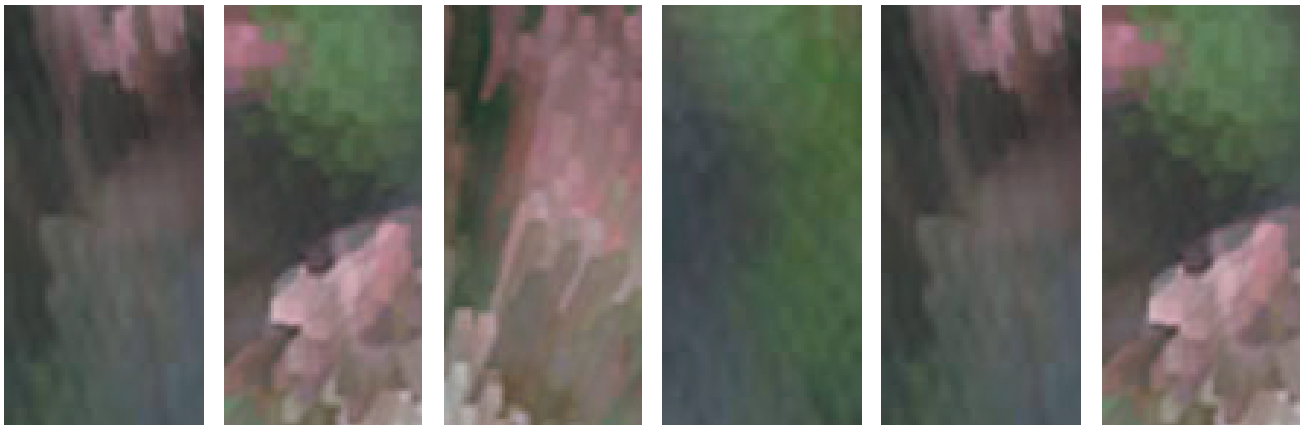


# PROCEEDINGS OF THE NSF WORKSHOP ON RELIABLE ENGINEERING COMPUTING

SEPTEMBER 15-17, 2004 | SAVANNAH, GEORGIA USA

**EDITORS**
Rafi L. Muhanna
Robert L. Mullen

### EDITORS

Rafi L. Muhanna, Georgia Institute of Technology
Robert L. Mullen, Case Western Reserve University

### WORKSHOP SPONSORS

National Science Foundation
CAST Division of AIChE
Society for Risk Analysis
Sun Microsystems
Georgia Institute of Technology

### HONORARY WORKSHOP CO-CHAIRS

Ramon E. Moore
Eldon R. Hansen

### WORKSHOP CHAIR

Rafi L. Muhanna, Georgia Institute of Technology

### WORKSHOP CO-CHAIR

Robert L. Mullen, Case Western Reserve University

### WORKSHOP SCIENTIFIC COMMITTEE

Götz Alefeld, University of Karlsruhe, Germany
Daniel Berleant, Iowa State University, IA, USA
David Bogle, University College London, UK
George Corliss, Marquette University, WI, USA
William Edmonson, University of Hampton, VA, USA
Scott Ferson, Applied Biomathematics, NY, USA
Roger Ghanem, Johns Hopkins University, MD, USA
Raphael Haftka, University of Florida, FL, USA
Baker Kearfott, University of Louisiana at Lafayette, LA, USA
Vladik Kreinovich, University of Texas at El Paso, TX, USA
Zissimos Mourelatos, Oakland University, MI, USA
P.S.V. Nataraj, Indian Inst. of Technology, India
Arnold Neumaier, University of Vienna, Austria
Efstratios Nikolaidis, University of Toledo, OH, USA
Mark Stadtherr, University of Notre Dame,IN, USA
William Walster, Sun Microsystems, CA, USA
Steve Wojtkiewicz, Sandia National Laboratories, USA

### LOCAL WORKSHOP ORGANIZING COMMITTEE

Kimberly Gaither
Natalie Cosner
Jillison Parks
Patricia Potter
Hao Zhang

# **PROCEEDINGS**: TABLE OF CONTENTS

## NSF WORKSHOP ON RELIABLE ENGINEERING COMPUTING

## NSF WORKSHOP ON RELIABLE ENGINEERING COMPUTING

These proceedings embody the papers presented at the first workshop hosted by The Center for Reliable Engineering Computing at the Georgia Institute of Technology. Both this workshop and the activities of the Center focus on emerging technologies for reliable engineering analysis and design. Reliable engineering computing, as we understand it, requires that computing systems accommodate several sources of uncertainty and errors with a focus on self-validating methods. In the case of a mechanical system model, uncertainties can originate from:

1) The appropriateness of the mathematical model to describe the physical system;
2) The discretization of mathematical model into a computational framework;
3) The inexact knowledge of input parameters of a problem; and
4) Errors introduced by the nature of computer finite arithmetic.

A reliable engineering analysis must include all of the above in providing both solutions and measures of the reliability of the results provided. To date, no all-encompassing framework exists for reliable engineering computing. This field will only progress through meetings such as this; a meeting that addresses different aspects of reliable engineering computing in both analysis and design. The participants represent a truly interdisciplinary group including mathematicians, computer scientists and engineers from a wide distribution of engineering disciplines. The participants are from academia, research institutions and industry and include both national and international experts.

The sponsors of this workshop are:

- National Science Foundation
- CAST Division of AIChE
- Society for Risk Analysis
- Sun Microsystems
- Georgia Institute of Technology

The organizers appreciate the support of the sponsors: this workshop would not have occurred without their contributions and commitment.

Rafi L. Muhanna
Robert L. Mullen
Editors

# INTRODUCTION

## Introductory Remarks on Reliable Engineering Computing

Ramon E. Moore (rmoore17@columbus.rr.com)
*Worthington, Ohio, USA*

"In a physical system model, uncertainties can originate from several sources"(Rafi Muhanna and Robert Mullen):

### 1. "The appropriateness of the mathematical model to describe the physical system"

Examples of the use of interval methods in this connection:

1)A physics problem at Lockheed (ca. 1960): Q: Is the strange behavior of the computer model due to round-off errors? A: After converting the program to run in interval arithmetic with outward rounding, it was determined that round-off error was very small in the original program. Result: the physicist took another look at the model equations and found that there was a missing term.

2)A long controversy between research groups at MIT and Cal Tech concerned whether the observed behavior of computer simulations was due to roundoff error or defects in the mathematical model, in the case of computer solutions of the Birkhoff-Rota complex partial differential/integral equations modeling the onset of turbulence in wind-shears. My graduate student Jeffrey Ely wrote a program for variable-precision interval arithmetic with outward rounding, and finally by using about 300 decimal place (nearly 1000 bits) in outwardly rounded interval arithmetic, was able to settle the controversy. It was NOT roundoff error, but the model itself. The model realistically determined the onset of turbulence at a reproduceable, finite time after the initial appearance of the wind-shear. I have always found it odd to suppose that anything we want to compute can be done carrying only some fixed number of digits or bits. Is 40 bits enough? 80 ? A thousand? It depends on what we are trying to compute. [J. Ely and G. R. Baker. High precision calculations of vortex sheet motion. J. Comp. Phys., 111:275-282, 1994].

The point of these examples is that outwardly rounded interval arithmetic automatically bounds roundoff error in any computation. As a consequence, if the interval results are adequately narrow, then no repetition is needed using higher precision arithmetic, carrying more digits. If the results are not narrow enough, then the computations can be repeated carrying more digits. If roundoff is the only source of error, we may eventually obtain

satisfactorily narrow interval results containing the corresponding infinite precision results that would come from using exact real arithmetic. Floating-point arithmetic by itself cannot provide such answers.

More important is the fact that interval computational methods can answer questions about the "appropriateness of the mathematical model" even when roundoff is not the only nor the main source of computational error in simulating the behavior of a proposed mathematical model on a computer. I will expand upon this in the following discussions of items 2,3, and 4 listed by the organizer of this workshop as origins of uncertainties in physical system models:

## 2. "The discretization of the mathematical model into a computational framework"

If there is an analytic expression for the discretization error, for example the mean value form of the remainder in a truncated Taylor series approximation, that too can be bounded by interval computation. In addition to that, a finite element method or a finite difference method may be looked at as a computational model of a physical process or structure, and interval methods can provide intervals containing the exact behavior of the computational model. In this way we test for suitable computational parameters (such as mesh size or the number of nodes) for which the model produces results in agreement with observed behavior of the real process or characteristics of the the real physical structure.

## 3. "The inexact knowledge of input parameters of a problem"

Suppose we have measured that input parameters fall within certain upper and lower limits, then we can use interval inputs for them, and interval computation is designed for just that sort of thing.

For example, if we measure a width as $w = 7.2 \pm 0.1$, length as $l = 14.9 \pm 0.1$, and height as $h = 405.6 \pm 0.2$, then the volume $V = w \times l \times h$ is within in the interval

$([7.1, 7.3] \times [14.8, 15.0]) \times [405.4, 405.8] = [42599.432, 44435.1] = 43517.266 \pm 917.834$ If desired, we could use outward rounding to retain containment with numbers having only one digit after the decimal point, and find that the volume is contained in the interval of numbers $[42599.4, 44435.1] = 43517.25 \pm 917.85$. This in turn is contained in the interval $[42599.4, 44435.2] = 43517.3 \pm 917.9$.

All I am saying in the above example is that IF all we know about $w$, $l$ and $h$ is that their values lie in the intervals given, THEN all we can say about $V$ is that it is in the interval obtained by the appropriate multiplications of the given endpoints of the input variables $w$, $l$ and $h$. If we need $V$ more accurately, then we have to measure $w$, $l$ and $h$ more accurately.

There are successful interval algorithms for doing this sort of thing for real engineering computations involving uncertain values of parameters, if those uncertain values are at least known to lie in certain intervals. See for example papers of Muhanna and Mullen; Stadtherr

et al; G. Fichtner, H. J. Reinhart, and D. W. T. Rippin;R. P. Broadwater, H. E. Shaalan and W. J. Fabrycky; C. H. Dou, W. Woldt, I. Bogardi, and M. Dahab; M. Hurme, M. Dohnal, and M. Jaervalaeinen; X. Lin, O. T. Melo, D. R. Hastie, et al; H. U. Koyluoglu, A. S. Cakmak, and S. R. K. Neilson; H. E. Shaalan and R. P. Broadwater; and many others, see e.g. http://www.cs.utep.edu/interval-comp/abstracts/list.html A recent noteworthy book on interval methods for global optimization is: Global Optimization Using Interval Analysis, E. Hansen and G. William Walster, published by Marcel Dekker, 2004. More generally, interval arithmetic, which is very simple, is incorporated as a tool in more sophisticated methods for analyzing uncertainty in engineering design, in which more information is available than simple upper and lower bounds on uncertainties of inputs. For example some recent works of Erik Antonsson and others use the "level interval algorithm", which is itself used internally by their "Method of Imprecision". There are frequent uses of interval arithmetic as a tool in such fuzzy systems analyses, see for example the special issue on Interfaces between Fuzzy Set Theory and Interval Analysis, of the journal Fuzzy Sets and Systems, Vol 135, No 1, April 2003. See also two special issues on Interval Analysis and Fuzzy Sets of the journal Reliable Computing, Vol 10, No. 4, 2004 and Vol. 10, No. 5 (to appear).

Interval arithmetic is also coming into use in probabilistic handling of uncertainty, see for example the two special issues of the journal Reliable Computing devoted to Dependable Reasoning about Uncertainty, guest-edited by D. Berleant, Vol 9, No.6, (Dec. 2003) and Vol 10, No. 2 (April 2004); and D. Berleant and J. Zhang, Representation and problem solving with the distribution envelope determination (DEnv) method, Reliability Engineering and System Safety, in press. By using step-function interval envelopes around probability density functions, we can sometimes compute useful envelopes for cumulative distributions without using costly Monte Carlo methods.

## 4. "Errors introduced by the nature of computer finite arithmetic"

Interval arithmetic with outward rounding (lower bounds rounded to the left on the number line and upper bounds rounded to the right) yields computer results which contain both the unknown infinite-precision real arithmetic results as well as the results which would be obtained by ordinary floating-point arithmetic for the same sequence of computer operations.

## References


1. Antonsson, E. K. and Otto, K.,"Improving Engineering Design with Fuzzy Sets" Chapter 38 in FUZZY INFORMATION ENGINEERING: A GUIDED TOUR OF APPLICATIONS, Edited by: D. Dubois, H. Prade, and R. R. Yager PUBLISHED BY: John Wiley & Sons, New York (712 p.) 1997
2. Broadwater, R. P., Shaalan, H. E. and Fabrycky, W. J., "Decision Evaluation with Interval Mathematics: A Power Distribution System Case Study", IEEE Transactions on Power Delivery, 9:59-65, 1994.
3. Dong, W. M., Chiang, W. L. and Wong, F. S. "Propagation of uncertainties in deterministic systems." Computers & Structures., 26(3):415-423, 1987.

4. Dou, C. H., Woldt, W., Bogardi, I. and Dahab, M., "Steady state groundwater flow simulation with imprecise parameters", Water Resources Research., 31(11):2709-2719, 1995.
5. Fichtner, G. Reinhart, H. J. and Rippin, D. W. T. "Design of Flexible Chemical Plants by the Application of Interval Mathematics". Computers and Chemical Engineering, 14:1311-1316, 1990.
6. Gau, C.-Y. and Stadtherr, M. A.,"New Interval Methodologies for Reliable Chemical Process Modeling" Comput. Chem. Eng., 26, 827-840 (2002).
7. Hurme, M., Dohnal, M. and Jaervalaeinen, M., "Qualitative Reasoning in Chemical and Safety Engineering", Computers and Chemical Engineering, 17:441-446, 1993.
8. Koyluoglu, H. U., Cakmak, A. S. and Neilson, S. R. K., "Interval algebra to deal with pattern loading and structural uncertainty". Journal of Engineering Mechanics, 121(11):1149-1157, 1995.
9. Kutscher, S. and Schulze, J., "Some Aspects of Uncertain Modelling Experiences in Applying Interval Mathematics to Practical Problems", In H. Bandemer, editor, Modelling Uncertain Data, pages 62-68, Akademie Verlag, Berlin, 1993.
10. Lin, X., Melo, O. T. Hastie, D. R. et al., "Case study of ozone production in a rural area of central ontario" Atmos. Environ., 26A(2):311-324, 1992.
11. Lin, Y. and Stadtherr, M. A.,"Advances in Interval Methods for Deterministic Global Optimization in Chemical Engineering" J. Global Optimization, in press (2003).
12. Moore, R. E., Methods and Applications of Interval Analysis, SIAM, Philadelphia, 1979.
13. Muhanna, R. L. and Mullen, R. L., "Uncertainty in Mechanics Problems - Interval-Based Approach," Journal of Engineering Mechanics, ASCE, Vol. 127, No. 6, pp 557-566, 2001.
14. Muhanna, R. L. and Mullen, R. L., Interval Methods for Reliable Computing, Chapter in Reliability Design Handbook, CRC Press LLC, ISBN 0-8493-1180-2, edited by Efsratios Nikolaidis and Dan Ghiocel, to be published February 2004.
15. Muhanna, R. L. and Mullen, R. L., "Fundamental Concepts for Interval-based Finite Element Formulations," Journal of Reliable Computing, Kluwer Academic Publishers, 2004, (tentatively accepted)
16. Mullen, R. L. and Muhanna, R. L., "Efficient Interval Methods for Finite Element Solution," High Performance Computing Systems and Applications, IEEE, editors: Almhana J. N. and Bhavsar V. C., 2002.
17. Shaalan, H. E. and Broadwater, R. P., "Using Interval Mathematics in Cost Benefit Analysis of Distribution Automation", Electric Power Systems Research, 27:145-152, 1993.
18. Muhanna, R. L. and Mullen, R. L."Development of Interval Based Methods for Fuzziness in Continuum Mechanics." Proceedings of ISUMA-NAFIPS'95, IEEE, pp 705-710, 1995.
19. Muhanna, R. L. and Mullen, R. L. "Formulation of fuzzy finite element methods for mechanics problems." Computer-Aided Civil and Infrastructure Engineering (previously Microcomputers in Civil Engineering), 14, 107-117, 1999.

# A method for outer interval solution of parametrized systems of linear equations

I. Skalna (skalna@galaxy.uci.agh.edu.pl)
*Department of Applied Informatics, University of Science and Technology, ul. Gramatyka 10, 30-067 Krakow,POLAND*

**Abstract.** Consider the systems of linear interval equations whose coefficients are affine-linear functions of interval parameters. Such systems, called parametrized systems of linear interval equations, are encountered in many practical problems, e.g in structure mechanics. A direct method for computing a tight enclosure for the solution set is proposed in this paper. It is proved that for systems with real matrix and interval right-hand vector the method generates the hull of the solution set. For such systems an explicit formula for the hull is also given. Finally some numerical examples are provided to demonstrate the usefulness of the method in structure mechanics.

**Keywords:** parametrization, intervals, linear equations, truss structures

## 1. Introduction

A system of linear interval equations

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

with coefficient matrix $[A] \in \mathbb{IR}^{n\times n}$ and right-hand vector $[b] \in \mathbb{IR}^n$ is defined as a family of linear equations

$$Ax = b,\ (A \in [A],\, b \in [b]). \tag{2}$$

The solution set of (1) is given by

$$\sum\Big([A],\,[b]\Big) = \{x \,|\, Ax = b,\ A \in [A],\, b \in [b]\}. \tag{3}$$

When computing inner and outer bounds for the solution set (3) it is implicitly assumed $A$ and $b$ to vary independently within $[A]$ and $[b]$. In practice there might be further constraints on matrices within $[A]$ and $[b]$. Taking into account these contraints leads to the parametrized systems of linear interval equations. Consider the family of linear algebraic systems of the following type

$$A(p)x = b(p), \tag{4}$$

with

$$A_{ij}(p) = \omega(i,\, j)^{\mathrm{T}} p \tag{5a}$$

$$b_j(p) = \omega(0,\, j)^{\mathrm{T}} p \tag{5b}$$

and $p \in [p] \in \mathbb{IR}^k$ [6]. Such systems are encountered in many practical applications, e.g. in structure mechanics [3], [7].

The family of systems (4) is usually written in the form

$$A([p])x = b([p]) \tag{6}$$

and is called parametrized system of linear interval equations.

The (united) solution set of the system (6) is defined as

$$\sum \Big(A([p]),\, b([p])\Big) = \{\, x \,|\, A(p)x = b(p),\, p \in [p]\} \tag{7}$$

If the solution set is bounded then the interval hull for it exists. In order to guarantee that the solution set is bounded matrix $A([p])$ must be regular (for all $p \in [p]$ $A(p)$ is regular). In practice it is usually required that the matrix $A([p])$ is an H-matrix.

In this paper a direct method for computing a tight enclosure for (7) is proposed. The method is based on the following inclusion

$$\diamondsuit\Big(\sum (A([p]),\, b([p]))\Big) \subseteq \tilde{x} + \langle [D] \rangle |Z| [-1,\, 1] \tag{8}$$

where

$$[Z]_i = \sum_{j=1}^{n} R_{ij} \Big(\omega(0,\, j) - \sum_{k=1}^{n} \tilde{x}_k \omega(j,\, k)\Big)^{\mathrm{T}} [p], \tag{9}$$

$$[D]_{ij} = \Big(\sum_{k=1}^{n} R_{ik} \omega(k,\, j)\Big)^{\mathrm{T}} [p], \tag{10}$$

$R = \mathrm{mid}\Big(A([p])\Big)^{-1}$ and $\tilde{x} = R\check{b}$.

It is proved that for systems with $\mathrm{rad}\Big(A([p])\Big) = 0$ the inclusion in (8) is an equality. For such systems an explicit formula for the hull of the solution set (7) is also given.

Finally some numerical examples of truss structures are provided to demonstrate the usefulness of the method in structure mechanics.

## 2. Basic notion

By $\mathbb{IR}$, $\mathbb{IR}^n$, $\mathbb{IR}^{n\times n}$ denote the set of real compact intervals, respectively interval vectors with $n$ components and the set of interval $n \times n$ matrices.

For interval $[a] = [\underline{a},\, \overline{a}] = \{x \,|\, \underline{a} \le x \le \overline{a}\}$ define the midpoint

$$\check{a} = \mathrm{mid}([a]) = (\underline{a} + \overline{a})/2$$

the radius

$$\mathrm{rad}([a]) = (\overline{a} - \underline{a})/2$$

and minimal absolute value (mignitude)

$$\langle [x] \rangle = \mathrm{mig}([x]) = \min\{|x| \mid x \in [x]\}.$$

An interval matrix $[A] \in \mathbb{IR}^{n\times n}$ is interpreted as a set of real $n \times n$ matrices

$$[A] = \{A \in \mathbb{R}^{n\times n} \,|\, A_{ij} \in [A]_{ij},\, i,j = 1,\, \ldots,\, n\,\}$$

An $n \times 1$ matrix is just an interval vector. In analogy to one-dimensional case certain real matrices are related to each interval matrix. Middle matrix $\mathrm{mid}([A])$ and the radius $\mathrm{rad}([A])$ are computed componentwise. For square interval matrices an Ostrowsky matrix $\langle [A] \rangle$ is defined with entries

$$\langle [A] \rangle_{ij} = \mathrm{mig}([A]_{ij}),\, i \neq j$$
$$\langle [A] \rangle_{ij} = -|[A]_{ij}|,\, i = j.$$

A square matrix $[A] \in \mathbb{IR}^{n\times n}$ is called regular if all $A \in [A]$ are nonsingular. If $\check{A}[A]$ is regular then $[A]$ is strongly regular.

An interval matrix $[A]$ is an H-matrix iff there exist a vector $u > 0$ such that

$$\langle [A] \rangle u > 0.$$

If $S$ is a bounded set of real matrices then $\inf S$ and $\sup S$ exist, and the *hull* of S,

$$\diamondsuit S = [\inf S,\, \sup S] = \diamondsuit S = \bigcap \{[Y] \mid [Y] \in \mathbb{IR},\ [Y] \supseteq S\}$$

is the tightest interval matrix enclosing $S$.

## 3. Minimal enclosure

In case of parametrized systems with real matrices, $\mathrm{rad}\Big(A([p])\Big) = 0$, the hull of the solution set (7) is given by an explicit formula.

THEOREM 1. *Let* $A([p])x = b([p])$, $[p] \in \mathbb{IR}^k$ $R = mid\Big(A([p])\Big)$ *and* $\tilde{x} = R \cdot mid\Big(b([p])\Big)$. *If* $rad\Big(A([p])\Big) = 0$ *then*

$$\diamondsuit\Big(\sum (A([p]),\, b([p]))\Big) = \tilde{x} + [Z]',$$

*where*

$$[Z]'_i = \sum_{j=1}^{n} \Big(R_{ij} \cdot \omega(0,\, j)\Big)^T [-rad([p]),\, rad([p])]. \tag{11}$$

*Proof.* Since $\mathrm{rad}([A]) = 0$, hence $A([p]) = A$, $\check{A} = A$, $R = A^{-1}$ $(RA = I)$ and $\tilde{x} = A^{-1}\check{b}$ $(A\tilde{x} = \check{b})$. Then one has

$$\begin{aligned}
\diamondsuit\Big(\sum (A([p]),\, b([p]))\Big) &= \diamondsuit\Big(\sum (A,\, b([p]))\Big) = \\
&= \tilde{x} + \diamondsuit\Big(\sum (A,\, b([p]) - A\tilde{x})\Big) = \\
&= \tilde{x} + \diamondsuit\Big(\sum (A,\, b([p]) - \check{b})\Big) = \\
&= \tilde{x} + \diamondsuit\Big(\{R(b(p) - \check{b}),\ p \in [p]\}\Big).
\end{aligned}$$

$$\diamondsuit\Big(\{R(b(p)-\check{b},\ p\in[p]\}\Big)_i = \diamondsuit\Big\{\sum_{j=1}^{n} R_{ij}(b(p)-\check{b})_j,\ p\in[p]\Big\} =$$

$$= \diamondsuit\Big\{\sum_{j=1}^{n} R_{ij}(\omega(0,\,j)^{\mathrm{T}}\cdot p-\omega(0,\,j)^{\mathrm{T}}\cdot\check{p}),\ p\in[p]\Big\} =$$

$$= \diamondsuit\Big\{\sum_{j=1}^{n} (R_{ij}\cdot\omega(0,\,j))^{\mathrm{T}}(p-\check{p}),\ p\in[p]\Big\} =$$

$$= \Big(\sum_{j=1}^{n} (R_{ij}\cdot\omega(0,\,j))^{\mathrm{T}}\Big)([p]-\check{p}) =$$

$$= \Big(\sum_{j=1}^{n} (R_{ij}\cdot\omega(0,\,j))^{\mathrm{T}}\Big)[-\mathrm{rad}([p]),\,\mathrm{rad}([p])].$$

The equality before the last one holds since every component $p_i$ occurs at most once in the preceding expression. □

## 4. Main result

Most of the methods for enclosing the solution set of parametized systems of equations are iterative [1], [2], [5], [6]. However, each iteration enlarges the enclosure because of the roundings has to be made in arithmetic operations. The method based on the formula (8) has polynomial complexity and computes the enclosure of the solution set (7) in one step, and hence has a great advantage over the iterative methods. In what follows the theoretical background for the method is presented.

THEOREM 2 (Neumaier [4]). *Let $[A]\in\mathbb{IR}^{n\times n}$. If $[A]$ is an H-matrix then for all $[b]\in\mathbb{IR}^n$ holds*

$$\diamondsuit\sum\Big([A],\,[b]\Big)\subseteq\langle[A]\rangle^{-1}[b][-1,\,1].$$

THEOREM 3. *Let $A([p])x=b([p])$ with $[p]\in\mathrm{IR}^k$, $R\in\mathbb{R}^{n\times n}$, and $\tilde{x}\in\mathbb{R}^n$. If $[D]$ given by formula (10) is an H-matrix then*

$$\diamondsuit\Big(\sum\Big(A([p]),\,b([p])\Big)\Big)\subseteq\tilde{x}+\langle[D]\rangle^{-1}|[Z]|[-1,\,1],\tag{12}$$

*where $[Z]$ is defined by formula (9).*

*Proof.* Vector $x\in\sum(A([p]),\,b([p]))$ iff there exists such $p\in[p]$ that $A(p)x=b(p)$. Since $[D]$ is an H-matrix then both sides of this equality can be multipled by $A(p)^{-1}$. Hence

$$x=A(p)^{-1}b(p)=\tilde{x}+A(p)^{-1}(b(p)-A(p)\tilde{x})=$$
$$=\tilde{x}+(R\cdot A(p))^{-1}(R(b(p)-A(p)\tilde{x})).$$

Since $R \cdot A(p) \in [D]$, $R(b(p) - A(p)\tilde{x}) \in [Z]$ then the following relation holds

$$(R \cdot A(p))^{-1}(R(b(p) - A(p)\tilde{x})) \in \diamondsuit\Big(\sum([D],\, [Z])\Big),$$

and hence

$$x \in \tilde{x} + \diamondsuit\Big(\sum([D],\, [Z])\Big). \tag{13}$$

Matrix $[D]$ is an H-matrix then by theorem 2

$$\diamondsuit\Big(\sum([D],\, [Z])\Big) \subseteq \langle [D] \rangle^{-1} |[Z]|[-1,\, 1]. \tag{14}$$

Equations (13) and (14) gives the thesis of the theorem. □

It is recomended to choose

$$R = \mathrm{mid}(A([p]))^{-1}$$

and

$$\tilde{x} = \mathrm{mid}(A([p]))^{-1} \cdot \mathrm{mid}(b([p]))$$

so that $[D]$ and $[Z]$ are of small norms (see theorem 4.1.10 [4]).

THEOREM 4. *Let $A([p])x = b([p])$, $[p] \in IR^k$ and $R = \check{A}^{-1}$, $\tilde{x} = R\check{b}.i$ If rad$(A([p])) = 0$ then*

$$\diamondsuit\Big(\sum (A([p]),\, b([p]))\Big) = \tilde{x} + \langle [D] \rangle^{-1} |[Z]|[-1,\, 1].$$

*where $[D]$ and $[Z]$ are given respectively by formula (10) and (9).*

*Proof.* To prove the theorem it suffices to show that

$$\tilde{x} + \langle [D] \rangle^{-1} |[Z]|[-1,\, 1] = \tilde{x} + [Z]'$$

where $[Z]'$ is given by (11).

Since $\mathrm{rad}(A([p])) = 0$, hence $R = \check{A}^{-1} = A^{-1}$ and then matrix $[D]$ and vector $[Z]$ takes the simpler form. $[D] = I$ and

$$[Z]_i = \diamondsuit\{R(b(p) - A\tilde{x}),\, p \in [p]\}_i = \diamondsuit\{R(b(p) - \check{b}),\, p \in [p]\}_i = [Z]'_i.$$

Hence

$$\tilde{x} + \langle [D] \rangle^{-1} |[Z]|[-1,\, 1] = \tilde{x} + |Z'|[-1,\, 1]. \tag{15}$$

Let now $\alpha = \sum_{j=1}^{n} (R_{ij} \cdot \omega(0,\, j))$. Then by symmetry of the interval $[-\mathrm{rad}([p]),\, \mathrm{rad}([p])]$ holds

$$\{|Z|'[-1,\, 1]\}_i = |\sum_{k=1}^{n} \alpha_k [-\mathrm{rad}([p]_k),\, \mathrm{rad}([p]_k)]|[-1,\, 1] =$$

$$|[-\sum_{k=1}^{n} \alpha_k \mathrm{rad}([p]_k),\, \sum_{k=1}^{n} \alpha_k \mathrm{rad}([p]_k)]|[-1,\, 1] =$$

$$|\sum_{k=1}^{n}\alpha_k \mathrm{rad}([p]_k)|[-1,\,1] = [-\sum_{k=1}^{n}\alpha_k \mathrm{rad}([p]_k),\ \sum_{k=1}^{n}\alpha_k \mathrm{rad}([p]_k)] =$$

$$\sum_{k=1}^{n}\alpha_k[-\mathrm{rad}([p]_k),\ \mathrm{rad}([p]_k)] = [Z]_i'$$

This and equation (15) gives the thesis of the theorem. □

Table I. Algorithm

$R := \mathrm{mid}(A([p]))^{-1}$;

$\tilde{x} := R \cdot \mathrm{mid}(b([p]))$;

$$[Z]_i = \sum_{j=1}^{n} R_{ij}\Big(\omega(0,\,j) - \sum_{k=1}^{n}\tilde{x}_k\omega(j,\,k)\Big)^{\mathrm{T}}[p]$$

$$[D]_{ij} := \Big(\sum_{\nu=1}^{n} R_{i\nu}\omega(\nu,\,j)\Big)^{\mathrm{T}}[p];$$

$\mathrm{outer} := \tilde{x} + [-1,\,1]\langle[D]\rangle^{-1}|[Z]|$

## 5. Examples

**Example 1.** Baltimore bridge (1870).
For the plane truss structure (all bars, loads and displacements are in the same x-y plane) shown in Figure 1 subjected to downward forces of 80 [kN] at node Nº 11, 120 [kN] at node Nº 12 and 80 [kN] at node Nº 15, the displacements of the nodes are computed. Young's modulus E= $2.1 \times 10^{11}$ [Pa] and cross-section area A= 0.004 [$m^2$]. The lenghts of the bar elements are shown in the figure (unity equals 1 [m]). Assume the stiffness of some of the bar elements (denoted in the figure with thick lines) to be uncertain by ±5%. To compute the displacement, the parametrized system of linear interval equations must be sovled. The results are in table II and III.

**Example 2.** Plane truss with uncertain stiffness of 8 bar elements.

The plane truss shown in Figure 2 is subjected to downward forces of 30 [kN] at nodes Nº 2, 3 and 4. All bar elements have the same Young's modulus E=$7 \cdot 10^{10}$ and cross-section area A=0.003 [$m^2$]. Assume the stiffness of 8 bar elements to be uncertain by ±5%. The resultimg intevals vectors are presented is tables IV and V.

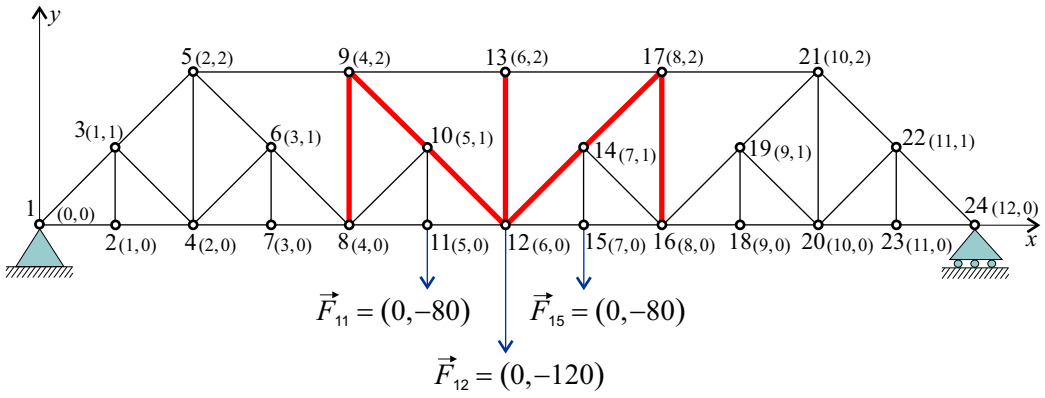



*Figure 1.* Scheme of the Baltimore bridge

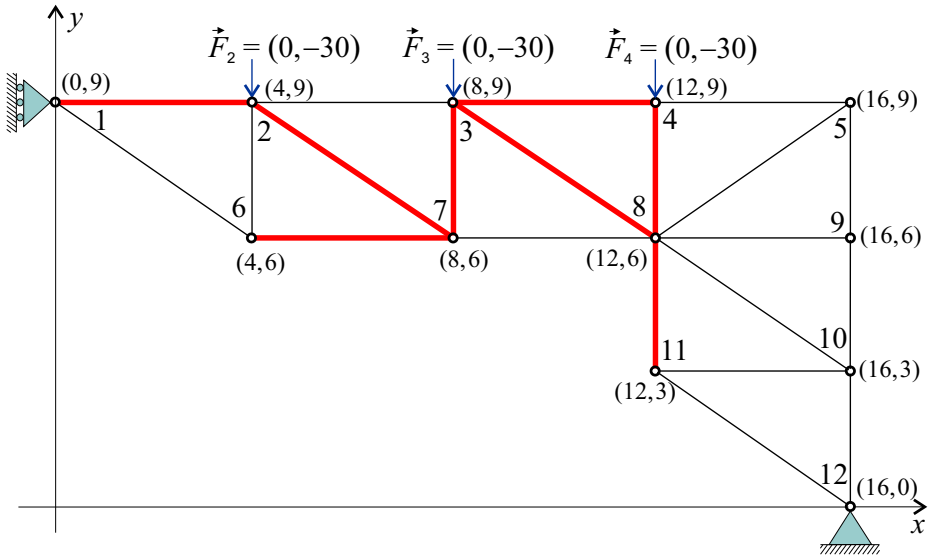



*Figure 2.* Truss structure with uncertain stiffnesses of 8 bar elements

**Example 3.** Plane truss with uncertain displacements of the supports.

The plane truss shown in Figure 3 has two supports: partial (sliding) support along y axis at node № 1 and full support at node № 12. Allow the movements of the supports at node 1 by $\Delta_{1-2} = 0.2$ [m] along y axis, at node 12 by 0.3 [m] along x axis and by 0.4 [m] along y axis. Now assume all movements to be uncertain by $\pm 5\%$ Uncetrain displacements of the support cause interval parameters appear only in the right-hand vector. In this case the method should give the hull. The results in tables VI and VII prove this to be true.

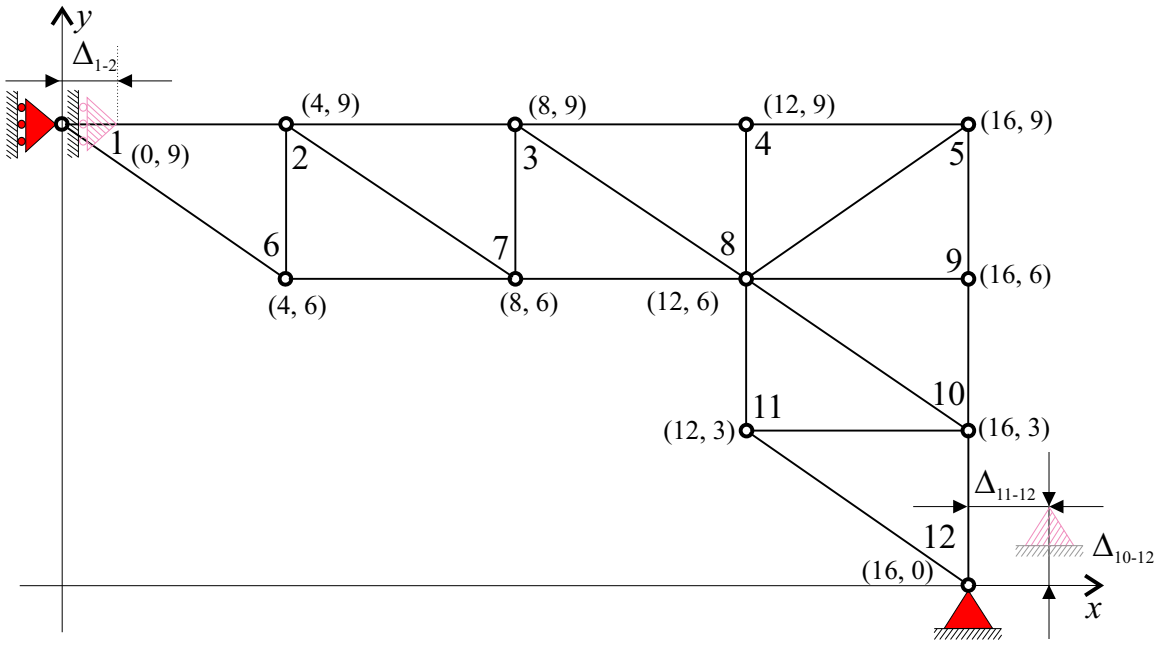



*Figure 3.* Truss structure with uncertain displacements of the supports

## 6. Results

The results produced by the method described in section 4 are presented in tables below. Column № 2 contains exact solution of non-interval system, comulmn № 3 contains the inner estimation obtained using the method of random sampling of parameter intervals (RSPI), column № 4 contains the results of the proposed method. Columns № 3 and 5 contain the relative error of the resulting intervals (in percent).

Table II. Example 1 ($x$-coords.)

| | $\mathbf{d}_0$ [$\times 10^{-5}$] | RSPI [$\times 10^{-5}$] | r[$*$]/$\mathbf{d}_0$ [%] | Method [$\times 10^{-5}$] | r[$*$]/$\mathbf{d}_0$ [%] |
|---|---|---|---|---|---|
| $d_2^x$ | 16.67 | 16.67 | 0 | 16.67 | 0 |
| $d_3^x$ | 190.24 | [189.72, 190.76] | 0.3 | [189.34, 191.13] | 0.5 |
| $d_4^x$ | 33.33 | 33.33 | 0 | 33.33 | 0 |
| $d_5^x$ | 300 | [298.96, 301.05] | 0.3 | [298.21, 301.79] | 0.6 |
| $d_6^x$ | 190.24 | [189.72, 190.76] | 0.3 | [189.34, 191.13] | 0.5 |
| $d_7^x$ | 50 | 50 | 0 | 50 | 0 |
| $d_8^x$ | 66.67 | 66.67 | 0 | 66.67 | 0 |
| $d_9^x$ | 233.33 | [232.29, 234.38] | 0.4 | [231.54, 235.12] | 0.8 |
| $d_{10}^x$ | 172.01 | [170.6, 173.38] | 0.8 | [169.84, 174.17] | 1.3 |
| $d_{11}^x$ | 104.76 | 104.76 | 0 | 104.76 | 0 |
| $d_{12}^x$ | 142.86 | 142.86 | 0 | 142.86 | 0 |
| $d_{13}^x$ | 142.86 | [141.81, 143.9] | 0.7 | [141.07, 144.65] | 1.3 |
| $d_{14}^x$ | 113.71 | [112.38, 114.93] | 1.1 | [111.54, 115.88] | 1.9 |
| $d_{15}^x$ | 180.95 | 180.95 | 0 | 180.95 | 0 |
| $d_{16}^x$ | 219.05 | 219.05 | 0 | 219.05 | 0 |
| $d_{17}^x$ | 52.38 | [51.34, 53.43] | 2 | [50.59, 54.17] | 3.4 |
| $d_{18}^x$ | 235.71 | 235.71 | 0 | 235.71 | 0 |
| $d_{19}^x$ | 95.48 | [94.96, 96] | 0.5 | [94.58, 96.37] | 0.9 |
| $d_{20}^x$ | 252.38 | 252.38 | 0 | 252.38 | 0 |
| $d_{21}^x$ | $-14.29$ | $[-15.33, -13.24]$ | 7.3 | $[-16.08, -12.5]$ | 12.5 |
| $d_{22}^x$ | 95.48 | [94.96, 96] | 0.5 | [94.58, 96.37] | 0.9 |
| $d_{23}^x$ | 269.05 | 269.05 | 0 | 269.05 | 0 |
| $d_{24}^x$ | 285.71 | 285.71 | 0 | 285.71 | 0 |

Table III. Example 1 ($y$-coords.)

| | $\mathbf{d}_0$ $[\times 10^{-5}]$ | RSPI $[\times 10^{-5}]$ | r$[*]/\mathbf{d}_0$ [%] | Method $[\times 10^{-5}]$ | r$[*]/\mathbf{d}_0$ [%] |
|---|---|---|---|---|---|
| $d_2^y$ | −237.38 | [−237.9, −236.86] | 0.2 | [−238.27, −236.48] | 0.4 |
| $d_3^y$ | −237.38 | [−237.9, −236.86] | 0.2 | [−238.27, −236.48] | 0.4 |
| $d_4^y$ | −394.28 | [−395.33, −393.24] | 0.3 | [−396.07, −392.49] | 0.5 |
| $d_5^y$ | −394.28 | [−395.33, −393.24] | 0.3 | [−396.07, −392.49] | 0.5 |
| $d_6^y$ | −551.18 | [−552.75, −549.62] | 0.3 | [−553.87, −548.5] | 0.5 |
| $d_7^y$ | −551.18 | [−552.75, −549.62] | 0.3 | [−553.87, −548.5] | 0.5 |
| $d_8^y$ | −721.9 | [−723.99, −719.81] | 0.3 | [−725.47, −718.32] | 0.5 |
| $d_9^y$ | −745.7 | [−747.96, −743.69] | 0.3 | [−749.81, −741.6] | 0.5 |
| $d_{10}^y$ | −840.7 | [−842.68, −838.87] | 0.2 | [−843.9, −837.5] | 0.4 |
| $d_{11}^y$ | −850.23 | [−852.2, −848.39] | 0.2 | [−853.43, −847.03] | 0.4 |
| $d_{12}^y$ | −890.06 | [−893.29, −887.27] | 0.3 | [−895.42, −884.69] | 0.6 |
| $d_{13}^y$ | −890.06 | [−893.29, −887.27] | 0.3 | [−895.99, −884.12] | 0.7 |
| $d_{14}^y$ | −840.7 | [−842.64, −838.75] | 0.2 | [−843.9, −837.5] | 0.4 |
| $d_{15}^y$ | −850.23 | [−852.17, −848.27] | 0.2 | [−853.43, −847.03] | 0.4 |
| $d_{16}^y$ | −721.9 | [−723.98, −719.8] | 0.3 | [−725.47, −718.32] | 0.5 |
| $d_{17}^y$ | −745.7 | [−748.02, −743.3] | 0.3 | [−749.81, −741.6] | 0.6 |
| $d_{18}^y$ | −551.18 | [−552.75, −549.61] | 0.3 | [−553.87, −548.5] | 0.5 |
| $d_{19}^y$ | −551.18 | [−552.75, −549.61] | 0.3 | [−553.87, −548.5] | 0.5 |
| $d_{20}^y$ | −394.28 | [−395.32, −393.23] | 0.3 | [−396.07, −392.49] | 0.5 |
| $d_{21}^y$ | −394.28 | [−395.32, −393.23] | 0.3 | [−396.07, −392.49] | 0.5 |
| $d_{22}^y$ | −237.38 | [−237.9, −236.85] | 0.2 | [−238.27, −236.48] | 0.4 |
| $d_{23}^y$ | −237.38 | [−237.9, −236.85] | 0.2 | [−238.27, −236.48] | 0.4 |

Table IV. Example 2 (x-coords.)

| | $\mathbf{d}_0$ $[\times 10^{-5}]$ | RSPI $[\times 10^{-5}]$ | r[∗]/$\mathbf{d}_0$ [%] | Method $[\times 10^{-5}]$ | r[∗]/$\mathbf{d}_0$ [%] |
|---|---|---|---|---|---|
| $d_2^x$ | −152.38 | [−160.4, −145.13] | 5 | [−160.4, −144.36] | 5.3 |
| $d_3^x$ | −228.57 | [−236.59, −221.32] | 3.3 | [−236.59, −220.55] | 3.5 |
| $d_4^x$ | −152.38 | [−163.76, −141.34] | 7.4 | [−165.26, −139.51] | 8.5 |
| $d_5^x$ | −76.19 | [−87.56, −65.15] | 14.7 | [−89.06, −63.32] | 16.9 |
| $d_6^x$ | 427.38 | [419, 435.56] | 1.9 | [416.48, 438.28] | 2.5 |
| $d_7^x$ | 427.38 | [419, 435.56] | 1.9 | [417.52, 437.24] | 2.3 |
| $d_8^x$ | 351.19 | [342.81, 359.37] | 2.4 | [341.33, 361.05] | 2.8 |
| $d_9^x$ | 351.19 | [342.81, 359.37] | 2.4 | [341.33, 361.05] | 2.8 |
| $d_{10}^x$ | 267.86 | [262.19, 273.57] | 2.1 | [261, 274.7] | 2.5 |
| $d_{11}^x$ | 115.48 | [109.81, 121.19] | 4.9 | [108.63, 122.32] | 5.9 |

Table V. Example 2 (y-coords.)

| | $\mathbf{d}_0$ $[\times 10^{-4}]$ | RSPI $[\times 10^{-4}]$ | r[∗]/$\mathbf{d}_0$ [%] | Method $[\times 10^{-4}]$ | r[∗]/$\mathbf{d}_0$ [%] |
|---|---|---|---|---|---|
| $d_1^y$ | −308.25 | [−312.45, −304.18] | 1.3 | [−315.09, −301.41] | 2.2 |
| $d_2^y$ | −251.27 | [−255.54, −247.22] | 1.6 | [−258.08, −244.46] | 2.7 |
| $d_3^y$ | −149.84 | [−152.8, −147.16] | 1.9 | [−153.43, −146.25] | 2.4 |
| $d_4^y$ | −37.14 | [−37.97, −36.34] | 2.2 | [−38.19, −36.1] | 2.8 |
| $d_5^y$ | 4.29 | 4.29 | 0 | 4.29 | 0 |
| $d_6^y$ | −251.27 | [−255.53, −247.22] | 1.7 | [−258.08, −244.46] | 2.7 |
| $d_7^y$ | −154.13 | [−157.19, −151.38] | 1.9 | [−158.32, −149.93] | 2.7 |
| $d_8^y$ | −32.86 | [−33.48, −32.24] | 1.9 | [−33.6, −32.11] | 2.3 |
| $d_9^y$ | 0 | 0 | 0 | 0 | 0 |
| $d_{10}^y$ | −4.29 | −4.29 | 0 | −4.29 | 0 |
| $d_{11}^y$ | 24.29 | [−25.04, −23.52] | 3.1 | [−25.2, −23.37] | 3.8 |

Table VI. Example 3 (x-coords.)

| | $\mathbf{d}_0$ $[\times 10^{-3}]$ | Method (hull) $[\times 10^{-3}]$ | r[*]/$d_0$ [%] |
|---|---|---|---|
| $d_2^x$ | 20 | [19, 21] | 5 |
| $d_3^x$ | 20 | [19, 21] | 5 |
| $d_4^x$ | 20 | [19, 21] | 5 |
| $d_5^x$ | 20 | [19, 21] | 5 |
| $d_6^x$ | 13.33 | [11.67, 15] | 12.5 |
| $d_7^x$ | 13.33 | [11.67, 15] | 12.5 |
| $d_8^x$ | 13.33 | [11.67, 15] | 12.5 |
| $d_9^x$ | 13.33 | [11.67, 15] | 12.5 |
| $d_{10}^x$ | 6.67 | [4.33, 9] | 35 |
| $d_{11}^x$ | 6.67 | [4.33, 9] | 35 |

Table VII. Example 3 (y-coords.)

| | $\mathbf{d}_0$ $[\times 10^{-3}]$ | Method (hull) $[\times 10^{-3}]$ | r[*]/$d_0$ [%] |
|---|---|---|---|
| $d_1^y$ | 35.56 | [28.44, 42.67] | 20 |
| $d_2^y$ | 26.67 | [21.33, 32] | 20 |
| $d_3^y$ | 17.78 | [14.22, 21.33] | 20 |
| $d_4^y$ | 8.89 | [7.11, 10.67] | 20 |
| $d_5^y$ | 0 | 0 | – |
| $d_6^y$ | 26.67 | [21.33, 32] | 20 |
| $d_7^y$ | 17.78 | [14.22, 21.33] | 20 |
| $d_8^y$ | 8.89 | [7.11, 10.67] | 20 |
| $d_9^y$ | 0 | 0 | – |
| $d_{10}^y$ | 0 | 0 | – |
| $d_{11}^y$ | 8.89 | [7.11, 10.67] | 20 |

## 7. Conclusions

The problem of solving parametrized systems of linear interval equations is very important in practical applications. Well known classical methods, such as interval version of Gauss Elimination or Preconditioned Interval Gauss-Seidel iteration fail since they compute enclosure for the solution set (3) which is generally much larger then solution set (7). A direct method for solving paraterized systems of linear interval equations based on the inclusion (8) was proposed and checked to be usefull in structure mechanics. The method produced tight enclosure for the solutions set of parametrized systems for all exemplary truss structures.

## References


1. C. Jansson. Interval linear systems with symmetric matrices, skew-symmetric matrices and dependencies in the right hand side. *Computing*, 46(3):265–274, 1991.
2. L. V. Kolev. A method for outer interval solution of linear parametric systems. *Reliable Computing*, 10:227–239, 2004.
3. Zenon Kulpa, Andrzej Pownuk, and Iwona Skalna. Analysis of linear mechanical structures with uncertainties by means of interval methods. *Computer Assisted Mechanics and Engineering Sciences*, 5(4):443–477, 1998.
4. Arnold Neumaier. *Interval Methods for Systems of Equations*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, UK, 1990.
5. E. D. Popova. On the solution of parametrised linear systems. *Scientific Computing, Validated Numerics, Interval Methods*, pages 127–138, 2001.
6. S. M. Rump. Verification methods for dense and sparse systems of equations. In Jürgen Herzberger, editor, *Topics in validated computations: proceedings of IMACS-GAMM International Workshop on Validated Computation, Oldenburg, Germany, 30 August–3 September 1993*, volume 5 of *Studies in Computational Mathematics*, pages 63–135, Amsterdam, The Netherlands, 1994. Elsevier.
7. Iwona Skalna. Methods for solving systems of linear equations of structure mechanics with interval parameters. *Computer Assisted Mechanics and Engineering Sciences*, 10(3):281–293, 2003.

# A Search Algorithm for Calculating Validated Reliability Bounds

Fulvio Tonon
*Department of Geology and Geophysics, University of Utah, 135 South 1460 East, Room 719, Salt Lake City, UT 84112, USA, e-mail: tonon@chpc.utah.edu*

**Abstract.** The search algorithm presented allows the CDF of a dependent variable to be bounded with 100% confidence, and allows for a guaranteed evaluation of the error involved. These reliability bounds are often enough to make decisions, and require a minimal number of function calls. The procedure is not intrusive, i.e. it can be equally applied when the function is a complex computer model (black box). The proposed procedure can handle input information consisting of probabilistic, interval-valued, set-valued, or random-set-valued information, as well as any combination thereof. The function as well as the joint pdf of the input variables can be of any type.

## 1. Introduction

Determining validated bounds for the Cumulative Distribution Function (CDF) of a function of random variables has attracted the attention of many scholars and a recent literature review may be found in [8]. Moore [24] and Moore [22] were probably the first ones to use interval analysis [23] to this end.

For example, Berleant and co-workers developed Statool [4-8], a computer program for obtaining bounds on the distributions of sums, products, and various other functions of random variables where the dependency relationship of the random variables need not be specified. Ferson [13] developed RiskCalc with similar capabilities. Independently, Lodwick and Jamison [19] presented a method for estimating and validating the cumulative distribution of a function of random variables (independent or dependent).

Dubois and Prade [12] firstly indicated how Random Set Theory might be used to bound the Cumulative Distribution Function (CDF) of a sum of two random variables. Tonon *et al.* [32] and Tonon [31] generalized this idea to a provide verified bounds to the CDF of a general function $y = f(u)$ where $u$ is a generic random vector. Random Set Theory allowed the abovementioned procedures developed by different authors to be put in a rigorous light. They also showed that their procedure can be used equally well when some components of $u$ are described as random variables, some others as intervals or Cartesian products, and some others as random sets. Additionally, a procedure was introduced to calculate the CDF of a particular value, $y^*$, of $y$; this procedure is meant to be used in reliability analyses and yields verified bounds on the reliability of a system. The motivation behind this procedure is that these bounds are often enough to make a decision, do not suffer from the shortcomings of Monte Carlo methods [14, 31], and often requires far less function calls than Monte Carlo methods.

In this paper, the procedure to calculate the CDF of a particular value, $y^*$, is advanced by introducing a searching algorithm with the aim of reducing the number of function calls. This is accomplished in Section 3. Before doing that and in order to establish common terminology and connect with [31] , Section 2 briefly restates the general procedure for calculating bounds on the entire CDF of $y$.

Section 4 specializes the searching algorithm to reliability analyses and Section 5 presents an application to the reliability analysis of a beam.

## 2. The entire CDF of $y=f(u)$ must be calculated

Let $pro(u)$ be the joint probability mass function of a discrete vector of input parameters $u = (u_1,...,u_p)$. Without loss of generality, it is assumed that the $i$-th parameter, $u_i$, belongs to interval

$I_i$, which may be infinite or semi-infinite. Consequently, $u$ is constrained within a $p$-dimensional box $D = I_1 \times,...,\times I_p$, where $\times$ indicates Cartesian product. The procedure proposed [31, 32] consists of the following three steps.

### 2.1 Step 1

Let $\{A_j, j = 1,...,N\}$ be a partition of $D$ and set

$$m\left(A_j\right) = \sum_{u:u\in A_j} pro\left(u\right) \tag{1}$$

If $u$ is not discrete but continuous, $pro(u)$ is a joint probability density function (PDF) and Equation (1) becomes

$$m\left(A_j\right) = \int_{A_j} pro\left(u\right) du \tag{2}$$

Since this procedure is based on Random Set Theory, subsets $A_j$ are called focal elements [31, 32]).

### 2.2 Step 2

Calculate the image $f(A_j)$ of each set $A_j$ through function $f$. In general, this problem can be solved by applying twice the techniques of global optimization (e.g. [17, 28, 29, 33]).

However, if one divides the $i$-th interval $I_i$ into $n_i$ subintervals, then $D$ is partitioned into $N = \prod_{i=1}^{p} n_i$ $p$-dimensional boxes $A_j$ obtained as Cartesian products of $p$ intervals (one per variable). As a consequence, $A_j$ has $2^p$ vertices, which we indicate as $v_k$, $k$=1,..., $2^p$. In this case, each parameter $u_i$ varies in an interval $L_i=[{}_L L_i, {}_R L_i]$, and the methods of Interval Analysis can be used to efficiently calculate $f(A_j)$. These methods are continuously improving, and the reader is referred to the web page (http://cs.utep.edu/interval-comp/main.html) as well as to the Journal of Reliable Computing for up-to-date information and references.

If the function $f$ is the response of a linearly elastic structure to static loads, then one can use the interval finite element formulation developed by Muhanna and Mullen [25, 26] to efficiently calculate $f(A_j)$. Finally, if $f$ is an eigenvalue of a linearly elastic structure one can use the procedure developed by Modares and Mullen [21] to efficiently calculate $f(A_j)$.

### 2.3 Step 3

Calculate the upper, $F_{y,upp}$, and lower, $F_{y,low}$, bounds on the cumulative distribution function (CDF) of $y$, $F_y$, as follows:

$$F_{y,low}\left(y\right) \le F_y\left(y\right) \le F_{y,upp}\left(y\right) \tag{3.a}$$

where

$$F_{y,upp}\left(y\right) = \sum_{A_j : y\ge \inf\left(f\left(A_j\right)\right)} m\left(A_j\right) \tag{3.b}$$

$$F_{y,low}\left(y\right) = \sum_{A_j : y\ge \sup\left(f\left(A_j\right)\right)} m\left(A_j\right) \tag{3.c}$$

The proposed procedure allows for an explicit evaluation of the error involved in the calculation of the whole cumulative distribution function, i.e.:

$$\mathrm{Err}_{\max} = \max_{y}\left(F_{y,upp}(y) - F_{y,low}(y)\right) \tag{4}$$

or the error at a particular value $y^*$:

$$\mathrm{Err}(y^*) = F_{y,upp}(y^*) - F_{y,low}(y^*) \tag{5}$$

### 3. Only the CDF of a particular value $y^*$ must be calculated

Consider the case in which the cumulative probability of only a particular element of $Y$, say $y^*$, is of interest, as is the case in reliability analyses (see Section 4). In this case, it is advisable to start off with a coarse partition of $D$ into subsets $A_i$. Let:

$$S_1 = \{A_i : \sup f(A_i) < y^*\}, \tag{6.a}$$

$$S_2 = \{A_i : \inf f(A_i) > y^*\} \tag{6.b}$$

$$S_3 = \{A_i : \inf f(A_i) < y^* < \sup f(A_i)\} \tag{6.c}$$

$$C_k = \bigcup_{A_i : A_i \in S_k} A_i \tag{6.d}$$

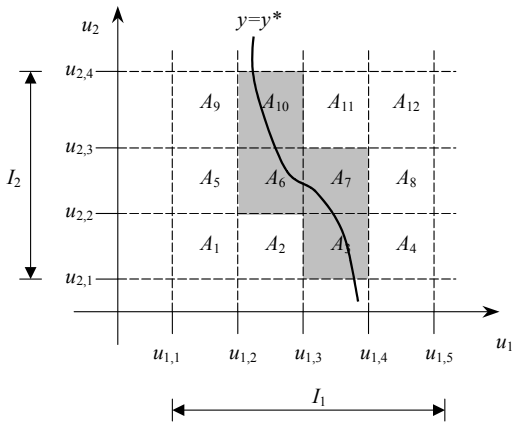



*Figure 1.* The image of focal elements $A_3$, $A_6$, $A_7$, and $A_{10}$ straddle $y^*$.

Of course, $D = \bigcup_{k=1}^{3} C_k$ .

To illustrate the procedure, consider a two-dimensional case ($p = 2$), the extension to larger dimensions being straightforward. It is assumed that the curve $f(u_1, u_2) = y^*$ intersect the boundary of $D$ at two points (as is the case in most reliability analyses). Figure 1 illustrates an example, in which:

$$S_1 = \{A_1, A_2, A_5, A_9\}$$

$$S_2 = \{A_4, A_8, A_{11}, A_{12}\}$$

$$S_3 = \{A_3, A_6, A_7, A_{10}\}$$

As can be seen in Figure 1, elements in $S_3$ are those (and only those) focal elements intersected by the contour curve $f(u_1, u_2) = y^*$.

The procedure for bracketing $F_y(y^*)$ can be summarized as follows:

1) Determine the focal elements in $S_3$ as explained in Section 3.1 below.
2) Determine $C_1$ as explained in Section 3.2 below.
3) Calculate the bounds on the CDF of $y^*$ as:

$$F_{y,low}(y^*) = \sum_{A_j \in S_1} m'(A_j) \le F_y(y^*) \le F_{y,upp}(y^*) = \sum_{\substack{A_j \in S_1 \\ A_j \in S_3}} m'(A_j) \qquad (7)$$

where $m'(A_j)$ is given in Eqs. (1) or (2). It is to be noted that the weights $m'(A_j)$ do not have to be calculated for each focal element of partition $D$, but only for the focal elements in $S_1$ and $S_3$. Because the cumulative probability of $y^*$ is in most cases small, the focal elements in $S_1$ and $S_3$ are less numerous than the focal elements in $S_2$. If the bounds in Eq. (7) are too large, the procedure in Section 3.3 below is followed.

### 3.1 Determining $C_3$

The determination of $C_3$ may be carried out in two parts:

1) Find a focal element belonging to $S_3$ along the boundary of $D$.
2) Find the remaining focal elements belonging to $S_3$.

Since region $C_1$ is in most cases smaller than region $C_2$, it is more efficient to start searching from the corner(s) belonging to $C_1$. Therefore, part one can be articulated into the following sequence:

1.1) WHILE $f(P) > y^*$, calculate function $f(P)$ at the corners $P$ of $D$. Let $P^* = P$.

1.2) Consider the local numbering of points along the boundary and of focal elements as in Figure 2. Let $j^*$ be the smallest $j$ such that:
$f(u_{0,j}) > y^*$.
IF such $j^*$ does not exist, THEN GOTO point 1.3.
Calculate the image of $A_{1,j^*}$ to check if it belongs to $S_3$.
IF $A_{1,j^*}$ belongs to $S_3$:
THEN calculate the image of $A_{1,j}$ for $j < j^*$ until $A_{1,j} \in S_1$.
ELSE $A_{1,j^*-1}$ belongs to $S_3$, and calculate the image of $A_{1,j}$ for $j < j^*-1$ until $A_{1,j} \in S_1$.
GOTO point 2.1.

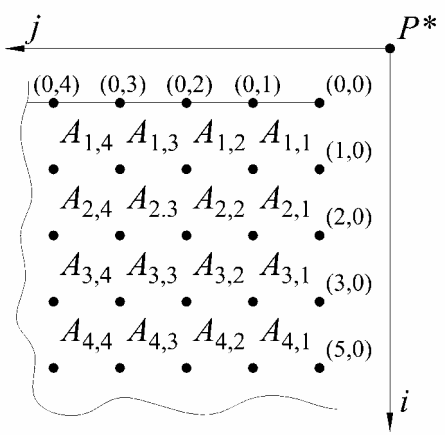



*Figure 2.* Local numbering of focal elements and points on the boundary of $D$.

1.3) If such $j^*$ does not exist, then let $i^*$ be the smallest $i$ such that: $f(u_{i,0}) > y^*$.
Calculate the image of $A_{i^*,1}$ to check if it belongs to $S_3$.
IF $A_{i^*,1}$ belongs to $S_3$:
THEN calculate the image of $A_{i,1}$ for $i < i^*$ until $A_{i,1} \in S_1$.
ELSE $A_{i^*-1,1}$ belongs to $S_3$, and calculate the image of $A_{i,1}$ for $i < i^*$-1 until $A_{i,1} \in S_1$.
GOTO point 2.1.

If the focal element(s) determined in step 1 was (were) along a boundary edge parallel to the $x$-axis, then let $i$ =1, and the other focal elements in $S_3$ are found using the following procedure:

2.1) For the current $i$, let $A_{i,j^*}$ be the focal element in $S_3$ with the largest $j$. Set $i = i + 1$. Calculate the image of $A_{i,j^*}$.
IF $A_{i,j^*} \in S_3$,
THEN, calculate the image of $A_{i,j}$ for $j > j^*$ until $A_{i,j} \in S_2$;
calculate the image of $A_{i,j}$ for $j < j^*$ until $A_{i,j} \in S_1$.
ELSE, calculate the image of $A_{i,j}$ for $j < j^*$ until $A_{i,j} \in S_1$.
IF $i = n_2$,
THEN STOP
ELSE GOTO point 2.1.

If the curve $f(u_1, u_2) = y^*$ is known to be concave (resp. convex) toward $P^*$, then Point 2.1 simplifies as follows:

2.1) For the current $i$, let $A_{i,j^*}$ be the focal element in $S_3$ with the largest $j$. Set $i = i + 1$. Calculate the image of $A_{i,j}$ for $j \le j^*$ (resp. $j \ge j^*$) until $A_{i,j} \in S_1$.
IF $i = n_2$,
THEN STOP
ELSE GOTO point 2.1.

If the focal element(s) determined in step 1 was (were) along a boundary edge parallel to the $y$-axis, then let $j$ =1, and the other focal elements in $S_3$ are found using the following procedure:

2.2) For the current $j$, let $A_{i^*,j}$ be the focal element in $S_3$ with the largest $i$. Set $j = j + 1$. Calculate the image of $A_{i^*,j}$.
IF $A_{i^*,j} \in S_3$,
THEN, calculate the image of $A_{i,j}$ for $i > i^*$ until $A_{i,j} \in S_2$;

calculate the image of $A_{i,j}$ for $i < i^*$ until $A_{i,j} \in S_1$.
ELSE, calculate the image of $A_{i,j}$ for $i < i^*$ until $A_{i,j} \in S_1$.
IF $j = n_1$,
THEN STOP
ELSE GOTO point 2.2.

If the curve $f(u_1, u_2) = y^*$ is known to be concave (resp. convex) toward $P^*$, then Point 2.2 simplifies as follows:

2.2) For the current $j$, let $A_{i^*,j}$ be the focal element in $S_3$ with the largest $i$. Set $j = j + 1$. Calculate the image of $A_{i^*,j}$ for $i \le i^*$ (resp. $i \ge i^*$) until $A_{i,j} \in S_1$.
IF $j = n_1$,
THEN STOP
ELSE GOTO point 2.2.

### 3.2 Determining $C_1$

$C_1$ completely lies on one side of the curve $f(u_1, u_2) = y^*$, and consequently, of set $C_3$. $C_3$ will be on the side containing corner $P^*$, which was determined at Point 1.1.

### 3.3 Discretization refinement

If the bounds (7) on the CDF of $y^*$ are too large, the discretization refinement is restricted to the focal elements in $S_3$. In fact, focal elements belonging to $S_1$ map to the left of $y^*$ on the real line, and therefore do not need to be further discretized because their contribution to $F_y(y^*)$ is already known. Likewise, focal elements belonging to $S_2$ map to the right of $y^*$ on the real line, and therefore do not need to be further discretized because they do not contribute to $F_y(y^*)$ altogether.

Additionally, it is useless to further discretize those focal elements in $S_3$ whose weight $m'(A_j)$ is very small as compared to the required precision on the CDF of $y^*$ (e.g. $m'(A_j) = 10^{-10}$ if the required precision is $10^{-4}$). Therefore, the number of focal elements that need to be further discretized is generally very small, which leads to drastic savings in the number of function calls.

Once the focal elements in $S_3$ have been further discretized into sub-elements, a procedure similar to that described in Section 3.1 can be used for determining the sub-elements belonging to $S_3$.

## 4. Reliability evaluation

Let $u=(u_1,...,u_p)$ be a vector of uncertain parameters that control the behavior of a given system. The safety of a system is quantified by the safety margin $z(u)$, such that [1, 2, 11, 16, 20]

- If $z<0$ the system is unsafe.
- If $z>0$ the system is safe.
- If $z=0$ the system is at a limit state condition.

The probability of failure of the system is defined as

$$\mathrm{Pro}_{fail} = \mathrm{Pro}(z < 0) \tag{8}$$

In general, a system is accepted if the probability of failure is smaller than a limit value

$$\mathrm{Pro}_{fail} < \mathrm{Pro}_{\lim} \tag{9}$$

If the input $u$ is known through its joint probability function, then the cumulative distribution function $F_z(z)$ of $z$ can be calculated and

$$\text{Pro}_{fail} = F_z(0) \tag{10}$$

For complex systems, the calculation of $F_z(z)$ may very cumbersome, and a bracketing of the failure probability can be obtained using the procedures presented in Section 3, leading to Eq. (7), i.e.

$$F_{z,low}(0) \le F_z(0) \le F_{z,upp}(0) \tag{11}$$

where

- $F_{z,low}$ is the lower cumulative distribution function of $z$;
- $F_{z,upp}$ is the upper cumulative distribution function of $z$.

Three cases can be distinguished:

1) If $\text{Pro}_{\text{lim}} < F_{z,low}(0)$, then the system is certainly unsafe, and it is not necessary to further increase the fineness of the discretization of the focal elements in $S_3$.
2) If $\text{Pro}_{\text{lim}} > F_{z,upp}(0)$, then the system is certainly safe, and it is not necessary to further increase the fineness of the discretization of the focal elements in $S_3$.
3) If $F_{z,low}(0) < \text{Pro}_{\text{lim}} < F_{z,upp}(0)$, then it is necessary to further increase the fineness of the discretization. As discussed in Section 3.3, one only needs to further discretize focal elements in set $S_3$.

As an alternative to the procedure proposed here, once the joint pdf of the uncertain input $u$ has been discretized as described in Section 2, one can use the efficient approximation technique developed by Bae *et al*. [3] to calculate approximate reliability bounds. However, these bounds do not offer a guaranteed envelope because Bae's procedure uses the Multi-Point Approximation method to construct a surrogate for the original safety margin using the Two-Point Adaptive Non-linear Approximation [34] as a local approximation.

## 5. Numerical example

Consider a beam of length $l$ = 5 m, fixed at one end, and subjected to a random concentrated load $u_1$ at the free end, and to a random distributed load $u_2$ along all its length. Let us assume $u_1 \sim N(10, 1)$ kN, and $u_2 \sim N(1, 0.3)$ kN/m, with a correlation coefficient of 0.5; a similar example is proposed by Ang and Tang ([1], Problem 4.18). The resistant moment at the fixed end is equal to $M$ = 90 kN·m. The safety margin of the bending resistance at the fixed end reads

$$z = M - (5 \cdot u_1 + 12.5 \cdot u_2) \tag{12}$$

and it is a normal variate with mean equal to 27.5 kN·m, and standard deviation $\sigma$ = 7.603 kN·m. This closed-form solution allows for a handy check of the results obtained with the proposed procedures because the exact probability of failure is $1.49 \cdot 10^{-4}$. Let us assume that the limit probability of failure is $10^{-5}$, and that one wants to determine whether the beam is safe or not under the given random loads.

Let us use the domain $D$ and the discretization shown in Figure 3. Following the procedure outlined in Section 3.2 (Point 1.1), function $z$ is evaluated at the corners of $D$ in clockwise order starting from $P$ = (-10, -1), until corner $P$ = (30, 3) yields $z(P) < 0$. We set $P^*$ = (30, 3), and (Point 1.2) by marching along $u_2$ = 3, we get $j^*$ = 4 (absolute coordinates: (10.5, 3)). By calculating the images of focal elements $A_{1,j}$ with $j \le j^* = 4$, it is found that $A_{1,4}$ and $A_{1,3}$ (absolute

numbering $A_{70}$ and $A_{80}$ in Figure 3) belong to $S_3$, whereas $A_{1,2}$ (absolute numbering $A_{90}$ in Figure 3) does not, and the search for focal elements of $S_3$ along the boundary is finished.

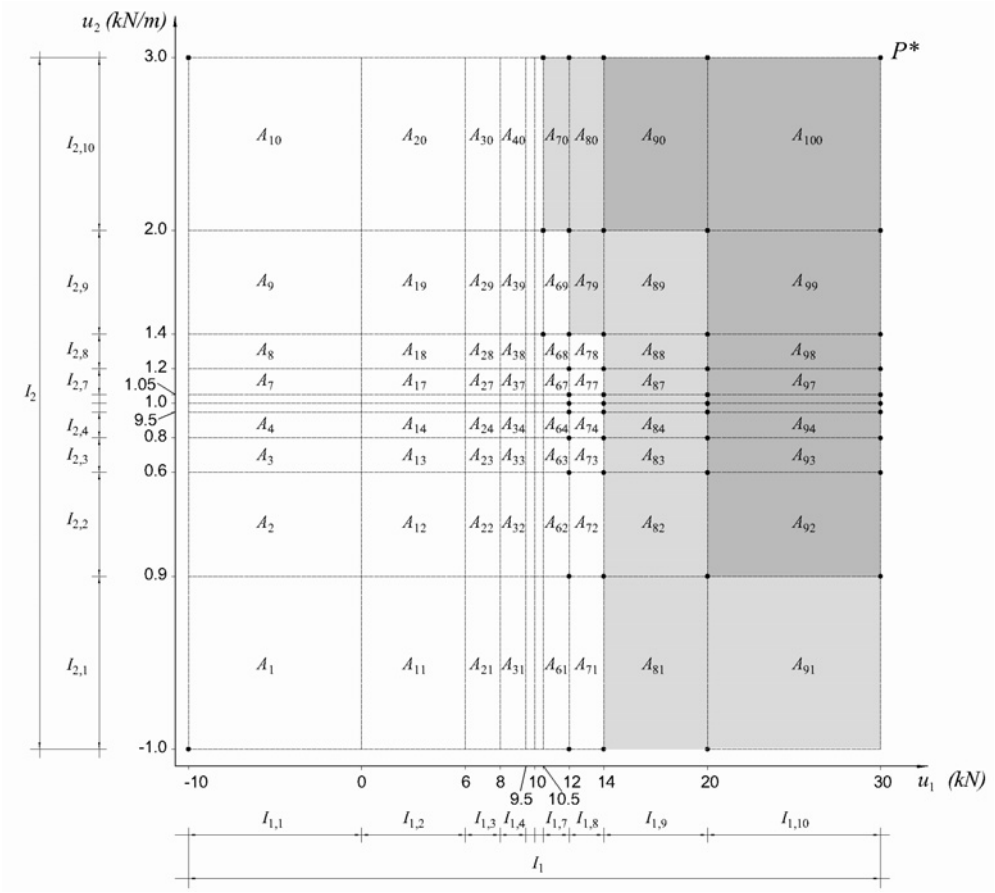



*Figure 3.* First discretization of set $D=I_1 \times I_2$ into focal elements for the calculation of the system reliability. A dot marks the points at which function $f$ must be evaluated. A light gray hatch identifies focal elements belonging to set $S_3$, whereas a dark gray hatch identifies focal elements belonging to set $S_1$.

Following Point 2.1, one gets $i = 1$, and $(i, j^*) = (1, 4)$. Let us set $i = 1+1 = 2$, and calculate the image of $A_{2,4}$; since $A_{2,4}$ belongs to $S_2$, the procedure does not calculate the images of $A_{2,j}$ for $j > j^*$ because all of these focal elements belong to $S_2$. The procedure calculates the images of $A_{2,j}$ for $j < j^*$ until $A_{2,j} \in S_1$, which occurs for $j = 1$, i.e. $A_{2,4} \in S_2$, $A_{2,3} \in S_3$, $A_{2,2} \in S_3$, $A_{2,1} \in S_1$. Since $i = 2 \neq n_2 = 10$, the procedure goes back to Point 2.1 with $i = 2$, and so on.

The results of the procedure are illustrated in Figure 3, in which a light gray hatch is used for the focal elements belonging to $S_3$, and a dark gray hatch is used for the focal elements belonging to $S_1$. Table 1 gives the weights $m'(A)$ for the elements of $S_3$. A total of 48 function evaluations were necessary to determine sets $C_1$, $C_2$, and $C_3$.

The calculated probability of failure is in the range

$$F_{z,low}(0) = 1.01 \cdot 10^{-2} < F_z(0) = 1.49 \cdot 10^{-4} < F_{z,upp}(0) = 2.57 \cdot 10^{-6}$$

*Table 1.* Focal elements in set $S_3$.

| Focal element (global numbering as in Figure 3) | Weight $m'(A_i)$ |
|---|---|
| 70 | $2.24 \cdot 10^{-4}$ |
| 79 | $9.66 \cdot 10^{-3}$ |
| 80 | $1.72 \cdot 10^{-4}$ |
| 81 | $6.29 \cdot 10^{-15}$ |
| 82 | $1.21 \cdot 10^{-9}$ |
| 83 | $2.14 \cdot 10^{-8}$ |
| 84 | $1.19 \cdot 10^{-7}$ |
| 85 | $1.02 \cdot 10^{-7}$ |
| 86 | $1.62 \cdot 10^{-7}$ |
| 87 | $1.13 \cdot 10^{-6}$ |
| 88 | $4.35 \cdot 10^{-6}$ |
| 89 | $2.32 \cdot 10^{-5}$ |
| 91 | $1.57 \cdot 10^{-15}$ |

Since $F_{z,low}(0) < \mathrm{Pro}_{\lim} < F_{z,upp}(0)$, it is necessary to increase the fineness of the discretization (Section 3.3). The weights of the focal elements in Table 1, are negligible except for $A_{70}$, $A_{79}$, $A_{80}$, and $A_{89}$; therefore, only the latter four focal elements are further discretized into $5 \times 5 = 25$ sub-focal elements each as depicted in Figure 4. A light gray hatch is used for the focal elements belonging to $S_3$, and a dark gray hatch is used for the focal elements belonging to $S_1$. The calculated probability of failure is in the range

$$F_{z,low}(0) = 4.51 \cdot 10^{-5} < F_z(0) = 1.49 \cdot 10^{-4} < F_{z,upp}(0) = 3.25 \cdot 10^{-4}$$

This range gives a guarantee that $\mathrm{Pro}_{\lim} < F_{z,low}(0)$, and therefore the beam is unsafe.

Additional 58 function evaluations were used in the discretization refinement. It is remarkable that only 48 + 58 = 106 function evaluations were necessary to perform a reliability analysis with 100% confidence, despite the very low value of the probability of failure.

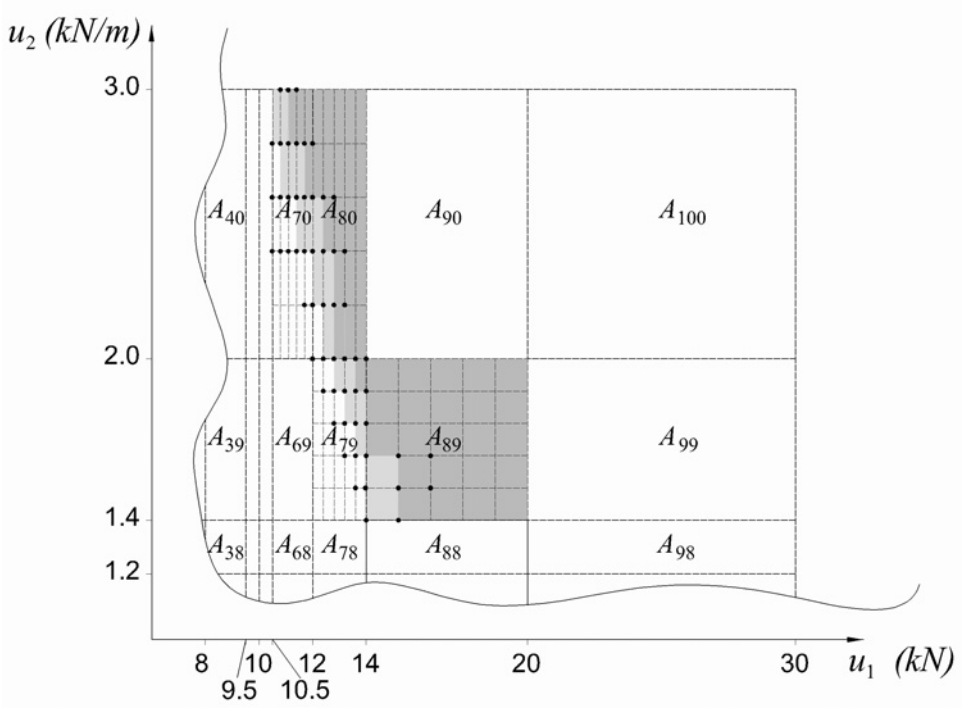



*Figure 4.* Second discretization of focal elements $A_{70}$, $A_{70}$, $A_{79}$, and $A_{89}$, into sub-focal elements for the refinement of the calculation of the system reliability. A dot marks the points at which function *f* must be evaluated. A light gray hatch identifies sub-focal elements belonging to set $S_3$, whereas a dark gray hatch identifies sub-focal elements belonging to set $S_1$.

On the other hand, the number of function evaluations necessary to achieve an error $e = F_{z,upp}(0) - F_{z,low}(0) = 2.8 \cdot 10^{-4}$ with confidence 1-δ with crude Monte Carlo is equal to [15]

$$n_C(e,\delta) = \frac{1}{4 \cdot \delta \cdot e^2}$$

Table 2 presents the number of Monte Carlo simulations for several values of the confidence level. It is evident that the procedure proposed leads to substantial computational savings. For example, if one requires that the reliability of reliability calculations be at least equal to the reliability of the structure being analyzed, then one should require a confidence level of 99.999%, which yields some 90 million function calls.

*Table 2.* Number of Monte Carlo simulations vs. confidence level.

| Confidence level (%) | $n_c$ |
|---|---|
| 90 | 8,929 |
| 95 | 17,857 |
| 99 | 89,286 |
| 99.9 | 892,857 |
| 99.99 | 8,928,571 |
| 99.999 | 89,285,710 |

## 6. Conclusions

The searching procedure presented allows one to bound the CDF of a dependent variable with 100% confidence, and allows for a guaranteed evaluation of the error involved in the calculations. These bounds are often enough to make decisions, and require a minimal number of function calls. The procedure is not intrusive, i.e. it can be equally applied when the function is a complex computer model (black box). The proposed procedure can handle input information consisting of probabilistic, interval-valued, set-valued, or random-set-valued information, as well as any combination thereof. The function as well as the joint pdf of the input variables can be of any type.

The application to a beam subjected to two random loads showed that the number of function calls is drastically reduced as compared to Monte Carlo methods. For example, only 106 function calls were necessary to conclude with 100% confidence that the CDF was greater than the specified limit value of the probability of failure, and that the beam was unsafe.

The drawback of the procedure presented is that it suffers from a dimensionality effect. However, Monte Carlo methods are not free from dimensionality effects. For example, Davis' and Rabinowitz's comment as follows on multiple integration by sampling when more than 12 variables are involved ([10], page 417): "Sophisticated methods of variance reduction appear to exhibit a dimensional effect and are probably ruled out in this range. Some authors feel that the dimensional effect may even play a role in crude [sampling] methods inasmuch as it may occur in the constant in the asymptotic error term." Indeed, Sloan and Wozniakowski [30] have shown that Monte Carlo may depend polynomially or even exponentially on the number of variables.

Current research is focusing on improving the efficiency of the procedures presented here by incorporating Bayesian philosophies and procedures [9, 18, 27] into the method's algorithm, with the aim of reducing the random variables in the problem to those that appreciably influence the output. Furthermore, adaptive techniques are being investigated in order to locally refine the discretization of those focal elements whose image straddles $y^*$.

Current applications aim at integrating the procedure presented with interval finite element formulations [21, 25, 26] for the efficient reliability analysis of structures.

## References


1. Ang A. H.-S. and Tang W.H.: *Probability Concepts in Engineering Planning and Design, Vol.1*, Wiley, New York, 1975.
2. Ang A. H.-S. and Tang W.H.: *Probability Concepts in Engineering Planning and Design, Vol.2*, Wiley, New York, 1984.
3. Bae H.-R., Grandhi R. V., and Canfield R. A.: An Approximation Approach for Uncertainty Quantification Using Evidence Theory, *Reliability Engineering and System Safety*, [accepted January 2004].
4. Berleant D.: Automatically verified arithmetic on probability distributions and intervals". In *Applications of Interval Computations* (R. B. Kearfott and V. Kreinovich eds.), Chap. 10, pp. 227-244, Kluwer Academic Publishers, Dordrecht, 1996.
5. Berleant D. and Goodman-Strauss, C.: Bounding the results of arithmetic operations on random variables of unknown dependency using intervals, *Reliable Computing* **4** (1998), pp. 147-165.
6. Berleant D.: Automatically verified reasoning with both intervals and probability density functions, *Interval Computations* **2** (1993), pp. 48-70.
7. Berleant, D. and Cheng, H., A software tool for automatically verified operations on intervals and probability distributions, *Reliable Computing* **4** (1998), pp. 71-82.

8. Berleant, D., Xie, L. and Zhang, J.: Statool: a tool for Distribution Envelope Determination (DEnv), an interval-based algorithm for arithmetic on random variables, *Reliable Computing* **9** (2003), pp. 91 – 108.

9. Craig, P. S. Goldstein, M., Rougier, J. and Seheult, A. H.: Bayesian forecasting for complex systems using computer simulators, *J. American Statis. Assn*. **96**(454) (2001), pp. 717-729.

10. Davis, P.J. and Rabinowitz, P.: *Methods of Numerical Integration – Second Edition*, Academic Press, London (UK), 1984.

11. Ditlevsen O. and Madsen H. O., *Structural Reliability Methods*, John Wiley & Sons, Chichester, 1996.

12. Dubois D. and Prade H.: Random sets and fuzzy interval analysis, *Fuzzy Sets and Systems* **42** (1991), pp. 87-101.

13. Ferson, S.: *RAMAS Rick Calc 4.0: Risk Assessment with Uncertain Numbers*. Lewis Press, Boca Raton (FL), 2002.

14. Ferson, S.: What Monte Carlo methods cannot do. *Journal of Human and Ecological Risk Assessment*, **2** (1996), pp. 990-1007.

15. Fishman, G. S.: *Monte Carlo: Concepts, Algorithms, and Applications*, Springer Verlag, New York, 1995.

16. Harr M.E.: *Reliability-based Design in Civil Engineering*. McGraw Hill, New York (1987). Reprinted by Dover, Minneola, 1996.

17. Kearfott R. B.: COCOS'02 - A Workshop on Global Constrained Optimization and Constraint Satisfaction October 2-4, 2002, Sophia-Antipolis, France, *Reliable Computing* **9** (2003), pp. 81 – 87.

18. Kennedy M. C. and O'Hagan A.: Bayesian calibration of computer models, *J. R. Statistical Soc B* **63**(3) (2001), pp. 425-464 (with discussion).

19. Lodwick, W. A. and Jamison K. D.: Estimating and validating the cumulative distribution of a function of random variables: toward the development of distribution arithmetic, *Reliable Computing* **9** (2003), pp.127 – 141.

20. Melchers R. E.: *Structural Reliability Analysis and Prediction* – Second Edition. John Wiley & Sons, Chichester, 1999.

21. Modares, M., and Mullen, R. L.: Free vibration of structures with interval uncertainty. Proc. 9th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability PMC2004 (Wojtkiewicz, S., Chanem, R. and Red-Horse, J. eds.), Albuquerque (NM), July 26-28, 2004, paper 7-103, ASCE, Reston, VA, 2004.

22. Moore A. S.: Interval risk analysis of real estate investment: a non-Monte Carlo approach. *Freiburger Interval-Berichte*, 85/3 (1985), pp. 23-49.

23. Moore R. E.: *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.

24. Moore R. E.: Risk analysis without Monte Carlo methods. *Freiburger Interval-Berichte*, 84/1 (1984), pp. 1-44.

25. Muhanna, R. L. and Mullen, R. L.: Fundamental concepts for interval-based finite element formulations, *Journal of Reliable Computing*, 2004, (tentatively accepted).

26. Muhanna, R. L. and Mullen, R. L.: Uncertainty in mechanics problems - Interval-based approach, *Journal of Engineering Mechanics*, ASCE, **127**(6) (2001), pp. 557-566.

27. Oakley, J. and O'Hagan, A.: Bayesian inference for the uncertainty distribution of computer model outputs, *Research Report No. 497/00*, Department of Probability and Statistics, University of Sheffield, 2000. Submitted to *Biometrika*.

28. Ratschek H. and Rokne J.: *Computer Methods for the Range of Functions*, Ellis Horwood Limited, Chirchester, 1984.

29. Ratschek H. and Rokne J.: *New Computer Methods for Global Optimization*. Ellis Horwood Limited, Chirchester, 1988.

30. Sloan, I. H and Wozniakowski, H.: When does Monte Carlo depend polynomially on the number of variables? Applied Mathematics Report AMR03/1, Department of Applied Mathematics, The University of South Wales, Sidney, Australia, 2003.

31. Tonon, F.: On the use of Random Set Theory to bracket the results of Monte Carlo simulations, *Reliable Computing* **10** (2004), pp. 107-137.

32. Tonon F., Bernardini A. and Mammino A.: Determination of parameters range in Rock Engineering by means of Random Set Theory, *Reliability Engineering & System Safety*, **70**(3) (2000), pp. 241-261.

33. Tuy H.: *Convex Analysis and Global Optimization*. Kluwer Academic Publisher, Dordrecht, 1998.

34. Wang, L. P. and Grandhi, R. V.: Improved two-point function approximations for design optimization, *AIAA Journal* **33**(9) (1995), 1720–1727.

# Requirements Analysis for Engineering Computation

Spencer Smith, Lei Lai and Ridha Khedri
*Computing and Software Department, McMaster University*

**Abstract.** This paper argues that the reliability of engineering computation can be significantly improved by adopting software engineering methodologies for requirements analysis and specification. The argument centers around the fact that the only way to judge the reliability of a system is by comparison to a specification of the requirements. This paper also points to methods for documenting the requirements. In particular, a requirements template is proposed for specifying engineering computation software. To make the mathematical specification easily understandable by all stakeholders, the technique of using tabular expressions is advocated. To clarify the presentation, a case study of the documentation for a system for analyzing statically determinant beams is presented.

**Keywords:** requirements analysis, requirements template, engineering computation, reliable computation, software quality

## 1. Introduction

Software engineers generally advocate that the first step in system development should be a systematic elicitation, analysis and documentation of the requirements, because it is much easier and cheaper to correct mistakes and misconceptions at the beginning of the process than it is to try and fix problems during implementation and maintenance. There is wide agreement in the software engineering community on the necessity of a complete and consistent software requirements document for evaluating any software system quality, including reliability [4]. Requirements documentation has been demonstrated to be effective in other application areas, such as with business applications [29] and for real-time systems, such as the U.S. Navy's A-7E military aircraft [16] and the shutdown systems of the Darlington nuclear generating station [26]. However the requirements stage of software development is often neglected when solving engineering computation problems. This paper argues that the reliability of engineering computation can be significantly improved by adopting software engineering methodologies for requirements analysis and specification.

The importance of requirements analysis and documentation is not only widely recognized by software engineers. All engineering disciplines understand the importance of documenting requirements for large and complex systems. For instance, an automobile manufacturer will gather requirements from customers to determine whether fuel efficiency is considered more important than luxury, or vise versa. Similarly, a structural engineer would not start designing a building until she had determined the following: How many floors will the building have? What are the expected loads? Will the building be used as a hospital, or a school, or a shopping mall, or for some other purpose? To judge the success of the design, it is necessary to take the requirements into account. For instance, the required reliability for a hospital is higher than for other structures, so the probability of a hospital collapsing due to an earthquake should be less than that for an office building

in the same situation. Given that engineers recognize the importance of requirements for large and complex systems, and that the size and complexity of software systems, including engineering computation systems, seems to be continually growing, it is necessary to ensure the quality of software systems via documenting their requirements.

The central argument of this paper is that a requirements specification document is necessary to judge the reliability of engineering computation software. Reliability is a measure of the dependability of a system. One definition says that "reliability of software is defined to be the ability of the software to behave consistently in a user-acceptable manner when subjected to an environment in which it was intended to be used"[5, page 310]. Another definition say that "reliability requirements deal with failures to provide service. They determine the maximum allowed software system failure rate, and can refer to the entire system or to one or more of its separate functions" [10, page 39]. The definitions of reliability depend on the existence of a specification of the requirements because one cannot judge correctness, user acceptability, or failure rates, without knowing the standard for comparison. In engineering computation, one needs to know exactly what problem the system is required to solve and the values for the acceptable tolerances, or it is impossible to judge whether the results are correct. Although the field of engineering computation has developed many excellent methodologies for producing efficient and accurate numerical results, the design decision for selecting the appropriate methodology are often made in an ad hoc manner because there is a lack of appropriately documented requirements to guide the decision. The existence of a complete and consistent requirements document can lead to better decisions for improving reliability and it can also improve other software qualities, such as usability, verifiability, maintainability, reusability and portability, which are sometimes neglected in engineering software.

The first section below presents background information, including an overview of software engineering methodologies for requirements elicitation, analysis and documentation. The background section also contains a brief summary of the syntax and semantics of tabular expressions, which are introduced in this document because they provide a relatively easy way of documenting complex requirements. After the background section the value of requirements documentation for engineering computation is explored in depth. Following this, the methodology promoted in this paper is made more concrete by presenting some excerpts from a requirements document for a software system to analyze statically determinate beams. The discussion of this example highlights the requirements template that was followed in constructing the requirements document. The final section consists of concluding remarks.

## 2. Background

The idea of this paper is to borrow guidelines from software engineering to improve engineering computation. To do this, it is first necessary to understand some aspects of software engineering. This section provides necessary information on some current software engineering methodologies. In particular, an overview of requirements elicitation, analysis

and documentation is given and a particular approach for documenting formal requirements is described: the technique of tabular notation. More detail on the software engineering methodologies discussed here can be found in [21].

## 2.1. Overview of Requirements Elicitation, Analysis and Documentation

This section highlights how requirements fit into the software development process by first providing a description of the waterfall model of software development. Following this, the software requirements activities are described for elicitation, analysis, documentation, validation and verification of requirements. A section is also provided to describe the end-product of the requirements phase, the document called the Software Requirements Specification (SRS). The final section describes a requirements template, which is a documentation approach used by software engineers to improve the quality of the SRS.

### 2.1.1. *Waterfall Model*

In a common model of the software development lifecycle the first phase involves gathering requirements, analyzing them and documenting them. This lifecycle model, which is graphically depicted in Figure 1, is termed the waterfall model because each stage flows into the next as the process moves downstream. The stages of the waterfall model consist of requirements, design and coding. The back and forth arrows represent the iterative process of validation and derivation at each phase before proceeding to the next. After the code is developed and validated against the design, it also has to be validated against the requirements. This is represented by the dashed arrow from "Code" to "Requirements". The acceptance testing, in which the system is used in the real world, is performed before the software is finally accepted. This is indicated by the dashed arrow from "Code" to "Real World". If the requirements are modified, the whole procedure has to be repeated.

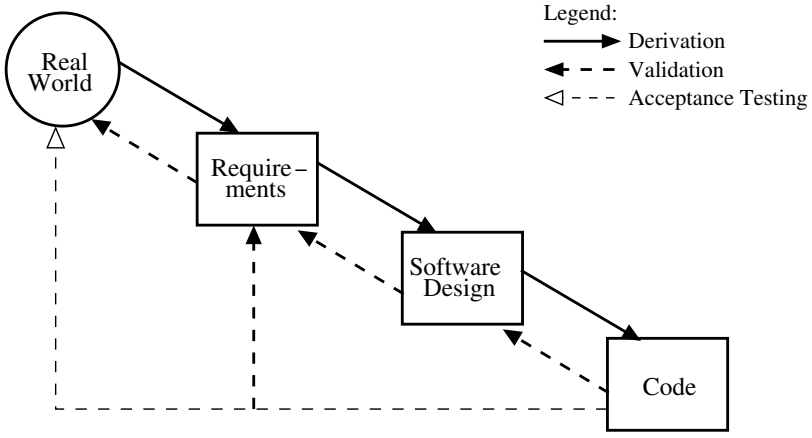



*Figure 1.* Waterfall model of the software development lifecycle

The waterfall model is not the only model of the software development process [12, pages 385-456], but it is the model that is cited here because it closely parallels how engineers typically think about their work flow. Moreover, as Section 3 will show, the software lifecycle model and the scientific method essentially follow the same waterfall process. The waterfall model is also well-suited for engineering computation problems because the waterfall model works well when the requirements are stable [12, page 409], which is certainly the case in engineering computation where the scientific theories of the laws of physics are slow to change. Another argument in favor of presenting the waterfall model is that even though the process of software development is never as rational as that presented in Figure 1, the advantages of a rational process can still be obtained by documenting the work products as if they were developed and written following the waterfall model [27].

2.1.2. *Software Requirements Activities*

To develop engineering computation software in a rational way, it is necessary to document the software requirements. A software requirement is a description of how the system should behave, or of a system property or attribute [33]. In [34], a software requirement is defined as a software capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.

Software requirements activities cover all of the activities involved in discovering, analyzing, documenting, and maintaining a set of requirements for a computer-based system, with an emphasis on using systematic and repeatable techniques [31, 33]. The software requirements activities begin with the software requirements elicitation. At this stage an attempt is made to work with the stakeholders to gather all of the information necessary for understanding the problem. After all of the requirements have been gathered they are analyzed, which involves refining and modeling the requirements. The goal of analysis is to discover problems, incompleteness and inconsistencies in the elicited requirements. The analysis is interleaved with the requirements elicitation phase. Some methodologies for requirements analysis include structured analysis, object-oriented analysis [22], goal based methods [22, 36], viewpoint methods [6] and component requirements analysis [13].

For the results of the requirements analysis to be useful for the subsequent development of the software, the requirements need to be documented in the software requirements specification document, which is discussed in the next section. To ensure quality, the software requirements must undergo a process of validation and verification to check the adequacy of the documented requirements. In the validation, the requirements model is examined to make sure that stakeholders' expectations are correctly captured. Verification involves checking the software requirements for certain properties, such as consistency, completeness, and modifiability.

In this paper, the term requirements analysis does double duty. It refers to the stage of requirements refinement and modeling described above, but it is also used as a shorthand to describe all of the requirements steps, from elicitation to verification and validation. In the literature, the term requirements engineering is sometimes used for this purpose. The meaning of the term requirements analysis should be clear from the context where the term is used.

In terms of quality, requirements should be reviewed to ensure that they are correct, unambiguous, complete, consistent, modifiable, verifiable, and traceable [17]. A good requirement should express "What" functionalities and qualities the system should have, but it should not mention "How" these requirements are to be accomplished. That is, the requirements should not impose design decisions. For instance, a requirement may specify an ordinary differential equation that must be solved, but it should not mention that a fourth order Runge-Kutta method should be employed. The requirements document should not tie the hands of the designer; she should be free to select any algorithm that will satisfy the requirements.

Besides the "What" versus "How" test, there are other tests that can be used to review requirements. One such test is the "what is ruled out" test. This test determines if a requirement actually makes a decision because if no alternatives are ruled out then no decision has really been made. Another test is the "negation" test. If the negation of a requirement represents a position that someone might argue for, then the original decision is likely to be meaningful. For instance, the statement that "the software should be reliable" has a negation that no one would argue for and thus the statement does not represent a good characterization of a requirement for the system.

2.1.3. *Software Requirements Specification (SRS)*

During the process of requirement gathering, the requirements need to be documented in a software requirement specification (SRS), which includes the external behavior of the system, the constraints placed on the implementation, the forethought about the lifecycle of the system, and the acceptable response to the undesired events [16]. The SRS is a document that clearly and precisely describes each of the essential requirements (functions, performance, constraints, and quality attributes) of the software and external interfaces [34].

Requirements documentation methods can be categorized according to their degree of formality, where formality is defined as the degree to which use is made of mathematical techniques and notations. The first group of methods are informal methods, which describe the requirements document in natural language. In principle, the requirements in natural language are universally understandable but, in practice, the meaning of requirements is not always obvious, because natural language is inherently ambiguous and analyzing such descriptions can be very difficult [19]. The second documentation methods are formal methods, which use mathematically formal syntax and semantics to specify system function and behavior. Example languages currently used in formal specifications are Z, VDM, CSP, etc. Formal methods do not have the ambiguity of natural language, but they can be time consuming to produce and as indicated in [2], formal methods do not help us "solve the difficulties caused by lack of understanding of the real world situation". The last group of methods are semi-formal methods that use diagrammatic modeling or object-oriented techniques. These methods are generally easier to develop and understand than formal methods. However, semi-formal methods are facing criticism for paying less attention to verification and validation of requirements [31]. At this time, requirements documentation methods are not well-developed and there is no universally accepted way of documenting requirements.

As the official statement of the system requirements for customers, end-users and software developers, the SRS provides many advantages during the lifecycle of the software project [17, 31, 33]. For instance, the SRS reflects the mutual understanding of the problem to be solved between the requirements analyst and the client and the SRS serves as a starting point for the software design phase because decisions are made explicitly before designing and coding. Other benefits of the SRS include the fact that it provides a basis for estimating costs and schedules and it allows validation and verification because it provides a baseline against which compliance can be measured. The SRS aids the software lifecyle because it facilitates incremental development. In many businesses, systems are built in increments; that is, the next generation inherits the features from the previous version, only enhancing the system with additional or improved features. The final benefit of an SRS is the financial benefit of finding problems early. If mistakes are found in the requirements stage, then they are much cheaper to fix than when they are found in a later stage of the software development. Empirical studies show that if one arbitrarily assigns unit cost to the effort required to detect and repair an error during the coding stage, then the cost to detect and repair an error during the requirements stage is between a fifth and tenth as much and the cost to detect and repair an error during maintenance is twenty times as much [5, page 25].

#### 2.1.4. *Requirements Template*

A requirements template provides a frame of reference, identifies needed information, and suggests an order of presentation so that the requirements can be best expressed [31]. The use of a template encourages a systematic procedure of requirements documentation. Since no single template can meet the needs of every requirements document, it is vital that the template be tailored to the needs of a particular audience [31].

The advantages of using requirements templates are discussed in [31, 33]. One advantage is that templates can increase the productivity of SRSs. Software can be developed to support the process of producing requirements documents conforming to the template. Furthermore, templates can increase the adequacy of SRSs because a well-organized format for the document acts as a checklist for writers of the SRS and reduces the chances of omitting information. Another benefit of a template is that it facilitates the communications among various SRS users, such as customers, developers, experts, etc., which in the context of engineering computation, will be researchers, software developers, physical modelers, computational scientist etc. Templates also provide the advantage of easing information handling by defining the content of each specific section. The readers can find information more easily and understand the relationships between different parts of the document. Finally, a template helps the process of software development by making it easier to compare two SRSs when they both conform to the same template.

There are several requirements specification frameworks that are designed for general purposes and contain good advice on how to write requirements and how to avoid problems [9, 23, 24, 7, 30, 34]. These templates are the result of many years of practice, consulting and research in requirement activities and thus provide a good foundation for software requirements documentation. They are subject to change and are usually not used without modification. The templates that have been developed to date focus on business applications

and real-time systems and do not address some of the issues of importance for engineering computation problems, which motivates the development of the new template discussed in Section 4.

### 2.2. Tabular Expressions

Tables are used for documenting the requirements in this paper because they improve readability so that formal documentation can be advocated to replace conventional documentation. Tabular expressions (or tabular notations) for computer programs and modules made their appearance in the late 1950s [18]. Multi-dimensional tabular expressions make it easier to consider every case separately while writing or reading a document, as opposed to the standard linear mathematical notation. The key ideas of tabular expressions, one of the cornerstones of the relational model for documenting the intended behavior of programs [6, 18, 25], were first developed in work for the US Navy and applied to the A-7E aircraft [15]. In the current case study, tabular expressions are used in the SRS for the beam analysis problem to formalize the specification of the system behavior. The advantages of tabular expressions are that they are well-structured, they can simplify the task of composition of table specifications to have a global or a dynamic view of the system's behavior, and they allow the achievement of SRS qualities attributes such as completeness and consistency.

A full review of tabular expressions is beyond the scope of this paper; details on the mathematics of tables can be found in [21]. An intuitive understanding of tables can be obtained by considering an illustrative example. The example is taken from the SRS for beam analysis. The SRS uses a table to specify the system response to input data for describing the beam problem and the constraints on this data. The example in Table I is for input of the distance from the left end of a beam to the point of application of a load ($x_1$). In this table $min_d$ and $max_d$ are the bounds on the admissible range of values for $x_1$, $@x_1$ is the symbol that represents $x_1$, $S_{GET}$ is the set of symbols of user interface variables that are accepted by the system, $ErrorMsg$ is a system output indicating the error mode, $ChangeOnly(var1, var2, ...)$ indicates that only the output variables $var1, var2, ...$ may change value, $i, j$ are the indexes of the table cells, $H_1$ and $H_2$ are headers that consist of an indexed set of cells, $G$ is a grid, and the composition rule is a relation expression that determines the relation represented by the table.

Table I. System Response to Constraints on Input Variable $x_1$

| Composition rule | $\cup_{i=1}^{4} H_2[i] \cap (\cap_{j=1}^{2} H_1[j] \ ; \ G[i,j])$ |
|---|---|

$H_1$

| | $S'_{GET} \cup =$ | $ErrorMsg' +=$ |
|---|---|---|
| $x_1 < 0$ | $\emptyset$ | $InvalidInput\_x_1$ |
| $0 \leq x_1 < min_d$ | $\emptyset$ | $x_1\_TooSmall$ |
| $x_1 > max_d$ | $\emptyset$ | $x_1\_TooLarge$ |
| $min_d \leq x_1 \leq max_d$ | $\{@x_1\}$ | $NULL$ |
| $H_2$ | $\wedge ChangeOnly(S_{GET}, ErrorMsg)$ | $G$ |

Although the mathematics of tabular expressions can be complex, the interpretation is natural and intuitive. In this example the value of $x_1$ determines the new values (indicated by a prime ($'$) symbol on a variables name) of $S_{GET}$ and $ErrorMsg$. For a given value of $x_1$ one should search for the matching predicate in $H_2$ and read the row in $G$ to determine what happens to the other variables. For instance, if $min_d \leq x_1 \leq max_d$ then, $S'_{GET} = S_{GET} \cup \{@x_1\}$ and $ErrorMsg' = ErrorMsg$.

## 3. Why Requirements Analysis for Engineering Computation?

Attempts to apply software engineering methodologies to engineering computation have usually payed little attention to the appropriate and rigorous documentation of requirements. Some ideas from software engineering have been applied to engineering computation, such as algebraic abstractions [1], object-oriented design [14], software components [8], and software patterns [3]. Although these approaches have advantages, the research usually focuses on the design and implementation and does not address how to improve the quality of engineering software from the requirements level. One exception to neglecting the requirements phase is a requirements analysis of data parallel applications [11]. Another exception documents the requirements of models of physical phenomena [20] using tabular expressions. However, this model of physical phenomena does not necessarily solve all of the problems for documenting the requirements of engineering software because the original idea of this model was developed for an embedded system, which has different needs than an engineering computation system. Moreover, this model allows the numerical methods, which are essentially implementation decisions, to be encompassed into the requirements documentation. This contradicts with the principle that requirements should not address "How", but only "What".

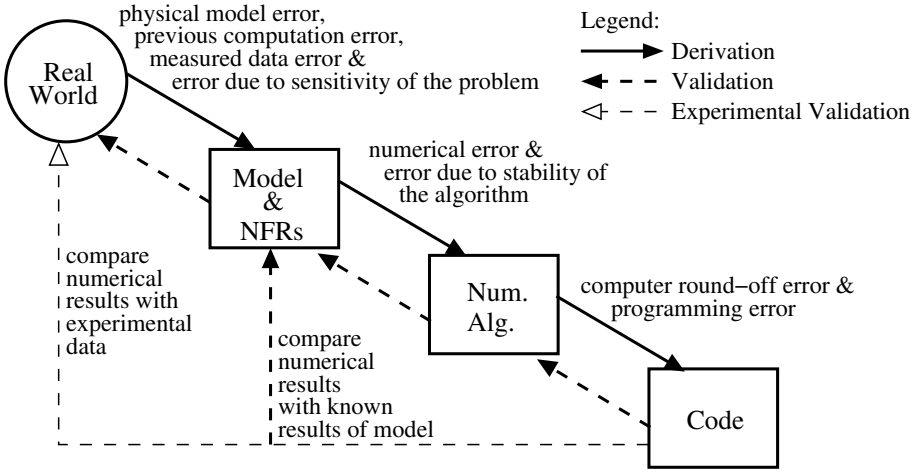



*Figure 2.* The Scientific Method

The apparent absence of studies on the requirements for engineering computation is not an indication that this is an unimportant topic. All of the benefits listed in Section 2.1.3 apply to engineering software, just as they do to other types of applications. Besides these arguments, there are also arguments that can be made that are specific to engineering software.

The argument that requirements analysis can improve the reliability of engineering computation is first made by observing the strong similarity between the waterfall model of the software lifecycle and the standard model of the scientific method. The strong similarity implies that engineering computation can benefit from methods that have proved to be successful in software engineering. Like the waterfall model, the typical work flow for the development of engineering modeling and simulation software can also be divided into several stages [32], as illustrated in Figure 2. First, physical modelers do the basic theoretical research, using assumptions to simplify the real world so that they can build mathematical models. The correctness of these models is validated against the original problem. Then computational scientists work on numerical algorithms, which are further developed by computer scientists. Again, the algorithms are tested against the mathematical models and the code is validated against the algorithms. There are two stages for correctness confirmation. First, the code should be validated against the mathematical model, which makes sure the implementation is a correct reflection of the model. Second, experiments are used to validate that the model embodied in the requirements is adequate for the intended use. If problems appear during the experimental validation this work flow recycles to the physical modelers and the same cycle is applied for the changes in the mathematical models.

In the above discussion, the terms validation and verification are used in the same sense that they are used in software engineering literature, which is different than the definitions sometimes used in the engineering computation literature. For instance, Roache [28, pages 19-36] reserves the word validation for experimental validation, but in the current

paper validation refers to the external validation of any phase in the software development process, which would be called verification by Roache. In the current paper the term verification refers to checking the internal properties, such as consistency and completeness, at each stage of the software development process and the "validation" of Roache's type is identified by the compound term "experimental validation".

There exists intrinsic similarity between the processes in Figure 1 and Figure 2; that is, each box and arrow in Figure 2 has a counterpart in Figure 1. This similarity motivates applying the innovative ideas from the discipline of software engineering to the field of engineering computation.

As stated in the introduction, another argument in favor of requirements documentation for engineering computation problems is that reliability can only be judged by comparison to explicitly stated requirements. Verification and validation (V&V) are difficult if it is unclear what standards the system is being verified and validated against. Clear requirements are necessary to know what the system should be inspected and tested for. Moreover, current V&V efforts focus on functional requirements, but the nonfunctional requirements, like accuracy, efficiency, portability etc. are as important and should also be tested.

## 4. SRS for Beam Analysis Software

The methodology for documenting the requirements for engineering computing problems is illustrated by the example of a system for analyzing beams. The central object in the beam analysis problem is a beam with two external forces and supports at ends $a$ and $b$, as presented in Figure 3. The beam, of length $L$, is in static equilibrium. The following information about the beam is given: the beam properties and the external loading ($F_1$ and $F_2$ located at positions and angles $x_1$, $x_2$, $\theta_3$ and $\theta_4$, respectively). The purpose of the system is to calculate the unknown applied forces or support reactions ($F_{ax}$, $F_{ay}$, $F_{bx}$ and $F_{by}$), the internal shear forces and bending moments, and the deflection of the beam. This software will be used as an educational tool for teaching statics and strength of material. Although the example is relatively simple there is enough complexity to illustrate the value of requirements documentation. The advantages discussed here should become even more pronounced as the complexity and size of the engineering computation problem grows.

The SRS template was constructed by borrowing ideas from the templates presented in Section 2.1.4, but it was also necessary to add original ideas to the new template. In some ways the new template is simpler than its predecessors. For instance, only one viewpoint needs to be considered for engineering software, but many different viewpoints need to be considered for business applications (e.g. the accounting viewpoint, the marketing viewpoint, etc.). Although different in the specific details, all engineering software can be abstracted as: input information then perform calculations and finally output the results. In business applications, on the other hand, the interaction of the system with the environment is typically more complex. Although the engineering computation SRS template is simpler in terms of the number of viewpoints than some templates, in other ways it is more complex. Certain issues are more important for an engineering computation domain, such as the

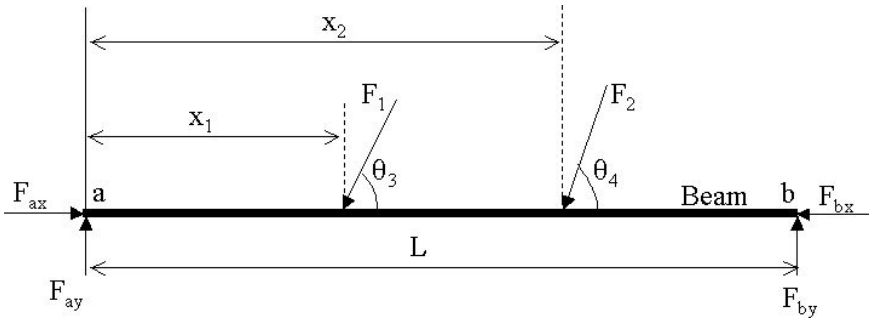



*Figure 3.* Free body diagram for beam with pin-pin supports

physics of the natural world, the sensitivity to the system of equations, the accuracy of the calculations, etc. How these issues are handled, along with other benefits of the proposed template, are examined in the subsections that follow. Each subsection discusses a different benefit of requirements documentation and then presents excerpts from the full SRS [21] to illustrate how the benefit is achieved.

### 4.1. The Template Provides Guidelines

The proposed requirements template supports and encourages a systematic process because it breaks the problem down into smaller steps and thus provides guidelines and a checklist for the issues that need to be addressed and the questions that need to be asked. The sections of the proposed template are shown in Figure 4. The sections encourage a systematic process by forcing the authors to consider each heading, even if the eventual decision is that the heading is inappropriate for a given problem. The division into sections, which is an example of applying the principle of separation of concerns, is an engineering approach to handling large and complex problems and for facilitating multidisciplinary collaboration. One example of separation of concerns is that the organization and purpose of the document (Sections 2.a and 2.c of the SRS template) are discussed separately from one another and from the rest of the document so that the discussion about the documentation itself will not complicate the discussion of the requirements.

Another example of separation of concerns is that the presentation of the functional requirements (Sections 4.a and 4.b of the SRS template) is separated from the presentation of the nun-functional requirements (Section 4.c of the SRS template). By separating functional (system behaviors) and non-functional requirements (overall system qualities) the analysis can focus on what the system is intended to do separately from thinking about what qualities the system should have. As an example, in the case study SRS, the decision that the system solves for unknown forces is separated from the requirement that, "The maximum response time of any interaction between the user and the system should be less than 1 second." The requirement to solve for unknown forces may exist in another systems, but in that other system the response time may be required to be faster than 1 second. Due to the separation of the two requirements, it would be straightforward to adapt the current

*Figure 4.* Table of Contents for the Beam Analysis SRS

documentation to reflect the needs of the new system. The clear separation of these two requirements improves the potential reusability of the documentation.

Another feature that improve reusability and make the document easier to maintain is the use of labels for cross-referencing. Whenever necessary the different parts of the SRS have their own label. The items that have labels include the following: the different sections of the SRS, the physical system descriptions (in SRS Section 4.a.iii), the goal statements (in SRS Section 4.a.iv), the assumptions (in SRS Section 4.b.i), the theoretical models (in SRS Section 4.b.ii), the instanced models of the system (in SRS Section 4.b.iv), and all of the tables and figures used in the SRS. As an example, physical system description PS1.a. says, "the shape of the beam is long and thin" and PS1.d says, "The transverse cross-section of the beam is rectangular". The use of labels allows the interrelationships between items in the SRS to be explicitly documented. Moreover, if there is a change in one portion of the document, it should be possible to determine what other portions of the document are affected, which improves reusability.

An additional feature of the proposed template that encourages a systematic approach and results in reusable documentation is the use of parameters instead of explicit values. Rather than say that the allowed range for $x_1$ is $0 \leq x_1 \leq 999999$, the range is written as $min_d \leq x_1 \leq max_d$. The actual values for these parameters are supplied in SRS Section 7, "Values of Auxiliary Constants". By not explicitly giving the values of the parameters in the body of the SRS, it is simple to change all occurrences of the values by changing the respective entry in SRS Section 7. This approach has the same advantages of using symbolic constants in a computer program: the value is easy to change and the symbolic name for the constant improves the documentation by being more meaningful than a number. Moreover, the use of parameters in this way encourages a systematic procedure. The discussion of

what value is appropriate for a constant can be separated from the decision that a variable should have some, as yet undetermined, limits on its value.

### 4.2. A Transition from General to Specific

The requirements template supports a systematic process by gradually taking the authors from general (abstract) to specific (concrete) concepts. For instance, as Figure 4 shows, the structure of the document proceeds from a general introduction to a specific system description. The "Introduction" section provides an overview of the entire SRS. After this the "General System Description" provides general information about the system, identifies the interfaces between the system and its environment, describes the user characteristics and system constraints. The next section, "Specific System Description", increases the level of detail and presents more concrete information. This section provides the physical system description, defines the system goals, presents a mathematical model of the system and documents the non-functional requirements. The structure of the template helps the SRS authors by allowing them to document the "big-picture" before thinking about the details.

The transition from general to specific also occurs with the refinement of the abstract system goals to a theoretical model and finally to a concrete instanced model of the system. The system goals state that given the beam properties and some of the external forces, the system should solve for:

**G1**. The unknown external forces applied to the beam;

**G2**. The functions of shear force and bending moment in the beam;

**G3**. The function of deflection along the beam.

These goals are refined further via a theoretical model that helps the reader to develop an understanding of the solution by introducing the theory and principles relevant to the problem. For instance, the goal **G1** is refined by the theoretical model for equilibrium, **T1**:

$$\textbf{(T1)} \quad \begin{cases} \sum F_{xi} = 0, \\ \sum F_{yi} = 0, \\ \sum M_i = 0, \end{cases}$$

where the forces and moments are all represented by their signed magnitude symbols: $F_{xi}$ represents the $i$th force component in the **x** direction, $F_{yi}$ represents the $i$th force component in the **y** direction, $M_i$ represents the $i$th moment component in the **z** direction. Similarly there are theoretical models **T2** and **T3**, which are detailed in [21], to refine the other two system goals.

To reinforce the fact that **G1** is more abstract than **T1**, it is worth mentioning that **T1** is not the only option for solving for the unknown forces. A theoretical model could be constructed that used the principle of virtual work, instead of using the equations of equilibrium.

The theoretical model alone does not provide enough information to solve for the unknown forces. It is necessary to introduce and define a more detailed model of the beam

Table II. Excerpt from Table for Solution of Unknown External Forces

| | | $H_1$ | |
|---|---|---|---|
| | | $S_{GET} = S_{sym} - S_{unkF}$ | $S_{GET} \neq (S_{sym} - S_{unkF})$ |
| $S_{unkF} \notin \mathbb{P}_3$ | - | $(ErrorMsg' = InvalidUnknown) \wedge ChangeOnly(ErrorMsg)$ | |
| $S_{unkF} = \{@F_{ax}, @F_{bx}, @F_{ay}\}$ | - | $ErrorMsg' = NoSolution \wedge ChangeOnly(ErrorMsg)$ | |
| $S_{unkF} = \{@F_{ax}, @F_{bx}, @F_{by}\}$ | - | $ErrorMsg' = NoSolution \wedge ChangeOnly(ErrorMsg)$ | $FALSE$ |
| $S_{unkF} = \{@F_{ax}, @F_{bx}, @F_1\}$ | - | $ErrorMsg' = NoSolution \wedge ChangeOnly(ErrorMsg)$ | |
| $S_{unkF} = \{@F_{ax}, @F_{bx}, @F_2\}$ | - | $ErrorMsg' = NoSolution \wedge ChangeOnly(ErrorMsg)$ | |
| $S_{unkF} = \{@F_{ax}, @F_{ay}, @F_1\}$ | $x_1 \neq 0 \wedge \theta_3 \neq 0 \wedge \theta_3 \neq 180$ | $F'_{ax} = \frac{-\cos\theta_3 F_2 x_2 \sin\theta_4 + \cos\theta_3 F_{by} L + F_2 \cos\theta_4 x_1 \sin\theta_3 + F_{bx} x_1 \sin\theta_3}{x_1 \sin\theta_3} \wedge F'_{ay} = -\frac{F_2 x_2 \sin\theta_4 - F_{by} L - F_2 \sin\theta_4 x_1 + F_{by} x_1}{x_1} \wedge F'_1 = \frac{-F_2 x_2 \sin\theta_4 + F_{by} L}{x_1 \sin\theta_3} \wedge ChangeOnly(S_{unkF})$ | |
| | *otherwise* | $(ErrorMsg' = Indeterminant) \wedge ChangeOnly(ErrorMsg)$ | |
| $S_{unkF} = \{@F_{ax}, @F_{ay}, @F_{by}\}$ | $L \neq 0$ | $F'_{ax} = F_1 \cos\theta_3 + F_2 \cos\theta_4 + F_{bx} \wedge F'_{ay} = \frac{-(-F_1 \sin\theta_3 L - F_2 \sin\theta_4 L + F_1 x_1 \sin\theta_3 + F_2 x_2 \sin\theta_4)}{L} \wedge F'_{by} = \frac{F_1 x_1 \sin\theta_3 + F_2 x_2 \sin\theta_4}{L} \wedge ChangeOnly(S_{unkF})$ | |
| | *otherwise* | $(ErrorMsg' = Indeterminant) \wedge ChangeOnly(ErrorMsg)$ | |
| $H_2$ | | $G$ | |

and the forces. In this case study the forces are represented as in Figure 3 and defined in SRS subsection "Data Definitions", with some of the forces decomposed into **x** and **y** components and others given as a magnitude and a direction. Another SRS subsection "Instanced Models" applies the theoretical models to the physical system. For instance, when moments are taken about point $a$, one instance of the model **M1** is:

$$\textbf{(M1)} \quad \begin{cases} F_{ax} - F_1 \cdot \cos\theta_3 - F_2 \cdot \cos\theta_4 - F_{bx} = 0 \\ F_{ay} - F_1 \cdot \sin\theta_3 - F_2 \cdot \sin\theta_4 + F_{by} = 0 \\ -F_1 \cdot x_1 \sin\theta_3 - F_2 \cdot x_2 \sin\theta_4 + F_{by} \cdot L = 0 \end{cases}$$

As was the case for the transition from **G1** to **T1**, there are other concrete options available for specifying **M1**. One of the advantages of the proposed template is that it supports reuse by allowing a new theoretical or instanced model to be introduced. Any of the documentation associated with the more abstract model or goal can remain unchanged in the new version of the SRS.

4.2.1. *Special Cases are Considered*
The proposed requirements template provides a systematic approach for eliciting and documenting requirements, which improves the user's confidence that all special cases have been considered, where special cases are defined as those exceptional cases that differ from the normally expected behaviour. In particular, the use of tabular expressions aids in detecting special cases, as shown by Table II. This table shows how the exceptions of divide by zero errors can be avoided by carefully documenting what input values cause problems for which cases. Tables have the advantageous property that they can be mathematically verified to ensure that the space that they cover is complete.

Table II is an extract from the full table, given in [21], that shows the system behaviour for solving for unknown forces, given the correct number of known forces. Only the statically determinant beam is considered by the system, thus the number of unknowns should be the same as the number of equilibrium equations; that is, there should be exactly 3 unknowns. The valid unknown force space can be defined by $\mathbb{P}_3$, where $\mathbb{P}_3 \triangleq \{S | S \subseteq S_F \wedge \sharp S = 3\}$ with $S_F$ being a set of symbols of force variables: $\{@F_{ax}, @F_{ay}, @F_{bx}, @F_{by}, @F_1, @F_2\}$. The @ symbol is used to distinguish between the symbol for a variable and the actual value of the variable. The $\sharp$ symbol is a unary operator that is applied to a set and returns the set's cardinality. The user of the system should specify 3 known forces from the set $S_F$ and the system will solve for the remaining 3. The solutions for each element in the set $\mathbb{P}_3$ ($C_6^3 = 20$ situations) are specified in [21]. (An advantage for this problem is that the closed-form solution is available to be analyzed before design.) Some of the cases may lead to an infinite number of solutions, which come from statically indeterminant beams. Since the system is only interested in the cases that have a unique solution, the indeterminant situations should be avoided. Table II, which shows a portion of the full table, documents how the forces are solved for some of the cases. Table II uses $S_{unkF}$ to represent a set of symbols of the unknown force variables: $S_{unkF} \in \{S | S \subseteq S_F\}$. In Table II, $S_{sym}$ is the set of symbols of all user interface variables that are either inputs or outputs of the system. The table specifies that all symbols, except those in $S_{unkF}$, must be in set $S_{GET}$ before the calculations proceeds. As Table II shows, divide by zero errors can be avoided, by checking the predicates in the second column of $H_2$.

## 4.3. Catalyzes Consideration of Issues Before Design

The requirements template not only encourages a systematic process for collecting, analyzing and documenting the requirements, it also improves the systematic application of the scientific method shown in Figure 2. There are sections in the requirements template that encourage the analysts to answer questions that will pay significant dividends during the design stage. For instance, the requirements template has a section (SRS Section 4.c.ii) where the sensitivity of the model is considered. The SRS template encourages the analyst to consider the sensitivity of the numerical model in advance of coding. If the model is very sensitive to input data errors, then it may not be worth constructing, or it may be necessary to approach the problem in a different manner. For instance, if the bean considered in this case study experiences a high axial load, then buckling may occur. In the current case study

an assumption is made that buckling failure will not occur, but if this is an important issue, according to the range of parameters permitted by the model, then a buckling analysis may be appropriate. The systematic presentation that tables provide is also helpful for considering the sensitivity of individual calculations. For instance, in Table II, the condition number of each explicit equation in the grid could be calculated to provide an estimate of the sensitivity of the calculations.

Besides the section on sensitivity, there are other sections of the SRS template (Figure 4) that encourage the analyst to think about important issues in advance of the design. These sections are all in the "Non-functional Requirements" section of the template. The achievement of non-functional requirements often involve trade-offs between system qualities, such as speed efficiency and accuracy, so it is valuable to start the discussion on these trade-offs early in the process. If adequate effort is placed into documenting what behaviour is expected of the system, then it should be easier to judge if the results produced by the system are reasonable. One particular requirement that the analyst should think about in advance of solving the problem is the tolerance allowed in the solution (Section 4.c.iii in the proposed SRS template). For the case study SRS the tolerance is specified by equations of the form $|\sum F_{xi}|/\sqrt{\sum {F_{xi}}^2} \leq \epsilon$, where $F_{xi}$ is the $i$th force component in the $\mathbf{x}$ direction and $\epsilon$ is the allowed tolerance. Another valuable set of pre-design non-functional requirements involves the SRS section "Solution Validation Strategies". Some possible validation strategies for numerical solutions,which are given in the case study SRS, include: solving the problem by different techniques (such as electronic spreadsheet or a graphical solution), and substituting the results back into the model to see if the formula is still satisfied.

A good design should take into account the likely changes that the system may undergo in the future. This is accommodated in the proposed template by Section 6 of the SRS, "List of Possible Changes in the Requirements". Some potential changes to the case study SRS include incorporating more than two applied forces, considering beams with other types of supports and considering the self-weight of the beam. The system designer will be able to use this information to produce a system that can potentially have a long life because it will be able to evolve to accommodate the likely future changes.

### 4.4. Reduces Ambiguity

One significant benefit of an SRS for an engineering computation problem, such as the beam analysis problem, is that it can reduce the ambiguity of the requirements. By explicitly documenting the requirements and by formalizing some of them, it becomes much easier for different experts to communicate, a review of the requirements is possible, and the eventual designer will not have to make arbitrary judgments about the system's required behaviour. One approach to making the SRS unambiguous is to use tabular expressions. As Table II shows, tabular expressions clearly specify the required behaviour for all cases when they are complete, which is a property that can be automatically verified. The SRS also has sections that are devoted to providing the necessary details to make the problem unambiguous. For instance, important reference material is given in the SRS sections "Table of Symbols" and "Abbreviations and Acronyms".

Two other sections that reduce ambiguity are the SRS sections "Scope of the Software Product" and "System Context". These sections are important because they respectively delineate what the system does and how the system fits into its external environment. For the case study, the scope is solving for unknown forces, shear, bending moment and deflection of a beam, but the documentation specifies that the system does not solve for internal axial forces. The scope also makes it clear that the system will be used for educational purposes. The system context section, on the other hand, documents the user responsibilities, such as preparing input information and using consistent units, versus the system responsibilities, such as detecting data type mismatch and determining if the inputs satisfy the required constraints.

When developing a new system, it is important to have the characteristics of the future users in mind. A beam analysis program will likely be designed differently by a practicing engineer versus a high school student. In the current case study, the decision was made, as documented in the SRS section "User Characteristics", to assume that the users of the system are first or second year university students in science or engineering.

An SRS is not meant to be read sequentially, instead it is a reference document that will be searched for specific pieces of information. For instance, a reader or reviewer may search for the definition of a particular term. Without the definition of the term, the reader will not know how to interpret the documentation. For this reason there are two sections in the SRS devoted to definitions: "Terminology Definition" and "Data Definitions". The terms defined include applied force, bending moment, deflection, equilibrium, free-body diagram, longitudinal centroid plane, magnitude of a vector and Young's modulus. The data definitions are used to define the mathematical variables that model objects in the physical system. For instance, there are definitions for the coordinate system, the dimension system, the beam, the reactive forces, the moments and the shear forces. These definitions remove ambiguity by giving a meaning to the symbols and by defining potentially confusing details, such as the sign conventions.

The advantage of unambiguous requirements is that they can end arguments about different designs. Practitioners argue over the relative merits of different designs based on their own implicit requirements. A developer may criticize a design because it is not efficient, but this criticism would not be justified if the designer clearly started out with requirements that clearly stated that precision, maintainability and portability are more important than efficiency. The specific trade-offs depend on the scope of the system. For instance, in the beam analysis case study the requirements for speed and accuracy do not need to be that strict because the system is not intended for a safety critical setting, but rather an educational one.

### 4.5. Range of Model Applicability is Identified

A significant benefit of appropriate and rigorous requirements documentation is that the range of the physical model's applicability can be clearly identified. One way that this is done is by documenting the assumptions that the model is based on, as discussed in the next section. Another way that the range of applicability is identified is by explicitly constraining

the input data. The input data should be constrained so that it is physically meaningful. In the SRS section "Data Constraints", physical constraints are applied to all variables, as appropriate. For instance, the location $x_1$ of the applied load $F_1$ is required to be in the range $0 \leq x_1 \leq L$ because any other value is physically meaningless. Another constraint, which is added to maintain the physical system description that the beam is long and thin (PS1.a), is that $0 < h \leq 0.1L$.

Other constraints are added to the input data as system constraints, which are constraints that are not motivated by physics, but rather by the range of values it is reasonable to expect in practice. Table I gives an example of system constraints on the variable $x_1$. The system also constrains the angle $\theta_3$ so that $0 \leq \theta_3 \leq 180$. This is not necessary for the physics of the problem, but allowing this range of angles, and a signed magnitude for the force, allows one to enter any possible loading, so the system restriction simplifies the user input, without putting any constraints on the range of problems the system can solve. In [21] the specification of the system behaviour for each input variable is given, along with a new tabular composition operation that allows the specifications to be combined, while still maintaining the property of domain coverage.

Restricting the input to the system to reasonable values can potentially avoid error cases and it can streamline the subsequent design stage. One of the hardest challenges in engineering computation is to devise algorithms for a general case. Why should a designer have to face this challenge, if information is known about the problem that will mean that only specific cases will ever occur? For instance, numerical problems could occur in the beam problem if the external forces have significantly different magnitudes, but this is not likely to occur in an engineering problem because the effect of the smallest magnitude forces would be negligible and typically not even modeled by an engineer. For this reason, the case study SRS has constraints on all of the forces of a form like the following constraint on $F_{ax}$: $(min_f \leq |F_{ax}| \leq max_f) \wedge (|F_{ax}| \neq 0) \wedge \forall(FF|@FF \in S_F \cdot FF \neq 0 \wedge \frac{max\{|F_{ax}|,|FF|\}}{min\{|F_{ax}|,|FF|\}} \leq 10^{r_f})$, where $min_f$ and $max_f$ are the system constraints for the minimum and maximum magnitude forces and $r_f$ is a positive integer that is the maximum exponent of base 10 for the ratio between the magnitudes of the largest and smallest forces.

Documenting the range of applicability of the model is also important for engineering computation problems other than the beam analysis problem. For instance, it is difficult to write code to solve any system of equations $Ax = b$, but this job becomes easier if $A$ is known to have special characteristics, such as being symmetric positive definite. Therefore, the requirements documentation should clearly show the restrictions on the data and the theory that will lead to the simplifications.

### 4.6. Clear Documentation of Assumptions

In engineering computation it is often the differing assumptions that distinguish one piece of work from another. Assumptions are necessary to build a physical model of the real world. Often the quality of the model depends on how reasonable the simplifying assumptions are. Given the importance of assumptions, an SRS for engineering computation problems should clearly label and document all assumptions, as in Section 4.b.i of the proposed template.

The assumptions defined in this subsection simplify the original problem and fill in missing information for the physical system so that a theoretical models can be developed and be properly applied. Some sample assumptions for the beam analysis SRS, using the numbering given in [21], include the following:

**A1.** The physics for this problem are in the field of Newton's classical mechanics.

**A2.** Thermal effects are neglected.

**A4.** The weight of the beam is neglected.

**A8.** Only the statically determinant cases for the beam are considered.

**A9.** Beam deformations are small compared to the original dimensions. Thus the slope of the beam's deflected shape is small compared to the unity, and the length variation of the beam in the longitudinal direction is neglected.

**A10.** The deflection of the beam is caused by bending moment only, the shear does not contribute.

**A14.** The second moment of area along the length of the beam is constant.

Since different models are distinguished by their differing assumptions, a clear indication of the dependence of the model on its assumptions will greatly facilitate reuse and future evolution of a given SRS. When the assumptions change the model generally changes, but, very often, some parts will stay the same and by identifying those parts one can easily reuse them. In the proposed SRS template the interdependence of the different components of the specification is documented by a traceability matrix, which is in the SRS Section 5. For the traceability matrix to be meaningful it is important that the assumptions be independent of one another.

The traceability matrix gives a "big picture" view of the relationships between sections "Physical System Descriptions", "Goal Statements", "Data Definitions", "Assumptions", "Theoretical Models" and "Instanced Models" in the case study SRS. These sections of the SRSs are shown because they contain the most essential information of the system functionalities. As an example, the relationship between "Goal Statements" and "Theoretical Models" can be refined by a specific relationship between one concrete goal statement and one concrete theoretical model, for instance **G1** and **T1**. These relationships are represented by ticks "$\surd$" in the cells of the matrix, as in Table III, which is a portion of the table presented in [21]. The physical system descriptions and the goals in the first column of the matrix are used by the data definitions or the theoretical models in the second column; the assumptions in the first row are used by the theoretical models, the instanced models and the data definitions in the second column; the theoretical models and the data definitions in the second column are used by the instanced models in the first row.

The traceability matrix of the "deflected beam problem" presented in this paper can be compared with the simpler case of a "rigid beam problem" to illustrate how requirement reuse can be implemented with the help of a traceability matrix. Compared with the deflected beam, the rigid beam has a new assumption:

Table III. Traceability Matrix for Deformed Beam

| Phy. Sys. /Goal | Data /Model | Assumption | | | | | | | | | | Model | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | A1 | A2 | ... | A4 | ... | A8 | A9 | A10 | ... | A14 | **M1** | ... |
| **G1** | **T1** | √ | | ... | | ... | √ | √ | | ... | | √ | ... |
| **G2** | **T2** | √ | | ... | | ... | √ | √ | | ... | | | ... |
| **G3** | **T3** | √ | | ... | | ... | | √ | √ | ... | | | ... |
| | **M1** | | √ | ... | | ... | | | | ... | | √ | ... |
| PS1.a | $L$ | | | ... | | ... | | | √ | ... | | √ | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

**A15.** The beam behaves as a rigid body.

This new assumption will require several changes to the SRS for the deflected beam. The modifications can be guided by the traceability matrix. To see this, one can consider how the flexible beam traceability matrix can be modified to obtain a traceability matrix for the rigid beam problem. The addition of the rigid body assumption means that the previous assumptions **A9 - A14** are no longer necessary. The new system does not require system goals **G2** and **G3**; therefore, theoretical models **T2**, and **T3** will be removed from the row headers of Table III. Removing the columns representing the unnecessary assumptions and the rows of the unnecessary models and symbols results in a traceability matrix that reflects the requirements specification of the "rigid beam problem" exactly. The details of this transition can be found in [21].

Besides supporting a transition to a simpler model, the traceability matrix also supports a move to a more complex model. For instance, assumption **A10** could be removed, which would mean that shear would also contribute to deflections and a more complex theory, such as Timoshenko beam theory [35, pages 224-230] would be required. In the transition to the more complex system the traceability matrix would guide the analyst as to what sections of the documentation need to be added and/or modified. For instance, Table III shows that the theoretical model **T3**, which is the model for simulating the deflection, will need to be changed. Also, PS1.a, which says that the "beam is long and thin" is no longer necessarily true as Timoshenko beam theory also applies to short and thick beams.

## 5. Concluding Remarks

This paper motivates, justifies and illustrates a method of writing requirements specifications for engineering computation problems that will improve their reliability. The motivation comes first from the fact that reliability of a system can only be accurately judged if there is an unambiguous statement of the behaviors and qualities that the system is required to have. Although there are many excellent numerical libraries and packages that implement accurate and efficient algorithms, the selection of an appropriate algorithm

is often based on implicit requirements. If explicit, appropriate and rigorous requirements documentation is available, then the overall quality of the design can be improved, not just with respect to reliability, but also with respect to usability, verifiability, maintainability, reusability and portability, which are sometimes neglected qualities in engineering software. Further motivation for the appropriateness of software engineering methodologies, such as requirements analysis, for engineering computation, comes from the fact that the waterfall model of software development closely parallels the usual model of the scientific method.

The justification for using the proposed template for engineering computation comes from observing how it supports and encourages a systematic process by providing guidelines, by providing a smooth transition from general to specific details, by increasing confidence that all special cases have been considered, and by encouraging the analyst to scrutinize their problem in advance of designing the computational system. Further justification for using an SRS for engineering computation comes from the benefit of reducing ambiguity, clearly identifying and documenting the range of model applicability and rigorously documenting the assumptions that simplify the real world to the point where theoretical and instanced models can be constructed. To improve the systematic process and to reduce ambiguity, this paper advocates the use of tabular expressions, which provide mathematical rigor, but at the same time have the benefit that they can be easily and intuitively understood.

To illustrate the proposed method of documenting requirements for engineering computation problems, this paper presents a case study for the analysis of a statically determinant beam. Although this problem is relatively simple, the findings of this study can be generalized because many engineering computation problems follow the same pattern. For instance, the scientific method is appropriate for most engineering problems and engineering software can often be abstracted by the following simple model: input information then perform calculations and finally output results. Moreover, the calculation step is similar between many engineering computation problems because it involves solving some given set of governing equations together with appropriate boundary and/or initial conditions. Given the similar pattern between the beam problem and other engineering problems, the method presented in this paper can be applied to larger and more complex problems. The advantages of the current method will greatly increase as the size and complexity of the problem grows because the value of a systematic approach increases with the number of details and the number of people involved.

## Acknowledgements

The financial support of the Natural Sciences and Engineering Research Council (NSERC) of Canada is gratefully acknowledged.

## References


1. Åhlander, K., M. Haveraaen, and H. Kaas: 2001, 'On the Role of Mathematical Abstractions for Scientific Computing'. In: *The Architecture of Scientific Software*. Boston, pp. 145–158. ISBN 0-7923-7339-1.
2. Berry, D.: 2002, 'Formal Methods: the Very Idea'. *Science of Computer Programming* **42**, 11–27.
3. Blilie, C.: 2002, 'Patterns in scientific software: an introduction'. *Computing in Science and Engineering* **4**(3), 48–53.
4. Davis, A. and et al.: 1993, 'Identifying and measuring quality in a software requirements specification'. In: *Proceedings of the 1st International Software Metrics Symposium*. pp. 141–152.
5. Davis, A. M.: 1993, *Software Requirements: Objects, Functions, and States*. Prentice Hall PTR, revision edition.
6. Desharnais, J., R. Khedri, and A. Mili: 2003, 'Representation, Validation and Integration of Scenarios Using Tabular Expressions'. *Formal Methods in System Design*. Accepted for publication (Final version submitted on July 02, 2002).
7. DoD: 1994, 'Software development and documentation, DoD Military Standard MIL-STD-498'. Technical report, US Department of Defense, Washington, D.C.
8. Dubois, P.: 2002, 'Designing Scientific Components'. *Computing in Science and Engineering* **4**(5), 84–90.
9. ESA: February 1991, 'ESA Software Engineering Standards, PSS-05-0 Issue 2'. Technical report, European Space Agency.
10. Galin, D.: 2004, *Software Quality Assurance: From theory to implementation*. Pearson Education Limited.
11. Gerlach, J.: 2002, 'Domain Engineering and Generic Programming for Parallel Scientific Computing'. Ph.D. thesis, Von der Fakultät IV - Elektrotechnik und Informatik der Technischen Universität Berlin.
12. Ghezzi, C., M. Jazayeri, and D. Mandrioli: 2003, *Fundamentals of software engineering*. Prentice Hall, 2nd ed. edition.
13. Grundy, J.: 1999, 'Aspect-oriented Requirements Engineering for Component-based Software Systems'. In: *Requirements Engineering, 1999. Proceedings. IEEE International Symposium on*. Limerick, Ireland, pp. 84–91.
14. Hannemann, R., J.H., M. Zellerhoff, and L. Klinkenbusch: 2001, 'Scientific Programming in Field Theory, Part 1'. *IEEE Computing in Science and Engineering* **3**(3).
15. Heninger, K., J. Kallander, D. Parnas, and J. Shore: 1978, 'Software Requirements for the A-7E Aircraft'. Nrl memorandum report, United States Naval Research Laboratory, Washington, D.C.
16. Heninger, K. L.: January 1980, 'Specifying Software Requirement for Complex System: New Techniques and Their Application'. *IEEE Transactions on Software Engineering, vol. 6, no. 1, pp. 2-13.*
17. IEEE: 1998, *Recommended Practice for Software Requirements Specifications, IEEE Std. 830*. IEEE.
18. Janicki, R. and R. Khedri: 2001, 'On a formal semantics of tabular expression'. *Science of Computer Programming* **39**(2-3), 189–213.
19. Kaindl, H., S. Brinkkemper, J. B. Jr., B. Farbey, and S. Greenspan: 2002, 'Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda'. *Requirements Engineering Journal* **7**(3).
20. Kreyman, K. and D. L. Parnas: 2002, 'On Documenting the Requirements for Computer Programs Based on Models of Physical Phenomena'. SQRL Report 1, Software Quality Research Laboratory, McMaster University.
21. Lai, L.: 2004, 'Requirements Documentation for Engineering Mechanics Software: Guidelines, Template and a Case Study'. Master's thesis, initial draft for McMaster University, Hamilton, Ontario, Canada.
22. Mylopoulos, J., L. Chung, and E. Yu: 1999, 'From object-oriented to goal-oriented requirements analysis'. *Communications of the ACM archive* **42**(1), 31–37.
23. NASA: 1989a, 'External interface requirements DID, SMAP-DID-P210, Release 4.3'. Technical report, National Aeronautics and Space Agency.

24. NASA: 1989b, 'Software requirements DID, SMAP-DID-P200-SW, Release 4.3'. Technical report, National Aeronautics and Space Agency.
25. Parnas, D. and J. Madey: 1990, 'Functional Documentation for Computer Systems'. *Colloque surl'inge'nierie de la qualite, Le Centre, Montreal* p. 7.
26. Parnas, D. L., G. J. K. Asmis, and J. Madey: 1991, 'Assessment of Safety-Critical Software in Nuclear Power Plants'. *Nuclear Safety* **32**(2), 189–198.
27. Parnas, D. L. and P. Clements: February 1986, 'A Rational Design Process: How and Why to Fake it'. *IEEE Transactions on Software Engineering, vol. 12, no. 2, p.251-257.*
28. Roache, P. J.: 1998, *Verification and validation in computational science and engineering.* hermosa publishers.
29. Robertson, S. and J. Robertson: 1999a, *Mastering the Requirements Process.* ACM Press.
30. Robertson, S. and J. Robertson: 1999b, *Mastering the Requirements Process*, Chapt. Volere Requirements Specification Template, pp. 353–391. Addison-Wesley Educ. Publ. Inc.
31. Sanga, B.: 2003, 'Assessing and Improving the Quality of Software Requirements Specification Documents (SRSDs)'. Thesis for M.Sc. program, Computing and Software Department, McMaster University, Hamilton, ON.
32. SciDAC: 2000, 'Scientific Discovery through Advanced Computing'. world wide web: http:// www.osti.gov/ scidac/ (accessed June, 2003). Office of Science, U.S. Department of Energy.
33. Sommerville, I. and P. Sawyer: 1997, *Requirement Engineering: A Good Practice Guide.* John Wiley & Sons Ltd. ISBN 0-471-97444-7.
34. Thayer, R. H. and M. Dorfman (eds.): 2000, *IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, Washington, DC, USA, 2nd ed. edition.
35. Timoshenko, S. and D. H. Young: 1968, *Elements of Strength of Materials.* D. Van Nostrand Company, Inc. 5 edition.
36. van Lamsweerde, A.: 2001, 'Goal-oriented requirements engineering: a guided tour'. In: *Proceedings of the 5th IEEE International Symposium on Requirements Engineering.* pp. 249–263.

# Activity Networks and Uncertainty Quantification: 2nd Order Probability for Solving Graphs of Concurrent and Sequential Tasks

Daniel Berleant, Jianzhong Zhang, and Gerald Sheblé
*Department of Electrical and Computer Engineering, Iowa State University, 2215 Coover Hall, Ames, Iowa 50011*
*e-mail: berleant@iastate.edu*

**Abstract**. Activity networks model the time to project completion based on the times to complete various subtasks, some of which can proceed concurrently and others of which are prerequisite to others. Uncertainty in the times to complete subtasks implies uncertainty in the overall time to complete the project. When the information about the times to complete subtasks is insufficient to fully specify a probability distribution but sufficient to bound the distribution, the problem of making conclusions about time to complete the entire project requires use of second-order probabilistic techniques. An interval-based technique for this is described, and applied to the problem of evaluating activity networks.

## 1. Introduction

**The Problem**. Determining the completion time of activity networks is of importance to engineering project management, and is the subject of an extensive and expanding body of works. Forecasts for activity durations must often be estimates, since the execution of an activity typically depends on various factors whose details are not knowable in advance. This leads naturally to modeling durations of activities with, for example, probability distributions. However, determining the distribution of the completion time of the entire network can then be non-trivial. Addition and maximization are typical algebraic operations on random variables occurring during evaluation of activity networks (Agrawal and Elmaghraby 2001). Distributions must be found for the sums of random variables whose distributions describe the durations of activities on a given path. Also, various paths may each have some chance of being the critical one, depending on the summed times of the activities comprising each path. This requires calculating distributions of the maximums of random variables, because when computing the time to complete concurrent tasks, the joint completion time is the maximum of the completion times of the concurrent tasks.

A further challenge is posed by the need to model the dependency relationships among the duration distributions of the various activities. A complete solution to the distribution of the network completion time would, in general, require specification of a multivariate joint distribution with one marginal for each activity duration distribution.

**Modeling activity networks**. In order to solve networks a variety of simplifying assumptions have been proposed. The most drastic of these is to model activity duration as numbers. The network completion time is then the completion time of the critical (longest duration) path through the network. However, this removes uncertainties that are essential to account for in understanding important properties of the network, such as risk of project delay and the consequent financial and other consequences. Hence it is better to retain distributions as representations of individual task durations. This suggests a less drastic simplification, namely statistical independence.

One type of independence assumption applies to understanding the completion time of a single task. A *factor* is something that contributes to the uncertainty in an activity duration such

as weather conditions (for construction projects), labor variabilities, etc. It is typically considered reasonable to assume that the factors contributing to completion time of a particular task act independently. Summing them thus involves determining the sum of independent random variables. Agrawal and Elmaghraby (2001) propose an algorithm and review another early algorithm proposed by Martin (1965).

The assumption that completion time distributions of different *activities* are independent has been the basis of considerable work. Robillard and Trahan (1977) show how activity network completion time distributions can be approximated efficiently under this assumption. They derive lower and upper bounds for the mean and variance of the completion time distributions. Kleindorfer (1971) bounds the time to complete the activity network with lower and upper bounding distributions under the same independence assumption. Kamburowski (1985) provides an upper bound on the expected project completion time for independent activity duration distributions, each a member of a large class of distributions. A more recent algorithm was proposed by Schmidt and Grossmann (2000) to obtain the distribution of the project duration under the independence assumption. A general-purpose algorithm for arithmetic operations on independent random variables is described and some previous algorithms are noted in Berleant (1993). The maximization operation is a particularly simple case since the cumulative probability that two tasks will be completed by any given time is the product of the probabilities of completion by that time for each task.

The problem with independence assumptions is that the completion times for different tasks are often not independent. For example, frequently tasks share factors that tend to affect the tasks' completion times similarly. Shipbuilding is an example of a domain where correlations are important (van Dorp and Duffey 1999). Thus it is important to consider how activity networks can be analyzed when the individual task completion time distributions are not independent.

Ahuja and Nadakumar (1985) and Padilla and Carr (1991) capture information about correlations among different activity completion times by identifying individual factors that affect the rate of progress across multiple activities. Examples of factors include weather, legal issues, environmental issues in construction projects, variability due to labor, etc. Each sample drawn from the distribution of such a factor affects the simulated duration of a number of activities. Woolery and Crandall (1983) allow effects of factors to vary over time. For example, weather may impose more uncertainty during some times of the year than others. Levitt and Kunz (1985) examine whether the actual completion times of tasks were lower or higher than expected, and adjust the projected completion times of future tasks that share factors with completed tasks whose completion times deviated from expectations. Wang and Demsetz (2000) propose an elicitation method for using expert judgements to estimate distribution functions for factors affecting task completion time. Tasks that share factors therefore have correlated durations. In the solution offered in this paper, each individual task completion time may be described with a number, an interval, or a distribution function.

In the case of distribution functions, two task completion times might be independent random variables, as when the tasks are performed in different environments and proceed independently. Alternatively, completion times might be positively correlated, as when both depend on the quality of management and proceed within the same managerial environment. Or, they could be negatively correlated, as when resource sharing means that faster completion of one implies slower completion of the other. Finally, various factors might interact to make completion times dependent in ways the details of which are lost by merely stating the amount of correlation. The solution offered in this paper avoids requiring the assumption that individual task completion times are independent or have any other dependency relationship. Project management is just one application of activity networks and, hence, of the technique described.

**Solving activity networks**. Diaz and Hadipriono (1993) compare five methods of activity network evaluation, finding significant differences among the results. For example, PERT tended

to give more optimistic estimates of project time overrun than the other four methods, and differences between Monte Carlo simulation methods and the other methods tended to be exacerbated by use of asymmetric distributions for activity durations because the details of the asymmetries were not captured by the other methods. The Narrow Reliability Bounds (NRB) method (Ditlevsen 1979) frequently does not contain the results of the other methods between its bounds.

The **t**ime to **c**omplete the **j**ob (TCJ) can be addressed analytically, numerically, or by simulation. Analytical approaches generally rely on approximations and/or simplifying assumptions, such as that distributions of individual task completion times are normal, or that moments of distributions characterize all that is necessary to know about them (e.g. Mehrotra et al. 1996).

Simulation and numerical methods pose fewer needs for such simplifications, but present other tradeoffs. Thus, it can be harder to accomodate potentially important aspects of a model within a numerical method than with simulation. An example is that combinatorial explosion favors simulation over numerical methods for certain problems involving multiple marginals. On the other hand, simulation has disadvantages relative to numerical methods as well. The best known is perhaps the risk of unreliable results due to insufficient iterations. Monte Carlo simulation in particular has several other problems as well (Ferson 1996). One potentially significant problem with simulation is the difficulty it can have in handling models where the distributions of random variables are incompletely known or are partially dependent on one another. For example, correlation values do not fully define a dependency since, in general, many different joint distributions may have the same value for correlation (Berleant and Zhang 2004(a)). Given a correlation value as an input, it is difficult in a simulation to avoid assuming just one dependency that satisfies the given correlation. Analysts may make such an assumption without noting this or even realizing it (Ferson et al. 2004).

The proper outcome of such a model specification that accounts for unknown dependencies is that distributions resulting from an analysis, such as sums or maximums of other distributions, cannot be fully specified. They can however be bounded with envelopes around the space within which the distribution curves must be. With simulation it is difficult to deal with envelopes because, since the distribution is not fully specified, it is not clear how to generate samples of the random variable in question. This issue can be circumvented by using appropriate numerical methods. One such method, the DEnv algorithm (Berleant and Zhang 2004(b)), is the basis of this paper.

**Summary**. Assumptions help in simplifying problems, but can be risky when not properly supported as results can be significantly affected. Validity of assumptions is therefore an important issue and one that has motivated considerable concern. Therefore it is important to consider how the computation of a distribution for activity network completion time can be affected by these assumptions. This report addresses that issue by accounting for such $2^{nd}$-order uncertainties by avoiding the requirement that dependencies among activity durations be specified. Results show that networks can exhibit a range of completion time distributions consistent with the input data. This illustrates the importance of assumptions in two ways: (1) that they often must be made in order to get results of acceptable specificity, and (2) that they must be made reliably to get reliable results.

## 2. Approach

Determining the time to complete all tasks in a network of tasks is easy when the time to complete each individual task in the network has a numerical value, harder when individual completion times are described using probability distributions, and still more challenging when these distributions are neither assumed independent nor assumed to have any other dependency

relationship. A method is described here for determining completion times of task networks in the last case. We begin by describing each task completion time with a probability distribution function, noting that this includes as a special case a completion time described with a precise number since a number may be represented as a step distribution function (Figure 1, *left*). We later generalize to the case of left and right envelopes enclosing a family of cumulative distribution functions (CDFs) which, as a special case, allows a completion time to be represented as an interval describing a range of plausible values with high and low bounds but no information about the probability distribution within those bounds (Figure 1, *right*).

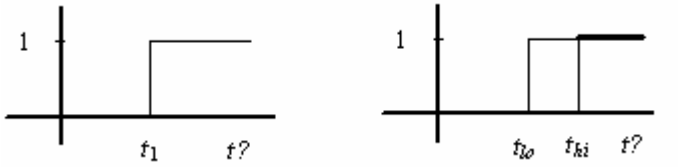



**Figure 1. (*Left*) the numerical value of time $t_1$ is a special case of a cumulative distribution function (CDF) which is 0 below $t_1$, and 1 at $t_1$ or above. (*Right*) an interval $[t_{lo}, t_{hi}]$ is a special case of a family of distributions containing any CDF which is 0 below $t_{lo}$ and 1 (at or) above $t_{hi}$.**

In real situations, two task completion times might be independent random variables, as when each is done in a different environment and they proceed independently. Alternatively, completion times might be highly positively correlated, as could occur if the tasks depend on the quality of management and proceed within the same managerial environment. As a third possibility, completion times might be quite negatively correlated, as could occur if the tasks proceed concurrently with shared personnel or other resources and faster completion of one entails slower completion of the other. A final and quite likely possibility is that various factors interact to make completion times dependent in a way that is difficult to characterize accurately. Therefore in the general case we wish to avoid assuming that individual task completion times are independent or have any other particular dependency relationship. A solution to this general case is offered.

The results have application to project management, where task completion time analyses can be useful as illustrated by the well-known PERT (Program Evaluation and Review Technique) method.

## 3. Solution for the case of two concurrent tasks

This section discusses the case of two concurrent tasks. Generalization to larger networks of tasks is discussed in Section 3.

Consider concurrent tasks X and Y, each beginning when the task environment is in a start state S and whose joint completion brings about desired finish state F (Figure 2). Let $F_x$ be the CDF of random variable $t_x$, the completion time of task X, and let $F_y$ be the CDF of random variable $t_y$, the completion time of task Y. We begin by reviewing solution strategies when $t_x$ and $t_y$ are independent, and then generalize by removing the independence assumption.

One solution strategy is the analytical one. The analytical approach to arithmetic on random variables is limited in the forms of the distributions it can handle and usually relies on the assumption of independence (e.g. Springer 1979). The Monte Carlo approach is a numerical strategy that does not produce definite bounds, does not handle cases where one operand is a CDF and the other an interval except under severe restrictions, does not handle the case of unknown dependency between random variables, and has other limitations (Ferson 1996). Numerical convolution (Ingram et. al 1968; Colombo and Jaarsma 1980) is an alternative numerical strategy that allows arithmetic operations to be applied to random variables with a wide variety of CDFs,

and has been extended to capture discretization error via error bounds that propagate through the calculations and lead to left and right envelopes around the true solution (Williamson and Downs 1990; Berleant 1993; Cooper et al. 1996). See Figure 3.

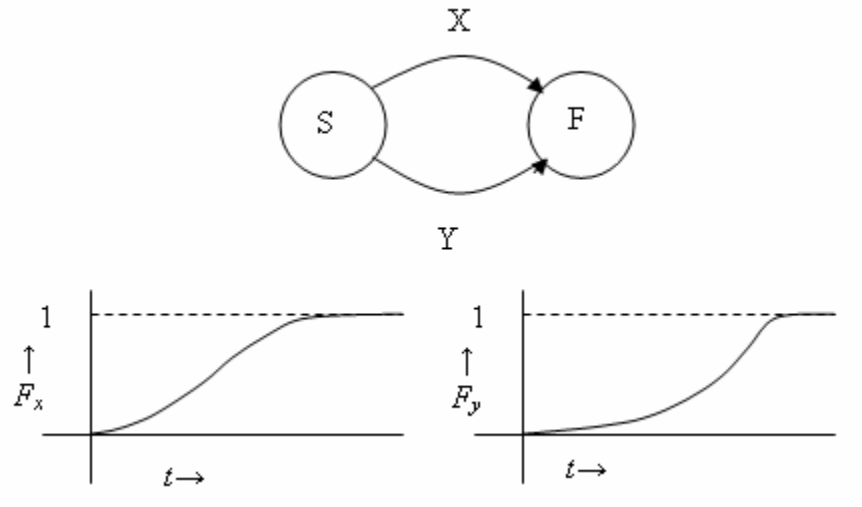



**Figure 2. PERT diagram showing a starting state S, a finish state F, and two tasks X and Y that must be completed to reach state F. Two different distribution functions $F_x$ and $F_y$ describe random variables $t_x$ and $t_y$, which represent the completion times of tasks X and Y.**

Envelopes consist of non-crossing CDFs that enclose the paths of all CDFs consistent with the problem. These envelopes are often called *probability bounds* (Ferson et al. 2002) and, because they do not cross, the right envelope has *first order stochastic dominance* over the left (Levy 1992). Coarse discretizations for random variables $t_x$ and $t_y$ (e.g. Figure 3) lead to correspondingly large discretization error and therefore more widely spaced left and right envelopes. Finer discretizations would result in left and right envelopes that have more and smaller steps and are closer together. The CDF for result $t$, the time to complete both tasks, must be some CDF enclosed by the left and right envelopes.

Left and right envelopes are each derived from a joint distribution table such as that shown in Figure 3. The probability mass shown associated with each interior cell of a joint distribution table is the product of the probability masses in its corresponding marginal cells if the operands are independent, but relaxing the assumption of independence leaves them undetermined. Therefore when the dependency relationship between the operands is unknown, the process illustrated in Figure 3 requires significant modification (Berleant and Goodman-Strauss 1998). Regardless of the dependency relationship between the marginals, the masses of the interior cells are constrained to some extent by the marginals, which require the masses of all the interior cells in a row to sum to the mass of the marginal cell at the right of that row, and the masses of the interior cells in a column to sum to the mass of the corresponding marginal cell at the bottom of that column. Consequently the summed mass of any particular subset of interior cells will typically have a range of possible values, and for a properly chosen subset the maximum or minimum of this range yields a point on the left or right envelope. More specifically, obtaining the height of the left envelope at time $t$ requires maximizing the collective probability mass of the interior cells whose intervals have low bounds below (or equal to) $t$ subject to the row and column constraints, because the mass of each of those cells either may (if the interval's high bound is above $t$) or must (if the interval's high bound is not above $t$ ) be in the cumulation at $t$. The process is analogous for finding the height of the right envelope: minimize the sum of the probability masses of the interior cells whose intervals have high bounds below or equal to $t$

(Berleant and Goodman-Strauss 1998). Figure 4 explains the process, which can be done by hand for a very small table although in the general case linear programming (LP) is more practical. The left and right envelopes have staircase-like forms. In Figure 4, for example, the heights of the left and right envelopes at *t=3.5* hold for all other values of *t* between 3 and 4. Because for staircase-shaped curves the heights for only a limited number of values of *t* need to be found to fully characterize the envelopes, the number of LP problems is correspondingly limited. Figure 4 also shows the full envelopes.

$t = max(t_x, t_y)$, **$t_x$ and $t_y$ independent**

| | | |
|---|---|---|
| $t \in [2,3]$ $p = 1/8$ | $t \in [2,4]$ $p = 1/8$ | $t_y \in [2,3]$ $p = ¼$ |
| $t \in [3,4]$ $p = 1/4$ | $t \in [3,4]$ $p = 1/4$ | $t_y \in [3,4]$ $p = ½$ |
| $t \in [4,5]$ $p = 1/8$ | $t \in [4,5]$ $p = 1/8$ | $t_y \in [4,5]$ $p = ¼$ |
| $t_x \in [1,2]$ $p = ½$ | $t_x \in [2,4]$ $p = ½$ | ← $t_x$ ↑ $t_y$ |

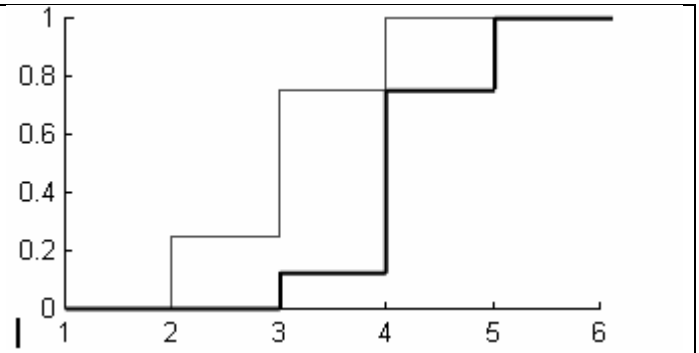



**Figure 3. Random variable $t_x$ is coarsely discretized (bottom row), and similarly for $t_y$ (right hand column). The binary operation appropriate to the task completion problem is $max(t_x,t_y)$ because, for any samples of $t_x$ and $t_y$, both tasks are complete when the one that finishes last is complete. The distribution of joint completion times is implicit in the set of interior cells (*unshaded*) of the joint distribution table, each of which is calculated from its corresponding marginal cells. For example, the upper left cell contains probability mass 1/8, which is the product of the probabilities of its marginal cells in the right hand column and bottom row, 1/4 and 1/2 respectively. The product is used because $t_x$ and $t_y$ are assumed independent (this assumption will be relaxed later). The upper left cell contains the interval [2,3] because its marginal cells have task X complete in time [1,2] and Y in time [2,3], so the time to complete both could potentially be anywhere within that interval. The cumulation over $t$ of the interior cells is bounded by the left and right envelopes shown, with the separation between the envelopes due to the undetermined distribution of each cell's mass across its interval which could, in extreme cases, be concentrated at the interval low or high bound (Berleant 1993).**

When linear programming is applied to minimization and maximization problems of this type the objective function is the sum of the probabilities of the subset of interior cells to be maximized or minimized, and the constraint set consists of one for each row and one for each column. A general-purpose linear programming algorithm such as the simplex method can be used, but a faster choice is the transportation simplex method, which applies to certain problems such as these containing only row and column constraints.

To apply the transportation simplex method to optimize the distribution of probability masses across interior cells, the cost coefficients of the cells in the subset whose probability mass is to be maximized or minimized are set to one, the cost coefficients of the remaining cells are set to zero, and the allocations of the cells are their probability masses. In our software implementation,

problems involving generating envelopes from a 16x16 joint distribution table require approximately 14 seconds using the simplex algorithm but only 1 second using the transportation simplex algorithm, on a Pentium III PC running Windows NT.

**Figure 4. An example.**

**Each interior cell interval in the following joint distribution table has bounds defined by $max(t_x,t_y)$ for its associated (shaded) marginal cell intervals. While interior cell probabilities are constrained by the marginal cell probabilities, they are not fully determined because no assumptions are made about the dependency relationship between $t_x$ and $t_y$.**

| | | |
|---|---|---|
| $t \in [2,3]$<br>$p_{11}$ | $t \in [2,4]$<br>$p_{12}$ | $t_y \in [2,3]$<br>P = ¼ |
| $t \in [3,4]$<br>$p_{21}$ | $t \in [3,4]$<br>$p_{22}$ | $t_y \in [3,4]$<br>P = ½ |
| $t \in [4,5]$<br>$p_{31}$ | $t \in [4,5]$<br>$p_{32}$ | $t_y \in [4,5]$<br>P = ¼ |
| $t_x \in [1,2]$<br>p = ½ | $t_x \in [2,4]$<br>p = ½ | ← ↑<br>$t_x$ $t_y$ |

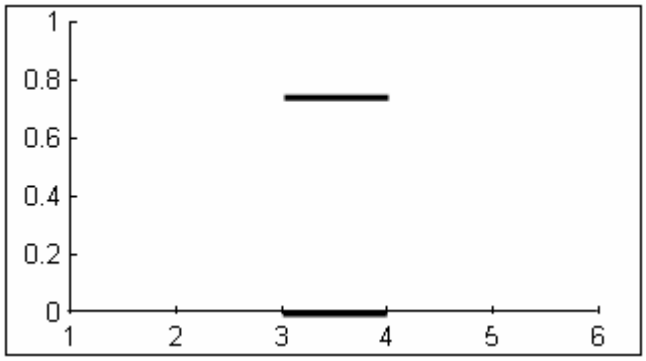
**Consider for example the cumulative probability of $t$ at 3.5. Bolded probabilities masses $p_{11}$, $p_{12}$, $p_{21}$, and $p_{22}$ can contribute to the left envelope of $t$ at 3.5, because the low bounds of the intervals in those cells are ≤3.5. Therefore those probabilities could all be in the cumulation at $t=3.5$, and in the extreme case that $p_{12}$, $p_{21}$, & $p_{22}$ happen to be concentrated at the low bounds of their intervals, will be (and to find points on the envelopes, we are interested in extreme cases). To maximize this cumulation of $p_{11}$, $p_{12}$, $p_{21}$, and $p_{22}$, their sum must be maximized (at the expense of non-bolded probabilities $p_{31}$ and $p_{32}$), yielding $p_{11}+p_{12},+p_{21}+p_{22}=3/4$ as shown in the following solution:**

| | | |
|---|---|---|
| $t \in [2,3]$<br>$p_{11}=¼$ | $t \in [2,4]$<br>$p_{12}=0$ | $t_y \in [2,3]$<br>P = ¼ |
| $t \in [3,4]$<br>$p_{21}=0$ | $t \in [3,4]$<br>$p_{22}=½$ | $t_y \in [3,4]$<br>P = ½ |
| $t \in [4,5]$<br>$p_{31}=¼$ | $t \in [4,5]$<br>$p_{32}=0$ | $t_y \in [4,5]$<br>P = ¼ |
| $t_x \in [1,2]$<br>p = ½ | $t_x \in [2,4]$<br>p = ½ | ← ↑<br>$t_x$ $t_y$ |

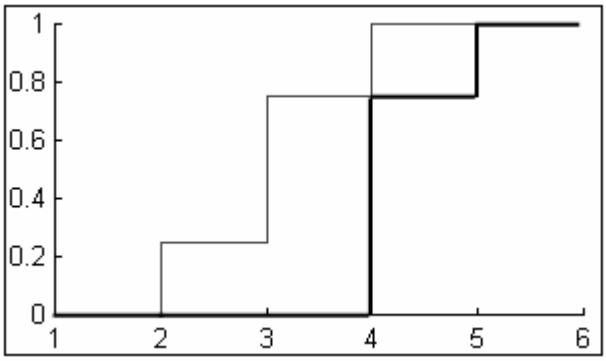
**For the other envelope, the (unary "sum" of) italicized probability mass $p_{11}$ is minimized, yielding 0 as shown in the following solution:**

| | | |
|---|---|---|
| $t \in [2,3]$<br>$p_{11}=0$ | $t \in [2,4]$<br>$p_{12}=¼$ | $t_y \in [2,3]$<br>P = ¼ |
| $t \in [3,4]$<br>$p_{21}=½$ | $t \in [3,4]$<br>$p_{22}=0$ | $t_y \in [3,4]$<br>P = ½ |
| $t \in [4,5]$<br>$p_{31}=0$ | $t \in [4,5]$<br>$p_{32}=¼$ | $t_y \in [4,5]$<br>P = ¼ |
| $t_x \in [1,2]$<br>p = ½ | $t_x \in [2,4]$<br>p = ½ | ← ↑<br>$t_x$ $t_y$ |

**These maximum and minimum cumulations of 3/4 and 0 hold not only for *t*=3.5 but also for all other *t* from 3 to 4, because no interior cell has an interval with an endpoint in that range, as graphed next.**




**Repeating this process for appropriate values of *t* yields the following full envelopes around $t=max(t_x,t_y)$.**




**Although the marginals used here are the same as in Figure 3, the envelopes are farther apart because the dependency between the random variables is unspecified, so the inferential power of the independence assumption is absent. The discretization, coarse in this example, also affects the degree of separation of the envelopes. Finer discretization would yield smaller steps in the envelopes and hence envelopes that are, on average, closer together.**

**Figure 4 (*end*).**

## 4. Generalizing the solution to networks of concurrent and sequential tasks

Extending the approach from two concurrent tasks to larger networks of tasks requires solving three problems: (1) determining the completion time of two tasks that run not concurrently but sequentially, (2) determining the completion time of three or more concurrent tasks, and (3) using results as inputs to obtain further downstream results. These problems may be solved as follows.

(1) To determine the completion time of two sequential tasks, their individual completion times are added, because one completes and then the next begins. To add them, the same procedure that was described earlier for concurrent tasks is applied except that the intervals in the interior cells of the joint distribution table are obtained by performing $t_x+t_y$ instead of $max(t_x,t_y)$. Thus for each joint distribution tables in Figure 4, the top left cell would contain the interval [3,5]=[1,2]+[2,3] instead of [2,3]=*max*([1,2],[2,3]).

(2) To handle three concurrent tasks, the result for two of them is calculated, and that result used as the completion time for a composite task that proceeds concurrently with the third task. In other words, for concurrent tasks X, Y, & Z, we wish to calculate $max(max(x_t,y_t), z_t)$. This is a case of using intermediate results as inputs, discussed next.

(3) To use a result as an input to another calculation, we must convert a pair of left and right envelopes, which is what a result looks like, into a set of intervals and associated probability masses, which is what a marginal in a joint distribution table looks like. To convert, first note that the envelopes consist of horizontal and vertical line segments. This allows the space they enclose to be partitioned into a stack of rectangles (Figure 5, *top*). Each rectangle defines an interval whose low bound is a value on the horizontal axis at which there is a vertical segment of the left envelope (forming the left side of the rectangle), and whose high bound is a value on the horizontal axis at which there is a vertical segment of the right envelope (forming the right side of the rectangle). The mass of the interval is the increment in the cumulative probability represented by the (bottom-to-top) height of the rectangle. The result of this partition process is a set of intervals and their associated probabilities, usable as a marginal in a joint distribution table for another arithmetic operation (Figure 5, *bottom*).

## 5. Using inferences from result envelopes

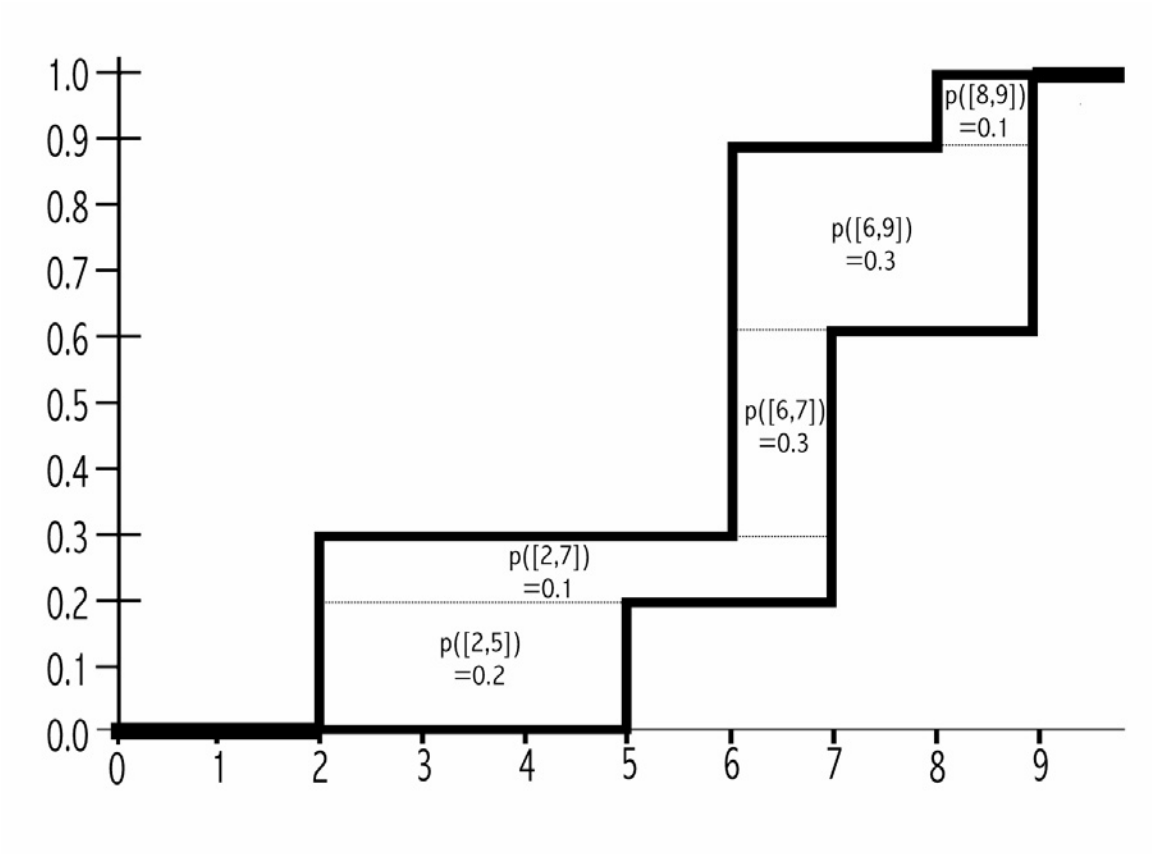
Consider three types of inference that may be drawn from a pair of left and right envelopes.

1) The probability of finishing all the tasks by some time $T_0$ is at least $P_0$ in Figure 6. Similarly, the probability of not finishing by time $T_0$ is at least $(1-P_1)$.

2) Suppose that $p(\text{some outcome}) \in [P,1]$. For example in Figure 6, $p(\text{task completion by time } T_1) \in [P_2,1]$. The interval $[P_2,1]$ is qualitatively different from a point estimate somewhere between $P_2$ and 1 that would derive from an analysis that produced a single distribution function instead of left and right envelopes. This is because, unlike a point estimate, $p \in [P_2,1]$ indicates the plausibility of two distinct scenarios with different implications, (1) certain completion (within the model limits), and (2) uncertain completion. Decisions about resource allocation on the overall project or about deadlines to contract for could depend on which scenario is correct, yet the implied opportunity to seek further information to enable discriminating, or at least to reduce the second order uncertainty in completion time would not be available from an analysis that produced a point probability estimate.

3) Consider the problem of determining the probability that one task will finish later than another, $p(t_y>t_x)$. The probability of one task or path taking longer than another is relevant in such applications as project management where task networks represent PERT

diagrams describing the prerequisite structure of tasks in a project. A simple example is two tasks that begin at the same time, as in Figure 1. A generalization is two tasks embedded in a network of tasks, such as in Figure 7 for final tasks CF and EF. In the generalization the tasks need not start at the same time, and the times at which they complete depend on both the task itself and any prerequisite tasks in the network. These prerequisite tasks may form a simple sequence as in the case of task EF with prerequisite partial path SDE, or contain concurrency as in the case of task CF with prerequisite, concurrent, partial paths SAC and SBC.




$t=max(t_z,t_w)$

| | | | | | |
|---|---|---|---|---|---|
| $t \in [2,5]$ $p=$ | $t \in [2,7]$ $p=$ | $t \in [6,7]$ $p=$ | $t \in [6,9]$ $p=$ | $t \in [8,9]$ $p=$ | $t_w \in [2,3]$ $p = 0.25$ |
| $t \in [3,5]$ $p=$ | $t \in [3,7]$ $p=$ | $t \in [6,7]$ $p=$ | $t \in [6,9]$ $p=$ | $t \in [8,9]$ $p=$ | $t_w \in [3,4]$ $p = 0.5$ |
| $t \in [4,5]$ $p=$ | $t \in [4,7]$ $p=$ | $t \in [6,7]$ $p=$ | $t \in [6,9]$ $p=$ | $t \in [8,9]$ $p=$ | $t_w \in [4,5]$ $p = 0.25$ |
| $t_z \in [2,5]$ $p = 0.2$ | $t_z \in [2,7]$ $p = 0.1$ | $t_z \in [6,7]$ $p = 0.3$ | $t_z \in [6,9]$ $p = 0.3$ | $t_z \in [8,9]$ $p = 0.1$ | $\leftarrow$ $\uparrow$ $t_z$ $t_w$ |

**Figure 5. Staircase shaped envelopes partitioned into a set of intervals and masses (*top*). These might represent a random variable $t_z=max(t_x,t_y)$, used as a marginal in the last row of a joint distribution table (*bottom*), and combined with the concurrent completion time $t_w$ of some other task W. The interior cell probabilities of the table are undetermined since no dependency relationship was defined between the marginals, and so cannot be given values.**

Solving this type of problem requires determining $p(t_y>t_x)$, where $t_x$ and $t_y$ are sample values of random variables for the time points at which two tasks X and Y, or CF and EF, etc., complete. To do this, and relate it to standard techniques, we first provide a continuous solution for the case of independent distributions, then give the discrete form of the solution, then an intervalized discrete form, and finally remove the independence assumption.

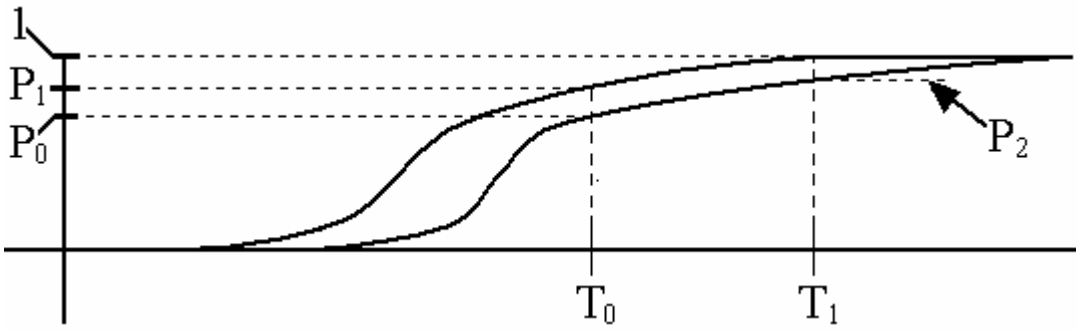



**Figure 6. Left and right envelopes associate probability intervals with time points. If the envelopes describe cumulative probability of task completion over time, then the probability of completion by time $T_0$ is within the interval $[P_0,P_1]$, and by time $T_1$, $[P_2,1]$.**

In the case of a continuous solution for independent distributions, if the density functions of the task completion times are $f_x(t)$ and $f_y(t)$ and sample completion times are $t_x$ and $t_y$, then

$$(1) \qquad p(t_y > t_x) = \int_{t=-\infty}^{\infty} \left( f_y(t)dt \int_{-\infty<t_0<t} f_x(t_0)dt \right).$$

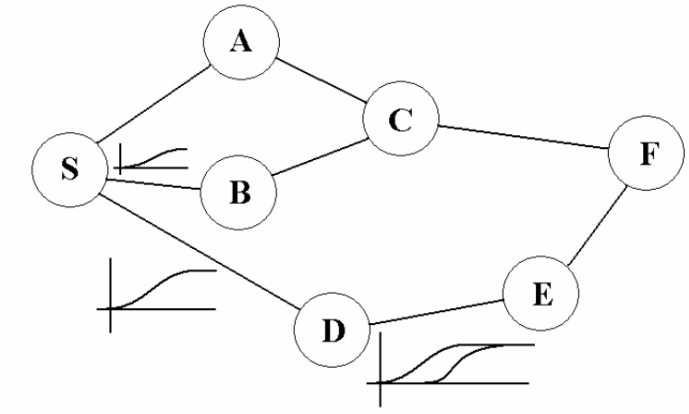



**Figure 7. A network of tasks. The times to complete tasks SB and SD are shown as cumulative distributions. The time to reach state E is the sum of the times to**

**complete SD and DE, and if the dependency relationship between the completion times for SD and DE are unknown the sum is a pair of envelopes rather than a single cumulative distribution.**

Intuitively, $\int_{-\infty<t_0<t} f_x(t_0)dt$ is the area under $f_x$ over all times earlier than some given time $t$, which is $p(t>t_x)$, or the probability that $t$ is later than the completion time $t_x$ of task X. The probability that the completion time of task Y is within a time period centered at $t$ with width $dt$ is $p(\ t_y \in t \pm \frac{1}{2}dt\ ) = f_y(t)dt$. The probability of both $(t>t_x)$ and $(t_y \in t \pm \frac{1}{2}dt)$ is therefore the product of their individual probabilities, $f_y(t)dt \int_{-\infty<t_0<t} f_x(t_0)dt$, and integrating this expression over all possibilities for $t$ gives equation (1).

Discretizing (1) gives $p(t_y > t_x) = \sum_{t=-\infty}^{\infty}\left( f_y(t)\Delta t \sum_{-\infty<t_0<t} f_x(t_0)\Delta t \right)$ for values of $t$ and $t_0$ spaced $\Delta t$ apart. This can be intervalized, bounding the discretization error and giving

(2) $$p(t_y > t_x) = \left[ \sum_{T_y}\left( p(T_y) \sum_{T_x,\ \underline{T_y}>\overline{T_x}} p(T_x) \right),\ \sum_{T_y}\left( p(T_y) \sum_{T_x,\ \overline{T_y}>\underline{T_x}} p(T_x) \right) \right]$$

where the $T_x$ and $T_y$ are intervals over $t_x$ and $t_y$ such as might appear in the marginals of a joint distribution table, $p(T_x)$ and $p(T_y)$ are their associated probability masses, and $\underline{T_x}, \underline{T_y}, \overline{T_x}$ and $\overline{T_y}$ are their low and high bounds.

As an example of equation (2) consider the joint distribution table in Figure 8. The low bound of $p(t_y>t_x)$ is the sum of the probability masses of cells labeled True, which is 0.789. The high bound is the sum of the masses of cells labeled True or Uncertain, which is 0.939, yielding $p(t_y > t_x) \in [0.789,0.939]$.

To remove the independence assumption, the masses of the interior cells are reapportioned among the interior cells within the limits imposed by the row and column constraints using linear programming to minimize the summed masses of the cells labeled True, giving a low bound of 0.61, and then reapportioned again to maximize the summed masses of the cells labeled True or Uncertain, giving a high bound of 1. The new result, $p(t_y > t_x) \in [0.61,1]$, as expected is wider than the earlier result of $p(t_y > t_x) \in [0.789,0.939]$, which benefited from assuming independence.

$t_y > t_x$, $t_x$ **and** $t_y$ **independent**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| True p=.005 | True p=.006 | True p=.008 | True p=.01 | True p=.021 | Uncertain p=.021 | Uncertain p=.01 | False p=.008 | False p=.006 | False p=.005 | **[10.1,11.1] p=.1** |
| True p=.01 | True p=.012 | True p=.016 | True p=.02 | True p=.042 | True p=.042 | Uncertain p=.02 | Uncertain p=.016 | False p=.012 | False p=.01 | **[11.1,12.1] p=.2** |
| True p=.02 | True p=.024 | True p=.032 | True p=.04 | True p=.084 | True p=.084 | True p=.04 | Uncertain p=.032 | Uncertain p=.024 | False p=.02 | **[12.1,13.1] p=.4** |
| True p=.01 | True p=.012 | True p=.016 | True p=.02 | True p=.042 | True p=.042 | True p=.02 | True p=.016 | Uncertain p=.012 | Uncertain p=.01 | **[13.1,14.1] p=.2** |
| True p=.005 | True p=.006 | True p=.008 | True p=.01 | True p=.021 | True p=.021 | True p=.01 | True p=.008 | True p=.006 | Uncertain p=.005 | **[14.1,15.1] p=.1** |
| **[5,6] p=.05** | **[6,7] p=.06** | **[7,8] p=.08** | **[8,9] p=.1** | **[9,10] p=.21** | **[10,11] p=.21** | **[11,12] p=.1** | **[12,13] p=.08** | **[13,14] p=.06** | **[14,15] p=.05** | ← $t_x$ ↑ $t_y$ |

**Figure 8. Joint distribution table representing $t_y > t_x$, for independent $t_x$ and $t_y$. Each interior cell is labeled** True **if $t_y > t_x$ for $t_y$ and $t_x$ in the intervals of the marginal cells of that interior cell,**, False **if instead $t_y < t_x$, and** Uncertain **if the marginal cell intervals overlap (indicating that the unspecified details of the distributions of the marginal cell masses over their intervals determine whether $t_y > t_x$ for all, some, or none of the interior cell mass).**

To restate an example, this process could be used to bound the probability that the completion time of task X will be later than that of task Y in a PERT diagram conforming to Figure 1. The process could also be used in a more complex example such as bounding the probability that task CF will complete later than task EF in Figure 7. The completion time of each of these tasks will be in the form of envelopes, which when converted to marginals will have overlapping intervals as in Figure 5. However any overlap is irrelevant to equation (2), which justifies Figure 8. Ultimately such results could support management decisions about resource allocation intended to optimize the overall completion time of the entire project.

## 6. Software

Crystal Ball (www.decisioneering.com) and @risk (www.palisade.com) are well-known commercial products that rely on Monte Carlo simulation, thereby inheriting the shortcomings of Monte Carlo simulation noted earlier in Section 2. RiskCalc (Ferson 2002) is a commercially available package that can do the operations on random variables used here, although its algorithm (Williamson and Downs, 1990) is different and more complicated than the one used here, some further details of which have been described by Berleant and Zhang (2004(a)). Our software, Statool, is downloadable from http://www.public.iastate.edu/~berleant/statool.html.

## 7. Conclusion

We have shown how to solve a simply stated problem with significant implications: determining completion times of networks of tasks in the absence of assumptions about both the forms of distribution functions and their independence or other dependency relationships. Results are left and right envelopes bounding the space of plausible CDFs. Completion times of individual tasks may be expressed as numbers, intervals, distribution functions, or left and right envelopes.

Real problems frequently pose a variety of uncertainties. Therefore methods for obtaining results with minimal assumptions and while accounting for uncertainty remain an important area of investigation.

## Acknowledgements

The authors are grateful to Helen Regan, National Center for Ecological Analysis and Synthesis, University of California, Santa Barbara, for numerous valuable comments.

## References

1. Agrawal, M.K. and Elmaghraby, S.E.: On Computing the Distribution Function of the Sum of Independent Random Variables, *Computers & Operations Research* **28** (5) (April 2001), pp. 473-483.
2. Ahuja, H.N. and Nandkumar, V.: Simulation Model to Forecast Project Completion Time, *Journal of Construction Engineering and Management* **111** (4) (1985), pp. 325-342.
3. Berleant, D.: Automatically Verified Reasoning with Both Intervals and Probability Density Functions, *Interval Computations* (1993 No. 2), pp. 48-70.
4. Berleant, D. and Goodman-Strauss, C.: Bounding the Results of Arithmetic Operations on Random Variables of Unknown Dependency Using Intervals, *Reliable Computing* **4** (2) (1998), pp. 147-165.
5. (a) Berleant, D. and Zhang, J.: Using Pearson Correlation to Improve Envelopes Around the Distributions of Functions, *Reliable Computing* **10** (2) (2004), pp. 139-161.
6. (b) Berleant, D. and Zhang, J.: Representation and Problem Solving with Distribution Envelope Determination (DEnv), *Reliability Engineering and System Safety* **85** (1-3) (2004), pp. 153-168.
7. Colombo, A.G. and Jaarsma, R.J.: A Powerful Numerical Method to Combine Random Variables, *IEEE Trans. On Reliability* **R-29** (2) (1980), pp. 126-129.
8. Cooper, J.A., Ferson, S., and Ginzburg, L.R.: Hybrid Processing of Stochastic and Subjective Uncertainty Data, *Risk Analysis* **16** (6) (1996) pp. 785-791.
9. Diaz, C.F. and Hadipriono, F.C.: Nondeterministic Networking Methods, *Journal Of Construction Engineering and Management* **119** (1) (March 1993), pp. 40-57.
10. Ditlevsen, O.: Narrow Reliability Bounds for Structural Systems, *Journal Of Structural Mechanics* **7** (4) (1979), pp. 453-472.
11. Ferson, S.: What Monte Carlo Methods Cannot Do, *Journal Of Human and Ecological Risk Assessment* **2** (4) (1996), pp. 990-1007.
12. Ferson, S.: RAMAS Risk Calc 4.0 Software: Risk Assessment with Uncertain Numbers, Lewis Publishers, 2002.
13. Ferson, S., et al.: Myths About Correlations and Dependencies and their Implications for Risk Analysis, Submitted 2004. Contact scott@ramas.com or berleant@iastate.edu.
14. Ingram, G.E., Welker, E.L., and Herrmann, C.R.: Designing for Reliability Based on Probabilistic Modeling Using Remote Access Computer Systems, in *Proc. 7th Reliability and Maintainability Conference*, 1968, American Society Of Mechanical Engineers, pp. 492-500.

15. Kamburowski, J.: An Upper Bound on the Expected Completion Time of PERT Networks, *European Journal of Operational Research* **21** (1985), pp. 206-212.
16. Kleindorfer, G.B.: Bounding Distributions for a Stochastic Acyclic Network, *Operations Research* **19** (1971), pp. 1586-1601.
17. Levitt, R.E. and Kunz, J.C.: Using Knowledge of Construction and Project Management for Automated Schedule Updating, *Project Management Journal* **16** (5), pp. 57-76.
18. Levy, H.: 1992. Stochastic Dominance and Expected Utility: Survey and Analysis, *Management Science* **38** (4), pp. 555-593.
19. Martin, J.J.: Distribution of Time Through a Directed, Acyclic Network, *Operations Research* **13** (1965), pp. 46-66.
20. Mehrotra, K., Chai, J., and Pillutla, S.: A Study of Approximating the Moments of the Job Completion Time in PERT Networks, *Journal of Operations Management* **14** (1996), pp. 277-289.
21. Padilla, E.M. and Carr, R.I.: Resource Strategies for Dynamic Project Management, *Journal of Construction Engineering and Management* **117** (2) (1991), pp. 279-293.
22. Robillard, P. and Trahan, M.: The Completion Time of PERT Networks, *Operations Research* **25** (1) (Jan.-Feb. 1977), pp. 15-29.
23. Schmidt, C.W. and Grossman, I.E.: The Exact Overall Time Distribution of a Project With Uncertain Task Durations, *European Journal of Operational Research* **126** (3) (Nov. 2000), pp. 614-636.
24. Springer, M.D.: *The Algebra of Random Variables,* John Wiley & Sons, Inc. New York, 1979.
25. Van Dorp, J.R. and Duffy, M.R.: Statistical Dependence in Risk Analysis for Project Networks Using Monte Carlo Methods, *International Journal Of Production Economics* **58** (1999), pp. 17-29.
26. Wang. W.-C. and Demsetz, L.A.: Model for Evaluating Networks Under Correlated Uncertainty – NETCOR, *Journal Of Construction Engineering and Management* **126** (6) (Nov./Dec. 2000), pp. 458-466.
27. Williamson, R. and Downs, T.: Probabilistic Arithmetic I: Numerical Methods for Calculating Convolutions and Dependency Bounds, *International Journal of Approximate Reasoning* **4** (1990), pp. 89-158.
28. Woolery, J.C. and Crandall, K.C.: Stochastic Network Model for Planning Scheduling, *Journal of Construction Engineering and Management* **109** (3) (1983), pp. 342-354.

# Methods For Interval Linear Equations

Eldon Hansen

**Abstract.** We discuss one known and five new interrelated methods for bounding the hull of the solution set of a system of interval linear equations. Each method involves a polynomial amount of computing; but requires considerably more effort than Gaussian elimination. However, each method can yield sharper results for appropriate problems. For certain problems, our methods can obtain sharp bounds for one or more (and perhaps all) components of the hull of the solution set.

## 1. Introduction

Consider a set of linear equations

$$Ax = b \tag{1.1}$$

where $A = [\underline{A}, \overline{A}]$ is an interval matrix of order $n$ and $b = [\underline{b}, \overline{b}]$ is an interval vector of $n$ components. The problem of determining the interval hull $h$ of the solution set is NP-hard. (See [5].)

In practice, one can accept non-sharp bounds on $h$ obtained by applying an interval version of Gaussian elimination. Only a polynomial amount of computing of order $O(n^3)$ is needed. Unfortunately, bounds obtained in this way can be far from sharp because of growth of interval widths caused by dependence. (See [2] or [4].). In fact, the method can fail even when $A$ is regular. See [6].

The author introduced preconditioning (see [1]) to reduce the effect of dependence in Gaussian elimination. In this procedure, the equation $Ax = b$ is multiplied by an approximate inverse $B$ of the center of $A$. Unfortunately, preconditioning generally enlarges the solution set. Thus, the hull $h^P$ of the preconditioned system $BAx = Bb$ contains $h$; but is generally larger than $h$.

Later, the author found that the hull of the preconditioned system could be determined exactly (except for roundoff). In [4], the procedure to do so is called the "hull method". The hull method requires somewhat more computing then applying Gaussian elimination to the preconditioned system. Nevertheless, it is the recommended method. The hull method fails only if the preconditioned matrix $BA$ is irregular.

Another way to get bounds on $h$ is to use a version of Gaussian elimination which involves use of what the author called "parameter dependent monotonicity". A procedure of this kind has been described by the author [3]. It requires more computing that ordinary Gaussian elimination; but yields sharper results. It does not involve preconditioning which enlarges the solution set.

In this paper, when we refer to "crude bounds" on the hull $h$, we mean bounds obtained by a method such as described above. The term is meant to imply that the bounds are

obtained by a relatively efficient method; and the bounds include the exact result, but are not sharp.

In subsequent sections, we introduce six interrelated methods for bounding $h$. They require a polynomial amount of computing; but considerably more computing than is needed to get crude bounds. Each requires solving linear programming problems. A weakness of our methods is that they are not applicable if the crude methods fail to produce bounds.

For a small fraction of possible linear systems, the methods in Sections 3 and 4 provide the exact (except for roundoff) hull of the solution set. For a slightly larger fraction of systems, the method of Section 4 provides exact bounds on one or more components of $h$.

We define and discuss sign-definiteness in the next section. All our methods use this property. In Section 5, we describe ways to precondition a system of equations to produce a desired kind of sign-definiteness. The methods are described in Sections 3, 4, and 6 through 9. In Section 10, we note that our methods can be used to bound the inverse of an interval matrix. Section 11 provides a suggested procedure for deciding how to use our methods. Section 12 discusses some special problems for which our methods are especially suited.

## 2. Sign-definiteness

We shall need a concept defined as follows:

$$A\widetilde{x} = [\underline{A}\widetilde{x}, \overline{A}\widetilde{x}] \tag{2.1}$$

That is, we are able to express $A\widetilde{x}$ in terms of endpoints of elements of $A$ because we know the signs of the components of $x$. Equation (1.1) can therefore be written $[\underline{A}\widetilde{x}, \overline{A}\widetilde{x}] = [\underline{b}, \overline{b}]$. Since $\widetilde{x}$ must be such that these intervals intersect. it follows that

$$\underline{A}\widetilde{x} \leq \overline{b} \text{ and } \overline{A}\widetilde{x} \geq \underline{b}. \tag{2.2}$$

To find a component $\underline{h}_i$ (and $\overline{h}_i$) for a given $i = 1, ..., n$, we can minimize (and maximize) $\widetilde{x}_i$ subject to the constraints (2.2). This linear programming problem can be solved by a polynomial amount of computing.

We were able to formulate this problem because, in (2.1), we could express $A\widetilde{x}$ as $[\underline{A}\widetilde{x}, \overline{A}\widetilde{x}]$. In the general case in which $x$ is SD and $\widetilde{x} \in x$, we have

$$a_{ij}\widetilde{x}_j = \begin{cases} [\underline{a}_{ij}\widetilde{x}_j, \overline{a}_{ij}\widetilde{x}_j] \text{ if } x_j \geq 0, \\ [\overline{a}_{ij}\widetilde{x}_j, \underline{a}_{ij}\widetilde{x}_j] \text{ if } x_j < 0. \end{cases} . \tag{2.3}$$

Therefore,

$$A\widetilde{x} = [\widetilde{A}_1\widetilde{x}, \widetilde{A}_2\widetilde{x}] \tag{2.4}$$

where the real (i.e., non-interval) matrices $\widetilde{A}_1$ and $\widetilde{A}_2$ are formed from appropriate endpoints of elements of the interval matrix $A$.

It is this feature which enables us to formulate three of the methods in this paper. A similar feature enables us to formulate the other three methods. See Section 4. In particular,

see Equation (4.1). In all six of our methods, we determine or produce a vector or subvector which is SD and use this fact to obtain an algorithm for bounding $h$. We rely on the crude methods to provide essential information about SD.

Each algorithm involves solving a set of linear programming problems. Thus, they require considerably more computing than the procedures in Section 1 for getting crude bounds.

## 3. First method

Assume we have computed crude bounds $x^B$ on $h$ by a method such as those described in Section 1. If $x^B$ is SD, then $h$ is SD; and we can use the known method of Section 2 to compute $h$ sharply. If $x^B$ lies in just a few orthants, a reasonable procedure is to use the method to obtain the part of $h$ in each of these orthants. The narrowest interval vector containing all such results is $h$.

If $h$ extends into all $2^n$ orthants, then this approach entails solving $2^n$ separate problems each involving $2n$ linear programming problems. This amount of effort is prohibitive even for moderate values of $n$. This is an example of an exponential amount of computing used to solve the NP-hard problem of determining $h$.

## 4. Second method

It is possible to define a linear programming problem in which the primal variables are components of $x$ and the dual variables are elements of the inverse of $A$. Since our second method concentrates on the inverse, the method can be considered as a kind of dual of the first method. In this sense, the third and fourth methods below are duals as are the fifth and sixth, respectively.

Given an interval matrix $A$, let $P$ denote the interval matrix which is the hull of the set of inverses of real matrices in $A$. Then for any real $\widetilde{A} \in A$, there exists $\widetilde{P} \in P$ such that $\widetilde{P}\widetilde{A} = I$. Note that this does not imply that for any $\widetilde{P} \in P$, there exists $\widetilde{A} \in A$ such that $\widetilde{P}\widetilde{A} = I$.

Let $p_i^T$ denote the $i$th row of $P$. If $p_i$ is SD, then we know how to express both $\widetilde{p}_i^T A$ and $\widetilde{p}_i^T b$ for any real $\widetilde{p}_i \in p_i$. For example, if $p_i \geq 0$, then

$$\widetilde{p}_i^T A = [\widetilde{p}_i^T \underline{A}, \widetilde{p}_i^T \overline{A}] \quad \text{and} \quad \widetilde{p}_i^T b = [\widetilde{p}_i^T \underline{b}, \widetilde{p}_i^T \overline{b}]. \tag{4.1}$$

Note that

$$x_i = \{\widetilde{p}_i^T \widetilde{b} : \widetilde{p}_i^T \widetilde{A} = e_i^T,\ \widetilde{A} \in A,\ \widetilde{b} \in b\}$$

where $e_i$ denotes the $i$th column of the identity matrix. Therefore, $\underline{h}_i$ is the solution of the linear programming problem

$$\min\ p_i^T \underline{b} \tag{4.2}$$

subject to $p_i^T \underline{A} \leq e_i^T$ and $p_i^T \overline{A} \geq e_i^T$; and $\overline{h}_i$ is the maximum of $p_i^T \overline{b}$ subject to the same constraints.

More generally, if $p_i$ is just SD rather than nonnegative, we know how to express $p_i^T A$ and $p_i^T b$ in a way similar to (4.1). See (2.3). Therefore, we can compute sharp values of $\underline{h}_i$ and $\overline{h}_i$.

If all the elements of $P$ are SD, we could compute all the components of $h$ sharply by solving $2n$ linear programming problems. However, we now show that the results can sometimes be obtained more simply.

If we differentiate the equation $AP = I$ with respect to an element $a_{rs}$ of $A$, we find that

$$\frac{\partial P_{ij}}{\partial a_{rs}} = -P_{ir}P_{sj} \quad (i, j, r, s = 1, ..., n). \tag{4.3}$$

When $P$ is SD, we know the signs of these derivatives. Therefore, we can determine the real matrix in $A$ whose inverse is $\underline{P}$ and the real matrix in $A$ whose inverse is $\overline{P}$. For example, if $P \geq 0$, then the derivatives given by (4.3) are negative. In this case, $\underline{P} = \overline{A}^{-1}$ and $\overline{P} = \underline{A}^{-1}$.

For certain conditions on $b$, the results of the optimization problem (4,2) (and the corresponding max) can be expressed simply. For example, if $P \geq 0$, we find

$$h = \begin{cases} [\overline{A}^{-1}\underline{b}, \underline{A}^{-1}\overline{b}] & \text{if } b \geq 0 \\ [\underline{A}^{-1}\underline{b}, \underline{A}^{-1}\overline{b}] & \text{if } 0 \in b \\ [\underline{A}^{-1}\underline{b}, \overline{A}^{-1}\overline{b}] & \text{if } b \leq 0 \end{cases} .$$

This special case is known. See page 108 of [6].

For this example, we are able to compute $h$ as the solution of two noninterval systems. It requires that $P$ be positive and that $b$ be rather special. For our method $P$ need only be SD, and $b$ is arbitrary. However, our method can require much more computing.

To formulate the linear programming problem (4.2) we must know the signs of all the components of row $p_i^T$ of $P$. Using a method described in Section 1, we can compute crude bounds on the $i$th row $p_i^T$ by solving $A^T p = e_i$. If the bounds are SD, then the exact row $p_i^T$ is SD. In this case, we can obtain sharp bounds on $x_i$ using the above method.

If desired, we can compute bounds on all the rows of $P$ by solving $A^T P^T = I$ and then compute bounds on all the columns of $P$ by solving $AP = I$. The intersection of the two results will generally be sharper than either result. This enhances the chance of proving that $p_i$ is SD.

Let $P^B$ denote a bound on $P$ obtained by a method such as that just described. Note that we can obtain crude bounds on $h$ by computing $P^B b$; but the bounds need not be sharp even if $P^B$ is sharp. Suppose $p_i$ fails to be SD in only a few of its components. We can divide each of these components into its negative and positive parts and solve for $x_i$ for each combination of cases in which $p_i$ is SD. Then $h$ is the hull of the union of results. Compare the similar statement in Section 3.

Let $(p_i^B)^T$ denote the $i$th row of $P^B$. In problem (4.2), the unknown real row vector $p_i$ must be bounded by the interval row vector $p_i^B$. If the method used to solve the linear programming problem (4.2) can benefit from additional constraints, we can use $p_{ij} \in p_{ij}^B$ $(j = 1, ..., n)$.

## 5. Preconditioning

We now consider methods of preconditioning which allow a preconditioned equation to be solved exactly in part or in whole. One type of method involves preconditioning a vector so that it is SD. See the methods in Sections 6 and 7. Another type eliminates the variables which are not SD so that the problem can be partially solved. The latter type can be regarded as preconditioning so that certain quantities are zeroed. It involves operations by an interval matrix; so it is perhaps misleading to call it preconditioning. See the methods in Sections 8 and 9.

We now consider the first type of preconditioning. Assume an interval vector $x = [\underline{x}, \overline{x}]$ is not SD. Define a real vector $q$ with components

$$q_i = \begin{cases} -\underline{x}_i & \text{if } \underline{x}_i < 0 < \overline{x}_i \text{ and } |\underline{x}_i| \le \overline{x}_i \\ 0 & \text{if } \underline{x}_i \ge 0 \text{ or } \overline{x}_i \le 0 \\ -\overline{x}_i & \text{if } \underline{x}_i < 0 < \overline{x}_i \text{ and } |\underline{x}_i| > \overline{x}_i \end{cases} \qquad (i = 1, ..., n).$$

Then the interval vector $y = x + q$ is the nearest SD interval vector to $x$ in some sense.

Assume $x$ is not SD. However, assume that at least one component of $x$ is strictly SD. By "strictly SD", we mean that the the interval is SD and neither endpoint is zero. An interval having an endpoint which is zero is SD, but not strictly SD. Let $j$ be the index such that $x_j$ is the component of largest mignitude. The mignitude of $x_j$ is

$$\text{mig}(x_j) = \begin{cases} \underline{x}_j & \text{if } \underline{x}_j > 0 \\ -\overline{x}_j & \text{if } \overline{x}_j < 0 \\ 0 & \text{otherwise} \end{cases} .$$

Define the vector $v(j)$ with components

$$v_i(j) = \begin{cases} q_i/\underline{x}_j & \text{if } x_j > 0 \\ q_i/\overline{x}_j & \text{if } x_j < 0 \end{cases}$$

and define the matrix

$$V_j = I + v(j)e_j^T. \tag{5.1}$$

Then $V_j x$ is SD.
Note that

$$V_j^{-1} = I - \tfrac{1}{2}v(j)e_j^T \tag{5.2}$$

Therefore, $V_j^{-1}$ is known exactly when $V_j$ has been determined.

In what follows, we do not actually use the interval vector (such as $x$) which we precondition so as to be SD. Instead, we use an unspecified real vector $\widetilde{x} \in x$. However, if we precondition so that the interval vector $x$ is SD, then the sign of any component of $\widetilde{x} \in x$ has the sign we impose on the corresponding component of $x$.

Suppose we precondition a matrix $A$ by multiplying by a real matrix $\widetilde{B}$. The product $M = \widetilde{B}A$ can be irregular even when $A$ is regular and $\widetilde{B}$ is nonsingular. The four methods we describe below all use some kind of preconditioning and in two cases $\widetilde{B}$ becomes an interval matrix. In each method, we assume that $\widetilde{B}$ is such that $M$ is regular.

A virtue of the preconditioning method just described is that the preconditioning matrix differs from the identity in only one row. Contrast this with preconditioning using the inverse of the center of $A$. See Section 1.

The matrix $V_j^{-1}$ in (5.2) will be used as a preconditioner. The smaller the norm of the vector $v(j)$ used to define $V_j^{-1}$, the closer $V_j^{-1}$ is to the identity. The enlargement of the solution set by preconditioning is less when $V_j^{-1}$ is nearer the identity.. If more than one component of $x$ is strictly SD, it is sometimes possible to define a matrix similar to $V_j^{-1}$ which is nearer the identity. We omit the details.

## 6. Third method

Our third method is obtained by introducing preconditioning into our first method. Suppose we have obtained crude bounds $x^B$ on the solution to $Ax = b$ and find that at least one component of $x^B$ is SD. Then the corresponding component of the hull $h$ is SD. Therefore we can determine a matrix $V$ as in Section 5 such that $Vx^B$ is SD. This assures that $Vh$ is SD.

Define $y = Vx$ and $M = AV^{-1}$. Then the solution $y$ of $My = b$ is SD and its hull can be found using the first method (in Section 3). We then obtain $x$ as $x = V^{=1}y$.

Presumably any kind of preconditioning can enlarge the solution set. It is natural to compare the method using this kind of preconditioning with a method in Section 1 used to get the crude bounds $x^B$. The latter method requires considerably less computing.

To get $x^B$, we can precondition by multiplying by an approximate inverse of the center $A_c$ of $A$. The closer $A_c$ is to the identity, the less the preconditioning step tends to enlarge the solution set. The closer $x^B$ is to being SD, the less the method just described enlarges the solution set. (A measure of how far $x^B$ is from SD is the norm of the vector $v(j)$ in Section 5.) The amount to which preconditioning enlarges the solution set depends on how far the preconditioner is from the identity matrix. Therefore, the comparative sharpness of results when preconditioning by $A_c^{-1}$ or by $V^{-1}$ depends strongly on the nature of the problem. A similar statement holds for the methods discussed below.

## 7. Fourth method

In the third method, we introduced a preconditioning procedure which produced an equation whose solution was SD. Therefore, we could apply the first method. In the same way, we can precondition so that the new equation can be solved by the second method (in Section 4). That is, the preconditioning is such that a row of the inverse of the generated matrix is SD.

The exact inverse $P$ of $A$ will usually be regular when $A$ is regular. Assume it is. Also assume that the bound $P^B$ (obtained as in Section 4) on $P$ is regular. Then for any $i = 1, ..., n$, at least one component of the $i$th row $(p_i^B)^T$ of $P^B$ must be SD. From Section 5, we can determine a matrix $V$ such that $(p_i^B)^T V$ is SD. This implies that $p_i^T V$ is SD.

Assume that we have determined $V$ such that row $i$ of $PV$ is SD. To precondition $Ax = b$, we multiply by the matrix $V^{-1}$. (Note that $V^{-1}$ is exactly known from (5.2) when $V$ is known.) The new coefficient matrix is $M = V^{-1}A$. Row $i$ of the inverse of $M$ is SD.

We can compute the $i$th component of the hull of $V^{-1}Ax = V^{-1}b$ using the second method (see Section 4).

It is not necessary to verify that row $i$ of the inverse of $M$ is SD. If it were computed to not be SD, it would still be correct to proceed as if it were.

Note that the result $x_i$ will generally not be a sharp bound on the corresponding component $h_i$ of the hull since preconditioning by $V^{-1}$ tends to enlarge the solution set.

There are two other reasons why a solution obtained by this method can fail to be sharp. First, the computed bounds on the inverse will generally not be sharp. The preconditioning matrix $V$ is determined so that a row of the bound $P^B$ is SD. Since $P^B$ is not a sharp bound on $P$, the matrix $V$ generally causes an "overshoot" when changing a non-SD element to SD. Therefore, preconditioning $A$ by $V^{=1}$ causes too large a change in $A$.

The other cause of loss of sharpness is more subtle. It occurs because $P$ contains matrices which are not inverses of matrices in $A$. The loss of sharpness is similar in nature to that just described.

## 8. Fifth method

Assume that one or more component of the crude bound $x^B$ is SD. Then the corresponding component(s) of the hull $h$ are SD. For simplicity, assume that for some integer $k$, we have $\underline{h}_i < 0 < \overline{h}_i$ for $i = 1, ..., k$ and $h_i$ is SD for $i = k+1, ..., n$. Partition $A$, $x$, and $b$ conformally so that the equation $Ax = b$ takes the form

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix}. \tag{8.1}$$

Here $x^T = (y^T, z^T)$ where $y$ has $k$ components and $z$ has $n - k$ components. The hull of the solution set of this system is such that the interval solution $z$ is SD.

Perform interval Gaussian elimination; but stop when $A_3$ is zeroed The result is an equation of the form

$$\begin{bmatrix} A_1' & A_2' \\ 0 & A_4' \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} c' \\ d' \end{bmatrix} \tag{8.2}$$

This equation can be written as the system

$A_1'y + A_2'z = c'$,

$A_4'z = d'$.

Since $z$ is SD, we can compute $z$ sharply from the equation $A_4'z = d'$ using the first method (in Section 3). We can then compute bounds on $y$ by backsolving $A_1'y + A_2'z = c'$.

When we perform the interval Gaussian elimination to obtain (8.2) from (8.1), interval widths will tend to grow; and we should precondition. Suppose we precondition by multiplying $Ax = b$ by an approximate inverse of the center of $A$. Then there is no point in using the method just described because we can determine the hull of such a preconditioned system sharply using the hull method. See [4].

Instead, we should precondition by an approximate inverse of the center of

$$\begin{bmatrix} A_1 & 0 \\ A_3 & I \end{bmatrix}$$

where $I$ denotes an identity matrix of order $n-k$. This tends to enlarge the solution set by less than preconditioning by an approximate inverse of the center of the entire matrix $A$.

## 9. Sixth method

We now consider a method which can be regarded as a kind of dual of the fifth method.

Suppose we compute crude bounds on the inverse $P$ of $A$ as described in Sections 1 and 4. To simplify discussion we fix our attention on the first row of $P$. We also simplify by assuming that $P_{1j}$ is SD for $j = 1, ..., k$ and that $\underline{P}_{1j} < 0 < \overline{P}_{1j}$ for $j = k+1, ..., n$.

Partition $A$ as

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}$$

where $A_1$ is $k$ by $k$ and $A_4$ is $n-k$ by $n-k$. We can perform Gaussian elimination on $A$ in such a way that $A_2$ becomes zero. This is achieved by multiplying by a matrix of the form

$$B = \begin{bmatrix} I & B_2 \\ 0 & B_4 \end{bmatrix}.$$

This matrix need not be explicitly generated. However, the operations to obtain $BA$ must also be perfomed on $b$ so that the new equation is $BAx = Bb$.

The first row of the inverse $PB^{-1}$ of $BA$ is such that its first $k$ components are the same as those of $P$ and, by assumption, are SD. The last $n-k$ components of the first row of $PB^{-1}$ are zero (and hence SD). Since the first row of the inverse of $BA$ is SD, we can determine the first component of the solution of $BAx = Bb$ by the second method (in Section 4).

Other components of $x$ can be bounded in a similar way.

## 10. Using the inverse

Suppose we have a matrix $P'$ which bounds the exact inverse $P$ of $A$. In Section 4, we discussed how to obtain $P'$. Note that $P'b$ bounds the hull $h$ of the solution set. The bound on $h$ would generally not be sharp even if $P'$ were the exact inverse $P$. This is because $P$ can contain matrices which are not inverses of any matrix in $A$. However, this provides another bound on $h$ which can be intersected with bounds obtained by methods such as those we have described.

We can bound $P$ more sharply than by the way described in Section 4 by using methods such as ours to solve the equations which $P$ must satisfy. Thus, we can solve for the $i$th column of $P$ by solving $Ax = e_i$. We can solve for the $i$th row of $P$ by solving $A^T x = e_i$. If we solve for both the rows and the columns, we can intersect the two results.

The wider the vector $b$ in $Ax = b$ the wider the solution set. From this point of view, the equation $Ax = e_i$ is ideal in that the right hand vector $e_i$ is real and all components

are zero except one. This suggests that there can be advantages in using a method which computes the inverse of $A$.

## 11. Choosing a method

In practice, we must decide whether to use any of the methods we have described; and, if so,which one(s). We first note that if the center $A_c$ of $A$ is "near" the identity matrix, then its inverse $B = A_c^{-1}$ is near the identity. In this case, there is little need to use any of our methods. We can use $B$ to precondition the system without unduly enlarging the solution set. We can then use the hull method to determine the hull of the preconditioned system. The question of what is meant by "near the identity matrix" will be left to a user. Alternatively, the center of $A$ might be near a matrix which is the identity with rows and columns permuted.

Suppose the center $A_c$ of $A$ is near the identity matrix. Then the center of $A^{-1}$ is near the identity. That is, its off-diagonal elements are likely to contain zero. Therefore, a row of $A^{-1}$ is unlikely to be SD. In this case, it is unlikely that our second method is applicable; and there is probably little point in using the fourth or sixth method.

For any problem, a reaonable first step is to obtain bounds $x^B$ on the hull $h$ using the hull method. One can also use the method from [3] and find the intersection of the two methods Thereafter, we might use the following procedural steps. They involve a great deal of computing; but the work is polynomially bounded.

(1) If $A_c$ is near the (perhaps permuted) identity matrix, accept the results of the hull method as a sufficiently sharp solution. Thus, go to step (10).

(2) If $x^B$ is SD in all but a few components, solve for $h$ using the first method (in Section 3). Then go to step (10).

(3) Use the hull method to obtain bounds on $A^{-1}$ by solving $Ax = e_i$ for $i = 1, ..., n$. (The method in [3] can also be used.) Also solve $A^T x = e_i$ for $i = 1, ..., n$ to bound $A^{T-1}$. Then intersect the two bounds on $A^{-1}$. Denote the resulting bound on $A^{-1}$ by $P^B$.

(4) For $i = 1, ..., n$, if all but a few components of row $i$ of $P^B$ are SD, solve for $h_i$ using the second method (in Section 4). If all components of $h$ are obtained in this way, go to step (10).

(5) Compute the bound $P^B b$ on $h$. Intersect it with $x^B$ and the result from step (4).

(6) If at least one component of $h$ has been shown to be SD, the third method (in Section 6) is applicable. Use it to bound $h$. Intersect the solution with the result of step (5).

(7) Use the fourth method (in Section 7) to bound $h_i$ for $i = 1, ..., n$. Skip any value of $i$ for which $h_i$ was obtained sharply in step (4). Intersect the result with the result from step (6).

(8) If at least one component of $h$ has been found to be SD, use the fifth method (in Section 8) to bound $h$. Intersect the result with the result from previous steps.

(9) Use the sixth method (in Section 9) to bound $h$. Intersect the result with the result from previous steps.

(10) Stop.

The amount of work to apply this procedure is not particularly excessive if the crude bounds reveal that $h$ or $A^{-1}$ is SD. If this is not the case, our procedure is useful only if sharpness is so important that a considerable amount of computing is warranted.

There are cases in which there is no need to use our methods. If $A$ is an M-matrix, then interval Gaussian elimination will obtain the hull $h$ sharply. See [5].

## 12. Some special cases

The essential requirement in our methods is that we are able to express certain products in which a factor is unknown except for its sign. For example, in the first method, we needed to be able to express $Ax$ when $x$ is unknown. If $x_j$ is SD, we can use (2.3) to express $a_{ij}x_j$ in terms of the endpoints of $a_{ij}$. But suppose that for a given value of $j$, the element $a_{ij}$ is real (i.e., a degenerate interval) for all $i = 1, ..., n$. Then the "endpoints" of $a_{ij}$ are equal; and $a_{ij}x_j$ is expressed in terms of their coincident value. Therefore, $x_j$ need not be SD.

If all but a few columns of $A$ are real, the first method can be used to determine the hull $h$ with a reasonable amount of computing. If a given column of $A$ is real, the corresponding component of $x$ does not have to be made SD in the third method, nor does it have to be eliminated in the fifth method.

Similar statements can be made for the dual methods. Now, however, we must be able to express both $p^TA$ and $p^Tb$ for a row $p^T$ of $P$. Consider the matrix $R = (P \quad b)$ which is the matrix $P$ augmented with the vector $b$ as an added column. If all but a few rows of $R$ are real, the second method can determine the hull. If a row (or rows) of $R$ is real, the corresponding component of a row of $P$ need not be SD in the fourth and sixth methods.

Even if every element of $A$ except one is real, then every element of $P$ can be a nondegenerate interval. It is unlikely that we shall know that a row of $R$ is real. Therefore, the dual methods generally do not "simplify" in this way.

## 13. A non-polynomial method

The methods we have described fail if the crude methods fail to obtain bounds on the solution. In this section, we describe how the hull of the solution set of $Ax = b$ can be obtained as the solution of an optimization problem. The optimization problem is not a linear programming problem; so the work to solve it is not polynomially bounded. We include it as an alternative for two reasons. First, it does not require that some other method provide crude bounds. Second, it is similar to the methods we have described. The difference is that the formulation of the optimization problem includes nonlinear constraints.

Equation (2.3) can be written as

$$a_{ij}x_j = [\min\{\underline{a}_{ij}x_j, \overline{a}_{ij}x_j\}, \max\{\underline{a}_{ij}x_j, \overline{a}_{ij}x_j\}]. \tag{14.1}$$

Since the maximum of two function can be exressed as their average plus half their difference, we have

$$\max\{\underline{a}_{ij}x_j, \overline{a}_{ij}x_j\} = \tfrac{1}{2}(\underline{a}_{ij}x_j + \overline{a}_{ij}x_j) + \tfrac{1}{2}|\overline{a}_{ij}x_j - \underline{a}_{ij}x_j| = m_{ij}x_j + \tfrac{1}{2}w_{ij}|x_j| \tag{14.2}$$

where $m_{ij} = \frac{1}{2}(\underline{a}_{ij} + \overline{a}_{ij})$ and $w_{ij} = \overline{a}_{ij} - \underline{a}_{ij}$. Similarly,

$$\min\{\underline{a}_{ij}x_j, \overline{a}_{ij}x_j\} = m_{ij}x_j - \tfrac{1}{2}w_{ij}|x_j|. \tag{14.3}$$

Row $i$ of the equation $Ax = b$ can be written

$\sum_{j=1}^{n} a_{ij}x_j = b_i \quad (i = 1, ..., n).$

From (14.1), (14.2), and (14.3), we therefore obtain

$\sum_{j=1}^{n} [m_{ij}x_j - \frac{1}{2}w_{ij}|x_j|,\ m_{ij}x_j + \frac{1}{2}w_{ij}|x_j|] = b_i.$

A point $x$ is in the solution set of $Ax = b$ only if the intervals in the left and right members intersect. This imposes the constraints

$$\sum_{j=1}^{n} (m_{ij}x_j - \tfrac{1}{2}w_{ij}|x_j|) \le \overline{b}_i, \tag{14.4a}$$

and

$$\sum_{j=1}^{n} (m_{ij}x_j + \tfrac{1}{2}w_{ij}|x_j|) \ge \underline{b}_i. \tag{14.4b}$$

for $i = 1, ..., n$.

We can obtain the $k$th component of the hull by minimizing and maximizing $x_k$ subject to the constraints (14.4). The function $|x_j|$ is not differentiable at $x_j = 0$. We can obtain constraints which are differentiable if we replace $|x_j|$ by $x_{j+n}$ and add the constraints

$x_{j+n}^2 - x_j^2 = 0 \quad (j = 1, ..., n).$

The problem now becomes: For $k = 1, ..., n$,

minimize and maximize $x_k$

subject to

$\sum_{j=1}^{n} (m_{ij}x_j - \frac{1}{2}w_{ij}x_{j+n}) \le \overline{b}_i \quad (i = 1, ..., n)$

$\sum_{j=1}^{n} (m_{ij}x_j + \frac{1}{2}w_{ij}x_{j+n}) \ge \underline{b}_i \quad (i = 1, ..., n)$

$x_{j+n}^2 - x_j^2 = 0 \quad (j = 1, ..., n).$

## 14. References

[1] Hansen, E. R., Interval arithmetic in matrix computations, part I, SIAM J. Numer. Anal. 2, 308-320, 1965.

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

[3] Hansen, E. R., Sharpening interval computations, paper presented at the First Scandinavian workshop on interval methods and their application, Copenhagen, 2003.

[4] Hansen, E. R. and Walster, G. W. Global Optimization Using Interval Analysis, (second ed.), Marcel Dekker, 2004.
[5] Heindl, G., Kreinovich, V., and Lakeyev, A. (1998), Solving linear interval systems is NP-hard even if we exclude overflow and underflow, Reliable Computing, 4, 383-388.
[6] Neumaier, A., Interval Methods for Systems of Equations, Cambridge University Press, London, 1990.

# Formulation for Reliable Analysis of Structural Frames

George Corliss (george.corliss@marquette.edu)
*Electrical and Computer Engineering, Marquette University*

Christopher Foley (chris.foley@marquette.edu)
*Civil and Environmental Engineering, Marquette University*

R. Baker Kearfott (rbk@louisiana.edu)
*Mathematics, University of Louisiana at Lafayette*

**Abstract.** Structural engineers use design codes formulated to consider uncertainty for both reinforced concrete and structural steel design. For a simple one-bay structural steel frame, we survey typical uncertainties and compute an interval solution for displacements and forces. The naive solutions have large over-estimations, so we explore the Mullen-Muhanna element-by-element strategy, scaling, and constraint propagation to achieve tight enclosures of the true ranges for displacements and forces in a fraction of the CPU time typically used for simulations. That we compute tight enclosures, even for large parameter uncertainties used in practice, suggests the promise of interval methods for much larger structures.

**Keywords:** structural steel frames, partially constrained connections, uncertain parameters, interval arithmetic, element-by-element, constraint propagation.

## 1. Introduction

Structural engineers have used design codes formulated to consider uncertainty for both reinforced concrete and structural steel design for several decades. The format for these design codes has been termed Load and Resistance Factor Design (LRFD). The LRFD format for structural steel design is founded upon first-order, second-moment reliability theory applied to structural loads and resistances (Cornell, 1969). LRFD-based design rests on the following definition for the probability of structural failure,

$$P_F = P\left((R - Q) < 0\right), \tag{1}$$

where

$R$ = a structure's resistance, which is considered a random variable, modeled using a known probability density function (PDF);
$Q$ = the load effect, which is also a random variable with known PDF.

The frequency distribution of the resulting random variable, $R-Q$, allows the definition of a safety margin against structural failure. The probability of failure expressed in Equation (1) is re-phrased as (Ravindra & Galambos, 1978),

$$P_F = P\left(\ln\left(R/Q\right) < 0\right) \ .$$

If one knew the probability distribution of $\ln\left(R/Q\right)$, determining the probability of failure for the structure would be very easy. Unfortunately, there are several random variables that

contribute to structural resistance as well as load effect. These contributors do not all follow the same PDF's, and the process of characterizing them also is uncertain. In this paper, we use interval arithmetic to compute reliable bounds for structure responses in the presence of uncertain parameters. In section 2, we discuss the nature of the uncertainties and realistic bounds.

The first-order second-moment method approximates the failure of a structure by the safety index (Ravindra & Galambos, 1978),

$$\beta = \frac{\text{mean}\left(\ln(R/Q)\right)}{\sigma\left(\ln(R/Q)\right)} \ ,$$

where $\sigma\left(\ln(R/Q)\right)$ is the standard deviation of the natural logarithm of the ratio of resistance to load. In a simplistic sense, the LRFD formulation seeks to define a probability of a failure using an acceptable number of $\beta$'s away from mean $\left(\ln(R/Q)\right)$. The acceptable value of $\beta$ for various structural components is determined using calibration with existing structural systems. In other words, the LRFD design procedures that were proposed, and are currently in use, provide a level of reliability against structural failure that is near that of structures designed using pre-LRFD criteria.

In the discipline of structural engineering, the engineer is often concerned with determining response quantities for which there is very small probability of exceedance. For example, one may be interested in the lateral displacement at the top of the frame shown in Figure 1 for which the probability of exceedance is 0.1%. Since life-safety is involved in design of structural systems, we may desire a very small coefficient of variation in this probabilistic estimate.

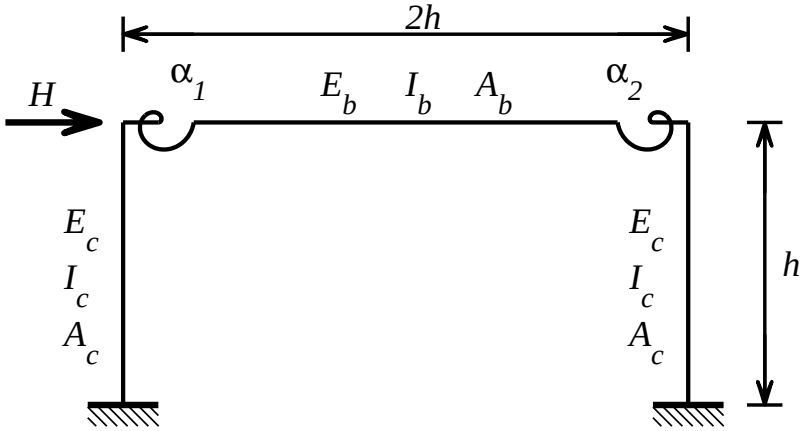



*Figure 1.* Simple one bay portal frame with partially constrained connections.

Monte-Carlo simulation is a traditional approach for establishing safety indices or probabilities of failure for structural systems. Unfortunately, simulation also includes a level of uncertainty in the results. Better results require more simulations. Soong and Grigoriu

(Soong & Grigoriu, 1993) have shown that the coefficient of variation in an estimated probability $\overline{P}$ can be written as

$$V_{\overline{P}} = \sqrt{\frac{1 - P_{\text{true}}}{N \cdot P_{\text{true}}}} \ , \tag{2}$$

where $P_{\text{true}}$ is the true probability, and $N$ is the number of simulations. Equation (2) can be solved for a required number of simulations using a desired probability and coefficient of variation. If our structural analysis determining the unlikely structural response should have a probability of failure of 0.001 and we need to have a small coefficient of variation in that estimate (e.g., 0.05), then by Equation (2), 399,600 simulations are necessary. That is, nearly 400,000 structural analyses are required to be able to determine structural response for an event with very low probability with high confidence. Other simulation techniques are available, e.g., importance sampling (Melchers, 2001). However, in general, simulation can be a highly expensive tool for understanding uncertainty in structural engineering.

Recent work (Mullen & Muhanna, 1999; Muhanna & Mullen, 2001) introduced intervals as a means for reliably accounting for uncertainty in structural engineering. The present study considers load and resistance uncertainty using interval-based structural analysis. The success of the present work foreshadows additional applications of interval methods in structural engineering to quantify uncertainty in progressive collapse, ground motion analysis, and other highly important endeavors. Furthermore, it is hoped that the interval-based results can be used to *quantify* any error present in structural engineering design as a result of first-order, second-moment reliability-based design methods for complex structures.

## 2. Development of Intervals for Load and Resistance

Structural loads and resistances frequently are defined using probability density function models for the frequency of occurrence of properties or loading magnitudes characterizing structural behavior. This section assigns intervals of known confidence for cross-sectional properties, loading, material properties, and connection response.

### 2.1. Lateral Wind Loading

The frequency of occurrence of extreme wind speeds is modeled using Fisher-Tippett Type 1 Extreme Value probability distributions (Simiu et al., 1978). To demonstrate the process, a hypothetical extreme wind record is used to generate wind speed intervals and then a wind pressure interval suitable for structural analysis. The mean peak wind speed (assumed here to be for 3-second gusts) and standard deviation for a 19-year record are

$$\overline{V}_{\text{3-sec}} = 62.7 \text{ mph} \qquad \sigma_{\text{3-sec}} = 8.63 \text{ mph}.$$

The PDF assumed allows one to compute peak wind speeds and confidence levels associated with those speeds that include sampling error due to the limited number of years for

which data is available. Buildings are often assumed to have service life spans of 50 years. If one is willing to accept that the wind speed used for design has a 5% chance of being exceeded in 50 years, one is establishing a 1,000 year mean recurrence interval wind. In other words, there is a 0.1% chance that the wind speeds used for design will be exceeded in any given year.

Given a number of data points in a peak wind speed record and a known probability density function describing frequency of occurrence, an estimate of the $N$-year peak wind speed and standard deviation in the estimate that includes sampling errors can be determined using (Simiu et al., 1978)

$$\widehat{V}^N_{\text{3-sec}} = \overline{V}_{\text{3-sec}} + \sigma_{\text{3-sec}}(y - 0.5772)\frac{\sqrt{6}}{\pi}, \text{ and} \tag{3}$$

$$\text{SD}\left(\widehat{V}^N_{\text{3-sec}}\right) = [1.645 + 1.462(y - 0.5772) + 1.1(y - 0.5772)^2]^{0.5} \frac{0.78\ \sigma_{\text{3-sec}}}{\sqrt{n}}\ , \tag{4}$$

where

$y = -\ln\left[-\ln\left(1 - \frac{1}{N}\right)\right]$;
$N$ = mean recurrence interval (years) for peak wind in question;
$\widehat{V}^N_{\text{3-sec}}$ estimated value of the $N$-year, peak 3-second wind;
$\text{SD}\left(\widehat{V}^N_{\text{3-sec}}\right)$ = standard deviation in the estimate for the $N$-year 3-second wind;
$\overline{V}^N_{\text{3-sec}}$ = sample mean for 3-second peak winds measured;
$\sigma_{\text{3-sec}}$ = sample standard deviation for 3-second peak winds measured;
$n$ = sample size in years.

We define an interval for peak wind speeds using Equations (3) and (4). Our target for the design analysis is to set a 0.1% probability that the peak winds used to assign lateral wind load magnitudes will be exceeded. As mentioned earlier, this equates to a 1,000 year mean recurrence interval wind, or $N$ = 1,000 years. The estimated value of the 1,000-year wind and the standard deviation in the estimate based upon the 19-year sample size are computed using equations (3) and (4):

$$\widehat{V}^N_{\text{3-sec}} = 105.29 \text{ mph, and } \text{SD}\left(\widehat{V}^N_{\text{3-sec}}\right) = 11.450 \text{ mph.} \tag{5}$$

Using the values given in Equation (5), we can assign intervals for peak 3-second wind speeds in a highly flexible manner. For example, assume that we wish to have two standard deviations of confidence in the peak 3-second wind speed. The interval of wind speeds corresponding to this is

$$82.39 \leq \widehat{V}^N_{\text{3-sec}} \leq 128.19 \text{ mph} \quad \text{or} \quad \widehat{V}^N_{\text{3-sec}} = 105.29 \pm 22.9 \text{ mph.} \tag{6}$$

This wind speed interval can be interpreted as follows. There is a 99.9% confidence that the peak wind speed will be less than or equal to 105.29 mph. However, this estimate is based

upon limited peak wind speed data. Therefore, the error in the estimate that bounds this level of confidence in the expected peak wind has been defined as two standard deviations above and below the estimate. Thus, one has two standard deviations of confidence that the 1,000 year wind will not be exceeded. One can then say there is an acceptably low probability of the wind speed exceeding 128.19 mph.

Building codes (ASCE, 2002) use peak wind speeds of known averaging time to convert these speeds into design pressures for building structures. The expression to carry out this conversion is based upon the classical work of Bernoulli (ASCE, 2002),

$$q = 0.00256 \cdot K_z \cdot K_{zt} \cdot K_d \cdot V^2 \cdot I \ . \tag{7}$$

For the sake of simplicity, we assume

$I = 1.0$ (importance factor)
$K_d = 0.85$ (directionality factor)
$K_{zt} = 1.0$ (topographic effect factor)
$K_z = 0.70$ (height factor).

Using the peak wind speed interval of Equation (6), the corresponding interval for the peak pressures computed using Equation (7) is

$$10.34 \leq q^{\text{peak}} \leq 25.03 \text{ psf} \quad \text{or} \quad q^{\text{peak}} = 17.685 \pm 7.345 \text{ psf}.$$

If we assume a structural system layout that contains the portal frame shown in Figure 1, we can compute an interval for the peak applied lateral loads at the top of the frame. If we assume that the height of the frame is 12 feet and the lateral load resisting portal frames are 50 feet apart, the peak lateral loads are expected to lie within

$$3{,}102 \leq H \leq 7{,}509 \text{ lbs} \quad \text{or} \quad H = 5{,}305.5 \pm 2{,}203.5 \text{ lbs}.$$

### 2.2. Member Material and Cross-Sectional Properties

The loading is only one aspect to the uncertainty in structural engineering problems. Material and cross-sectional properties for component members within the structure are also subject to uncertainty. The portal frame shown in Figure 1 contains one beam member and two column members.

The beam members are W18×35, with mean cross-sectional area and second moment of area (AISC, 2001)

$$A_b = 10.3 \text{ in}^2 \quad \text{and} \quad I_b = 510 \text{ in}^4 \ .$$

(Cecen, 1974) and (Ravindra & Galambos, 1978) suggest statistical data for describing the fabrication-related variation in $A_b$ and $I_b$:

$\mu_F = 1.0$ (mean)
$V_F = 0.05$ (coefficient of variation),

which lead to

$\sigma_F = 0.05$ (standard deviation).

Cross-sectional properties are assumed to follow a normal statistical distribution (Ravindra & Galambos, 1978). Therefore, two standard deviations above and below the mean ensure approximately 95% confidence that the parameters lie within the stated interval. Therefore, mid-point and interval radii are

$$A_b = 10.3 \pm 1.03 \text{ in}^2 \quad \text{and} \quad I_b = 510 \pm 51 \text{ in}^4.$$

Columns are W10×49 members, with mean cross-section properties (AISC, 2001)

$$A_c = 14.4 \text{ in}^2 \quad \text{and} \quad I_c = 272 \text{ in}^4.$$

Using the same argument as that used for the beams above, intervals that consider uncertainty in the cross-sectional properties of the column members are

$$A_c = 14.4 \pm 1.44 \text{ in}^2 \quad \text{and} \quad I_c = 272 \pm 27.2 \text{ in}^4.$$

Uncertainty in material properties (e.g., material modulus) are often described using a normally distributed random variable (Ravindra & Galambos, 1978) with mean

$E = 29{,}000{,}000 \text{ lb/in}^2$.

(Cecen, 1974) and (Ravindra & Galambos, 1978) suggest the following statistical data for describing the variation in $E$:

$\mu_F = 1.0$ (mean), and
$V_F = 0.06$ (coefficient of variation),

which lead to

$\sigma_F = 0.06$ (standard deviation).

Two standard deviations above and below the mean ensure approximately 95% confidence that the true values of the parameters lie within the stated interval. Therefore, interval mid-point and radius are

$$E = 29{,}000{,}000 \pm 3{,}480{,}000 \text{ lb/in}^2.$$

### 2.3. Connections

The framework considered in Figure 1 also includes connections at the beam ends that are assumed to be partially restrained. These connections will not force the 90 degree angle made between beams and columns to remain 90 degrees after deformation of the frame laterally. These are often modeled as nonlinear springs. However, for simplicity and demonstration of concept, we assume the springs are linear.

Physical testing is used to determine the stiffness and strength characteristics of structural steel connections found in real structures. Unfortunately, there have been very few studies undertaken to quantify the statistical variation in connection response. (Deierlein et al., 1991) report examination of statistical parameters for a typical structural steel connection classified as partially-restrained. This connection is the top-and-seat angle connection with web cleats (TSAW).

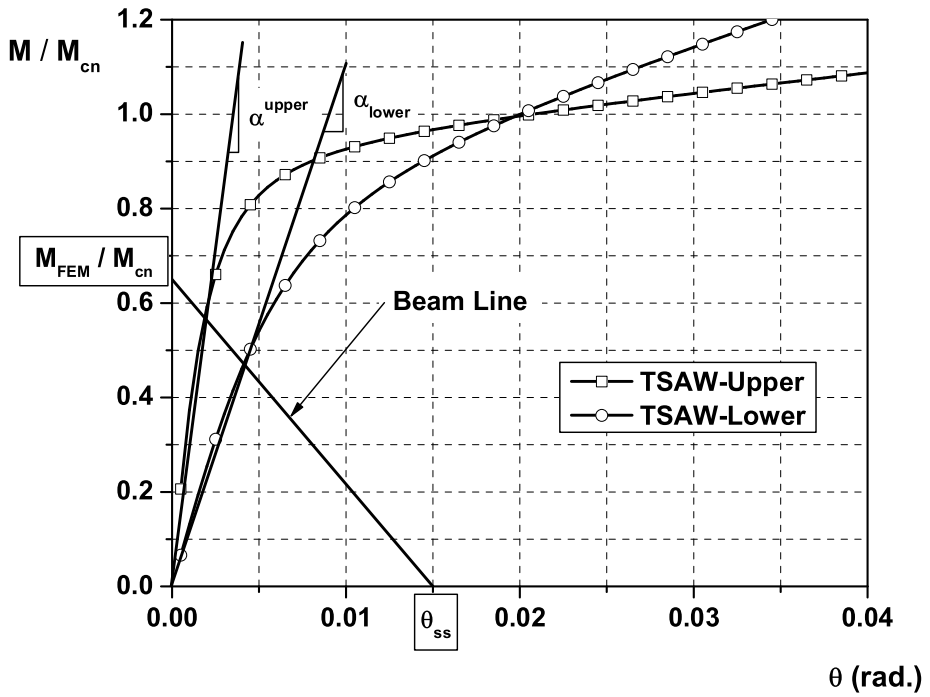



*Figure 2.* Beam-line approach for linearizing connection stiffness.

Uncertainty in nonlinear response of the TSAW connections was found to be adequately described using a normally distributed random variable (Deierlein et al., 1991). Linearization of connection response for purposes of structural analysis is commonly accomplished using the beam-line approach. The beam-line approach is schematically illustrated in Figure 2. The approach is well documented, and details will not be presented here. We assume that repeated loading and unloading of the connections results in shake-down to the linear connection stiffness established using the beam-line.

The connection stiffness uncertainty used in the present study is generated using the upper and lower-bound nonlinear connection curves for the TSAW connections discussed in (Deierlein et al., 1991). These two curves (shown in Figure 2) constitute boundaries for which there is 95% confidence that the expected connection behavior is captured. This corresponds to plus or minus two standard deviations from the mean. The connection curves are normalized with respect to the connection capacity, $M_{\rm cn}$. For the present study, $M_{\rm cn} = 0.4M_{\rm pb}$, where $M_{\rm pb}$ is the plastic moment capacity of the connected beam.

Using the beam-line concept and the W18×35 beam member, the linear connection stiffness magnitudes corresponding to the upper- and lower-bound connection curves are

$$\alpha^{\rm upper} = 403{,}965 \text{ k} \cdot \text{in/rad}$$
$$\alpha_{\rm lower} = 150{,}957 \text{ k} \cdot \text{in/rad}.$$

The midpoint and radius for the connection stiffness 95% confidence interval are

$$\alpha = 277{,}461 \pm 126{,}504 \text{ k} \cdot \text{in/rad} .$$

For simplicity, we assume that the connection of the columns to the foundation is rigid, although the present formulation can account for variability in connection response at the foundation.

## 3. Frame Components

We model each component of the frame shown in Figure 1 in an object-oriented manner, following the notation of (Hibbeler, 2002). We describe the structural components in an object-oriented manner, foreshadowing both the mathematical analysis to follow and the implementation in computer code.

### 3.1. Component: Member

Let $\cdot_N$ denote values at the near node and $\cdot_F$ denote values at the far node. We use $r_{N\hat{z}}$ instead of Hibbeler's $d_{N\hat{z}}$ to reserve $d_{N\hat{z}}$ for 3-dimensional frames.

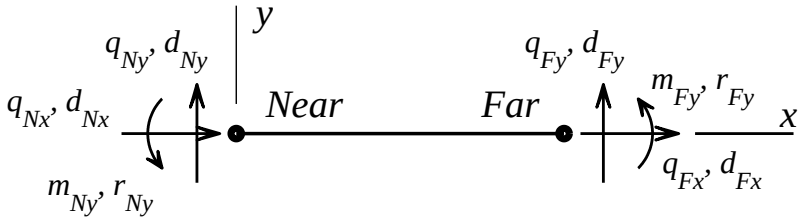



*Figure 3.* Member local forces, moments, displacements, and rotations (after Hibbeler 2002).

**Attributes** (in local ($\hat{\cdot}$) or global coordinates):

- Displacements: $d_{N\hat{x}}$, $d_{N\hat{y}}$, $d_{F\hat{x}}$, $d_{F\hat{y}}$ (local) or $d_{Nx}$, $d_{Ny}$, $d_{Fx}$, $d_{Fy}$ (global)
- Rotations: $r_{N\hat{z}}$, $d_{F\hat{z}}$ (local) or $r_{Nz}$, $d_{Fz}$ (global)
- Forces: $q_{N\hat{x}}$, $q_{N\hat{y}}$, $q_{F\hat{x}}$, $q_{F\hat{y}}$ (local) or $q_{Nx}$, $q_{Ny}$, $q_{Fx}$, $q_{Fy}$ (global)
- Moments: $m_{N\hat{z}}$, $m_{F\hat{z}}$ (local) or $m_{Nz}$, $m_{Fz}$ (global)

**Properties**: Frame-member stiffness equation:

$$\begin{bmatrix} \frac{AE}{L} & 0 & 0 & \frac{-AE}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & \frac{-12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & \frac{-6EI}{L^2} & \frac{2EI}{L} \\ \frac{-AE}{L} & 0 & 0 & \frac{AE}{L} & 0 & 0 \\ 0 & \frac{-12EI}{L^3} & \frac{-6EI}{L^2} & 0 & \frac{12EI}{L^3} & \frac{-6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & \frac{-6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \begin{bmatrix} d_{N\hat{x}} \\ d_{N\hat{y}} \\ r_{N\hat{z}} \\ d_{F\hat{x}} \\ d_{F\hat{y}} \\ r_{F\hat{z}} \end{bmatrix} = \begin{bmatrix} q_{N\hat{x}} \\ q_{N\hat{y}} \\ m_{N\hat{z}} \\ q_{F\hat{x}} \\ q_{F\hat{y}} \\ m_{F\hat{z}} \end{bmatrix}$$

or $\kappa(A, E, L, \mathbf{d}-\mathbf{q}) = \mathbf{k}'\mathbf{d}-\mathbf{q} = 0$. Typical values for frame parameters and applied loading are (see §2):

$$\begin{aligned} &E_b = E_c = 29{,}000{,}000 \pm 3{,}480{,}000 \text{ lbs/in}^2 \ (12\%) \\ &I_b = 510 \pm 51 \text{ in}^4; I_c = 272 \pm 27.2 \text{ in}^4 \ (10\%) \\ &A_b = 10.3 \pm 10.3 \text{ in}^2; A_c = 14.4 \pm 1.44 \text{ in}^2 \ (10\%) \\ &H = 5{,}305.5 \ \pm 2{,}203.5 \text{ lbs } (41.6\%) \\ &\alpha = 277{,}461{,}000 \ \pm 126{,}504{,}000 \text{ lb-in/rad } (45.6\%) \\ &L_c = 144 \text{ in}; \ L_b = 2L_c \ . \end{aligned} \tag{8}$$

Local coordinates are transformed to global coordinates by transformation matrices. For each member, let $\lambda_x = \cos\theta$ and $\lambda_y = \cos\phi$, so that $\lambda_x^2 + \lambda_y^2 = 1$. Let

$$T = \begin{bmatrix} \lambda_x & \lambda_y & 0 & 0 & 0 & 0 \\ -\lambda_y & \lambda_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_x & \lambda_y & 0 \\ 0 & 0 & 0 & -\lambda_y & \lambda_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Then $TT^T = T^T T =$ Identity.

### 3.2. Component: End

An "End" is an end of a Member (or a Joint). The Ends define the topology of the structure. Our End corresponds somewhat to the usual notion of a Node, except that we use "Connections" to join Members and Joints.

**Attributes** (in global coordinates):

- Displacements: $d_x$, $d_y$
- Rotations: $r_z$
- Forces: $q_x$, $q_y$, $q_{Ex}$, $q_{Ey}$

– Moments: $m_z$, $m_{Ez}$,

where $\cdot_E$ denotes externally applied forces and moments.

**Properties**:

– Can be incident with two or more Members and Joints
– Displacements $d_x$, $d_y$, and $r_z$ are equal for all Members and Joints incident on an End
– Forces $q_{Ex} + q_x$, $q_{Ey} + q_y$, and moments $m_{Ez} + m_z$ each sum to zero for all Members and Joints incident on an End

### 3.3. Component: Joint

**Attributes** (in local or global coordinates):

– Displacements: $d_{N\hat{x}}$, $d_{N\hat{y}}$, $d_{F\hat{x}}$, $d_{F\hat{y}}$
– Rotations: $r_{N\hat{z}}$, $d_{F\hat{z}}$
– Forces: $q_{N\hat{x}}$, $q_{N\hat{y}}$, $q_{F\hat{x}}$, $q_{F\hat{y}}$
– Moments: $m_{N\hat{z}}$, $m_{F\hat{z}}$

**Properties**:

– Length = 0
– Joins one Member to another
– Global displacements $d_x$ are equal for incident End and Member
– Global displacements $d_y$ are equal for incident End and Member
– Global forces $q_x$ are equal for incident End and Member
– Global forces $q_y$ are equal for incident End and Member
– Local rotations and moments satisfy

$$\begin{bmatrix} \alpha & -\alpha \\ -\alpha & \alpha \end{bmatrix} \begin{bmatrix} r_{N\hat{z}} \\ r_{F\hat{z}} \end{bmatrix} = \begin{bmatrix} m_{N\hat{z}} \\ m_{F\hat{z}} \end{bmatrix}$$

## 4. Assembly

Following the usual practice (e.g., (Hibbeler, 2002)), we assemble a linear system corresponding to the structure in Figure 4, identifying equal displacements and summing appropriate forces, to ensure that both equilibrium and compatibility of displacements are satisfied.

Columns 1 – 5:

$$\begin{bmatrix}
\frac{12E_cI_c}{L_c^3}+\frac{A_bE_b}{L_b} & 0 & \frac{6E_cI_c}{L_c^2} & 0 & 0 \\
0 & \frac{A_cE_c}{L_c}+\frac{12E_bI_b}{L_b^3} & 0 & \frac{6E_bI_b}{L_b^2} & \frac{6E_bI_b}{L_b^2} \\
\frac{6E_cI_c}{L_c^2} & 0 & \frac{4E_cI_c}{L_c}+\alpha & -\alpha & 0 \\
0 & \frac{6E_bI_b}{L_b^2} & -\alpha & \frac{4E_bI_b}{L_b}+\alpha & \frac{2E_bI_b}{L_b} \\
0 & \frac{6E_bI_b}{L_b^2} & 0 & \frac{2E_bI_b}{L_b} & \frac{4E_cI_c}{L_c}+\alpha \\
-\frac{A_bE_b}{L_b} & 0 & 0 & 0 & 0 \\
0 & -\frac{12E_bI_b}{L_b^3} & 0 & -\frac{6E_bI_b}{L_b^2} & -\frac{6E_bI_b}{L_b^2} \\
0 & 0 & 0 & 0 & -\alpha
\end{bmatrix}$$

Columns 6 – 8:

$$\begin{bmatrix}
-\frac{A_bE_b}{L_b} & 0 & 0 \\
0 & -\frac{12E_bI_b}{L_b^3} & 0 \\
0 & 0 & 0 \\
0 & -\frac{6E_bI_b}{L_b^2} & 0 \\
0 & -\frac{6E_bI_b}{L_b^2} & -\alpha \\
\frac{A_bE_b}{L_b}+\frac{12E_cI_c}{L_c^3} & 0 & \frac{6E_cI_c}{L_c^2} \\
0 & \frac{12E_bI_b}{L_b^3}+\frac{A_cE_c}{L_c} & -\frac{6E_bI_b}{L_b^2} \\
\frac{6E_cI_c}{L_c^2} & -\frac{6E_bI_b}{L_b^2} & \frac{4E_cI_c}{L_c}+\alpha
\end{bmatrix}
\begin{bmatrix} d2_x \\ d2_y \\ r2_z \\ r5_z \\ r6_z \\ d3_x \\ d3_y \\ r3_z \end{bmatrix}
=
\begin{bmatrix} H \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{9}$$

The global stiffness matrix $K$ given by Equation (9) has condition number cond($K$) = 4.7e+04. Solving using mid-point values of parameters given in Equation (8) yields

| | Displacement $d_x$ | Displacement $d_y$ | Rotation $r_z$ |
|---|---|---|---|
| Connection 2 | 0.15356843 | 0.00033236 | -0.00096285 |
| Connection 3 | 0.15102784 | -0.00033236 | -0.00094313 |
| Connection 5 | | | -0.00045995 |
| Connection 6 | | | -0.00044556 |

| | Force $q_x$ | Force $q_y$ | Moment $m_z$ |
|---|---|---|---|
| Connection 1 | -2670.516 | -963.856 | 245019.992 |
| Connection 4 | -2634.984 | 963.856 | 241381.602 |

The ranges given in Equation (8) for the parameters in this system suggest using interval arithmetic (Moore, 1966; Moore, 1979; Neumaier, 1990). Interval arithmetic computes with

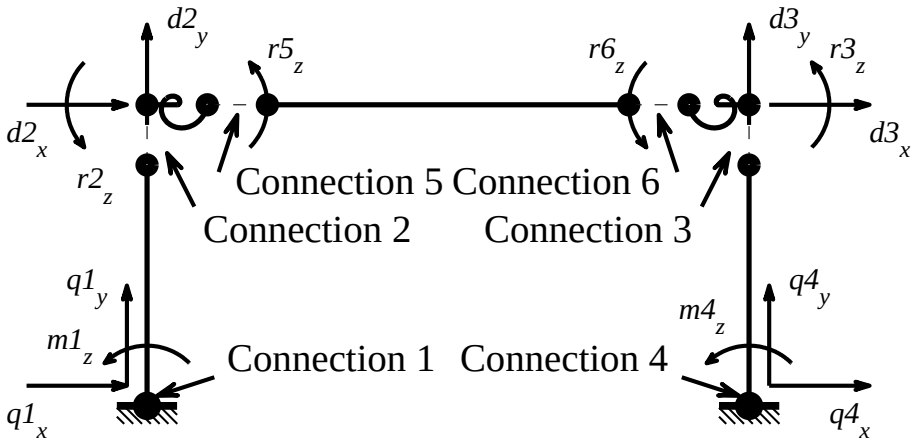



*Figure 4.* Element-by-element assembly.

guaranteed lower and upper bounds. It accounts for uncertain parameters and roundoff errors in computation. Our problem has uncertain parameters, and the condition number of 4.7e+04 for even such a simple frame suggests that roundoff is a potential concern, especially as we scale to larger structures.

In interval arithmetic, operations are defined set-wise. That is, if $[a] = [\underline{a}, \overline{a}]$ and $[b] = [\underline{b}, \overline{b}]$ are intervals,

$$[a] + [b] = \{a + b : a \in [a],\ b \in [b]\} = [\underline{a} + \underline{b}, \overline{a} + \overline{b}].$$

In a practical implementation, the additions of the endpoints are done using IEEE outwardly directed rounding. Other operations and elementary functions are defined similarly.

Initially, we use intervals of uncertainty 1% of those given in Equation (8). For example, instead of using $H = 5{,}305.5 \pm 2{,}203.5$ lbs, we use $5{,}305.5 \pm 22.035$ lbs. We form the global stiffness matrix $K$ given by Equation (9) using interval values of the parameters and solve.

Table I gives the naive interval solution of the one-bay frame problem. The column "Float" contains the floating point solutions to the system whose coefficients are given by the midpoints of the parameter intervals. The column "Interval" contains the solution computed by an interval linear equation solver applied to Equation (9) with interval coefficients. The column "Midpoint ± Radius" contains the same intervals as the column "Interval," except that they are expressed in a midpoint ± radius form, rather than an endpoint form.

The "Interval" solutions contain the true values, but narrower enclosures are better than wide ones. Naive interval computations, as we have done here, are prone to overestimation. For the rows labeled "Tight:," we solved the $2^{10}$ extremal individual problems formed by taking lower and upper bounds of the intervals for each of the 10 parameters in this system. Since the system is linear, the solution to **any** combination of parameter values taken from the respective intervals must lie in the convex hull of the solutions to the extremal problems. This is **not** simulation since parameter values are chosen not at random but as extremal values. The column "Relative overestimation" is the width of the "Interval"

Table I. Naive interval solution of the one-bay frame problem.

| Disp. | Float | Interval<br>True range | Midpoint ± Radius<br>Rel. overest. |
|---|---|---|---|
| $d2_x$ | 0.153568 | [ 0.09375783, 0.21337873 ] | 0.1535683 ± 0.05981 |
| | Tight: | [ 0.15237484, 0.15476814 ] | 76.34% |
| $d2_y$e+3 | 0.332364 | [ 0.19060424, 0.47412283 ] | 0.3323635 ± 0.1418 |
| | Tight: | [ 0.32940418, 0.33533906 ] | 83.52% |
| $r2_z$e+3 | -0.962852 | [ -1.3531968, -0.57250484 ] | -0.9628508 ± 0.3903 |
| | Tight: | [ -0.97085151, -0.95490139 ] | 79.42% |
| $r5_z$e+3 | -0.459955 | [ -0.6557609, -0.26414725 ] | -0.4599541 ± 0.1958 |
| | Tight: | [ -0.4638112, -0.45611532 ] | 83.47% |
| $r6_z$e+3 | -0.445563 | [ -0.64100045, -0.2501251 ] | -0.4455628 ± 0.1954 |
| | Tight: | [ -0.44930811, -0.4418354 ] | 86.05% |
| $d3_x$ | 0.151028 | [ 0.091230936, 0.21082444 ] | 0.1510277 ± 0.0598 |
| | Tight: | [ 0.14985048, 0.15221127 ] | 77.62% |
| $d3_y$e+3 | -0.332364 | [ -0.47412283, -0.19060424 ] | -0.3323635 ± 0.1418 |
| | Tight: | [ -0.33533906, -0.32940418 ] | 83.52% |
| $r3_z$e+3 | -0.943133 | [ -1.3330326, -0.55323186 ] | -0.9431322 ± 0.3899 |
| | Tight: | [ -0.95100335, -0.93531196 ] | 81.02% |

solution not contained in the "Tight" solution, scaled by the "Float" solution, and expressed as a percentage. Given that intervals are guaranteed to enclose the true answers, the goal is to compute enclosures with as little over-estimation as possible.

We observe

- "Interval" solutions contain the approximate "Float" solutions and the "Tight" solutions, illustrating the claim of enclosure.
- "Interval" solutions are hopelessly pessimistic. The relative over-estimations are too large to have practical utility.
- We used parameter uncertainties of 1% of the intervals given in Equation (8). If we use 4%, the interval linear solver fails because the global stiffness matrix includes matrices that are singular since we have perturbed by 4% elements of a matrix with condition 4.7e+04.
- In the Matlab environment we used, the "Interval" solution takes 1,200 times the CPU time for the approximate solution. That CPU cost should be compared with the CPU cost of 400,000 simulation runs, which do not provide the reliability of the interval results.

Rather than conclude interval arithmetic is not practical, we conclude that we must be more clever in its application. The rest of this paper leads us through a sequence of increasingly

sophisticated formulations until we are able to solve a system equivalent to Equation (9) with parameter uncertainties 1.5 times the widths of the intervals given in Equation 8.

## 5. Element-by-Element

The excessive overestimation in Table I comes from the "dependency problem" common in evaluation of expressions in interval arithmetic. For example, if we take $[x] = [-1, 2]$, Table II shows that even for some simple expressions, mathematically equivalent expressions do not give the same interval results because the set-wise definition of interval operations does not recognize that the same interval appearing in different contexts must be the same value. The interval operator $-$ cannot distinguish $[x]-[x]$, which equals 0, from $[x]-[y]$ with $[x] = [y]$, which does not. In general, expressions in which each variable appears only once (Single Use Expression, SUE) are evaluated with no over-estimation. In naive Gaussian elimination with back substitution applied to a system of order $n$, the coefficient $K_{1,1}$ appears in the symbolic expression for $d1_x$ a total of $\mathcal{O}(n^2)$ times, hardly a Single Use Expression.

Table II. Overestimation from dependencies in expressions with $[x] = [-1, 2]$.

| | | | | |
|---|---|---|---|---|
| $x - x$ | [-3, 3] | vs. | 0 | [0, 0] |
| $x * x$ | [-2, 4] | vs. | $x^2$ | [0, 4] |

(Mullen & Muhanna, 1999) suggested an element-by-element approach for structural engineering trusses. Instead of a finite element formulation, they introduced extra variables and added extra equations to the system to reduce the interval dependencies. We apply the Mullen-Muhanna element-by-element approach to frames. The difference is in the way we assemble the global stiffness matrix. From an object-oriented perspective, "End" becomes an inherent attribute of the Member and Joint classes. Each Member and Joint in the structure becomes its own block in the global stiffness matrix, with both displacements and forces at each end as unknowns. "Node" becomes a new Connector class, adding rows to the global stiffness matrix rows expressing that adjacent ends have identical displacements and rotations and that forces and moments at each connection sum to zero.

Joint $J_1$ global stiffness matrix:

$$
\begin{aligned}
& d3_x - d4_x = 0; \quad d3_y - d4_y = 0 \\
& \alpha r3_z - \alpha r4_z - m3_z = 0 \\
& q3_x + q4_x = 0; \quad q3_y + q4_y = 0 \\
& -\alpha r3_z + \alpha r4_z - m4_z = 0 \ .
\end{aligned}
$$

Member $M_1$ global stiffness matrix:

$$\begin{bmatrix} \frac{12E_cI_c}{L_c^3} & 0 & -\frac{6E_cI_c}{L_c^2} & -\frac{12E_cI_c}{L_c^3} & 0 & -\frac{6E_cI_c}{L_c^2} \\ 0 & \frac{A_cE_c}{L_c} & 0 & 0 & -\frac{A_cE_c}{L_c} & 0 \\ -\frac{6E_cI_c}{L_c^2} & 0 & \frac{4E_cI_c}{L_c} & \frac{6E_cI_c}{L_c^2} & 0 & \frac{2E_cI_c}{L_c} \\ -\frac{12E_cI_c}{L_c^3} & 0 & \frac{6E_cI_c}{L_c^2} & \frac{12E_cI_c}{L_c^3} & 0 & \frac{6E_cI_c}{L_c^2} \\ 0 & -\frac{A_cE_c}{L_c} & 0 & 0 & \frac{A_cE_c}{L_c} & 0 \\ -\frac{6E_cI_c}{L_c^2} & 0 & \frac{2E_cI_c}{L_c} & \frac{6E_cI_c}{L_c^2} & 0 & \frac{4E_cI_c}{L_c} \end{bmatrix} \begin{bmatrix} d1_x \\ d1_y \\ r1_z \\ d2_x \\ d2_y \\ r2_z \end{bmatrix} - \begin{bmatrix} q1_x \\ q1_y \\ m1_z \\ q2_x \\ q2_y \\ m2_z \end{bmatrix} = 0 \ .$$

Members $M_2$ and $M_3$ and Joint $J_2$ are handled similarly.

Connector $C_2$ (we'll see $C_1$ later) connecting Member $M_1$ with Joint $J_1$ requires equality of incident displacements:

$$[d2_x, d2_y, r2_z]^T - [d3_x, d3_y, r3_z]^T = 0 \ ,$$

and that incident forces sum to zero:

$$q2_x + q3_x = H; \quad q2_y + q3_y = 0; \quad m2_z + m3_y = 0 \ .$$

This is the first non-zero right hand side so far. Connector $C_3$ connecting Joint $J_1$ with Member $M_2$ requires equality of incident displacements and that incident forces sum to zero:

$$[d4_x, d4_y, r4_z]^T - [d5_x, d5_y, r5_z]^T = 0$$
$$[q4_x + q5_x, q4_y + q5_y, m2_z + m3_y]^T = 0 \ .$$

Connectors $C_4$ and $C_5$ are handled similarly. Connector $C_1$ fixes Member $M_1$ to the ground, and connector $C_6$ fixes Member $M_3$:

$$[d1_x, d1_y, r1_z]^T = 0; \quad [d10_x, d10_y, r10_z]^T = 0 \ .$$

For simplicity of exposition, we retain the last two sets of equations corresponding to displacements and rotations known to be zero. The solution of the element-by-element global stiffness system using intervals of uncertainty 1% of those given in Equation (8) in interval arithmetic is shown in Table III. The condition number is 1.2e+17. This condition number leads one to suspect that, in exact arithmetic, the matrix may be exactly singular.

With such a large condition number, it is surprising that we get essentially the same answers as before, but it is disappointing that the interval radii are not significantly smaller than for the naive interval solution shown in Table I. However, there are many common terms in many of the matrix coefficients. For example, see the global stiffness matrix for Member $M_1$. We can factor them out and take advantage of subdistributivity.

Table III. Interval solution the Mullen-Muhanna element-by-element approach.

| Disp. | Float | Interval<br>True range | Midpoint ± Radius<br>Rel. overest. |
|---|---|---|---|
| $d2_x$ | 0.153568 | [ 0.09246203, 0.21467453 ] | 0.1535683 ± 0.06111 |
| | Tight: | [ 0.15237484, 0.15476814 ] | 78.02% |
| $d2_y$e+3 | 0.332364 | [ 0.18751797, 0.4772091 ] | 0.3323635 ± 0.1448 |
| | Tight: | [ 0.32940418, 0.33533906 ] | 85.38% |
| $r2_z$e+3 | -0.962852 | [ -1.361667, -0.56403468 ] | -0.9628508 ± 0.3988 |
| | Tight: | [ -0.97085151, -0.95490139 ] | 81.18% |
| $r5_z$e+3 | -0.459955 | [ -0.66002154, -0.25988661 ] | -0.4599541 ± 0.2001 |
| | Tight: | [ -0.4638112, -0.45611532 ] | 85.32% |

## 6. Subdistributivity

In interval arithmetic, we have

$$a(b+c) \subseteq ab + ac \quad \text{(subdistributivity)}.$$

For example, $[-1,2]*([4,5]+[-3,-2]) = [-3,6] \subseteq [-1,2]*[4,5]+[-1,2]*[-3,-2]$. Hence, to get tighter enclosures, we want to extract common factors whenever possible, as suggested by (Mullen & Muhanna, 1999) for trusses.

For example, consider equations 9 and 12 from the Joint $J_1$ stiffness matrix and equations 21 and 24 from the Joint $J_2$ stiffness matrix. Let $d_{61} := r3_z - r4_z$ and $d_{62} := r7_z - r8_z$. Then

$$\begin{aligned}
&\text{Eq. 9 \& 21: } \alpha d_{61} - m3_z = 0; \quad \alpha d_{62} - m7_z = 0\\
&\text{Eq. 12: } \alpha d_{61} + m4_z = 0 \text{ or } m3_z + m4_z = 0\\
&\text{Eq. 24: } \alpha d_{62} + m8_z = 0 \text{ or } m7_z + m8_z = 0\\
&\text{Eq. 61 \& 62: } r3_z - r4_z - d_{61} = 0; \quad r7_z - r8_z - d_{62} = 0\ .
\end{aligned}$$

Next, consider the Member $M_1$ global stiffness matrix:

$$\begin{bmatrix}
\frac{12E_cI_c}{L_c^3} & 0 & -\frac{6E_cI_c}{L_c^2} & -\frac{12E_cI_c}{L_c^3} & 0 & -\frac{6E_cI_c}{L_c^2}\\
0 & \frac{A_cE_c}{L_c} & 0 & 0 & -\frac{A_cE_c}{L_c} & 0\\
-\frac{6E_cI_c}{L_c^2} & 0 & \frac{4E_cI_c}{L_c} & \frac{6E_cI_c}{L_c^2} & 0 & \frac{2E_cI_c}{L_c}\\
-\frac{12E_cI_c}{L_c^3} & 0 & \frac{6E_cI_c}{L_c^2} & \frac{12E_cI_c}{L_c^3} & 0 & \frac{6E_cI_c}{L_c^2}\\
0 & -\frac{A_cE_c}{L_c} & 0 & 0 & \frac{A_cE_c}{L_c} & 0\\
-\frac{6E_cI_c}{L_c^2} & 0 & \frac{2E_cI_c}{L_c} & \frac{6E_cI_c}{L_c^2} & 0 & \frac{4E_cI_c}{L_c}
\end{bmatrix}
\begin{bmatrix} d1_x\\ d1_y\\ r1_z\\ d2_x\\ d2_y\\ r2_z \end{bmatrix}
-
\begin{bmatrix} q1_x\\ q1_y\\ m1_z\\ q2_x\\ q2_y\\ m2_z \end{bmatrix}
= 0\ .$$

Let

$$d_{63} := \frac{A_c E_c}{L_c}(d1_y - d2_y) \qquad d_{65} := \frac{2E_c I_c}{L_c}(r1_z + r2_z)$$
$$d_{64} := \frac{6E_c I_c}{L_c^2}(d1_x - d2_x) \qquad d_{66} := \frac{2E_c I_c}{L_c} r1_z \ ,$$

which leads to a considerably simpler system

$$\begin{aligned}
&\text{Eq. 1}: \frac{2}{L_c} d_{64} - \frac{3}{L_c} d_{65} - q1_x = 0 \\
&\text{Eq. 2}: d_{63} - q1_y = 0 \\
&\text{Eq. 3}: \ -d_{64} + d_{65} + d_{66} - m1_z = 0 \\
&\text{Eq. 4 \& 5}: q1_x + q2_x = 0; \quad q1_y + q2_y = 0 \\
&\text{Eq. 6}: \ -d_{64} + 2d_{65} - d_{66} - m2_z = 0 \\
&\text{Eq. 63}: d1_y - d2_y - \frac{L_c}{A_c E_c} d_{63} = 0 \\
&\text{Eq. 64}: d1_x - d2_x - \frac{L_c^2}{6E_c I_c} d_{64} = 0 \\
&\text{Eq. 65}: r1_z + r2_z - \frac{L_c}{2E_c I_c} d_{65} = 0 \\
&\text{Eq. 66}: r1_z - \frac{L_c}{2E_c I_c} d_{66} = 0 \ .
\end{aligned}$$

The global stiffness matrices for Members $M_2$ and $M_3$ are handled similarly. The solution of the element-by-element global stiffness system using intervals of uncertainty 1% of those given in Equation (8) in interval arithmetic is shown in Table IV. Cond($K$) = 1.2e+17.

Table IV. Interval solution the Mullen-Muhanna element-by-element approach.

| Disp. | Float | Interval<br>True range | Midpoint ± Radius<br>Rel. overest. |
|---|---|---|---|
| $d2_x$ | 0.153568 | [ 0.15206288, 0.15507492 ] | 0.1535689 ± 0.001506 |
| | Tight: | [ 0.15237484, 0.15476814 ] | 0.40% |
| $d2_y$e+3 | 0.332364 | [ 0.32918317, 0.33554758 ] | 0.3323654 ± 0.003182 |
| | Tight: | [ 0.32940418, 0.33533906 ] | 0.13% |
| $r2_z$e+3 | -0.962852 | [ -0.97485786, -0.95084958 ] | -0.9628537 ± 0.012 |
| | Tight: | [ -0.97085151, -0.95490139 ] | 0.84% |
| $r5_z$e+3 | -0.459955 | [ -0.46757208, -0.45234116 ] | -0.4599566 ± 0.007615 |
| | Tight: | [ -0.4638112, -0.45611532 ] | 1.63% |

These results in Table IV are about two orders of magnitude tighter than the interval element-by-element method shown in Table III. Further, we can solve the system with relative uncertainty 1.5 times the intervals of uncertainty given in Equation (8), compared with 0.01 before. That allows us to handle practical engineering tolerances.

So far, we have replicated the work of Mullen and Muhanna, except applied to frames instead of trusses. Their work speaks of factoring the stiffness equations, although they might not have done so in exactly the same way we have done it. To further reduce interval over-estimation, we try scaling the equations and applying constraint propagation.

## 7. Scaling

In traditional numerical analysis, seeing solution components varying over eight orders of magnitude and a condition number of 1.2e+17 is a warning sign. Let's try scaling variables to have similar magnitudes.

It appears that scaling all forces by the input force $H = 5305.5$ would be good. We cannot do that by simply replacing $H = 5305.5$ by $H = 1$ because $H$ appears only on the right hand side, so that replacement would have no effect on cond($K$).

In the right hand side, we can replace the interval $[H]$ by its midpoint $\tilde{H}$ and then multiply the solution by $[H]/\tilde{H} = [1-\delta, 1+\delta]$. That replaces one of the interval parameters by a degenerate interval in the computation of the solution. The result is a very slight further reduction in the uncertainties of the solution.

Next, in each equation, we replace each variable force ($q1_x$, ...) by force/$\tilde{H}$. Each of the intermediate variables $d_{61}$, ..., $d_{74}$ introduced in the Subdistributivity section are of the same order as forces, so we scale them by $\tilde{H}$, too. Hence in the global stiffness matrix, each coefficient of a force or a newly introduced intermediate variable is multiplied by $\tilde{H}$, unless all of the terms of that equation are in the scaled set. That reduces cond($K$) from about 1.7e+17 to 2.8e+08 and yields a further reduction in the widths the Mullen-Muhanna results of Table IV, as shown in Table V.

Table V. Interval solution using scaled element-by-element approach with parameter uncertainties 1% of those in Equation (8).

| Disp. | Float | Interval<br>True range | Midpoint ± Radius<br>Rel. overest. |
|---|---|---|---|
| $d2_x$ | 0.153568 | [ 0.15294597, 0.15419182 ] | 0.1535689 ± 0.0006229 |
| | Tight: | [ 0.1531698, 0.15396904 ] | 0.29% |
| $d2_y$e+3 | 0.332364 | [ 0.33111682, 0.33361393 ] | 0.3323654 ± 0.001249 |
| | Tight: | [ 0.3311227, 0.33360764 ] | 0.004% |
| $r2_z$e+3 | -0.962852 | [ -0.96945816, -0.95624927 ] | -0.9628537 ± 0.006604 |
| | Tight: | [ -0.96583881, -0.95988319 ] | 0.75% |
| $r5_z$e+3 | -0.459955 | [ -0.46515166, -0.45476159 ] | -0.4599566 ± 0.005195 |
| | Tight: | [ -0.46141645, -0.45849491 ] | 1.62% |

Table VI shows the solution to the same system as Table V, except that we use the practical parameter uncertainties given in Equation (8). If we multiply the uncertainties given in Equation (8) by 1.7, we get solution enclosures shown in Table VII. In either case,

we can compute bounds, but bounds are quite over-estimated, and they include values of the wrong sign, an observation which leads us to consider constraint propagation.

Table VI. Interval solution using scaled element-by-element approach with parameter uncertainties from Equation (8).

| Disp. | Float | Interval<br>True range | Midpoint ± Radius<br>Rel. overest. |
|---|---|---|---|
| $d2_x$ | 0.153568 | [ 0.022924888, 0.29366922 ] | 0.1582971 ± 0.1354 |
| | Tight: | [ 0.12130751, 0.20804041 ] | 119.8% |
| $d2_y$e+3 | 0.332364 | [ 0.11407836, 0.57891094 ] | 0.3464947 ± 0.2324 |
| | Tight: | [ 0.21526742, 0.47234932 ] | 62.51% |
| $r2_z$e+3 | -0.962852 | [ -2.5286276, 0.55857565 ] | -0.985026 ± 1.544 |
| | Tight: | [ -1.4124783, -0.73502904 ] | 250.3% |
| $r5_z$e+3 | -0.459955 | [ -1.7359689, 0.77691797 ] | -0.4795255 ± 1.256 |
| | Tight: | [ -0.6216869, -0.3157181 ] | 479.8% |

Table VII. Interval solution using scaled element-by-element approach with parameter uncertainties 1.7 times of those in Equation (8).

| Disp. | Float | Interval<br>True range | Midpoint ± Radius<br>Rel. overest. |
|---|---|---|---|
| $d2_x$ | 0.153568 | [ -2.4422952, 2.7778163 ] | 0.1677605 ± 2.61 |
| | Tight: | [ 0.10506254, 0.29253671 ] | 3277% |
| $d2_y$e+3 | 0.332364 | [ -3.726848, 4.4766635 ] | 0.3749077 ± 4.102 |
| | Tight: | [ 0.12406102, 0.59793831 ] | 2326% |
| $r2_z$e+3 | -0.962852 | [ -33.038469, 30.980172 ] | -1.029148 ± 32.01 |
| | Tight: | [ -2.2934663, -0.62850325 ] | 6476% |
| $r5_z$e+3 | -0.459955 | [ -27.257812, 26.220056 ] | -0.5188781 ± 26.74 |
| | Tight: | [ -0.76558971, -0.18556419 ] | 11500% |

CPU times for the $58 \times 58$ interval solution shown in Tables V - VII are about 2,000 times the CPU time required to solve the approximate $8 \times 8$ system of Equation (9) with midpoint values of the parameters. The figure of 2,000 times can reasonably be compared with the nearly 400,000 simulation runs to achieve even comparable confidence intervals. Nonetheless, the cost of the interval computation is at least partially due to the Matlab programming environment in which these experiments were performed. Direct coding with a language with an interval datatype, even if that datatype is implemented with operator overloading, probably would result in an order-of-magnitude speedup of the interval computations. These results also point to a need for a quality suite of interval sparse matrix routines.

## 8. Constraint Propagation

The solution enclosures in Tables VI and VII show large uncertainty, and they include non-physical values, e.g., compression rather than tension. The stiffness matrix is close to singular. Constraint propagation can help, because we can always intersect with physically known constraints.

Constraint propagation originated in the field of logic programming. (Van Hentenryck et al., 1997) is an excellent explanation of constraint propagation in an interval context. The idea is best illustrated by an example. Suppose we seek roots in $[-4, 4]$ of $f(x) = x^2+x-5 = 0$. Solve for the linear occurrence of $x$, $x = 5 - x^2$. On the right, substitute $x = [-4, 4]$: $x = 5 - [-4, 4]^2 = 5 - [0, 16] = [-11, 5]$. That is, if a root $x^*$ of $x^2 + x - 5 = 0$ lies in the interval $[-4, 4]$, then it must also lie in the interval $[-11, 5]$, not an especially helpful result.

Next, solve for the quadratic occurrence of $x$, $x = \pm\sqrt{5 - x}$. On the right, substitute $x = [-4, 4]$: $x = \pm\sqrt{5 - [-4, 4]} = \pm\sqrt{[1, 9]} = [-3, -1] \cup [1, 3]$. That is, if a root $x^*$ of $x^2 + x - 5 = 0$ lies in the interval $[-4, 4]$, then it must also lie in the interval $[-3, -1]$ or the interval $[1, 3]$. Further iteration of $x = \sqrt{5 - x}$ from $x = [1, 3]$ yields

$$\begin{aligned} x \;=\; & [1.41421356237309, 2.00000000000000] \\ & [1.73205080756887, 1.89361728911280] \\ & [1.76249332222485, 1.80774699347866] \\ & [1.78668771936265, 1.79930727719730] \\ & [1.78904799343189, 1.79257141577047] \\ & [1.79092953078269, 1.79191294614669] \;, \end{aligned}$$

which is converging to the root $x^* = 1.79128784747792$.

Constraint propagation can be viewed as discarding candidate solutions that are infeasible with respect to already known information. It is Gauss-Seidel iteration, except that we solve each equation for each variable. Constraint propagation is especially attractive for sparse systems, such as ours.

To describe a generic constraint propagation algorithm for a linear system $Ax = b$, we denote the set of unknowns by $x = (x_i)$ and use $A_j x = b_j$, for a single equation.

```
Loop until converged
    Loop for each j
        In principle, consider equation j: A_j x = b_j
        For each variable x_i which appears in A_j x = b_j, "solve" for x_i
            Let [x_i] := [x_i] ⋂ expression for x_i evaluated with [x]
            If the intersection is empty, there is no solution. STOP
    If any intersection is smaller, the solution is converging
```

There are various strategies for choosing the order of iteration of equations and variables and many implementation details, which we ignore here.

Using the the practical parameter uncertainties given in Equation (8) and starting with the solution enclosures shown in Table VI, we get the results shown in Table VIII. Constraint

propagation tightened the enclosure of $d2_y$, but did not improve significantly on the quality of the interval solution.

Table VIII. Constraint propagation starting with solutions from Table VI.

| Disp. | Float | Interval<br>True range | Midpoint ± Radius<br>Rel. overest. |
|---|---|---|---|
| $d2_x$ | 0.153568 | [ 0.022924888, 0.29366922 ] | 0.1582971 ± 0.1354 |
| | Tight: | [ 0.12130751, 0.20804041 ] | 119.8% |
| $d2_y$e+3 | 0.332364 | [ 0.17032525, 0.57868876 ] | 0.374507 ± 0.2042 |
| | Tight: | [ 0.21526742, 0.47234932 ] | 45.52% |
| $r2_z$e+3 | -0.962852 | [ -2.5286276, 0.55857565 ] | -0.985026 ± 1.544 |
| | Tight: | [ -1.4124783, -0.73502904 ] | 250.3% |
| $r5_z$e+3 | -0.459955 | [ -1.7359689, 0.77691797 ] | -0.4795255 ± 1.256 |
| | Tight: | [ -0.6216869, -0.3157181 ] | 479.8% |

Starting with $[0.5, 1.5]$ times the approximate solution using midpoint values, using **no** interval system solver, and five iterations of constraint propagation, we get the results shown in Table IX. We achieve relative over-estimations that are comparable to the relative uncertainties in the parameters, in spite of a condition number of 2.8e+08.

Table IX. Constraint propagation starting with $[0.6, 1.4]$ times the approximate solution using midpoint values.

| Disp. | Float | Interval<br>True range | Midpoint ± Radius<br>Rel. overest. |
|---|---|---|---|
| $d2_x$ | 0.153568 | [ 0.076784216, 0.23035265 ] | 0.1535684 ± 0.07678 |
| | Tight: | [ 0.12130751, 0.20804041 ] | 43.52% |
| $d2_y$e+3 | 0.332364 | [ 0.166182, 0.49854599 ] | 0.332364 ± 0.1662 |
| | Tight: | [ 0.21526742, 0.47234932 ] | 22.65% |
| $r2_z$e+3 | -0.962852 | [ -1.4442773, -0.48142577 ] | -0.9628515 ± 0.4814 |
| | Tight: | [ -1.4124783, -0.73502904 ] | 29.64% |
| $r5_z$e+3 | -0.459955 | [ -0.68993206, -0.22997735 ] | -0.4599547 ± 0.23 |
| | Tight: | [ -0.6216869, -0.3157181 ] | 33.48% |

## 9. Conclusions and Future Directions

To the structural engineering community, along with the work of (Mullen & Muhanna, 1999) and (Muhanna & Mullen, 2001), we have demonstrated the feasibility of interval techniques. One set of interval computations can guarantee to enclose the displacements, rotations, forces, and moments that could be observed from any combination of values of

cross-sectional properties, loading, material properties, and connections, even for quite large uncertainties. In subsequent work, we will extend these techniques to non-linear behaviors and to larger structures.

To the interval community, we have demonstrated a variety of techniques to achieve relatively tight enclosures of the solution to a realistic (although small) problem, even in the face of parameter uncertainties over 40%. We used an element-by-element approach, which adds equations specifying that two variables are the same, rather than simplifying by identifying them with the same variable. We used symbolic rearrangement, scaling of the equations, and constraint propagation. In subsequent work, to handle larger systems, we will explore sparsity-preserving preconditioning, more effective and efficient constraint propagation, and branch-and-bound-like strategies for subdividing the ranges of wide interval-valued parameters.

## References


AISC. Manual of Steel Construction, Load and Resistance Factor Design, 3rd Edition. American Institute of Steel Construction, Chicago, Ill., 2001.

ASCE. Minimum design loads for buildings and other structures, ASCE 7–02. American Society of Civil Engineers, Reston, Va., 2002.

H. Cecen. Computer-aided, reliability based, optimum design of multi-story steel frames. Master's thesis, Washington University, St. Louis, Mo., 1974.

C.A. Cornell. A probability-based structural code. *Journal of the American Concrete Institute*, 66(12), December 1969.

Gregory G. Deierlein, S.-H. Hsieh, Y.-J. Shen, and John F. Abel. Nonlinear analysis of steel frames with semi-rigid connections using capacity spectrum approach. Technical Report NCEER–91–0008, National Center for Earthquake Engineering Research, SUNY at Buffalo, Buffalo, NY, 1991.

Russell C. Hibbeler. *Structural Analysis, Fifth Edition.* Prentice Hall, Englewood Cliffs, NJ, 2002.

Robert E. Melchers. *Structural Reliability Analysis and Prediction, 2nd Edition.* John Wiley and Sons, New York, 2001.

Ramon E. Moore. *Interval Analysis.* Prentice Hall, Englewood Cliffs, NJ, 1966.

Ramon E. Moore. *Methods and Applications of Interval Analysis.* SIAM, Philadelphia, PA, 1979.

Rafi L. Muhanna and Robert L. Mullen. Uncertainty in mechanics problems – interval-based approach. *Journal of Engineering Mechanics*, 2001.

Robert L. Mullen and Rafi L. Muhanna. Bounds of structural response for all possible loading conditions. *Journal of Structural Engineering*, 1999.

Arnold Neumaier. *Interval Methods for Systems of Equations.* Cambridge University Press, Cambridge, 1990.

Mayasandra K. Ravindra and Theodore V. Galambos. Load and resistance factor design for steel. *Journal of the Structural Division*, 104, 1978.

Emil Simiu, Jacques Bietry, and James J. Filliben. Sampling errors in estimation of extreme winds. *Journal of the Structural Division*, 104(3), 1978.

T.T. Soong and Mircea Grigoriu. *Random Vibration of Mechanical and Structural Systems.* Prentice Hall, Englewood Cliffs, NJ, 1993.

Pascal Van Hentenryck and Laurent Michel and Yves Deville. *Numerica: A Modeling Language for Global Optimization.* MIT Press, Cambridge, Mass., 1997.

# Effects of Error, Variability, Testing and Safety Factors on Aircraft Safety

E. Acar, A. Kale and R.T. Haftka
*Department of Mechanical and Aerospace Engineering,University of Florida, Gainesville, FL 32611-6250*
*e-mail: eacar@ufl.edu, akale@ufl.edu and haftka@ufl.edu*

**Abstract:** In this paper we aim to clarify the interaction between error, variability, testing and safety factors on the safety of aircraft structures by using an error model that includes errors made in the calculation of loads and stresses, and also errors in material and geometric parameters. The effect of various representative safety measures taken while designing aircraft structures following the deterministic approach codes in the FAA regulations is investigated. Uncertainties include errors, such as in predicting the response (stress, deflection etc.) of the structure and variability in materials, loading and geometry. Two error models, one is simple and the other is more detailed, are used and the results of these two models are compared. We use a simple model of failure of a representative aircraft structure. In addition, we explore the effectiveness of certification tests for improving safety. It is found that certification tests reduce the calculated failure probabilities by updating the modeling error. We find that these tests are most effective when safety factors are low and when most of the uncertainty is due to systemic errors rather than variability.

## Nomenclature

| | | |
|---|---|---|
| *c.o.v.* | = | Coefficient of variation |
| $e_m$, $e_P$, $e_\sigma$, $e_t$ and $e_w$ | = | Error factor for material failure stress, load, stress, thickness and width |
| $e_{total}$ | = | Cumulative effect of various errors |
| $P_{act}$, $P_{calc}$ *and* $P_d$ | = | Actual, calculated and design load |
| $\sigma_{f\,design}$, $t_{design}$ and $w_{design}$ | = | Design values of failure stress, thickness and width |
| $\sigma_{f\,built}$, $t_{built}$ and $w_{built}$ | = | Average values of failure stress, thickness and width of the components built by an aircraft company |
| $\sigma_{f\,actual}$, $t_{actual}$ and $w_{actual}$ | = | Actual values of failure stress, thickness and width |
| $S_{F\,avg}$ | = | Fleet-average safety factor |
| $k$ | = | Error multiplier |
| $\overline{P}_{nt}$ and $\overline{P}_t$ | = | Average value of probability of failure without and with certification |
| MEF and SEF Cases | = | Multiple Error Factor and Single Error Factor Cases |

## 1. Introduction

Aerospace structures have traditionally been designed using a deterministic approach based on FAA regulations. The safety of the structures has been achieved by combining safety factors with tests of material and structural components. There is a growing interest to replace safety factors by reliability-based design. However, there is no consensus on how to make transition from deterministic design to reliability-based design. An important step in this transition is to understand the way safety is built into aircraft structures now, via explicit safety factors, use of conservative material properties and by testing. Safety measures are intended to compensate for errors and variability. Errors reflect inaccurate modeling of physical phenomena, errors in structural analysis, errors in load calculations, or use of materials and tooling in construction that are different from those specified by the designer. Thus, the errors affect all copies of structural components in the entire fleet of aircraft of the same model. On the other hand, variability reflects variation in material properties, geometry, or loading between different copies of the same structure on different aircraft in the fleet.

Our previous paper (Kale et al, 2004) sought to clarify the interaction between the error, variability and testing on the overall probability of failure. We started with a structural design employing all considered safety measures. The effect of variability in geometry, loads, and material properties was incorporated by the appropriate random variables. For errors we used a simplified model that represented the overall error by a single random variable used in the calculation of stress. In this paper, we use a more detailed model in which we consider individual error components in load calculation, stress calculation, material properties and geometry parameters. The objective of the paper is to observe differences between the use of the simple model and the more detailed model.

As in our previous paper, we transform the errors into random variables by considering the design of multiple aircraft models. As a consequence, for each model the structure is different. It is as if we pretend that there are hundreds of companies (Airbus, Boeing, Bombardier, Embraer etc.) each designing essentially the same airplane, but each having different errors in their structural analysis and manufacturing. For each model we simulate certification testing. If the airplane passes the test, then an entire fleet of airplanes with the same design is assumed to be built with different members of the fleet having different geometry, loads, and material properties based on assumed models for variability in these properties. That is, the uncertainty due to variability is simulated by considering multiple realizations of the same design, and the uncertainty due to errors is simulated by designing different structures to carry the loads specified by the FAA.

We consider only stress failure due to extreme loads, which can be simulated by an unstiffened panel designed under uniaxial loads. No testing of components prior to certification is analyzed for this simple example.

## 2. Structural uncertainties

A good analysis of different sources of uncertainty is provided by Oberkampf et al. (2000). Here we simplify the classification with a view to the question of how to control uncertainty. We propose in Table 1 a classification that distinguishes between (1) uncertainties that apply equally to the entire fleet of an aircraft model and (2) uncertainties that vary for the individual aircraft. The distinction is important because safety measures usually target one or the other.

Similarly, the uncertainty in the failure of a structural member can also be divided into two types: systemic errors and variability. Systemic errors reflect inaccurate modeling of physical phenomena, errors in structural analysis, errors in load calculations, or use of materials and tooling in construction that are different from those specified by the designer. Systemic errors affect

all the copies of the structural components built using the same model and are therefore fleet-level uncertainties. The other type of uncertainty reflects variability in material properties, geometry, or loading between different copies of the same structure and is called here individual uncertainty.

**Table 1. Uncertainty Classification**

| Type of uncertainty | Spread | Cause | Remedies |
|---|---|---|---|
| Systemic error (modeling errors) | Entire fleet of components designed using the model | Errors in predicting structural failure and differences between properties used in design and average fleet properties. | Testing and simulation to improve math model and the solution. |
| Variability | Individual component level | Variability in tooling, manufacturing process, and flying environments. | Improve tooling and construction. Quality control. |

## 3. Safety Measures

Aircraft structural design is still done by and large using code-based design rather than probabilistic approaches. Safety is improved through conservative design practices that include use of safety factors and conservative material properties. It is also improved by tests of components and certification tests that can reveal inadequacies in analysis or construction. In the following we detail some of these safety measures.

**Safety Margin:** Traditionally all aircraft structures are designed with a safety factor to withstand 1.5 times the limit load without failure.

**A-Basis Properties:** In order to account for uncertainty in material properties, the Federal Aviation Administration (FAA) recommends the use of conservative material properties. This is determined by testing a specified number of coupons selected at random from a batch of material. The A-basis property is determined by calculating the value of a material property exceeded by 99% of the population with 95% confidence.

**Component and Certification tests:** Component tests and certification tests of major structural components reduce stress and material uncertainties for given extreme loads due to inadequate structural models. These tests are conducted in a building block procedure. First, individual coupons are tested, and then a sub assembly is tested followed by a full-scale test of the entire structure. Since these tests cannot apply every load condition to the structure, they leave uncertainties with respect to some loading conditions. It is possible to reduce the probability of failure by performing more tests to reduce uncertainty or by extra structural weight to reduce stresses. If certification tests were designed together with the structure, it is possible that additional tests would become cost effective because they would allow reduced structural weight.

## 4. Errors in Stress, Load, Geometry and Material Allowable

### *4.1. Errors in design*

We assume that different aircraft companies like Airbus, Boeing, Bombardier, Embraer, etc. essentially design the same airplane. Before performing a stress analyses, we first assume that these companies perform aerodynamic analyses to determine the loads acting on aircraft. However, the loads calculated by an aircraft company, $P_{calc}$, differ from the loads corresponding to FAA design specifications, $P_d$. The error made in load calculation, $e_p$, is different from one company to another. Throughout all error factor definitions we consistently formulate the expressions such that positive error factor implies a conservative decision. Based on this, $P_{calc}$ is expressed in terms of $P_d$ as:

$$P_{calc} = (1 + e_P) P_d \tag{1}$$

Besides the error in load calculation, an aircraft company has also errors in stress calculation. Considering a small part of the aircraft structure, we can represent it as an unstiffened panel such that the value of stress calculated by stress analysis team, $\sigma_{calc}$, is expressed in terms of the load values calculated by the aerodynamics team, $P_{calc}$, the design width, $w_{design}$, and thickness, $t$. Hence, introducing the term $e_\sigma$ representing error in the stress analysis we can write

$$\sigma_{calc} = (1 + e_\sigma) \frac{P_{calc}}{w_{design}\, t} \tag{2}$$

Equation (2) is used by the designer to calculate the design thickness $t_{design}$ required to carry the calculated design load times the safety factor $S_F$. That is

$$t_{design} = \left(1 + e_\sigma\right)\left(1 + e_P\right) \frac{S_F P_d}{w_{design}\, \sigma_a} \tag{3}$$

where $\sigma_a$ is the value of allowable stress used in the design. This allowable stress is based on A-basis properties (see Appendix 1) for the design material.

### *4.2. Errors in implementation (difference between design value and fleet average)*

The error factors $e_\sigma$ and $e_\sigma$ represent the errors made in the design stage. In addition, there will be some implementation errors in the geometric and material parameters. These implementation errors represent the difference between the values of these parameters in an average airplane (fleet-average) built an aircraft company and the design values of these parameters. Since we represent a small part of the aircraft structure as an unstiffened panel, the geometry parameters are the width and the thickness of the panel. Errors in panel width, $e_w$, represent the deviation of the values of panel width designed by an individual aircraft company, $w_{design}$, from the average value of panel width and thickness of panels built by the company, $w_{built}$. Thus we have

$$w_{built} = (1 + e_w)\, w_{design} \tag{4}$$

Similarly the built thickness value will differ from the design value such that

$$t_{built} = (1 + e_t)\, t_{design} \tag{5}$$

In addition average built material parameters and the design material parameters will be different from each other. In particular, the failure stresses $\sigma_f$ are related as

$$\sigma_{f\,built} = (1 - e_m)\, \sigma_{f\,design} \tag{6}$$

The relationship between the allowable and failure stresses is that the allowable stress is the A-basis value of failure stress. The detailed explanation on the computation of A-basis value is given in Appendix 1. The formulation of Eq.(6) is different from Eqs. (1, 2, 4 and 5) in that the sign in front of the error factor $e_m$ is negative. The reason is that we consistently formulate the expressions such that positive error factor implies a conservative decision.

*4.3. Fleet average safety factor*

The fleet average of stress experienced by a panel under the correct design loads is

$$\sigma_{d-avg} = \frac{P_d}{t_{built}\, w_{built}} \tag{7}$$

Substituting from Eqs.(3-5) into Eq. (7) we have

$$\sigma_{d-avg} = \frac{1}{(1 + e_\sigma)(1 + e_P)(1 + e_w)(1 + e_t)} \frac{\sigma_a}{S_F} \tag{8}$$

Then, we can define a fleet average safety factor

$$S_{F\,avg} = \frac{\sigma_{f\,built}}{\sigma_{d-avg}} \tag{9}$$

Combining Eqs. (6) and (9) yields

$$S_{F\,avg} = S_F \frac{\sigma_{f\,design}}{\sigma_a} \left(1 + e_{total}\right) \tag{10}$$

where

$$e_{total} = \left[(1 + e_\sigma)(1 + e_P)(1 + e_w)(1 + e_t)\ (1 - e_m)\right] - 1 \tag{11}$$

Here $e_{total}$ represents the cumulative effect of the various errors on the safety factor for the average airplane (fleet average) built by a company. Equation (10) shows that when there are no errors, the average safety factor is larger than $S_F$ due to conservative allowable stress (A-Basis properties). The error factors are random variables represented by distribution type, their average values and their bounds as given in Table 2. In addition there is variability in the material and

geometric properties and the load experienced in actual flight between individual aircraft in the fleet. This will be discussed next.

**Table 2.** Distribution of error factors and their bounds

| Error factors | Distribution Type | Mean | Bounds |
|---|---|---|---|
| Error in stress calculation, $e_\sigma$ | Uniform | 0.0 | ± 5% |
| Error in load calculation, $e_P$ | Uniform | 0.0 | ± 10% |
| Error in width, $e_w$ | Uniform | 0.0 | ± 1% |
| Error in thickness, $e_t$ | Uniform | 0.0 | ± 2% |
| Error in material allowable, $e_m$ | Uniform | 0.0 | ± 20% |

Table 2 presents nominal values for the error factors. In the Results section of the paper we will vary these error bounds and investigate the effects of these changes on the probability of failure. As seen in Table 2, the error having the largest bound in its distribution is the error in material failure stress, because it includes also the likelihood of unexpected failure modes.

*4.4. Variability*

In the previous sections, we analyzed the different types of errors made in the design and implementation stages representing the differences between the fleet average values of geometry, material and loading parameters and their corresponding design values. These parameters, however, vary from one aircraft to another in the fleet. For instance, we assume that the actual value of thickness of a panel in an aircraft is defined by the fleet average thickness value by

$$t_{act} = U\left(t_{built}; 3\%\, bounds\right) \tag{12}$$

Here '*U*' indicates that the distribution is uniform, '$t_{built}$' is the average value of thickness (fleet average) and '3% bounds' defines that the lower bound for thickness value is the average value minus 3% of the average and the upper bound for thickness value is the average value plus 3% of the average. Note that the thickness error in Table 2 is uniformly distributed with bounds of ± 2%. Thus the difference between all thicknesses over the fleets of all companies is up to ± 5%. However, the combination of error and variability is not a uniform distribution. Table 3 presents the assumed distributions for variabilities.

**Table 3.** Distribution of random variables having variability

| Variables | Distribution type | Mean | Scatter |
|---|---|---|---|
| Actual service load, $P_{act}$ | Lognormal | $P_d = 100$ | (10%) c.o.v.* |
| Actual panel width, $w_{act}$ | Uniform | $w_{built} = 1$ | (1%) bounds |
| Actual panel thickness, $t_{act}$ | Uniform | $t_{built}$ | (3%) bounds |
| Actual failure stress, $\sigma_{f\,act}$ | Lognormal | $\sigma_{f\,built} = 150$ | (10%) c.o.v.* |

* c.o.v. = coefficient of variation

## 5. Certification Tests

We simulate the effect of safety measures mentioned in Section 3 by assuming the statistical distribution of the uncertainties and incorporating them in approximate probability calculations and a two-level Monte Carlo simulation (see Figure 1), with different aircraft models being considered at the upper level, and different instances of the same aircraft at the lower level. To simulate the epistemic uncertainty, we assume that we have a large number of nominally identical aircraft being designed (e.g. by Airbus, Boeing, Bombardier, Embraer etc.), with the errors being fixed for each aircraft. We consider a simple example of an unstiffened panel designed for strength under uniaxial tensile loads. The Monte Carlo simulation works as follows.

After the structural component has been designed with random errors in stress, load, width, allowable stress and thickness (step A in Fig. 1), we simulate certification testing for the aircraft (step B in Fig. 1). Here we assume that the component will not be built with complete fidelity to the design due to variability in geometric properties. That is, the actual values of these parameters $w_{act}$ and $t_{act}$ and will be different from the fleet-average values $w_{built}$ and $t_{built}$ due to variability. The panel is then loaded with the design axial force of $S_F$ times $P_{calc}$, and the stress in the panel is recorded (step C in Fig. 1). If this stress exceeds the failure stress $\sigma_{f\,act}$ (itself a random variable with an average value $\sigma_{f\,built}$, see Table 3.) then the design is rejected, otherwise it is certified for use. That is, the airplane is certified if the following inequality is satisfied and we can build multiple copies of the airplane.

$$\sigma - \sigma_f = \frac{S_F P_{calc}}{w_{act}\, t_{act}} - \sigma_{f\,act} \le 0 \tag{13}$$

or

$$S_F (1 + e_P) P_d \le R_{act} = w_{act}\, t_{act}\, \sigma_{f\,act} \tag{14}$$

where the left side denotes the applied load, and the right side the load bearing capacity or "resistance" $R_{act}$. As noted earlier the terms $w_{act}$, $t_{act}$ and $\sigma_{f\,act}$ in Eq. (14) reflect the variability in geometric and material properties (see Table 3). The distribution types and the distribution parameters of the random variables used in design and certification are listed in Table 3.

*Effect of Certification Tests on Distributions of $e_{total}$ and $S_{F\ avg}$*

The fleet-average safety factor (see Eq. (10)) is defined in terms of safety factor, failure stress ratio and total error property. Amongst those terms only the error term is subject to change due to certification testing. One can argue that the way certification tests reduce the probability of failure is by changing the distribution of the error factor $e_{total}$. Without certification testing, we assume uniform distributions for the error factors in stress, load, width and thickness. However, designs based on unconservative models are more likely to fail certification, and so the distribution of $e_{total}$ becomes conservative for structures that pass certification. In order to quantify this effect, we calculated the updated distribution of the error factor $e_{total}$. The updated distribution is calculated by Monte Carlo simulations.

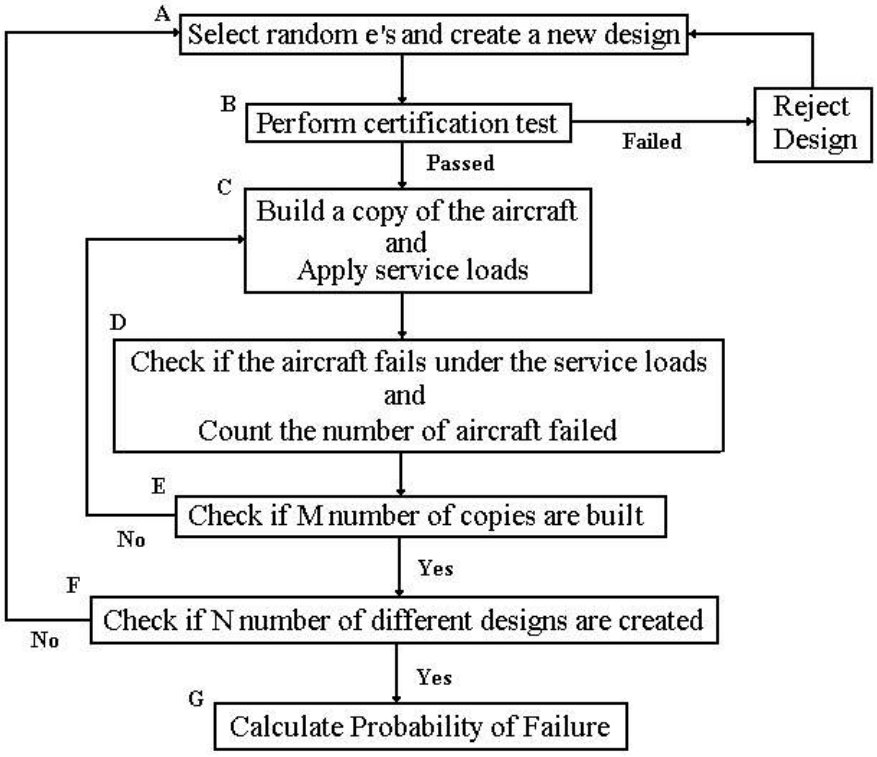



**Figure 1.** Flowchart for Monte Carlo simulation of component design and failure

As noted earlier, in our previous paper (Kale et al, 2004) we represented the overall error with a single parameter, hereinafter the "Single Error Factor Case (SEF case)", and used uniform distribution for the initial distribution of this error. However, in the present work we use a more complex representation of error with individual error factors, hereinafter the "Multiple Error Factor Case (MEF case)", and we represent initial distribution of each error factor with uniform distribution. For the SEF Case we obtained updated the distribution of error term using Bayesian updating. However, since we use a more complex model to represent error in this study, updating by analytical means is quite difficult. In addition, we prefer to update average-safety factor, $S_{F\ avg}$. Revisiting the expression for average safety factor (see Eq. (10)), we see that only random variables are $\sigma_a$ and $e_{total}$. Since the variability in $\sigma_a$ is very small compared to $e_{total}$, the distribution of

$S_{F\,avg}$ and $e_{total}$ are nearly the same. We calculated the updated distribution of average safety factor thru Monte Carlo simulations of sample size 10,000. Initial and updated distribution of $S_{F\,avg}$ are shown in Fig.2.

Figure 2 shows us how certification tests update the distribution of average safety factor for SEF and MEF cases. For SEF case the uniform initial distribution is updated such that the likelihood of higher values of average safety factor is increased. That is the components built with low safety factor are rejected in certification tests. Similarly for the MEF case, the initial distribution of average safety factor is shifted to up and right indicating that the components with high safety factors are favored via certification testing.

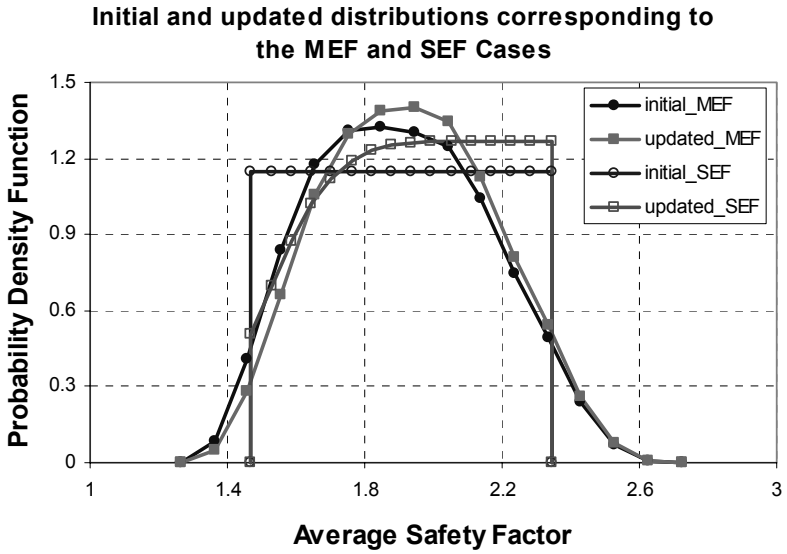



**Figure 2.** Comparing initial and updated distribution of $S_{Favg}$ between SEF and MEF cases. The single error is chosen as to match the standard deviation of the safety factor MEF.

## 6. Probability of Failure Calculation

After the component passes the certification test, we subject the component in each airplane to actual random maximum (over a lifetime) service loads (step D in Fig. 1) and decide whether it fails using Eq. (15).

$$P_{act} \geq R = t_{act}\, w_{act}\, \sigma_{f\,act} \tag{15}$$

Here, $P_{act}$ is the actual load acting under service, and $R$ is the resistance or load capacity of the structure in terms of the random width $w_{act}$, thickness $t_{act}$ and failure stress, $\sigma_{f\,act}$.

This procedure of design and testing is repeated (steps A-B in Fig.1) for $N$ different aircraft models. Here N different design is the representative of different designs of different aircraft

companies (outer loop of Monte Carlo simulation). For each new model, different random error factors $e_\sigma$, $e_P$, $e_w$ $e_t$ and $e_m$ are picked for the design representing the different error factors for the different aircraft companies.

The inner loop in Figure 1 (steps C-E in Fig.1) represents the simulation of a population of $M$ airplanes (hence structural components) that all have the same design. However, each component is different due to variability in geometry, failure stress, and loading (step D). We subject the component in each airplane to actual random maximum (over a lifetime) service loads (step E) and calculate whether it fails using Eq. (15).

We count the number of panels failed for each airplane, and add up all the failures. The failure probability is calculated by dividing the number of failures by the number of airplane models that passed certification, times the number of copies of each model.

## 7. Results

We first investigate the effect of error bounds on the probability of failure of panels. Since we have 5 different contribution to total error in the analysis, we scale all error components with a single multiplier, $k$,

$$e_{total} = \left[(1+ke_\sigma)\,(1+ke_P)\,(1+ke_w)\,(1+ke_t)\,(1-ke_m)\right]-1 \tag{16}$$

and explore the effect of $k$ on failure probability. We calculated the average value and coefficient of variation of probability of failure values for the panels designed with A-basis properties and safety factor of 1.5.

**Table 4.** Average and coefficient of variation (over N=500 companies) of probability of failure for the components designed with A-basis properties and $S_F$=1.5. Monte Carlo simulations with N=500, M=20,000.

| $k$ | $\overline{P}_{nt}$ and c.o.v.($P_{nt}^*$) | $\overline{P}_t$ and c.o.v.($P_t^*$) | $\overline{P}_{nt} - \overline{P}_t$ | $\overline{P}_t / \overline{P}_{nt}$ |
|---|---|---|---|---|
| 0.5 | 1.510 x10$^{-5}$ (330 %) | 1.483 x10$^{-5}$ (336 %) | 2.740 x10$^{-7}$ | 0.982 |
| 0.75 | 9.000 x10$^{-5}$ (289 %) | 8.053 x10$^{-5}$ (313 %) | 9.474 x10$^{-6}$ | 0.895 |
| 1 | 3.626 x10$^{-4}$ (448 %) | 2.464 x10$^{-4}$ (596 %) | 1.163 x10$^{-4}$ | 0.679 |
| 1.5 | 5.275 x10$^{-3}$ (366 %) | 1.294 x10$^{-3}$ (496 %) | 3.981 x10$^{-3}$ | 0.245 |
| 2 | 3.106 x10$^{-2}$ (318 %) | 1.905 x10$^{-3}$ (694 %) | 2.916 x10$^{-2}$ | 0.061 |

* $P_{nt}$ and $P_t$ are the probability of failure without and with certification testing, respectively.

Table 4 shows that as the error in analyses increases, i.e. $k$ increases, the average values probability of failures (both with and without certification) of the components are also increases. The coefficient of variation of failure probability is very large. With N=500, the coefficient of variation of the average between repeated Monte Carlo simulations should be reduced by $\sqrt{500}$=22. This would still indicates variations of up to 30% in the values in Table 4.

The last two columns of Table 4 present the effect of certification tests on failure probabilities. For this purpose we used two measures; the difference of failure probabilities and the ratio of failure probabilities. In our previous work, we have shown that the difference may be more meaningful when the probability of failure is high since it indicates the amount of aircraft that is saved by the use of certification tests. As we can see from the 4$^{th}$ column when the error increases, the difference between the two failure probabilities also increases pointing out that the certification

tests become more effective. The results given in the last column demonstrate that the trend in the probability ratio is also similar to the previous trend; when the error increases the ratio of the two probabilities decreases showing the increase in the effectiveness of the certification tests.

As noted earlier, in our previous paper (Kale et al, 2004) we represented the overall error with a single error factor (SEF), and used uniform distribution for the initial distribution of this error. However, in this work we use a more complex representation of error with multiple error factors (MEF), we represent initial distribution of each error factor with uniform distribution. In this case, the distribution of total error is no more uniform (see Figure 2). In order to compare the two approaches, we first calculate the mean and standard deviation of the initial total error factor, $e_{total}^{ini}$. The mean value of the total error factor is close to zero so that we use zero mean value and equal standard deviation value for a uniform distribution of total error. The equivalent error bound s for uniform distribution corresponding to different error multiplier *k* is listed of Table 5.

**Table 5.** Equivalent error bounds for the SEF case corresponding to the same standard deviation in the MEF case

| *k* | Average $e_{total}^{ini}$ | Standard Deviation of $e_{total}^{ini}$ | from the SEF Case → to the MEF Case | Average $e_{total}^{ini}$ | Bound of error for $e_{total}^{ini}$ |
|---|---|---|---|---|---|
| 0.5 | $1.00 \times 10^{-3}$ | 0.064 | | 0 | 11.1 |
| 0.75 | $1.42 \times 10^{-3}$ | 0.100 | | 0 | 17.3 |
| 1 | $1.12 \times 10^{-3}$ | 0.132 | | 0 | 22.9 |
| 1.5 | $2.97 \times 10^{-3}$ | 0.200 | | 0 | 34.6 |
| 2 | $1.07 \times 10^{-2}$ | 0.271 | | 0 | 46.9 |

Using the equivalent error bounds of SEF Case given in the right portion of Table 5 we calculate the average values of probabilities of failure without and after certification test for SEF case and we compare them in Table 6 with corresponding failure probabilities of MEF case from Table 4. In addition, the comparison of the probability of failures for the two cases is presented in Fig. 3.

**Table 6.** Comparison of failure probabilities for SEF and MEF case

| *k* | $\overline{P}_{nt}^{MEF}$ ( $\times 10^{-4}$) | $\overline{P}_{t}^{MEF}$ ( $\times 10^{-4}$) | $P_f$ Ratio* | $\overline{P}_{nt}^{SEF}$ ( $\times 10^{-4}$) | $\overline{P}_{t}^{SEF}$ ( $\times 10^{-4}$) | $P_f$ Ratio* |
|---|---|---|---|---|---|---|
| 0.5 | 0.151 | 0.148 | 0.982 | 0.147 | 0.139 | 0.948 |
| 0.75 | 0.900 | 0.805 | 0.895 | 0.620 | 0.525 | 0.848 |
| 1 | 3.626 | 2.464 | 0.679 | 2.579 | 1.592 | 0.617 |
| 1.5 | 52.57 | 12.94 | 0.245 | 37.06 | 5.671 | 0.153 |
| 2 | 310.6 | 19.05 | 0.061 | 314.6 | 5.733 | 0.018 |

* $P_f$ Ratio is the ratio of failure probabilities; $\overline{P}_t / \overline{P}_{nt}$

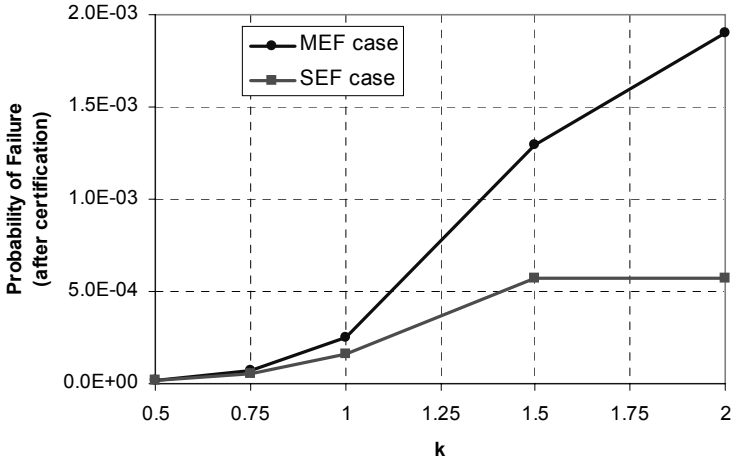



**Figure 3.** After certification failure probabilities for SEF and MEF case

When we compare the probability of failure without certification the results are similar for both the MEF Case and SEF Case (see columns 2 and 5, Table 6). Note that the differences between the corresponding columns are of the same order as the scatter in the Monte Carlo simulations. Comparing the failure probabilities after certification, we notice that the MEF Case leads to higher probability of failure values hence $\overline{P}_t / \overline{P}_{nt}$ ratios. That is the additional detail of the MEF reduces the effectiveness of the certification testing. This is due to the fact that in the SEF case (Kale et al, 2004) the certification testing is performed with the average value of actual load, $P_d$ (see Table 3 for the definition of $P_d$). However, in the MEF case certification testing is performed with the calculated load, $P_{calc}$ (see Eq. (1) for the expression for $P_{calc}$). Therefore, one component of the error can not be exposed by certification testing. This effect is also apparent when we compare the average safety factor values for these two cases in Table 7 and in Fig. 4.

**Table 7.** Comparison of Average Safety Factor for two cases

| $k$ | $\left(\overline{S}_{F_{avg}}\right)_{nt}^{MEF}$ | $\left(\overline{S}_{F_{avg}}\right)_{t}^{MEF}$ | $S_F$ Ratio* | $\left(\overline{S}_{F_{avg}}\right)_{nt}^{SEF}$ | $\left(\overline{S}_{F_{avg}}\right)_{t}^{SEF}$ | $S_F$ Ratio* |
|---|---|---|---|---|---|---|
| 0.5 | 1.909 | 1.911 | 1.001 | 1.907 | 1.910 | 1.002 |
| 0.75 | 1.910 | 1.920 | 1.005 | 1.907 | 1.920 | 1.007 |
| 1 | 1.909 | 1.938 | 1.015 | 1.907 | 1.942 | 1.018 |
| 1.5 | 1.912 | 2.015 | 1.054 | 1.907 | 2.031 | 1.065 |
| 2 | 1.907 | 2.093 | 1.097 | 1.907 | 2.149 | 1.127 |

* $S_F$ Ratio is the ratio of average safety factors without and with certification

Comparing the average safety factors, $S_{F\,avg}$, after certification corresponding to the MEF and SEF Cases (columns 3 and 6, Table 7), we see that average safety factor values corresponding to SEF Case is larger which will in turn lead to smaller probability of failure (see Table 6).

Looking at the columns 4 and 7 we see and expected trend in the values of $S_F$ Ratios. Both the ratios corresponding to the MEF Case and SEF Case increases with the increase of error bounds, rendering certification tests more effective.

The main reason for lower safety in the MEF case is the reduced effect of certification on design thickness as seen in Table 8.

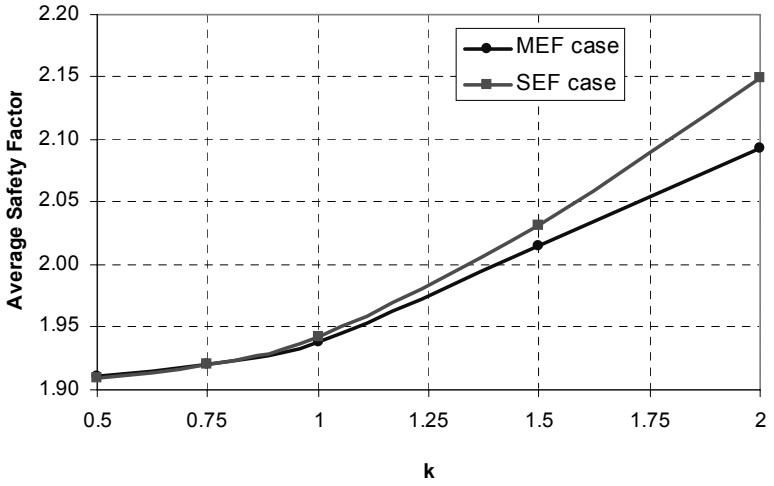



**Figure 4.** After certification failure probabilities for SEF and MEF case

**Table 8.** Comparison of design thicknesses for two cases

| $k$ | $\left(\bar{t}_{design}\right)_t^{MEF}$ | $\left(\bar{t}_{design}\right)_t^{SEF}$ | $t_{design}$ Ratio* |
|---|---|---|---|
| 0.5 | 1.273 | 1.274 | 1.001 |
| 0.75 | 1.274 | 1.280 | 1.005 |
| 1 | 1.276 | 1.295 | 1.015 |
| 1.5 | 1.282 | 1.354 | 1.056 |
| 2 | 1.282 | 1.434 | 1.118 |

* $t_{design}$ Ratio is the ratio of average design thickness for the MEF and SEF Cases

Table 8 illustrates the effect of error multiplier $k$ on the average design thicknesses after certification of the components corresponding to the MEF and SEF Cases. When we compare average design thicknesses, we see that components corresponding to SEF Case are designed thicker compared to MEF Case leading to low probability of failure values.

Finally, we change the variability in failure stress and investigate the effect of this change in probability of failure. The results are presented in Table 9.

**Table 9.** Comparison of Failure Probabilities for the MEF Case corresponding to different variability in failure stress $\sigma_f$

| c.o.v. ($\sigma_f$) | $\left(\overline{S}_{F_{avg}}\right)_{nt}$ | $\left(\overline{S}_{F_{avg}}\right)_{t}$ | Average $\left(t_{design}\right)_{nt}$ | Average $\left(t_{design}\right)_{t}$ | $\overline{P}_{nt}$ ( x$10^{-4}$) | $\overline{P}_{t}$ ( x$10^{-4}$) | $P_f$ Ratio |
|---|---|---|---|---|---|---|---|
| 0 % | 1.503 | 1.650 | 1.001 | 1.007 | 79.19 | 0.306 | 0.004 |
| 5 % | 1.691 | 1.761 | 1.127 | 1.131 | 10.50 | 1.297 | 0.124 |
| 10 % | 1.909 | 1.938 | 1.275 | 1.276 | 3.626 | 2.464 | 0.679 |
| 15 % | 2.152 | 2.166 | 1.434 | 1.435 | 3.624 | 3.231 | 0.892 |
| 20 % | 2.450 | 2.458 | 1.628 | 1.628 | 3.576 | 3.370 | 0.943 |

Table 9 displays the effect of variability in failure stress, $\sigma_f$, on the average safety factor and probability of failure for the MEF Case. We observe that the average safety factor and design thickness increases with the increase of variability in failure stress. On the other hand, probability of failure increases with the increase of variability. Comparing the design thicknesses with and without certification cases and also from $P_f$ ratio given in the last column of Table 10 we observe that certification tests become less effective as variability increases. Figure 5 also shows the diminishing of the efficiency of testing as variability grows.

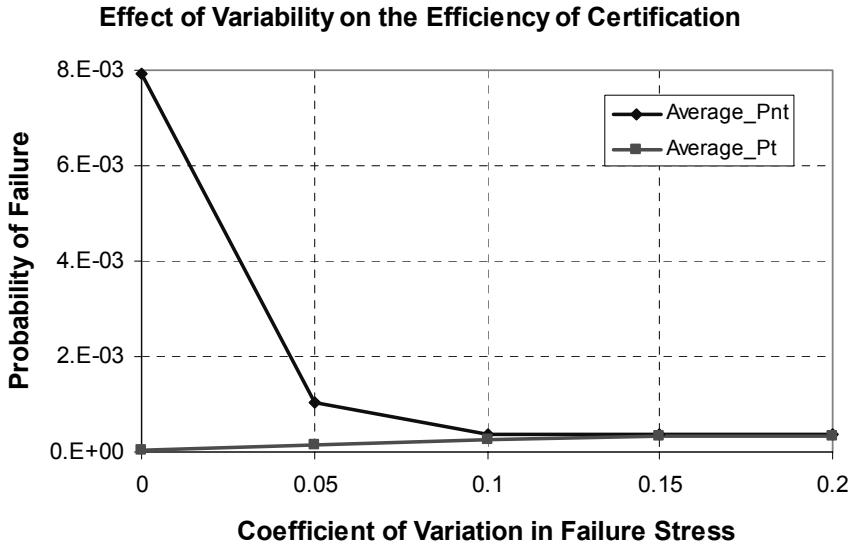



**Figure 5.** Effect of variability in failure stress on MEF case

## 8. Concluding Remarks

The interaction between error, variability and testing on the probability of failure of aircraft structures was analyzed. We used stress failure due to extreme loads, which can be simulated by an unstiffened panel designed under uniaxial loads. Monte Carlo simulations were performed to account for both fleet-level uncertainties (such as errors in analytical models) and individual uncertainties (such as variability in material properties).

In our previous paper (Kale et al, 2004) we sought to clarify the interaction between error, variability and testing by the use of a simple model of error, lumping it into a single error component in the calculation of stresses.. In this paper we used a more realistic error model such that errors in load and stress calculation, and also errors in material and geometric properties were modeled using uniform distributions for their initial distributions and compared the results with our previous paper's results. The same as in our previous paper, the variability in the material and geometric properties and in the loading was included in the analysis by modeling the variabilities with random numbers and their distributions.

In our previous paper, we had found that the effect of tests is most important when errors in analytical models are high and when the variability between airplanes is low. These observations also apply to the results obtained in this paper. We expressed the effectiveness of the certification tests is expressed by the ratio of the probability of failure with the test, $P_t$, to the probability of failure without tests, $P_{nt}$. Using this ratio we have shown that the effectiveness of certification tests increases when the error in the analysis is large. We changed the bound of error in material properties, $e_m$, in which we included the likelihood of occurrence of unexpected failure modes and the difference in the behavior of material in coupon tests and in the actual service, and have shown that the reduction in bounds in $e_m$ is also an indication of safer designs. It was an expected result since the safer the design, the lesser the need for testing. In addition, we played with the variability of failure stress and have shown that the increase of variability increased the probability of failure and made certification tests less effective.

Another observation from study is that this new more realistic error model led to an increase in average safety factor (fleet-average) thereby an increase in the probability of failure. In addition, the certification testing for this new case, we called as MEF Case, found to be less effective since we used the calculated load values in testing of components instead of using actual loads as we did in our previous paper.

The effect of building-block type tests that are conducted before certification was not assessed here. However, these tests reduce the errors in the analytical models, and on that basis we determined that they can reduce the probability of failure by one or two orders of magnitude.

## Acknowledgement

This work was supported in part by NASA University Research, Engineering and Technology Institute (URETI) and NASA Langley Research Center.

## References

Ang, A., H-S. and Tang, W.H., "Probability Concepts in Engineering Planning and Design, Volume I : Basic Principles," John Wiley & Sons, 1975.

Kale, A., Acar, E. Haftka, R.T., and Stroud, W.J., "Why Airplanes are so Safe Structurally? Effect of Various Safety Measures on Structural Safety of Aircraft," 45th Structures, Structural Dynamics and Materials Conference, 19-22 April 2004, Palm Springs California.

Oberkampf, W.L., Deland, S.M., Rutherford, B.M., Diegert, K.V., and Alvin, K.F., Estimation of Total Uncertainty in Modeling and Simulation, Sandia Report SAND2000-0824, 2000.

## Appendix 1

**A-Basis property –** A-basis value is the value exceeded by 99% of the population with 95% confidence. This is given by

$$\text{A-basis} = \mu - \sigma \times k_1 \tag{A1}$$

where $\mu$ is the mean, $\sigma$ is the standard deviation and $k_1$ is the tolerance coefficient for normal distribution given by Equation A2

$$k_1 = \frac{z_{1-p} + \sqrt{z_{1-p}^2 - ab}}{a}$$
$$a = 1 - \frac{z_{1-\gamma}^2}{2(N-1)}\ ;\ b = z_{1-p}^2 - \frac{z_{1-\gamma}^2}{N} \tag{A2}$$

where $N$ is the sample size and $z_{1-p}$ is the critical value of normal distribution that is exceeded with a probability of $1- p$. The tolerance coefficient $k_1$ for a lognormal distribution is obtained by first transforming the lognormally distributed variable to a normally distributed variable. Equation A1 and A2 can be used to obtain an intermediate value. This value is then converted back to the lognormally distributed variable using inverse transformation.

In order to obtain the A-basis values, 15 panels are randomly selected from a batch. Here, the uncertainty in material property is due to allowable stress. The mean and standard deviation of 15 random values of allowable stress is calculated and used in determining the A-basis value of allowable stress.

# An Efficient Unified Approach for Reliability and Robustness in Engineering Design

Zissimos P. Mourelatos
*Mechanical Engineering Department Oakland University, Rochester, MI 48309*
*e-mail: mourelat@oakland.edu*

Jinghong Liang
*Mechanical Engineering Department Oakland University, Rochester, MI 48309*

**Abstract.** Mathematical optimization plays an important role in engineering design, leading to greatly improved performance. Deterministic optimization however, can lead to undesired choices because it neglects input and model uncertainty. Reliability-based design optimization (RBDO) and robust design improve optimization by considering uncertainty. A design is called reliable if it meets all performance targets in the presence of variation/uncertainty and robust if it is insensitive to variation/uncertainty. Ultimately, a design should be optimal, reliable, and robust. Usually, some of the deterministic optimality is traded-off in order for the design to be reliable and/or robust. This paper describes the state-of-the-art in assessing reliability and robustness in engineering design and proposes a new unifying formulation. The principles of deterministic optimality, reliability and robustness are first defined. Subsequently, the design compromises for simultaneously achieving optimality, reliability and robustness are illustrated. Emphasis is given to a unifying probabilistic optimization formulation for both reliability-based and robust design, including variation of all performance measures. The robust engineering problem is investigated as a part of a "generalized" RBDO problem. Because conventional RBDO optimizes the mean performance, its objective is only a function of deterministic design variables and the means of the random design variables. The conventional RBDO formulation is expanded to include performance variation as a design criterion. This results in a multi-objective optimization problem even with a single performance criterion. A preference aggregation method is used to compute the entire Pareto frontier efficiently. Examples illustrate the concepts and demonstrate their applicability.

## 1. Introduction

Deterministic mathematical optimization has led to greatly improved performance in all areas of engineering design. It can however, lead to undesired choices, if uncertainty/variation is ignored. In deterministic design we assume that there is no uncertainty in the design variables and/or modeling parameters. Therefore, there is no variability in the simulation outputs. However, there exists inherent input and parameter variation that results in output variation. Deterministic optimization typically yields optimal designs that are pushed to the limits of design constraint boundaries, leaving little or no room for tolerances (uncertainty) in manufacturing imperfections, modeling and design variables. Therefore, deterministic optimal designs that are obtained without taking into account uncertainty are usually unreliable. Input variation is fully accounted for in Reliability-Based Design Optimization (RBDO) and robust design.

In RBDO, probability distributions describe the stochastic nature of the design variables and model parameters. Variations are represented by standard deviations (typically assumed to be constant) and a *mean* performance measure is optimized subject to *probabilistic* constraints. RBDO can be a powerful tool which can assist in design under uncertainty, since it provides

optimum designs in the presence of uncertainty in design variables/parameters and simulation models. For this reason, it has been extensively studied [1-8].

Robust designs methods are also widely used because they can improve the quality of products and processes [9]. Robust design minimizes performance variation without eliminating the sources of variation [10]. The product quality is commonly defined using a quality loss function [10,11]. Various methods have been proposed for estimating the product quality loss [12-16] using the mean and standard deviation of the response (performance measure). A review of existing robust optimization methods can be found in [17,18].

It is common in product design to have multiple performance measures. A robust design therefore, must simultaneously minimize the variation of all performance measures using a multi-objective optimization approach. It is however, common to use a single-objective robust design formulation by either minimizing a heuristic quality loss function [18,19] or form a single objective utilizing weighting factors in a weighted-sum approach [20]. We will show in this work that the weighted-sum approach may lead to inaccurate results. A detailed examination of the weighted-sum approach drawbacks is provided in [21]. There are only a few multi-objective approaches to robust design [17,22-24].

Reliability and robustness are attributes of design under uncertainty. It makes sense therefore, to combine them in a *unified,* multi-objective approach where the mean and variation of multiple performance measures are simultaneously minimized, subject to probabilistic constraints for design feasibility. Such an approach is proposed in this paper. The concept of a unified methodology for reliability and robustness is not new, as references [17,18,25] for example, indicate. However, major simplifications are usually made. In general, researchers use one or both of 1) the weighted-sum simplification for the general multi-objective problem and 2) a simplified representation of the probabilistic design feasibility using the worst-case scenario [23,17] or the moment matching formulation [26,16]. Furthermore, a first-order Taylor expansion is usually performed for estimating the performance variance. This linearization approach does a fairly good job in estimating the expected value of the nonlinear objective function. However, it can be quite inaccurate in estimating its higher moments as is the standard deviation [27]. Moreover, it is limiting in that it does not provide us with the correct probability distribution information of the objective function.

In this paper, a computationally efficient unified approach to reliability and robustness is proposed which alleviates the described shortcomings of the available methods. A preference aggregation method [28-30] is used to choose the "best" solution of a multi-objective optimization problem based on designer preferences. The performance variation is assessed by a percentile difference method originally proposed in [25]. The percentiles are efficiently calculated using a variation of the Advanced Mean Value method [31]. The "best" design is calculated using an efficient single-loop probabilistic optimization method [32]. Examples illustrate the methodology.

## 2. Definition of Optimality, Reliability and Robustness

In deterministic design optimization an objective function is usually minimized subject to certain constraints which define a feasible region. A conventional deterministic optimization problem with inequality constraints only, is stated as

$$\begin{aligned} &\min_{\mathbf{d}} f(\mathbf{d}) \\ \text{s.t.}\quad &G_i(\mathbf{d}) \geq 0\,, \quad i = 1,...,n \\ &\mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U \end{aligned} \tag{1}$$

where $\mathbf{d} \in R^k$ is the vector of deterministic design variables. A bold letter indicates a vector.

In optimization under uncertainty the task is to minimize (or maximize) an objective while 1) all constraints are satisfied and 2) the performance of the design is insensitive to the existing variation or uncertainty. *Variation* or *stochastic uncertainty* is defined as that "irreducible" uncertainty which, being inherent in the physical system, ought not to depend on the amount of available statistical data. It is usually modeled probabilistically. In this work, a design is called *reliable* if it meets all performance targets in the presence of variation/uncertainty and *robust* if it is insensitive to variation/uncertainty. Ultimately, *a design should be optimal, reliable, and robust*. Usually, some of the deterministic optimality is traded-off in order for the design to be reliable and/or robust.

A typical RBDO problem is formulated as

$$\begin{aligned} &\min_{\mathbf{d},\boldsymbol{\mu}_\mathbf{X}} f(\mathbf{d},\boldsymbol{\mu}_\mathbf{X},\boldsymbol{\mu}_\mathbf{P}) \\ \text{s.t.}\quad &P(G_i(\mathbf{d},\mathbf{X},\mathbf{P})\geq 0)\geq R_i = 1-p_{f_i}, \quad i=1,...,n \\ &\mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U \\ &\boldsymbol{\mu}_\mathbf{X}^L \leq \boldsymbol{\mu}_\mathbf{X} \leq \boldsymbol{\mu}_\mathbf{X}^U \end{aligned} \tag{2}$$

where $\mathbf{X}\in R^m$ is the vector of random design variables and $\mathbf{P}\in R^q$ is the vector of random design parameters. According to the used notation, an upper case letter indicates a random variable or a random parameter and a lower case letter indicates a realization of a random variable or parameter. If the target probability of failure $p_f$ is approximated using the target reliability index $\beta_t$ and the standard normal cumulative distribution function $\Phi$, the actual reliability level for the $i^{th}$ deterministic constraint $G_i(\mathbf{d},\mathbf{X},\mathbf{P})\geq 0$ is $R_i = 1-p_{f_i}$ where

$$p_{f_i} = P(G_i(\mathbf{d},\ \mathbf{X},\ \mathbf{P}) \leq 0) = F_{G_i}(0) \leq \Phi(-\beta_{t_i}) \tag{3}$$

and $F_{G_i}(\ )$ is the cumulative distribution function of $G_i$.

The principles of deterministic optimality and reliability are demonstrated graphically in Fig. 1 using a hypothetical design. The design compromises for achieving optimality and reliability are illustrated. For a hypothetical design with two constraints in two dimensions, the deterministic optimum is denoted by point **A** in Fig. 1. It is the constrained optimum, where the objective is minimized and both constraints are active. If the two design variables are random with their means specified by the deterministic optimum, all possible design realizations fall within a closed domain (indicated for simplicity by a circle centered at point **A**) due to the variation of the two design variables. In this case, a large percentage of design realizations violate at least one constraint or performance target, rendering design **A** unreliable. To achieve reliability, the circle around point **A** must be moved inside the feasible domain with its center at point **B**. As the circle moves within the feasible domain, the design simultaneously becomes more reliable and less optimal. The circle must be moved to accommodate the uncertainty indicated by its radius. At the reliable design **B**, the circle may be tangent to a number of performance targets which become "probabilistically" active. The process of moving the circle from the deterministic optimum to the reliability optimum is known as Reliability-Based Design Optimization. It can be implemented mathematically by solving Problem (2).

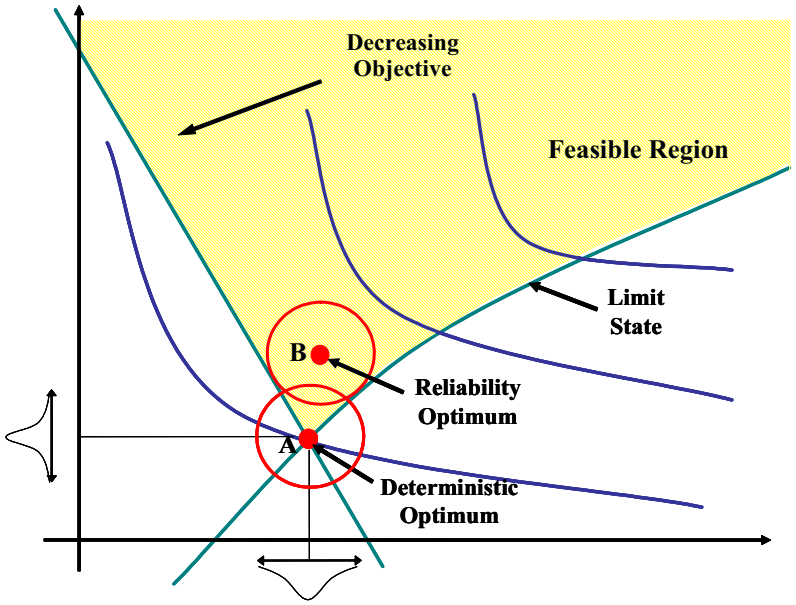



Figure 1. Geometric interpretation of deterministic and reliable designs

At point **B** of Fig. 1, the design is at its reliability optimum where the objective function is optimized given that the circle around **B** is within the bounds of the constraints. However, it may not be robust. It is robust if the performance of each design realization within the circle is as close to constant as possible, indicating insensitivity to variation. Robustness can therefore, be achieved by placing the final optimum at a region, where the response is "flat" or insensitive to the design variables. This is illustrated in Fig. 2 for a hypothetical one-dimensional design. Assuming the variation/uncertainty of the design variable **x** is constant, the variation of the response is much smaller if $\mathbf{x} = \mathbf{x}_2$. It should be noted that the reliable and robust design is usually (although not necessarily), suboptimal to the reliable design **B** which is in turn suboptimal to the deterministic design **A**. This is the design trade-off among optimality, reliability and robustness.

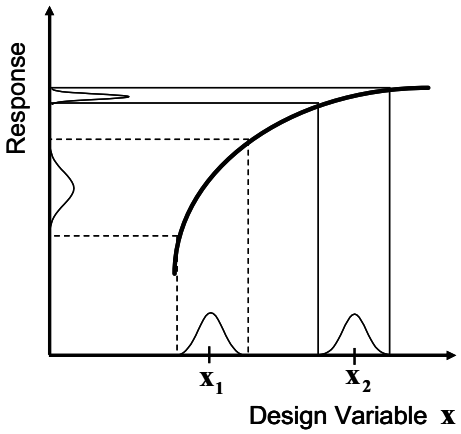



Figure 2. Geometric interpretation of robust design

The robust design problem is in general, expressed as

$$\min_{\mathbf{d},\boldsymbol{\mu}_\mathbf{X}} \mathbf{V_f} = [V_{f_1}, V_{f_2}, ..., V_{f_m}] \tag{4}$$

$$\text{s.t.} \quad P\left(G_i(\mathbf{d}, \mathbf{X}, \mathbf{P}) \geq 0\right) \geq R_i = 1 - p_{f_i}, \quad i = 1, ..., n$$

$$\mathbf{d}^L \le \mathbf{d} \le \mathbf{d}^U$$
$$\boldsymbol{\mu}_{\mathbf{X}}^L \le \boldsymbol{\mu}_{\mathbf{X}} \le \boldsymbol{\mu}_{\mathbf{X}}^U$$

where $V$ indicates a variation measure. For example, $V_{f_1}$ is a variation measure of the first objective $f_1$ of the multi-objective Problem (4). Section 4 discusses the commonly used variation measures and explains what we propose in this work.

A unifying formulation for reliability and robustness is described in Section 4. The solution methodology is based on a "generalized" RBDO formulation which includes robustness considerations. For this reason, an overview of the existing RBDO methods is given next.

## 3. Overview of Reliability-Based Design Optimization

Optimization is concerned with achieving the best outcome of a given objective while satisfying certain restrictions. It has been observed that the deterministic optimum design does not necessarily have high reliability. To ensure that the optimum design is also reliable, the optimization formulation must include reliability constraints. Such a formulation is commonly referred as Reliability-Based Design Optimization (RBDO). Problem (2) is a typical RBDO formulation.

The classical RBDO method is the so-called double-loop approach. It employs two nested optimization loops; the design optimization loop (outer) and the reliability assessment loop (inner). The latter is needed for the evaluation of each probabilistic constraint. There are two different methods for the reliability assessment; the Reliability Index Approach (RIA) [2] and the Performance Measure Approach (PMA) [6,7]. Although either approach can be used, PMA is in general more efficient, especially for high reliability problems [7]. Every time the design optimization loop calls for a constraint evaluation, a reliability assessment loop is executed which searches for the Most Probable Point (MPP) in the standard normal space, based on First-Order Reliability Methods (FORM).

The PMA-based RBDO problem, which is practically the inverse of the RIA-based RBDO problem, is stated as [7]

$$\min_{\mathbf{d},\boldsymbol{\mu}_{\mathbf{X}}} f\left(\mathbf{d},\boldsymbol{\mu}_{\mathbf{X}},\boldsymbol{\mu}_{\mathbf{P}}\right) \tag{5}$$
$$\text{s.t. } G_{p_i} = (F_{G_i}{}^{-1}(\Phi(-\beta_{t_i})) \ge 0\,,\quad i = 1,...,n$$
$$\mathbf{d}^L \le \mathbf{d} \le \mathbf{d}^U$$
$$\boldsymbol{\mu}_{\mathbf{X}}^L \le \boldsymbol{\mu}_{\mathbf{X}} \le \boldsymbol{\mu}_{\mathbf{X}}^U$$

where Eq. (3) has been used to transform each probabilistic constraint to an equivalent non-negative constraint for a performance measure $G_p$. $G_p$ is a function of the target reliability index $\beta_t$. It is calculated from the following reliability minimization problem

$$G_p = \min_{\mathbf{U}} G(\mathbf{U}) \tag{6}$$
$$\text{s.t } \|\mathbf{U}\| = \beta_t$$

where the vector $\boldsymbol{U}$ represents the random variables in the standard normal space.

Using a percentile formulation, the general RBDO formulation of Eq. (2), can be equivalently stated as [6,7]

$$\min_{\mathbf{d},\boldsymbol{\mu}_{\mathbf{X}}} f\left(\mathbf{d},\boldsymbol{\mu}_{\mathbf{X}},\boldsymbol{\mu}_{\mathbf{P}}\right) \tag{7}$$
$$\text{s.t.} \quad G_i^R(\mathbf{d},\mathbf{X},\mathbf{P}) \ge 0,\quad i = 1,\cdots,n$$
$$\mathbf{d}^L \le \mathbf{d} \le \mathbf{d}^U\ ,\quad \boldsymbol{\mu}_{\mathbf{X}}^L \le \boldsymbol{\mu}_{\mathbf{X}} \le \boldsymbol{\mu}_{\mathbf{X}}^U$$

where $G^R$ is the R-percentile of the constraint $G(\mathbf{d},\mathbf{X},\mathbf{P})$. It is defined as

$$P(G(\mathbf{d},\mathbf{X},\mathbf{P}) \geq G^R) = R \tag{8}$$

where $R$ is the target reliability for the constraint. Note that $P(G(\mathbf{d},\mathbf{X},\mathbf{P}) \geq 0) \geq R$ if $G^R \geq 0$. Therefore, $G^R \geq 0$ provides an equivalent expression of the probabilistic constraints in Eq. (2). After the MPP is calculated, the R-percentile is given by

$$G^R = G\left(\mathbf{d}, \mathbf{X}_{MPP}, \mathbf{P}_{MPP}\right). \tag{9}$$

The RBDO Problems (5) or (7) involve nested optimization loops which may hinder on their computational efficiency. For this reason, two new classes of RBDO formulations have been recently proposed. The first class decouples the RBDO process into a sequence of a deterministic design optimization followed by a set of reliability assessment loops [33,34]. The series of deterministic and reliability loops is repeated until convergence. The second class of RBDO methods converts the problem into an equivalent, single-loop deterministic optimization [35,32], leading therefore, to significant efficiency improvements.

### 3.1. Decoupled RBDO

Among the decoupled RBDO methods, the Sequential Optimization and Reliability Assessment (SORA) [33] method is the most promising. It uses the reliability information from the previous cycle to shift the violated deterministic constraints in the feasible domain. This is done sequentially until convergence is achieved. SORA employs a sequence of decoupled deterministic optimization and reliability assessment loops which are performed in series. At the end of a deterministic design optimization, the reliability of each constraint is assessed. If the reliability of a particular constraint is less than the specified target, a "shifting" vector is calculated which is used to push the constraint boundary in the feasible domain. The "shifted" constraints are then used to perform a new deterministic design optimization. The series of deterministic and reliability assessment loops continues until convergence is achieved; i.e. the objective function is minimized and the target reliability of each constraint is met. At convergence the shifting distance is zero. For the reliability assessment in SORA, either of the RIA or PMA approaches can be used. Detailed information is provided in [33].

### 3.2. Single-Loop RBDO

Based on the percentile formulation of Eq. (7), a computationally efficient single-loop RBDO method has been recently developed [32]. The method relates the $\boldsymbol{\mu}_\mathbf{X}, \boldsymbol{\mu}_\mathbf{P}$ and $\mathbf{X}$, $\mathbf{P}$ vectors using the KKT optimality conditions of the inner reliability loops. In that case, the constraint gradient and the β hyper-sphere gradient must be collinear and pointing in opposite directions at the MPP point [32]. This is expressed as $\mathbf{X} = \boldsymbol{\mu}_\mathbf{X} - \boldsymbol{\sigma}\beta_t\boldsymbol{\alpha}$ and $\mathbf{P} = \boldsymbol{\mu}_\mathbf{P} - \boldsymbol{\sigma}\beta_t\boldsymbol{\alpha}$. Problem (7) can be therefore, expressed as

$$\begin{aligned}
&\min_{\mathbf{d},\boldsymbol{\mu}_\mathbf{X}} \quad f(\mathbf{d}, \boldsymbol{\mu}_\mathbf{X}, \boldsymbol{\mu}_\mathbf{P}) \\
&\text{s.t. } G_i(\mathbf{d}, \mathbf{X}_i, \mathbf{P}_i) \geq 0 \qquad i = 1,.....,n \\
&\quad \mathbf{X}_i = \boldsymbol{\mu}_\mathbf{X} - \boldsymbol{\sigma}\beta_{t_i}\boldsymbol{\alpha}_i, \quad \mathbf{P}_i = \boldsymbol{\mu}_\mathbf{p} - \boldsymbol{\sigma}\beta_{t_i}\boldsymbol{\alpha}_i \\
&\quad \boldsymbol{\alpha}_i = \boldsymbol{\sigma} * \nabla G_{i\,\mathbf{X},\mathbf{P}}(\mathbf{d}, \mathbf{X}_i, \mathbf{P}_i) / \| \boldsymbol{\sigma} * \nabla G_{i\,\mathbf{X},\mathbf{P}}(\mathbf{d}, \mathbf{X}_i, \mathbf{P}_i) \| \\
&\quad \mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U, \quad \boldsymbol{\mu}_\mathbf{X}^L \leq \boldsymbol{\mu}_\mathbf{X} \leq \boldsymbol{\mu}_\mathbf{X}^U
\end{aligned} \tag{10}$$

where $\boldsymbol{\mu}_\mathbf{X}, \boldsymbol{\mu}_\mathbf{P}$ are the mean values of vectors $\mathbf{X}$ and $\mathbf{P}$, $\beta_{t_i}$ is the target reliability index for the ith constraint, $\boldsymbol{\alpha}_i$ is the normalized gradient of the i[th] constraint and $\boldsymbol{\sigma}$ is the standard deviation vector of random variables $\mathbf{X}$ and parameters $\mathbf{P}$.

It should be noted that the single-loop RBDO method does not search for the MPP of each

constraint. Instead, the MPP of each active constraint is correctly identified at the optimum. This dramatically improves the efficiency without compromising the accuracy. The main advantage of the method is the elimination of the repeated reliability loops and its excellent convergence properties since it is based on an equivalent deterministic optimization. Detail information on the single-loop method is provided in [32].

A modified version of the single-loop probabilistic optimization of Problem (10) is used in this work for the proposed unified reliability and robust design formulation.

## 4. A Unified Formulation for reliability and Robustness

A computationally efficient unified method to reliability and robustness, based on a multi-objective optimization problem, is presented in this section. The method addresses all shortcomings of the existing methods as described in the introduction section. The preference aggregation method of Section 4.1 is used to choose the "best" solution of the multi-objective optimization problem based on designer preferences. The performance variation is assessed by a percentile difference method originally proposed in [25]. The percentiles are efficiently calculated using a variation of the Advanced Mean Value method as described in Section 4.2. The "best" design is calculated using the single-loop probabilistic optimization method of Section 3.2.

A multi-objective formulation for reliability and robustness is proposed by combining the RBDO and robust design formulations of Problems (2) and (4), respectively. The variation measure $V_{f_i}$ of objective (performance) $f_i$ is expressed by the spread of its PDF, which is simply the percentile difference $\Delta R_{f_i} = f_i^{R_2} - f_i^{R_1}$ where $f_i^{R_1}$ and $f_i^{R_2}$ are a low and high percentile of $f_i$, respectively. For example, the 5th and 95th percentiles can be used. There are several advantages of using the percentile difference instead of the standard deviation in assessing variability in robust design. The percentile is related to the probability at the tail areas of the distribution and therefore, it provides more information than the standard deviation. It considers for example, the skewness of the distribution while the standard deviation only captures the dispersion around the mean value. Also, with the percentile formulation, we can easily know the confidence level of the design robustness, which is simply equal to $R_2 - R_1$.

Using the percentile difference as a variation measure, the unified reliability and robustness formulation is stated by the following multi-objective optimization problem

$$
\begin{aligned}
&\min_{\mathbf{d},\boldsymbol{\mu}_\mathbf{X}} f\left(\mathbf{d},\boldsymbol{\mu}_\mathbf{X},\boldsymbol{\mu}_\mathbf{P}\right) \\
&\min_{\mathbf{d},\boldsymbol{\mu}_\mathbf{X}} \boldsymbol{\Delta}\mathbf{R}_\mathbf{f} = [\Delta R_{f_1}, \Delta R_{f_2}, ..., \Delta R_{f_m}] \\
\text{s.t.}\quad & P\left(G_i\left(\mathbf{d},\mathbf{X},\mathbf{P}\right) \geq 0\right) \geq 1 - p_{f_i}, \quad i = 1,...,n \\
& \mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U \\
& \boldsymbol{\mu}_\mathbf{X}^L \leq \boldsymbol{\mu}_\mathbf{X} \leq \boldsymbol{\mu}_\mathbf{X}^U
\end{aligned}
\tag{11}
$$

where $\Delta R_f = f^{R_2} - f^{R_1}$. The first objective minimizes a mean performance and the remaining objectives minimize the distribution spread of all performance measures. The trade-off between all objectives can play an important role in the selection of the best design. It is common to perform this trade-off using a weighted-sum approach which usually leads to undesired results.

A multi-objective design problem generally has a set of possible "best" solutions, known as the Pareto set or Pareto frontier. The Pareto set contains all feasible points for which there is no other point which performs better on all objectives. To decide which of all the Pareto points is the best design, the objectives must be traded off against each other.

Some researchers have proposed to use compromise programming (CP) [17,22,23] to address the trade-off mentioned above. The basic idea in CP is to identify the entire Pareto frontier, and allow the user to choose among the Pareto points. CP commences with the identification of an ideal solution (*utopia point*), where each attribute under consideration simultaneously achieves its optimum value. As the ideal point is unachievable in general, the designer seeks a solution which is as close as possible to the ideal point. The closest feasible point to the utopia point for a given weight set is guaranteed to belong to the set of Pareto points. By varying the weighting factors, the full set of Pareto points can be obtained [17]. In contrast, a weighted-sum (WS) method used for the same purpose may fail to locate all Pareto points. It has been shown [36] that for every Pareto point of a convex multi-objective optimization problem there exists a (nonzero) weight vector $\mathbf{w} > 0$ such that this Pareto point is an optimal solution of the Weighted-sum Problem (WSP). However, not every Pareto solution of a general (nonconvex) problem can be found by solving the corresponding WSP. Also, it has been concluded from numerical experiments [17] that even for convex multi-objective optimization problems, an evenly distributed set of weights fails to produce an even distribution of points from all parts of the Pareto set if a weighted-sum aggregation is used. Details on compromise programming in engineering design can be found in [17].

### 4.1. Preference Aggregation Method

An alternative to compromise programming is preference aggregation. A family of aggregation functions for modeling all possible trade-offs in engineering design has been presented in [28-30]. Methods using these aggregation functions to aggregate preferences of designers on performance measures are called *preference aggregation methods*. In this work, a preference aggregation method is used to address the robust design of Problem (11). It has been mentioned that the weighted-sum methods have serious drawbacks for optimization [17,21] because they are usually unable to reach some Pareto solutions. The recovery of the entire Pareto frontier may be computationally intractable and even if it is available, it may be beyond the capacity of the human designer to choose the best point from the Pareto set. The preference aggregation method surmounts these difficulties. This is the main reason we use preference aggregation methods instead of compromise programming in this work.

*Preference aggregation* is a formal approach for reconciling multiple conflicting criteria in design [28,29]. *Preference functions* or *preferences* are defined for each criterion and the various functions are *aggregated* into a single overall preference function by means of *aggregation operators*. Preference functions take values on [0,1], where a preference of 0 indicates a criterion is unacceptable, while a preference of 1 denotes complete satisfaction. A set of properties was offered in [28] that seem intuitively reasonable for combining preferences in engineering design, indicating that decisions can have different *trade-offs*. The set includes the annihilation, idempotency, monotonicity, commutativity and continuity properties which are mathematically described as

$$h\left[(0,w_1),(h_2,w_2)\right] = h\left[(h_1,w_1),(0,w_2)\right] = 0 \qquad \text{(annihilation)} \qquad (12)$$

$$h\left[(h_1,w_1),(h_1,w_2)\right] = h_1 \qquad \text{(idempotency)} \qquad (13)$$

$$h\left[(h_1,w_1),(h_2,w_2)\right] \le h\left[(h_1,w_1),(h_2^*,w_2)\right] \text{ if } h_2 \le h_2^* \qquad \text{(monotonicity)} \qquad (14)$$

$$h\left[(h_1,w_1),(h_2,w_2)\right] = h\left[(h_2,w_2),(h_1,w_1)\right] \qquad \text{(commutativity)} \qquad (15)$$

$$h\left[(h_1,w_1),(h_2,w_2)\right] = \lim_{h_1^* \to h_1} h\left[(h_1^*,w_1),(h_2,w_2)\right]. \qquad \text{(continuity)} \qquad (16)$$

where $(h_1,w_1)$ and $(h_2,w_2)$ are the individual preference functions to be aggregated and their corresponding importance weights and $h$ is the aggregate preference function.

The aggregation properties distinguish between *compensating* trade-offs, where high performance on one criterion can make up for lower performance on another, and *non-*

*compensating* trade-offs, in which the lowest performances should be raised first. It was shown [29] that the level of compensation can vary continuously, and that there is a family of aggregation operators $h^s$ that satisfies the original set of properties for design (Eqs 12-16) and can capture all possible trade-offs. For a two-attribute design problem, $h^s$ is given by

$$h^s\left[(h_1, w_1),(h_2, w_2)\right] = \left(\frac{w_1 h_1^s + w_2 h_2^s}{w_1 + w_2}\right)^{1/s}. \tag{17}$$

The parameter $s$ can be interpreted as a measure of the level of compensation, or trade-off. Higher values of $s$ indicate a greater willingness to allow high preference for one criterion to compensate for lower values of another. It is shown [29] that if $s \to 0$, the aggregation of the two preferences provides maximum compensation. In this case, Eq. (17) reduces to the following geometric product of the two preferences $h_1$, $h_2$

$$h^s = h^s_{prod} = \left[h_1^{w_1} h_2^{w_2}\right]^{1/(w_1+w_2)}. \tag{18}$$

To the contrary, if $s \to -\infty$ the aggregation of the two preferences provides no compensation at all and Eq. (17) reduces to

$$h^s = \min(h_1, h_2). \tag{19}$$

When the parameters $s$ and $w$ (or equivalently, weight ratio $w_2 / w_1$) are correctly chosen, the "best" design can be located by maximizing $h^s$.

The weighted sum is a special case with $s$ =1. It has been shown [29] that for any Pareto optimal point in a given set, there is always a choice of a weight ratio and a level of compensation $s$ that selects that point as the most preferred. It has been also shown that for any fixed $s$, there are Pareto sets in which some Pareto points can *never* be selected by any choice of weights. In particular, the weighted-sum approach ($s$ =1) may not be able to select all Pareto points. In order to avoid this arbitrary and meaningless use of weights, a rigorous, provable procedure of "indifference points" has been developed for calculating the proper trade-off parameters [30].

### 4.2. Percentile Calculation using the Advanced Mean Value Method

The percentiles $f^{R_1}$ and $f^{R_2}$ of a performance measure $f$ (see Eq. 11) can be in general calculated using two reliability calculations for estimating the two Most Probable Points corresponding to $R_1$ and $R_2$. In this work, a computationally more efficient method is used based on the Advanced Mean Value (AMV) method [31].

The AMV method has been originally proposed as a computationally efficient method for generating the cumulative distribution function (CDF) of performance $f$. It uses a simple correction to compensate for errors introduced from a Taylor series truncation. The performance $f(\mathbf{X})$ is first linearized around the mean design point. A limit state function is then defined as

$$g(\mathbf{X}) = f(\mathbf{X}) - f_0 \tag{20}$$

where $f_0$ is a particular value of the performance function. Based on the CDF definition, we have the following first-order relation

$$P(f \le f_0) = P(g \le 0) = \Phi(-\beta), \tag{21}$$

where $\Phi$ is the standard normal cumulative distribution function and $\beta$ is the reliability index.

For the calculation of the R-percentile $f^R$, the reliability index $\beta$ is calculated from $\Phi(-\beta) = R$ if $R \ge 50\%$ and $\Phi(-\beta) = 1 - R$ if $R \le 50\%$.

Using the linear approximation of $g(\mathbf{X})$ at the mean value point $\boldsymbol{\mu}_{\mathbf{X}}$, the MPP is given by

$$\mathbf{U}^* = -\beta\boldsymbol{\sigma}_\mathbf{X}\frac{\nabla g(\boldsymbol{\mu}_\mathbf{X})}{|\boldsymbol{\sigma}_\mathbf{X}\nabla g(\boldsymbol{\mu}_\mathbf{X})|} = -\beta\boldsymbol{\sigma}_\mathbf{X}\frac{\nabla f(\boldsymbol{\mu}_\mathbf{X})}{|\boldsymbol{\sigma}_\mathbf{X}\nabla f(\boldsymbol{\mu}_\mathbf{X})|}. \quad (22)$$

in the standard normal space, if the random variables $\mathbf{X}$ are normally distributed with $\mathbf{X} \sim N(\boldsymbol{\mu}_\mathbf{X}, \boldsymbol{\sigma}_\mathbf{X})$. In the original X space, the MPP coordinates are

$$\mathbf{X}^* = \mathbf{U}^*\boldsymbol{\sigma}_\mathbf{X} + \boldsymbol{\mu}_\mathbf{X}. \quad (23)$$

For non-normal random variables, a non-linear transformation is needed.

The AMV method "corrects" the relation of Eq. (21) as

$$P(f \le f(\mathbf{X}^*)) = \Phi(-\beta) \quad (24)$$

by replacing the $f_0$ value at which the reliability index $\beta$ is calculated by $f(\mathbf{X}^*)$. Based on Eq. (24), the R-percentile is equal to $f(\mathbf{X}^*)$.

The described R-percentile calculation using the AMV method requires only one extra function evaluation (i.e. $f(\mathbf{X}^*)$). The gradient of $f(\mathbf{X})$ at the mean design point $\boldsymbol{\mu}_\mathbf{X}$ (see Eq. 22) is usually known, if a gradient-based optimization method is used for solving the robust optimization problem.

## 5. Examples

### 5.1. A Mathematical Example

A simple mathematical example is first used to demonstrate the proposed methodology for reliability and robustness, using preference aggregation methods to handle the trade-off between reliability and robustness. The following mathematical problem, first appeared in [17], is used

$$\min_{\mathbf{x}} f(\mathbf{x}) = (x_1 - 4)^3 + (x_1 - 3)^4 + (x_2 - 5)^2 + 10$$
$$\text{s.t. } G(\mathbf{x}) = -x_1 - x_2 + 6.45 \le 0$$
$$1 \le x_i \le 10, \quad i = 1,2.$$

Assuming that only the variation of the objective is important, the reliable/robust problem is formulated as,

$$\min_{\boldsymbol{\mu}_\mathbf{x}} f$$
$$\min_{\boldsymbol{\mu}_\mathbf{x}} \Delta R_f(\mathbf{X})$$
$$\text{s.t. } P(G(\mathbf{X}) \ge 0) \ge R$$
$$P(1 \le \mathbf{X} \le 10) \ge R.$$

The two design variables are assumed normally distributed with $X_i \sim N(\mu_{x_i}, 0.4), \ i = 1,2$. The percentile difference is calculated as $\Delta R_f = f^{R_2} - f^{R_1}$ with $R_2 = 95\%$ and $R_1 = 5\%$. Two separate single-objective optimization problems are first solved in order to establish the *utopia* point. Each problem is composed of one of the two objectives and all constraints. The first problem is the conventional RBDO problem. It minimizes $f$ subject to the probabilistic constraints. Its solution yields an optimum objective of $\mu_f^* = 5.4745$ at the design vector $\boldsymbol{\mu}_\mathbf{x}^* = [2.2, 5.9471]$. The superscript * indicates optimal value. The second problem minimizes $\Delta R_f$. It is a single-objective, purely robust optimization problem. Its solution yields an

optimum objective value and design vector of $\Delta R_f^* = 2.8982$ and $\boldsymbol{\mu}_{\mathbf{x}}^* = [3.4668, 5.5332]$, respectively.

Because we have two objectives, all optimal solutions belong to a Pareto set. For the calculation of the Pareto set, the two objectives are aggregated using the preference aggregation method of Section 4.1. Two preference functions $h_1$ and $h_2$ are first defined for $f$ and $\Delta R_f$, respectively. Fig. 3 shows $h_1$, which has the following linear form

$$h_1(\mu_f) = \begin{cases} \dfrac{3\mu_f^* - \mu_f}{3\mu_f^* - \mu_f^*} & \text{if} \quad \mu_f \le 3\mu_f^* \\ 0 & \text{if} \quad 3\mu_f^* < \mu_f \end{cases}.$$

Note that for all feasible designs, the mean objective value $\mu_f$ is *always* greater or equal to $\mu_f^*$. A "cut-off" value of $\mu_f = 3\mu_f^*$ is used, assuming that if $\mu_f > 3\mu_f^*$ the design is unacceptable. Therefore, $h_1 = 1$ if $\mu_f = \mu_f^*$ and $h_1 = 0$ if $\mu_f > 3\mu_f^*$. The preference function $h_2$ for the second objective $\Delta R_f$ is defined in a similar manner. The assumed "cut-off" value is equal to $\Delta R_f = 8\Delta R_f^*$.

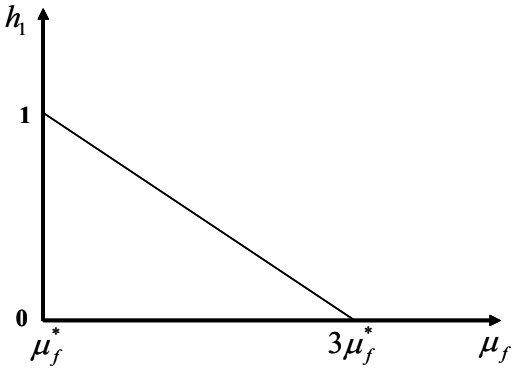



Figure 3. Preference function $h_1$ for the mathematical example

The two objectives are aggregated using

$$h = \left( \frac{w_1 h_1^s + w_2 h_2^s}{w_1 + w_2} \right)^{1/s}; \; w_1 + w_2 = 1, \; 0 \le w_1 \le 1.$$

The overall preference $h$ is maximized by solving the following probabilistic optimization problem

$$\max_{\boldsymbol{\mu}_{\mathbf{x}}} h \tag{25}$$

$$\text{s.t. } P(G(\mathbf{x}) \ge 0) \ge R$$

$$P(1 \le x_i \le 10) \ge R, \quad i = 1,2,$$

using the single-loop RBDO algorithm of Section 3.2.

The results are summarized in Fig. 4 and Table 1. For illustration purposes, we have assumed $s = -1$. Fig. 4 shows the trade-off between $\mu_f / \mu_f^*$ and $\Delta R_f / \Delta R_f^*$. Note that the Pareto set does not cover the entire range between the reliable and robust optima. For $w_1 \geq 0.48$ (see Table 1), the overall design is dominated by the reliable design. Note that all designs are well spaced along the Pareto set.

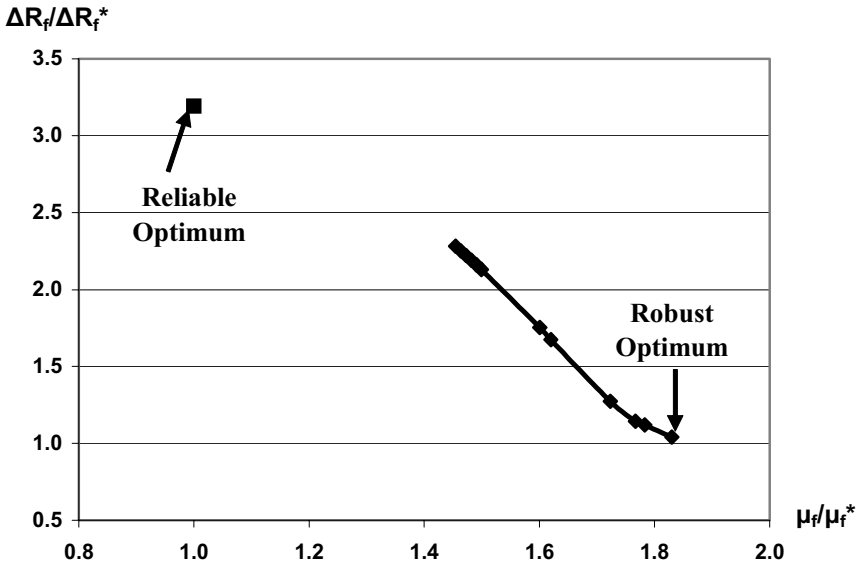



Figure 4. Pareto set for the mathematical example, using the preference aggregation method

Table 1 shows the exact values for some points on the Pareto set and the corresponding values of the optimal point $\mu_x = \lfloor \mu_{x_1} \quad \mu_{x_2} \rfloor$, overall preference function *h* and the value of constraint $G$ for $0 \leq w_1 \leq 1$. As indicated by its zero value, constraint $G$ is always active.

Table 1. Pareto set details for the mathematical example, using the preference aggregation method

| $w_1$ | $w_2$ | $\mu_f / \mu_f^*$ | $\Delta R_f / \Delta R_f^*$ | $\mu_{x_1}$ | $\mu_{x_2}$ | $h$ | $G$ |
|---|---|---|---|---|---|---|---|
| ***0.00[a]*** | ***1.00*** | ***1.8300*** | ***1.0399*** | ***3.4775*** | ***4.6695*** | ***0.9943*** | ***0.0000*** |
| 0.10 | 0.90 | 1.7833 | 1.1190 | 3.3635 | 4.9443 | 0.9260 | -0.1608 |
| 0.20 | 0.80 | 1.7673 | 1.1447 | 3.3036 | 4.9353 | 0.8761 | -0.0918 |
| 0.30 | 0.70 | 1.7233 | 1.2747 | 3.1721 | 4.9785 | 0.8343 | -0.0036 |
| 0.40 | 0.60 | 1.6199 | 1.6758 | 2.9455 | 5.2015 | 0.8040 | 0.0000 |
| 0.41 | 0.59 | 1.6006 | 1.7530 | 2.9104 | 5.2367 | 0.8019 | 0.0000 |
| 0.42 | 0.58 | 1.4993 | 2.1312 | 2.7498 | 5.3973 | 0.7990 | 0.0000 |
| 0.43 | 0.57 | 1.4907 | 2.1606 | 2.7377 | 5.4094 | 0.7980 | 0.0000 |
| 0.44 | 0.56 | 1.4819 | 2.1903 | 2.7254 | 5.4216 | 0.7972 | 0.0000 |
| 0.45 | 0.55 | 1.4730 | 2.2201 | 2.7131 | 5.4340 | 0.7965 | 0.0000 |
| 0.46 | 0.54 | 1.4638 | 2.2501 | 2.7007 | 5.4463 | 0.7960 | 0.0000 |
| 0.47 | 0.53 | 1.4545 | 2.2802 | 2.6882 | 5.4588 | 0.7956 | 0.0000 |
| 0.48 | 0.52 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 0.8083 | 0.0000 |
| 0.49 | 0.51 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 0.8113 | 0.0000 |
| 0.50 | 0.50 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 0.8143 | 0.0000 |
| 0.60 | 0.40 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 0.8457 | 0.0000 |
| 0.70 | 0.30 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 0.8797 | 0.0000 |
| 0.80 | 0.20 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 0.9164 | 0.0000 |
| 0.90 | 0.10 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 0.9564 | 0.0000 |
| ***1.00[b]*** | ***0.00*** | ***1.0000*** | ***3.1923*** | ***2.2000*** | ***5.9471*** | ***1.0000*** | ***0.0000*** |

**a: Robust design; b: Reliable design**

For comparison purposes, the same problem is solved using the weighted-sum approach. The following probabilistic optimization problem is solved instead of Problem (25)

$$\min_{\boldsymbol{\mu}_x} f = \left( w_1 \frac{\mu_f}{\mu_f^*} + w_2 \frac{\Delta R_f}{\Delta R_f^*} \right)$$

$$\text{s.t. } P(G(\mathbf{x}) \geq 0) \geq R$$

$$P\left(1 \leq x_i \leq 10\right) \geq R, \quad i = 1,2 .$$

The results are summarized in Fig. 5 and Table 2. As shown in Fig. 5, the weighted-sum approach fails to identify a large portion of the Pareto set. It only identifies a small region around the robust optimum extreme.

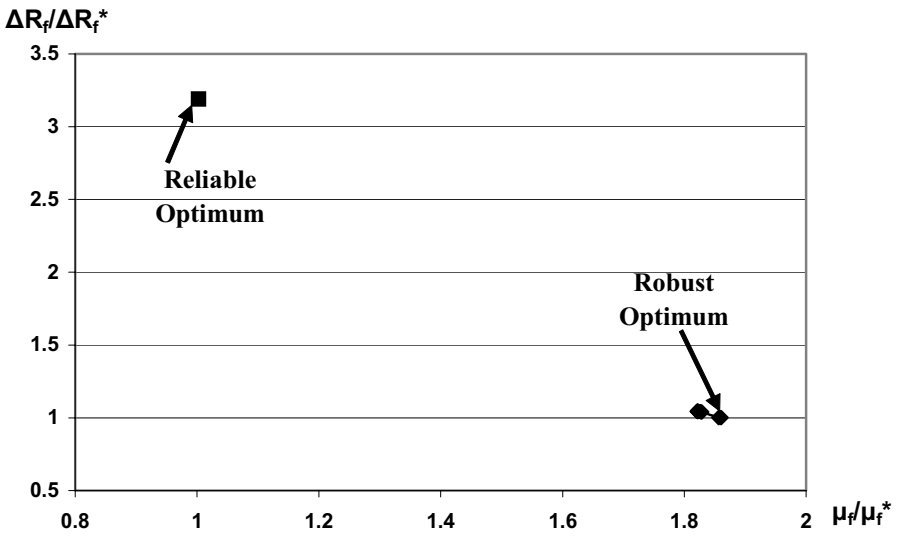



Figure 5. Pareto set for the mathematical example, using the weighted-sum method

Table 2. Pareto set details for the mathematical example, using the weighted-sum method

| $w_1$ | $w_2$ | $\mu_f / \mu_f^*$ | $\Delta R_f / \Delta R_f^*$ | $\mu_{x_1}$ | $\mu_{x_2}$ | $f$ | $G$ |
|---|---|---|---|---|---|---|---|
| ***0.00[a]*** | ***1.00*** | ***1.8600*** | ***1.0000*** | ***3.4666*** | ***5.5359*** | ***1.0000*** | ***0.8554*** |
| 0.10 | 0.90 | 1.8566 | 1.0002 | 3.4637 | 5.5213 | 1.0858 | 0.8379 |
| 0.20 | 0.80 | 1.8284 | 1.0403 | 3.4727 | 4.6744 | 1.1979 | 0.0000 |
| 0.30 | 0.70 | 1.8263 | 1.0410 | 3.4669 | 4.6802 | 1.2766 | 0.0000 |
| 0.40 | 0.60 | 1.8224 | 1.0433 | 3.4555 | 4.6916 | 1.3549 | 0.0000 |
| 0.41 | 0.59 | 1.8226 | 1.0431 | 3.4561 | 4.6909 | 1.3627 | 0.0000 |
| 0.42 | 0.58 | 1.8215 | 1.0440 | 3.4528 | 4.6943 | 1.3705 | 0.0000 |
| 0.43 | 0.57 | 1.8218 | 1.0437 | 3.4538 | 4.6933 | 1.3783 | 0.0000 |
| 0.44 | 0.56 | 1.8211 | 1.0442 | 3.4517 | 4.6954 | 1.3861 | 0.0000 |
| 0.45 | 0.55 | 1.8212 | 1.0441 | 3.4521 | 4.6950 | 1.3938 | 0.0000 |
| 0.46 | 0.54 | 1.0026 | 3.1901 | 2.2025 | 5.9446 | 2.1838 | 0.0000 |
| 0.47 | 0.53 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 2.1619 | 0.0000 |
| 0.48 | 0.52 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 2.1400 | 0.0000 |
| 0.49 | 0.51 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 2.1181 | 0.0000 |
| 0.50 | 0.50 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 2.0962 | 0.0000 |
| 0.60 | 0.40 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 1.8769 | 0.0000 |
| 0.70 | 0.30 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 1.6577 | 0.0000 |
| 0.80 | 0.20 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 1.4385 | 0.0000 |
| 0.90 | 0.10 | 1.0000 | 3.1923 | 2.2000 | 5.9471 | 1.2192 | 0.0000 |
| ***1.00[b]*** | ***0.00*** | ***1.0000*** | ***3.1923*** | ***2.2000*** | ***5.9471*** | ***1.0000*** | ***0.0000*** |

**a: Robust design; b: Reliable design**

### 5.2. A Cantilever Beam Example

In this example, a cantilever beam in vertical and lateral bending [37] is used (see Fig. 6). The beam is loaded at its tip by the vertical and lateral loads Y and Z, respectively. Its length L is equal to 100 in. The width *w* and thickness *t* of the cross-section are random design variables. The

objective is to minimize the weight of the beam. This is equivalent to minimizing $w * t$, assuming that the material density and the beam length are constant.

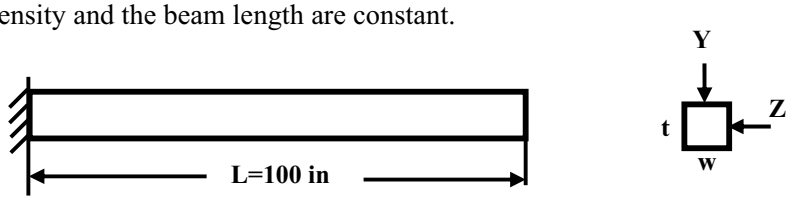



Figure 6. Cantilever beam under vertical and lateral bending.

One non-linear failure mode is used representing yielding at the fixed end of the cantilever. The RBDO problem is formulated as,

$$\min_{\mu_w,\mu_t} f = \mu_w * \mu_t$$

$$\text{s.t. } P(G_1(\mathbf{X}) \geq 0) \geq R$$

$$0 \leq \mu_w, \mu_t \leq 5$$

where the limit state $G_1(y,Z,Y,w,t) = y - (\frac{600}{wt^2} * Y + \frac{600}{w^2 t} * Z)$ represents the failure mode. The random design variables $w$ and $t$ are normally distributed with $\sigma_w = \sigma_t = 0.225$ . Y, Z, y and E are normally distributed random parameters with Y~ N (1000, 100) lb, Z~ N (500,100) lb, y~ N (40000, 2000) psi and E~ N $(29*10^6, 1.45*10^6)$ psi; y is the random yield strength, Z and Y are mutually independent random loads in the vertical and lateral directions respectively, and E is the Young modulus. A reliability index $\beta = 3$ is used.

For the reliable/robust problem one more objective is added representing the variation of the beam tip displacement. The formulation is as follows,

$$\min_{\mu_w,\mu_t} f = \mu_w * \mu_t$$

$$\min_{\mu_w,\mu_t} \Delta R_\delta(w,t,E,Y,Z)$$

$$\text{s.t. } P(G_1(\mathbf{X}) \geq 0) \geq R$$

$$0 \leq \mu_w, \mu_t \leq 5$$

where the tip displacement $\delta$ is given by $\delta(w,t,E,Y,Z) = \frac{4L^3}{Ewt}\sqrt{(\frac{Y}{t^2})^2 + (\frac{Z}{w^2})^2}$ .

Two objectives are simultaneously minimized subject to one probabilistic constraint. If the beam cross-sectional area is minimized, the beam stiffness is also minimized which usually leads to a large variation of the tip displacement. It is expected therefore, to have a trade-off between the two objectives. The percentile difference for the tip displacement is calculated as $\Delta R_\delta = \delta^{R_2} - \delta^{R_1}$ with $R_2 = 95\%$ and $R_1 = 5\%$ . Similarly to the previous example, two separate single-objective optimization problems are first solved in order to establish the *utopia* point. The first problem (conventional RBDO) yields an optimum objective of $\mu_f^* = 11.2884$ for the design vector $[w^*, t^*] = [2.9421, 3.8369]$. The second single-objective, purely robust optimization problem yields a solution of $\Delta R_\delta^* = 0.1440$ for $[w^*, t^*] = [5,5]$.

For the calculation of the Pareto set, two linear preference functions $h_1$ and $h_2$ are used, corresponding to the two objectives $f$ and $\Delta R_\delta$. They are all defined similarly to the previous

example. Their "cut-off" values are $\mu_f = 3\mu_f^*$, $\Delta R_\delta = 15\Delta R_\delta^*$ for $h_1$ and $h_2$, respectively. The two objectives are aggregated using

$$h = \left( \frac{w_1 h_1^s + w_2 h_2^s}{w_1 + w_2} \right)^{1/s}; \; w_1 + w_2 = 1, \; 0 \le w_1 \le 1$$

which is maximized by solving the following probabilistic optimization problem

$$\max_{\mu_w, \mu_t} h$$
$$\text{s.t. } P(G_1(\mathbf{X}) \ge 0) \ge R$$
$$0 \le \mu_w, \mu_t \le 5,$$

using the single-loop RBDO method of Section 3.2.

The results for this example are summarized in Figure 7 and Tables 3 and 4. A compensation level of $s = -1$ is assumed. The Pareto frontier of Fig. 7 shows the trade-off between $\mu_f / \mu_f^*$ and $\Delta R_\delta / \Delta R_\delta^*$. The designs are almost equally spaced between the two extremes of reliable and robust designs.

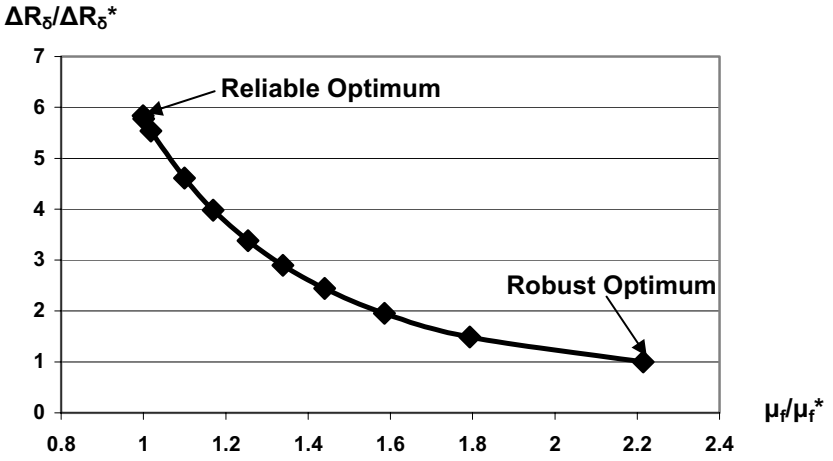



Figure 7. Pareto set for the beam example; trade-off between $\Delta R_\delta / \Delta R_\delta^*$ and $\mu_f / \mu_f^*$

Table 3 shows the exact values of the Pareto points and the corresponding values of constraint $G_1$ for ten equally spaced segments of the $0 \le w_1 \le 1$ domain. As indicated by their positive values, constraint $G_1$ is inactive for $0 \le w_1 \le 0.8$ and active for $0.9 \le w_1$ .

Table 3. Pareto set details for the beam example

| $w_1$ | $w_2$ | $\mu_f / \mu_f^*$ | $\Delta R_\delta / \Delta R_\delta^*$ | $G_1$ |
|---|---|---|---|---|
| ***0.0[a]*** | ***1.0*** | ***2.2147*** | ***0.9999*** | ***26054.0*** |
| 0.1 | 0.9 | 1.7930 | 1.4881 | 22672.9 |
| 0.2 | 0.8 | 1.5860 | 1.9506 | 19922.2 |
| 0.3 | 0.7 | 1.4403 | 2.4413 | 17111.9 |
| 0.4 | 0.6 | 1.3385 | 2.8985 | 14574.7 |
| 0.5 | 0.5 | 1.2541 | 3.3780 | 12053.2 |
| 0.6 | 0.4 | 1.1696 | 3.9830 | 8837.4 |
| 0.7 | 0.3 | 1.0994 | 4.6126 | 5635.8 |
| 0.8 | 0.2 | 1.0182 | 5.5360 | 1121.7 |
| 0.9 | 0.1 | 1.0002 | 5.7771 | 0.0 |
| ***1.0[b]*** | ***0.0*** | ***0.9993*** | ***5.8376*** | ***0.0*** |

**a: Robust design; b: Reliable design**

Table 4. Pareto set details for the beam example (Cont.)

| $w_1$ | $w_2$ | $h_1$ | $h_2$ | $h$ | $w$ | $t$ |
|---|---|---|---|---|---|---|
| ***0.0[a]*** | ***1.0*** | ***0.3927*** | ***1.0000*** | ***1.0000*** | ***5.0000*** | ***5.0000*** |
| 0.1 | 0.9 | 0.6035 | 0.9651 | 0.9106 | 4.1995 | 4.8198 |
| 0.2 | 0.8 | 0.7070 | 0.9321 | 0.8763 | 3.7938 | 4.7189 |
| 0.3 | 0.7 | 0.7798 | 0.8970 | 0.8583 | 3.6199 | 4.4914 |
| 0.4 | 0.6 | 0.8307 | 0.8644 | 0.8506 | 3.4970 | 4.3207 |
| 0.5 | 0.5 | 0.8729 | 0.8301 | 0.8510 | 3.3765 | 4.1928 |
| 0.6 | 0.4 | 0.9152 | 0.7869 | 0.8592 | 3.2651 | 4.0437 |
| 0.7 | 0.3 | 0.9503 | 0.7420 | 0.8765 | 3.1734 | 3.9109 |
| 0.8 | 0.2 | 0.9909 | 0.6760 | 0.9065 | 3.0397 | 3.7810 |
| 0.9 | 0.1 | 0.9999 | 0.6588 | 0.9507 | 2.9978 | 3.7665 |
| ***1.0[b]*** | ***0.0*** | ***1.0000*** | ***0.6545*** | ***1.0000*** | ***2.9319*** | ***3.8477*** |

**a: Robust design; b: Reliable design**

The constraint $G_1$ is always inactive except for designs close to the reliable design where it becomes active. Table 4 shows the values of the optimal design vector $[w,t]$ and the corresponding preference functions $h_1$ and $h_2$ and the overall aggregate preference function $h$ for the points on the Pareto set.

## 6. Summary and Conclusions

A computationally efficient unified method for reliability and robustness, based on a multi-objective optimization formulation, has been presented. The preference aggregation method is used to choose the "best" solution of the multi-objective optimization problem based on designer preferences. The proposed methodology addresses the shortcomings of the commonly used weighted-sum method which may fail to identify regions of the Pareto set of optimal solutions. Furthermore, it does not require the calculation of the entire Pareto set. It can identify the "best" design on the Pareto set, based on designer preferences and a rigorous, provable procedure of "indifference points."

The performance variation is assessed by a percentile difference method. The percentiles are efficiently calculated using a variation of the Advanced Mean Value method which provides much more accurate results compared with the commonly used Taylor series expansion for calculating the variance of a performance measure. The "best" design (optimal, reliable and robust) is calculated using an efficient single-loop probabilistic optimization method. Two examples illustrated the benefits of the proposed method and demonstrated its applicability.

## Acknowledgments

The present study was partially funded by General Motors Research and Development Center and the Automotive Research Center, a U.S. Army Center of Excellence in Modeling and Simulation of Ground Vehicles. The support is gratefully acknowledged. The authors like to thank Prof. Michael J. Scott from the University of Illinois at Chicago for the valuable discussions on the use of preference aggregation methods in robust design.

## References


1. Reddy, M. V., Granhdi, R. V. and Hopkins, D. A., "Reliability Based Structural Optimization: A Simplified Safety Index Approach," *Computers and Structures*, **53**(6), 1407-1418, 1994.
2. Lee, J. O., Yang, Y. O. and Ruy, W. S., "A Comparative Study on Reliability Index andTarget Performance Based Probabilistic Structural Design Optimization," *Computers and Structures*, **80,** 257-269, 2002.
3. Wu, Y.-T., "Computational Methods for Efficient Structural Reliability and Reliability Sensitivity Analysis," *AIAA Journa*l, **32**(8), 1717-1723, 1994.
4. Wu, Y. –T. and Wang, W., "A New Method for Efficient Reliability-Based Design Optimization," Proceedings of 7th Special Conference on Probabilistic Mechanics & Structural Reliability, 274-277, 1996.
5. Zou, T., Mourelatos, Z. P., Mahadevan, S. and J. Tu, "Component and System Reliability Analysis Using an Indicator Response Surface Monte Carlo Approach," Proceedings of ASME Design Engineering Technical Conferences, DETC2003/DAC-48708, 2003.
6. Tu, J., Choi, K. K. and Park, Y. H., "A New Study on Reliability-Based Design Optimization", *ASME Journal of Mechanical Design*, **121**, 557-564, 1999.
7. Youn, B. D., Choi, K. K. and Park, Y. H., "Hybrid Analysis Method for Reliability-Based Design Optimization," *ASME Journal of Mechanical Design*, **125**(2), 221-232, 2003.
8. Wu, Y.-T. and Wang, W., "Efficient Probabilistic Design by Converting Reliability Constraints to Approximately Equivalent Deterministic Constraints, " *Journal of Integrated Design and Process Sciences*, **2**(4), 13-21, 1998.
9. Phadke, M. S., Quality Engineering Using Robust Design, Prentice Hall, NJ, 1989.
10. Taguchi, G., Elsayed, E. and Hsiang, T., *Quality Engineering in Production Systems*, McGraw-Hill, NY, 1989.
11. Chandra, M. J., Statistical Quality Control, *CRC Press*, Boca Raton, FL, 2001.
12. Parkinson, D. B., "Robust Design by variability Optimization," *Quality and Reliability Engineering International*, **13**, 97-102, 1997.
13. Bennett, G. and Gupta, L. C., "Least Cost Tolerance," I*nternational Journal of Production Research,* **8**(1), 65-74, 1969.
14. Forouraghi, B., "Worst-Case Tolerance Design and Quality Assurance via Genetic Algorithms," *Journal of Optimization Theory and Applications,* **113**(2), 251-268, 2000.
15. Jung, D. H. and Lee, B. C., "Development of a Simple and Efficient Method for Robust Optimization," *International Journal for Numerical Methods in Engineering,* **53,** 2201-2215, 2002.
16. Du, X. and Chen, W., "Towards a Better Understanding of Modeling Feasibility Robustness in Engineering Design," *ASME Journal of Mechanical Design*, **122**, 385-394, 2000.
17. Chen, W., Wiecek, M. M. And Zhang, J., "Quality Utility – A Compromise Programming Approach to Robust Design," *ASME Journal of Mechanical Design*, **121**(2), 179-187, 1999.
18. Youn, B. D. and Choi, K. K., "Performance Moment Integration Approach for Reliability-Based Robust Design Optimization," Proceedings of ASME Design Engineering Technical Conferences (DETC), Paper # DETC2004/DAC-57471, 2004.
19. Ramakrishnan, B. and Rao, S. S., "A Robust Optimization Approach using Taguchi's Loss Function for Solving Nonlinear Optimization Problems," Advances in Design Automation, ASME DE-32-1, 241-248, 1991.
20. Stoebner, A. M. and Mahadevan, S., "Robustness in Reliability-Based Design," Proceedings of the 41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Atlanta, GA, 2000.

21. Das, I. and Dennis, J., "A Closer Look at Drawbacks of Minimizing Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems," *Structural Optimization,* **14**(1), 63-69, 1997.
22. Bras, B. A. and Mistree, F., "A Compromise Decision Support Problem for Robust and Axiomatic Design," *ASME Journal of Mechanical Design*, **117**(1), 10-19, 1995.
23. Chen, W., Allen, J. K., Mistree, F. And Tsui, K.-L., "A Procedure for Robust Design; Minimizing Variations Caused by Noise factors and Control Factors," *ASME Journal of Mechanical Design,* **118**(4), 478-485, 1996.
24. Dai, Z., Scott, M. J. and Mourelatos, Z. P., "Robust Design using Preference Aggregation Methods," Proceedings of ASME Design Engineering Technical Conferences (DETC), Paper # DETC2003/DAC-48715, 2003.
25. Du, X., Sudjianto, A. and Chen, W., "An Integrated Framework for Optimization under Uncertainty using Inverse Reliability Strategy," Proceedings of ASME Design Engineering Technical Conferences (DETC), Paper # DETC2003/DAC-48706, 2003.
26. Parkinson, A., Sorensen, C. and Pourhassan, N., "A General Approach for Robust Optimal Design," *ASME Journal of Mechanical Design*, **115**(1), 74-80, 1993.
27. Kokkolaras, M., Mourelatos, Z. P. and Papalambros P. Y., "Design Optimization of Hierarchically Decomposed Multilevel Systems under Uncertainty," Proceedings of ASME Design Engineering Technical Conferences (DETC), Paper # DETC2004/DAC-57357, 2004.
28. Otto, K. N. and Antonsson, E. K., "Trade-Off Strategies in Engineering Design," *Research in Engineering Design*, **3**(2), 87-104, 1991.
29. Scott, M. J. and Antonsson, E. K., "Aggregation Functions for Engineering Design Trade-Offs," *Fuzzy Sets and Systems*, **99**(3), 253-264, 1998.
30. Scott, M. J. and Antonsson, E. K., "Using Indifference Points in Engineering Decisions," Proceedings of ASME Design Engineering Technical Conferences, Paper# DETC2000/ DTM-14559, 2000.
31. Wu, Y.-T., Millwater, H. R. And Cruse, T. A., "Advanced Probabilistic Structural Analysis Method of Implicit Performance Functions," *AIAA Journal*, **28**(9), 1663-1669, 1990.
32. Liang, J., Mourelatos, Z. P., and Tu, J., "A Single-Loop Method for Reliability-Based Design Optimization," Proceedings of ASME Design Engineering Technical Conferences, Paper# DETC2004/ DAC-57255, 2004.
33. Du, X. and Chen, W., "Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design," *ASME Journal of Mechanical Design*, **126**(2), 225-233, 2004.
34. Royset, J. O., Der Kiureghian, A. and Polak, E., "Reliability-based optimal structural design by the decoupling approach," *Reliability Engineering & System Safety*, **73**, 213-221, 2001.
35. Chen, X. and Hasselman, T. K. and Neill, D. J., "Reliability Based Structural Design Optimization for Practical Applications," Proceedings of the 38th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 1997.
36. Hacker, K. and Lewis, K., "Robust Design Through the Use of a Hybrid Genetic Algorithm," Proceedings of ASME Design Engineering Technical Conferences (DETC), Paper # DETC2002/DAC-34108, 2002.
37. Wu, Y.-T., Shin, Y., Sues, R. and Cesare, M., "Safety – Factor Based Approach for Probabilistic – Based Design Optimization," 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Seattle, WA, 2001.

# Monte-Carlo-Type Techniques for Processing Interval Uncertainty, and Their Engineering Applications

V. Kreinovich, J. Beck, C. Ferregut, A. Sanchez, G. R. Keller, M. Averill and S. A. Starks
*College of Engineering and NASA Pan-American Center for Earth and Environmental Studies (PACES), University of Texas, El Paso, TX 79968, USA (*vladik@cs.utep.edu*)*

**Abstract.** In engineering applications, we need to make decisions under uncertainty. Traditionally, in engineering, statistical methods are used, methods assuming that we know the probability distribution of different uncertain parameters. Usually, we can safely linearize the dependence of the desired quantities $y$ (e.g., stress at different structural points) on the uncertain parameters $x_i$ – thus enabling sensitivity analysis. Often, the number $n$ of uncertain parameters is huge, so sensitivity analysis leads to a lot of computation time. To speed up the processing, we propose to use special Monte-Carlo-type simulations.

**Keywords:** interval uncertainty, Monte-Carlo techniques, engineering applications

## 1. Introduction

Typically, in engineering applications, we need to make decisions under uncertainty. In addition to measurement errors, some uncertainty comes from the fact that we do not know how exactly the engineering devices that we produced will be used: e.g., we have limits $L_i$ on the loads $l_i$ in different rooms $i$, but we do not know how exactly these loads will be distributed – and we want to make sure that our design is safe for all possible $l_i \le L_i$.

Traditionally, in engineering, statistical methods are used, methods assuming that we know the probability distribution of different uncertain parameters. Usually, we can safely linearize the dependence of the desired quantities $y$ (e.g., stress at different structural points) on the uncertain parameters $x_i$ – thus enabling sensitivity analysis.

Often, the number $n$ of uncertain parameters is huge – e.g., in ultrasonic testing, we record (= measure) signal values at thousands moments of time. To use sensitivity analysis, we must call the model $n$ times – and if the model is complex, this leads to a lot of computation time. To speed up the processing, we can use Monte-Carlo simulations. Their main advantage is that for Monte-Carlo techniques, the required number of calls to a model depends only on the desired accuracy $\varepsilon$ and not on $n$ – so for large $n$, these methods are much faster.

In real life, we often do not know the exact probability distribution of measurement errors; we also do not know the distribution of user loads – and if we knew, it would be a disaster to, e.g., design a building that is stable against random loads, but could fall down with a rare (but allowable) combination of loads. In such cases, usually, all we know is the *intervals* of possible values of the corresponding parameters: e.g., we know that the load $l_i$ is in $[0, L_i]$.

In such situations, we can use sensitivity analysis, we can use interval techniques – but for large $n$, this takes too long. To speed up, we developed a new Monte-Carlo-type technique

for processing interval uncertainty (Trejo and Kreinovich, 2001; Kreinovich and Ferson, 2004).

In this paper, we will describe this new technique, discuss its applications to engineering problems, describe its limitations, and explain how these limitations can be overcome.

## 2. Formulation of the Problem

In many real-life situations, we are interested in the value of a quantity $y$ that is difficult (or even impossible) to measure directly. In this cases, a natural idea is to measure easier-to-measure quantities $x_1, \ldots, x_n$ that are related to the desired quantity $y$, and try to estimate $y$ based on the results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ of these measurements. To be able to produce such an estimate, we need to have an algorithm $f(x_1, \ldots, x_n)$ that, based on the values $x_1, \ldots, x_n$ of the directly measured quantities, reconstructs the value $y$ of the desired quantity as $y = f(x_1, \ldots, x_n)$. Once we have such an algorithm, we plug in the measured values of $x_i$ into this algorithm $f$, and get the following estimate for $y$: $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.

Measurements are never 100% accurate; as a result, the actual values $x_i$ of the measured quantities may somewhat differ from the measured values. In other words, we know the inputs to the algorithm $f$ only with some (measurement-related) uncertainty. Because of this input uncertainty $\widetilde{x}_i \neq x_i$, our estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ is, in general, different from the actual value $y = f(x_1, \ldots, x_n)$ of the desired quantity. In other words, uncertainty in the inputs leads to the uncertainty in the output as well. It is therefore desirable to estimate this output uncertainty. So, we arrive at the following problem:

- *We know:*
  - the algorithm $f(x_1, \ldots, x_n)$;
  - the measured values $\widetilde{x}_1, \ldots, \widetilde{x}_n$; and
  - the information about the uncertainty $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$ of each direct measurement.
- *We must estimate:* uncertainty $\Delta y = \widetilde{y} - y$ of the algorithm's output.

In order to solve this problem, we must know what are the possible types of information that we can have about the uncertainty of each measurement error $\Delta x_i$.

We do not know the exact values of the measurement errors $\Delta x_i$; as a result, in real life, we may have (and often we do have) several situations in which we get exactly exactly the same measurement results $\widetilde{x}_1, \ldots, \widetilde{x}_n$, but the actual values $x_1, \ldots, x_n$ of the measured quantity are different. Thus, to describe the uncertainty, we need to know:

- what are the possible values of $\Delta x_i$, and
- how often can different possible values occur.

In the ideal case, when we have a complete description of uncertainty, we know the exact frequency (probability) of all possible error combinations $(\Delta x_1, \ldots, \Delta x_n)$. In other words,

we know the exact probability distribution of the set of all $n$-dimensional vectors $\Delta x = (\Delta x_1, \ldots, \Delta x_n)$. Often, the measurement errors corresponding to different measurements are independent, so it is sufficient to know the distribution of each variable $x_i$. This distribution can be described, e.g., by a cumulative density function (cdf) $F_i(t) \stackrel{\rm def}{=} {\rm Prob}(x_i \le t)$.

Most traditional methods of processing uncertainty in science and engineering (see, e.g., (Wadsworth, 1990)) are based on the assumption that we have a *probabilistic uncertainty*, i.e., that the error distributions are independent, and that we know the probability distribution $F_i(t)$ for each of the variables $x_i$. However, in real life, we often do not have all this information.

In some real-life situations, we do not have any information about the frequency of different measurement error $\Delta x_i$; all we know is the range $[\Delta_i^-, \Delta_i^+]$ of possible values of this error. In this case, the only information that we have about the actual measured value $x_i = \widetilde{x}_i - \Delta x_i$ of $i$-th quantity is that $x_i$ must be in the interval $[\underline{x}_i, \overline{x}_i]$, where we denoted $\underline{x}_i \stackrel{\rm def}{=} \widetilde{x}_i - \Delta_i^+$ and $\overline{x}_i \stackrel{\rm def}{=} \widetilde{x}_i - \Delta_i^-$. The corresponding uncertainty is called *interval uncertainty*; see, e.g., (Moore, 1979; Kearfott, 1996; Kearfott and Kreinovich, 1996; Jaulin et al., 2001).

So far, we have describe two extreme situations:

- in the case of probabilistic uncertainty, we have a complete information on which values $\Delta x_i$ are possible, and what are the frequencies of different possible values;
- in the case of interval uncertainty, we only know the range of possible values of $\Delta x_i$, we do not have any information about the frequencies at all.

In many real-life cases, we have an intermediate situation: we have some (partial) information about the frequencies (probabilities) of different values of $\Delta x_i$, but we do not have the complete information about these frequencies.

How can we describe such situations? To describe the complete information about the probabilities of different values of $\Delta x_i$, we must describe, for every real number $t$, the value $F_i(t)$ of the corresponding cdf. Thus, when we have a partial information about these probabilities, it means that, instead of the exact value $F_i(t)$, we only have the range $[\underline{F}_i(t), \overline{F}_i(t)]$ of possible values of $F_i(t)$. Thus, to describe such an intermediate situation, we must describe the *bounds* $\underline{F}_i(t)$ and $\overline{F}_i(t)$ for the cdf. These bounds are called *probability boxes* (or *p-boxes*, for short) (Ferson, 2002).

Both probability distributions and intervals can be described as a particular case of p-boxes:

- a probability distribution $F_i(t)$ can be described as a degenerate p-box $[F_i(t), F_i(t)]$; and
- an interval $[a^-, a^+]$ can be described as a p-box $[\underline{F}_i(t), \overline{F}_i(t)]$ in which:
  - $\underline{F}_i(t) = 0$ for $t < a^+$ and $\underline{F}_i(t) = 1$ for $t \ge a^+$;
  - $\overline{F}_i(t) = 0$ for $t < a^-$ and $\overline{F}_i(t) = 1$ for $t \ge a^-$.

So, p-boxes are the most general way of representing these types of uncertainty.

Another way to describe partial information about the uncertainty is by using the Dempster-Shafer approach. In this approach, for each variable $x_i$, instead of a single interval $[\underline{x}_i, \overline{x}_i]$, we have several intervals $[\underline{x}_i^{(k)}, \overline{x}_i^{(k)}]$ with probabilities $p_i^{(k)}$ attached to each such interval (so that for every $i$, $p_1^{(k)} + p_2^{(k)} + \ldots = 1$). For example, we may have several experts who provide us with different intervals $[\underline{x}_i^{(k)}, \overline{x}_i^{(k)}]$, and $p_i^{(k)}$ is the probability that $k$-th expert is right. The collection of intervals with probabilities attached to different intervals constitutes a DS knowledge base.

Thus, depending on the information that we have about the uncertainty in $x_i$, we can have five different formulations of the above problem:

- we know the probability distribution $F_i(t)$ for each variable $x_i$, we know that these distributions are independent, and we must find the distribution $F(t)$ for $y = f(x_1, \ldots, x_n)$;
- we know the interval $[\underline{x}_i, \overline{x}_i]$ of possible values of each variable $x_i$, and we must find the interval $[\underline{y}, \overline{y}]$ of possible values of $y$;
- we know the p-boxes $[\underline{F}_i(t), \overline{F}_i(t)]$ that characterize the distribution of each variable $x_i$, we know that the corresponding distributions are independent, and we must find the p-box $[\underline{F}(t), \overline{F}(t)]$ that describe the variable $y$;
- we know the DS knowledge bases

  $$\langle [\underline{x}_i^{(1)}(t), \overline{x}_i^{(1)}(t)], p_i^{(1)} \rangle, \langle [\underline{x}_i^{(2)}(t), \overline{x}_i^{(2)}(t)], p_i^{(2)} \rangle, \ldots$$

  that characterize the distribution of each variable $x_i$, we know that the corresponding distributions are independent, and we must find the DS knowledge base that describe the variable $y$;
- we may also have different types of uncertainty for different variables $x_i$: e.g., we may have probabilistic uncertainty or $x_1$ and interval uncertainty for $x_2$.

It is also reasonable to consider the formulations in which the corresponding distributions may be dependent.

There exist efficient methods for solving these problems; see, e.g., (Ferson, 2002) and references therein (in particular, for interval uncertainty, see (Moore, 1979; Kearfott, 1996; Kearfott and Kreinovich, 1996; Jaulin et al., 2001)). Many of these methods are based on the fact that we know the algorithm $f$; so, instead of applying this algorithm step-by-step to the measured values $\widetilde{x}_1, \ldots, \widetilde{x}_n$, we apply this same algorithm step-by-step to the corresponding "uncertain numbers": probability distributions, intervals, and/or p-boxes.

In several practical situations, however, the algorithm is given as a *black box*: we do not know the sequence of steps forming this algorithm; we can only plug in different values into this algorithm and see the results. This situation is reasonably frequent:

– with commercial software, where the software's owners try to prevent competitors from using their algorithms, and

– with classified security-related software, where efficient security-related algorithms are kept classified to prevent the adversary from using them.

In some practical cases, the situation is made even more difficult by the fact that the software $f(x_1, \ldots, x_n)$ is so complex and requires so much time to run that it is only possible to run it a few times. This complex black-box situation is what we will analyze in this text.

*Comment.* It is worth mentioning that even for a black-box function, it may be possible to run more simulations if we do the following:

– first, we use the actual black-box function $f(x_1, \ldots, x_n)$ to provide an approximating easier-to-compute model $f_{\text{approx}}(x_1, \ldots, x_n) \approx f(x_1, \ldots, x_n)$, and

– then, we use this approximate model to estimate the uncertainty of the results.

So, if our preliminary computations show that we need more simulations that the black-box function can give us, it does not necessarily mean that the corresponding uncertainty estimation method cannot be applied to our case: we may still be able to apply it to the approximate function $f_{\text{approx}}$.

## 3. From Traditional Monte-Carlo Techniques for Probabilistic Uncertainty to Monte-Carlo-Type Techniques for Interval Uncertainty: What Was Previously Known

*Probabilistic uncertainty: Monte-Carlo techniques.* Let us first consider the case of the probabilistic uncertainty, when we know that the values $\Delta x_i$ are distributed according to the cdf $F_i(t)$, and that the corresponding random variables $\Delta x_i$ are independent. In this case, we are interested to know the distribution $F(t)$ of $\Delta y$.

In the probabilistic case, a natural idea is to use Monte-Carlo simulations. Specifically, on each iteration $k$:

– for each input variable $x_i$, we simulate the values $x_i^{(k)}$ distributed according to the known distribution $F_i(t)$;

– then, we plug the simulated values $x_i^{(k)}$ the algorithm $f$, and thus get the value $y^{(k)} = f(x_1^{(1)}, \ldots, x_n^{(k)})$.

After $N$ iterations, we get $N$ values $y^{(k)}$.

Since the inputs $x_i^{(k)}$ are independently distributed according to the corresponding input distributions $F_i(t)$, the outputs $y^{(k)}$ are distributed according to the desired distribution $F(t)$. Thus, the $N$ values $y^{(k)}$ are a sample from the unknown distribution $F(t)$. It is therefore necessary to extract information about $F(t)$ from this sample.

*Interval uncertainty: case of linearization.* Let us now consider the case of interval uncertainty.

In the interval case, we have intervals $[\underline{x}_i, \overline{x}_i]$ of possible values of each input $x_i$, and we are interested in finding the corresponding interval $[\underline{y}, \overline{y}]$ of possible values of $y$.

It is convenient to represent each interval $[\underline{x}_i, \overline{x}_i]$ by its midpoint $x_i^{\rm mid} \stackrel{\rm def}{=} \dfrac{\underline{x}_i + \overline{x}_i}{2}$ and by its half-width $\Delta_i \stackrel{\rm def}{=} \dfrac{\overline{x}_i - \overline{x}_i}{2}$, so that each such interval takes the form $[x_i^{\rm mid} - \Delta_i, x_i^{\rm mid} + \Delta_i]$. In this representation, instead of the original variables $x_i$ that take values from $\underline{x}_i$ to $\overline{x}_i$, it is often convenient to consider auxiliary variables $\delta x_i \stackrel{\rm def}{=} x_i - x_i^{\rm mid}$ that take values from $-\Delta_i$ to $\Delta_i$.

When the function $f(x_1, \ldots, x_n)$ is reasonable smooth and the box $[\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_n, \overline{x}_n]$ is reasonably small, then on this box, we can reasonably approximate the function $f$ by its linear terms:

$$f(x_1^{\rm mid} + \delta x_1, \ldots, x_n^{\rm mid} + \delta x_n) \approx y^{\rm mid} + \delta y,$$

where $\delta y \stackrel{\rm def}{=} c_1 \cdot \delta x_1 + \ldots + c_n \cdot \delta x_n$, $y^{\rm mid} \stackrel{\rm def}{=} f(x_1^{\rm mid}, \ldots, x_n^{\rm mid})$, and $c_i \stackrel{\rm def}{=} \dfrac{\partial f}{\partial x_i}$. One can easily show that when each of the variables $\delta x_i$ takes possible values from the interval $[-\Delta_i, \Delta_i]$, then the largest possible value of the linear combination $\delta y$ is

$$\Delta = |c_1| \cdot \Delta_1 + \ldots + |c_n| \cdot \Delta_n, \tag{1}$$

and the smallest possible value of $\delta y$ is $-\Delta$. Thus, in this approximation, the interval of possible values of $\delta y$ is $[-\Delta, \Delta]$, and the desired interval of possible values of $y$ is $[y^{\rm mid} - \Delta, y^{\rm mid} + \Delta]$.

*Interval uncertainty: sensitivity analysis.* For small $n$, we can use the following *sensitivity analysis* method – a method that is applicable not only for approximately linear functions $f(x_1, \ldots, x_n)$, but also for all functions that are monotonic (increasing or decreasing) with respect of each of its variables. Specifically, in the sensitivity analysis method:

- First, we apply $f$ to the results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ of direct measurements, resulting in the value $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.

- Then, for each of $n$ inputs $x_i$, we modify this input to $x_i' \neq \widetilde{x}_i$ and, leaving other inputs, apply $f$ again. By comparing the values $f(\widetilde{x}_1, \ldots, \widetilde{x}_i, x_i', \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n)$ and $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$, we decide whether $f$ in increasing or decreasing in $x_i$.

– Finally, we apply $f$ two more times to get the desired bounds for $y$ as follows: $\underline{y} = f(x_1^-, \ldots, x_n^-)$ and $\overline{y} = f(x_1^+, \ldots, x_n^+)$, where:

  - for the variables $x_i$ for which $f$ increases with $x_i$, we take $x_i^- = \underline{x}_i$ and $x_i^+ = \overline{x}_i$, and
  - for the variables $x_i$ for which $f$ decreases with $x_i$, we take $x_i^- = \overline{x}_i$ and $x_i^+ = \underline{x}_i$.

The main disadvantage of this method is that it requires $n$ calls to the program $f$. Often, the number $n$ of uncertain parameters is huge – e.g., in ultrasonic testing, we record (= measure) signal values at thousands moments of time. To use sensitivity analysis, we must call the model $n$ times – and if the model is complex, this leads to a lot of computation time.

*Interval case: Cauchy deviates method.* One way to speed up computations is to use the following Cauchy deviate method. This method works when the function $f(x_1, \ldots, x_n)$ is reasonable smooth and the box $[\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_n, \overline{x}_n]$ is reasonably small, so that on this box, we can reasonably approximate the function $f$ by its linear terms.

This method uses Cauchy distribution with a parameter $\Delta$, i.e., a distribution described by the following density function: $\rho(x) = \dfrac{\Delta}{\pi \cdot (x^2 + \Delta^2)}$. It is known that if $\xi_1, \ldots, \xi_n$ are independent variables distributed according to Cauchy distributions with parameters $\Delta_i$, then, for every $n$ real numbers $c_1, \ldots, c_n$, the corresponding linear combination $c_1 \cdot \xi_1 + \ldots + c_n \cdot \xi_n$ is also Cauchy distributed, with the parameter $\Delta$ described by the formula (1).

Thus, if we for some number of iterations $N$, we simulate $\delta x_i^{(k)}$ $(1 \le k \le N)$ as Cauchy distributed with parameter $\Delta_i$, then, in the linear approximation, the corresponding differences

$$\delta y^{(k)} \stackrel{\text{def}}{=} f(x_1^{\text{mid}} + \delta x_1^{(k)}, \ldots, x_n^{\text{mid}} + \delta x_n^{(k)}) - y^{\text{mid}}$$

are distributed according to the Cauchy distribution with the parameter $\Delta$. The resulting values $\delta y^{(1)}, \ldots, \delta y^{(N)}$ are therefore a sample from the Cauchy distribution with the unknown parameter $\Delta$. Based on this sample, we can estimate the value $\Delta$.

Simulation can be based on the functional transformation of uniformly distributed sample values: $\delta x_i^{(k)} = \Delta_i \cdot \tan(\pi \cdot (r_i - 0.5))$, where $r_i$ is uniformly distributed on the interval $[0, 1]$.

In order to estimate $\Delta$, we can apply the Maximum Likelihood Method which leads to the following equation:

$$\frac{1}{1 + \left(\dfrac{\delta y^{(1)}}{\Delta}\right)^2} + \ldots + \frac{1}{1 + \left(\dfrac{\delta y^{(N)}}{\Delta}\right)^2} = \frac{N}{2}.$$

The left-hand side of this equation is an increasing function that is equal to $0 (< N/2)$ for $\Delta = 0$ and $> N/2$ for $\Delta = \max\left|\delta y^{(k)}\right|$; therefore the solution to this equation can be found by applying a bisection method to the interval $\left[0, \max\left|\delta y^{(k)}\right|\right]$.

*How many iterations do we need for the ideal estimate.* In (Trejo and Kreinovich, 2001; Kreinovich and Ferson, 2004), we found the number of iterations $N$ that would provide the desired accuracy (usually, 20% accuracy in estimating $\Delta$). The difference between the actual value $\Delta$ and its estimate $\widetilde{\Delta}$ is distributed, for large $N$, according to normal distribution, with 0 mean and standard deviation $\sigma_e = \Delta \cdot \sqrt{2/N}$. Thus, e.g., to get a 20% accuracy $0.2 \cdot \Delta$ with 95% certainty (corresponding to $2\sigma_e$), we need $N = 200$ runs.

After 200 runs, we can conclude that $\Delta \leq 1.2 \cdot \widetilde{\Delta}$ with certainty 95%.

Thus, the required number of calls to a model depends only on the desired accuracy $\varepsilon$ and not on $n$ – so for large $n$, these methods are much faster.

## 4. Applications: Brief Overview

We have applied the Cauchy deviate techniques to the following engineering examples:

- Environmental and power engineering: safety analysis of complex systems (Kreinovich and Ferson, 2004). In this example, $x_1, \ldots, x_n$ are the parameters of the system that are only known with interval uncertainty such as the thickness of the wall of the drum that contains radioactive waste. The program $f(x_1, \ldots, x_n)$ (usually given as a black box) describes how the desired parameters such as the radioactivity level at different places depend on $x_i$.

- Civil engineering: building safety. This example is similar to the models considered in (Muhanna and Mullen, 2001; Muhanna and Mullen, 2001a) and references therein. In this example, $x_1, \ldots, x_n$ are the loads on a structure for each of which we only know the tolerance intervals, and the elastic parameters of this structure which are only known with interval uncertainty. The program $f(x_1, \ldots, x_n)$ (often commercial and thus, given as a black box) is a finite-element model that describes how the stresses in the corresponding structure (e.g., building) depend on $x_i$.

- Petroleum and geotechnical engineering: estimating the uncertainty of the solution to the inverse problem caused by the measurement errors (Doser et al., 1998). In this example, $x_1, \ldots, x_n$ are the traveltimes of the seismic signals between the source and the sensor (and possibly other measurement results). The program $f(x_1, \ldots, x_n)$ solves the inverse problem, i.e., uses the traveltimes $x_i$ to estimate the density $y$ at different locations and at different depths. To be more accurate, the program reconstructs the speed of sound at different locations and at different depths, and then uses the known (approximate) relationship between the speed of sound and the density to reconstruct the desired density.

In all these cases, we got reasonable estimates:

- In the environmental and civil engineering applications, we got the same results as sensitivity analysis, but much faster.

– In geotechnical engineering, the dependence of the accuracy on the location and depth fits much better with the geophysicists' understanding than the previous accuracy results obtained under the assumption that all the measurement errors are independent and normally distributed.

## 5. Limitations of the Existing Cauchy Deviate Techniques and How These Limitations Can Be Overcome

### 5.1. Limitations

Cauchy deviate technique is based on the following assumptions:

– that the measurement errors are small, so we can safely linearize the problem;

– that we only have interval information about the uncertainty, and

– that we can actually call the program $f$ 200 times.

In real-life engineering problems, these assumptions may not be satisfied. In this section, we describe how we can modify the Cauchy deviate technique so as to overcome these limitations.

### 5.2. What If We Cannot Perform Many Iterations

*Problem.* In many real-life engineering problems, we do not have the possibility to run the program $f$ 200 times. In this case, we can still use the Cauchy deviates estimates with the available amount of $N$ iterations, but we need to come up with new formulas that translate the numerical estimate into the enclosure for $\Delta$.

*Case when $N$ is large enough.* In this case, the difference $\widetilde{\Delta} - \Delta$ is still Gaussian, we can conclude that $\Delta \le \widetilde{\Delta} \cdot \left(1 + k_0 \cdot \sqrt{\frac{2}{N}}\right)$ (where $k_0 = 2$), with certainty 95%. (If we want, e.g., 99.9% certainty, which corresponds to 3 sigma, then we should take $k_0 = 3$.)

Thus, e.g., for $N = 50$, we conclude that $\Delta \le 1.4 \cdot \widetilde{\Delta}$. This is not such a bad estimate.

*Case of very small number of iterations: idea.* When the number of iterations is even smaller, then we can no longer assume that the distribution of the error $\widetilde{\Delta} - \Delta$ is Gaussian. In this case, to find the bounds on $\Delta$ with, e.g., 95% certainty, we must perform numerical experiments.

The possibility of such experiments is caused by the fact that, as we have mentioned in the above description of the Cauchy deviates method, the distribution of the results $\delta y^{(k)}$ always follows the Cauchy distribution, no matter how small $N$ is.

So, to find out the confidence bounds on the Cauchy deviate estimates, it is sufficient to make experiments with the Cauchy distribution. The Cauchy distribution with a parameter $\Delta$ can be obtained by multiplying the Cauchy-distributed random variable with parameter $\Delta_0 = 1$ by the number $\Delta$. Thus, it is sufficient to test the method on Cauchy deviates with parameter 1.

For each $N$ and $\alpha$, we want to find $k(N,\alpha)$ for which $\Delta \le k(N,\alpha) \cdot \widetilde{\Delta}$ with certainty $1-\alpha$, i.e., for which $\widetilde{\Delta} \ge (1/k(N,\alpha)) \cdot \Delta$ with probability $1-\alpha$. Since we will be using Cauchy distribution with $\Delta = 1$, we must thus find $k(N,\alpha)$ for which $\widetilde{\Delta} \ge 1/k(N,\alpha)$ with probability $1-\alpha$.

To find such value, we do the following. We pick a large number of iterations $M$ (the relative accuracy of our estimate of $k(N,\alpha)$ will be $\approx 1/\sqrt{M}$). Then:

– For each $m$ from 1 to $M$:

  • we simulate Cauchy distribution (with parameter $\Delta_0 = 1$) $N$ times, producing $N$ numbers

$$\delta y_1^{(m)} = \tan(\pi \cdot (r_1^{(m)} - 0.5)), \ldots, \delta y_N^{(m)} = \tan(\pi \cdot (r_N^{(m)} - 0.5));$$

  • we then apply the above Maximum Likelihood Method to find $\widetilde{\Delta}_m$ as the solution to the following equation:

$$\frac{1}{1 + \left(\dfrac{\delta y_1^{(m)}}{\widetilde{\Delta}_m}\right)^2} + \ldots + \frac{1}{1 + \left(\dfrac{\delta y_N^{(m)}}{\widetilde{\Delta}_m}\right)^2} = \frac{N}{2};$$

  we solve this equation by applying a bisection method to the interval $\left[0, \max_i \left|\delta y_i^{(m)}\right|\right]$.

– After that, we sort the values $\widetilde{\Delta}_m$ into an increasing sequence

$$\widetilde{\Delta}_{(1)} \le \ldots \le \widetilde{\Delta}_{(M)}.$$

– We take the value $\widetilde{\Delta}_{(\alpha \cdot M)}$ for which the probability to be greater than this number is exactly $1-\alpha$, and estimate $k(N,\alpha)$ as $1/\widetilde{\Delta}_{(\alpha \cdot M)}$.

*Simulation results.* We wrote a C program that implements this algorithm. For $\alpha = 0.05$, the results of applying this program are:

– For $N = 20$, we get $k \approx 1.7$, which fits very well with the above Gaussian-based formula $k_{\text{norm}} \approx 1 + 2 \cdot \sqrt{2/20} \approx 1.7$.

– For $N = 10$, we get $k \approx 2.1$, which is slightly higher than the Gaussian-based formula $k_{\text{norm}} \approx 1 + 2 \cdot \sqrt{2/10} \approx 1.9$.

– For $N = 5$, we get $k \approx 5$, which is already much higher than the Gaussian-based value $k_{\rm norm} \approx 1 + 2 \cdot \sqrt{2/5} \approx 2.3$.

## 5.3. p-Boxes and Dempster-Shafer Knowledge Bases: An Idea

*Formulation of the problem.* In the previous sections, we described and analyzed different methods for estimating uncertainty in the cases when we have probabilistic or interval uncertainty in the inputs. What if the uncertainty in each input $x_i$ is characterized, e.g., by the Dempster-Shafer knowledge bases?

*Why this problem is difficult.* One reason why this problem is difficult is that it is not even clear how we can represent the DS knowledge base corresponding to the output.

Indeed, a DS knowledge base for each input variable $x_i$ means that we may have different intervals $[\underline{x}_i^{(k)}, \overline{x}_i^{(k)}]$, with different probabilities $p_i^{(k)}$. For each combination of intervals, $[\underline{x}_1^{(k_1)}, \overline{x}_1^{(k_1)}], \ldots, [\underline{x}_n^{(k_n)}, \overline{x}_n^{(k_n)}]$, we can use the known techniques to find the corresponding interval $[\underline{y}^{(k_1,\ldots,k_n)}, \overline{y}^{(k_1,\ldots,k_n)}]$ for the output. Since we know the probability $p_i^{(k_i)}$ of each interval $[\underline{x}_i^{(k_i)}, \overline{x}_i^{(k_i)}]$, and we assume that these probabilities are independent, we can compute the probability $p^{(k_1,\ldots,k_n)}$ of the corresponding output interval as the product $p^{(k_1,\ldots,k_n)} = p_1^{(k_1)} \cdot \ldots \cdot p_n^{(k_n)}$.

At first glance, this may sound like a reasonable solution to our problem, but in reality, this solution is not practical at all: even in the simplest case, when for each variable, we have two possible intervals, for $n = 50$ inputs, we will have an astronomical number of $2^{50} \approx 10^{15}$ output intervals $[\underline{y}^{(k_1,\ldots,k_n)}, \overline{y}^{(k_1,\ldots,k_n)}]$.

Thus, although the resulting uncertainty is still a DS uncertainty, we can no longer represent it as we represented the uncertainty for each input: by listing all the intervals and the corresponding probabilities.

Thus, not only it is not clear how to compute the resulting uncertainty, it is not even clear what exactly we want to compute.

*Can we use the fact that DS uncertainty is a generalization of interval uncertainty?* Our idea comes from the fact that the Dempster-Shafer uncertainty is a generalization of interval uncertainty, a generalization in which, for each inputs $x_i$, instead of a single interval $[\underline{x}_i, \overline{x}_i]$, we have several possible intervals $[\underline{x}_i^{(k)}, \overline{x}_i^{(k)}]$, with different probabilities $p_i^{(k)}$. For the interval uncertainty, in a realistic case when the black-box function is linearizable, we can use the Cauchy deviates method to estimate the interval uncertainty of the output. Let us see whether it is possible – at least, under some reasonable assumptions – to extend the Cauchy deviates method to the more general Dempster-Shafer case.

*Analysis.* The fact that the black-box function is linearizable means that we have $f(x_1, \ldots, x_n) = \widetilde{y} + \sum_{i=1}^{n} c_i \cdot (x_i - \widetilde{x}_i)$, where $\widetilde{y} \stackrel{\rm def}{=} f(\widetilde{x}_n, \ldots, \widetilde{x}_n)$ and for every $i$, $c_i$ denotes the

(unknown) value of the partial derivative $\partial f/\partial x_i$ of the black-box function $f(x_1, \ldots, x_n)$ with respect to $i$-th input $x_i$.

If we know the exact values $x_1, \ldots, x_n$ of all the inputs, then we can simply plug in the values $x_i$ and get the desired value.

If for each $i$, we know the interval $[x_i^{\rm mid} - \Delta_i, x_i^{\rm mid} + \Delta x_i]$, then, in the linearized case described above, the corresponding range of $y$ can be described by the interval $[y^{\rm mid} - \Delta, y^{\rm mid} + \Delta]$, where:

$$y^{\rm mid} = \widetilde{y} + \sum_{i=1}^{n} c_i \cdot (y_i^{\rm mid} - \widetilde{y}_i); \tag{2}$$

$$\Delta = \sum_{i=1}^{n} |c_i| \cdot \Delta_i. \tag{3}$$

In the Dempster-Shafer case, for each $i$, instead of a single pair $(y_i^{\rm mid}, \Delta_i)$, we have different pairs with different probabilities. Due to the formulas (2) and (3), the vector $(y^{\rm mid}, \Delta)$ is a linear combination of the vectors $(y_i^{\rm mid}, \Delta_i)$ corresponding to different inputs $x_i$.

If one of these vectors was prevailing, then we would have a single input (or a few dominating inputs), and there would be no need to consider the uncertainty in all $n$ inputs. Thus, the only case when this problem makes sense is when the contributions of all $n$ vectors is approximately of the same size (or at least the same order of magnitude). In this case, the vector $(y^{\rm mid}, \Delta)$ is a linear combination of $n$ independent vectors of approximately the same size.

This situation is exactly the case covered by the Central Limit Theorem, the case when in the limit $n \to \infty$, we have a normal 2-D distribution and hence, for sufficient large $n$, with a good approximation, we can assume that the pair $(y^{\rm mid}, \Delta)$ is normally distributed.

*Comment: strictly speaking, the distribution is almost normal but not exactly normal.* From the purely *theoretical* viewpoint, the distribution of the pairs $(y^{\rm mid}, \Delta)$ cannot be exactly normal, because:

- the interval half-width $\Delta$ is always non-negative, while
- for every normally distributed random variable, there is a non-zero probability that this value attains negative values.

However, *in practice*, every normal distribution with mean $\mu$ and standard deviation $\sigma$ is located within the interval $[\mu - k\cdot\sigma, \mu + k\cdot\sigma]$ with practically a certainty, i.e., with probability $\approx 1$:

- for $k = 3$, the probability to be outside the 3 sigma interval is $\approx 0.1\%$;
- for $k = 6$, the probability to be outside the 3 sigma interval is $\approx 10^{-6}\%$; etc.

Thus, if $\mu \geq k \cdot \sigma$, then, for all practical purposes, the half-width $\Delta$ is indeed always non-negative.

*Resulting idea.* It is therefore reasonable to conclude that for large $n$, the uncertainty in $y$ can be characterized as follows: we have different intervals $[y^{\rm mid} - \Delta, y^{\rm mid} + \Delta]$, and the probability of an interval is described by a 2-D normal distribution on the $(y^{\rm mid}, \Delta)$ plane.

To describe a 2-D normal distribution, it is sufficient to know 5 parameters: the means and standard deviations of both variables and the covariance (that describes their dependence).

*Discussion: are we abandoning the idea of non-parametric estimates?* At first glance, it may seem like we are abandoning our approach: we started with the idea of having non-parametric estimates, and we ended up with a 5-parametric family.

However, realistically, to exactly describe a generic distribution, we must use infinitely many parameters. In reality, we only have finitely many runs of the black-box function $f$ with reasonable accuracy, and based on their results, we can only estimate finitely many parameters anyway.

Even in the ideal case of Monte-Carlo tests, we need $N$ experiments to get a value of each parameter with an accuracy of $1/\sqrt{N}$. Thus, to get a reasonably low accuracy of 30% (everything worse makes it order-of-magnitude qualitative estimate), we need $\approx 10$ runs.

With 50 runs, we can therefore determine the values of no more than 5 parameters anyway. The above 5-parametric family is reasonable, its justification is very similar to the justification of the Gaussian distribution – the main workhorse of statistics – so why not use it?

*How can we determine the parameters of this model?* If we simply take the midpoints $x_i^{(k){\rm mid}}$ of the corresponding intervals in our simulations, then the resulting value $y^{(k)}$ are normally distributed, with the distribution corresponding to $y^{\rm mid}$. We can therefore estimate the mean and standard deviation of $y^{\rm mid}$ as simply the sample mean and the sample variance of the values $y^{(1)}, y^{(2)}, \ldots$

For $\Delta$, from the formula (3), we conclude that

$$E[\Delta] = \sum_{i=1}^{n} |c_i| \cdot E[\Delta_i] \tag{4}$$

and

$$\sigma[\Delta] = \sqrt{\sum_{i=1}^{n} |c_i|^2 \cdot \sigma^2[\Delta_i]}. \tag{5}$$

Due to the formula (4), we can use the Cauchy deviates technique to estimate $E[\Delta]$ if for each input $x_i$, we use the average half-width

$$E[\Delta_i] = p_i^{(1)} \cdot \Delta_i^{(1)} + p_i^{(1)} \cdot \Delta_i^{(1)} + \ldots$$

of the corresponding interval.

Due to the fact that $|c_i|^2 = c_i^2$, the formula (5) means that we can compute $\sigma[\Delta]$ by using the standard Monte-Carlo simulation technique: namely, we simulate $\delta x_i$ to be normally distributed with 0 mean and standard deviation $\sigma[\Delta_i]$, then the resulting value of $\delta y =$

$\sum c_i \cdot \delta x_i$ is also normally distributed, with the standard deviation equal to (5). We can thus estimate (5) as a sample variance of the corresponding simulated values $\delta y^{(k)}$.

We thus know how to estimate 4 of 5 parameters that describe the desired uncertainty. The only remaining problem is how to estimate the covariance between $y^{\rm mid}$ and $\Delta$. For this, we propose the following idea.

The non-zero covariance means, in particular, that the conditional average $E[\Delta \,|\, y^{\rm mid} \le E[y^{\rm mid}]]$ of $\Delta$ over the cases when $y^{\rm mid}$ is smaller than its average $E[y^{\rm mid}]$ is different from the conditional average $E[\Delta \,|\, y^{\rm mid} \ge E[y^{\rm mid}]]$ of $\Delta$ over the cases when $y^{\rm mid}$ is larger than its average $E[y^{\rm mid}]$. From the difference between these two conditional averages, we can determine the desired value of the covariance.

To compute the conditional averages, we can use the Cauchy deviates idea. Namely, at each simulation, for each variable $x_i$, we select one of the intervals $[\underline{x}_i^{(k)}, \overline{x}_i^{(k)}]$ with the corresponding probability $p_i^{(k)}$, and we apply the black box function $f$ to the centers of the corresponding intervals, to get the result $y^{\rm mid}$. We then apply the Cauchy techniques with the corresponding intervals and get the value distributed according to the Cauchy distribution with the width corresponding to selected intervals for $x_i$.

The main difference between what we propose to do here and the previously described Cauchy deviates methods is the following:

- in the previously described Cauchy deviates method, we combine all the results of Cauchy simulation into a single sample, and we then compute the parameter $\Delta$ based on this sample;
- in the proposed methods, we separate the results of Cauchy simulation into two different samples:
  - a sample containing all the cases in which $y^{\rm mid} \le E[y^{\rm mid}]$, and
  - a sample containing all the cases in which $y^{\rm mid} \ge E[y^{\rm mid}]$.

In the previous described approach, in all simulations, we had *the same* interval width, so the results of the simulation belong to the same Cauchy distribution. In the new method, we have *different* widths with different probabilities, so the resulting distribution is a combination of different Cauchy distributions, with different probabilities.

For each sample, we can safely assume that the distribution of the width $\Delta$ is a Gaussian distribution, with mean $\mu$ and standard deviation $\sigma$. Thus, our sample corresponds to the combination in which the Cauchy distribution with parameter $\Delta$ occurs with the Gaussian probability density $\dfrac{1}{\sqrt{2 \cdot \pi} \cdot \sigma} \cdot \exp\left(-\dfrac{(\Delta - \mu)^2}{2\sigma^2}\right)$. Cauchy-distributed random variable $\xi$ with the parameter $\Delta$ can be described by its characteristic function $E[\exp(\mathrm{i} \cdot \omega\xi)] = \exp(-|\omega| \cdot \Delta)$. Thus, the above-described probabilistic combination of Cauchy distributions can be described by the corresponding probabilistic combination of these characteristic functions:

$$E[\exp(\mathrm{i} \cdot \omega \cdot \xi)] = \int \frac{1}{\sqrt{2 \cdot \pi} \cdot \sigma} \cdot \exp\left(-\frac{\Delta - \mu}{2\sigma^2}\right) \cdot \exp(-|\omega| \cdot \Delta)\, \mathrm{d}\Delta. \tag{6}$$

By separating the full square in the integrated expression, one can show that this integral is equal to:

$$\exp\left(\frac{1}{2}\cdot\sigma^2\cdot\omega^2-\mu\cdot|\omega|\right). \tag{7}$$

We can estimate the characteristic function by its sample value

$$E[\exp(\mathrm{i}\cdot\omega\cdot\xi)]\approx\frac{1}{N}\cdot\sum_{k=1}^{N}\cos(\omega\cdot y^{(k)})$$

(Since the expression (7) is real, it makes sense to only consider the real part of $\exp(\mathrm{i}\cdot\omega\cdot\xi)$, i.e., $\cos(\omega\cdot\xi)$.)

So, we arrive at the following algorithm for computing $\mu$ and $\sigma$ from the sample values $y^{(1)},\ldots,y^{(N)}$:

- for different real values $\omega_1,\ldots,\omega_k>0$, compute $l(\omega_k)\stackrel{\text{def}}{=}-\ln(c(\omega_k))$, where $c(\omega_k)\stackrel{\text{def}}{=}\frac{1}{N}\cdot\sum\limits_{k=1}^{N}\cos(\omega\cdot y^{(k)})$;
- use the Least Squares Method to find the values $\mu$ and $\sigma$ for which

$$\mu\cdot\omega_k-\frac{1}{2}\sigma^2\cdot\omega_k^2\approx l(\omega_k).$$

The resulting value $\mu$ is the average $\Delta$.

Thus, when we repeat this algorithm for both samples, we get the two desired conditional averages of $\Delta$ – from which we can then compute the covariance.

*What about p-boxes?* It is known that a p-box can be described as a DS knowledge base. Namely, a p-box $[\underline{F}(t),\overline{F}(t)]$ is a generalization of a cdf function $F(t)$. A cdf function can be represented by an explicit formula, or it can be represented if we list, for uniformly spaced levels $p=0,\Delta p,2\cdot\Delta p,\ldots,1.0$ (e.g., for $p=0,0.1,0.2,\ldots,0.9.1.0$), the corresponding quantiles, i.e., values $t$ for which $F(t)=p$. In mathematical terms, quantiles are the values of the inverse function $f(p)=F^{-1}(t)$ at equally spaced values $p$.

The variable with a probability distribution $F(t)$ can be approximately described as follows: we have the values $f(0)$, $f(\Delta p)$, etc., with equal probability $\Delta p$.

Similarly, a p-box can be alternatively represented by listing, for each $p$, the interval $[\underline{f}(p),\overline{f}(p)]$ of the possible quantile values. Here:

- the function $\underline{f}(p)$ is an inverse function to $\overline{F}(t)$, and
- the function $\overline{f}(p)$ is an inverse function to $\underline{F}(t)$.

This description, in effect, underlies some algorithms for processing p-boxes that are implement in RAMAS software (Ferson, 2002).

Because of this description, we can interpret the p-box as the DS knowledge base, in which, with equal probability $\Delta p$, we can have intervals $[\underline{f}(0), \overline{f}(0)]$, $[\underline{f}(\Delta p), \overline{f}(\Delta p)]$, etc.

Thus, whatever method we have for DS knowledge bases, we can apply it to p-boxes as well.

*How can we describe the resulting p-boxes?* We have just mentioned that, in principle, we can interpret each p-box as a DS knowledge base, and apply the above DS-based method to describe th uncertainty of the output. The result, however, is a DS knowledge base. How can we describe the corresponding "Gaussian" DS knowledge base as a p-box?

It is known that for a DS knowledge base, i.e., for a probabilistic distribution on the set of intervals $[\underline{x}, \overline{x}]$:

- The probability $F(t) = \mathrm{Prob}(X \le t)$ attains its largest possible value $\overline{F}(t)$ if for each interval, we take the smallest possible value $\underline{x}$.
- Similarly, the probability $F(t) = \mathrm{Prob}(X \le t)$ attains its smallest possible value $\underline{F}(t)$ if for each interval, we take the largest possible value $\overline{x}$.

Thus:

- $\overline{F}(t)$ is a probability distribution for the lower endpoints $y^{\min} - \Delta$, and
- $\underline{F}(t)$ is a probability distribution for the upper endpoints $y^{\min} + \Delta$ of the corresponding intervals.

Since the 2-D distribution of the pairs $(y^{\mathrm{mid}}, \Delta)$ is Gaussian, the distributions of both linear combinations $y^{\min} - \Delta$ and $y^{\min} + \Delta$ are Gaussian as well.

Therefore, as a result of this procedure, we get a p-box $[\underline{F}(t), \overline{F}(t)]$ for which both bounds $\underline{F}(t)$ and $\overline{F}(t)$ correspond to Gaussian distributions.

*Comment: strictly speaking, the distributions are almost normal but not exactly normal.* Let us denote the cdf of the standard Gaussian distribution, with 0 mean and standard deviation 1 by $F_0(t)$. Then, an arbitrary Gaussian distribution, with mean $\mu$ and standard deviation $\sigma$, can be described as $F(t) = F_0((t-\mu)/\sigma)$. In particular, if we denote:

- the mean and the standard deviations of the Gaussian distribution $\underline{F}(t)$ by $\underline{\mu}$ and $\underline{\sigma}$, and
- the mean and the standard deviations of the Gaussian distribution $\overline{F}(t)$ by $\overline{\mu}$ and $\overline{\sigma}$,

then we conclude that $\underline{F}(t) = F_0((t-\underline{\mu})/\underline{\sigma})$ and $\overline{F}(t) = F_0((t-\overline{\mu})/\overline{\sigma})$.

From the theoretical viewpoint, for thus defined functions $\underline{F}(t)$ and $\overline{F}(t)$, we cannot always have $\underline{F}(t) \le \overline{F}(t)$, because, due to monotonicity of $F_0(t)$, this would be equivalent to $\dfrac{t-\underline{\mu}}{\underline{\sigma}} \le \dfrac{t-\overline{\mu}}{\overline{\sigma}}$ for all $t$, i.e., to one straight line being always below the other – but this is only possible when they are parallel.

However, as we have mentioned while describing the similar situation with the DS knowledge bases, in practice, we can have this inequality if we ignore the values $t$ for which $F_0(t)$ is very small – and thus, not practically possible.

Alternatively, we can assume that the inequality $\underline{F}(t) \le \overline{F}(t)$ holds for all $t$ – but the distributions $\underline{F}(t)$ and $\overline{F}(t)$ are only approximately – but not exactly – normal.

*What if we have different types of uncertainty for different inputs?* If we have different types of uncertainty for different inputs, we can transform them to p-boxes (Ferson, 2002) – hence, to DS knowledge bases – and use a similar approach.

## 5.4. Cauchy Deviates Methods for Non-Linear Functions $f(x_1, \ldots, x_n)$

*Case of weak non-linearity.* In some cases, we cannot reasonably approximate $f$ by a linear expression on the entire box, but we can divide the box into a few subboxes on each of which $f$ is approximately linear. For example, if the dependence of $f$ on one of the variables $x_i$ is strongly non-linear, then we can divide the interval $[\underline{x}_i, \overline{x}_i]$ of possible values of this variable into two (or more) subintervals, e.g., $[\underline{x}_i, x_i^{\rm mid}]$ and $[x_i^{\rm mid}, \overline{x}_i]$, and consider the corresponding subboxes

$$[\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_{i-1}, \overline{x}_{i-1}] \times [\underline{x}_i, x_i^{\rm mid}] \times [\underline{x}_{i+1}, \overline{x}_{i+1}] \times \ldots \times [\underline{x}_n, \overline{x}_n]$$

and

$$[\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_{i-1}, \overline{x}_{i-1}] \times [x_i^{\rm mid}, \overline{x}_i] \times [\underline{x}_{i+1}, \overline{x}_{i+1}] \times \ldots \times [\underline{x}_n, \overline{x}_n].$$

By using the Cauchy deviates methods, we compute the range of $f$ over each of these subboxes, and then take the union of the resulting range intervals.

*Quadratic case.* Linearization technique is based on the assumption that the measurement errors $\Delta x_i$ and/or uncertainties are so small that we can safely ignore terms that are quadratic (or of higher order) in $\Delta x_i$. If the measurement errors are larger, so that we can no longer reasonably approximate $f$ by a linear expression, a natural next step is to take quadratic terms into consideration while still ignoring cubic and higher-order terms: $f(x_1^{\rm mid} + \delta x_1, \ldots, x_n^{\rm mid} + \delta x_n) \approx y^{\rm mid} + \delta y$, where

$$\delta y \stackrel{\rm def}{=} \sum_{i=1}^{n} c_i \cdot \delta x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot \delta x_i \cdot \delta x_j, \tag{8}$$

where $c_i$ are the same as for the linearized case and $c_{ij} \stackrel{\rm def}{=} \frac{1}{2} \cdot \frac{\partial^2 f}{\partial x_i \partial x_j}$.

In general, computing the exact bound for a quadratic function of $n$ variables in case of interval uncertainty is an NP-hard problem (Vavasis, 1991; Kreinovich et al., 1997). Luckily, in many practical case, the dependence of $f$ on $x_i$ is monotonic (see, e.g., (Lakeyev and Kreinovich, 1995)), so we can use, e.g., the above-described sensitivity analysis technique.

The problem with the sensitivity analysis technique, as we have mentioned, is that this technique requires $n$ calls to the program $f$, which for large $n$ may be too long. It is therefore desirable to modify the Cauchy deviate technique so that it can be used for quadratic functions as well.

*Analysis of the problem.* We consider the case when the function $f(x_1, \ldots, x_n)$ is monotonic in each variable $x_i$.

If the function $f$ is increasing in $x_i$, then the derivative $\dfrac{\partial f}{\partial x_i}$ is always positive; in particular, it is positive at the central point $(x_1^{\rm mid}, \ldots, x_n^{\rm mid})$, so $c_i > 0$. In this case, the maximum of $f$ is attained when $\delta x_i = \Delta_i$ and $x_i = \overline{x}_i = x_i^{\rm mid} + \Delta_i$.

Similarly, when the function $f$ is decreasing in $f$, then $c_i < 0$ and the maximum is attained when $\delta x_i = -\Delta_i$ and $x_i = x_i^{\rm mid} - \Delta_i$. In both cases, the largest possible value $\Delta^+$ of the difference $\delta y$ is attained when for every $i$, we have $\delta x_i = \varepsilon_i \cdot \Delta_i$, where $\varepsilon_i \stackrel{\rm def}{=} {\rm sign}(c_i)$. Substituting this expression for $\delta x_i$ into the above formula for $\delta y$, we conclude that

$$\Delta^+ = \sum_{i=1}^{n} c_i \cdot \varepsilon_i \cdot \Delta_i + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \cdot \varepsilon_i \cdot \varepsilon_j \cdot \Delta_i \cdot \Delta_j =$$

$$\sum_{i=1}^{n} |c_i| \cdot \Delta_i + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \cdot \varepsilon_i \cdot \varepsilon_j \cdot \Delta_i \cdot \Delta_j. \tag{9}$$

Similarly, the smallest possible value $\delta y_{\rm min}$ of $\delta y$ is attained when $\delta x_i = -\varepsilon_i \cdot \Delta_i$, hence $\Delta^- \stackrel{\rm def}{=} |\delta y_{\rm min}|$ is equal to:

$$\Delta^- = \sum_{i=1}^{n} |c_i| \cdot \Delta_i - \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \cdot \varepsilon_i \cdot \varepsilon_j \cdot \Delta_i \cdot \Delta_j. \tag{10}$$

We would like to use a Cauchy-type method to find the bounds (9) and (10). For this, we consider, for every pairs of vectors $z = (z_1, \ldots, z_n)$ and $t = (t_1, \ldots, t_n)$, the following auxiliary expression:

$$\frac{f(x^{\rm mid} + z + t) - f(x^{\rm mid} + z - t)}{2} =$$

$$\frac{1}{2} \cdot f(x_1^{\rm mid} + z_1 + t_1, \ldots, x_n^{\rm mid} + z_n + t_n) -$$

$$\frac{1}{2} \cdot f(x_1^{\rm mid} + z_1 - t_1, \ldots, x_n^{\rm mid} + z_n - t_n). \tag{11}$$

Substituting $\delta x_i = z_i + t_i$ into the formula (8), we conclude that

$$f(x^{\rm mid} + z + t) = y^{\rm mid} + \sum_{\imath=1}^{n} c_i \cdot (z_i + t_i) + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \cdot (z_i + t_i) \cdot (z_j + t_j), \tag{12}$$

and similarly,

$$f(x^{\rm mid}+z-t)=y^{\rm mid}+\sum_{\imath=1}^{n}c_i\cdot(z_i-t_i)+\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}\cdot(z_i-t_i)\cdot(z_j-t_j),\qquad(13)$$

hence

$$\frac{1}{2}\cdot(f(x^{\rm mid}+z+t)-f(x^{\rm mid}+z-t))=\sum_{i=1}^{n}\left(c_i+2\cdot\sum_{j=1}^{n}c_{ij}\cdot z_j\right)\cdot t_i.\qquad(14)$$

This expression is linear with respect to $t_1,\ldots,t_n$. Therefore, we can use the existing linear Cauchy algorithm in order to find bounds for this expression as a function of $t_i$ when $|t_i|\le\Delta_i$.

Let $g(z)=g(z_1,\ldots,z_n)$ denote the result of applying the linear Cauchy method to the expression (14) considered as as a function of $t$; then,

$$g(z)=\sum_{i=1}^{n}\left|c_i+2\cdot\sum_{j=1}^{n}c_{ij}\cdot z_j\right|\cdot\Delta_i.$$

Since the function $f$ is monotonic on the box, its derivative $\dfrac{\partial f}{\partial x_i}$ has the same sign at all the points from the box. Hence, the sign of the derivative $c_i+2\cdot\sum\limits_{j=1}^{n}c_{ij}\cdot z_j$ at the point

$$x^{\rm mid}+z=(x_1^{\rm mid}+z_1,\ldots,x_n^{\rm mid}+z_n)$$

is the same as the sign $\varepsilon_i$ of the derivative $c_i$ at the midpoint $x^{\rm mid}=(x_1^{\rm mid},\ldots,x_n^{\rm mid})$ of the box. Since $|E|=\mathrm{sign}(E)\cdot E$ for every expression $E$, we thus conclude that

$$\left|c_i+2\cdot\sum_{j=1}^{n}c_{ij}\cdot z_j\right|=\varepsilon_i\cdot\left(c_i+2\cdot\sum_{j=1}^{n}c_{ij}\cdot z_j\right),$$

hence

$$g(z)=\sum_{i=1}^{n}|c_i|\cdot\Delta_i+2\cdot\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}\cdot\varepsilon_i\cdot\Delta_i\cdot z_j.\qquad(15)$$

In particular, for $z=0=(0,\ldots,0)$, we get $g(0)=\sum\limits_{i=1}^{n}|c_i|\cdot\Delta_i$.

¿From (12) and (14), we conclude that

$$f(x^{\rm mid}+z)-f(x^{\rm mid}-z)=2\cdot\sum_{i=1}^{n}c_i\cdot z_i.$$

We can therefore construct a new function $h(z)$ as follows:

$$h(z)\stackrel{\rm def}{=}\frac{1}{2}\cdot(g(z)-g(0)+f(x^{\rm mid}+z)-f(x^{\rm mid}-z))=$$

$$\sum_{i=1}^{n} c_i \cdot z_i + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \cdot \varepsilon_j \cdot \Delta_j \cdot z_i. \tag{16}$$

This expression is linear with respect to $z_1, \ldots, z_n$. Therefore, we can use the existing linear Cauchy algorithm in order to find bounds for this expression as a function of $z_i$ when $|z_i| \le \Delta_i$. As a result, we get the estimate

$$H \stackrel{\text{def}}{=} \sum_{i=1}^{n} \left| c_i + \sum_{j=1}^{n} c_{ij} \cdot \varepsilon_j \cdot \Delta_j \right| \cdot \Delta_i.$$

Since the function $f$ is monotonic on the box, its derivative $\dfrac{\partial f}{\partial x_i}$ has the same sign at all the points from the box. Hence, the sign of the derivative $c_i + \sum\limits_{j=1}^{n} c_{ij} \cdot \varepsilon_j \cdot \Delta_j$ at the point

$$(x_1^{\text{mid}} + \frac{1}{2} \cdot \varepsilon_1 \cdot \Delta_1, \ldots, x_n^{\text{mid}} + \frac{1}{2} \cdot \varepsilon_n \cdot \Delta_n)$$

is the same as the sign $\varepsilon_i$ of the derivative $c_i$ at the midpoint $x^{\text{mid}} = (x_1^{\text{mid}}, \ldots, x_n^{\text{mid}})$ of the box. Since $|E| = \text{sign}(E) \cdot E$ for every expression $E$, we thus conclude that

$$\left| c_i + \sum_{j=1}^{n} c_{ij} \cdot \varepsilon_j \cdot \Delta_j \right| = \varepsilon_i \cdot \left( c_i + \sum_{j=1}^{n} c_{ij} \cdot \varepsilon_j \cdot \Delta_j \right),$$

hence

$$H = \sum_{i=1}^{n} |c_i| \cdot \Delta_i + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \cdot \varepsilon_i \cdot \Delta_i \cdot \varepsilon_j \cdot \Delta_j,$$

which is exactly the above expression for $\Delta^+$. The value $\Delta^-$ can now be computed as $2g(0) - \Delta^+$.

We thus arrive at the following algorithm for computing $\Delta^+$ and $\Delta^-$.

*Algorithm.* As an auxiliary step, we first design an algorithm that, given a vector $z = (z_1, \ldots, x_n)$, computes $g(z)$. This algorithm consists of applying the linear Cauchy deviate method to the auxiliary function $t \to \dfrac{1}{2} \cdot (f(x^{\text{mid}} + z + t) - f(x^{\text{mid}} + z - t))$ and the values $t_i \in [-\Delta_i, \Delta_i]$. The linear Cauchy methods requires $N$ calls to the auxiliary function (where $N$ depends on the desired accuracy), and each call to the auxiliary function means 2 calls to the program $f$; so, overall, we need $2N$ calls to $f$.

The algorithm itself works as follows:

- First, we apply the algorithm $g(z)$ to the vector $0 = (0, \ldots, 0)$, thus computing the value $g(0)$.
- Second, we apply the linear Cauchy deviate method to the auxiliary function $h(z) = \dfrac{1}{2} \cdot (g(z) - g(0) + f(x^{\text{mid}} + z) - f(x^{\text{mid}} - z))$; the result is the desired value $\Delta^+$.

– Finally, we compute $\Delta^-$ as $2g(0) - \Delta^+$.

What is the computational complexity of this algorithm? How many calls to the program $f$ did we make?

– In the first stage, we made a single call to $g$, so this stage requires $2N$ calls to $f$.

– The second stage requires $N$ calls to $h$. Each call to $h$ means 2 calls to $f$ and 1 call to $g$; each call to $g$, as we have mentioned, requires $2N$ calls to $f$. Thus, overall, each call to $h$ requires $2 + 2N$ calls to $f$; in total, the second stage requires $N \cdot (2 + 2N)$ calls to $f$.

– On the final stage, there are no calls to $f$.

So, overall, this algorithm requires $2N + n \cdot (2 + 2N) = 2N \cdot (N + 2)$ calls to $f$.

For example, if we want the 20% accuracy on average, we need $N = 50$, so this algorithm would require $\approx 5000$ calls to $f$. Thus, when we have $n \ll 5000$ variables, it is faster to use the sensitivity analysis method, but when we have $n \gg 5000$ variables, this Monte-Carlo-type method is faster.

If we want 20% accuracy with certainty 95%, then we need $N = 200$. In this case, the above quadratic method requires $\approx 80000$ calls to $f$, so this method is faster only if we have $n \gg 80000$ variables.

## Acknowledgements

This work was supported in part by NASA under cooperative agreement NCC5-209, by the Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-1-0365, by NSF grants EAR-0112968, EAR-0225670, and EIA-0321328, by the Army Research Laboratories grant DATM-05-02-C-0046, and by a research grant from Sandia National Laboratories as part of the Department of Energy Accelerated Strategic Computing Initiative (ASCI).

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

## References

Doser, D. I., K. D. Crain, M. R. Baker, V. Kreinovich, and M. C. Gerstenberger Estimating uncertainties for geophysical tomography, *Reliable Computing*, 1998, 4(3)241–268.

Ferson, S. *RAMAS Risk Calc 4.0: Risk Assessment with Uncertain Numbers*, CRC Press, Boca Raton, Florida, 2002.

Kreinovich, V., and S. Ferson, A New Cauchy-Based Black-Box Technique for Uncertainty in Risk Analysis, *Reliability Engineering and Systems Safety*, 2004, 85(1–3): 267-279.

Kreinovich, V., A. Lakeyev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1997.

Lakeyev, A. V., and V. Kreinovich, If Input Intervals Are Small Enough, Then Interval Computations Are Almost Always Easy, *Reliable Computing*, 1995, Supplement (Extended Abstracts of APIC'95: International Workshop on Applications of Interval Computations, El Paso, TX, Febr. 23–25, 1995), pp. 134–139.

Jaulin L., M. Keiffer, O. Didrit, and E. Walter, *Applied Interval Analysis*, Springer-Verlag, Berlin, 2001.

Kearfott, R. B., *Rigorous Global Search: Continuous Problems*, Kluwer, Dordrecht, 1996.

Kearfott R. B., and V. Kreinovich, editors, *Applications of Interval Computations*, Kluwer, Dordrecht, 1996.

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

Muhanna, R. L., and R. L. Mullen, Uncertainty in mechanics problems – interval-based approach, *ASCE Journal of Engineering Mechanics*, 2001, 127(6):557–566.

Muhanna, R. L., and R. L. Mullen, Treatment in geometric tolerances in finite element analysis, *International Journal of Advanced Manufacturing Systems*, 2001, 127(6):557–566.

Trejo, R., and V. Kreinovich, Error Estimations for Indirect Measurements, In: S. Rajasekaran et al., editors, *Handbook on Randomized Computing*, Kluwer, 2001, pp. 673–729.

Vavasis, S. A., *Nonlinear Optimization: Complexity Issues*, Oxford University Press, N.Y., 1991.

Wadsworth, H. M. Jr., editor, *Handbook of statistical methods for engineers and scientists*, McGraw-Hill Publishing Co., N.Y., 1990.

# Optimal Multilevel System Design under Uncertainty

M. Kokkolaras (mk@umich.edu)
*Department of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan*

Z.P. Mourelatos (mourelat@oakland.edu)
*Department of Mechanical Engineering, Oakland University, Rochester, Michigan*

P.Y. Papalambros (pyp@umich.edu)
*Department of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan*

**Abstract.** In this paper we consider hierarchically decomposed multilevel systems, and extend previous deterministic methodologies for optimal and consistent design of such systems to account for the presence of uncertainties. Specifically, we use the probabilistic formulation of the analytical target cascading process to solve the multilevel problem, and use an advanced mean value-based technique to estimate uncertainty propagation. The proposed methodology is demonstrated by means of a simple yet illustrative optimal bi-level system design example.

**Keywords:** design optimization, hierarchical multilevel systems, analytical target cascading, design under uncertainty, propagation of uncertainties

## 1. Introduction

Optimal design of complex engineering systems can be accomplished only by decomposition. The system is partitioned into subsystems, the subsystems are partitioned into components, the components into parts, and so on. This decomposition process results in a multilevel hierarchy of elements that comprise the system.

Deterministic optimization approaches assume that complete information of the problem is available, and that design decisions can be implemented. These assumptions imply that optimization results are as good (and therefore useful) as the design and simulation/analysis models used to obtain them, and that they are meaningful only if they can be realized exactly.

In reality, these assumptions do not hold. We are rarely in a position to represent a physical system without using approximations, have complete knowledge on all of its parameters, or control the design variables with high accuracy. It is therefore necessary to treat all quantities associated with uncertainty as stochastic.

In this paper, we consider hierarchically decomposed multilevel systems, and we extend deterministic methodologies for optimal and consistent design of such systems to account for the presence of uncertainties. Our objective is to introduce the concept of uncertainty, model its propagation through the multilevel hierarchy, set the ground for the application of "single-element" optimization under uncertainty methods in multilevel systems, and identify needs for future research.

To the best of our knowledge, no research work on addressing the presence of uncertainties in hierarchically decomposed multilevel systems has been reported in the literature.

However, there is ongoing work to take uncertainties into consideration in the multidisciplinary optimization (MDO) framework [1–10]. Most of these references utilize a simple first-order Taylor expansion to calculate the mean and variance of the response in robust multidisciplinary design or use "worst case" concepts based on first-order sensitivity to evaluate the performance range of a multidisciplinary system.

Although the calculation of the response mean and variance using first-order sensitivity may be adequate for robustness calculations, it does not provide enough statistical information to consider design feasibility under uncertainty. As will be illustrated in this paper, probabilistic representation of the constraints requires complete probabilistic distributions of the system output.

Reliability analysis using probabilistic distributions has been used in MDO [11–13]. Reliability analysis introduces an additional iteration loop resulting in coupled optimization problems that are computationally expensive. Response surfaces have been used to reduce the computational effort [1]. Decoupled reliability and optimization procedures in an MDO framework have been also proposed using approximate probabilistic constraint representations [12]. In general, a double-loop optimization process exists in reliability-based MDO analysis, which repeatedly calls expensive system-level multidisciplinary analyses. A single-loop collaborative reliability analysis method has been recently proposed in [11]. A Most Probable Point (MPP) reliability analysis method is combined with the collaborative disciplinary analyses to automatically satisfy the interdisciplinary consistency in reliability analyses. A single reliability optimization loop uses equality constraints to enforce disciplinary compatibility. Despite the use of a single optimization loop, it is a computationally expensive, "all-at-once" procedure due to the presence of the equality discipline constraints.

It is important to differentiate our research work from that related to multidisciplinary design optimization (MDO). MDO approaches are non-hierarchical in the sense that the optimal design problems are not decomposed according to disciplines into multilevel hierarchies. Discipline outputs are inputs to other disciplines and *vice versa*. This is the significant difference between MDO and our work. In hierarchically decomposed multilevel systems outputs of lower-level elements are inputs to higher-level elements, but not *vice versa*.

The paper is organized as follows. In the next section we present a methodology for optimal design of hierarchical multilevel systems, and extend its formulation to account for uncertainties. In Section 3 we address the issue of modeling uncertainty propagation in multilevel hierarchies and present some analytical examples. A simple yet illustrative simulation-based example is used in Section 4 to demonstrate our methodology for hierarchical multilevel system design. Finally, concluding remarks are summarized in Section 5.

## 2. Optimal Design of Hierarchically Decomposed Multilevel Systems

Our framework for hierarchical multilevel system optimization under uncertainty is based on analytical target cascading (ATC). In this section we first review the deterministic formulation of ATC, and then we present its extension to account for uncertainties.

### 2.1. Deterministic Formulation

ATC is a mathematical methodology for translating ("cascading") overall system design targets to element specifications based on a hierarchical multilevel decomposition [14–16]. The objective is to assess relations and identify possible trade-offs among elements early in the design development process, and to determine specifications that yield consistent system design with minimized deviation from design targets.

The ATC process is proven to be convergent when using a specific class of coordination strategies [17], and has been successfully applied to a variety of optimal design problems, e.g., [18–21].

We refer the reader to the above references for a detailed description of ATC. Here, we will briefly present the concept and the general mathematical formulation. In ATC a minimum deviation optimization problem is formulated and solved for each element in the multilevel hierarchy that reflects the decomposed optimal system design problem, *cf.* Figure 1. Therefore, responses of lower-level elements are inputs into higher-level elements.

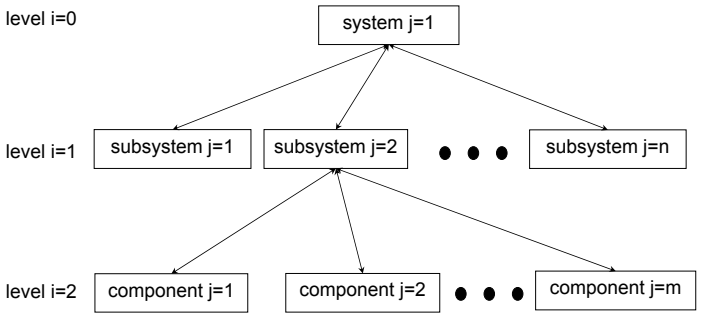



*Figure 1.* Example of hierarchically decomposed multilevel system

The ATC process aims at minimizing the gap between what higher-level elements "want" and what lower-level elements "can". If design variables are shared among some elements at the same level, their consistency is coordinated by their parent element at the level above.

The mathematical formulation of problem $p_{ij}$, where $i$ and $j$ denote level and element, respectively, is

$$\begin{aligned} \min_{\tilde{\mathbf{x}}_{ij},\epsilon^r_{ij},\epsilon^y_{ij}} \quad & \|\mathbf{r}_{ij} - \mathbf{r}^u_{ij}\|_2^2 + \|\mathbf{y}_{ij} - \mathbf{y}^u_{ij}\|_2^2 + \epsilon^r_{ij} + \epsilon^y_{ij} \\ \text{subject to} \quad & \textstyle\sum_{k=1}^{n_{ij}} \|\mathbf{r}_{(i+1)k} - \mathbf{r}^l_{(i+1)k}\|_2^2 \le \epsilon^r_{ij} \end{aligned} \tag{1}$$

$$\begin{aligned} \textstyle\sum_{k=1}^{n_{ij}} \|\mathbf{y}_{(i+1)k} - \mathbf{y}^l_{(i+1)k}\|_2^2 \leq \epsilon^y_{ij} \\ \mathbf{g}_{ij}(\mathbf{r}_{(i+1)1}, \ldots, \mathbf{r}_{(i+1)n_{ij}}, \mathbf{x}_{ij}, \mathbf{y}_{ij}) \leq \mathbf{0} \\ \mathbf{h}_{ij}(\mathbf{r}_{(i+1)1}, \ldots, \mathbf{r}_{(i+1)n_{ij}}, \mathbf{x}_{ij}, \mathbf{y}_{ij}) = \mathbf{0} \\ \text{with} \quad \mathbf{r}_{ij} = \mathbf{f}_{ij}(\mathbf{r}_{(i+1)1}, \ldots, \mathbf{r}_{(i+1)n_{ij}}, \mathbf{x}_{ij}, \mathbf{y}_{ij}), \end{aligned}$$

where the vector of optimization variables $\tilde{\mathbf{x}}_{ij}$ consists of $(n_{ij})$ children response design variables $\mathbf{r}_{(i+1)1}, \ldots, \mathbf{r}_{(i+1)n_{ij}}$, local design variables $\mathbf{x}_{ij}$, local shared design variables $\mathbf{y}_{ij}$ (i.e., design variables that this element shares with other elements at the same level), and coordinating variables for the shared design variables of the children $\mathbf{y}_{(i+1)1}, \ldots, \mathbf{y}_{(i+1)n_{ij}}$, and where $\mathbf{g}_{ij}$ and $\mathbf{h}_{ij}$ denote local design inequality and equality constraints, respectively. Tolerance optimization variables $\epsilon^r$ and $\epsilon^y$ are introduced to coordinate responses and shared variables, respectively. Superscripts $u$ $(l)$ are used to denote response and shared variable values that have been obtained at the parent (children) problem(s), and have been cascaded down (passed up) as design targets (consistency parameters), *cf.* Figure 2.

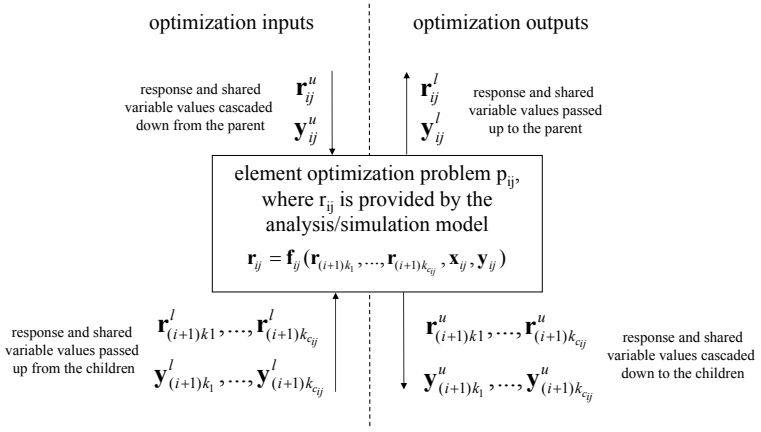



*Figure 2.* ATC information flow at element $j$ of level $i$

Assuming that all the parameters have been updated using the solutions obtained at the parent- and children-problems, Problem (1) is solved to update the parameters of the parent- and children-problems. This process is repeated until the tolerance optimization variables in all problems cannot be reduced any further.

## 2.2. Non-deterministic Formulations

In this section, the ATC formulation is modified to account for uncertainties. Stochastic quantities are represented by random variables and parameters (denoted by upper case latin symbols). For the sake of simplicity, in the following formulations we will assume that all design variables are random and that there exist no random parameters.

2.2.1. *Stochastic Formulation*
In the stochastic formulation, each random variable is represented by a parameter that describes its probabilistic characteristics. Typically, this parameter is the first moment, or mean, of the random variable. Responses and other functions of random variables are expressed as expected values. Thus, Problem (1) becomes

$$\begin{aligned}
\min_{\mu_{\tilde{\mathbf{X}}_{ij}},\epsilon^R_{ij},\epsilon^Y_{ij}} \quad & \|E[\mathbf{R}_{ij}] - \mu^u_{\mathbf{R}_{ij}}\|_2^2 + \|\mu_{\mathbf{Y}_{ij}} - \mu^u_{\mathbf{Y}_{ij}}\|_2^2 + \epsilon^R_{ij} + \epsilon^Y_{ij} \qquad (2)\\
\text{subject to} \quad & \textstyle\sum_{k=1}^{n_{ij}} \|\mu_{\mathbf{R}_{(i+1)k}} - E[\mathbf{R}_{ij}]^l\|_2^2 \le \epsilon^R_{ij}\\
& \textstyle\sum_{k=1}^{n_{ij}} \|\mu_{\mathbf{Y}_{(i+1)k}} - \mu^l_{\mathbf{Y}_{(i+1)k}}\|_2^2 \le \epsilon^Y_{ij}\\
& E[\mathbf{g}_{ij}(\mathbf{R}_{(i+1)1},\ldots,\mathbf{R}_{(i+1)n_{ij}},\mathbf{X}_{ij},\mathbf{Y}_{ij})] \le \mathbf{0}\\
& E[\mathbf{h}_{ij}(\mathbf{R}_{(i+1)1},\ldots,\mathbf{R}_{(i+1)n_{ij}},\mathbf{X}_{ij},\mathbf{Y}_{ij})] = \mathbf{0}\\
\text{with} \quad & \mathbf{R}_{ij} = \mathbf{f}_{ij}(\mathbf{R}_{(i+1)1},\ldots,\mathbf{R}_{(i+1)n_{ij}},\mathbf{X}_{ij},\mathbf{Y}_{ij}),
\end{aligned}$$

where $E[\cdot]$ denotes the expectation operator.

In words, this formulation attempts to

1. Match the expected values of the local responses with the targets cascaded from the higher level; these targets are the optimal values of the random design variables, i.e., the means, of the higher-level problem.

2. Match the optimal values of the random response design variables, i.e., the means, with the expected values of the children responses.

3. Match the optimal values of the local and children random shared variables, i.e., the means, with the target values cascaded from the higher and lower levels, respectively.

The challenge in solving stochastic optimization problems such as Problem (2) is that evaluating expectations requires knowledge of the probability density functions of the random variables and evaluation of multidimensional integrals.

The solution of Problem (2) satisfies the design inequality and equality constraints in an average sense, but does not provide any information on the percentage of constraint violations due to uncertainty. In practical applications, however, there is a need to satisfy the constraints at a specified target reliability level.

2.2.2. *Probabilistic Formulation*
The constraints are thus reformulated. We now require that the probability of satisfying a constraint under the presence of uncertainties greater than some appropriately selected threshold, or, alternatively, that the probability of violating a constraint is less than some pre-specified probability of failure. The formulation of Problem (2) becomes

$$\begin{aligned}
\min_{\mu_{\tilde{\mathbf{X}}_{ij}},\epsilon^R_{ij},\epsilon^Y_{ij}} \quad & \|E[\mathbf{R}_{ij}] - \mu^u_{\mathbf{R}_{ij}}\|_2^2 + \|\mu_{\mathbf{Y}_{ij}} - \mu^u_{\mathbf{Y}_{ij}}\|_2^2 + \epsilon^R_{ij} + \epsilon^Y_{ij} \qquad (3)\\
\text{subject to} \quad & \textstyle\sum_{k=1}^{n_{ij}} \|\mu_{\mathbf{R}_{(i+1)k}} - E[\mathbf{R}_{ij}]^l\|_2^2 \le \epsilon^R_{ij}
\end{aligned}$$

$$\sum_{k=1}^{n_{ij}} \|\mu_{\mathbf{Y}_{(i+1)k}} - \mu^{l}_{\mathbf{Y}_{(i+1)k}}\|_2^2 \leq \epsilon_{ij}^{Y}$$
$$P[\tilde{\mathbf{g}}_{ij}(\mathbf{R}_{(i+1)1}, \ldots, \mathbf{R}_{(i+1)n_{ij}}, \mathbf{X}_{ij}, \mathbf{Y}_{ij}) > 0] \leq \mathbf{P}_f,$$
$$\text{with} \qquad \mathbf{R}_{ij} = \mathbf{f}_{ij}(\mathbf{R}_{(i+1)1}, \ldots, \mathbf{R}_{(i+1)n_{ij}}, \mathbf{X}_{ij}, \mathbf{Y}_{ij}),$$

where $P[\cdot]$ denotes probability measure and $\mathbf{P}_f$ is a vector of prespecified probability of failure thresholds.

Note that the mathematical formulation of Problem (3) does not contain equality constraints. Equality constraints do not make sense in a probabilistic framework (it is meaningless to require that a function takes exactly a specific value under the presence of uncertainty, since the probability of a continuous random variable taking an exact value is zero), one has to introduce some slack and treat equality constraints as inequality constraints. For example, if in a deterministic framework it is required that $h(\mathbf{x}) = 0$, in a probabilistic framework it is required that $\mid h(\mathbf{X}) \mid \leq \delta$, where $\delta$ is a small positive constant, so that the constraint is formulated as $P[\mid h(\mathbf{X}) \mid -\delta > 0] \leq P_f$. Therefore, we rewrite equality constraints as inequality constraints and unite the two constraint function vectors into one, denoted by $\tilde{\mathbf{g}}$.

Problem (3) can be solved with any of the available commercial software packages or the methods reported recently in the literatures, e.g., the hybrid mean value (HMV) method or the sequential optimization and reliability assessment (SORA) method [22, 23]. We adopt a recently developed single-loop method that is as accurate as the HMV and the SORA methods, but much more efficient [24].

## 3. Propagation of Uncertainties

The responses of the elements in the multilevel hierarchy are typically nonlinear functions of the elements' inputs, which include random variables and parameters. Thus, responses are themselves random variables, whose expected value must be computed to evaluate objective and constraints when solving probabilistic optimization problems. Moreover, estimated variance of responses is required if robustness considerations are included.

In a multilevel hierarchy, responses of lower-level subsystems are inputs to higher-level subsystems. Therefore, it is necessary to obtain probability distribution information required for the solution of the higher-level problems. This is an issue of outmost importance in design optimization of hierarchically decomposed multilevel systems. An efficient and accurate mechanism is required for propagating probabilistic information in the form of cumulative distribution and probability density functions throughout the hierarchy.

### 3.1. Estimating Moments Using the Mean-Value First-Order Second-Moment Method

In an initial effort, a mean-value first-order second-moment (MVFOSM) approach was adopted to estimate the mean and standard deviation of a nonlinear function of random

variables [25]. Specifically, a first-order Taylor expansion about the current design, represented by the mean vector $\mu_{\mathbf{X}}$ of the random variables $\mathbf{X}$, was used to linearize a nonlinear random response $R$:

$$R = f(\mathbf{X}) \approx f(\mu_{\mathbf{X}}) + \sum_{i=1}^{n} \frac{\partial f(\mu_{\mathbf{X}})}{\partial X_i}(X_i - \mu_{X_i}), \tag{4}$$

where $n$ is the dimension of the vector $\mathbf{X}$. Assuming that all the random variables are statistically independent (uncorrelated), the first-order approximations of the mean and the variance of $R$ were given by

$$E[R] = \mu_R \approx f(\mu_{\mathbf{X}}) \tag{5}$$

and

$$Var[R] = \sigma_R^2 \approx \sum_{i=1}^{n} \left( \frac{\partial f(\mu_{\mathbf{X}})}{\partial X_i} \right)^2 \sigma_{X_i}^2, \tag{6}$$

respectively.

The advantage of this approach, besides efficiency, is that it allowed us to assume that the responses are normally distributed if all input random variables and parameters were normal. Therefore, propagation of uncertainty in ATC was modeled as a linear process. With the distribution information known, all that was necessary was the estimation of the first two moments, which characterize a normal distribution completely. The validity of the successive linearizations during the ATC process was ensured by virtue of the ATC consistency constraints that do not allow large deviations from current designs.

To our knowledge, this linearization approach is currently embedded in all state-of-the-art software packages for optimization under uncertainty. As will be demonstrated shortly, the linearization approach does a fairly good job in estimating the expected value of nonlinear functions of random variables. However, it can be quite inaccurate in estimating higher moments, e.g., the standard deviation. Moreover, it is limiting in that it does not provide us with the correct probability distribution information of the random nonlinear responses.

It is also important to note that if the linearization approach is used to compute expectations in the stochastic formulation, Problems (1) and (2) generate identical solutions. There is no value in solving the stochastic ATC formulation if expectations are not computed exactly, which requires accurate probability distribution information and multidimensional integrations. This is an additional reason that may explain why the probabilistic constraint formulation is used universally today to solve non-deterministic problems.

### 3.2. Generating Distributions Using the Advanced Mean Value Method

In this paper, we utilize the advanced mean value (AMV) method to generate the cumulative distribution function (CDF) of a nonlinear response. The AMV method [26] is a computationally efficient method for generating the CDF of nonlinear functions of random variables. It improves the Mean Value (MV) prediction (Section 3.1) by using a simple correction to compensate for errors introduced from the Taylor series truncation. A response performance

function $R = f(\mathbf{X})$ is linearized as shown in Eq. (4) and its first and second order moments $\mu_R$ and $\sigma_R$ are calculated using Eqs. (5) and (6), respectively.

A limit state function is then defined as

$$g(\mathbf{X}) = f(\mathbf{X}) - f_0, \tag{7}$$

where $f_0$ is a particular value of the performance function. The reliability index $\beta$ is then given by

$$\beta = \frac{\mu_g}{\sigma_g}, \tag{8}$$

where $\mu_g = \mu_R - f_0$ and $\sigma_g = \sigma_R$. The CDF value of $f$ at $f_0$ is calculated from the first-order relation

$$P[f \leq f_0] = P[g \leq 0] = \Phi(-\beta), \tag{9}$$

where $\Phi$ is the standard normal cumulative distribution function. It is emphasized that Eq. (8) is equivalent to calculating the most probable point (MPP) using the linear approximation of Eq. (4). The MPP in the standard normal space is given by

$$\mathbf{U}^* = -\beta \frac{\nabla g(\mathbf{X})}{|\nabla g(\mathbf{X})|}. \tag{10}$$

In the original $X$ space, the MPP coordinates vector is

$$\mathbf{X}^* = \mathbf{U}^* \sigma_{\mathbf{x}} + \mu_{\mathbf{x}}, \tag{11}$$

where $\mu_{\mathbf{x}}$ and $\sigma_{\mathbf{x}}$ are the mean and standard deviation vectors, respectively, of the vector of random variables $\mathbf{X}$.

In the AMV method, the following relation is used instead of Eq. (9):

$$P[f \leq f(\mathbf{X}^*)] = \Phi(-\beta), \tag{12}$$

i.e., the $f_0$ value at which the reliability index $\beta$ is calculated is replaced by $f(\mathbf{X}^*)$.

To generate the CDF of $R = f(\mathbf{X})$, the Most Probable Point is first approximated using the simple MV method, which has minimal computational requirements relative to existing MPP-based reliability analysis methods. Once all MPP's $\mathbf{X}_i^*$ for an appropriately discretized range of the performance function at points $f_i$ are obtained, the so-called MPP locus (MPPL) is identified, and is equivalent with the CDF of $R = f(\mathbf{X})$. Subsequently, a single function evaluation $f(\mathbf{X}_i^*)$ is used at each CDF level $i$ to correct the CDF value obtained with the MV method. This so-called AMV-based method is computationally efficient since it requires only a single linearization of the performance function at the mean value and an additional function evaluation at each CDF level (discretized $f$ range at values $f_i$). It is also very accurate as repeatedly demonstrated in the literature [27–29]. Note that the MPPL-based CFD generation concept has been reported before, but is was based on a less efficient MPP determining procedure [30].

With the CDF available, one can differentiate numerically to obtain the probability density function (PDF). We use central differences to obtain second-order accurate approximations. Finally, to compute moments, we integrate numerically, using spline interpolation

to estimate response values that lie between the available PDF values. As will be shown by means of several analytical examples, this method is quite accurate.

### 3.3. Examples

The MVFOSM-based and AMV-based methods were used to estimate the first two moments of several nonlinear analytical expressions. All random variables were assumed to be normal. Test functions and input statistics are presented in Table I and results are summarized in Table II. One million samples were used for the Monte Carlo simulations.

Table I. Test functions and input statistics

| # | Expression | Input Statistics |
|---|---|---|
| 1 | $X_1^2 + X_2^2$ | $X_1 \sim N(10, 2)$, $X_2 \sim N(10, 1)$ |
| 2 | $-\exp(X_1 - 7) - X_2 + 10$ | $X_{1,2} \sim N(6, 0.8)$ |
| 3 | $1 - \frac{X_1^2 X_2}{20}$ | $X_{1,2} \sim N(5, 0.3)$ |
| 4 | $1 - \frac{(X_1+X_2-5)^2}{30} - \frac{(X_1-X_2-12)^2}{30}$ | $X_{1,2} \sim N(5, 0.3)$ |
| 5 | $1 - \frac{80}{X_1^2+8X_2+5}$ | $X_{1,2} \sim N(5, 0.3)$ |

Table II. Estimated moments and errors relative to Monte Carlo simulation (MCS) results

| # | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\mu_{\text{lin}}$ | 200.0 | 3.6321 | −5.25 | −1.0333 | −0.1428 |
| $\mu_{\text{AMV}}$ | 203.4 | 3.6029 | −5.3495 | −1.0380 | −0.1454 |
| $\mu_{\text{MCS}}$ | 205.0 | 3.4921 | −5.3114 | −1.0404 | −0.1448 |
| $\epsilon_{\text{lin}}$ [%] | −2.44 | 4.00 | −1.15 | −0.68 | −1.30 |
| $\epsilon_{\text{AMV}}$ [%] | −0.78 | 3.17 | 0.71 | −0.23 | 0.41 |
| $\sigma_{\text{lin}}$ | 44.72 | 1.9386 | 0.8385 | 0.1166 | 0.00627 |
| $\sigma_{\text{AMV}}$ | 45.20 | 0.9013 | 0.8423 | 0.1653 | 0.00631 |
| $\sigma_{\text{MCS}}$ | 45.10 | 0.9327 | 0.8407 | 0.1653 | 0.00630 |
| $\epsilon_{\text{lin}}$ [%] | −0.84 | 107.85 | −0.26 | 29.46 | −0.47 |
| $\epsilon_{\text{AMV}}$ [%] | 0.22 | −3.36 | 0.19 | 0 | 0.15 |

By inspecting Table II, it can be seen that while the mean-related errors of the linearization approach are within acceptable limits, standard deviation errors can be quite large. The AMV-based moment estimation method performs always better, and never exhibits unacceptable errors. Moreover, the AMV-method provides accurate probability distribution information of nonlinear responses. For example, Figure 3 depicts the CDF

and PDF, respectively, of function # 1, obtained using both the MVFOSM-based and the AMV-based method. It can be seen that, using the linearization approach, the nonlinear response would be incorrectly assumed as normally distributed.

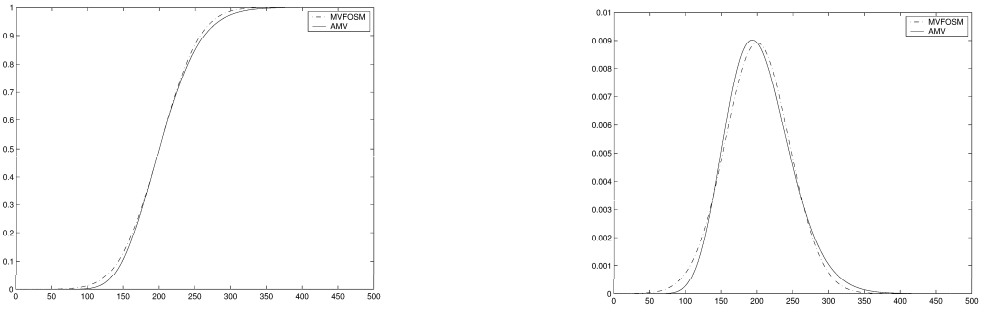



*Figure 3.* Cumulative distribution and probability density functions for analytical example #1

### 3.4. Propagating Uncertainty in ATC

Our methodology for propagating uncertainty information during the ATC process can be summarized in the following steps:

1. Start at the bottom level of the hierarchy, where probability distribution on the input random variables and parameters is assumed as known. If such information is not available at the bottom level, start at the lowest level possible where such information is available.

2. Solve the probabilistic design optimization problems for the level specified in step 1.

3. Use the approach described in Section 3.2 to obtain distribution information for the response variables that are inputs to higher-level ("parent") problems.

4. Using the information obtained at step 3, solve the parent problems. Note that the CDFs and PDFs of lower-level ("children") responses that constitute optimization variables in the parent problems are required for solving these problems correctly. Second moment (variance) information alone is inadequate to guarantee proper solution process and uncertainty propagation throughout the hierarchy (as opposed, e.g., to "single"-element robust design optimization).

5. Move your way to the top of the hierarchy.

6. Once you have reached the top-level problem start moving towards the bottom using previous solutions to update parameters as shown in Figure 2.

7. Keep iterating until all $\epsilon$ values in all problems in the hierarchy have been reduced as much as possible, i.e., have converged to a steady state value. Note that the $\epsilon$ variables

are deterministic, as are the constraints they appear in. While uncertainties are taken into account in the probabilistic design constraints, the non-deterministic ATC process aims at coordinating values of shared variables and responses in an average sense.

Since the linearization approach is sufficiently accurate for estimating expected values, it can be used to reduce computational cost. However, the AMV-based method is so efficient, that it is suggested for use in estimating expected values to improve accuracy and thus, possibly, the convergence rate of the ATC process.

## 4. Example

The probabilistic formulation of the ATC process (Problem (3)) is used to solve a simple yet illustrative simulation example. We consider a V6 gasoline engine as the system, which is "decomposed" into a subsystem that represents the piston-ring/cylinder-liner subassembly of a single cylinder. The system simulation predicts engine performance in terms of brake-specific fuel consumption. Although the engine has six cylinders, they are all designed to be identical. For this reason, we only consider one subsystem.

The associated bi-level hierarchy, shown in Figure 4, includes the engine as a system at the top level and the piston-ring/cylinder-liner subbassembly as a subsystem at the bottom level. The ring/line subassembly simulation takes as inputs the surface roughness of the

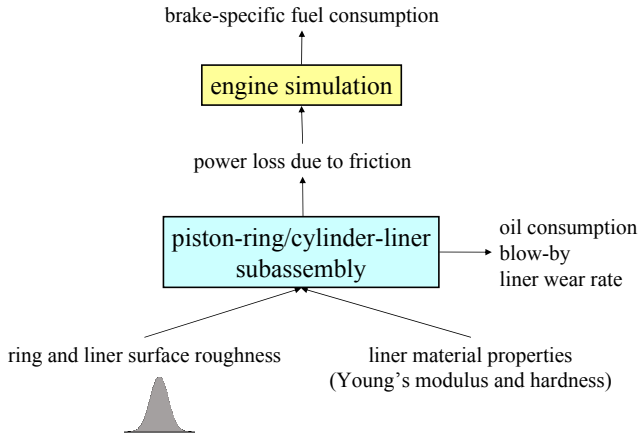



*Figure 4.* Hierarchical bi-level system

ring and the liner and the Young's modulus and hardness and computes power loss due to friction, oil consumption, blow-by, and liner wear rate. The root mean square (RMS) of asperity height is used to represent asperity roughness, which is assumed to be normally distributed. The engine simulation takes then as input the power loss and computes brake-specific fuel consumption of the engine. Commercial software packages were used to perform the simulations. A detailed description of the problem can be found in [25].

### 4.1. Problem Formulation

Due to the simplicity of the given problem structure, we will use here a modified version of the notation introduced earlier. Since there are only two levels with only one element in each, we skip element indices and denote the upper-level element with subscript 0 and the lower-level element with subscript 1. We use second indices to denote entries in the design variable vector of the lower-level element optimization problem. The design problem is to find optimal mean values $\mu_{X_{11}}$ and $\mu_{X_{12}}$ for the piston-ring and cylinder-liner surface roughness random variables $X_{11}$ and $X_{12}$, respectively, and optimal values for the deterministic design variables representing the material properties (Young's modulus $x_{13}$ and hardness $x_{14}$) of the liner that yield minimized expected value of brake-specific fuel consumption $R_0$. The optimal design is subject to constraints on liner wear rate, oil consumption, and blow-by. The power loss due to friction $R_1$ links the two levels.

The top- and bottom-level ATC problems are formulated as

$$\begin{aligned} \min_{\mu_{R_1},\epsilon^R} \quad & (E[R_0]-T)^2+\epsilon^R \\ \text{subject to} \quad & (\mu_{R_1}-E[R_1]^l)^2 \le \epsilon^R \\ \text{with} \quad & R_0 = f_0(R_1) \end{aligned} \tag{13}$$

and

$$\begin{aligned} \min_{\mu_{X_{11}},\mu_{X_{12}},x_{13},x_{14}} \quad & (E[R_1]-\mu^u_{R_1})^2 \\ \text{subject to} \quad & P[\text{liner wear rate} > 2.4\times10^{-12}\ m^3/s] \le P_f \\ & P[\text{blow-by} > 4.25\times10^{-5}\ kg/s] \le P_f \\ & P[\text{oil consumption} > 15.3\times10^{-3}\ kg/hr] \le P_f \\ & P[X_{11} < 1\mu m] \le P_f \\ & P[X_{11} > 10\mu m] \le P_f \\ & P[X_{12} < 1\mu m] \le P_f \\ & P[X_{12} > 10\mu m] \le P_f \\ & 340\ GPa \ge x_{13} \ge 80\ GPa \\ & 240\ BHV \ge x_{14} \ge 150\ BHV \\ \text{with} \quad & R_1 = f_1(X_{11},X_{12},x_{13},x_{14}), \end{aligned} \tag{14}$$

respectively. The standard deviation of the surface roughnesses was assumed to be 1.0 $\mu m$, and remained constant throughout the ATC process. The assigned probability of failure $P_f$ was 0.13%, which corresponds to the target reliability index $\beta = 3$. The fuel consumption target $T$ was simply set to zero to achieve the best fuel economy possible.

Note that since the random variables are normally distributed, the associated linear probabilistic bound constraints can be reformulated as deterministic. For example,

$$P[X_{11} < 1\mu m] \le P_f \Leftrightarrow P[X_{11} - 1\mu m < 0] \le P_f \Leftrightarrow$$

$$\Phi(0 - \frac{\mu_{X_{11}} - 1\mu m}{\sigma_{X_{11}}}) \leq \Phi(-\beta) \Rightarrow -\frac{\mu_{X_{11}} - 1\mu m}{\sigma_{X_{11}}} \leq -\beta \Leftrightarrow$$
$$\frac{\mu_{X_{11}} - 1\mu m}{\sigma_{X_{11}}} \geq \beta \Leftrightarrow \mu_{X_{11}} - 1\mu m \geq \beta\sigma_{X_{11}} \Leftrightarrow$$
$$\mu_{X_{11}} \geq 1\mu m + \beta\sigma_{X_{11}} \Leftrightarrow \mu_{X_{11}} \geq 4\mu m$$

Similarly, the other three probabilistic bound constraints in Problem (14) can be reformulated as

$$\mu_{X_{11}} \leq 7\mu m; \quad \mu_{X_{12}} \geq 4\mu m; \quad \mu_{X_{12}} \leq 7\mu m.$$

### 4.2. Results

It is desired to minimize power loss due to friction in order to optimize engine operation and thus maximize fuel economy. Therefore, it was anticipated that the bottom-level optimization problem would yield a design with as smooth surfaces (low surface roughnesses) as possible.

The probabilistic ATC process of solving Problems (14) and (13) iteratively converged after two iterations. The obtained optimal ring/liner subassembly design is shown in Table III. The ring surface roughness optimal value is at its probabilistic lower minimum,

Table III. Optimal ring/liner subassembly design

| Variable | Description | Value |
|---|---|---|
| $X_{11}$ | Ring surface roughness, $[\mu m]$ | 4.00 |
| $X_{12}$ | Liner surface roughness, $[\mu m]$ | 6.15 |
| $x_{13}$ | Liner Young's modulus, $[GPa]$ | 80 |
| $x_{14}$ | Liner hardness, $[BHV]$ | 240 |

while the liner's Young's modulus and hardness optimal values are at their deterministic lower and upper bounds, respectively.

The liner surface roughness is not, however, at its lower bound because the problem is bounded by the oil consumption constraint. A certain degree of surface roughness is required to maintain an optimal oil film thickness in order to avoid excessive oil consumption. For this reason, the associated constraint is active, and the surface roughness of the liner is an interior optimizing argument.

An interesting theoretical issue arises. How do we define activity for probabilistic constraints? The definition of constraint activity in deterministic optimization is the following: A constraint is active if removing it or moving its boundary affects the location of the optimum. In probabilistic design, a constraint is active if the reliability index associated with the constraint's MPP is equal to the target reliability index. In other words, the constraint's MPP lies on the target reliability circle.

A Monte Carlo simulation was performed to assess the accuracy of the reliability analyses of the probabilistic constraints. One million samples were generated using the mean and standard deviation values of the design variables, and the constraints were evaluated using these samples to calculate the probability of failure. Results are summarized in Table IV.

Table IV. Reliability analysis results

| Constraint | Active | $P_f$ | MCS $P_f$ |
|---|---|---|---|
| Liner wear rate | No | $\leq 0.13$ % | 0 % |
| Blow-by | No | $\leq 0.13$ % | 0 % |
| Oil consumption | Yes | 0.13 % | 0.16 % |

The obtained design is actually 0.03% less reliable than found. This error is due to the first-order reliability approximation used in the probabilistic optimization problem.

Propagation of uncertainty was modeled using the approach described in Section 3.2. Table V summarizes the estimated moments for the two responses of the bi-level hierarchy.

Table V. Estimated moments and errors relative to Monte Carlo simulation (MCS) results for the simulation example

| Response | Power loss | Fuel consumption |
|---|---|---|
| $\mu_{\text{lin}}$ | 0.3950 | 0.5341 |
| $\mu_{\text{AMV}}$ | 0.3922 | 0.5431 |
| $\mu_{\text{MCS}}$ | 0.3932 | 0.5432 |
| $\epsilon_{\text{lin}}$ [%] | 0.45 | $-0.01$ |
| $\epsilon_{\text{AMV}}$ [%] | $-0.25$ | $-0.01$ |
| $\sigma_{\text{lin}}$ | 0.0481 | 0.00757 |
| $\sigma_{\text{AMV}}$ | 0.0309 | 0.00760 |
| $\sigma_{\text{MCS}}$ | 0.0311 | 0.00759 |
| $\epsilon_{\text{lin}}$ [%] | 54.6 | $-0.25$ |
| $\epsilon_{\text{AMV}}$ [%] | $-0.64$ | 0.13 |

The linearization approach results are included to illustrate the large error that this approach introduces to the top-level problem. This happens because the power loss function is highly nonlinear. In fact, its PDF is multi-modal, as illustrated in Figure 5. Figure 5 also depicts the histogram obtained by Monte Carlo simulation using one million samples; note that the perpendicular axis of the histogram must be divided by 1,000,000 to obtain the probability density relative to the sample size. The agreement is quite satisfactory and illustrates the usefulness of the AMV-based approach to propagate uncertainty for highly nonlinear functions. The fuel consumption is almost a linear function of the power loss.

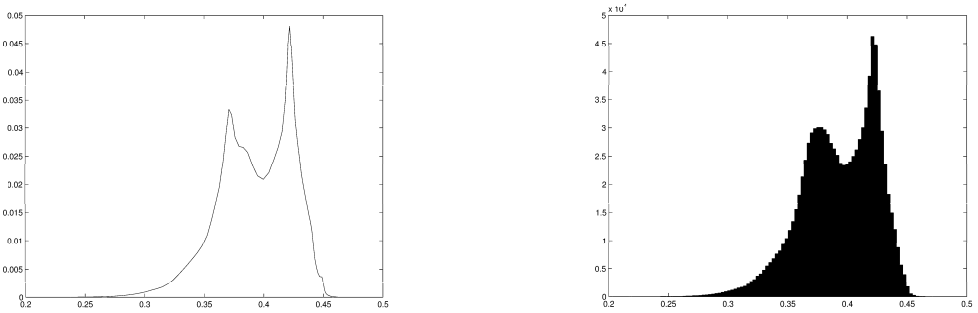


*Figure 5.* Power loss PDF (left) and histogram obtained using Monte Carlo simulation (right)

## 5. Summary and Conclusions

We have presented a methodology for design optimization of hierarchically decomposed multilevel systems under uncertainty. We extended the deterministic formulation of analytical target cascading (ATC) to account for uncertainties. We modeled the propagation of uncertainty in the ATC process by using the advanced mean value (AMV) method to generate accurate probability distributions of nonlinear responses. We demonstrated the presented methodology by means of a simple yet illustrative engine design example. The proposed methodology for simulation-based optimal system design by decomposition is not related to multidisciplinary design optimization (MDO) methods in either its deterministic or its probabilistic formulation. Stochastic formulations are meaningful only if expectations of nonlinear responses are computed exactly, which requires probability distribution information of the input random variables and parameters and accurate multidimensional integrations. Probabilistic formulations are suggested for practical applications. The linearization approach for propagating uncertainties yields inaccurate second moment estimations and is inadequate for multilevel optimization under uncertainty since it does not provide probability distribution information that is necessary for solving higher-level problems.

## References


1. R.H. Sues, D.R. Oakley, and G.S Rhodes. "Multidisciplinary stochastic optimization". In *Proceedings of the 10th Conference on Engineering Mechanics*, pages 934–937, Boulder, Colorado, 1995.
2. D.R. Oakley, R.H. Sues, and G.S. Rhodes. "Performance optimization of multidisciplinary mechanical systems subject to uncertainties". *Probabilistic Engineering Mechanics*, 13(1):15–26, 1998.
3. X. Gu, J.E. Renaud, and S.M Batill. "An investigation of multidisciplinary design subject to uncertainty". In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Sumposium on Multidisciplinary Analysis and Optimization*, St. Louis, Missouri, 1998. Paper no. AIAA-1998-4747.
4. P.N. Koch, T.W. Simpson, J.K. Allen, and F. Mistree. "Approximations for multidisciplinary design optimization". *Journal of Aircraft*, 36(1):275–286, 1999.

5. X. Gu, J.E. Renaud, S.M. Batill, R.M. Brach, and A.S. Budhiraja. "Worst case propagated uncertainty of multidisciplinary systems in robust design optimization". *Structural and Multidisciplinary Optimization*, 20(3):190–213, 2000.
6. X. Gu, J.E. Renaud, L.M. Ashe, S.M. Batill, A.S. Budhiraja, and L.J. Krajewski. "Decision-based collaborative optimization under uncertainty". In *Proceedings of the 27th ASME Design Automation Conference*, Baltimore, Maryland, 2001. Paper no. DETC2000/DAC-14297.
7. X. Du and W. Chen. "Methodology for managing the effect of uncertainty in simulation-based systems design". *AIAA Journal*, 38(8):1471–1478, 2000.
8. X. Gu and J.E. Renaud. "Implicit uncertainty propagation for robust collaborative optimization". In *Proceedings of the 27th ASME Design Automation Conference*, Pittsburgh, Pennsylvania, 2001. Paper no. DETC2001/DAC-21118.
9. X. Gu and J.E. Renaud. "Implementation study of implicit uncertainty propagation in decomposition-based optimization". In *Proceedings of the 9th AIAA/USAF/NASA/ISSMO Sumposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia, 2002. Paper no. AIAA-2002-5416.
10. X. Du and W. Chen. "Efficient uncertainty analysis methods for multidisciplinary robust design". *AIAA Journal*, 40(3):545–552, 2002.
11. X. Du and W. Chen. "Collaborative reliability analysis under the framework of multidisciplinary systems design". *Optimization and Engineering*. In press.
12. R.H. Sues and M.A. Cesare. "An innovative framework for reliability-based MDO". In *Proceedings of the 41th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Atlanta, Georgia, 2000.
13. R.H. Sues, M.A. Cesare, S.S. Pageau, and Y.T. Wu. "Reliability-based optimization considering manufacturing and operational uncertainties". *Journal of Aerospace Engineering*, 14:166–174, 2001.
14. N.F. Michelena, H.M. Kim, and P.Y. Papalambros. "A system partitioning and optimization approach to target cascading". In *Proceedings of the 12th International Conference on Engineering Design*, Munich, Germany, 1999.
15. H.M. Kim. *Target Cascading in Optimal System Design*. PhD thesis, University of Michigan, 2001.
16. H.M. Kim, N.F. Michelena, P.Y. Papalambros, and T. Jiang. "Target cascading in optimal system design". *ASME Journal of Mechanical Design*, 125(3):474–480, 2003.
17. N.F. Michelena, H. Park, and P.Y. Papalambros. "Convergence properties of analytical target cascading". *AIAA Journal*, 41(5):897–905, 2003.
18. H.M. Kim, M. Kokkolaras, L.S. Louca, G.J. Delagrammatikas, N.F. Michelena, Z.S. Filipi, P.Y. Papalambros, J.L. Stein, and D.N. Assanis. "Target cascading in vehicle redesign: A class VI truck study". *International Journal of Vehicle Design*, 29(3):1–27, 2002.
19. M. Kokkolaras, R. Fellini, H.M. Kim, N.F. Michelena, and P.Y. Papalambros. "Extension of the target cascading formulation to the design of product families". *Structural and Multidisciplinary Optimization*, 24(4):293–301, 2002.
20. H.M. Kim, D.G. Rideout, P.Y. Papalambros, and J.L. Stein. "Analytical target cascading in automotive vehicle design". *ASME Journal of Mechanical Design*, 125(3):481–489, 2003.
21. R. Choudhary, P.Y. Papalambros, and A. Malkawi. "Analytical target cascading in building performance analysis". *Building and Environment*. In review.
22. B.D. Youn, K.K. Choi, and Y.H. Park. "Hybrid analysis method for reliability-based design optimization". *Journal of Mechanical Design*, 125:221–232, 2003.
23. X. Du and W. Chen. "Sequential optimization and reliability assessment method for efficient probabilistic design". *Journal of Mechanical Design*. In press.
24. J. Liang, Z.P. Mourelatos, and J. Tu. "A single-loop method for reliability-based design optimization". In *Proceedings of the 30th ASME Design Automation Conference*, Salt Lake City, Utah, 2004. Submitted for consideration.
25. K.Y. Chan, M. Kokkolaras, P.Y. Papalambros, S.J. Skerlos, and Z. Mourelatos. "Propagation of uncertainty in optimal design of multilevel systems: Piston-ring/cylinder-liner case study". In *Proceedings of the 2004 SAE World Congress*, Detroit, Michigan, March 8-11. Paper No. 2004-01-1559.

26. Y.T. Wu, H.R. Millwater, and T.A. Cruse. "Advanced probabilistic structural analysis method of implicit performance functions". *AIAA Journal*, 28(9):1663–1669, 1990.
27. Y.T. Wu and O.H. Burnside. "Validation of the NESSUS probabilistic analysis computer program". In *Proceedings of the 29th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference*, Williamsburg, Virginia, 1988.
28. T.A. Cruse, Y.T. Wu, J.B. Dias, and K.R. Rajagopal. "Probabilistic structural analysis methods and applications". *Computers and Structures*, 30(1-2):163–170, 1988.
29. Y.T. Wu, O.H. Burnside, and T.A. Cruse. "Probabilistic methods for structural response analysis". In W.K. Liu and T. Belytschko, editors, *Computational Mechanics of Probabilistic and Reliability Analysis*. Elmepress International, 1989.
30. X. Du and W. Chen. "A most probable point-based method for efficient uncertainty analysis". *Journal of Design and Manufacturing Automation*, 1(1-2):47–66, 2001.

# Calculating Risk of Cost Using Monte Carlo Simulations with Fuzzy Parameters in Civil Engineering

MICHAŁ BĘTKOWSKI
*Department of Civil Engineering, Silesian University of Technology,Gliwice, Poland,*
*mb@zeus.polsl.gliwice.pl*

ANDRZEJ POWNUK
*Department of Civil Engineering, Silesian University of Technology,Gliwice, Poland,*
*pownuk@zeus.polsl.gliwice.pl, http://zeus.polsl.gliwice.pl/ pownuk*

August 1, 2004

**Abstract.** Risk is a part of almost all civil engineering projects. Usually there is a difference between the real and the estimated cost of the civil engineering projects. Unfortunately, in civil engineering applications usually we do not have enough data to calculate probabilistic characteristics [13]. There are also different methods of modeling of uncertainty [6, 4]. In this paper probabilistic characteristics are modeled by fuzzy numbers, which are defined by some expert. The resulting cost is described by probability density functions with fuzzy characteristics (for example mean or standard deviation). Using assessment from different (or even one) experts we can estimate the uncertainty of the probability density function of total costs and the risk. Then using modified Monte-Carlo simulation and the alpha cut method we can calculate the results.

**Keywords:** risk, costs, uncertainty, impresise probability

## 1. Introduction

Risk us an integral part of each civil engineering project. We can define it as possibility of occurrence of loss. One of the most popular type contracts in Poland is (guaranteed maximum price or cost contract) . At this time task and costs are predicted on the basis on deterministic unit costs [13].

Tasks and unit costs are deterministic. Unfortunately, in reality schedule tasks and unit costs may change because of the influence of different and usually uncertain factors[2].

## 2. Calculating of cost of civil engineering projects

Today in Poland the cost of civil engineering project is calculated by using pure deterministic methods which are based on some catalogues [11], set of prices [8, 9] and/or different norms. Existing practical methods of calculating costs are pure deterministic.

The total cost can be calculated as:

$$c_T = DC + IC + P + T \tag{1}$$

where $DC$ - direct costs (labor, material, equipment), $IC$ - indirect cost (costs of management, cost of insurance etc.), $P$ - profit and risk of the project, $T$ – taxes.

In this paper only direct costs $DC$ are taken into account.

$$DC = \sum_{i=1}^{n} DC_i \tag{2}$$

where $DC_i$ direct costs of each tasks.

## 3. Probabilistic definition of risk

Many variables have impact upon cost overruns. The prime variables have been commonly identified as: unpredictable weather, inflationary material cost, inaccurate materials estimates, complexity of project, contractor's lack experience, poor labor productivity, project changes [10, 6].

The risk of cost is equal to the probability that the real cost $c_T$ is grater than assumed cost $c_{T,0}$ (maximal).

$$R = P\left\{c_T > c_{T,0}\right\} = 1 - P\left\{c_T \leq c_{T,0}\right\} \tag{3}$$

If we know the probability density function $f_{c_T}(x)$ of the random variable $c_T$ then

$$R = 1 - \int_{-\infty}^{c_{T,0}} f_{c_T}(x)\, dx = 1 - \Phi_{c_T}(c_{T,0}) \tag{4}$$

where

$$\Phi_{c_T}(x) = \int_{-\infty}^{x} f_{c_T}(t)\, dt \tag{5}$$

is a cumulative distribution function of the random variable $c_T$.

It should be emphasize that the influence of the uncertainty to the final cost is very difficult to estimate by using pure probabilistic methods due to lack of credible statistical data.

## 4. Calculating of risk of direct costs

At this moment the direst costs are calculated on the basis on standards [13] which are very general and they do not take into account different factors which have influence on their values. Because of that there is a difference between the real costs and predicted costs.

Risk is calculated as constant value which is introduced in order to cover the losses. The final result of the calculation is a fixed value.

The final price is a result of negotiation between investor and contractors.

Important information for the contractor is the following:

- what is the level of risk which accompany assumed maximal level of direct costs.
- what is the minimal cost for which the risk can be accepted.

Knowledge about influence of random parameters of the system would be a very good in negotiations.

There are many programs which enable to calculate project risk (for example Pert Master, Risk, MS Project etc.) in pure probabilistic sense. However in practice it is very difficult to obtain reliable statistical data, because of that the results of the calculations are not credible.

## 5. Example of analysis of risk

Let's assume that contractor would like to realize some civil engineering project for fixed price. The project consist of: determine tasks, alternative tasks and additional tasks.

One can called the task deterministic if occurrence of it is certain.

Let's assume that we have two tasks. If in each realization of that process we can get only one of them, then we can call these tasks alternative.

If the task may occur in each realization with some probability then we can call that task additional.

### 5.1. Preparation of data

Valuation of identified tasks was made on the basis of [8, 9, 11]. Then the data was aggregated with taking into account technology of realization and allocation of risk. The results are presented in the table 1.

The model of the system consists of some node. Each node is characterized by some cost$c_i$. One can also define some relations between the elements. Both route thru the graph and the costs $c_i$ are random. The process can be shown as a Petri nets[14] on the Fig. 1.

The tasks are represented by rectangles, conditions are represented as circle and the arrows show the direction of movement in the graph. On some connections there is information about the probability of occurrence of each variant.

According to many numerical experiments adequacy of the cost estimation can be characterized by using beta Pert distribution. Beta Pert distribution can be define by using most optimistic cost $c_o$, most likely cost $c_m$and most pessimistic $c_p$ [3, 12, 1] .

$$\alpha = \frac{4 \cdot (c_m - c_o)}{c_p - c_o}, \quad \beta = 4 - \alpha \tag{6}$$

where $\alpha, \beta$ are parameters of beta distribution.

$$f_{\alpha,\beta}(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\,\Gamma(\beta)} (1 - x)^{\beta - 1} x^{\alpha - 1}, \quad x \in [0,\ 1] \tag{7}$$

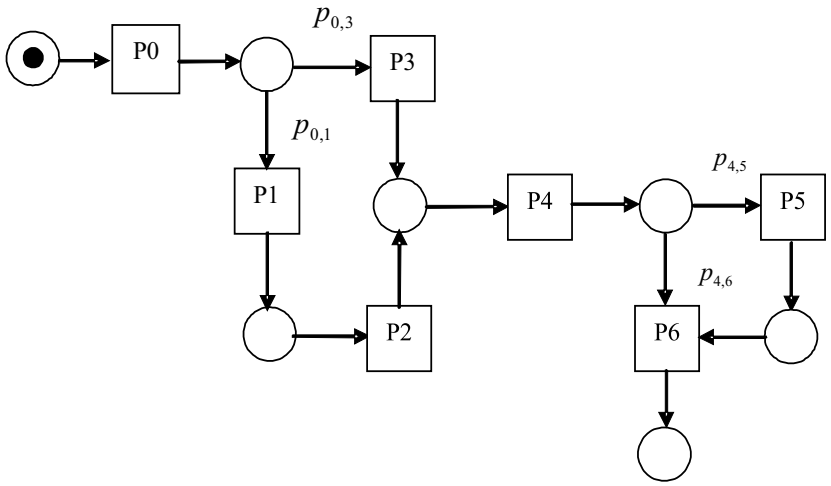



*Figure 1.* Graphical representation of the process

Table I. Tasks description

| No. | Name of tasks | Costs [PLN] | Remarks: |
|---|---|---|---|
| 1 | **P0** | 217.500 | deterministic task |
| 2 | **P1** | 132.000 | alternative task with P3 |
| 3 | **P2** | 187.700 | alternative task with P3 |
| 4 | **P3** | 420.000 | alternative task with P1, P2 |
| 5 | **P4** | 261.700 | deterministic task |
| 6 | **P5** | 43.200 | additional task |
| 7 | **P6** | 125.300 | deterministic task |

In calculation it is necessary to use the PDF which is defined on the interval $[c_o,c_p]$ i.e.

$$f(x) = \frac{1}{c_p - c_o} f_{\alpha,\beta}\left(\frac{x - c_o}{c_p - c_o}\right). \tag{8}$$

Beta Pert distribution is widely used to modeling of uncertainty of cost because of it is very intuitive (can be defined using $c_o, c_m, c_p$).

However usually we do not know the numbers $c_o, c_m, c_p$ precisely. However, usually it is possible to estimate upper and lower bounds its values by using expert knowledge.

$$c_o^- \le c_o \le c_o^+, \quad c_m^- \le c_m \le c_m^+, \quad c_p^- \le c_p \le c_p^+ \tag{9}$$

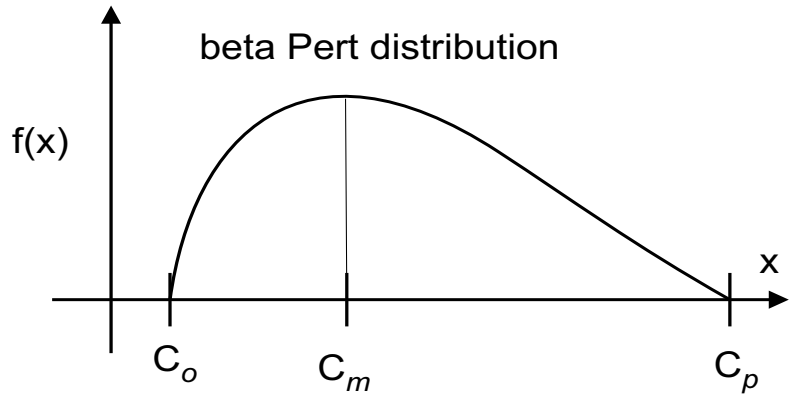



*Figure 2.* Beta distribution

This information is very imprecise. In order to make the calculations more precisely fuzzy numbers can be applied.

Let's assume that we would like to define fuzzy numbers $c_{o,F}, c_{m,F}, c_{p,F}$ which represent the number $c_o, c_m, c_p$. We assume that we know the expert(syrveyor-E1, planner-E2 , site agent-E3) opinions $c_o(\omega_i), c_m(\omega_i), c_p(\omega_i)$ for each expert $\omega_1, \omega_2, ..., \omega_n \in \Omega$. We can treat $\omega_i$ as elementary event of some probability spaceΩ. Examples of such expert opinions are shown in the table 2 and 3. The expert opinions are interval valued $c_p(\omega_i), c_m(\omega_i), c_o(\omega_i)$ or set they are simply numbers i.e.

$$c_p(\omega_i), c_m(\omega_i), c_o(\omega_i) \in I(R) \tag{10}$$

Alpha cut of fuzzy numbers $c_{p,F}, c_{m,F}, c_{o,F}$ can be constructed by using confidence intervals [7]. For given $\alpha$ level the appropriate $\alpha$ cut $c_{p,F,\alpha}, c_{m,F,\alpha}, c_{o,F,\alpha}$ should satisfy the following condition.

$$P\{\omega_i : c_p(\omega_i) \cap c_{p,F,\alpha} \neq \emptyset\} = 1 - \alpha \tag{11}$$

$$P\{\omega_i : c_m(\omega_i) \cap c_{m,F,\alpha} \neq \emptyset\} = 1 - \alpha \tag{12}$$

$$P\{\omega_i : c_o(\omega_i) \cap c_{o,F,\alpha} \neq \emptyset\} = 1 - \alpha \tag{13}$$

In the simplest case it is possible to apply triangular fuzzy numbers which are defined in the following way:

$$c^-_{p,F,0} = min\{c^-_p(\omega_i) : \omega_i \in \Omega\} \tag{14}$$

$$c^+_{p,F,0} = min\{c^+_p(\omega_i) : \omega_i \in \Omega\} \tag{15}$$

$$c^-_{m,F,0} = min\{c^-_m(\omega_i) : \omega_i \in \Omega\}, \tag{16}$$

$$c^+_{m,F,0} = min\{c^+_m(\omega_i) : \omega_i \in \Omega\} \tag{17}$$

$$c^-_{o,F,0} = min\{c^-_o(\omega_i) : \omega_i \in \Omega\}, \tag{18}$$

$$c^+_{o,F,0} = min\{c^+_o(\omega_i) : \omega_i \in \Omega\} \tag{19}$$

The vertex of triangle fuzzy number can be defined using generalized main value.

$$c_{p,F,1} = \sum_{\omega_i \in \Omega} P\{\omega_i\} \cdot mid\left(c_p\left(\omega_i\right)\right), \tag{20}$$

$$c_{m,F,1} = \sum_{\omega_i \in \Omega} P\{\omega_i\} \cdot mid\left(c_p\left(\omega_i\right)\right), \tag{21}$$

$$c_{o,F,1} = \sum_{\omega_i \in \Omega} P\{\omega_i\} \cdot mid\left(c_p\left(\omega_i\right)\right). \tag{22}$$

Table II. Table of cost evaluation

| Process name: P1 | | | |
|---|---|---|---|
| Cost: 217.500 | | | |
| Person: E2 | | | |
| Percent of cost [%] | Min | Mid | Max |
| 75 | | | |
| 80 | | | |
| 85 | | | |
| 90 | | | |
| 95 | X | | |
| 100 | | | |
| 105 | | X | |
| 110 | | | |
| 115 | | | X |
| 120 | | | X |
| 125 | | | |
| 130 | | | |
| 135 | | | |
| 140 | | | |

The fuzzy numbers are which will be used in calculations are given in the table 5 and 6.

Table III. Fuzzy probability of alternative costs

| Probability of occurrence of alternative | | | | |
|---|---|---|---|---|
| lp. | Task | Degree of membership | $p^-$ | $p^+$ |
| | $P_2$ | $\alpha = 0$ | 0.35 | 0.55 |
| | | $\alpha = 1/3$ | 0.3889 | 0.5222 |
| | | $\alpha = 2/3$ | 0.4407 | 0.4944 |
| | | $\alpha = 1$ | 0.4667 | 0.4667 |
| | $P_5$ | $\alpha = 0$ | 0.15 | 0.35 |
| | | $\alpha = 1/3$ | 0.1778 | 0.3111 |
| | | $\alpha = 2/3$ | 0.2148 | 0.2722 |
| | | $\alpha = 1$ | 0.2333 | 0.2333 |

The total cost can be calculated as a sum of random variable with uncertain parameters $\mathbf{h} \in \hat{\mathbf{h}}_\alpha$.

$$c_T(\omega, \mathbf{h}) = \sum_{i=1}^{n} \chi_i(\omega, \mathbf{h}) \cdot c_i(\omega, \mathbf{h}) \tag{23}$$

where $\chi : \Omega \times \hat{\mathbf{h}}_\alpha(\omega, \mathbf{h}) \to \chi(\omega, \mathbf{h}) \in \{0,\ 1\}$, $c_i : \Omega \times \hat{\mathbf{h}}_\alpha \ni (\omega, \mathbf{h}) \to c_i(\omega, \mathbf{h}) \in R$, $c_T : \Omega \times \hat{\mathbf{h}}_\alpha \ni (\omega, \mathbf{h}) \to c_T(\omega, \mathbf{h}) \in R$ are some random variables with uncertain parameters, $\hat{\mathbf{h}}_\alpha = \left[h_1^-, h_1^+\right] \times \left[h_2^-, h_2^+\right] \times ... \times [h_m^-, h_m^+] \subseteq I(R^m)$ is an interval vector.

Extreme values of the risk can be defined in the following way:

$$\hat{R}_\alpha(c_{T,0}) = \left[R_\alpha^-(c_{T,0}), R_\alpha^+(c_{T,0})\right] \tag{24}$$

$$\hat{R}_\alpha(c_{T,0}) = \left\{P\{\omega : c_T(\omega, \mathbf{h}) > c_{T,0}, \omega \in \Omega\} : \mathbf{h} \in \hat{\mathbf{h}}_\alpha\right\} \tag{25}$$

or

$$\hat{R}_\alpha(c_{T,0}) = \left\{1 - \Phi_{c_T}(c_{T,0}, \mathbf{h}) : \mathbf{h} \in \hat{\mathbf{h}}_\alpha\right\} \tag{26}$$

where

$$\Phi_{c_T}(c_{T,0}, \mathbf{h}) = P\{\omega \in \Omega{:}c_T(\omega, \mathbf{h}) \leq c_{T,0}\} \tag{27}$$

Fuzzy membership function $\mu(x|R_F(c_T))$ of the risk of cost $R_F(c_T)$ can be described using the following formula:

Table IV. Fuzzy costs

| **Alpha** | **Task** | **min** | | **mid** | | **max** | |
|---|---|---|---|---|---|---|---|
| | | $c_o^-$ | $c_o^+$ | $c_m^-$ | $c_m^+$ | $c_p^-$ | $c_p^+$ |
| $\alpha = 0$ | | 184,88 | 228,38 | 206,63 | 250,13 | 250,13 | 293,63 |
| $\alpha = 1/3$ | **P0** | 190,91 | 219,91 | 212,66 | 241,66 | 254,95 | 283,95 |
| $\alpha = 2/3$ | | 196,95 | 211,45 | 218,70 | 233,20 | 259,78 | 274,28 |
| $\alpha = 1$ | | 202,99 | 202,99 | 224,74 | 224,74 | 264,61 | 264,61 |
| $\alpha = 0$ | | 112,20 | 125,40 | 125,40 | 151,80 | 151,80 | 178,20 |
| $\alpha = 1/3$ | **P1** | 114,40 | 123,20 | 129,07 | 146,67 | 154,73 | 172,33 |
| $\alpha = 2/3$ | | 116,60 | 121,00 | 132,73 | 141,53 | 157,66 | 166,46 |
| $\alpha = 1$ | | 118,80 | 118,80 | 136,40 | 136,40 | 160,59 | 160,59 |
| $\alpha = 0$ | | 159,55 | 197,09 | 178,32 | 215,86 | 215,86 | 253,40 |
| $\alpha = 1/3$ | **P2** | 164,76 | 189,78 | 182,48 | 207,51 | 221,07 | 246,09 |
| $\alpha = 2/3$ | | 169,97 | 182,48 | 186,65 | 199,16 | 226,28 | 238,79 |
| $\alpha = 1$ | | 175,18 | 175,18 | 190,82 | 190,82 | 231,49 | 231,49 |
| | **P3** | 420,00 | 420,00 | 420,00 | 420,00 | 420,00 | 420,00 |
| $\alpha = 0$ | | 222,45 | 274,79 | 248,62 | 300,96 | 327,13 | 379,47 |
| $\alpha = 1/3$ | **P4** | 232,62 | 267,51 | 258,79 | 293,68 | 334,39 | 369,26 |
| $\alpha = 2/3$ | | 242,79 | 260,23 | 268,96 | 286,40 | 341,66 | 359,05 |
| $\alpha = 1$ | | 252,96 | 252,96 | 279,13 | 279,13 | 348,92 | 348,85 |
| $\alpha = 0$ | | 36,72 | 41,04 | 41,04 | 45,36 | 45,36 | 49,68 |
| $\alpha = 1/3$ | **P5** | 37,44 | 40,32 | 41,76 | 44,64 | 46,08 | 48,96 |
| $\alpha = 2/3$ | | 38,16 | 39,60 | 42,48 | 43,92 | 46,80 | 48,24 |
| $\alpha = 1$ | | 38,88 | 38,88 | 43,20 | 43,20 | 47,52 | 47,52 |
| $\alpha = 0$ | | 93,98 | 119,04 | 119,04 | 131,57 | 144,10 | 169,16 |
| $\alpha = 1/3$ | **P6** | 98,84 | 115,55 | 121,12 | 129,48 | 146,88 | 163,58 |
| $\alpha = 2/3$ | | 103,71 | 112,07 | 123,21 | 127,39 | 149,66 | 158,01 |
| $\alpha = 1$ | | 108,58 | 108,58 | 125,30 | 125,30 | 152,44 | 152,44 |

$$\mu\left(x|R_F\left(c_T\right)\right) = sup\left\{\alpha : x \in \hat{R}_\alpha\left(c_{T,0}\right)\right\} \tag{28}$$

## 6. Approximate algorithm of calculation of fuzzy probability

We can find the approximate values of the fuzzy set$R_F\left(c_T\right)$ using alpha cut method and the formula (26).

1) For the discrete values $0 \leq \alpha_1 \leq \alpha_2 \leq ... \leq \alpha_k \leq 1$ calculate alpha cut of the uncertain parameters $\hat{\mathbf{h}}_{\alpha_1}$, $\hat{\mathbf{h}}_{\alpha_2}$, ..., $\hat{\mathbf{h}}_{\alpha_k}$.

2) Divide the intervals $\hat{h}_{\alpha_i,1}, \hat{h}_{\alpha_i,2}, ..., \hat{h}_{\alpha_i,m}$ into $k$ parts.

3) For each combination of the parameters $\left(h_{\alpha_i,1,j_1}, h_{\alpha_i,2,j_2}, ..., h_{\alpha_i,m,j_m}\right) = \mathbf{h}_{\alpha_i,j_1,j_2,...,j_m}$ calculate the cumulative distribution function $\Phi_{c_T}\left(c, \mathbf{h}_{\alpha_i,j_1,j_2,...,j_m}\right)$. The approximate value of the alpha cut $\hat{R}_{\alpha_i}\left(c_{T,0}\right)$ can be calculated. In the following way.

$$R^{-}_{\alpha_i}\left(c_{T,0}\right) = \tag{29}$$

$$= min\left\{1 - \Phi_{c_T}\left(c_{T,0}, \mathbf{h}_{\alpha_i,j_1,j_2,...,j_m}\right) : j_1, ..., j_m \in \{1, ..., k\}\right\} \tag{30}$$

$$R^{+}_{\alpha_i}\left(c_{T,0}\right) = \tag{31}$$

$$= max\left\{\ 1 - \Phi_{c_T}\left(c_{T,0}, \mathbf{h}_{\alpha_i,j_1,j_2,...,j_m}\right) : j_1, ..., j_m \in \{\ 1, ..., k\}\right\}, \tag{32}$$

4) Approximate value of the fuzzy membership function $\mu\left(x|R_F\left(c_T\right)\right)$ is given by the following formula

$$\mu\left(x|R_F\left(c_{T,0}\right)\right) = sup\left\{\alpha_i : x \in \hat{R}_{\alpha_i}\left(c_{T,0}\right)\right\}, \tag{33}$$

## 7. Computer implementation of the algorithm

Algorithm which was described above was implemented in C++ language and can be run on Linux and Windows. To generation of random numbers GSL library was applied.

The models can be described by using BPFPRAL language (Bętkowski Pownuk Fuzzy Probability Risk Analysis Language) [5] . As and example below is show the code of simulator which is shown on the Fig. 1

```
Node
NumberOfNode 0, NumberOfChildren 2, Children 1 3, Probability 0.415,
IntervalProbability 0.088, xMinMin 198.766, xiMnMax 206.016, xMidMin
215.688, xMidMax 219.313, xMaxMin 231.391, xMaxMax 238.641, ProbabilityGrids 3
End

Node
NumberOfNode 1, NumberOfChildren 1, Children 2, xMinMin 125.761, xMinMax
130.161, xMidMin 133.830, xMidMax 138.230, xMaxMin 147.030, xMaxMax 153.63
End

Node
```

```
NumberOfNode 2, NumberOfChildren 1, Children 4, xMinMin 171.533, xMinMax
177.789, xMidMin 186.136, xMidMax 189.264, xMaxMin 206.983, xMaxMax 213.24
End

Node
PointValue, NumberOfNode 3, NumberOfChildren 1, Children 4, xMinMin 420.0,
xMinMax 420.0, xMidMin 420.0, xMidMax 420.0, xMaxMin 420.0, xMaxMax 420.0,
NumberOfGrid 1
End

Node
NumberOfNode 4, NumberOfChildren 2, Children 5 6, Probability 0.224,
IntervalProbability 0.088, xMinMin 239.159, xMinMax 247.882, xMidMin
252.244, xMidMax 260.967, xMaxMin 282.863, MaxMax 295.948, NumberOfGrid 2,
ProbabilityGrids 3
End

Node
NumberOfNode 5, NumberOfChildren 1, Children 6, xMinMin 38.52, xMinMax
40.68, xMidMin 42.84, xMidMax 44.28, xMaxMin 47.40, xMaxMax 48.84
End

Node
NumberOfNode 6, xMinMin 121.123, xMinMax 125.3, xMidMin 126.344, xMidMax
130.521, xMaxMin 140.267, xMaxMax 146.532, NumberOfGrid 2
End

Results
Xmin 820, Xmax 1120, NumberOfSimulations 10000, NumberOfClasses 20,
NumberOfGrid 2, DistributionType 2
End
```

In presented example only one alpha cut was described. In order to get full description of fuzzy probability model it is necessary to repeat these calculations for each alpha cut.

In the program we can define the upper and lower bounds of $c_o, c_m, c_p$ in the following way:

$$cMinMin \leq c_o \leq cMinMax \tag{34}$$

$$cMidMin \leq c_m \leq cMidMax \tag{35}$$

$$cMaxMin \leq c_p \leq cMaxMax \tag{36}$$

The meaning of other instructions is explaind in the BPFPRAL user manual.

## 8. Numerical results of the calculations

For the example which is shown on the Fig. 1 and is also described in the BPFPRAL language above. In the numerical experiment 10000 Monte Carlo simulations was used for each combination of uncertain parameters in each alpha cut. Extreme values of risk

and probability density function of cost were calculated by using 262144 combinations of uncertain parameters.

Table V. Numerical results

| **Cost** | **Probability** | | | | | | |
|---|---|---|---|---|---|---|---|
| | $\alpha = 0$ | | $\alpha = 1/3$ | | $\alpha = 2/3$ | | $\alpha = 1$ |
| | Min | Max | Min | Max | Min | Max | |
| 8350 - 8500 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8500 - 8650 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8650 - 8800 | 0,989 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8800 - 8950 | 0,941 | 1 | 0,997 | 1 | 0,997 | 1 | 0,997 |
| 8950 - 9100 | 0,784 | 1 | 0,954 | 1 | 0,964 | 1 | 0,964 |
| 9100 - 9250 | 0,528 | 1 | 0,808 | 1 | 0,85 | 1 | 0,85 |
| 9250- 9400 | 0,172 | 1 | 0,515 | 1 | 0,719 | 1 | 0,719 |
| 9400 - 9550 | 0,006 | 1 | 0,179 | 1 | 0,481 | 0,989 | 0,634 |
| 9550 - 9700 | 0 | 0.992 | 0,019 | 0,99 | 0,102 | 0,824 | 0,574 |
| 9700 - 9850 | 0 | 0.927 | 0 | 0,78 | 0,007 | 0,531 | 0,531 |
| 9850 - 10000 | 0 | 0,691 | 0 | 0,514 | 0 | 0,514 | 0,514 |
| 10000 - 10150 | 0 | 0,406 | 0 | 0,406 | 0 | 0,406 | 0,406 |
| 10150 - 10300 | 0 | 0,259 | 0 | 0,259 | 0 | 0,259 | 0,259 |
| 10300 - 10450 | 0 | 0,176 | 0 | 0,176 | 0 | 0,176 | 0,176 |
| 10450 - 10600 | 0 | 0,111 | 0 | 0,111 | 0 | 0,111 | 0,111 |
| 10600 - 10750 | 0 | 0,031 | 0 | 0,031 | 0 | 0,031 | 0,031 |
| 10750 -10900 | 0 | 0,009 | 0 | 0,009 | 0 | 0,009 | 0,009 |
| 10900 - 11050 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11050 - 11200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The envelopes of the risk curves for particular alpha level equal to 0.33 are shown below.

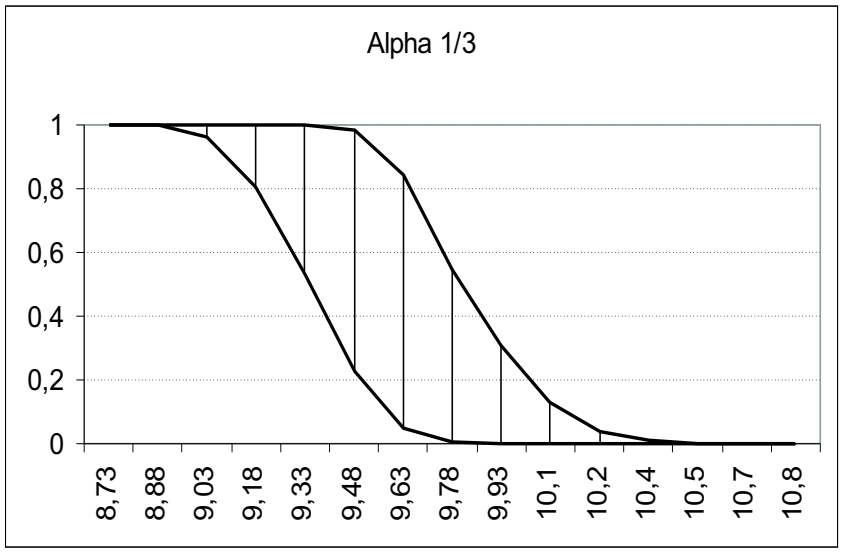



*Figure 3.* Uncertain risc curve for $\alpha = 1/3$

Now we can show the shape of fuzzy risk surfaces for particular alpha levels on the Fig. 4, 5, 8 .

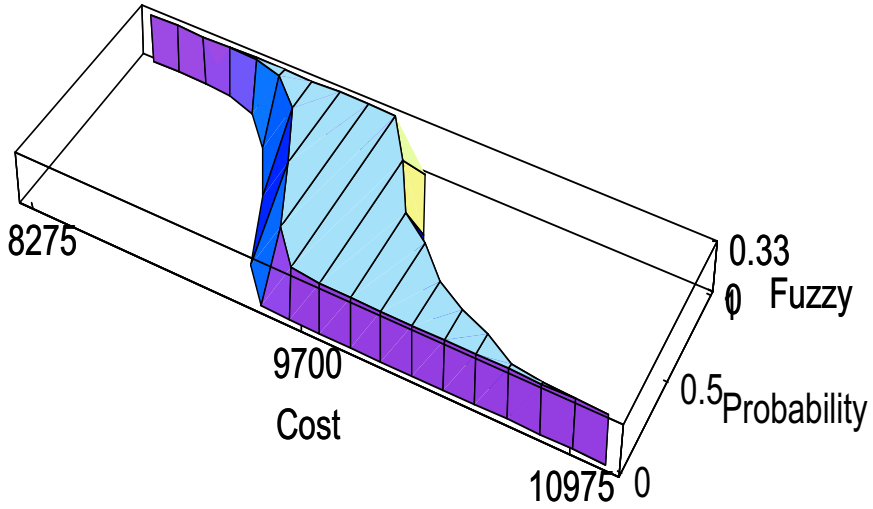



*Figure 4.* Fuzzy probability surface for $\alpha = 1/3$

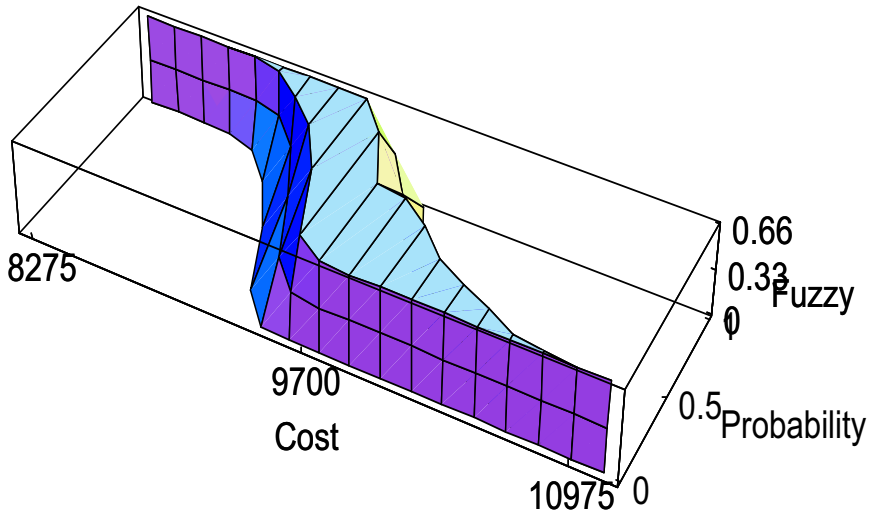



*Figure 5.* Fuzzy probability surface for $\alpha = 2/3$

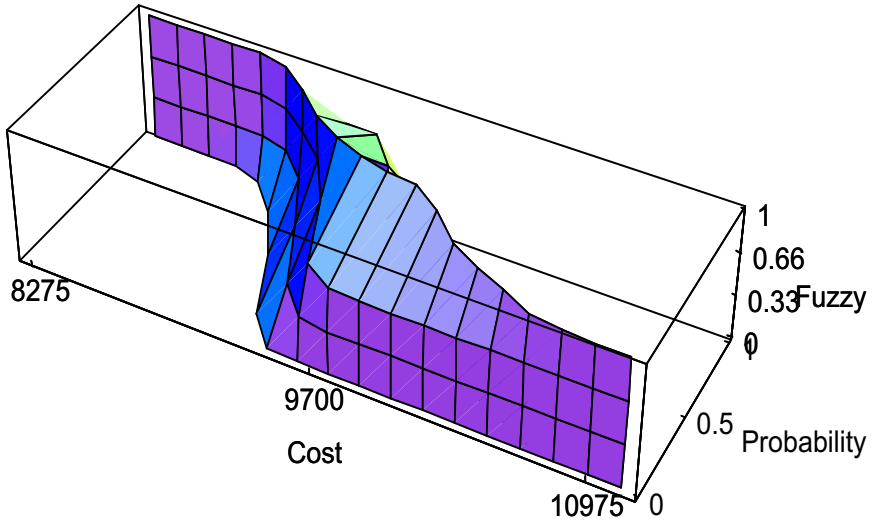



*Figure 6.* Fuzzy probability surface for $\alpha = 1$

## 9. Conclusions

Presented method allows estimating the direct cost risk of civil engineering projects in the case when there are no credible data. In presented algorithm the costs can be deterministic, probabilistic, fuzzy number. It is also possible to take into account the cost which is modeled by probability density function with fuzzy parameters. Unfortunately, at this moment the computational complexity of the algorithm grows exponentially with respect to the number of the fuzzy parameters. The method shows the relation between the assumed maximal direct costs, the risk of overrun and the uncertainty of the statistical data.

## References


1. AbouRikz, S.M., Halpin, D. and Wilson, J.: Fitting beta distribution based simple data, *Journal of Construction Engineering and Management*, **120**(2):288-305, 1993.
2. Akintola, A.: Analysis of factors influencing project cost estimating practice. *Construction Management and Economics*, **18**(1):77-89, 2000.
3. Battersy, A.: *Network Analiz for Planning and Scheduling*, 3rd End. Macmillan, London, 1970.
4. Berleant, D., Cheong, M.-P., Chu, Ch., Guan Y., Kamal A., Shedble, G., Ferson, S., Peters and James, F.: Dependable Handling of Uncertainty, *Reliable Computing*, **9**(6):407-418, 2003.
5. Bętkowski, M. and Pownuk, A.: *BPFPRAL ver. 1.8.2 - user manual*, Gliwice, Poland, 2004.
6. Bizon-Górecka, J.: *Risk management methodology in construction production*, University of Technology and Agriculture in Bydygoszcz, Bydygoszcz, Poland, 1998 (in Polish).
7. Dubois, D., Foulloy, L., Mauris, G. and Prade, H.: Probability-Possibility Transformations, Triangular Fuzzy Sets and Probabilistic Inequalitiesy. *Reliable Computing*, **9**(6):273-29, 2004.
8. *Informacyjny zestaw cen czynnikow produkcji budowlanej*, I kwartal 2004, Orgbud Serwis, Poznan 2004 (in Polish).
9. *Informacyjny Zestaw Wskanikw Nakadw na Obiekty Budowlane I kwartal 2004*, Orgbud Serwis, Poznan 2004 (in Polish).
10. Kaming, P. F., Olomolaiye, P. O., Holt, G. D., Harris F. C.: Factors influencing construction time and cost overruns on high-rise projects in Indonesia, *Construction Management and Economics*, **15**(1):83 - 94, 1997.
11. MSWIA, *Katalog Nakladow Rzeczowych nr 202 Konstrukcje budowlane -tom 1*, , Warszawa 1995 (in Polish).
12. Riggs, L.S.: Numierical approach for generating beta random variables, *Journal of Computing in Civil Engineering*, **3**(2):183-91, 1989.
13. Stowarzyszenie Biur Kosztorysowania Budowlanego, *Srodowiskowe Metody Kosztorysowania Robt Budowlanych* , Warszawa, grudzień 2001 (in Polish).
14. Starke, P. H.: Petrinetze. Deutscher Verlag der Wissenschaften, DDR, Berlin, 1980 (in German).

# Towards Combining Probabilistic and Interval Uncertainty in Engineering Calculations

S. A. Starks, V. Kreinovich, L. Longpré, M. Ceberio, G. Xiang, R. Araiza, J. Beck, R. Kandathi, A. Nayak and R. Torres
*NASA Pan-American Center for Earth and Environmental Studies (PACES), University of Texas, El Paso, TX 79968, USA (*vladik@cs.utep.edu*)*

**Abstract.** In many engineering applications, we have to combine probabilistic and interval errors. For example, in environmental analysis, we observe a pollution level $x(t)$ in a lake at different moments of time $t$, and we would like to estimate standard statistical characteristics such as mean, variance, autocorrelation, correlation with other measurements. In environmental measurements, we often only know the values with interval uncertainty. We must therefore modify the existing statistical algorithms to process such interval data. Such modification are described in this paper.

**Keywords:** probabilistic uncertainty, interval uncertainty, engineering calculations

## 1. Formulation of the Problem

*Computing statistics is important.* In many engineering applications, we are interested in computing statistics. For example, in environmental analysis, we observe a pollution level $x(t)$ in a lake at different moments of time $t$, and we would like to estimate standard statistical characteristics such as mean, variance, autocorrelation, correlation with other measurements. For each of these characteristics $C$, there is an expression $C(x_1, \ldots, x_n)$ that enables us to provide an estimate for $C$ based on the observed values $x_1, \ldots, x_n$. For example, a reasonable statistic for estimating the mean value of a probability distribution is the population average $E(x_1, \ldots, x_n) = \dfrac{1}{n}(x_1 + \ldots + x_n)$; a reasonable statistic for estimating the variance $V$ is the population variance $V(x_1, \ldots, x_n) = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} (x_i - \bar{x})^2$, where $\bar{x} \stackrel{\text{def}}{=} \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} x_i$.

*Interval uncertainty.* In environmental measurements, we often only know the values with interval uncertainty. For example, if we did not detect any pollution, the pollution value $v$ can be anywhere between 0 and the sensor's detection limit $DL$. In other words, the only information that we have about $v$ is that $v$ belongs to the interval $[0, DL]$; we have no information about the probability of different values from this interval.

Another example: to study the effect of a pollutant on the fish, we check on the fish daily; if a fish was alive on Day 5 but dead on Day 6, then the only information about the lifetime of this fish is that it is somewhere within the interval $[5, 6]$; we have no information about the probability of different values within this interval.

In non-destructive testing, we look for outliers as indications of possible faults. To detect an outlier, we must know the mean and standard deviation of the normal values – and

these values can often only be measured with interval uncertainty (see, e.g., (Rabinovich, 1993; Osegueda et al., 2002)). In other words, often, we know the result $\widetilde{x}$ of measuring the desired characteristic $x$, and we know the upper bound $\Delta$ on the absolute value $|\Delta x|$ of the measurement error $\Delta x \stackrel{\rm def}{=} \widetilde{x} - x$ (this upper bound is provided by the manufacturer of the measuring instrument), but we have no information about the probability of different values $\Delta x \in [-\Delta, \Delta]$. In such situations, after the measurement, the only information that we have about the actual value $x$ of the measured quantity is that this value belongs to interval $[\widetilde{x} - \Delta, \widetilde{x} + \Delta]$.

In geophysics, outliers should be identified as possible locations of minerals; the importance of interval uncertainty for such applications was emphasized in (Nivlet et al., 2001; Nivlet et al., 2001a). Detecting outliers is also important in bioinformatics (Shmulevich and Zhang, 2002).

In bioinformatics and bioengineering applications, we must solve systems of linear equations in which coefficients come from experts and are only known with interval uncertainty; see, e.g., (Zhang et al., 2004).

In biomedical systems, statistical analysis of the data often leads to improvements in medical recommendations; however, to maintain privacy, we do not want to use the exact values of the patient's parameters. Instead, for each parameter, we select fixed values, and for each patient, we only keep the corresponding range. For example, instead of keeping the exact age, we only record whether the age is between 0 and 10, 10 and 20, 20 and 30, etc. We must then perform statistical analysis based on such interval data; see, e.g., (Kreinovich and Longpré, 2003; Xiang et al., 2004).

*Estimating statistics under interval uncertainty: a problem.* In all such cases, instead of the actual values $x_1, \ldots, x_n$, we only know the intervals $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1], \ldots, \mathbf{x}_n = [\underline{x}_n, \overline{x}_n]$ that contain the (unknown) actual values of the measured quantities. For different values $x_i \in \mathbf{x}_i$, we get, in general, different values of the corresponding statistical characteristic $C(x_1, \ldots, x_n)$. Since all values $x_i \in \mathbf{x}_i$ are possible, we conclude that all the values $C(x_1, \ldots, x_n)$ corresponding to $x_i \in \mathbf{x}_i$ are possible estimates for the corresponding statistical characteristic. Therefore, for the interval data $\mathbf{x}_1, \ldots, \mathbf{x}_n$, a reasonable estimate for the corresponding statistical characteristic is the range

$$C(\mathbf{x}_1, \ldots, \mathbf{x}_n) \stackrel{\rm def}{=} \{C(x_1, \ldots, x_n) \,|\, x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}.$$

We must therefore modify the existing statistical algorithms so that they would be able to estimate such ranges. This is a problem that we solve in this paper.

*This problem is a part of a general problem.* The above range estimation problem is a specific problem related to a combination of interval and probabilistic uncertainty. Such problems – and their potential applications – have been described, in a general context, in the monographs (Kuznetsov, 1991; Walley, 1991); for further developments, see, e.g., (Rowe, 1988; Williamson, 1990; Berleant, 1993; Berleant, 1996; Berleant and Goodman-Strauss, 1998; Ferson et al., 2001; Ferson, 2002; Berleant et al., 2003; Lodwick and Jamison, 2003; Moore and Lodwick, 2003; Regan et al., (in press)) and references therein.

## 2. Analysis of the Problem

*Mean.* Let us start our discussion with the simplest possible characteristic: the mean. The arithmetic average $E$ is a monotonically increasing function of each of its $n$ variables $x_1, \ldots, x_n$, so its smallest possible value $\underline{E}$ is attained when each value $x_i$ is the smallest possible ($x_i = \underline{x}_i$) and its largest possible value is attained when $x_i = \overline{x}_i$ for all $i$. In other words, the range $\mathbf{E}$ of $E$ is equal to $[E(\underline{x}_1, \ldots, x_n), E(\overline{x}_1, \ldots, \overline{x}_n)]$. In other words, $\underline{E} = \dfrac{1}{n}(\underline{x}_1 + \ldots + \underline{x}_n)$ and $\overline{E} = \dfrac{1}{n}(\overline{x}_1 + \ldots + \overline{x}_n)$.

*Variance: computing the exact range is difficult.* Another widely used statistic is the variance. In contrast to the mean, the dependence of the variance $V$ on $x_i$ is not monotonic, so the above simple idea does not work. Rather surprisingly, it turns out that the problem of computing the exact range for the variance over interval data is, in general, NP-hard (Ferson et al., 2002; Kreinovich, (in press)) which means, crudely speaking, that the worst-case computation time grows exponentially with $n$. Moreover, if we want to compute the variance range with a given accuracy $\varepsilon$, the problem is still NP-hard. (For a more detailed description of NP-hardness in relation to interval uncertainty, see, e.g., (Kreinovich et al., 1997).)

*Linearization.* ¿From the practical viewpoint, often, we may not need the exact range, we can often use approximate linearization techniques. For example, when the uncertainty comes from measurement errors $\Delta x_i$, and these errors are small, we can ignore terms that are quadratic (and of higher order) in $\Delta x_i$ and get reasonable estimates for the corresponding statistical characteristics. In general, in order to estimate the range of the statistic $C(x_1, \ldots, x_n)$ on the intervals $[\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n]$, we expand the function $C$ in Taylor series at the midpoint $\widetilde{x}_i \stackrel{\text{def}}{=} (\underline{x}_i + \overline{x}_i)/2$ and keep only linear terms in this expansion. As a result, we replace the original statistic with its linearized version $C_{\text{lin}}(x_1, \ldots, x_n) = C_0 - \sum\limits_{i=1}^{n} C_i \cdot \Delta x_i$, where $C_0 \stackrel{\text{def}}{=} C(\widetilde{x}_1, \ldots, \widetilde{x}_n)$, $C_i \stackrel{\text{def}}{=} \dfrac{\partial C}{\partial x_i}(\widetilde{x}_1, \ldots, \widetilde{x}_n)$, and $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$. For each $i$, when $x_i \in [\underline{x}_i, \overline{x}_i]$, the difference $\Delta x_i$ can take all possible values from $-\Delta_i$ to $\Delta_i$, where $\Delta_i \stackrel{\text{def}}{=} (\overline{x}_i - \underline{x}_i)/2$. Thus, in the linear approximation, we can estimate the range of the characteristic $C$ as $[C_0 - \Delta, C_0 + \Delta]$, where $\Delta \stackrel{\text{def}}{=} \sum\limits_{i=1}^{n} |c_i| \cdot \Delta_i$.

In particular, for variance, $C_i = \dfrac{\partial V}{\partial x_i} = \dfrac{2}{n}(\widetilde{x}_i - \bar{\widetilde{x}})$, where $\bar{\widetilde{x}}$ is the average of the midpoints $\widetilde{x}_i$. So, here, $V_0 = \dfrac{1}{n} \sum\limits_{i=1}^{n} (\widetilde{x}_i - \bar{\widetilde{x}})^2$ is the variance of the midpoint values $\widetilde{x}_1, \ldots, \widetilde{x}_n$, and $\Delta = \dfrac{2}{n} \sum\limits_{i=1}^{n} |\widetilde{x}_i - \bar{\widetilde{x}}| \cdot \Delta_i$.

It is worth mentioning that for the variance, the ignored quadratic term is equal to $\frac{1}{n}\sum_{i=1}^{n}(\Delta x_i)^2 - (\overline{\Delta x})^2$, where $\overline{\Delta x} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\Delta x_i$, and therefore, can be bounded by 0 from below and by $\Delta^{(2)} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\Delta_i^2$ from above. Thus, the interval $[V_0 - \Delta, V_0 + \Delta + \Delta^{(2)}]$ is a guaranteed enclosure for $\mathbf{V}$.

*Linearization is not always acceptable.* In some cases, linearized estimates are not sufficient: the intervals may be wide so that quadratic terms can no longer be ignored, and/or we may be in a situation where we want to guarantee that, e.g., the variance does not exceed a certain required threshold. In such situations, we need to get the exact range – or at least an enclosure for the exact range.

Since, even for as simple a characteristic as variance, the problem of computing its exact range is NP-hard, we cannot have a feasible-time algorithm that always computes the exact range of these characteristics. Therefore, we must look for the reasonable classes of problems for which such algorithms are possible. Let us analyze what such classes can be.

*First class: narrow intervals.* As we have just mentioned, the computational problems become more complex when we have wider intervals. In other words, when intervals are narrower, the problems are easier. How can we formalize "narrow intervals"? One way to do it is as follows: the actual values $x_1, \ldots, x_n$ of the measured quantity are real numbers, so they are usually different. The data intervals $\mathbf{x}_i$ contain these values. When the intervals $\mathbf{x}_i$ surrounding the corresponding points $x_i$ are narrow, these intervals do not intersect. When their widths becomes larger than the distance between the original values, the intervals start intersecting.

*Definition.* Thus, the ideal case of "narrow intervals" can be described as the case when no two intervals $\mathbf{x}_i$ intersect.

*Second class: slightly wider intervals.* Slightly wider intervals correspond to the situation when few intervals intersect, i.e., when for some integer $K$, no set of $K$ intervals has a common intersection.

*Third class: single measuring instrument.* Since we want to find the exact range $\mathbf{C}$ of a statistic $C$, it is important not only that intervals are relatively narrow, it is also important that they are approximately of the same size: otherwise, if, say, $\Delta x_i^2$ is of the same order as $\Delta x_j$, we cannot meaningfully ignore $\Delta x_i^2$ and retain $\Delta x_j$. In other words, the interval data set should not combine high-accurate measurement results (with narrow intervals) and low-accurate results (with wide intervals): all measurements should have been done by a single measuring instrument (or at least by several measuring instruments of the same type).

How can we describe this mathematically? A clear indication that we have two measuring instruments (MI) of different quality is that one interval is a proper subset of the other one: $[\underline{x}_i, \overline{x}_i] \subseteq (\underline{x}_j, \overline{x}_j)$.

*Definition.* So, if all pairs of non-degenerate intervals satisfy the following *subset property* $[\underline{x}_i, \overline{x}_i] \not\subseteq (\underline{x}_j, \overline{x}_j)$, we say that the measurements were done *by a single MI.*

*Comment.* This restriction only refers to inexact measurement results, i.e., to non-degenerate intervals. In additional to such interval values, we may have exact values (degenerate intervals). For example, in geodetic measurements, we may select some point ("benchmark") as a reference point, and describe, e.g., elevation of each point relative to this benchmark. For the benchmark point itself, the relative elevation will be therefore exactly equal to 0. When we want to compute the variance of elevations, we want to include the benchmark point too. ¿From this viewpoint, when we talk about measurements made by a single measuring instrument, we may allow degenerate intervals (i.e., exact numbers) as well.

A reader should be warned that in the published algorithms describing a single MI case (Xiang et al., 2004), we only considered non-degenerate intervals. However, as one can easily see from the published proofs (and from the idea of these proofs, as described below), these algorithms can be easily modified to incorporate possible exact values $x_i$.

*Fourth class: same accuracy measurement.* In some situations, it is also reasonable to consider a specific case of the single MI case when all measurements are performed with exactly the same accuracy, i.e., in mathematical terms, when all non-degenerate intervals $[\underline{x}_i, \overline{x}_i]$ have exactly the same half-width $\Delta_i = \dfrac{1}{2} \cdot (\overline{x}_i - \underline{x}_i)$.

*Fifth class: several MI.* After the single MI case, the natural next case is when we have several MI, i.e., when our intervals are divided into several subgroups each of which has the above-described subset property.

*Sixth class: privacy case.* Although these definitions are in terms of measurements, they make sense for other sources of interval data as well. For example, for privacy data, intervals either coincide (if the value corresponding to the two patients belongs to the same range) or are different, in which case they can only intersect in one point. Similarly to the above situation, we also allow exact values in addition to ranges; these values correspond, e.g., to the exact records made in the past, records that are already in the public domain.

*Definition.* We will call interval data with this property – that every two non-degenerate intervals either coincide or do nor intersect – *privacy case.*

*Comment.* For the privacy case, the subset property is satisfied, so algorithms that work for a single MI case work for the privacy case as well.

*Seventh class: non-detects.* Similarly, if the only source of interval uncertainty is detection limits, i.e., if every measurement result is either an exact value or a *non-detect*, i.e., an interval $[0, DL_i]$ for some real number $DL_i$ (with possibly different detection limits for different

sensors), then the resulting non-degenerate intervals also satisfy the subset property. Thus, algorithms that work for a single MI case work for this "non-detects" case as well.

Also, an algorithm that works for the general privacy case also works for the non-detects case when all sensors have the same detection limit $DL$.

## 3. Results

*Variance: known results.* The lower bound $\underline{V}$ can be always computed in time $O(n \cdot \log(n))$ (Granvilliers et al., 2004).

Computing $\overline{V}$ is, in general, an NP-hard problem; $\overline{V}$ can be computed in time $2^n$. If intervals do not intersect (and even if "narrowed" intervals $[\widetilde{x}_i - \Delta_i/n, \widetilde{x}_i + \Delta_i/n]$ do not intersect), we can compute $\overline{V}$ in time $O(n \cdot \log(n))$ (Granvilliers et al., 2004). If for some $K$, no more than $K$ interval intersect, we can compute $\overline{V}$ in time $O(n^2)$ (Ferson et al., 2002; Kreinovich, (in press)).

For the case of a single MI, $\overline{V}$ can be computed in time $O(n \cdot \log(n))$; for $m$ MIs, we need time $O(n^{m+1})$ (Xiang et al., 2004).

*Variance: main ideas behind the known results.* The algorithm for computing $\underline{V}$ is based on the fact that when a function $V$ attains a minimum on an interval $[\underline{x}_i, \overline{x}_i]$, then either $\dfrac{\partial V}{\partial x_i} = 0$, or the minimum is attained at the left endpoint $x_i = \underline{x}_i$ – then $\dfrac{\partial V}{\partial x_i} > 0$, or $x_i = \overline{x}_i$ and $\dfrac{\partial V}{\partial x_i} < 0$. Since the partial derivative is equal to $(2/n) \cdot (x_i - \bar{x})$, we conclude that either $x_i = \bar{x}$, or $x_i = \underline{x}_i > \bar{x}$, or $x_i = \underline{x}_i < \bar{x}$. Thus, if we know where $\bar{x}$ is located in relation to all the endpoints, we can uniquely determine the corresponding minimizing value $x_i$ for every $i$: if $\overline{x}_i \le \bar{x}$ then $x_i = \overline{x}_i$; if $x_i \le \underline{x}_i$, then $x_i = \underline{x}_i$; otherwise, $x_i = \bar{x}$. The corresponding value $\bar{x}$ can be found from the condition that $\bar{x}$ is the average of all the selected values $x_i$.

So, to find the smallest value of $V$, we can sort all $2n$ bounds $\underline{x}_i, \overline{x}_i$ into a sequence $x_{(1)} \le x_{(2)} \le \ldots$; then, for each zone $[x_{(k)}, x_{(k+1)}]$, we compute the corresponding values $x_i$, find their variance $V_k$, and then compute the smallest of these variances $V_k$.

For each of $2n$ zones, we need $O(n)$ steps, so this algorithm requires $O(n^2)$ steps. It turns out that the function $V_k$ decreases until the desired $k$ then increases, so we can use binary search – that requires that we only analyze $O(\log(n))$ zones – find the appropriate zone $k$. As a result, we get an $O(n \cdot \log(n))$ algorithm.

For $\overline{V}$, to the similar analysis of the derivatives, we can add the fact that the second derivative of $V$ is $\ge 0$, so there cannot be a maximum inside the interval $[\underline{x}_i, \overline{x}_i]$. So, in principle, to compute $\overline{V}$, it is sufficient to consider all $2^n$ combinations of endpoints. When few intervals intersect, then, when $\overline{x}_i \le \bar{x}$, we take $x_i = \underline{x}_i$; when $\bar{x} \le \underline{x}_i$, we take $x_i = \overline{x}_i$; otherwise, we must consider both possibilities $x_i = \underline{x}_i$ and $x_i = \overline{x}_i$.

For the case of a single MI, we can sort the intervals in lexicographic order: $\mathbf{x}_i \le \mathbf{x}_j$ if and only if $\underline{x}_i < \underline{x}_j$ or ($\underline{x}_i = \underline{x}_j$ and $\overline{x}_i \le \overline{x}_j$). It can be proven that the maximum of $V$ is

always attained if for some $k$, the first $k$ values $x_i$ are equal to $\underline{x}_i$ and the next $n-k$ values $x_i$ are equal to $\overline{x}_i$. This result is proven by reduction to a contradiction: if in the maximizing vector $x=(x_1,\ldots,x_n)$, some $\overline{x}_i$ is preceding some $\underline{x}_j$, $i<j$, then we can increase $V$ while keeping $E$ intact – which is in contradiction with the assumption that the vector $x$ was maximizing. Specifically, to increase $V$, we can do the following: if $\Delta_i \le \Delta_j$, we replace $\overline{x}_i$ with $\underline{x}_i = \overline{x}_i - 2\Delta_i$ and $\underline{x}_j$ with $\underline{x}_j + 2\Delta_i$; otherwise, we replace $\underline{x}_j$ with $\overline{x}_j = \underline{x}_j + 2\Delta_j$ and $\overline{x}_i$ with $\overline{x}_i - 2\Delta_j$.

As a result, to find the maximum of $V$, it is sufficient to sort the intervals (this takes $O(n\cdot\log(n))$ time), and then, for different values $k$, check vectors $(\underline{x}_1,\ldots,\underline{x}_k,\overline{x}_{k+1},\ldots,\overline{x}_n)$. The dependence of $V$ on $k$ is concave, so we can use binary search to find $k$; binary search takes $O(\log(n))$ steps, and for each $k$, we need linear time, so overall, we need time $O(n\cdot\log(n))$.

In case of several MI, we sort intervals corresponding to each of $m$ MI. Then, to find the maximum of $V$, we must find the values $k_1,\ldots,k_m$ corresponding to $m$ MIs. There are $\le n^m$ combinations of $k_i$s, and checking each combination requires $O(n)$ time, so overall, we need time $O(n^{m+1})$.

*Variance: new results.* Sometimes, most of the data is accurate, so among $n$ intervals, only $d \ll n$ are non-degenerate intervals. For example, we can have many accurate values and $m$ non-detects. In this situation, to find the extrema of $V$, we only need to find $x_i$ for $d$ non-degenerate intervals; thus, we only need to consider $2d$ zones formed by their endpoints. Within each zone, we still need $O(n)$ computations to compute the corresponding variance.

So, in this case, to compute $\underline{V}$, we need time $O(n\cdot\log(d))$, and to compute $\overline{V}$, we need $O(n\cdot 2^d)$ steps. If narrowed intervals do not intersect, we need time $O(n\cdot\log(d))$ to compute $\overline{V}$; if for some $K$, no more than $K$ interval intersect, we can compute $\overline{V}$ in time $O(n\cdot d)$.

For the case of a single MI, $\overline{V}$ can be computed in time $O(n\cdot\log(d))$; for $m$ MIs, we need time $O(n\cdot d^m)$.

In addition to new algorithms, we also have a new NP-hardness result. In the original proof of NP-hardness, we have $\widetilde{x}_1=\ldots=\widetilde{x}_n=0$, i.e., all measurement results are the same, only accuracies $\Delta_i$ are different. What if all the measurement results are different? We can show that in this case, computing $\overline{V}$ is still an NP-hard problem: namely, for every $n$-tuple of real numbers $\widetilde{x}_1,\ldots,\widetilde{x}_n$, the problem of computing $\overline{V}$ for intervals $\mathbf{x}_i=[\widetilde{x}_i-\Delta_i,\widetilde{x}_i+\Delta_i]$ is still NP-hard.

To prove this result, it is sufficient to consider $\Delta_i = N\cdot\Delta_i^{(0)}$, where $\Delta_i^{(0)}$ are the values used in the original proof. In this case, we can describe $\Delta x_i = \widetilde{x}_i - x_i$ as $N\cdot\Delta x_i^{(0)}$, where $\Delta_i^{(0)} \in [-\Delta_i^{(0)},\Delta_i^{(0)}]$. For large $N$, the difference between the variance corresponding to the values $x_i=\widetilde{x}_i+N\cdot\Delta x_i^{(0)}$ and $N^2$ times the variance of the values $\Delta x_i^{(0)}$ is bounded by a term proportional to $N$ (and the coefficient at $N$ can be easily bounded). Thus, the difference between $\overline{V}$ and $N^2\cdot\overline{V}^{(0)}$ is bounded by $C\cdot N$ for some known constant $C$. Hence, by computing $\overline{V}$ for sufficiently large $N$, we can compute $\overline{V}^{(0)}$ with a given accuracy $\varepsilon>0$, and we already know that computing $\overline{V}^{(0)}$ with given accuracy is NP-hard. This reduction proves that our new problem is also NP-hard.

*Covariance: known results.* In general, computing the range of covariance $C_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}) \cdot (y_i - \bar{y})$ based on given intervals $\mathbf{x}_i$ and $\mathbf{y}_i$ is NP-hard (Osegueda et al., 2002). When boxes $\mathbf{x}_i \times \mathbf{y}_i$ do not intersect – or if $\geq K$ boxes cannot have a common point – we can compute the range in time $O(n^3)$ (Beck et al., 2004).

The main idea behind this algorithm is to consider the derivatives of $C$ relative to $x_i$ and $y_i$. Then, once we know where the point $(\bar{x}, \bar{y})$ is in relation to $x_i$ and $y_i$, we can uniquely determine the optimizing values $x_i$ and $y_i$ – except for the boxes $\mathbf{x}_i \times \mathbf{y}_i$ that contain $(\bar{x}, \bar{y})$. The bounds $\underline{x}_i$ and $\overline{x}_i$ divide the $x$ axis into $2n+2$ intervals; similarly, the $y$-bounds divide the $y$-axis into $2n+2$ intervals. Combining these intervals, we get $O(n^2)$ zones. Due to the limited intersection property, for each of these zones, we have finitely many ($\leq K$) indices $i$ for which the corresponding box intersects with the zone. For each such box, we may have two different combinations: $(\underline{x}_i, \underline{y}_i)$ and $(\overline{x}_i, \overline{y}_i)$ for $\overline{C}$ and $(\underline{x}_i, \overline{y}_i)$ and $(\overline{x}_i, \underline{y}_i)$ for $\underline{C}$. Thus, we have finitely many ($\leq 2^K$) possible combinations of $(x_i, y_i)$ corresponding to each zone. Hence, for each of $O(n^2)$ zones, it takes $O(n)$ time to find the corresponding values $x_i$ and $y_i$ and to compute the covariance; thus, overall, we need $O(n^3)$ time.

*Covariance: new results.* If $n - d$ measurement results $(x_i, y_i)$ are exact numbers and only $d$ are non-point boxes, then we only need $O(d^2)$ zones, so we can compute the range in time $O(n \cdot d^2)$.

In the privacy case, all boxes $\mathbf{x}_i \times \mathbf{y}_i$ are either identical or non-intersecting, so the only case when a box intersects with a zone is when the box coincides with this zone. For each zone $k$, there may be many ($n_k$) such boxes, but since they are all identical, what matters for our estimates is how many of them are assigned one of the possible $(x_i, y_i)$ combinations and how many the other one. There are only $n_k + 1$ such assignments: 0 to first combination and $n_k$ to second, 1 to first and $n_k - 1$ to second, etc. Thus, the overall number of all combinations for all the zones $k$ is $\sum\limits_k n_k + \sum\limits_k 1$, where $\sum n_k = n$ and $\sum\limits_k 1$ is the overall number of zones, i.e., $O(n^2)$. For each combination of $x_i$ and $y_i$, we need $O(n)$ steps. Thus, in the privacy case, we can compute both $\underline{C}$ and $\overline{C}$ in time $O(n^2) \cdot O(n) = O(n^3)$ (or $O(n \cdot d^2)$ if only $d$ boxes are non-degenerate).

Another polynomial-time case is when all the measurements are exactly of the same accuracy, i.e., when all non-degenerate $x$-intervals have the same half-width $\Delta_x$, and all non-degenerate $y$-intervals have the same half-width $\Delta_y$. In this case, e.g., for $\overline{C}$, if we have at least two boxes $i$ and $j$ intersecting with the same zone, and we have $(x_i, y_i) = (\underline{x}_i, \underline{y}_i)$ and $(x_j, y_j) = (\overline{x}_j, \overline{y}_j)$, then we can swap $i$ and $j$ assignments – i.e., make $(x_i', y_i') = (\overline{x}_i, \overline{y}_i)$ and $(x_j', y_j') = (\underline{x}_j, \underline{y}_j)$ – without changing $\bar{x}$ and $\bar{y}$. In this case, the only change in $C_{xy}$ comes from replacing $x_i \cdot y_i + x_j \cdot y_j$. It is easy to see that the new value $C$ is larger than the old value if and only if $z_i > z_j$, where $z_i \stackrel{\text{def}}{=} \widetilde{x}_i \cdot \Delta_y + \widetilde{y}_i \cdot \Delta_x$. Thus, in the true maximum, whenever we assign $(\underline{x}_i, \underline{y}_i)$ to some $i$ and $(\overline{x}_i, \overline{y}_j)$ to some $j$, we must have $z_i \leq z_j$. So, to get the largest value of $C$, we must sort the indices by $z_i$, select a threshold $t$, and assign $(\underline{x}_i, \underline{y}_i)$ to all the boxes with $z_i \leq t$ and $(\overline{x}_j, \overline{y}_j)$ to all the boxes $j$ with $z_j > t$. If $n_k \leq n$ denotes the overall number of all the boxes that intersect with $k$-th zone, then we have $n_k + 1$ possible

choices of thresholds, hence $n_k + 1$ such assignments. For each of $O(n^2)$ zones, we test $\le n$ assignments; testing each assignment requires $O(n)$ steps, so overall, we need time $O(n^4)$.

If only $d$ boxes are non-degenerate, we only need time $O(n \cdot d^3)$.

*Detecting outliers: known results.* Traditionally, in statistics, we fix a value $k_0$ (e.g., 2 or 3) and claim that every value $x$ outside the $k_0$-sigma interval $[L, U]$, where $L \stackrel{\text{def}}{=} E - k_0 \cdot \sigma$, $U \stackrel{\text{def}}{=} E + k_0 \cdot \sigma$ (and $\sigma \stackrel{\text{def}}{=} \sqrt{V}$), is an outlier; thus, to detect outliers based on interval data, we must know the ranges of $L$ and $U$. It turns out that we can always compute $\underline{U}$ and $\overline{L}$ in $O(n^2)$ time (Kreinovich et al., 2003a; Kreinovich et al., 2004). In contrast, computing $\overline{U}$ and $\underline{L}$ is NP-hard; in general, it can be done in $2^n$ time, and in quadratic time if $\le K$ intervals intersect (even if $\le K$ appropriately narrowed intervals intersect) (Kreinovich et al., 2003a; Kreinovich et al., 2004).

For every $x$, we can also determine the "degree of outlier-ness" $R$ as the smallest $k_0$ for which $x \notin [E - k_0 \cdot \sigma, E + k_0 \cdot \sigma]$, i.e., as $|x - E|/\sigma$. It turns out that $\overline{R}$ can be always computed in time $O(n^2)$; the lower bound $\underline{R}$ can be also computed in quadratic time if $\le K$ narrowed intervals intersect (Kreinovich et al., 2003a).

*Detecting outliers: new results.* Similar to the case of variance, if we only have $d \ll n$ non-degenerate intervals, then instead of $O(n^2)$ steps, we only need $O(n \cdot d)$ steps (and instead of $2^n$ steps, we only need $O(n \cdot 2^d)$ steps).

For the case of a single MI, similarly to variance, we can prove that the maximum of $U$ and the minimum of $L$ are attained at one of the vectors $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$; actually, practically the same proof works, because increasing $V$ without changing $E$ increases $U = E + k_0 \cdot \sqrt{V}$ as well. Thus, to find $\overline{U}$ and $\underline{L}$, it is sufficient to check $n$ such sequences; checking each sequence requires $O(n)$ steps, so overall, we need $O(n^2)$ time. For $m$ MI, we need $O(n^{m+1})$ time.

If only $d \ll n$ intervals are non-degenerate, then we need, correspondingly, time $O(n \cdot d)$ and $O(n \cdot d^m)$.

*Moments.* For population moments $\frac{1}{n} \cdot \sum_{i=1}^{n} x_i^q$, known interval bounds on $x^q$ leads to exact range. For central moments $M_q = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - \bar{x})^q$, we have the following results (Kreinovich et al., 2004a). For even $q$, the lower endpoint $\underline{M}_q$ can be computed in $O(n^2)$ time; the upper endpoint $\overline{M}_q$ can always be computed in time $O(2^n)$, and in $O(n^2)$ time if $\le K$ intersect. For odd $q$, if $\le K$ intervals do not intersect, we can compute both $\underline{M}_q$ and $\overline{M}_q$ in $O(n^3)$ time.

If only $d$ out of $n$ intervals are non-degenerate, then we need $O(n \cdot 2^d)$ time instead of $O(2^n)$, $O(n \cdot d)$ instead of $O(n^2)$, and $O(n \cdot d^2)$ instead of $O(n^3)$.

For even $q$, we can also consider the case of a single MI. The arguments work not only for $M_q$, but also for a generalized central moment $M_\psi \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \psi(x_i - E)$ for an arbitrary convex function $\psi(x) \ge 0$ for which $\psi(0) = 0$ and $\psi''(x) > 0$ for all $x \ne 0$. Let us first show

that the maximum cannot be attained inside an interval $[\underline{x}_i, \overline{x}_i]$. Indeed, in this case, at the maximizing point, the first derivative

$$\frac{\partial M_\psi}{\partial x_i} = \frac{1}{n} \cdot \psi'(x_i - E) - \frac{1}{n^2} \cdot \sum_{j=1}^{n} \psi'(x_j - E)$$

should be equal to 0, and the second derivative

$$\frac{\partial^2 M_\psi}{\partial x_i^2} = \frac{1}{n} \cdot \psi''(x_i - E) \cdot \left(1 - \frac{2}{n}\right) + \frac{1}{n^3} \cdot \sum_{j=1}^{n} \psi''(x_j - E)$$

is non-positive. Since the function $\psi(x)$ is convex, we have $\psi''(x) \geq 0$, so this second derivative is a sum of non-negative terms, and the only case when it is non-negative is when all these terms are 0s, i.e., when $x_j = E$ for all $j$. In this case, $M_\psi = 0$ which, for non-degenerate intervals, is clearly not the largest possible value of $M_\psi$.

So, for every $i$, the maximum of $M_\psi$ is attained either when $x_i = \underline{x}_i$ or when $x_i = \overline{x}_i$. Similarly to the proof for the variance, we will now prove that the maximum is always attained for one of the vectors $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$. To prove this, we need to show that if $x_i = \overline{x}_i$ and $x_j = \underline{x}_j$ for some $i < j$ (and $\underline{x}_i \leq \underline{x}_j$), then the change described in that proof, while keeping the average $E$ intact, increases the value of $M_\psi$. Without losing generality, we can consider the case $\Delta_i \leq \Delta_j$. In this case, the fact that $M_\psi$ increase after the above-described change is equivalent to: $\psi(\underline{x}_i + 2\Delta_i - E) + \psi(\underline{x}_j - E) \leq \psi(\underline{x}_i - E) + \psi(\underline{x}_j + 2\Delta_i - E)$, i.e., that $\psi(\underline{x}_i + 2\Delta_i - E) - \psi(\underline{x}_i - E) \leq \psi(\underline{x}_j + 2\Delta_j - E) - \psi(\underline{x}_j - E)$. Since $\underline{x}_i \leq \underline{x}_j$ and $\underline{x}_i - E \leq \underline{x}_j - E$, this can be proven if we show that for every $\Delta > 0$ (and, in particular, for $\Delta = 2\Delta_i$), the function $\psi(x + \Delta) - \psi(x)$ is increasing. Indeed, the derivative of this function is equal to $\psi'(x + \Delta) - \psi'(x)$, and since $\psi''(x) \geq 0$, we do have $\psi'(x + \Delta) \geq \psi'(x)$.

Therefore, to find $\overline{M}_\psi$, it is sufficient to check all $n$ vectors of the type $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$, which requires $O(n^2)$ steps. For $m$ MIs, we similarly need $O(n^{m+1})$ steps.

*Summary.* These results are summarized in the following table. In this table, the first row corresponds to a general case, other rows correspond to different classes of problems described in Section 2:

| class number | class description |
|---|---|
| 0 | general case |
| 1 | narrow intervals: no intersection |
| 2 | slightly wider intervals<br>$\leq K$ intervals intersect |
| 3 | single measuring instrument (MI):<br>subset property –<br>no interval is a "proper" subset of the other |
| 4 | same accuracy measurements:<br>all intervals have the same half-width |
| 5 | several ($m$) measuring instruments:<br>intervals form $m$ groups,<br>with subset property in each group |
| 6 | privacy case:<br>intervals same or non-intersecting |
| 7 | non-detects case:<br>only non-degenerate intervals are $[0, DL_i]$ |

| # | $E$ | $V$ | $C_{xy}$ | $L, U, R$ | $M_{2p}$ | $M_{2p+1}$ |
|---|---|---|---|---|---|---|
| 0 | $O(n)$ | NP-hard | NP-hard | NP-hard | NP-hard | ? |
| 1 | $O(n)$ | $O(n \cdot \log(n))$ | $O(n^3)$ | $O(n^2)$ | $O(n^2)$ | $O(n^3)$ |
| 2 | $O(n)$ | $O(n^2)$ | $O(n^3)$ | $O(n^2)$ | $O(n^2)$ | $O(n^3)$ |
| 3 | $O(n)$ | $O(n \cdot \log(n))$ | ? | $O(n^2)$ | $O(n^2)$ | ? |
| 4 | $O(n)$ | $O(n \cdot \log(n))$ | $O(n^4)$ | $O(n^2)$ | $O(n^2)$ | ? |
| 5 | $O(n)$ | $O(n^{m+1})$ | ? | $O(n^{m+1})$ | $O(n^{m+1})$ | ? |
| 6 | $O(n)$ | $O(n \cdot \log(n))$ | $O(n^3)$ | $O(n^2)$ | $O(n^2)$ | ? |
| 7 | $O(n)$ | $O(n \cdot \log(n))$ | ? | $O(n^2)$ | $O(n^2)$ | ? |

The case when only $d$ out of $n$ data points are intervals is summarized in the following table:

| # | $E$ | $V$ | $C_{xy}$ | $L, U, R$ | $M_{2p}$ | $M_{2p+1}$ |
|---|---|---|---|---|---|---|
| 0 | $O(n)$ | NP-hard | NP-hard | NP-hard | NP-hard | ? |
| 1 | $O(n)$ | $O(n \log(d))$ | $O(n \cdot d^2)$ | $O(n \cdot d)$ | $O(nd)$ | $O(nd^2)$ |
| 2 | $O(n)$ | $O(nd)$ | $O(n \cdot d^2)$ | $O(n \cdot d)$ | $O(nd)$ | $O(nd^2)$ |
| 3 | $O(n)$ | $O(n \log(d))$ | ? | $O(n \cdot d)$ | $O(nd)$ | ? |
| 4 | $O(n)$ | $O(n \log(d))$ | $O(n \cdot d^3)$ | $O(n \cdot d)$ | $O(nd)$ | ? |
| 5 | $O(n)$ | $O(nd^m)$ | ? | $O(n \cdot d^m)$ | $O(nd^m)$ | ? |
| 6 | $O(n)$ | $O(n \log(d))$ | $O(n \cdot d^2)$ | $O(n \cdot d)$ | $O(nd)$ | ? |
| 7 | $O(n)$ | $O(n \log(d))$ | ? | $O(n \cdot d)$ | $O(nd)$ | ? |

*Weighted mean and weighted average.* In the above text, we considered the case when we only know the upper bound $\Delta_i$ on the overall measurement error. In some real-life situations (see, e.g., (Rabinovich, 1993)), we know the standard deviation $\sigma_i$ of the random error component and the bound $\Delta_i$ on the absolute value of the systematic error component. If we had no systematic errors, then we would able to estimate the mean $E$ by solving the corresponding Least Squares problem $\sum \sigma_i^{-2} \cdot (x_i - E)^2 \to \min\limits_E$, i.e., as $E_w = \sum\limits_{i=1}^{n} p_i \cdot x_i$, where $p_i \stackrel{\text{def}}{=} \dfrac{\sigma_i^{-2}}{\sum\limits_{j=1}^{n} \sigma_j^{-2}}$. In this case, the variance can be estimated as $V_w = \sum\limits_{i=1}^{n} p_i \cdot (x_i - E_w)^2 = \sum\limits_{i=1}^{n} p_i \cdot x_i^2 - E_w^2$. Due to the presence of systematic errors, the actual values $x_i$ may be anywhere within the intervals $[\underline{x}_i, \overline{x}_i] \stackrel{\text{def}}{=} [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. Thus, we arrive at the problem of estimating the range of the above expressions for weighted mean and weighted variance on the interval data $[\underline{x}_i, \overline{x}_i]$.

The expression for the mean is monotonic, so, similar to the average, we substitute the values $\underline{x}_i$ to get $\underline{E}_w$ and the values $\overline{x}_i$ to get $\overline{E}_w$.

For the weighted variance, the derivative is equal to $2p_i \cdot (x_i - E_w)$, and the second derivative is always $\geq 0$, so, similarly to the above proof for the non-weighted variance, we conclude that the minimum is always attained at a vector $(\overline{x}_1, \ldots, \overline{x}_k, E_w, \ldots, E_w, \underline{x}_{k+l}, \ldots, \overline{x}_n)$. So, by considering $2n + 2$ zones, we can find $\underline{V}_w$ in time $O(n^2)$.

For $\overline{V}_w$, we can prove that the maximum is always attained at values $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$, so we can always find it in time $O(2^n)$. If no more than $K$ intervals intersect, then, similarly to the non-weighted variance, we can compute $\overline{V}_w$ in time $O(n^2)$.

*Robust estimates for the mean.* Arithmetic average is vulnerable to outliers: if one of the values is accidentally mis-read as $10^6$ times larger than the others, the average is ruined. Several techniques have been proposed to make estimates robust; see, e.g., (Huber, 2004). The best known estimate of this type is the median; there are also more general *L-estimates* of the type $\sum\limits_{i=1}^{n} w_i \cdot x_{(i)}$, where $w_1 \geq 0, \ldots, w_n \geq 0$ are given constants, and $x_{(i)}$ is the $i$-th value in the ordering of $x_1, \ldots, x_n$ in increasing order. Other techniques include *M-estimates*, i.e., estimates $a$ for which $\sum\limits_{i=1}^{n} \psi(|x_i - a|) \to \max\limits_{a}$ for some non-decreasing function $\psi(x)$.

Each of these statistics $C$ is a (non-strictly) increasing function of each of the variables $x_i$. Thus, similarly to the average, $\mathbf{C} = [C(\underline{x}_1, \ldots, \underline{x}_n), C(\overline{x}_1, \ldots, \overline{x}_n)]$.

*Robust estimates for the generalized central moments.* When we discussed central moments, we considered generalized central moments $M_\psi = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} \psi(x_i - E)$ for an appropriate convex function $\psi(x)$. In that description, we assumed that $E$ is the usual average.

It is also possible to consider the case when $E$ is not the average, but the value for which $\sum\limits_{i=1}^{n} \psi(x_i - E) \to \min\limits_{E}$. In this case, the robust estimate for the generalized central moment takes the form

$$M_\psi^{\rm rob} = \min_E \left( \frac{1}{n} \cdot \sum_{i=1}^{n} \psi(x_i - E) \right).$$

Since the function $\psi(x)$ is convex, the expression $\sum\limits_{i=1}^{n} \psi(x_i - E)$ is also convex, so it only attains its maximum at the vertices of the convex box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_b$, i.e., when for every $i$, either $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$. For the case of a single MI, the same proof as for the average $E$ enables us to conclude that the maximum of the new generalized central moment is also always attained at one of $n$ vectors $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$, and thus, that this maximum can be computed in time $O(n^2)$. For $m$ MIs, we need time $O(n^{m+1})$.

*Correlation.* For correlation, we only know that in general, the problem of computing the exact range is NP-hard (Ferson et al., 2002d).

## 4. Additional Issues

*On-line data processing.* In the above text, we implicitly assumed that before we start computing the statistics, we have all the measurement results. In real life, we often continue measurements after we started the computations. Traditional estimates for mean

and variance can be easily modified with the arrival of the new measurement result $x_{n+1}$: $E' = (n \cdot E + x_{n+1})/(n+1)$ and $V' = M' - (E')^2$, where $M' = (n \cdot M + x_{n+1}^2)/(n+1)$ and $M = V + E^2$. For the interval mean, we can have a similar adjustment. However, for other statistics, the above algorithms for processing interval data require that we start computation from scratch. Is it possible to modify these algorithms to adjust them to on-line data processing? The only statistic for which such an adjustment is known is the variance, for which an algorithm proposed in (Wu et al., 2003; Kreinovich et al., (in press)) requires only $O(n)$ steps to incorporate a new interval data point.

In this algorithm, we store the sorting corresponding to the zones and we store auxiliary results corresponding to each zone (finitely many results for each zone). So, if only $d$ out of $n$ intervals are non-degenerate, we only need $O(d)$ steps to incorporate a new data point.

*Fuzzy data.* Often, in addition to (or instead of) the guaranteed bounds, an expert can provide bounds that contain $x_i$ with a certain degree of confidence. Often, we know several such bounding intervals corresponding to different degrees of confidence. Such a nested family of intervals is also called a *fuzzy set*, because it turns out to be equivalent to a more traditional definition of fuzzy set (Nguyen and Kreinovich, 1996; Nguyen and Walker, 1999) (if a traditional fuzzy set is given, then different intervals from the nested family can be viewed as $\alpha$-cuts corresponding to different levels of uncertainty $\alpha$).

To provide statistical analysis of fuzzy-valued data, we can therefore, for each level $\alpha$, apply the above interval-valued techniques to the corresponding $\alpha$-cuts (Martinez, 2003; Nguyen et al., 2003).

*Can we detect the case of several MI?* For the several MI case, we assumed that measurement are labeled, so that we can check which measurements were done by each MI; this labeling is used in the algorithms. What if we do not keep records on which interval was measured by which MI; can we then reconstruct the labels and thus apply the algorithms?

For two MI, we can: we pick an interval and call it $\mathrm{MI}_1$. If any other interval is in subset relation with this one, then this new interval is $\mathrm{MI}_2$. At any given stage, if one of the un-classified intervals is in subset relation with one of the already classified ones, we classify it to the opposite class. If none of the un-classified intervals is in subset relation with classified ones, we pick one of the un-classified ones and assign to $\mathrm{MI}_1$. After $\leq n$ iterations, we get the desired labeling.

In general, for $m$ MI, the labeling may not be easy. Indeed, we can construct a graph in which vertices are intervals, and vertices are connected if they are in a subset relation. Our objective is to assign a class to each vertex so that connected vertices cannot be of the same class. This is exactly the coloring problem that is known to be NP-hard (Garey and Johnson, 1979).

*Parallelization.* In the general case, the problem of computing the range $\mathbf{C}$ of a statistic $C$ on interval data $\mathbf{x}_i$ requires too much computation time. One way to speed up computations is to use parallel computations.

If we have a potentially unlimited number of parallel processors, then, for the mean, the addition can be done in time $O(\log(n))$ (Jaja, 1992). In $O(n \cdot \log(n))$ and $O(n^2)$ algorithms for computing $\underline{V}$ and $\overline{V}$, we can perform sorting in time $O(\log(n))$, then compute $V_k$ for each zone in parallel, and find the largest of the $n$ resulting values $V_k$ in parallel (in time $O(\log(n))$). The sum that constitutes the variance can also be computed in parallel in time $O(\log(n))$, so overall, we need $O(\log(n))$ time.

Similarly, we can transform polynomial algorithms for computing the bounds for covariance, outlier statistics ($L, U$, and $R$), and moments into $O(\log(n))$ parallel algorithms.

In the general case, to find $\overline{V}$ and other difficult-to-compute bounds, we must compute the largest of the $N \stackrel{\text{def}}{=} 2^n$ values corresponding to $2^n$ possible combinations of $\underline{x}_i$ and $\overline{x}_i$. This maximum can be computed in time $O(\log(N)) = O(n)$. This does not mean, of course, that we can always physically compute $\overline{V}$ in linear time: communication time grows exponentially with $n$; see, e.g., (Morgenstein and Kreinovich, 1995).

It is desirable to also analyze the case when we have a limited number of processors $p \ll n$.

*Quantum algorithms.* Another way to speed up computations is to use quantum computing. In (Martinez, 2003; Kreinovich and Longpré, 2004), we describe how quantum algorithms can speed up the computation of $\mathbf{C}$.

*What if we have partial information about the probabilities? Enter p-boxes.* In the above text, we assumed that the only information that we have about the measurement error $\Delta x$ is that this error is somewhere in the interval $[-\Delta, \Delta]$, and that we have no information about the probabilities of different values from this interval. In many real-life situations, we do not know the exact probability distribution for $\Delta x$, but we have a partial information about the corresponding probabilities. How can we describe this partial information?

To answer this question, let us recall how the complete information about the probability distribution is usually described. A natural way to describe a probability distribution is by describing its cumulative density function (cdf) $F(t) \stackrel{\text{def}}{=} \text{Prob}(\Delta x \le t)$. In practice, a reasonable way to store the information about $F(t)$ is to store *quantiles*, i.e., to fix a natural number $n$ and to store, for every $i$ from 0 to $n$, the values $t_i$ for which $F(t_i) = i/n$. Here, $t_0$ is the largest value for which $F(t_0) = 0$ and $t_n$ is the smallest value for which $F(t_n) = 1$, i.e., $[t_0, t_n]$ is the smallest interval on which the probability distribution is located with probability 1.

If we only have partial information about the probabilities, this means that – at least for some values $t$ – we do not know the exact value of $F(t)$. At best, we know an interval $\mathbf{F}(t) = [\underline{F}(t), \overline{F}(t)]$ of possible values of $F(t)$. So, a natural way to describe partial information about the probability distribution is to describe the two functions $\underline{F}(t)$ and $\overline{F}(t)$. This pair of cdfs is called a *p-box*; see, e.g., a book (Ferson, 2002). In addition to the theoretical concepts, this book describes the software tool for processing different types of uncertainty, a tool based on the notion of a p-box.

Similarly to the case of full information, it is reasonable to store the corresponding quantiles, i.e., the values $\underline{t}_i$ for which $\overline{F}(\underline{t}_i) = i/n$ and the values $\overline{t}_i$ for which $\underline{F}(\overline{t}_i) = i/n$.

(The reason why we switched the notations is because $\underline{F}(t) \le \overline{F}(t)$ implies $\underline{t}_i \le \overline{t}_i$.) This is exactly the representation used in (Ferson, 2002).

*What if we have partial information about the probabilities? Processing p-boxes and how the above alorithms can help.* Once we have a probability distribution $F(t)$, natural questions are: what is the mean and the variance of this distribution? A p-box means that several different distributions are possible, and for different distributions, we may have different values of means and variance. So, when we have a p-box, natural questions are: what is the range of possible values of the mean? what is the range of possible values of the variance?

The mean $E$ is a monotonic function of $F(t)$; so, for the mean $E$, the answer is simple: the mean of $\underline{F}(t)$ is the desired upper bound $\overline{E}$ for $E$, and the mean of $\overline{F}(t)$ is the desired lower bound $\underline{E}$ for $E$. The variance $V$ is not monotonic, so the problem of estimating the variance is more difficult.

For the case of the exact distribution, if we have the quantiles $t(\alpha)$ corresponding to all possible probability values $\alpha \in [0,1]$, then we can describe the mean of the corresponding probability distribution as $E = \int t(\alpha)\, d\alpha$, and the variance as $V = \int (t(\alpha) - E)^2\, d\alpha$. If we only know the quantiles $t_1 = t(1/n), \ldots, t_n = t(n/n)$, then it is reasonable to replace the integral by the corresponding integral sum; as a result, we get the estimates $E = \dfrac{1}{n} \sum\limits_{i=1}^{n} t_i$ and $V = \dfrac{1}{n} \sum\limits_{i=1}^{n} (t_i - E)^2$.

In these terms, a p-box means that instead of the exact value $t_i$ of each quantile, we have an interval of possible values $[\underline{t}_i, \overline{t}_i]$. So, to find the range of $V$, we must consider the range of possible values of $V$ when $t_i \in [\underline{t}_i, \overline{t}_i]$. There is an additional restriction that the values $t_i$ should be (non-strictly) increasing: $t_i \le t_{i+1}$.

The resulting problem is very similar to the problems of estimating mean and variance of the interval data. In this case, intervals satisfy the subset property, i.e., we are in the case that we called the case of single MI. The only difference between the current problem of analyzing p-boxes and the above problem is that in the above problem, we looked for minimum and maximum of the variance over all possible vectors $x_i$ for which $x_i \in \mathbf{x}_i$ for all $i$, while in our new problem, we have an additional monotonicity restriction $t_i \le t_{i+1}$. However, the solutions to our previous problems of computing $\underline{V}$ and $\overline{V}$ for the case of a single MI are actually attained at vectors that are monotonic. Thus, to find the desired value $V$, we can use the same algorithm as we described above.

Specifically, to find $\underline{V}$, we find $k$ for which the variance of the vector $t = (\overline{t}_1, \ldots, \overline{t}_k, \overline{t}, \ldots, \overline{t}, \underline{t}_{k+l}, \ldots, \underline{t}_n)$ for which the variance is the smallest. To find $\overline{V}$, we find $k$ for which the variance of the vector $t = (\underline{t}_1, \ldots, \underline{t}_k, \overline{t}_{k+1}, \ldots, \overline{t}_n)$ for which the variance is the largest. Intuitively, this makes perfect sense: to get the smallest $V$, we select the values $t_i$ as close to the average $\overline{t}$ as possible; to get the largest $V$, we select the values $t_i$ as far away from the average $\overline{t}$ as possible. In both case, we can compute $\underline{V}$ and $\overline{V}$ in time $O(n \cdot \log(n))$.

The above algorithm describes a heuristic estimate based on approximating an integral with an integral sum. To get reliable bounds, we can take into consideration that both bounds $\underline{F}(t)$ and $\overline{F}(t)$ are monotonic; thus, we can always replace the p-box by a larger

p-box in which the values $t(\alpha)$ are piecewise-constant: namely, we take $\mathbf{t}'_i = [\underline{t}_{i-1}, \overline{t}_i]$ for each $i$. For this new p-box, the integral sum coincides with the integral, so the range $[\underline{V}, \overline{V}]$ produced by the above algorithm is exactly the range of the variance over all possible distributions from the enlarged p-box. It is therefore guaranteed to contain the range of possible values of the variance $V$ for the original p-box.

*What if we have partial information about probabilities? Multi-dimensional case.* How can we describe partial information about probabilities in multi-dimensional case? A traditional analogue of a cdf is a multi-dimensional cdf

$$F(t_1, \ldots, t_p) = \mathrm{Prob}(x_1 \le t_1 \,\&\, \ldots \,\&\, x_p \le t_p);$$

see, e.g., (Wadsworth, 1990). The problem with this definition is that often multi-D data represent, e.g., vectors with components $x_1, \ldots, x_p$. The components depend on the choice of coordinates. As a result, even if a distribution is symmetric – e.g., a rotation-invariant Gaussian distribution – the description in terms of a multi-D cdf is *not* rotation-invariant.

It is desirable to come up with a representation that preserves such a symmetry. A natural way to do it is to store, for each half-space, the probability that the vector $\vec{x} = (x_1, \ldots, x_p)$ is within this half-space. In other words, for every unit vector $\vec{e}$ and for every value $t$, we store the probability $F(\vec{e}, t) \stackrel{\mathrm{def}}{=} \mathrm{Prob}(\vec{x} \cdot \vec{e} \le t)$, where $\vec{x} \cdot \vec{e} = x_1 \cdot e_1 + \ldots + x_n \cdot e_n$ is a scalar (dot) product of the two vectors. This representation is clearly rotation-invariant: if we change the coordinates, we keep the same values $F(\vec{e}, t)$; the only difference is that we store each value under different (rotated) $\vec{e}$. Moreover, this representation is invariant under arbitrary linear transformations.

Based on this information, we can uniquely determine the probability distribution. For example, if the probability distribution has a probability density function (pdf) $\rho(\vec{x})$, then this pdf can be reconstructed as follows. First, we determine the characteristic function $\chi(\vec{\omega}) \stackrel{\mathrm{def}}{=} E[\exp(\mathrm{i} \cdot (\vec{x} \cdot \vec{\omega}))]$, where $E[\cdot]$ stands for the expected value. To get the value of $\chi(\vec{\omega})$, we apply the 1-D Fourier transform, to the values $F(\vec{e}, t)$ for different $t$, where $\vec{e} \stackrel{\mathrm{def}}{=} \vec{\omega}/\|\vec{\omega}\|$ is a unit vector in the direction of $\vec{\omega}$. Then, we can find $\rho(\vec{x})$ by applying the $p$-dimensional Inverse Fourier Transform to $\chi(\vec{\omega})$.

It is therefore reasonable to represent a partial information about the probability distribution by storing, for each $\vec{e}$ and $t$, the bounds $\underline{F}(\vec{e}, t)$ and $\overline{F}(\vec{e}, t)$ that describe the range of possible values for $F(\vec{e}, t)$.

It is worth mentioning that since for continuous distributions, $F(\vec{e}, t) = 1 - F(-\vec{e}, -t)$, we have $\underline{F}(\vec{e}, t) = 1 - \overline{F}(-\vec{e}, -t)$. So, it is sufficient to only describe $\overline{F}(\vec{e}, t)$, the lower bounds $\underline{F}(\vec{e}, t)$ can then be uniquely determined (or, vice versa, we can only describe the values $\underline{F}(\vec{e}, t)$; then the values $\overline{F}(\vec{e}, t)$ will be uniquely determined).

In order to transform this idea into an efficient software tool, we need to solve two types of problems. First, we must solve algorithmic problems: develop algorithms for estimating the ranges of statistical characteristics (such as moments) for the corresponding multi-D p-boxes.

Second, we must solve implementation problems. Theoretically, to uniquely describe a probability distribution, we need to know infinitely many values $F(\vec{e}, t)$ corresponding to infinitely many different vectors $\vec{e}$ and infinitely many different numbers $t$. In practice, we can only store finitely many values $F(\vec{e}, t)$ corresponding to finitely many vectors $\vec{e}$.

In principle, we can simply select a rectangular grid and store the values for the vectors $\vec{e}$ from this grid. However, the selection of the grid violates rotation-invariance and thus, eliminates the advantage of selecting this particular multi-D analogue of a cdf. It turns out that there is a better way: instead of using a grid, we can use rational points on a unit sphere. There exists efficient algorithms for generating such points, and the set of all such points is almost rotation-invariant: it is invariant with respect to all rotations for which all the entries in the corresponding rotation matrix are rational numbers (Oliverio, 1996; Trautman, 1998).

*Beyond p-boxes?* A p-box does not fully describe all kinds of possible partial information about the probability distribution. For example, the same p-box corresponds to the class of all distributions located on an interval $[0, 1]$ and to the class of all distributions located at two points 0 and 1.

Similarly, in the multi-D case, if we only use the above-defined multi-D cdfs, we will not be able to distinguish between a set $S$ (= the class of all probability distributions localized on the set $S$ with probability 1) and its convex hull. To provide such a distinction, we may want, in addition to the bounds on the probabilities $\text{Prob}(f(x) \le t)$ for all *linear* functions $f(x)$, to also keep the bounds on the similar probabilities corresponding to all *quadratic* functions $f(x)$.

Let us show that this addition indeed enables us to distinguish between different sets $S$. Indeed, for every point $x$, to check whether $x \in S$, we ask, for different values $\varepsilon > 0$, for the upper bound for the probability $\text{Prob}(d^2(x, x_0) \le \varepsilon^2)$, where $d(x, x_0)$ denotes the distance between the two points. If $x \notin S$, then for sufficiently small $\varepsilon$, this probability will be 0; on the other hand, if $x \in S$, then it is possible that we have a distribution located at this point $x$ with probability 1, so the upper bound is 1 for all $\varepsilon$ (Nguyen et al., 2000).

In 1-D case, the condition $f(x) \le t$ for a non-linear quadratic function $f(x)$ is satisfied either inside an interval, or outside an interval. Thus, in 1-D case, our idea means that in addition to cdf, we also store the bounds on the probabilities of $x$ being within different intervals. Such bounds are analyzed, e.g., in (Berleant, 1993; Berleant, 1996; Berleant and Goodman-Strauss, 1998; Berleant et al., 2003).

## Acknowledgements

This work was supported in part by NASA under cooperative agreement NCC5-209, by the Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-1-0365, by NSF grants EAR-0112968, EAR-0225670, and EIA-0321328, by the Army Research Laboratories grant

DATM-05-02-C-0046, and by a research grant from Sandia National Laboratories as part of the Department of Energy Accelerated Strategic Computing Initiative (ASCI).

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

## References

Beck, B., V. Kreinovich, and B. Wu, Interval-Valued and Fuzzy-Valued Random Variables: From Computing Sample Variances to Computing Sample Covariances, In: M. Lopez, M. A. Gil, P. Grzegorzewski, O. Hrynewicz, and J. Lawry, editor, *Soft Methodology and Random Information Systems*, pages 85–92, Springer-Verlag, Berlin Heidelberg New York Tokyo, 2004.

Berleant, D., Automatically verified arithmetic with both intervals and probability density functions, *Interval Computations*, 1993, (2):48–70.

Berleant, D., Automatically verified arithmetic on probability distributions and intervals, In: R. B. Kearfott and V. Kreinovich, editors, *Applications of Interval Computations*, Kluwer, Dordrecht, 1996.

Berleant, D., and C. Goodman-Strauss, Bounding the results of arithmetic operations on random variables of unknown dependency using intervals, *Reliable Computing*, 1998, 4(2):147–165.

Berleant, D., L. Xie, and J. Zhang, Statool: A Tool for Distribution Envelope Determination (DEnv), an Interval-Based Algorithm for Arithmetic on Random Variables, *Reliable Computing*, 2003, 9(2):91–108.

Ferson, S. *RAMAS Risk Calc 4.0: Risk Assessment with Uncertain Numbers*, CRC Press, Boca Raton, Florida, 2002.

Ferson, S., L. Ginzburg, V. Kreinovich, and M. Aviles, Exact Bounds on Sample Variance of Interval Data. *Extended Abstracts of the 2002 SIAM Workshop on Validated Computing*, Toronto, Canada, 2002, pp. 67–69

Ferson, S., L. Ginzburg, V. Kreinovich, and M. Aviles, *Exact Bounds on Finite Populations of Interval Data*, University of Texas at El Paso, Department of Computer Science, Technical Report UTEP-CS-02-13d, 2002, http://www.cs.utep.edu/vladik/2002/tr02-13d.pdf

Ferson, S., L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, Computing Variance for Interval Data is NP-Hard, *ACM SIGACT News*, 33(2):108–118, 2002.

Ferson, S., D. Myers, and D. Berleant, *Distribution-free risk analysis: I. Range, mean, and variance*, Applied Biomathematics, Technical Report, 2001.

Garey, M. E., and D. S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, Freeman, San Francisco, 1979.

Granvilliers, L., V. Kreinovich, and N. Müller, Novel Approaches to Numerical Software with Result Verification", In: R. Alt, A. Frommer, R. B. Kearfott, and W. Luther, editors, *Numerical Software with Result Verification*, International Dagstuhl Seminar, Dagstuhl Castle, Germany, January 19–24, 2003, Revised Papers, Springer Lectures Notes in Computer Science, 2004, Vol. 2991, pp. 274–305.

Huber, P. J., *Robust statistics*, Wiley, New York, 2004.

Jájá, J. *An Introduction to Parallel Algorithms*, Addison-Wesley, Reading, MA, 1992.

Kreinovich, V. Probabilities, Intervals, What Next? Optimization Problems Related to Extension of Interval Computations to Situations with Partial Information about Probabilities, *Journal of Global Optimization* (in press).

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

Kreinovich, V., and L. Longpré, Computational complexity and feasibility of data processing and interval computations, with extension to cases when we have partial information about probabilities, In: V. Brattka, M. Schroeder, K. Weihrauch, and N. Zhong, editors, *Proc. Conf. on Computability and Complexity in Analysis CCA'2003*, Cincinnati, Ohio, USA, August 28–30, 2003, pp. 19–54.

Kreinovich, V., and L. Longpré, Fast Quantum Algorithms for Handling Probabilistic and Interval Uncertainty, *Mathematical Logic Quarterly*, 2004, 50(4/5):507–518.

Kreinovich, V., L. Longpré, S. Ferson, and L. Ginzburg, *Computing Higher Central Moments for Interval Data*, University of Texas at El Paso, Department of Computer Science, Technical Report UTEP-CS-03-14b, 2004, http://www.cs.utep.edu/vladik/2003/tr03-14b.pdf

Kreinovich, V., L. Longpré, P. Patangay, S. Ferson, and L. Ginzburg, Outlier Detection Under Interval Uncertainty: Algorithmic Solvability and Computational Complexity, In: I. Lirkov, S. Margenov, J. Wasniewski, and P. Yalamov, editors, *Large-Scale Scientific Computing*, Proceedings of the 4-th International Conference LSSC'2003, Sozopol, Bulgaria, June 4–8, 2003, Springer Lecture Notes in Computer Science, 2004, Vol. 2907, pp. 238–245

Kreinovich, V., H. T. Nguyen, and B. Wu, On-Line Algorithms for Computing Mean and Variance of Interval Data, and Their Use in Intelligent Systems, *Information Sciences* (in press).

Kreinovich, V., P. Patangay, L. Longpré, S. A. Starks, C. Campos, S. Ferson, and L. Ginzburg, Outlier Detection Under Interval and Fuzzy Uncertainty: Algorithmic Solvability and Computational Complexity, *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society NAFIPS'2003*, Chicago, Illinois, July 24–26, 2003, pp. 401–406.

Kuznetsov, V. P., *Interval Statistical Models*, Radio i Svyaz, Moscow, 1991 (in Russian).

Lodwick, W. A., and K. D. Jamison, Estimating and Validating the Cumulative Distribution of a Function of Random Variables: Toward the Development of Distribution Arithmetic, *Reliable Computing*, 2003, 9(2):127–141.

Martinez, M., L. Longpré, V. Kreinovich, S. A. Starks, and H. T. Nguyen, Fast Quantum Algorithms for Handling Probabilistic, Interval, and Fuzzy Uncertainty, *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society NAFIPS'2003*, Chicago, Illinois, July 24–26, 2003, pp. 395–400.

Moore, R. E., and W. A. Lodwick, Interval Analysis and Fuzzy Set Theory, *Fuzzy Sets and Systems*, 2003, 135(1):5–9.

Morgenstein, D., and V. Kreinovich, Which algorithms are feasible and which are not depends on the geometry of space-time, *Geombinatorics*, 1995, 4(3):80–97.

Nguyen, H. T., and V. Kreinovich, Nested Intervals and Sets: Concepts, Relations to Fuzzy Sets, and Applications, In: R. B. Kearfott and V. Kreinovich, editors, *Applications of Interval Computations*, Kluwer, Dordrecht, 1996, pp. 245–290

Nguyen, H. T., and E. A. Walker, *First Course in Fuzzy Logic*, CRC Press, Boca Raton, Florida, 1999.

Nguyen, H. T., T. Wang, and V. Kreinovich, Towards Foundations of Processing Imprecise Data: From Traditional Statistical Techniques of Processing Crisp Data to Statistical Processing of Fuzzy Data, In: Y. Liu, G. Chen, M. Ying, and K.-Y. Cai, editors, *Proceedings of the International Conference on Fuzzy Information Processing: Theories and Applications FIP'2003*, Beijing, China, March 1–4, 2003, Vol. II, pp. 895–900.

Nguyen, H. T., B. Wu, and V. Kreinovich, Shadows of Fuzzy Sets – A Natural Approach Towards Describing 2-D and Multi-D Fuzzy Uncertainty in Linguistic Terms, *Proc. 9th IEEE Int'l Conference on Fuzzy Systems FUZZ-IEEE'2000*, San Antonio, Texas, May 7–10, 2000, Vol. 1, pp. 340–345.

Nivlet, P., F. Fournier, and J. Royer, A new methodology to account for uncertainties in 4-D seismic interpretation, *Proc. 71st Annual Int'l Meeting of Soc. of Exploratory Geophysics SEG'2001*, San Antonio, TX, September 9–14, 2001, 1644–1647.

Nivlet, P., F. Fournier, and J. Royer, Propagating interval uncertainties in supervised pattern recognition for reservoir characterization, *Proc. 2001 Society of Petroleum Engineers Annual Conf. SPE'2001*, New Orleans, LA, September 30–October 3, 2001, paper SPE-71327.

Oliverio, P., Self-generating Pythagorean quadruples and $n$-tuples, *Fibonacci Quarterly*, May 1996, pp. 98–101.

Osegueda, R., V. Kreinovich, L. Potluri, R. Aló, Non-Destructive Testing of Aerospace Structures: Granularity and Data Mining Approach, *Proc. FUZZ-IEEE'2002*, Honolulu, HI, May 12–17, 2002, Vol. 1, pp. 685–689

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

Regan, H., S. Ferson, and D. Berleant, Equivalence of five methods for bounding uncertainty, *Journal of Approximate Reasoning* (in press).

Rowe, N. C., Absolute bounds on the mean and standard deviation of transformed data for constant-sign-derivative transformations, *SIAM Journal of Scientific Statistical Computing*, 1988, 9:1098–1113.

Shmulevich, I., and W. Zhang, Binary analysis and optimization-based normalization of gene expression data, *Bioinformatics*, 2002, 18(4):555–565.

Trautman, A., Pythagorean Spinors and Penrose Twistors, In: S. A. Hugget et al., editors, *The Geometric Universe; Science, Geometry, and the Work of Roger Penrose*, Oxford Univ. Press, Oxford, 1998.

Wadsworth, H. M. Jr., editor, *Handbook of statistical methods for engineers and scientists*, McGraw-Hill Publishing Co., N.Y., 1990.

Wu, B., H. T. Nguyen, and V. Kreinovich, Real-Time Algorithms for Statistical Analysis of Interval Data, *Proceedings of the International Conference on Information Technology InTech'03*, Chiang Mai, Thailand, December 17–19, 2003, pp. 483–490.

Walley, P., *Statistical Reasoning with Imprecise Probabilities*, Chapman & Hall, N.Y., 1991.

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

Xiang, G., S. A. Starks, V. Kreinovich, and L. Longpré, New Algorithms for Statistical Analysis of Interval Data, *Proceedings of the Workshop on State-of-the-Art in Scientific Computing PARA'04*, Lyngby, Denmark, June 20–23, 2004, Vol. 1, pp. 123–129.

Zhang, W., I. Shmulevich, and J. Astola, *Microarray Quality Control*, Wiley, Hoboken, New Jersey, 2004.

# Structural Design under Fuzzy Randomness

MICHAEL BEER
*Dept. of Mechanical Engineering and Materials Science, Rice University, Houston, TX 77005, USA, email: mb18@rice.edu*

MARTIN LIEBSCHER, BERND MÖLLER
*Institute of Statics and Dynamics of Structures, Dresden University of Technology, Mommsenstr. 13, 01062 Dresden, Germany, email: martin.liebscher@mailbox.tu-dresden.de, moeller@rcs.urz.tu-dresden.de*

**Abstract.** In this paper a procedure for designing structures under uncertainty is presented. The uncertainty model of fuzzy randomness is employed to take into account the uncertainty of structural parameters in a realistic and comprehensive manner. This uncertainty model includes real valued random variables and fuzzy variables as special cases. Objective uncertainty and subjective uncertainty are processed simultaneously.

Algorithms of fuzzy structural analysis (processing of fuzzy variables in structural analysis) and fuzzy probabilistic safety assessment (processing of fuzzy random variables, real random variables, and fuzzy variables in safety assessment) are used to compute fuzzy structural responses and fuzzy safety prognoses, which are the backbone of the new design concept. Comparing fuzzy structural responses and the fuzzy safety level with permissible values, discrete permissible and nonpermissible parameter vectors are identified. These are introduced into a fuzzy cluster analysis to obtain permissible and nonpermissible clusters (continuous sets of real parameter vectors with similar properties), which represent the basis for generating uncertain structural design parameters.

This concept is referred to as fuzzy cluster design. It can be combined with arbitrary fundamental solutions for deterministic and probabilistic structural safety analyses. For instance, well developed algorithms of Monte Carlo simulation and codes of nonlinear structural analysis can be incorporated in the procedure.

The algorithm of fuzzy cluster design is presented in detail and demonstrated by way of a numerical example.

**Keywords:** Fuzzy variables; Fuzzy random variables; Fuzzy structural analysis; Fuzzy probabilistic safety assessment; Fuzzy clustering; Uncertain structural design; Nonlinear structural design.

## 1. Introduction

Structural engineering focuses on computing structural responses, assessing structural safety, and determining parameters for structural design that meet all relevant requirements. For these purposes, the structural engineer has to apply appropriate structural models, suitably-matched computational models and reliable structural parameters close to reality. Computational models must be capable of numerically simulating the system behavior of the chosen structural model. Such models have already been developed up to a high quality level and are available as nonlinear numerical procedures for solving many problems. Structural models and structural parameters, however, have to be established in the particular case on the basis of plans, drawings, measurements, observations, experiences, expert knowledge, codes, and standards. As a rule certain information regarding structural models and precise values of structural parameters do not exist. Structural models and structural parameters are characterized by uncertainty. Human errors in the manufacture, the use and maintenance of constructions, expert evaluations, and insufficient information sources represent only some examples of uncertain influences. In order to perform realistic structural analysis and safety assessment this uncertainty must be appropriately taken into consideration.

Various concepts are available for mathematically describing and quantifying uncertainty such as, e.g., probability theory [17], including subjective probability approach [31] and Bayes methods [6], interval mathematics [1], convex modeling [5], theory of rough sets [23], fuzzy set theory [2], theory of fuzzy random variables [15] and chaos theory [13]. In the scientific literature the new uncertainty models are not only controversially discussed [9] but also increasingly implemented for the solution of practice-relevant problems [4, 8, 12, 16, 22, 24, 29] These different developments of uncertainty models do not directly contradict each other but rather constitute an entirety.

The procedure presented in this paper takes account of uncertainty that may be quantified using the concept of fuzzy randomness. Structural parameters are modeled as fuzzy random variables, real random variables, or fuzzy variables. These are simultaneously processed in special procedures of fuzzy structural analysis and fuzzy probabilistic safety assessment to yield the associated uncertain structural responses and uncertain safety levels. The uncertainty of the structural parameters is apparent in the results. Generally, arbitrary computational models may be employed as deterministic and probabilistic fundamental solutions in these special procedures. If necessary, sophisticated nonlinear codes for structural analysis such as described in [26] and well developed algorithms of computational stochastics such as Monte Carlo simulation [27] may be applied to obtain results close to reality.

The uncertain results from fuzzy structural analysis and fuzzy probabilistic safety assessment provide a suitable basis for deriving an uncertain structural design. With the aid of a fuzzy cluster analysis algorithm permissible and nonpermissible clusters are detected in the space of the design parameters. These are taken as the basis to generate modified uncertain structural parameters, which represent alternative design variants. By comparing requirements regarding structural responses and safety levels with the fuzzy results associated with these design variants, their permissibility and quality is finally assessed to find the optimum uncertain structural design.

This fuzzy cluster design concept does not make any demands on the underlying deterministic and probabilistic fundamental solutions for fuzzy structural analysis and fuzzy probabilistic safety assessment. As it solves the inverse problem of structural analysis and safety assessment numerically, it is generally applicable and may also be employed to solve deterministic design problems in arbitrary nonlinear cases.

The basis and algorithmic development of fuzzy cluster design are discussed in detail in the sequel.

## 2. Processing Uncertainty in Structural Safety Assessment

For introducing uncertain structural parameters into structural analysis and safety assessment these must be quantified. Depending on the characteristic of the uncertainty appropriate mathematical models are applied. The uncertain parameters are classified according to the cause of their uncertainty. If exclusively informal or lexical uncertainty appears, the uncertainty characteristic fuzziness may be stated. If the uncertain parameter considered is partly influenced by stochastic uncertainty, but cannot be described clearly using random variables, then the characteristic fuzzy randomness may be assigned. Thereby, real valued random variables may be treated as a special case of fuzzy

random variables. These represent uncertain variables whose uncertainty exclusively arises from stochastic causes.

Uncertain parameters whose uncertainty characteristic has been identified as fuzziness are treated on the basis of fuzzy set theory. They are described as fuzzy variables $\tilde{x}$ quantified by membership functions $\mu(x)$ [2, 32], see Figure 1. The specification of the membership function is referred to as fuzzification. General algorithms for fuzzification cannot be provided as it basically represents a subjective assessment [20, 30].

Uncertainty with the characteristic fuzzy randomness is described, quantified, and processed on the basis of the theory of fuzzy random variables. For quantifying fuzzy random variables $\tilde{X}$ fuzzy probability distributions $\tilde{F}(x)$ are introduced [18, 30], see Figure 1. These may be understood as being a bunch of real valued probability distributions assessed by membership values $\mu$ indicating their degree of plausibility. The fuzzy probability distribution functions are analytically described by introducing fuzzy functional parameters $\tilde{p}$ into the common equations for distribution functions. Additionally, fuzzy functional types of distributions may be defined. The fuzzy parameters $\tilde{p}$ of the fuzzy probability distributions represent fuzzy variables characterized by membership functions $\mu(p)$. These fuzzy parameters may be determined on the basis of common statistical methods in combination with procedures from fuzzy set theory[18, 20, 30].

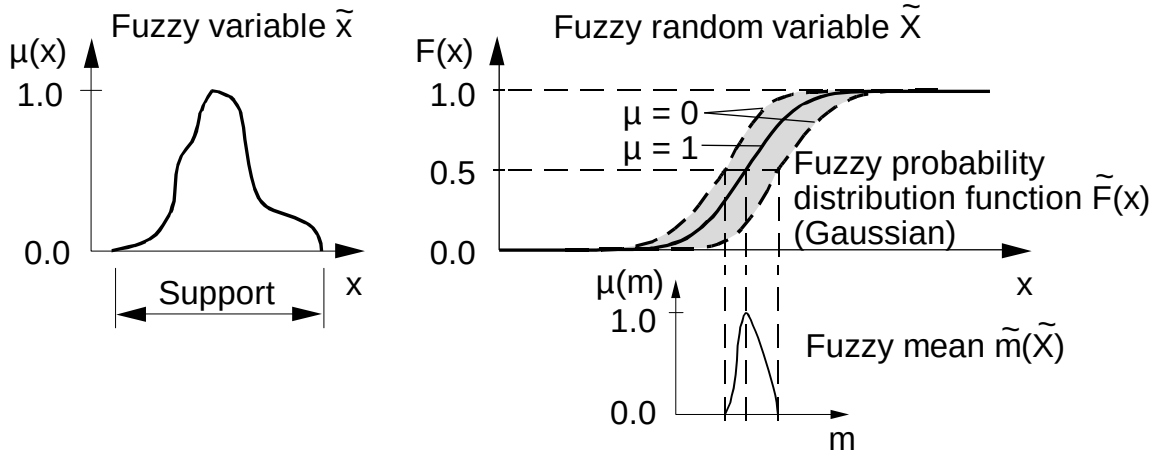



*Figure 1.* Fuzzy variable $\tilde{x}$ and fuzzy random variable $\tilde{X}$

For the purpose of fuzzy cluster design, the uncertain structural parameters are mapped to fuzzy structural responses and fuzzy safety levels, which are compared with required or permissible values as design constraints.

Fuzzy structural responses are obtained from fuzzy structural analysis [18, 20]. Fuzzy structural analysis implies the analysis of a structure with the aid of a crisp (or uncertain) algorithm applied to fuzzy variables for structural parameters. It may formally be described as the mapping of the fuzzy input vector $\underline{\tilde{x}}$ consisting of the fuzzy structural parameters $\tilde{x}_k$ to the vector $\underline{\tilde{z}}$ containing the fuzzy structural responses $\tilde{z}_j$,

$$\underline{\tilde{x}} \rightarrow \underline{\tilde{z}} \, , \tag{1}$$

see Figure 2. The mapping 1 is realized with the aid of the mapping model f,

$$\underline{z} = (z_1, \, ..., \, z_j, \, ..., \, z_m) = f(x_1, \, ..., \, x_k, \, ..., \, x_r) \, , \tag{2}$$

which is represented here by a deterministic algorithm for statical or dynamic structural analysis as a deterministic fundamental solution.

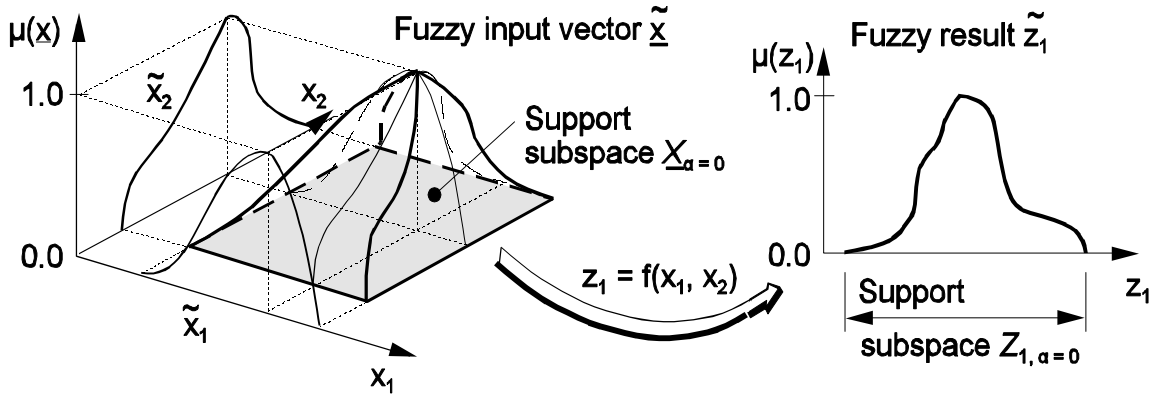



*Figure 2.* Mapping of fuzzy input variables to a fuzzy result variable

Fuzzy safety levels are computed by applying the concept of fuzzy probabilistic safety assessment. The developed Fuzzy First Order Reliability Method (FFORM) [18, 21] is selected here. Fuzzy random variables together with fuzzy structural parameters are introduced into the algorithms of FFORM to compute a fuzzy reliability index $\tilde{\beta}$. In terms of fuzzy analysis this may be formulated as the mapping

$$\underline{\tilde{x}}^e = (\tilde{x}_1, \, ..., \, \tilde{x}_k, \, ..., \, \tilde{x}_r, \, \tilde{p}_1, \, ..., \, \tilde{p}_t, \, ..., \, \tilde{p}_q) \rightarrow \tilde{\beta} \, , \tag{3}$$

of an extended fuzzy input vector $\underline{\tilde{x}}^e$ to the fuzzy reliability index $\tilde{\beta}$ representing a special fuzzy result variable. The extended fuzzy input vector $\underline{\tilde{x}}^e$ not only contains fuzzy structural parameters $\tilde{x}_k$ but also comprises the fuzzy parameters $\tilde{p}_t$ of the fuzzy random variables $\tilde{X}_s$. For purposes of notation convenience, all fuzzy input variables are subsequently denoted by $\tilde{x}_k$, whereas all fuzzy result variables are designated as $\tilde{z}_j$.

For processing the fuzziness of the fuzzy input vector a generally applicable as well as efficient numerical algorithm has been developed and formulated in terms of $\alpha$-level optimization [18, 20]. This concept permits to implement an arbitrary nonlinear deterministic fundamental solution without any special properties. The membership scale of the fuzzy input vector is discretized ($\alpha$-discretization), which leads to a certain number of $\alpha$-level sets

$$\underline{X}_\alpha = \{\underline{x} \in \underline{\tilde{x}} \mid \mu(\underline{x}) \geq \alpha\} \, . \tag{4}$$

The mapping 1 may then be described by

$$\underline{X}_\alpha \;\rightarrow\; \underline{Z}_\alpha \,\forall\, \alpha \;\in\; (0,\, 1] \;. \tag{5}$$

The $\alpha$-level sets $\underline{X}_\alpha$ are mapped to the associated $\alpha$-level sets $\underline{Z}_\alpha$ of the fuzzy results. For $\alpha$ = 0 (as an abbreviation for the limit $\alpha \rightarrow +0$) this mapping concerns the support subspaces $\underline{X}_{\alpha=0}$ and $\underline{Z}_{\alpha=0}$, which contain all elements of the fuzzy input vector and the fuzzy result, respectively, see Figure 2. The resulting $\alpha$-level sets $\underline{Z}_\alpha$ are described with the aid of their boundings. For their determination a modified evolution strategy has been developed and combined with a repeated solution of the associated optimization problem.

As a consequence, the fuzzy results are obtained with numerically determined membership functions. This means that fuzzy results are not available as connected sets but represent discrete sets of randomly distributed points in the space of the fuzzy result variables. Additionally, the parameter coordinates in the space of the fuzzy input variables which belong to these result points are known.

## 3. Fuzzy Cluster Design

### 3.1. CONCEPT

The basic idea of fuzzy cluster design is to derive an appropriate structural design accounting for the uncertainty of design parameters. It is thus reasonable to make use of the numerical results from fuzzy structural analysis and fuzzy probabilistic safety assessment. In contrast to deterministic computations, these results not only provide a single input-output dependency but offer some systematic insight into relationships between structural parameters and structural responses and safety levels over a certain range of parameter values. If the uncertain structural parameters are initially defined in such a way that they cover a proper design domain, then the information gained from fuzzy structural analysis and fuzzy probabilistic safety assessment may be used to find continuous sets of suitable design parameter vectors within this defined domain.

As a result of the mapping 5, a set of input points $\underline{\mathrm{x}}_\mathrm{i}$ from the support subspace $\underline{X}_{\alpha=0}$ of $\underline{\tilde{\mathrm{x}}}$ and a set of result points $\underline{\mathrm{z}}_\mathrm{i}$ from the support subspace $\underline{Z}_{\alpha=0}$ of $\underline{\tilde{\mathrm{z}}}$ are obtained. The result points $\underline{\mathrm{z}}_\mathrm{i}$ are assigned to the input points $\underline{\mathrm{x}}_\mathrm{i}$ by means of the mapping model according to 2 on a point-by-point basis,

$$\underline{\mathrm{z}}_\mathrm{i} \;=\; \mathrm{f}(\underline{\mathrm{x}}_\mathrm{i}) \;. \tag{6}$$

From $\alpha$-level optimization, the inverse assignment is also available point-by-point,

$$\underline{\mathrm{x}}_\mathrm{i} \;=\; \mathrm{f}^{-1}(\underline{\mathrm{z}}_\mathrm{i}) \;. \tag{7}$$

The fact that these dependencies are only known in a discrete form excludes a closed solution of the design problem; continuous sets of permissible design parameter vectors cannot be determined by simply applying 7. The pointwise information from 6 and 7, however, permits designing structures virtually directly with the aid of cluster analysis methods.

The points $\underline{x}_i$, i = 1, ..., n, are lumped together in the discrete point set $M_x$, whereas the points $\underline{z}_i$, i = 1, ..., n, are combined to form the discrete point set $M_z$. According to the design constraints $CT_h$ with h = 1, ..., q, the point set $M_z$ is subdivided into two disjoint subsets $M_{z^+}$ and $M_{z^-}$ comprising exclusively permissible points $\underline{z}_i^+ = (z_{i,1}^+, ..., z_{i,m}^+)$ or only nonpermissible points $\underline{z}_i^- = (z_{i,1}^-, ..., z_{i,m}^-)$, respectively. As a consequence the point set $M_x$ also becomes decomposed due to the dependencies in 6 and 7. The resulting disjoint subsets $M_{x^+}$ and $M_{x^-}$ of $M_x$ contain only permissible points $\underline{x}_i^+ = (x_{i,1}^+, ..., x_{i,r}^+)$ or only nonpermissible points $\underline{x}_i^- = (x_{i,1}^-, ..., x_{i,r}^-)$, respectively. A cluster analysis is then separately applied to both the permissible points $\underline{x}_i^+$ and the nonpermissible points $\underline{x}_i^-$. The obtained clusters comprised of permissible points represent alternative structural design variants [3, 18, 19], see Figure 3.

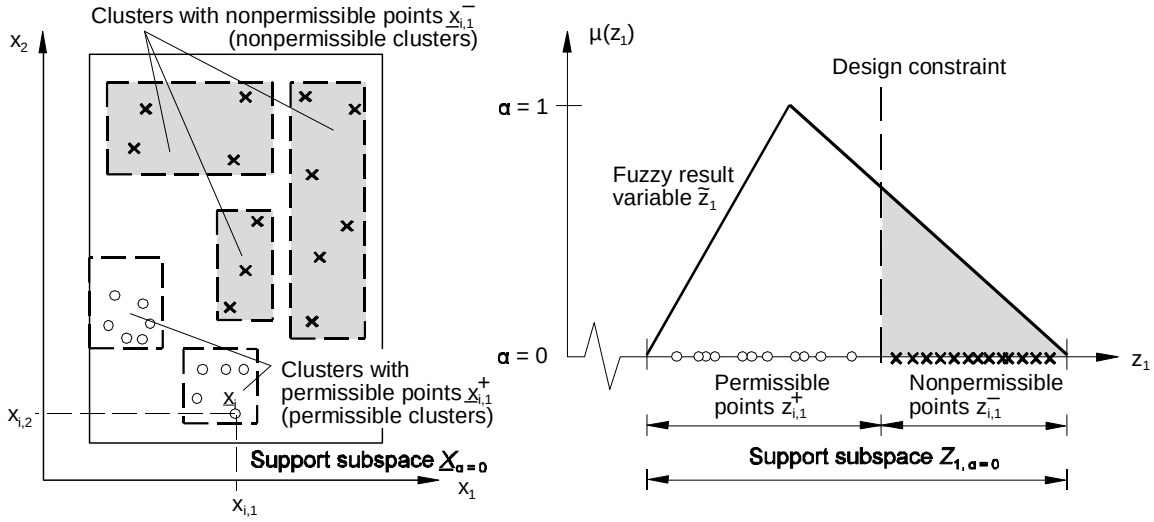



*Figure 3.* Clustering permissible and nonpermissible input points

The concept presented may be applied to designing structures based on fuzzy structural analysis as well as on fuzzy probabilistic safety assessment. Depending on the particular design problem fuzzy structural parameters or fuzzy parameters of fuzzy probability distributions may be selected as design parameters. Design constraints may be formulated for fuzzy structural responses or for the fuzzy safety level. Furthermore, the concept of fuzzy structural design for the first time provides a tool for designing structures directly and independently of the computational model. That is, every arbitrary nonlinear structural analysis algorithm may be taken as a basis.

### 3.2. FUZZY CLUSTER ANALYSIS

For generating permissible and nonpermissible clusters arbitrary established cluster analysis algorithms may be taken as a basis. The aim of these algorithms is to generate clusters from a given set of data, which are referred to as objects. Objects that are similar to each other are lumped together to

form one cluster, whereas objects that are dissimilar are assigned to different clusters. Each cluster should only contain objects of high similarity. That is, the distances d(i, j) between its objects $\square_i$ and $\square_j$ should be as small as possible. Objects from different clusters are characterized by dissimilarity, the distances between these objects should be as large as possible. In structural design with clusters the input points $\underline{x}_i^+$ and $\underline{x}_i^-$ in the support subspace $\underline{X}_{\alpha=0}$ are the objects.

For determining clusters from point sets a variety of methods are available [10, 14, 28]. Fuzzy cluster methods are particularly suitable for designing structures. These are characterized by the fact that the assignment of objects to clusters is fuzzy, which yields fuzzy clusters. The degree with which an object $\square_i$ belongs to a particular cluster $C_v$ is expressed by the membership value $\mu_{iv} \in [0, 1]$. An object may simultaneously be assigned to different clusters with different membership values.

Herein, the fuzzy cluster method presented in [14] is selected as a basis for fuzzy cluster design. In this method the determination of the cluster configuration is formulated as the nonlinear optimization problem

$$\sum_{v=1}^{k} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \mu_{iv}^{r} \cdot \mu_{jv}^{r} \cdot d\left(\square_i, \square_j\right)}{2 \cdot \sum_{j=1}^{n} \mu_{jv}^{r}} \Rightarrow \text{Min} \tag{8}$$

with the equality and inequality constraints

$$\mu_{iv} \geq 0;\ i = 1, ..., n;\ v = 1, ..., k \tag{9}$$

and

$$\sum_{v=1}^{k} \mu_{iv} = 1;\ i = 1, ..., n\ . \tag{10}$$

The number k of clusters must be predefined. Then, the objective function is minimized by iteratively improving the membership values $\mu_{iv}$ and $\mu_{jv}$. The exponent $r \in [1, \infty)$ controls the influence of the membership values sought. For large values r the result of the cluster analysis exhibits a strong fuzziness characterized by equal membership values $\mu_{iv} = k^{-1}$ for all objects. In contrast to this, a value r close to unity results in an almost crisp clustering. For applications the exponent r = 2 is frequently chosen.

In order to determine a suitable number k of clusters the cluster analysis is usually repeated with a varying number of clusters. The results are then compared based on quality measures to find the optimum solution. The absolute value of the quality measure, however, should not be taken into consideration as the sole criterion. Additionally, the relative improvement in the quality measure when sucessively increasing the number of clusters should be accounted for. In the present study the partition coefficient and the separation degree are applied as major quality measures [10].

The partition coefficient evaluates the ”clearness” with which the objects are assigned to clusters. The more uncertain an assignment is obtained, the worse the cluster configuration is assessed to be. It is defined as

$$PC = \frac{1}{n} \sum_{v=1}^{k} \sum_{i=1}^{n} \mu_{iv}^{2}\ . \tag{11}$$

As an enhancement, the normalized partition coefficient

$$\mathrm{PCN} = 1 - \frac{\mathrm{k}}{\mathrm{k}-1} \cdot (1 - \mathrm{PC}) \tag{12}$$

is defined [28], which takes values from the interval [0, 1]. An appropriate clustering is characterized by a PCN close to unity.

The separation degree measures the separation of the clusters. That is, the average distance between the clusters and their assigned objects is evaluated with respect to the squared minimum distance between different clusters,

$$\mathrm{SD} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} \sum_{\mathrm{v}=1}^{\mathrm{k}} \mu_{\mathrm{iv}}^2 \cdot \mathrm{d}\left(\square_{\mathrm{i}}, \square_{\mathrm{p_v}}\right)}{\mathrm{n} \cdot \min\left[\mathrm{d}^2\left(\square_{\mathrm{p_v}}, \square_{\mathrm{p_w}}\right) \mid \square_{\mathrm{p_v}}, \square_{\mathrm{p_w}} \in \mathrm{C}_1, \ ..., \ \mathrm{C}_{\mathrm{k}}; \ \square_{\mathrm{p_v}} \neq \square_{\mathrm{p_w}}\right]} . \tag{13}$$

The objects $\square_{\mathrm{p_v}}$ and $\square_{\mathrm{p_w}}$ denote the prototypes of the clusters $\mathrm{C_v}$ and $\mathrm{C_w}$, respectively, which may be interpreted as ”artificial, representative objects” of the clusters. In the desired case of a high similarity between the objects of the particular clusters and, simultaneously, of a distinctive dissimilarity between objects from different clusters, the separation degree 13 takes on small values.

### 3.3. Algorithmic Procedure

The concept from Section 3.1 together with the method of fuzzy cluster analysis from Section 3.2 is now taken as the basis to formulate an algorithm for fuzzy cluster design. This is comprised of the following six steps:

*Step I.* Fuzzy structural parameters and fuzzy parameters of fuzzy probability distributions that are chosen as fuzzy design parameters are initially defined in such a way that they cover a proper design domain. The parameter values taken into account by this means must be technically reasonable and practically realizable. Design constraints $\mathrm{CT_h}$, h = 1, ..., q, are formulated according to the particular problem. Fuzzy structural analysis and fuzzy probabilistic safety assessment are performed on the basis of $\alpha$-level optimization. The final results as well as the intermediate results, which are accumulated during the included repeated optimization, are summarized in the point sets $M_{\mathrm{x}} = \{\underline{\mathrm{x}}_{\mathrm{i}};\ \mathrm{i} = 1, \ ..., \ \mathrm{n}\}$ and $M_{\mathrm{z}} = \{\underline{\mathrm{z}}_{\mathrm{i}};\ \mathrm{i} = 1, \ ..., \ \mathrm{n}\}$ and plotted in the support subspaces $\underline{X}_{\alpha=0}$ and $\underline{Z}_{\alpha=0}$, respectively.

*Step II.* The result points $\underline{\mathrm{z}}_{\mathrm{i}}$in $\underline{Z}_{\alpha=0}$ are evaluated by checking the design constraints $\mathrm{CT_h}$. This permits the separation of the point set $M_{\mathrm{z}}$ into the subset $M_{\mathrm{z}^+}$ comprising all points $\underline{\mathrm{z}}_{\mathrm{i}}^{+}$ proven to be permissible and the subset $M_{\mathrm{z}^-}$ containing all nonpermissible points $\underline{\mathrm{z}}_{\mathrm{i}}^{-}$,

$$M_{\mathrm{z}} = \{M_{\mathrm{z}^+}, \ M_{\mathrm{z}^-}\} \ . \tag{14}$$

With the aid of the inverse assignment 7, which is known from $\alpha$-level optimization, the assigned permissible points $\underline{\mathrm{x}}_{\mathrm{i}}^{+}$ and nonpermissible points $\underline{\mathrm{x}}_{\mathrm{i}}^{-}$ in $\underline{X}_{\alpha=0}$ are determined. Accordingly, the point set $M_{\mathrm{x}}$ is subdivided into $M_{\mathrm{x}^+} = \left\{\underline{\mathrm{x}}_{\mathrm{i}}^{+}\right\}$ and $M_{\mathrm{x}^-} = \left\{\underline{\mathrm{x}}_{\mathrm{i}}^{-}\right\}$,

$$M_{\mathrm{x}} = \{M_{\mathrm{x}^+}, \ M_{\mathrm{x}^-}\} \ . \tag{15}$$

*Step III.* Fuzzy cluster analysis is applied separately to the point sets $M_{x+}$ and $M_{x-}$. The set $M_{x+}$ is decomposed into $k_1$ permissible clusters. From the set $M_{x-}$ $k_2$ nonpermissible clusters are determined. The search for an appropriate cluster configuration is performed in two stages. First, clustering is carried out for a varying number of clusters $k_1$ and $k_2$. Second, the obtained cluster configurations are assessed based on the numerical quality measures 12 and 13 to select the most suitable clustering. In addition to the values of these quality measures, their relative variation when changing the number of clusters is taken into consideration. Demanding a minimum cluster membership $\mu_{iv}$ of all points $\underline{x}_i^+$ and $\underline{x}_i^-$ the size of the clusters and hence the intersections between permissible and nonpermissible clusters may be reduced. Permissible clusters may be merged to obtain superclusters covering a bigger domain of permissible points $\underline{x}_i^+$, which may be advantageous when finally defining the structural design.

*Step IV.* The $k_1$ permissible clusters $C_v$ are taken as the basis for constructing modified uncertain structural parameters of the first generation. Intersections with nonpermissible clusters are removed from the permissible ones. The remaining reduced clusters $C_{v,red}$ then comprise exclusively permissible parameter combinations to a high probability. The boundaries of the reduced permissible clusters are used to form the $\alpha$-level sets $X^{[v]}_{1,\,\alpha=0}$, ..., $X^{[v]}_{r,\,\alpha=0}$ for constructing the modified fuzzy design parameters $\tilde{x}^{[v]}_1$, ..., $\tilde{x}^{[v]}_r$ with $v = 1, \ldots, k_1$. Each reduced permissible cluster yields one sequence of $\alpha$-level sets $X^{[v]}_{1,\,\alpha=0}$, ..., $X^{[v]}_{r,\,\alpha=0}$. That is, a total of $k_1$ sequences are generated. The associated support subspaces $\underline{X}^{[v]}_{\alpha=0}$ may intersect each other. With the sets $X^{[v]}_{1,\,\alpha=0}$, ..., $X^{[v]}_{r,\,\alpha=0}$ the supports of the modified fuzzy design parameters are already determined. The membership functions of the $\tilde{x}^{[v]}_1$, ..., $\tilde{x}^{[v]}_r$ may generally be constructed with arbitrary shapes. As simple shapes are usually preferred, e.g., fuzzy triangular numbers may be used. Their mean values may be determined, e.g., by considering the cluster centers, the prototypes, or other specifically selected points. Moreover, the curves of the membership functions may be oriented to, e.g., the membership values of the points resulting from fuzzy cluster analysis, the local density of points in the cluster, or other characteristics and properties of the cluster. Design parameter values that are preferred due to constructional convenience may be furnished with higher membership values. Generally speaking, all concepts of fuzzification may be applied [18]. The obtained sequences of modified fuzzy design parameters $\tilde{x}^{[v]}_1$, ..., $\tilde{x}^{[v]}_r$ represent alternative structural design variants [v] of the first generation. These design variants may be partly included in each other.

*Step V.* Clustering permissible points and generating reduced permissible clusters do not guaranty that all design variants include exclusively permissible design parameter values. For the purpose of verification fuzzy structural analysis and fuzzy probabilistic safety assessment are carried out for each design variant. That is, the sequences of modified fuzzy design parameters $\tilde{x}^{[v]}_1$, ..., $\tilde{x}^{[v]}_r$, v = 1, ..., $k_1$, are introduced into $\alpha$-level optimization one after the other. This leads to $k_1$ sequences of modified fuzzy results $\tilde{z}^{[v]}_1$, ..., $\tilde{z}^{[v]}_m$, v = 1, ..., $k_1$, of the first generation with associated point sets $M^{[v]}_x = \left\{\underline{x}^{[v]}_i;\ i = 1,\ ...,\ n^{[v]}\right\}$ and $M^{[v]}_z = \left\{\underline{z}^{[v]}_i;\ i = 1,\ ...,\ n^{[v]}\right\}$.

*Step VI.* For each of the $k_1$ structural design variants of the first generation all points $\underline{z}_i^{[v]}$; $i = 1, ..., n^{[v]}$ obtained from $\alpha$-level optimization are evaluated by means of the design constraints $CT_h$. A design variant is considered permissible only if all points $\underline{z}_i^{[v]}$; $i = 1, ..., n^{[v]}$ comply with the requirements according to the $CT_h$. As soon as one point $\underline{z}_i^{[v]}$ does not meet all $CT_h$, the associated design variant [v] is considered as not permissible. If the remaining $k_1^+$ permissible design variants do not represent satisfying results for some reason or if no permissible design variants have been found, $k_1^+ = 0$, fuzzy cluster analysis according to Step III may be repeated to obtain modified design variants of the second generation. Taking account of the additional points $\underline{x}_i^{[v]}$ and $\underline{z}_i^{[v]}$ from the verification analyses and selecting an other cluster configuration may so lead to more satisfying results. The $k_1^+$ alternative permissible design variants are compared to find out the most suitable one. For this comparison the numerical assessment criteria considered in the subsequent section may be used. Finally, the selected design variant is taken as the basis for defining the structural design. The final structural design parameters must lie within the uncertainty of the permissible design parameters. They may be crisp or uncertain depending on the conditions for their specification.

### 3.4. Assessment of Alternative Design Variants

For assessing the alternative permissible structural design variants obtained in Step VI of the design procedure (Section 3.3) arbitrary criteria may be formulated and combined with each other. The following two criteria, however, may represent the most important ones being of major interest in almost all design problems.

*Criterion I - Constraint Distance.* This criterion assesses the degree of exploitation of the design constraints for a design variant. The fuzzy result variables $\tilde{z}_1^{[v]}$, ..., $\tilde{z}_m^{[v]}$ are defuzzified with the aid of suitably selected methods, see [18, 25]. For example, the centroid method and the defuzzification algorithms after Chen [7] and Jain [11] may be applied. The Chen/Jain algorithms permit a "biased" defuzzification. That is, the defuzzification can be focused on small values (e.g., when defuzzifying a fuzzy reliability index) or on large values (e.g., when evaluating an internal force). Defuzzification leads to the crisp values $z_{1\,0}^{[v]}$, ..., $z_{j\,0}^{[v]}$, ..., $z_{m\,0}^{[v]}$ for the results. Assessment Criterion I is then defined as the sum of the weighted distances $d^{[v]}(.)$ between the crisp values $z_{j\,0}^{[v]}$ and the design constraints $CT_h$,

$$A^{[v]} = \sum_{h=1}^{q} w_h \cdot d\left(CT_h,\ \left\{z_{j\,0}^{[v]}\right\}\right) \tag{16}$$

with $w_h$ representing real weighting factors. In the simple but common case that a particular design constraint $CT_h$ is given for each fuzzy result $\tilde{z}_j^{[v]}$ alone (that is, q = m and h = j) as a permissible value *perm*_$z_j$, the sum 16 may be rewritten as

$$A^{[v]} = \sum_{j=1}^{m} w_j \cdot \left| perm\_z_j - z_{j\,0}^{[v]} \right| . \tag{17}$$

The weighting factors $w_h$ are introduced for taking account of the importance of the particular constraints. A small value $A^{[v]}$ characterizes a structural design that exploits the constraints to a

high degree. On the other hand, a large $A^{[v]}$ indicates a structural design providing some reserves, e.g., in load-bearing capacity, which may be considered as advantageous.

*Criterion II - Robustness.* The designed structure is considered as being robust if the sensitivity of the result variables (structural responses or safety measures) is low with regard to fluctuations of the design parameters. That is, a robust structure is characterized by a low uncertainty of the fuzzy results $\tilde{z}_j^{[v]}$ in relation to the uncertainty of the fuzzy design parameters $\tilde{x}_k^{[v]}$. The uncertainty of fuzzy values is computed on the basis of an analog to Shannon's entropy [2, 32]. This yields the absolute uncertainty of $\tilde{z}_j^{[v]}$ as [18]

$$H(\tilde{z}_j^{[v]}) = -k \cdot \int_{z_j^{[v]} \in \tilde{z}_j^{[v]}} g\left(\mu(z_j^{[v]})\right) dz_j^{[v]} \; ; \; k > 0 \tag{18}$$

with

$$g\left(\mu(z_j^{[v]})\right) = \mu(z_j^{[v]}) \cdot \ln\left(\mu(z_j^{[v]})\right) + \left(1-\mu(z_j^{[v]})\right) \cdot \ln\left(1-\mu(z_j^{[v]})\right) \; . \tag{19}$$

The sensitivity of a structure is then defined as

$$B^{[v]} = \sum_{j=1}^{m} u_m \sum_{k=1}^{r} \frac{H(\tilde{z}_j^{[v]})}{H(\tilde{x}_k^{[v]})} \tag{20}$$

with the weighting factors $u_m$ for emphasizing particular result variables $\tilde{z}_j^{[v]}$ depending on their importance. A low value $B^{[v]}$ indicates a low sensitivity (high robustness) of the structure according to design variant [v].

## 4. Examples

### 4.1. Problem Definition

The presented concept of fuzzy structural design is demonstrated for the plane reinforced concrete frame shown in Figure 4. The geometrical and physical nonlinear behavior of the structure is numerically simulated on the basis of the analysis algorithms in [26]. Physical nonlinearities are accounted for by using material laws for reinforcement steel and concrete after OETES, which are also provided in [26]. Tension stiffening and the effects of stirrup reinforcement are accounted for in the concrete material law. As for geometrical nonlinearities, the quadratic terms in the deformation-displacement dependencies are taken into account in addition to considering equilibrium for the deformed system. This enables the algorithm to simulate large displacements and moderate rotations. The stiffness of the system is determined by numerical integration incorporated into an incremental iterative approach. The selected computational model is thus capable of considering all essential nonlinearities. The load bearing behavior of the structure is numerically simulated in a sufficiently realistic manner.

The system is modeled using three bars. Fifty integration increments are chosen for each bar and each cross section is subdivided into 60 layers. The loading process is comprised of dead weight,

horizontal load $\mathrm{P_H}$, vertical nodal loads $\nu \cdot \mathrm{P_{V0}}$, and the line load $\nu \cdot \mathrm{p_0}$. After applying dead weight the horizontal load $\mathrm{P_H}$ is introduced. Finally, $\mathrm{P_{V0}}$ and $\mathrm{p_0}$ are increased incrementally using the load factor $\nu$.

Uncertainty is present in the load factor $\nu$ and in the rotational spring stiffness $\mathrm{k}_\varphi$. For different uncertainty models applied to these structural parameters an appropriate structural design is determined with the aid of fuzzy structural analysis and fuzzy probabilistic safety assessment.

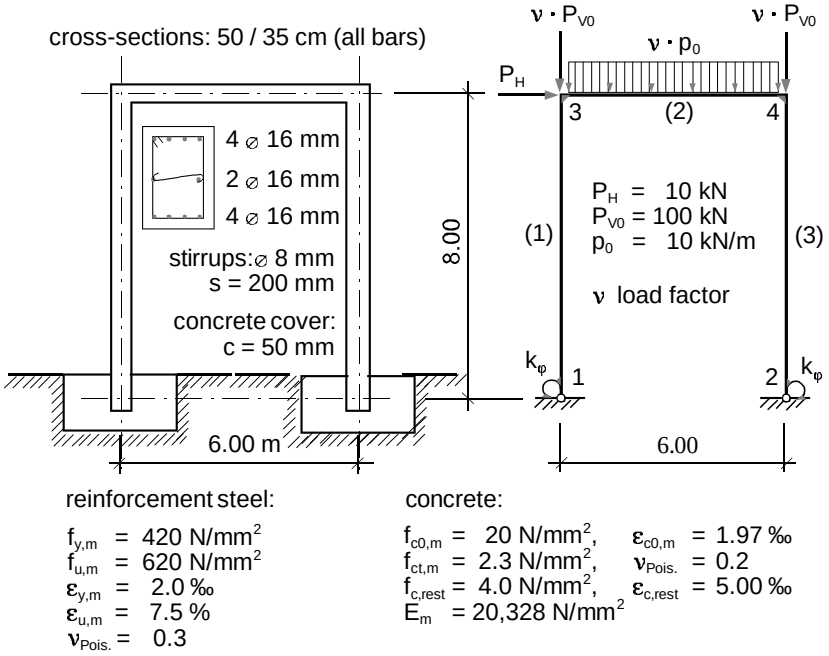



*Figure 4.* Reinforced concrete frame

## 4.2. Design with the Aid of Fuzzy Structural Analysis

The uncertainty of the structural parameters $\nu$ and $\mathrm{k}_\varphi$ is modeled as fuzziness. Load factor and rotational spring stiffness represent fuzzy design parameters. These are described with the aid of fuzzy triangular numbers,

$$\tilde{\nu} \;=\; < 5.5, 5.9, 6.7 > , \tag{21}$$

$$\tilde{\mathrm{k}}_\varphi \;=\; < 5.0, 9.0, 13.0 > \left[\frac{\mathrm{MNm}}{\mathrm{rad}}\right] . \tag{22}$$

Design constraint $\mathrm{CT}_1$ is a serviceability requirement for the maximum horizontal displacement $\mathrm{v_H}(3)$ of node 3. The permissible displacement is defined as $perm\_\mathrm{v_H}(3) = 4.0$ cm.

Fuzzy structural analysis [18, 20] yields a nonlinear fuzzy load-displacement dependency and, finally, the fuzzy result variable $\tilde{\mathrm{v}}_\mathrm{H}(3)$ for the horizontal displacement of node 3, see Figure 5.

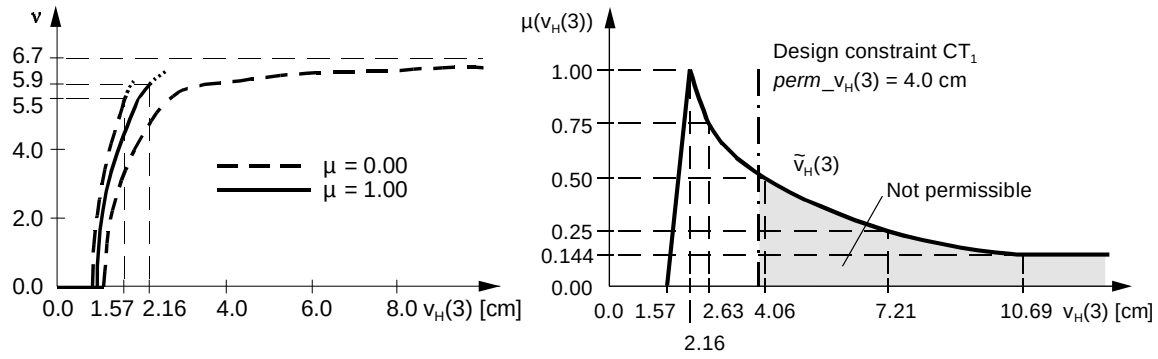



*Figure 5.* Fuzzy load-displacement dependency and fuzzy result $\tilde{\mathrm{v}}_{\mathrm{H}}(3)$

Moreover, from fuzzy structural analysis 75 points $(\nu,\ \mathrm{k}_{\varphi})_{\mathrm{i}}$ from the space of the fuzzy design parameters and the associated result values $\mathrm{v}_{\mathrm{H,i}}(3)$ are known. The fuzzy result $\tilde{\mathrm{v}}_{\mathrm{H}}(3)$ contains both permissible and nonpermissible points, see Figure 5. Accordingly, 62 permissible and 13 nonpermissible points are identified in the space of the fuzzy design parameters.

Fuzzy cluster analysis is separately applied to these permissible and nonpermissible points. A suitable cluster configuration is obtained with three permissible clusters and one nonpermissible cluster, see Figure 6.

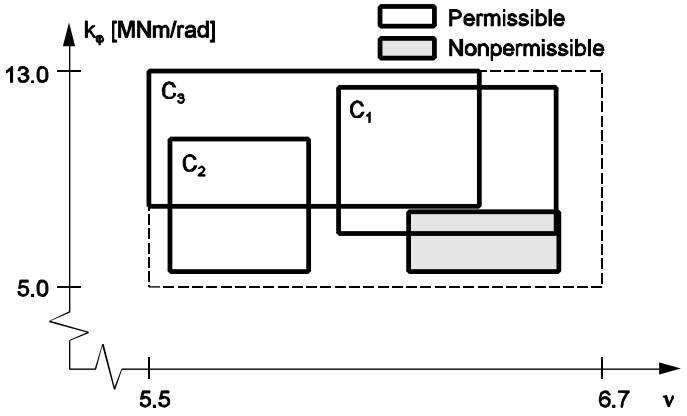



*Figure 6.* Cluster configuration

A minimum cluster membership of $\mu_{\mathrm{iv}} = 0.30$ is hereby prescribed for the assignment of objects to the clusters. Cluster $\mathrm{C}_1$ possesses some intersection with the nonpermissible cluster and is thus excluded from the further design procedure. Clusters $\mathrm{C}_2$ and $\mathrm{C}_3$ cover only permissible points $(\nu,\ \mathrm{k}_{\varphi})_{\mathrm{i}}^{+}$ and are considered as capable for generating modified fuzzy design parameters.

This yields two resulting alternative design variants,

$$\tilde{\nu}^{[2]} = < 5.56, 5.74, 5.92 > , \tag{23}$$

$$\tilde{\mathrm{k}}_{\varphi}^{[2]} = < 5.60, 8.00, 10.40 > \left[\frac{\mathrm{MNm}}{\mathrm{rad}}\right] , \tag{24}$$

and

$$\tilde{\nu}^{[3]} = < 5.50, 5.94, 6.38 > , \tag{25}$$

$$\tilde{\mathrm{k}}_{\varphi}^{[3]} = < 8.00, 10.50, 13.00 > \left[\frac{\mathrm{MNm}}{\mathrm{rad}}\right] . \tag{26}$$

For verifying the permissibility of the design variants fuzzy structural analysis is carried out for both pairs of modified design parameters. The associated fuzzy results meet the design constraint regarding the permissible displacement in either case, see Figure 7. Defuzzifying the fuzzy displacements $\tilde{\mathrm{v}}_{\mathrm{H}}^{[2]}(3)$ and $\tilde{\mathrm{v}}_{\mathrm{H}}^{[3]}(3)$ after JAIN and computing the sensitivity of the structure lead to

– cluster $C_2$: $\mathrm{v}_{\mathrm{H}\,0}^{[2]}(3)$ = 2.49 cm, $\mathrm{B}^{[2]}$ = 3.81

– cluster $C_3$: $\mathrm{v}_{\mathrm{H}\,0}^{[3]}(3)$ = 2.54 cm, $\mathrm{B}^{[3]}$ = 2.03

The results show no significant difference in the constraint distance, whereas the sensitivity of design variant [2] is almost twice as high as the sensitivity of design variant [3]. The fuzzy design parameters according to 25 and 26 corresponding to cluster $C_3$ are thus selected as being the most suitable ones. These fuzzy structural design parameters have to be ensured by an appropriate construction of the system, whereby the remaining uncertainty initialized by noncontrollable parameters, such as soil stiffness, must lie within the design uncertainty.

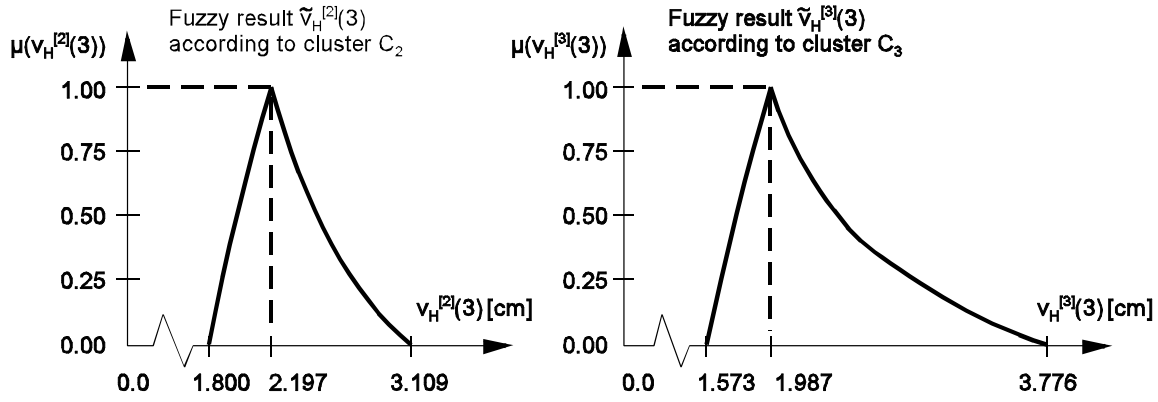



*Figure 7.* Fuzzy results according to the alternative design variants

### 4.3. DESIGN WITH THE AID OF FUZZY PROBABILISTIC SAFETY ASSESSMENT

The reinforced concrete frame shown in Figure 4 is designed with an alternative modeling of the structural parameters. Both the load factor $\nu$ and the rotational spring stiffness $\mathrm{k}_\varphi$ are described with fuzzy random variables,

$$\nu \;\rightarrow\; \widetilde{\mathrm{X}}_1 \,, \tag{27}$$

$$\mathrm{k}_\varphi \;\rightarrow\; \widetilde{\mathrm{X}}_2 \,. \tag{28}$$

The load factor is assumed to follow an extreme value distribution of Ex-Max Type I. The spring stiffness is described by a logarithmic normal distribution. The expected values m and standard deviations σ of both distributions are modeled as fuzzy triangular numbers,

$$\tilde{\mathrm{m}}_{\mathrm{X}_1} \;=\; < 5.70, 5.90, 6.30 > \,, \tag{29}$$

$$\tilde{\sigma}_{\mathrm{X}_1} \;=\; < 0.08, 0.11, 0.15 > \,, \tag{30}$$

and

$$\tilde{\mathrm{m}}_{\mathrm{X}_2} \;=\; < 8.50, 9.00, 10.00 > \left[\frac{\mathrm{MNm}}{\mathrm{rad}}\right] , \tag{31}$$

$$\tilde{\sigma}_{\mathrm{X}_2} \;=\; < 1.00, 1.35, 1.50 > \left[\frac{\mathrm{MNm}}{\mathrm{rad}}\right] . \tag{32}$$

The minimum value of the logarithmic normal distribution of $\mathrm{k}_\varphi$ is assumed to be crisp with $\mathrm{x}_{0,2} = 0\mathrm{MNm/rad}$. The fuzzy parameters $\tilde{\mathrm{m}}_{\mathrm{X}_1}$, $\tilde{\sigma}_{\mathrm{X}_1}$, $\tilde{\mathrm{m}}_{\mathrm{X}_2}$, and $\tilde{\sigma}_{\mathrm{X}_2}$ are chosen as fuzzy design parameters. Design constraint $\mathrm{CT}_1$ is the required reliability index $req\_\beta = 3.8$ for global system failure.

The actual safety level is computed on the basis of the Fuzzy First Order Reliability Method (FFORM)[18, 21]. This yields the fuzzy reliability index $\widetilde{\beta}$ shown in Figure 8. From $\alpha$-level optimization a total of 609 result points are known in $\widetilde{\beta}$, which are subdivided into 414 permissible points and 195 nonpermissible points by means of the design constraint $\mathrm{CT}_1$. Also, the associated permissible and nonpermissible points in the four-dimensional space of the fuzzy design parameters are known from $\alpha$-level optimization.

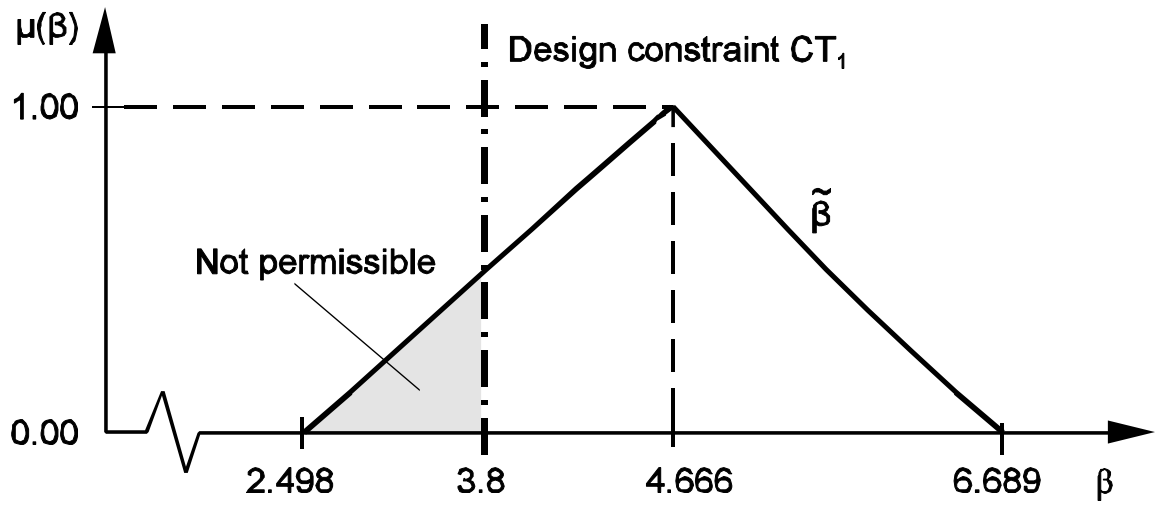



*Figure 8.* Fuzzy result $\widetilde{\beta}$ and design constraint $\mathrm{CT}_1$

The fuzzy cluster method is separately applied to permissible and nonpermissible points in the space of the fuzzy design parameters. The number of clusters is varied within the interval $[1,\ 10]$. The obtained cluster configurations are assessed on the basis of the numerical quality measures from Section 3.2. As a compromise a clustering with $\mathrm{k}_1$ = 6 permissible clusters is considered as being suitable, see Figure 9.

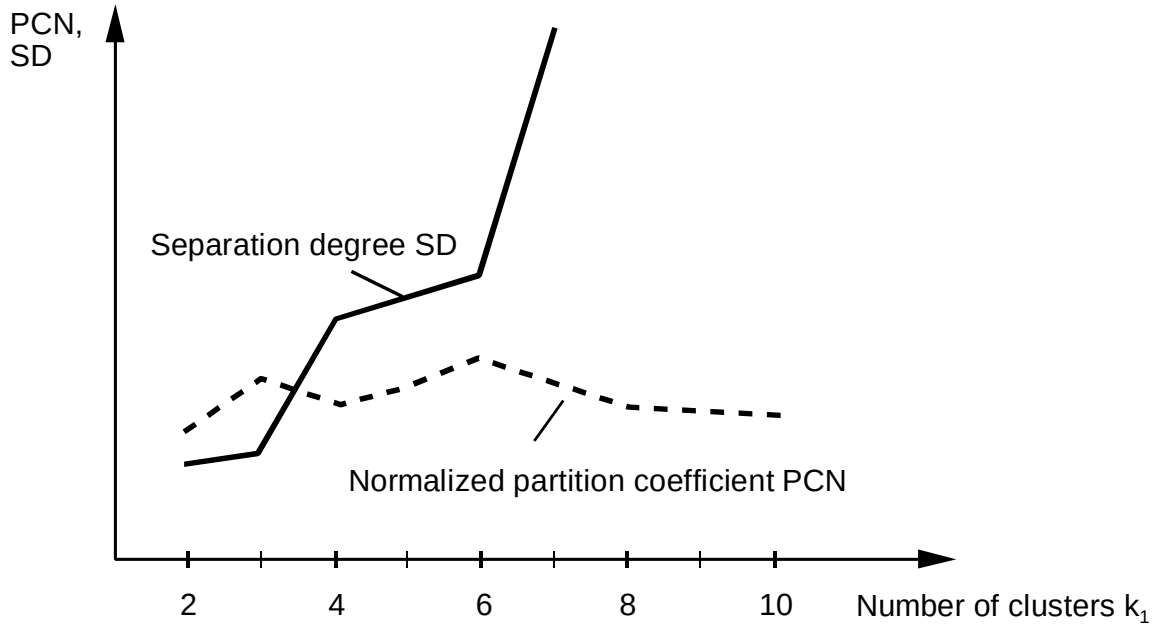



*Figure 9.* Assessing cluster configurations for permissible points

For the number of nonpermissible clusters $k_1$ = 7 is selected. With a minimum cluster membership of $\mu_{iv} = 0.25$ a cluster configuration without intersections between permissible and nonpermissible clusters is obtained. The permissible clusters $C_1$, $C_2$, $C_5$, and $C_6$ are very small in size. They cover only a very limited value range of permissible design parameters and are thus not taken as a basis for generating alternative design variants. Clusters $C_3$ and $C_4$ are approximately congruent and are merged to constitute the supercluster $C_{3,4}$. Hence, only one set of modified fuzzy design parameters is obtained,

$$\tilde{m}_{X_1}^{[3,4]} = < 5.85, 5.90, 6.00 > , \tag{33}$$

$$\tilde{\sigma}_{X_1}^{[3,4]} = < 0.08, 0.10, 0.11 > , \tag{34}$$

and

$$\tilde{m}_{X_2}^{[3,4]} = < 8.50, 9.50, 10.00 > , \tag{35}$$

$$\tilde{\sigma}_{X_2}^{[3,4]} = < 1.30, 1.40, 1.50 > \ . \tag{36}$$

For verifying this design variant FFORM is applied again with the modified fuzzy design parameters. This yields the modified fuzzy reliability index $\tilde{\beta}^{[3,4]}$, see Figure 10. All elements of $\tilde{\beta}^{[3,4]}$ comply with the design constraint $CT_1$. The modified fuzzy design parameters define modified fuzzy probability distribution functions for the modified fuzzy random variables $\tilde{X}_1$ and $\tilde{X}_2$.

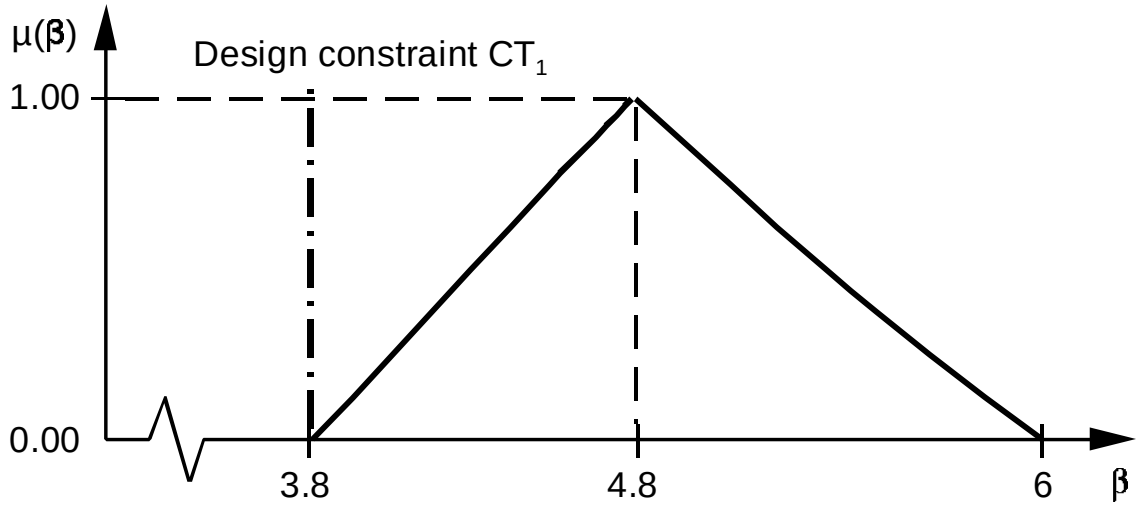



*Figure 10.* Modified fuzzy result variable $\widetilde{\beta}^{[3,4]}$

To ensure the permissibility of the structural design, the final design parameters must be elements of the modified fuzzy design parameters. This may be realized by acquiring more information about $\nu$ and $k_{\varphi}$. For example, additional prior information or additional sample elements drawn subsequently may be accounted for with the aid of a Bayesian approach. Also, a larger sample may be taken as the basis for an interval estimation of the parameters. The extent to which additional information is needed can be determined iteratively or inferred by inverting the parameter estimation problem. As a result, a minimum sample size may so be determined.

## 5. Conclusions

In this paper a concept for designing structures based on nonlinear fuzzy structural analysis (processing of fuzzy variables in nonlinear structural analysis) and fuzzy probabilistic safety assessment (processing of fuzzy random variables, real random variables, and fuzzy variables in safety assessment) has been provided. Thereby, uncertainty is accounted for as fuzziness, randomness, and fuzzy randomness of structural parameters. This is exploited to analyze an initially defined proper range of design parameter values with regard to the associated structural responses and safety levels. With the aid of fuzzy cluster analysis (to determine clusters of design parameter vectors as continuous sets of real vectors with similar properties) alternative uncertain structural design variants are generated from the analysis results. These are compared on the basis of numerical criteria to select the optimum design variant. For instance, a robust structural design may be found in this manner. The final design parameters may then be determined within the fuzziness of the optimum uncertain design variant. They are not necessarily crisp but may contain uncertainty that fits into the uncertainty of

the design variant. This is particularly suitable if the information about these parameters is limited, e.g., due to a small sample size.

This concept permits designing structures directly in combination with arbitrary nonlinear computational models and under consideration of nonstochastic uncertainty.

## Acknowledgements

The authors gratefully acknowledge the support of the Alexander von Humboldt-Foundation (AvH) and the German Research Foundation (DFG).

## References

1. Alefeld, G. and Herzberger, J.: *Introduction to Intervall Computations.* Academic Press, New York, 1983.
2. Bandemer, H. and Gottwald, S.: *Einführung in FUZZY-Methoden.* Wissenschaftliche Taschenbücher Mathematik/Physik, Band 305, Akademie-Verlag, Berlin 1989.
3. Beer, M., Möller, B. and Liebscher, M.: Processing uncertainty in structural analysis, design, and safety assessment. In: Ayyub, B.M. and Attoh-Okine, N.O. (eds.): *Fourth International Symposium on Uncertainty Modeling and Analysis (ISUMA)*, College Park, MD, 34 - 39, CD-ROM Doc. 07, 1 - 6, 2003.
4. Ben-Haim, Y.: *Robust Reliability in the Mechanical Sciences.* Springer-Verlag, Berlin, Heidelberg, New York 1996.
5. Ben-Haim, Y. and Elishakoff, I.: *Convex Models of Uncertainty in Applied Mechanics.* Elsevier, Amsterdam 1990.
6. Bernardo, J.M. and Smith, A.F.M.: *Bayesian Theory*. Wiley, Chichester, New York, Brisbane, Toronto, Singapore, 1994.
7. Chen, S. H.: Ranking fuzzy numbers with maximizing and minimizing set. *Fuzzy Sets and Systems* **17** (1985), pp. 113-129.
8. Dessombz, O., Thouverez, F., Laineé, J.-P. and Jézéquel, L.: Analysis of Mechanical Systems using Interval Computations applied to Finite Elements Methods. *Journal of Sound and Vibration* **239** (5) (2001), pp. 949 - 968.
9. Elishakoff, I.: *Whys and Hows in Uncertainty Modelling, Probability, Fuzziness and Anti-Optimization*. Springer-Verlag, Wien, New York 1999.
10. Höppner, F., Klawonn, F., Kruse, R. and Runkler, T.: *Fuzzy cluster analysis: Methods for classification, data analysis and image recognition*. Wiley, 1999.
11. Jain, R.: Decisionmaking in the presence of variables. IEEE Transactions on Systems, *Man, and Cybernetics* **6** (1976), pp. 698-703.
12. Kadane, J.B., Moreno, E., Perez, M.E. and Pericchi, L.R.: Applying non-parametric robust Bayesian analysis to non-optionated judicial neutrality. *Journal of Statistical Planning and Inference* **102** (2002), pp. 425 - 439.
13. Kapitaniak, T.: *Chaos for Engineers: Theory, Applications, and Control*. 2nd revised edition, Springer-Verlag, Berlin, Heidelberg, New York 2000.
14. Kaufmann, L. and Rousseeuw, P. J.: *Finding groups in data: An introduction to cluster analysis*. 3. Auflage, Wiley, 1990.
15. Krätschmer, V.: A unified approach to fuzzy random variables. *Fuzzy Sets and Systems* **123** (2001), pp. 1 - 9.
16. Li Bing, Zhu Meilin and Xu Kai: A practical engineering method for fuzzy reliability analysis of mechanical structures. *Reliability Engineering and System Safety* **67** (2000), pp. 311 - 315.
17. Madsen, H.O., Krenk, S. and Lind, N.C.: *Methods of Structural Safety*. Prentice-Hall, Englewood Cliffs, New Jersey 1986.
18. Möller, B. and Beer, M.: *Fuzzy Randomness - Uncertainty Models in Civil Engineering and Computational Mechanics*. Springer, Berlin, Heidelberg, New York, 2004.

19. Möller, B., Beer, M. and Liebscher, M.: Fuzzy Cluster Design - A New Way for Structural Design. In: Topping, B.H.V. (ed.): *Proceedings of the Ninth International Conference on Civil and Structural Engineering Computing*, Egmond aan Zee, The Netherlands Civil-Comp Press, Stirling, UK, CD-ROM, paper 119, pp. 1 - 14, 2003.
20. Möller, B., Graf, W. and Beer, M.: Fuzzy structural analysis using $\alpha$-level optimization. *Computational Mechanics* **26** (6) (2000), pp. 547 - 565.
21. Möller, B., Graf, W. and Beer, M.: Safety assessment of structures in view of fuzzy randomness. *Computers and Structures* **81** (15) (2003), pp. 1567 - 1582.
22. Noor, A.K., Starnes JR., J.H. and Peters, J.M.: Uncertainty analysis of stiffened composite panels. *Composite Structures* **51** (2001), pp. 139 - 158.
23. Pawlak, Z.: Rough sets. *Theoretical aspects of reasoning about data*. Kluwer Academic Publishers, 1991.
24. Penmetsa, R.C. and Grandhi, R.V.: Efficient estimation of structural reliability for problems with uncertain intervals. *Computers and Structures* **80** (12) (2002), pp. 1103 - 1112.
25. Rommelfanger, H.: *Entscheiden bei Unschärfe – fuzzy decision Support-Systeme*. Springer-Verlag, Berlin, Heidelberg 1988.
26. Schneider, R.: *Stochastische Analyse und Simulation des nichtlinearen Verhaltens ebener Stabtragwerke mittels M-N-Q-Interaktionsmodell.* Veröffentlichungen des Lehrstuhls für Statik, Heft 2, TU Dresden 2001.
27. Schuëller, G.I. and Spanos, P.D. (eds): *Monte Carlo Simulation: Proceedings of the International Conference on Monte Carlo Simulation*, Monaco, 2000, A.A. Balkema, Lisse, Exton, PA, 2001.
28. Timm, H.: *Fuzzy-Clusteranalyse: Methoden zur Exploration von Daten mit fehlenden Werten sowie klassifizierten Daten*. Ph.D. thesis, Magdeburg 2002.
29. Tonon, F. and Bernardini, A.: A random set approach to the optimization of uncertain structures. *Computers and Structures* **68** (6) (1998), pp. 583 - 600.
30. Viertl, R.: *Statistical Methods for Non-Precise Data*. CRC Press, Boca Raton, New York, London, Tokyo, 1996.
31. Wright, G. (ed.): *Subjective probability*. Wiley, Chichester 1994.
32. Zimmermann, H.-J.: *Fuzzy set theory and its applications*. Kluwer Academic Publishers, Boston, London 1992.

# Handling Uncertainty in the Development and Design of Chemical Processes

David Bogle, David Johnson, and Sujan Balendra
*Dept of Chemical Engineering, University College London,Torrington Place, London, U.K. WC1E 7JE*
*e-mail:* d.bogle@ucl.ac.uk

**Abstract.** The paper presents a stochastic methodology for handling uncertainty in process development as part of a general framework for batch and continuous process models. The method combines systematic modelling procedures with Hammersley sampling based uncertainty analysis and a range of sample-based sensitivity analysis techniques, used to quantify predicted performance uncertainty and identify key uncertainty contributions. The methodology was implemented on a batch reactor process and some clear recommendations as to how to reduce the uncertainty in the main output variables are obtained. The paper concludes with some discussion about an alternative approach to use instead error bars from experimental data as intervals and using interval methods to determine the best 'worst-case' design.

## 1. Introduction

In the development of new chemical manufacturing processes, particularly in the pharmaceutical industry, there is a large element of process uncertainty since detailed knowledge of the chemical reaction mechanisms and of the power and effectiveness of separation devices (to purify the product and recover raw materials) is often limited. Data is obtained from the laboratory during the identification stage of a new product (for example a new drug) and this is used during the manufacturing process development stage. Much data is generated but often not useful for the development of the large scale manufacturing process. In some cases large amounts of data are available but often single data points with confidence limits are obtainable in the form of interval bounds. Using a structured approach with the computational process design tools, which are used extensively, the uncertainty can be managed and improved process performance may be obtained. The methodology proposed is based on a stochastic formulation but the use of interval methods which have a natural role arising from the form of data used is also discussed briefly.

This work was undertaken for process development in the pharmaceutical industry. New products are constantly being proposed and the decision about whether to proceed with development depends on the efficacy of the drug but also whether the manufacturing process will be possible and will make a profit. It is therefore necessary to develop new processes but this is often done without regard to the data being obtained in the laboratory and without consideration of the accuracy of that data. The objective was to develop a model based approach that could help identify major causes of uncertainty and hence help to direct when better data is required. Much good data is developed but often the data required for manufacturing process development is not available or of poor quality.

Pharmaceutical processes typically consist of a sequence of unit operations, for example reactors and separation devices. It is important that the methodology is able to handle a large sequence of units as well as single units. The main causes of uncertainty in process development are in the data obtained about reaction and separation which are then used in the model and also in the quality of the raw materials that are used in the reactions. Assumptions about models are also uncertain which can cause the mathematical structure to be incorrect as well as the model parameters, for example in the case of the order of the reaction kinetics.

While the approach was developed with pharmaceutical processes in mind it could in fact be used for any type of process.

## 2. A Methodology for Design Under Uncertainty: Combined risk analysis and systematic model development

The proposed methodology aims to introduce some form of management of the uncertainty associated with the model representations of the current process knowledge. It is assumed that the conceptual process design (equipment allocation and design) is already decided. The deterministic process models may exhibit non-linear and dynamic characteristics as may be expected in typical pharmaceutical processes. However, spatially distributed models are not considered for computational reasons.

In the face of large amounts of uncertainty predicted in the important process output criteria, three issues have been considered:

(i) reduction of the uncertainty by improving current models/parameter estimations associated with the key contributing uncertainty factors identified,
(ii) manipulation of the available process decisions (operating policy) to improve process robustness to model uncertainty,
(iii) consideration of process alternatives.

Issue (i) concerns the gathering of additional information for systematic model development for more reliable models. Issues (ii) and (iii) concern the optimisation and comparison of uncertain but integrated processes sequences. This will be dealt with in a future paper.

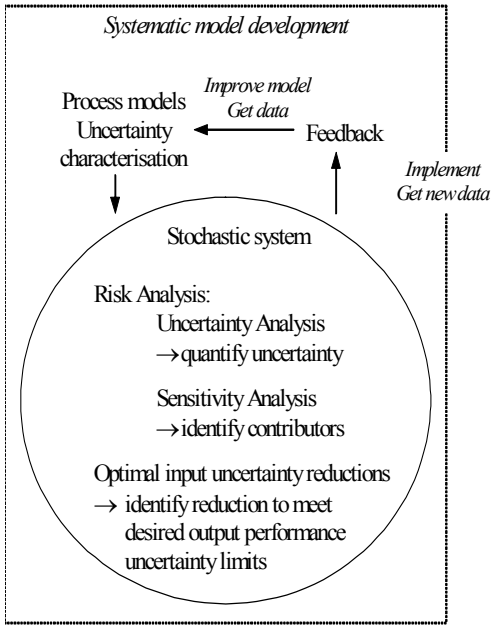



Figure 1. Management of uncertainty in a model-based approach to integrated design under uncertainty for pharmaceutical processes.

The elements of the Risk Analysis approach have been combined with systematic procedures for the development of deterministic process models (Figure 1). A stochastic representation of the complete model of the integrated process sequence is generated to quantify modelling uncertainties and to identify and rank the most important contributors in the uncertain (but

structured) system with respect to the important system responses. The suggestion is that this information can be used to drive the general direction for data collection (within process development) to improve the key models and reduce the uncertainty in the most significant areas. As more data becomes available the methodology allows the tracking of the effect of increased knowledge in certain process models and the effect this has on the complete system, in an iterative manner. A key issue is the flexibility of the approach to incorporate new data into the analyses. A more detailed schematic of the approach is shown in Figure 2. The reader is referred to Hangos and Cameron (2001) for further detail about conventional model development.

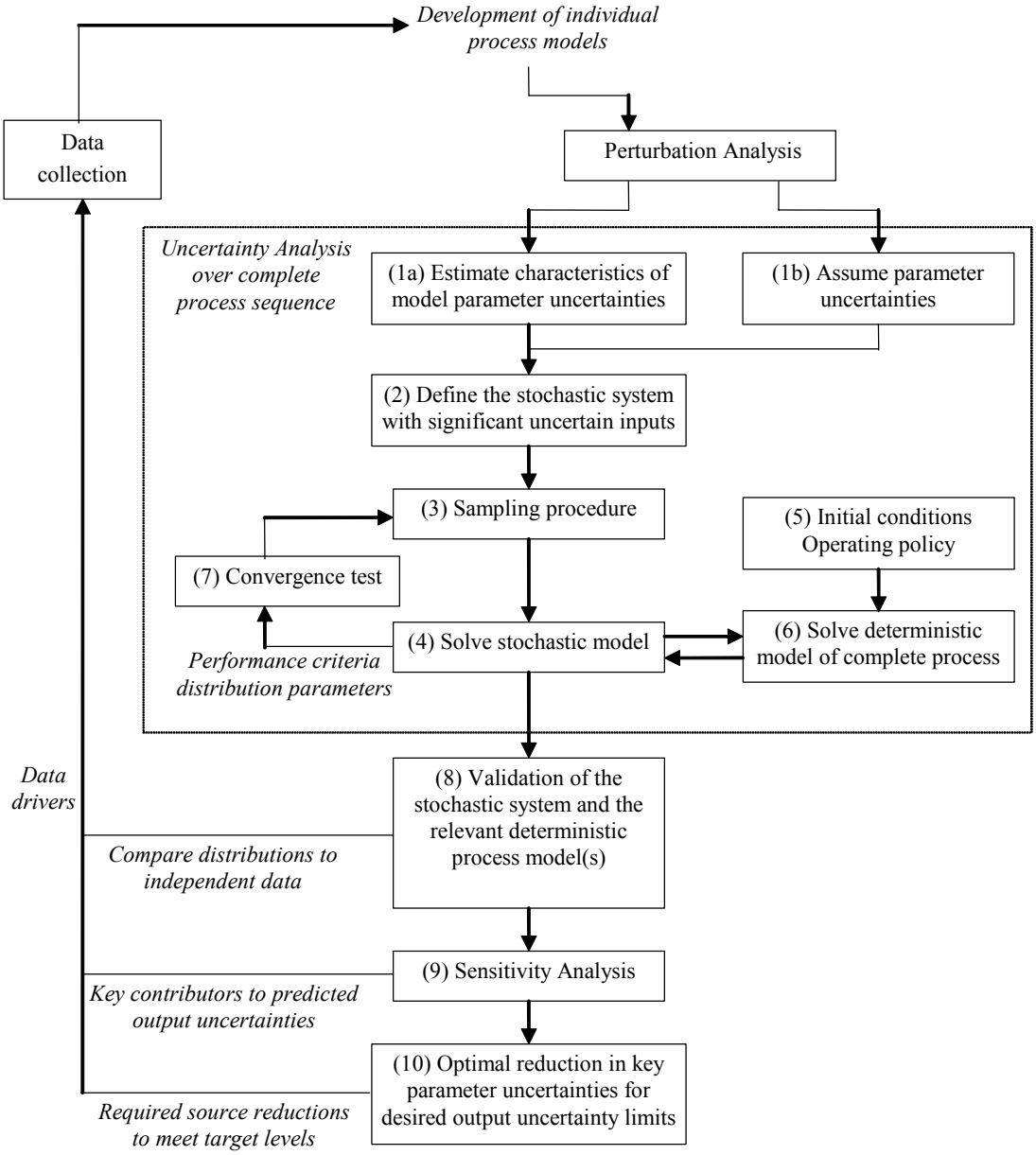



Figure 2. Schematic for the systematic model development incorporating the Risk Analysis approach under uncertainty.

Once a model has been developed and before the Risk Analysis methods are implemented a screening procedure is used to determine which of the parameter uncertainties in the complete

sequence model may be potentially significant regarding the response variables in the stochastic representation of the system. This is necessary when the number of parameters increases and it is necessary to limit the number of dimensions in the following Risk Analysis. For this purpose a Perturbation Analysis (one at a time approach) is used since it is a common and easy method to implement. The deterministic model is systematically simulated at positive and negative deviations from the nominal parameter estimates. The magnitude of the deviations may be based on the judgement of the developer (for example these could be the approximate precision ranges for different types of model parameters as suggested by Hangos and Cameron, 2001), or at estimated confidence limits if available.

## 3. Uncertainty Analysis

Following the selection of the significant uncertain parameters, Steps 1 to 7 in figure 2 comprise the elements of Uncertainty Analysis. The information required for the stochastic distributions of the input variables needs to be developed typically from sparse data sets so the methodology includes the development of these distributions. In Step 1 the approach used for the quantitative estimation of the uncertainties in these parameters is determined by the data available for parameter estimation which may be based on three different information sources:

- analysis of the performance of the model building based on experimental measurement data (Step 1a), allows the estimation of uncertainty in the parameter estimates using confidence intervals or regions,
- expert technical judgement is needed when quantitative data is not available for systematic model building and models are assumed whose parameter values are instead based on observations and/or assumed along with associated confidence intervals and probability distributions (Step 1b),
- either specific published information or information from which judgements can be inferred (Step 1a or 1b).

If parameters are estimated or assumed independently of each other, the joint sampling space may be described as a hyper-rectangle where each dimension represents one uncertain input bounded by its respective upper and lower confidence limits. If no data is available for model parameter estimation, confidence limits around the nominal values are assumed as some percentage of the nominal. For uncertainty in independent parameters of assumed nominal values, to be characterised by normal distributions, the standard deviation is assumed at some percentage of the nominal value. Confidence limits around the nominal value can be assumed at some number of standard deviations (typically two or three deviations for approximately 95 or 99.9% probability of containment according to Chebyshev's rule).

Least squares regression is a commonly used parameter estimation method for which confidence intervals can be simply stated assuming normally distributed uncertainty. Although likelihood and lack of fit are more accurate methods for estimating parameter confidence regions for non-linear models, Donaldson and Schnabel (1987) conclude from their general study on regression parameter confidence regions that the linearization methods provide the most concise representation of information required to construct confidence intervals and regions.

For a model that is non-linear in its parameters, individual confidence intervals can be approximated assuming a linearization of the model about its optimal estimated parameter values, $\theta'$,

$$\left|\theta_p - \theta'_p\right| \le \left(\hat{V}_{pp}\right)^{\frac{1}{2}} t_{N-P,1-\frac{\alpha}{2}} \tag{1}$$

where subscript p is the index of the input uncertainty ($\theta$), V is the covariance matrix, and the values of the confidence limits are defined where the value of t is taken from the Student's t-test

distribution with N-P degrees of freedom (number of regression data points, N, and number of parameters, P, in the regression model), assuming a desired level of confidence, 1-α. In a multi-parameter model where the parameters are estimated simultaneously, a joint confidence region provides a more appropriate measure of the (normally distributed) uncertainty space that would be a hyper-rectangle comprising the individual confidence intervals. Similarly, for a non-linear model, a hyper-ellipsoidal joint confidence region is approximated by,

$$\left(\theta-\theta'\right)^T \hat{V}^{-1}\left(\theta-\theta'\right) \le PF_{N,N-P,1-\alpha} \tag{2}$$

where θ is a vector of the model parameters and the value of F is taken from the F distribution. This is the distribution of a random variable, F, defined as the ratio of two independent chi-squared random variables divided by their respective degrees of freedom.
Linearization methods for the estimation of confidence intervals and regions require the estimation of the parameter covariance matrix. Donaldson and Schnabel (1987) state that the most common and easily computed estimate for the covariance matrix is,

$$\hat{V} = s^2\left(J\left(\theta'\right)^T J'\right)^{-1} \tag{3}$$

where $J\left(\theta'\right)$ is the Jacobian matrix of the model predictions at the optimal parameter estimates (i.e. the N × P matrix with the (n, p)th element estimated by $\partial f(x_n, \theta)/\partial\theta_p$ at θ′, for N data points and P parameters), $s^2$ is the estimated residual variance computed from the residual sum of squares (RSS) between the regression model predictions, $\hat{\Phi}$, at the optimum parameter estimates and the measurement data, Φ,

$$s^2 = \frac{RSS\left(\theta'\right)}{\left(N-P\right)} \tag{4}$$

$$RSS\left(\theta'\right) = \sum_{n=1}^{N}\left(\Phi_n - \hat{\Phi}_n\left(\theta'\right)\right)^2 \tag{5}$$

and where $\hat{V}_{pp}$ is the ppth element of the covariance matrix, $\hat{V}$, and is the variance estimate of the pth model parameter (input uncertainty). J is estimated numerically using the first order Taylor's approximation J by introducing deviations into each optimal parameter value in turn and re-evaluating the change in the predicted dependent variable at each data point.
Given the covariance matrix it is straightforward to determine the correlation matrix, $\hat{C}$,

$$\hat{V} = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \cdots & \rho_{1P}\sigma_1\sigma_P \\ \rho_{21}\sigma_2\sigma_1 & \sigma_2^2 & \cdots & \rho_{2P}\sigma_2\sigma_P \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{P1}\sigma_P\sigma_1 & \rho_{P2}\sigma_P\sigma_2 & \cdots & \sigma_P^2 \end{bmatrix} \quad \Rightarrow \quad \hat{C} = \begin{bmatrix} 1.0 & \rho_{12} & \cdots & \rho_{1P} \\ \rho_{21} & 1.0 & \cdots & \rho_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{P1} & \rho_{P2} & \cdots & 1.0 \end{bmatrix} \tag{6}$$

where ρ is the correlation coefficient and σ is the parameter standard deviation (determined from the square root of the parameter variances, $\sigma^2$, in the leading diagonal of the covariance matrix).

Step 2 defines the stochastic system considered by combining the deterministic process model sequence with the uncertain parameter characterisations as obtained in step 1.

A sampling procedure is invoked in Step 3 in order to approximate the uncertain system. In this methodology, the quasi-Monte Carlo Hammersley Sequence Sampling (HSS, Diwekar and Kalagnanam, 1997) is implemented. Sampling approaches are the most flexible because of their capacity to capture different perspectives of risk, the examination of the entire Θ-space, and they are not severely limited by the number of dimensions, of which HSS appears to be the most efficient. A unit hyper-rectangle of dimension P is sampled using HSS in Step 3.

Diwekar and Kalagnanam (1997) define the M points of the Hammersley sequence variant, $e_p(m)$ in a P-dimensional hyper-cube. Rank correlation coefficients are a meaningful way to describe dependencies between stochastic inputs. The desired rank correlation matrix, $\hat{C}^*$, of a matrix of independently generated sample input column vectors, X, is set as equal to $\hat{C}$ (the desired correlation matrix of X). A new matrix, K, is defined which has the same dimension as X but is independent of X, to give a correlation matrix close to the identity matrix. These are inverted over the standard normal cumulative distribution and the elements in K are rearranged to obtain the correlation structure defined by $\hat{C}$, to give a matrix, $K^*$. Not only is it necessary that the correlation matrix of K is close to the identity matrix but also that the correlation and rank correlation matrices of $K^*$ should be close to each other.

The stochastic system is solved in Step 4, to obtain probability distributions and distribution parameters for the desired process performance variables (i.e. the desired output variables). This is achieved by sequential simulation of the deterministic model in Step 6 at each observation of the uncertain parameters and at the initial conditions and operating conditions fixed in Step 5, given the matrix of stochastic input observations with any induced correlation structures ($X^*$) and the deterministic model of the complete flowsheet. To terminate the solution of the stochastic model, a convergence test is employed (Step 7). The convergence test used in this methodology is a tolerance limit on the average sum of squared deviations measured in a distribution parameter, w, over all previous iterations up to the current iteration observation, $m_i$. This limit, Δw, for the qth predicted process output quality criterion is shown in Equation 7,

$$\Delta w_q = \frac{1}{m_i}\sqrt{\sum_{m=1}^{m_i}\left(w_{q,m} - w_{q,m_i}\right)^2} \le \varepsilon_{w,q} \qquad (7)$$

where w is the mean or variance estimate from all the previous observations and ε is a permitted tolerance. The test requires that tolerances on the mean and variance parameters characterising the distributions in the key outputs are met.

## 4. Validation

The individual models of the process sequence need to be validated with available independent data of good quality. In the case of using data from a pilot plant run, data for individual operations may not be available since measurements are not taken at all points in the sequence. Here, validation may only be possible over sub-sequences of integrated models. In Step 8, a form of statistical model validation compares distributions of performance predicted from Uncertainty Analysis with independent data to validate the uncertain sequence model. Both the location and spread of the predicted distributions in relation to the independent data are important in the validation. Independent data may already be available from previous runs or if resources permit from specific model validation runs (for specific operations). As stated by Basu et al. (1999)

there should be plenty of opportunities to obtain more data for this purpose given the nature of pharmaceutical process development.

Following this, Sensitivity Analysis is used to estimate the ranking priority of the key stochastic inputs contributing to the uncertainty in the stochastic process output criteria (Step 9). In an efficient manner the sensitivity techniques employed in this methodology reuse the sample results generated from Uncertainty Analysis to avoid the need for any further simulations of the deterministic model.

Standardised regression coefficients (SRC) may be compared to correlation coefficients (CC) to avoid the affect of spurious correlations to which the CCs are susceptible. SRCs may be calculated either from first determining the linear regression coefficients, $b_p$, and then multiplying these by the parameter sample standard deviation, $s_\theta$, and dividing by the output standard deviation, $s_\Phi$,

$$SRC_p = \frac{b_p s_{\theta_p}}{s_\Phi} \tag{8}$$

or by standardising the raw sample data and then applying the regression,

$$\theta_{p,std,m} = \frac{\theta_{p,m} - \overline{\theta}_p}{s_{\theta_p}} \quad , \quad \hat{\Phi}_{std,m} = \frac{\hat{\Phi}_m - \overline{\Phi}}{s_\Phi}$$

$$\hat{\Phi}_{std,m} = \sum_{p=1}^{P} SRC_p \theta_{p,std,m} \tag{9}$$

where $\overline{\theta}$ and $\overline{\Phi}$ are the sample means of $\theta$ and $\hat{\Phi}$, and the subscript 'std' represents a standardised value. To avoid over-fitting problems in determining SRCs, stepwise regression procedures are employed.

The input sample set is split into a number of disjoint intervals which each contain an equal number of observations. In this way the conditional means of the outputs at given values of the inputs can be approximated for the first order $CR_p$ for $\theta_p$,

$$CR_p^2 = \frac{Var_{\theta_p}\left\{E\left\{\Phi\middle|\theta_p\right\}\right\}}{Var\left\{\Phi\right\}} = 1 - \frac{E_{\theta_p}\left\{Var\left\{\Phi\middle|\theta_p\right\}\right\}}{Var\left\{\Phi\right\}} \tag{10}$$

where $\Phi$ is the vector of deterministic model performance outputs, $\theta_p$ is the vector of observations in the pth uncertain input, $Var_{\theta_p}$ and $E_{\theta_p}$ are the variance and expectation condition for $\theta_p$. If there is any element of doubt then scatter plots between individual input-output pairs can be viewed, though these may also be susceptible to spurious correlations.

Following identification of the critical uncertain parameters from Sensitivity Analysis, the methodology provides the possibility to determine the minimum reduction in these uncertainties required to meet desired levels of reduction in the uncertainty contained in the performance criteria of the original system (Step 10). A quantitative estimate of the minimum extent of reductions required in the important uncertainty sources to meet desired output uncertainty levels can be provided by formulating a stochastic optimisation problem. In addition, trade-off curves between different parameter reductions can be plotted by solving at different levels of desired performance uncertainty reduction.

By defining the decision variables as the fractions of the original values (before uncertainty reduction) of the parameters which characterise the spread of the parameter uncertainties and formulating the objective function, $\Im$, as a summed term of these decisions, the desired problem formulation is obtained. Since only normal and bounded range parameter uncertainties are currently considered in the combined modelling and Risk Analysis part of the methodology, the

space size characterising parameters, δ, in the optimisation are the standard deviation, σ, for normally distributed uncertainties, $p_N$, and the deviation of the limits, $\theta^{UB}$ and $\theta^{LB}$, from the mean, μ, for bounded range uncertainties, $p_U$. The values of these decisions are passed to the HSS sampling sub-routine which locates observations within the redefined uncertainty space. The new stochastic model is solved given the fixed initial conditions, operating policy and remaining set of parameters. The objective is maximised subject to inequality constraints which permit a fraction, α, of the original level of the uncertainties observed in the original output variables. A stochastic optimisation algorithm is used to solve the following problem:

$$\max_{\delta_{p_N},\delta_{p_U}} \quad \Im = \sum_{p_N=1}^{P_N} \delta_{p_N} + \sum_{p_U=1}^{P_U} \delta_{p_U} \tag{11}$$

Subject to the deterministic model equations and constraints and

$$FW\left(\Phi_q\right) \le \alpha_q FW\left(\Phi_q\right)' \qquad \forall \quad q = 1\ldots Q$$

$$0 < \delta_{p_N} \le 1 \qquad \forall \quad p_N = 1\ldots P_N$$

$$0 < \delta_{p_U} \le 1 \qquad \forall \quad p_U = 1\ldots P_U$$

$$\Theta_N = \left\{ \theta_{p_N} \middle| N\left(\mu'_{p_N}, \delta_{p_N}\sigma'_{p_N}\right) \right\} \qquad \forall \quad p_N = 1\ldots P_N$$

$$\Theta_U = \left\{ \theta_{p_U} \middle| U\left(\mu'_{p_U} - \delta_{p_U}\Delta\theta_{p_U}, \mu'_{p_U} + \delta_{p_U}\Delta\theta_{p_U}\right) \right\} \qquad \forall \quad p_U = 1\ldots P_U$$

$$\Delta\theta_{p_U} = \theta_{p_U}^{UB'} - \mu'_{p_U} = \mu'_{p_U} - \theta_{p_U}^{LB'} \qquad \forall \quad p_U = 1\ldots P_U$$

where the indices o, s, m, d and q are associated with the initial conditions, process stages, parameter scenarios, operating policy variables (z) and performance criteria (Φ). The time invariant policy variables, υ, the time dependent variables, ν, and the stage duration times, $t_f$, remain fixed at the values specified in the prior Risk Analysis. The measure of uncertainty in the performance criterion, Φ, is the width between the 5 and 95% fractiles, FW(Φ). The prime represents the original value before uncertainty reduction. The total uncertainty space, Θ, is the combined space of the normal and uniform spaces $\Theta_N$ and $\Theta_U$. The Matlab Sequential Quadratic Programming routine was used to solve these problems.

It is assumed that the original values of the distribution means (nominal parameter values) are maintained. If the stochastic problem contains decisions in linearly correlated inputs, it is assumed that a change in the spread of one of the correlated parameters gives an equivalent change in the others, while maintaining the same correlation structure.

The solutions of these problems can provide a quantitative idea of the required efforts in reducing specific parameter uncertainties compared to returns in performance uncertainties, which may be used to support data collection decisions.

This information, combined with that obtained from model validation (Step 8) and Sensitivity Analysis (Step 9), provides a useful breakdown of the information required to focus relative experiment planning and data collection efforts towards improving a specific process model within the sequence (by inferring the key uncertain phenomena associated with the identified process sub-sequence and parameter uncertainties), with respect to the possible relative benefits which may be obtained in doing so. The data driver feedback loop shows the position of experiment planning and data collection in Figure 2, though specific decisions regarding these procedures are not explicitly considered in this work.

## 5. Case study: A multiphase semi-batch reactor process

This case study is based on an exothermic multiphase reaction process and kinetic model investigated by Sano et al. (1998). This case study is a single unit operation but the methodology has been used on a sequence of unit operations (Johnson). The process is for the production of a pharmaceutical intermediate, formed from the amination of a bromopropyl compound. Sano et al developed a kinetic model based on reaction calorimetry data obtained under laboratory conditions in order to determine the optimum feasible and safe operating policy. There is considerable uncertainty in many of the experimental parameters and even in the assumptions underlying the model.

Solid particles of the active pharmaceutical ingredient (API) bromopropyl feed compound (A) reside in an organic solvent (methanol) inside the reaction vessel. A fixed volume of a 50 wt% aqueous dimethylamine reagent (B) is added to the vessel at a constant flowrate under continuous agitation. The solids gradually dissolve and react with the dimethylamine. A diagram of the process is shown in Figure 3. The exact physico-chemical phenomena for this process are not known. The reaction consists of a parallel-series reaction in which the dimethylamine reacts with the dissolved API feed to form the desired intermediate (C) which in turn reacts with the active feed (A) to form a dimeric byproduct (D) in parallel,

| | |
|---|---|
| $A + B \rightarrow C$ | Main reaction |
| $A + C \rightarrow D$ | Sub-reaction |

By-product D is known to be very difficult to remove in the downstream purification stages. Intrinsic first order reaction kinetics are assumed in the deterministic process model proposed by Sano et al. (1998) but this is a source of uncertainty. An initial rate limiting period due to the dissolution of solids B, was observed to be independent of solvent concentration and agitation speed within the range of conditions approved. A crude approximation of first order kinetics (with Arrhenius constant and activation energy) is assumed in the model for this dissolution controlled period. This period was observed to last until approximately 55% conversion of A for all the conditions considered, at which point the reaction appeared to be limited by the intrinsic reaction kinetics.

The kinetic model is combined with a standard semi-batch reactor model with constant volume addition (of reagent B). The model equations are given in the Appendix. Consideration of the cooling capacity of the reactor resulted in a limiting relationship between the operating policy variables of feed B addition time, $t_{add}$, and isothermal temperature, $T_{iso}$. For the purposes of this study, this relationship is well approximated with $T_{iso}$ as a quadratic function of $t_{add}$ since data regarding the energy balance is unavailable, where the nominal values of the constants $C_1$, $C_2$ and $C_3$ are 7.06, -43.50 and 352.67 respectively.

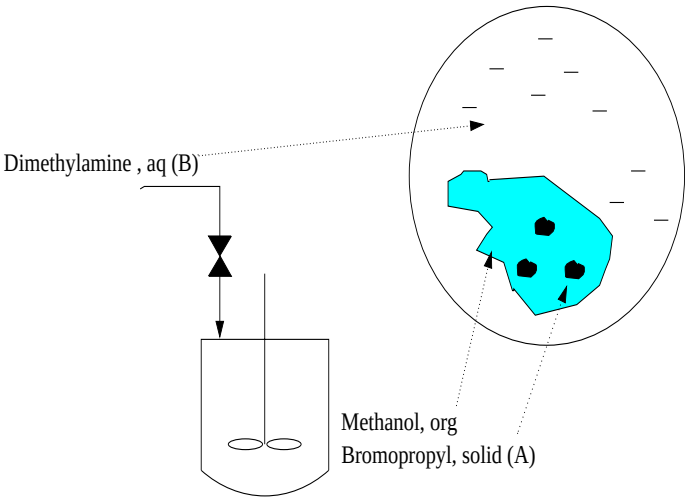



Figure 3. Multiphase batch reaction process

One of the process development objectives for which the model would be used is to help determine the best operating conditions for maximum product yield, $Y_C$. A reaction time, $t_f$, of less than 8 hours (terminated when the rate of conversion of A falls below 0.1%) and a final yield in the impurity, $Y_D$, of below 2% must be maintained. The model was optimised to obtain a nominal set of operating conditions which maximise the yield of C.

Of course uncertainty in the model parameters could have a large effect on any results predicted by the model. This may be of particular importance regarding the optimal operating policy determined subject to the desired limits on process performance. Hence the methodology presented in figure 2 was implemented on the case study.

Perturbation Analysis indicates 11 uncertain parameters which appear to have a non-negligible influence on yield of C ($Y_C$), yield of D ($Y_D$) and the final time ($t_f$): the kinetic rate law parameters ($Ea_{1,int}$, $A_{1,int}$, $Ea_{2,int}$, $A_{2,int}$, $Ea_{1,diss}$, $A_{2,diss}$), the conversion related transition point from dissolution controlled kinetics to intrinsically controlled kinetics, ($X_{A,diss}$), the molar ratio of active feed ($m_{A0}$) to reagent feed ($m_{B0}$) and the quadratic constants of the safety constraint ($C_1$, $C_2$ and $C_3$). The assumed uncertainties of these parameters are quantified in Table 1. Correlations are assumed between the activation energy (Ea) and the natural logarithm of the Arrhenius coefficient (A) parameters for each reaction rate constant and between the safety constraint constants.

A total of 490 scenarios were required to satisfy the convergence criterion of 0.5% error in the mean and variance parameters for both $Y_D$ and $t_f$. The key results of an Uncertainty Analysis under the nominal optimum isothermal operating conditions are shown in Table 2. Under uncertainty in the model parameters the process is predicted to perform particularly poorly regarding violation of the safety constraint for $T_{iso}$, with an expected probability of passing of only 0.281. The probability of passing the $Y_D$ constraint (at most 2%) is only 0.670 with an expected extent of violation of 0.245%. However, the corresponding values for $t_f$ appear to perform better (at most 8 hours).

Table 1. Uncertainty characterisation in the parameters of Case Study.

| | Normal distribution, N(μ, σ, $\hat{C}$) |
|---|---|
| $Ea_1$, $A_1$ | $N\left(\begin{matrix} Ea_1^* = 90.46\times10^4 & \sigma_{Ea_1^*} = 10\%Ea_{1s}^* \\ A_1^* = 4.68\times10^{15} & , \sigma_{A_1^*} = 10\%A_1^* \end{matrix}, \hat{C} = \begin{bmatrix} 1.00 & 0.99 \\ 0.99 & 1.00 \end{bmatrix}\right)$ |
| $Ea_2$, $A_2$ | $N\left(\begin{matrix} Ea_2^* = 65.97\times10^4 & \sigma_{Ea_2^*} = 10\%Ea_2^* \\ A_2^* = 1.00\times10^{10} & , \sigma_{A_2^*} = 10\%A_2^* \end{matrix}, \hat{C} = \begin{bmatrix} 1.00 & 0.99 \\ 0.99 & 1.00 \end{bmatrix}\right)$ |
| $Ea_{1,diss}$, $A_{1,diss}$ | $N\left(\begin{matrix} Ea_{1,diss}^* = 78.98\times10^4 & \sigma_{Ea_{1,diss}^*} = 10\%Ea_{1,diss}^* \\ A_{1,diss}^* = 1.76\times10^{13} & , \sigma_{A_{1,diss}^*} = 10\%A_{1,diss}^* \end{matrix} \hat{C} = \begin{bmatrix} 1.00 & 0.99 \\ 0.99 & 1.00 \end{bmatrix}\right)$ |
| $X_{A,diss}$ | N(0.55 ± 0.05) |
| $m_{A0,ratio}$ | N(0.357 ± 10% nominal) |
| $C_1$, $C_2$, $C_3$ | $N\left(\begin{matrix} C_1^* = 7.06 & \sigma_{C_1} = 1.15 \\ C_2^* = -43.50 & \sigma_{C_2} = 4.77 \\ C_3^* = 352.67 & \sigma_{C_3} = 4.53 \end{matrix}, \hat{C} = \begin{bmatrix} 1.00 & -0.99 & 0.97 \\ -0.99 & 1.00 & -0.99 \\ 0.97 & -0.99 & 1.00 \end{bmatrix}\right)$ |

Table 2. Uncertainty Analysis results under nominal optimum isothermal operating conditions

| | $Y_C$ (%) | $Y_D$ (%) | $t_f$ (hr) | safety constraint |
|---|---|---|---|---|
| Expected value | 96.34 | 1.79 | 6.33 | - |
| Expected extent of constraint violation | - | 0.245 | 0.118 | 0.440 |
| Probability of passing | - | 0.670 | 0.876 | 0.281 |

Sensitivity Analysis (step 9) shows which of the parameters are identified as key to the important output criteria. Correlated parameters are not included in the analysis since the presence of strongly correlated inputs invalidates the linear regression for the standardised regression coefficients. Approximate correlation ratios, Figure 4, show that the variance in the activation energy of the intrinsic parallel reaction, $Ea_2$ which is parameter number 2 in the figure (and the Arrhenius parameter, $A_2$, through correlation and the assumption of a linear joint confidence region), is the key uncertain parameter affecting the uncertainty in the prediction for both Yield D and Yield C. No single uncertain parameter is identified as being the main contributor to the uncertainty in the final time, $t_f$.

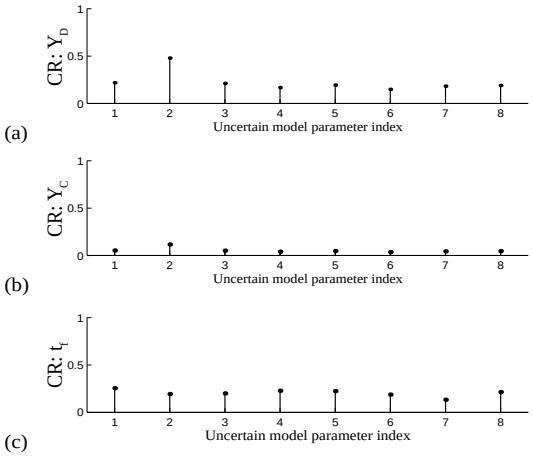



Figure 4. Sensitivity Analysis results under nominal optimum isothermal operating conditions

Further information quantifying the potential uncertainty reduction requirements to meet levels of reduction in the $Y_C$, $Y_D$ and $t_f$ criteria (step 10) is obtained from the solution of the following problem at different levels of desired output uncertainty reduction:

$$\max_{\delta_{p_N},\delta_{p_U}} \sum_{p_N=1}^{P_N} \delta_{p_N} + \sum_{p_U=1}^{P_U} \delta_{p_U}$$

subject to

deterministic process stage model equations and the following conditions on the 5% and 95%fractile requirements of the products ranges, constraints on δ to ensure the range is not exceeded, and the uncertainty description (θ)

$$FW_{5\%,95\%,Y_C} \le \alpha_{Y_C} FW'_{5\%,95\%,Y_C}$$

$$FW_{5\%,95\%,Y_D} \le \alpha_{Y_D} FW'_{5\%,95\%,Y_D}$$

$$FW_{5\%,95\%,t_f} \le \alpha_{t_f} FW'_{5\%,95\%,t_f}$$

$$0 < \delta_{p_N} \le 1,$$

$$0 < \delta_{p_U} \le 1$$

$$\Theta_{p_U} = \left\{ \theta_{p_U} \middle| U\left(\mu'_{p_U} - \delta_{p_U}\Delta p_U, \mu'_{p_U} + \delta_{p_U}\Delta p_U\right) \right\}$$

$$\Delta p_U = p_U^{UB'} - \mu'_{p_U} = \mu'_{p_U} - p_U^{LB'}$$

$$\Theta_{p_N} = \left\{ \theta_{p_N} \middle| N\left(\mu'_{p_N}, \delta_{p_N}\sigma'_{p_N}\right) \right\}$$

where $p_N = \left\{ Ea_{1,\text{int}}, Ea_{2,\text{int}}, Ea_{1,diss}, A_{1,\text{int}}, A_{2,\text{int}}, A_{1,diss}, C_1, C_2, C_3 \right\}$

$$p_U = \left\{ m_{A0,ratio}, X_{A,diss} \right\}$$

The stochastic optimisation in step 10 determines the minimum reduction in the uncertain parameters, $\theta_{p_N}$ and $\theta_{p_U}$ required in order to maintain stochastic inequality constraints for α% reductions in the widths of the predicted 5-95% fractile intervals for $Y_C$, $Y_D$ and $t_f$ from their original values. The parameters manipulated are the fractions of the original standard deviations of the normally distributed uncertain parameters, $Ea_{1,\text{int}}$, $Ea_{2,\text{int}}$, $Ea_{1,\text{diss}}$, $C_1$, $C_2$, $C_3$ and those of the

original bounding widths about the means of the uniformly distributed uncertain parameters, $m_{A0,ratio}$ and $X_{A,diss}$. Equivalent reductions in the uncertainty in the correlated parameters ($A_{1,int}$, $A_{2,int}$, $A_{1,diss}$) are assumed, to maintain the original correlation structures.

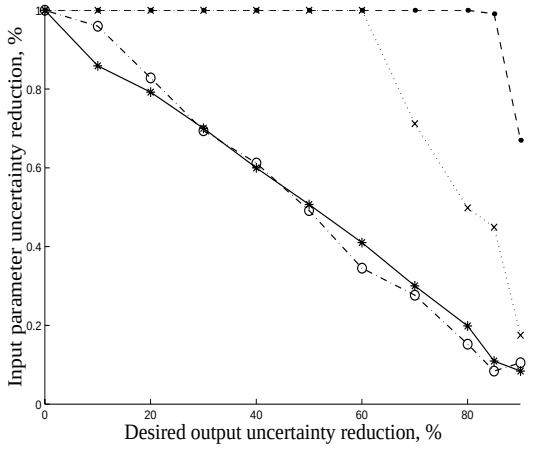



Figure 5. Model parameter uncertainty reductions meeting desired output criteria uncertainty reduction levels.
Key: *— = $\sigma_{Ea1,int}$, ×···· = $\sigma_{Ea1,diss}$, •--- = $\Delta m_{A0,ratio}$, o·-· = $\sigma_{Ea2,int}$.

The key results of these optimisation problems are shown in Figure 5. Reduction of the uncertainty in the intrinsic product and by-product activation energies (and the correlated Arrhenius constants) is shown to reduce the uncertainty in the output criteria to levels of around 60% of the original predicted uncertainties, at a constant rate. As uncertainty in the key input parameters is reduced in order to meet the desired levels of uncertainty in the output criteria, the contributions of the uncertainty in other input parameters become important and additionally need to be reduced. This is indicated in Figure 5 at levels of 60% and 90% output uncertainty reductions, where the respective optimal solutions state that reductions in the uncertainty in the dissolution activation energy (and the corresponding Arrhenius constant) and the API feed ratio become relatively more important than in $Ea_{1,int}$ and $Ea_{2,int}$. This has clear implications for where further more accurate experimental data should be obtained.

## 6. An interval based approach

The approach as implemented above relies on developing statistical information about the data items (in the sampling and repeated solution of the stochastic model), often from few data points, and the expensive machinery of stochastic optimisation.

In the case of process development shown above data is often obtained as a measurement with error or uncertainty bounds. These bounds give an important indication of the uncertainty of the measurement but it is only with in depth knowledge can someone know whether the degree uncertainty in the measurement is going to have a significant effect in process development. The measurements are often taken by development chemists who are not involved in development of the manufacturing process and so while they will have a feeling for the effects on the chemistry they will not necessarily know the effect on the manufacturing process.

There is again a role for systematically incorporating the uncertainty into the development process using a model based approach. If the model based approach presented in figure 2 is used but with intervals rather than stochastic distributions a systematic approach can be developed. A

weakness is that if only intervals are used the approach could be very conservative but it should be able to indicate which uncertainty in which measurements have the most effect and whether the uncertainty in the design can be significantly improved by better measurement. If data is now provided as a measurement with error bounds (intervals), optimization could also be achieved by application of interval global optimisation algorithms.

Two important distinctions are identified in formulating flowsheeting problems. In the equation-oriented formulation the flowsheet is treated as a set of mass/energy balance equations that are solved simultaneously. The alternative sequential modular approach views the flowsheet as interconnected black boxes. Both approaches have their advantages; however the modular approach has a particular advantage in that it matches more closely the natural structure of the flowsheet. Modular approaches are in general more popular in the chemical industry. Using modular flowsheets built from general models Byrne and Bogle (2000) showed how interval methods could be used in conjunction with this type of system. Modular flowsheets are constructed with generic unit modules that can provide the interval bounds, linear bounds, derivatives and derivative bounds using extended arithmetic types. Using interval analysis and automatic differentiation as the arithmetic types, lower bounding information is used in a branch and bound network.

The approach shown in Figure 2 could be modified to exploit this interval information using interval optimization techniques to solve the optimization problems. Step 1 requires obtaining intervals instead of distributions of the model parameters. Step two defines instead a deterministic system but with intervals for the uncertain parameters (such as the activation energies) and uncertain outputs (such as yield in the example above). The sampling procedure is no longer necessary since the optimization is done in terms of the interval bounds only. Step 5 remains as for the stochastic problem and step six involves obtaining the globally optimal solution for the deterministic problem using the real data points. In step 4 the models are used to obtain the interval bounds on the output variables and a sensitivity analysis can be performed to determine the key predicted output uncertainties and hence reduce the dimensionality of the subsequent optimization problem. Finally an interval optimization problem should be solved to determine the optimal reduction in input uncertainty that will keep the output uncertainties within their desired limits.

This approach has the advantage of requiring only data and error bounds and can use the interval optimisation software that is available. Error bounds can be conservative and this approach will help to indicate when it would be most appropriate to really try and improve the accuracy of measurements by more careful procedures or by obtaining more sophisticated measuring equipment.

## Conclusions

A systematic approach for incorporating uncertainty in process design has been presented. A stochastic optimisation problem is solved using distributions in the parameter uncertainties to determine where the key uncertainties in the data lie. This was applied to a multiphase batch reactor problem shown here and has also been applied to a pharmaceutical process involving 15 unit operations in sequence (Johnson). The methodology produced some clear recommendations about which measurements would best be improved to reduce the uncertainty in the output variables which are key for ensuring that the quality of the product is acceptable.

Since much data is often obtained from the laboratory with error bounds we have also discussed briefly how the problem could be cast as an interval optimisation problem which would determine where error bounds on particular data points were causing particular uncertainty in process development.

## References


1. Basu P.K., R.A. Mack and J.M. Vinson, Consider a new approach to pharmaceutical process development, 1999, *Chem. Eng. Prog*., Vol. 95, No. 8, 82-90.
2. Byrne R.P. and Bogle I.D.L. (2000) Global optimisation of modular process flowsheets, Ind Eng Chem Res 2000 39 4296-4301
3. Diwekar U.M. and E.S. Rubin, Stochastic modelling of chemical processes, 1991, *Computers Chem. Engng,* Vol. 15, No. 2, 105-114.
4. Donaldson J.R. and R.B. Schnabel, Computational experience with confidence regions and confidence intervals for nonlinear least squares, 1987, *Technometrics*, Vol. 29, No. 1, 67-82.
5. Johnson (2003) Integrated Design Under Uncertainty for Pharmaceutical Processes. PhD thesis, University of London.
7. Hangos K. and I. Cameron, Process modelling and model analysis, 2001, Academic Press, London.
8. Sano T., T. Sugaya T and M. Kasai, Process improvement in the production of a pharmaceutical intermediate using a reaction calorimeter for studies of the reaction kinetics of amination of a bromopropyl compound, 1998, *Organic process research and development*, Vol. 2, 169-174.


## Appendix

The deterministic model for the multiphase batch reactor

$$T_{iso,\max} \geq 7.06\left(t_{add}\right)^2 - 43.50\left(t_{add}\right) + 352.67$$

$$\frac{dm_A}{dt} = -k_1 \frac{m_A m_B}{V} - k_2 \frac{m_A m_C}{V}$$

$$\frac{dm_B}{dt} = \upsilon_{feed} \frac{m_{B0,feed}}{V_{B0,feed}} - k_1 \frac{m_A m_B}{V}$$

$$\frac{dm_C}{dt} = k_1 \frac{m_A m_B}{V} - k_2 \frac{m_A m_C}{V}$$

$$\frac{dm_D}{dt} = k_2 \frac{m_A m_C}{V}$$

$$k_{r,\text{int}} = A_{r,\text{int}} \exp\left(-\frac{Ea_{r,\text{int}}}{T_{iso}}\right) \qquad \text{for} \qquad r = 1,2$$

$$k_{1,diss} = A_{1,diss} \exp\left(-\frac{Ea_{1,diss}}{T_{iso}}\right)$$

$$k_{2,diss} = k_{2,\text{int}} \frac{k_{1,diss}}{k_{1,\text{int}}}$$

$$\beta_{diss} = \begin{cases} 1 & if \quad X_A \leq X_{diss} \\ 0 & if \quad X_A > X_{diss} \end{cases}$$

$$k_r = \beta_{diss} k_{r,diss} + \left(1 - \beta_{diss}\right) k_{r,\text{int}} \qquad \text{for} \qquad r = 1,2$$

$$\frac{dV}{dt} = \upsilon_{feed}\Big|_{t_0, t_{add}} = 0 \Big|_{t_{add}, t_f}$$

$$\upsilon_{feed} = \frac{V_{B0,feed}}{t_{add}}$$

$$T_{iso} \geq C_1\left(t_{add}\right)^2 - C_2\left(t_{add}\right) + C_3 \qquad \text{(safety constraint)}$$

$$X_A = \frac{m_{A0} - m_A}{m_{A0}}$$

$$\frac{dX_A}{dt} = \left( \frac{k_1 m_A m_B}{V m_{A0}} + \frac{k_2 m_A m_C}{V m_{A0}} \right) \leq 0.001 \quad hr^{-1}$$

$$T_{iso,\max} = C_1\left(t_{add}\right)^2 + C_2 t_{add} + C_3$$

Initial conditions (inside reactor)

$$m_{A0} = 1.075 \quad moles$$

$$m_{B0} = 0, \qquad m_{C0} = 0, \qquad m_{D0} = 0$$

$$V = 0.7 \quad dm^3$$

$$X_{A0} = 0$$

The subscripts diss and int denote dissolution and intrinsic kinetic controlled periods, and $k_{2,diss}$ is assumed to follow a similar temperature relationship as $k_{1,diss}$ relative to its intrinsic value.

# Solving Interval Constraints in Computer-Aided Design

Yan Wang
*NSF Center for e-Design,University of Pittsburgh 1048 Benedum Hall Pittsburgh, PA 15261*
*e-mail: ywang@engr.pitt.edu*

**Abstract**. Currently available CAD systems require geometric parameters to have fixed values. Valid range information on parameters cannot be properly represented and embedded in existing CAD data. Specifying fixed parameter values implicitly adds rigid constraints on the geometry, which have the potential to introduce conflicts during later design stages.
In this paper, a geometric modeling scheme based on nominal interval representation and analysis is presented to represent design uncertainty and inexactness. Parameters are represented by nominal intervals, which contain the information of nominal values, lower bounds, and upper bounds. Interval constraints represent inexactness at the early design stages, uncertainty in the detailed design, as well as the boundary information for design optimization.

To solve under-constrained and over-constrained interval problems, iteration-based equation solving methods are used. A generalized nonlinear constraint solving method based on linear enclosure is developed for fast convergence. Inequalities are transformed into equations and can be solved uniformly. Interval subdivision and constraint re-specification methods are developed for design refinement. Active and inactive constraints are differentiated in sensitivity analysis.

## 1. Introduction

During the process of design, various parameters are specified, which include geometric parameters (e.g. dimension, coordinate, and tolerance) and non-geometric ones (e.g. material characteristics, tooling speed, and expected life). Current CAD systems only allow geometric parameters to have fixed values, such as the position of a point in 3D space, the direction of a line, and the distance between two axes. Instead of simply assigning one real value to a parameter, there are some advantages to give an interval value to each parameter in a CAD model, which means that the parameter can take any valid value between the lower and upper bounds of the interval.

Fixed parameter values generate some problems. First, fixed-value constraints bring up conflicts easily at later design stages. Specifying determined parameter values implicitly adds rigid constraints on geometry. The rigid constraints reduce the freedom of geometric entities to the minimal level. These dominant constraints will be carried to other design stages and become the sources of conflicts. To resolve the conflicts, some parameter values have to be changed. This trial-and-error cycle will continue until no conflicts are found. If an interval instead of a fixed value is assigned to a parameter so that any real value within the interval is valid, the degrees of freedom of geometric entities are increased. As more constraints are imposed onto the designed object during the process of design, the freedom of geometric entities will be restricted gradually. The allowable intervals of parameter values are reduced by stages. There will be fewer chances for conflicts to occur, and several cycles of modification can be saved.

Second, the requirement of fixed parameter values makes the development of conceptual design tools difficult. At the conceptual design stage, actual values of parameters may not be known. Usually it is not important to specify fixed values of certain parameters at this stage yet. Current CAD systems require that parameter values be fully specified and fixed, thus they are not effective tools for conceptual design. It is quite challenging to develop a practically usable

Computer-Aided Conceptual Design tool based on the current scheme of fixed parameter values. Nevertheless, if a parameter is specified as a range, the problem of parameter partial integrity can be solved, i.e., it is not necessary to fix all values of parameters. This increases the flexibility of the geometric shape, and the inexactness of design is modeled.

Third, the specifications of valid parameter range are not captured by fixed-value data. Current design optimization process often occurs after parameters are specified at the detailed design stage, while the original intention of feasible ranges of parameters from upstream design activities is not transferable with the fixed-value scheme. Parameter bounds have to be added separately for optimization purpose. However, with the interval representation, the inherent range information is directly applicable for parameter optimization. Parameter intervals appropriately represent design intent of feasibility, thus integrating the sketching and optimization of design.

In real design situations, there are some uncertainty factors in CAD modeling. The dimensions and shape of the designed objects are computed and stored digitally in CAD systems. Representing an infinite number of real numbers by a finite number of bits requires approximation. Not all decimal numbers can be represented in binary format exactly. Rounding errors come from the approximation. Cancellation errors occur because of catastrophic and benign cancellation. The precision of numbers in a computer depends on the word size and floating-point representation. Variation exists among different systems with different architectures. Uncertainty also comes from the measurement and tolerance of human perception during the parameter specification. The real value of measurement is the ideal case that is hard to realize from the statistical point of view. In the interval geometry representation, a computer-generated value can be looked as a sample from the range of values, while the CAD data of a designed object is a sample from the population of models. Parameter intervals capture the uncertainty characteristics of design.

In this paper, a nominal interval constraint representation (NICR) scheme based on nominal interval values is described to represent design uncertainty and inexactness. It represents soft constraints, thus reducing the chance of conflicts during constraint imposition. It provides a generic numerical parameter scheme to represent inexactness at the early design stages, uncertainty in the detailed design, as well as the boundary information for design optimization.

## 2. Background

Methods of interval analysis have started being used in computer graphics, including rasterizing parametric surfaces (Mudur and Koparkar [1]), ray tracing of parametric surfaces (Toth [2]) and implicit surfaces (Kalra and Barr [3]), collision detection of polyhedral objects (Moore and Wilhelms [4]) and surfaces (Von Herzen et al. [5], Duff [6], Snyder et al. [7, 8]). In design and engineering applications, Bliek [9] studied the interval analysis to ensure the numerical reliability in design computation. Rao and Berke [10] used interval arithmetic in imprecise structural analysis. Rao and Cao [11] applied interval analysis in design optimization of mechanical systems. Muhanna and Mullen [12, 13] developed interval-based finite-element formulation methods for uncertainty in solid and structural mechanics. Interval arithmetic and analysis provide efficient and scalable methods to solve constraint systems. Related to interval representation, set-based modeling [14], probabilistic modeling, and fuzzy logic are also applied in engineering design.

In CAD applications, Sederberg and Farouki [15] used interval arithmetic in approximating Bezier curves. Maekawa and Patrikalakis [16, 17] used interval Bezier curves to solve shape interrogation problems. Hu *et al*. [18, 19] developed rounded-interval arithmetic (RIA) to ensure numerical robustness in Boolean operations and boundary evaluation. Further, Abrams *et al*. [20], Shen and Patrikalakis [21] applied RIA in interval non-uniform rational B-splines. Tuohy *et al*. [22] applied interval methods for interpolating measured data with B-spline curves and surfaces. Wallner *et al*. [23] used intervals to bound errors in geometric construction. Chen and Lou [24]

proposed methods to bound interval Bezier curve with lower degree interval Bezier curve. Lin *et al*. [25] investigated the boundary evaluation of interval Bezier curve. The above research concentrates on the improvement of model's robustness. Interval parameters embody rounding and cancellation errors during floating-point computation.

From a different perspective, the NICR presented here allows all numerical values of parameters including coordinates, dimensions, and other values to be nominal interval numbers. Thus, it allows incomplete and inexact of design specification especially at the conceptual design stage. *Soft* constraints compared to traditional fixed-value *rigid* constraints can be represented. Parameters have intrinsic bounds for design optimization. Design intent of parameter validity can be integrated into constraints. The probabilistic property of variables is captured in interval constraints, which provides a unified representation for different model applications, such as Monte Carlo simulation for finite element analysis, tolerance analysis and synthesis, and producibility analysis

## 3. Nominal interval constraints

In NICR, we define interval number X as X = [xL, xN, xU] which contains lower bound value xL, nominal value xN, and upper bound value xU. The nominal value is usually corresponding to the specified fixed value in current CAD systems.

The introduction of the nominal value into an interval is necessary for CAD modeling. The nominal value represents the actual user specification if the parameter is fixed. It allows current CAD modeling system to adopt interval parameters so that ICR can be integrated with current fixed-value schemes and visualization methods. Furthermore, the nominal value is allowed to change within the range when user specifies a different value. This allows more user interaction and captures more specification information. For example, a 2D point P([*1,2,3*],[*4,5,6*]) can be displayed at (*2,5*). When P is fixed, its coordinates are ([*2,2,2*],[*5,5,5*]). To simplify the notation, we can use a real number for a degenerated interval.

### 3.1 Basic Nominal Interval Definitions

An n dimensional real number space is denoted as Rn. An n dimensional interval number space is denoted as IRn. $X = [x_L, x_N, x_U] = \{x | x_L \le x \le x_U, x_L \le x_N \le x_U\}$, where xL $\in$ R, xN $\in$ R, xU $\in$ R, and X $\in$ IR.

Given that A = [aL, aN, aU], B = [bL, bN, bU], and $\wedge$ is logical and, we have the following relations:

- equivalence: $A = B \Leftrightarrow (a_L = b_L) \wedge (a_U = b_U)$.
- nominal equivalence: $A := B \Leftrightarrow (a_L = b_L) \wedge (a_N = b_N) \wedge (a_U = b_U)$.
- strictly greater than or equal to: $A \sim\!\ge B \Leftrightarrow a_L \ge b_U$.
- strictly greater than: $A \sim\!> B \Leftrightarrow a_L > b_U$.
- strictly less than or equal to: $A \sim\!\le B \Leftrightarrow a_U \le b_L$.
- strictly less than: $A \sim\!< B \Leftrightarrow a_U < b_L$.
- inclusion: $A \subseteq B \Leftrightarrow (a_U \le b_U) \wedge (a_L \ge b_L)$, $A \subset B \Leftrightarrow (a_U < b_U) \wedge (a_L > b_L)$.

The relations of intervals are illustrated in Figure 1. *0* = [*0,0,0*] is *zero interval*. Interval *A* is *positive* (*negative*), iff $A \sim\!> 0$ ($A \sim\!< 0$). If the nominal value of $A = [a_L, a_N, a_U]$ is not of concern, *A* can simply be denoted as $[a_L, a_U]$.

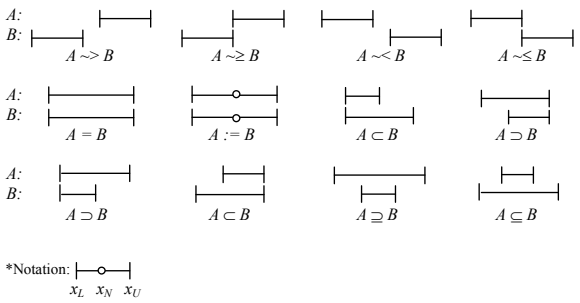



Figure 1: Relations between intervals

Interval $A = [a_L, a_N, a_U]$ is *empty*, denoted as $A = \varnothing$, iff $a_L > a_U$. $A$ is *invalid* when $a_N > a_U$, or $a_L > a_N$, or A is empty. The basic arithmetic and set operations are defined as:

- $A \cap B = \{x \mid x \in A \text{ and } x \in B, x \in \mathbf{R}\}$. If $A \cap B \neq \varnothing$, it can be derived by $A \cap B = [\max\{a_L, b_L\}, (\max\{a_L, b_L\} + \min\{a_U, b_U\})/2, \min\{a_U, b_U\}]$.
- $A \cup B = \{x \mid x \in A \text{ or } x \in B, x \in \mathbf{R}\}$. If $A \cap B \neq \varnothing$, it can be derived by $A \cup B = [\min\{a_L, b_L\}, (\min\{a_L, b_L\} + \max\{a_U, b_U\})/2, \max\{a_U, b_U\}]$.
- $A \setminus B = \{x \mid x \in A \text{ and } x \notin B, x \in \mathbf{R}\}$.
- $A + B = [a_L + b_L, a_N + b_N, a_U + b_U]$.
- $A - B = [a_L - b_U, a_N - b_N, a_U - b_L]$.
- $A \cdot B = [\min\{a_L b_L, a_L b_U, a_U b_L, a_U b_U\}, a_N b_N, \max\{a_L b_L, a_L b_U, a_U b_L, a_U b_U\}]$.
- $\frac{1}{B} = \left\{ \frac{1}{y} \middle| y \in B, 0 \notin B \right\}$.
- $$\frac{A}{B} = \begin{cases} A \cdot \frac{1}{B} & (0 \notin B) \\ [-\infty, 0, +\infty] & (B = 0) \\ [\frac{a_U}{b_L}, \frac{a_U}{b_L}, +\infty] & (a_U \le 0, b_L < 0, b_U = 0) \\ [-\infty, \frac{a_U}{b_U}, \frac{a_U}{b_U}] \cup [\frac{a_U}{b_L}, \frac{a_U}{b_L}, +\infty] & (a_U \le 0, b_L < 0, b_U > 0) \\ [-\infty, \frac{a_U}{b_U}, \frac{a_U}{b_U}] & (a_U \le 0, b_L = 0, b_U > 0) \\ [-\infty, 0, +\infty] & (a_L < 0, a_U > 0, b_L \le 0, b_U \ge 0) \\ [-\infty, \frac{a_L}{b_L}, \frac{a_L}{b_L}] & (a_L \ge 0, b_L < 0, b_U = 0) \\ [-\infty, \frac{a_L}{b_L}, \frac{a_L}{b_L}] \cup [\frac{a_L}{b_U}, \frac{a_L}{b_U}, +\infty] & (a_L \ge 0, b_L < 0, b_U > 0) \\ [\frac{a_L}{b_U}, \frac{a_L}{b_U}, +\infty] & (a_L \ge 0, b_L = 0, b_U > 0) \end{cases}.$$

The width of an interval is a real number, denoted as $\text{wid}(A) = a_U - a_L$. $\text{wid}(\varnothing) = 0$. Some other notations are $\text{ub}(A) = a_U$, $\text{lb}(A) = a_L$, and $\text{nom}(A) = a_N$.

### 3.2 Sampling Relation between Real Number and Interval Number

The intervals capture the uncertainty of design. A real number $x$ is a *sample* of interval $X$, iff $x \in X$. The value of a parameter, which is generated by computer or selected by human designer, is a sample of the corresponding set of values within the interval. Therefore, one CAD interval model is allowed to generate different shapes because of parameter intervals. Implicitly, a CAD interval model defines a set of geometric shapes that automatically accommodate geometry variation.

Some *strict* relations exist among intervals, which are related to real number samples. $X\Re Y \Leftrightarrow \forall x \in X, \forall y \in Y, x\Re y$ . $X\Re Y$ denotes that $X$ has a *strict* relation $\Re$ with $Y$ ($X \in \boldsymbol{IR}$, $Y \in \boldsymbol{IR}$).

- (*strict equivalence*): $A \sim= B \Leftrightarrow \forall x \in A, \forall y \in B, x = y$ .
- (*strictly greater than or equal to*): $A \sim\geq B \Leftrightarrow \forall x \in A, \forall y \in B, x \geq y$ .
- (*strictly greater than*): $A \sim> B \Leftrightarrow \forall x \in A, \forall y \in B, x > y$ .
- (*strictly less than or equal to*): $A \sim\leq B \Leftrightarrow \forall x \in A, \forall y \in B, x \leq y$ .
- (*strictly less than*): $A \sim< B \Leftrightarrow \forall x \in A, \forall y \in B, x < y$ .

Besides strict relations, some *global* relations exist in interval arithmetic evaluation and problem solving. $X\Im Y \Leftrightarrow \forall x \in X, \exists y \in Y, x\Im y$ . $X\Im Y$ denotes that $X$ has a *global* relation $\Im$ with $Y$ ($X \in \boldsymbol{IR}$, $Y \in \boldsymbol{IR}$).

Global relations ensure the *feasibility* of interval arithmetic operations and solutions. The goal of solving interval problems is to find a region that includes all feasible solutions. The corresponding process is to eliminate certainly infeasible points from a given region so as to make it as compact as possible. The global relations make global solution and optimization of interval analysis possible. For example, the four basic arithmetic operations of intervals follow the rule of global relation and generate the global solution with a compact bound. This is the default relation in interval analysis. Strict inequalities are special cases of global inequalities. Function evaluation and problem solving in interval analysis are normally based on global relations.

- (*global equivalence*): $A = B \Leftrightarrow \forall x \in A, \exists y \in B, x = y$ .
- (*greater than or equal to*): $A \geq B \Leftrightarrow a_L \geq b_L$ . Equivalently, $A \geq B \Leftrightarrow \forall x \in A, \exists y \in B, x \geq y$ .
- (*greater than*): $A > B \Leftrightarrow a_L > b_L$ . Equivalently, $A > B \Leftrightarrow \forall x \in A, \exists y \in B, x > y$ .
- (*less than or equal to*): $A \leq B \Leftrightarrow a_U \leq b_U$ . Equivalently, $A \leq B \Leftrightarrow \forall x \in A, \exists y \in B, x \leq y$ .
- (*less than*): $A < B \Leftrightarrow a_U < b_U$ . Equivalently, $A < B \Leftrightarrow \forall x \in A, \exists y \in B, x < y$ .

In a multidimensional interval space, an interval vector can be defined in $\boldsymbol{IR}^n$ with each component as an interval value, and an interval matrix is defined in $\boldsymbol{IR}^m \times \boldsymbol{IR}^n$ with each element as an interval value. Corresponding to a real function $f$: $\boldsymbol{R}^n \rightarrow \boldsymbol{R}^m$, if $f_{set}(\mathbf{X})$ denotes $\{f(\boldsymbol{x}) \mid \boldsymbol{x} = (x_1, x_2, \ldots, x_n), x_i \in X_i\ (i = 1, ..., n), \mathbf{X} = (X_1, X_2 ..., X_n), \mathbf{X} \in \boldsymbol{IR}^n\}$, the inclusion function for $f$ at $\mathbf{X}$ $F$: $\boldsymbol{IR}^n \rightarrow \boldsymbol{IR}^m$ if $f_{set}(\mathbf{X}) \subseteq F(\mathbf{X})$. A natural inclusion function $f(\mathbf{X})$ for $f(\boldsymbol{x})$ is obtained by replacing each occurrence of the variable $x_i$ by interval variable $X_i$. It is based on the inclusion isotonicity of the interval operations [26] and the property of pre-declared inclusions [27]. Generally, the natural inclusion function $f(\mathbf{X})$ for $f(\boldsymbol{x})$ is not tight enough because of dependency between variables and wrapping effect [28].

Interval vectors with same dimensions can be ranked and sorted.

- Interval vector **A** and **B** are in a non-decreasing order,
  $\mathbf{A} \prec \mathbf{B}$, where $\mathbf{A} = (A_1, A_2, \cdots, A_n), \mathbf{B} = (B_1, B_2, \cdots, B_n)$, if $A_n \leq B_n$, and
  $\neg(A_i < B_i) \rightarrow (A_{i-1} \leq B_{i-1})$ recursively apply, starting from $i = n$.
- Interval vector **A** and **B** are in a non-increasing order,
  $\mathbf{A} \succ \mathbf{B}$, where $\mathbf{A} = (A_1, A_2, \cdots, A_n), \mathbf{B} = (B_1, B_2, \cdots, B_n)$, if $A_n \geq B_n$, and
  $\neg(A_i > B_i) \rightarrow (A_{i-1} \geq B_{i-1})$ recursively apply, starting from $i = n$.
- $\text{maxwid}(\mathbf{A}) = \max_i(\text{wid}(A_i))$, where $\mathbf{A} = (A_1, A_2, \cdots, A_n)$.
- $\text{minwid}(\mathbf{A}) = \min_i(\text{wid}(A_i))$, where $\mathbf{A} = (A_1, A_2, \cdots, A_n)$.

### 3.3 Geometry Description

With the inherent capability of modeling variation, NICR has some special properties that make it different from current geometric modeling schemes. It models uncertainty and inexactness in the process of design. As the available ranges of parameters are narrowed down gradually, uncertainty is ruled out and decisions can be made throughout the design process until design is finalized. Changing current constraints or adding extra constraints would lead to different geometries. As illustrated in Figure 2, the shape of a 2D rectangular object may vary based on coordinates of four corner points within their allowable intervals.

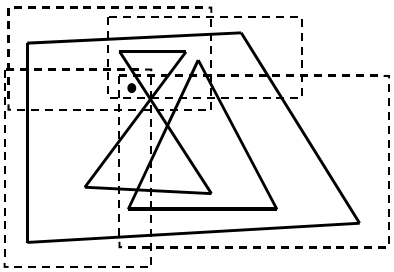


Figure 2: Constraint-driven geometry in interval modeling

Current parametric modeling scheme has strict requirements on the number of constraints. Only well-constrained geometry can be properly solved. The concept of under-constrained geometry in traditional parametric or variational design is not critical in NICR. *Soft* constraint is applied to geometry implicitly at every step of value specifications. The effect of adding more constraints is to reduce the allowable region of geometric entities systematically so that the final geometry can be fixed.

Over-constrained situation is also allowed in NICR. As illustrated in Figure 3, if the geometric constraints in a bracket design are specified as: the position of $P_0$; distances between $P_0$ and $P_1$, $P_1$ and $P_2$, $P_2$ and $P_3$, and $P_3$ and $P_0$; $L_0$ is perpendicular to $L_1$ as well as to $L_3$; and $L_0$ is horizontal, current CAD systems will complain that this geometry is over-constrained,

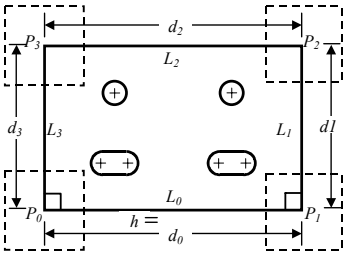



Figure 3: An example of over-constrained geometry in bracket design

In NICR, only those constraints that cause no feasible regions generate conflicts. Intervals loosen the current requirement on the number of constraints and give a different view of specifying parameters. Some of the previous over-constrained problems will no longer be over-constrained.

## 4. Solving interval constraints

To incorporate interval geometric modeling methodology into current CAD systems, several fundamental issues related to geometric computation should be addressed. These include linear and nonlinear equation representation and solution, which are essential for transformation operation, surface intersection, and constraint solving, etc. The process of solving systems of equations or inequalities is also called contraction. It starts with initial values of intervals, which are rough estimates of variable values. Subintervals that do not contain solutions are then eliminated, and intervals are "contracted". This process proceeds iteratively until there is no further improvement. Interval operations involve more steps and procedures than real arithmetic operations. Time and space efficient algorithms for solving interval constraint are needed.

### 4.1 Interval Linear Equations

Commonly used numerical methods for solving real-value linear equations can be extended to solve interval-value linear equations, such as Gaussian elimination and triangular factorization. But matrix-based methods only solve well-constrained problems. In contrast, iteration-based methods such as Jacobi iteration and Gauss-Seidel iteration have no requirement on the number of constraints. An algorithm for solving linear equations with nominal intervals is presented here, which is an extension of the Gauss-Seidel method, as listed in Figure 4. Different from methods of Alefeld and Herzberger [29], and Hansen and Sengupta [30], under-constrained and over-constrained linear systems are the major considerations here.

To solve

$$\sum_{j=1}^{n} A_{ij} X_j = Y_i \quad i = 1,2,...m \tag{1}$$

where $X_1$, $X_2$, …, $X_n$ are interval variables, $A_{ij}$ is interval constant for each $i$ and $j$, and $Y_1$, $Y_2$, …, $Y_m$ are interval constants. If an empty interval is derived during the process, there is no solution within the given initial intervals.

```
INPUT:  Interval matrix A
        Interval vector Y
OUTPUT: Interval vector X

Interval V
int i, j, k
REPEAT until stop criterion is met
    FOR each 1 <= i <= m
        FOR each 1 <= j <= n
            IF Aij=0
                continue next j iteration
            ENDIF
            V = 0
            FOR each 1<=k<j
                V = V+Aik*Xk
            ENDFOR
            FOR each j+1<=k<=n
                V = V+Aik*Xk
            ENDFOR
            V = (Yi - V)/Aij
            Xj = Xj ∩ V
        ENDFOR
    ENDFOR
```

Figure 4: Algorithm of extended Gauss-Seidel method for solving linear equations (1)

### 4.2 Interval Nonlinear Equations

Nonlinear equation systems can be solved by the fix-point method, forward-backward propagation, Newton's method, and Krawczyk method, etc. Given the requirement that a constraint solving system should be flexible on the number of constraints yet with fast convergence, a linear enclosure method is presented here. This algorithm is more general than Kolev's method [31, 32]. Kolev's method only considers the degenerated case where the right-hand sides of constants are all *0*s. The evaluation based on linear enclosure has sharper bounds than the one based on the interval Newton's method if the widths of intervals are non-trivial. Methods using coefficient matrix inverse operation are not applicable for under-constrained and over-constrained problems. Let us consider the interval nonlinear equation system

$$F_i(\mathbf{X}) = C_i \quad i = 1,2,...l\,, \tag{2}$$

where $\mathbf{X}$ is the interval variable vector $[X_1, X_2, \ldots, X_n]^{\mathrm{T}}$ and $C_i$ is a constant interval. The following steps are needed to solve the system:

*STEP 1*: Transform each equation of (2) to separable form to eliminate dependency among variables;

*STEP 2*: Find the linear enclosure of each of the univariate nonlinear functions and form a linear equation system;

*STEP 3*: Solve the linear system by the algorithm of Section 4;

*STEP 4*: If stopping criterion is met, stop. Otherwise, repeat from STEP 2 to STEP 4.

**STEP 1:**

According to Yamamura's algorithm [33], functions that are composed of four basic arithmetic operations (+, –, ×, /), unary operations (sin, exp, log, sqrt, etc.), and the power operation (^) can be transformed into the separable form by introducing necessary functions. For example, $f = f_1 \times f_2$ can be transformed to $f = (y^2 - f_1^2 - f_2^2)/2$ and $y = f_1 + f_2$; $f = f_1 / f_2$ can be transformed to $f = (y^2 - f_1^2 - 1/f_2^2)/2$ and $y = f_1 + 1/f_2$; and $f = (f_1)^{f_2}$ can be transformed to $f = exp(y_1)$, $y_1 = (y_2^2 - (log(f_1))^2 - f_2^2)/2$, and $y_2 = log(f_1) + f_2$. In geometric modeling, most of constraints/functions can be transformed into the separable form.

Thus equations (2) can be transformed to

$$\sum_{j=0}^{n} f_{ij}(X_j) = D_i \quad i = 1,2,...,m\,, \tag{3}$$

where $X_1, X_2, \ldots, X_n$ are interval variables and $D_1, D_2, \ldots, D_m$ are interval constants.

**STEP 2:**
Linear enclosure of $f_{ij}(x_j)$ is found within the initial interval of $X_j^{(0)}$ for each $i$ and $j$ as follows. Let $X_j^{(0)} = [x_L^j, x_N^j, x_U^j]$, we can have

$$f_{ij}^S = f_{ij}\left(x_L^j\right), \text{ and} \tag{4}$$

$$f_{ij}^T = f_{ij}\left(x_U^j\right). \tag{5}$$

Let

$$a_{ij} = \frac{f_{ij}^T - f_{ij}^S}{x_U^j - x_L^j}. \tag{6}$$

The *linear enclosure* of $f_{ij}(x_j)$ can be defined as

$$E_{ij}(x) = B_{ij} + a_{ij}x \quad for\ x \in X_j^{(0)}, \tag{7}$$

such that

$$f_{ij}(x) \in E_{ij}(x) \quad for\ \forall x \in X_j^{(0)}, \tag{8}$$

as illustrated in Figure 5.

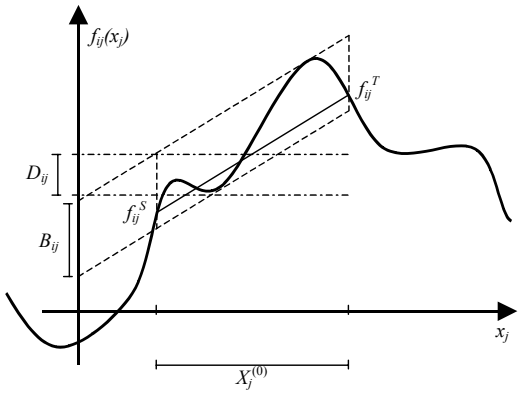



Figure 5: Linear enclosure of nonlinear interval function

To find a $B_{ij}$ with the minimum width with the given $a_{ij}$, derivation of $f_{ij}(x)$ is used if $f_{ij}(x)$ is continuous and differentiable within interval $X_j^{(0)}$. The problem is reduced to solve real value nonlinear equation and find out solutions of

$$f'_{ij}(x) = a_{ij} \quad for\ x \in X_j^{(0)}. \tag{9}$$

Given that $f_{ij}(x)$ is continuous and differentiable for most geometric relations, equations (9) have at least one solution. The Secant method can be used to solve the equation efficiently. Having been transformed to the separable form, $f'_{ij}(x)$ is a univariate polynomial function or a function with unary operations for most geometric constraints. For polynomial functions, roots can be isolated within disjointed intervals individually based on Descartes' rule of signs before equations are solved. Descartes' bound gives the upper bound of the number of positive roots of a polynomial. Once polynomial functions are solved, solutions to unary functions such as *sin* and *cos* can be easily found.

Let $P(x)$ be a polynomial with real coefficients, the following transformations are defined:

- (*Reverse transformation*): $R[P(x)] = x^n P(1/x)$ where $n$ is the degree of $P$.
- (*Translation transformation*): $T_t[P(x)] = P(x+t)$ for $t \in \boldsymbol{R}$.

- (*Homothetic transformation*): $H_c[P(x)] = P(cx)$ for $c \in \boldsymbol{R}$.

Based on the algorithm of Collins *et al*. [34, 35], $P_{ij}(x)$ for $x \in X_j^{(0)}$ is transformed to $P_{ij}^{0}(x)$ for $x \in [0, 1]$ by $P_{ij}^{0}(x) = H_{b\text{-}a}[T_a[P_{ij}(x)]]$ where $a$ is the lower bound of $X_j^0$ while $b$ is the upper bound of $X_j^0$. The roots of $P_{ij}(x)$ for $x \in X_j^0$ have one-to-one correspondence with the roots of $P_{ij}^{0}(x)$ for $x \in [0, 1]$. A list of root intervals or exact roots can be obtained by calling *RootIsolation*($P_{ij}^{0}$, 0, 0) listed in Figure 6. For each root interval or exact root with information of (*depth, index*) in the list, there is an corresponding $x \in [\frac{(b-a)index}{2^{depth}} + a, \frac{(b-a)(index+1)}{2^{depth}} + a]$ for root intervals or $x = \frac{(b-a)index}{2^{depth}} + a$ for exact roots such that $P_{ij}(x)=0$.

```
INPUT:  Polynomial P with n degree
        int depth
        int index
OUTPUT: RootIntervalList

IF P(0) = 0
    RootIntervalList.addExactRoot(depth, index)
ENDIF
IF P(1) = 0
    RootIntervalList.addExactRoot(depth, index+1)
ENDIF
Polynomial Q = T1[R(P)]
IF DecartesBound(Q) = 1
    RootIntervalList.addRootInterval(depth, index)
ELSEIF DecartesBound(Q) >= 2
    Polynomial P1 = 2^n H1/2[P]
    RootIsolation(P1, depth+1, 2*index)
    Polynomial P2 = T1[P1]
    RootIsolation(P2, depth+1, 2*index+1)
ENDIF
```

Figure 6: `RootIsolation` procedure based on Descartes' rule of signs

Thus, interval $X_j^{(0)}$ can be subdivided into small intervals containing an individual root. Let $g(x) = f'_{ij}(x) - a_{ij}$. Solutions to (9) within interval $X_j^{(0)}$ can be found by (10) iteratively.

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{g(x_n) - g(x_{n-1})} g(x_n) \quad n = 1,2,3,... \tag{10}$$

Suppose $x_{jp}$ ($p$=1, 2, …, $P$) is the $p^{\text{th}}$ solution of equation (9), and $x_{j0} = x_L^j$. Let $B_{ij} = [b_L^{ij}, b_N^{ij}, b_U^{ij}]$, where

$$b_U^{ij} = \max_p \{f_{ij}(x_{jp}) - a_{ij}x_{jp}, p = 0,1,2,..., P\}, \tag{11a}$$

$$b_N^{ij} = f_{ij}(x_{j0}) - a_{ij}x_{j0}, \tag{11b}$$

$$b_L^{ij} = \min_p \{f_{ij}(x_{jp}) - a_{ij}x_{jp}, p = 0,1,2,..., P\}. \tag{11c}$$

From equation (8), we have

$$f_{ij}(X_j) \subseteq E_{ij}(X_j) \quad for\ i = 1,2,...,m, \tag{12}$$

thus,

$$\sum_{j=1}^{n} f_{ij}(X_j) \subseteq \sum_{j=1}^{n} E_{ij}(X_j) = \sum_{j=1}^{n} (B_{ij} + a_{ij}X_j) \quad for\ i = 1,2,...,m. \tag{13}$$

**STEP 3:**
Solving (3) thus is reduced to solving linear equations (14) iteratively.

$$\sum_{j=1}^{n}\left(B_{ij}+a_{ij}X_j\right)=D_i \quad for\ i=1,2,...,m. \tag{14}$$

This linear system can be solved using the algorithm described in Section 4. Because the coefficient $a_{ij}$'s are degenerated intervals, only one iteration is needed to solve the linear equations. Suppose $Y_j$ is the $j^{th}$ variable solution of (14) in the $k^{th}$ iteration. By formula (15), the initial value of $X_j$ in the $(k+1)^{th}$ iteration is calculated. If an empty interval is derived, the original system has no solution within the given initial intervals ($X_1^{(0)}$, $X_2^{(0)}$, ..., $X_n^{(0)}$).

$$X_j^{(k+1)}=X_j^{(k)}\cap Y_j \quad for\ j=1,2,...,n. \tag{15}$$

**STEP 4:**
When the stopping criterion, such as the width of intervals has no further improvement (16a) or the intervals are sharp enough (16b), is met, the iteration is stopped. Otherwise, go back to (3) to find out the new linear enclosures within the updated intervals and repeat the procedure starting from STEP 2.

$$\left|\sum_{j=1}^{n}\mathrm{wid}(X_j^{(k+1)})-\sum_{j=1}^{n}\mathrm{wid}(X_j^{(k)})\right|<\varepsilon_1 \quad \text{for iteration } k. \tag{16a}$$

$$\left|\sum_{j=1}^{n}\mathrm{wid}(X_j^{(k)})\right|<\varepsilon_2 \quad \text{for iteration } k. \tag{16b}$$

### 4.2 Interval Inequalities

Inequalities can be solved by the methods for equations. Consider a set of linear or nonlinear inequalities

$$F_i(\mathbf{X})\le C_i \quad i=1,2,...l, \tag{17}$$

where **X** is the interval variable vector and $C_i$ is a constant interval, inequalities are transformed into equations

$$F_i(\mathbf{X})+S_i=C_i \quad i=1,2,...l, \tag{18}$$

where $S_i$ is a slack variable with initial value of $[0,0,+\infty]$. Similarly,

$$F_i(\mathbf{X})\ge C_i \quad i=1,2,...l, \tag{19}$$

can be transformed into

$$F_i(\mathbf{X})+S_i=C_i \quad i=1,2,...l, \tag{20}$$

where $S_i$ is a slack variable with initial value of $[-\infty,0,0]$. Inequalities can be easily integrated into systems of equalities, which is another property of interval constraint representation.

## 5. Design refinement

One important aspect related to interval representation of allowance is the over estimation of allowance. An interval vector simply encloses the allowable region by a hyper cube, which usually includes some infeasible region. During the function evaluation, inclusion functions are likely to give a set that is larger than the actual solution set. Design refinement is needed to generate more delicate design if desirable details have not been achieved yet. There are two ways to refine design: interval subdivision and constraint re-specification. Interval subdivision is to divide existing interval regions into unions of subintervals to achieve the refined views of current design. Constraint re-specification is to modify some of constraints or to add extra valid constraints to contract feasible regions.

### 5.1 Interval subdivision

Interval subdivision (also called *subpaving*) substitutes an interval vector with multiple interval vectors such that the corresponding real space region is subdivided into multiple smaller regions to cover the actual solution set more compactly. As shown in Figure 7, the interval vector **X** can be bisected recursively and subintervals are tested individually if they belong to the actual solution set. The actual solution set is approximated by the union of subinterval regions.

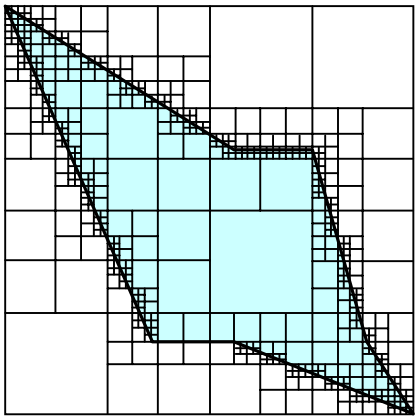


Figure 7: Two-dimensional interval vector subdivision

To represent subdivision of intervals concisely, a *power interval* can be used. An *n*-dimensional power interval with degrees of *m*, denoted as $\mathbf{P}^{(m,\ n)}$, is an ordered list of *m* non-overlapped interval vectors of *n*-dimensional, i.e., $\mathbf{P}^{(m,\ n)} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m]$, where $\mathbf{X}_i \in \boldsymbol{IR}^n$ ($i = 1, \ldots, m$), $\text{minwid}(\mathbf{X}_i \cap \mathbf{X}_j) = 0$ ($i \neq j$), and $\mathbf{X}_i \prec \mathbf{X}_{i+1}$ ($i = 1, \ldots, m-1$).

Consider a design problem $\boldsymbol{f}(\mathbf{X}) = \mathbf{Y}$. The target is to find the actual solution set $\mathbf{S} \subseteq \mathbf{X}$ with the minimal size such that $\boldsymbol{f}(\mathbf{S}) = \mathbf{Y}$. Interval arithmetic only gives a valid solution **D** with $\boldsymbol{f}(\mathbf{D}) \supseteq \mathbf{Y}$. If the valid solution is represented by power intervals, refinement can be looked as degree elevation of power intervals. If the original solution to a problem is found as an *n*-dimensional vector $\mathbf{X} = [X_1, X_2, \ldots, X_n]$, the corresponding power interval is $\mathbf{P}_{(0)}^{(1,n)} = [\mathbf{X}]$. One elevation operation will bisect **X**, with each interval vector being deleted and new subintervals inserted. Feasibility of each new subinterval then can be tested. The procedure of subdivision is shown in Figure 8.

```
INPUT:  Power Interval P^(m,n)
        Interval vector Y
        Mapping function f
OUTPUT: Power Interval P^(k,n)

IF stop criterion is met
    Return P^(m,n)
ELSE
    j = m*n
    Q^(j,n) = Bisect(P^(m,n))
    FOR 1 <= i <= m*n
        IF f(Q^(j,n)(i)) ⊄ Y
            Delete(Q^(j,n)(i))
        ENDIF
    ENDFOR
    Subdivide(Q^(j,n),Y,f)
ENDIF
```

Figure 8: `Subdivide` procedure for power interval elevation

### 5.2 Constraint Re-specification

Another way to contract a solution is to modify or add valid constraints to narrow down feasible regions. Feasibility and effectiveness should be considered simultaneously. Constraint modification depends on sensitivity analysis, while adding constraints is largely dependent on user's specification. One basic question is how to differentiate *active* and *inactive* constraints. Active constraints scope the actual range of solution while inactive constraints have certain level of slackness. At the beginning of interval computation, all constraints are active if a sufficiently large initial region is given. As the iteration proceeds, some constraints turn to be inactive. The decision of which constraints to be modified is based on the selection of active constraints.

**Lemma:** For a constraint set $\boldsymbol{p} = \{\boldsymbol{f}(\mathbf{X}) = \mathbf{Y}$ and $\boldsymbol{g}(\mathbf{X}) = \mathbf{Z}\}$, the subset $\boldsymbol{f}(\mathbf{X}) = \mathbf{Y}$ with respect to a solution $\mathbf{D} \subset \mathbf{X}$ is inactive if $\boldsymbol{f}(\mathbf{D}) \subset \mathbf{Y}$ and $\boldsymbol{g}(\mathbf{D}) \supseteq \mathbf{Z}$.

*Proof:*

Suppose $\mathbf{S}_1$ and $\mathbf{S}_2$ are actual solution sets of $\boldsymbol{f}$ and $\boldsymbol{g}$ respectively, and $\mathbf{S}$ is the actual solution set of $\boldsymbol{p}$. Given that $\boldsymbol{f}(\mathbf{S}_1) = \mathbf{Y}$ and $\boldsymbol{f}(\mathbf{D}) \subset \mathbf{Y}$, because of the property of inclusion monotonic, $\mathbf{S}_1 \supset \mathbf{D}$. Similarly, $\mathbf{D} \supseteq \mathbf{S}_2$. Thus, $\mathbf{S}_1 \supset \mathbf{S}_2$. □

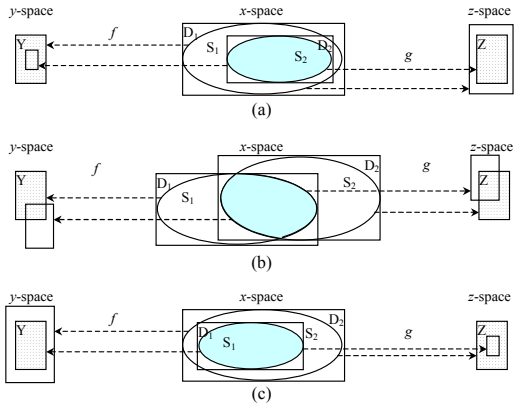



Figure 9: Relations of two constraint subsets

As illustrated by Figure 9, subset $\boldsymbol{f}$ is inactive and $\boldsymbol{g}$ is active in case (a); both $\boldsymbol{f}$ and $\boldsymbol{g}$ are active in case (b); and $\boldsymbol{f}$ is active and $\boldsymbol{g}$ is inactive in case (c).

## 6. A numerical example

The NICR kernel is implemented in C++ with an object-oriented programming style. The kernel includes the fundamental structure and arithmetic operations of the nominal intervals. It also includes the implementation of the algorithms described in previous sections for solving linear and nonlinear constraint systems as well as design refinement. The NICR kernel is integrated and tested in a geometric modeling system, which is based on ACIS® kernel.

As a demonstration, the design of the bracket in Figure 3 is used as a numerical example. The designer specifies the nominal value, lower bound, and upper bound of each coordinate and parameter. Geometric constraints are assigned to generate the outline of the bracket, which is over-constrained in the sense of the traditional parametric modeling. The interval geometric modeler then calculates the ranges of geometric points based on the algorithms of solving interval linear and nonlinear equations.

Figure 10 lists the constraint equations in Figure 3 (b), which are transformed to separable form. Based on the algorithm in Section 0, this over-constrained nonlinear equation system is solved. The numerical results are listed in Table 1.

| | |
|---|---|
| $x_0 = a_0$ | $y_2 - y_3 + v_3 = 0$ |
| $y_0 = b_0$ | $u_4^2 + v_4^2 = d_3^2$ |
| $y_0 - y_1 = 0$ | $-x_0 + x_3 + u_4 = 0$ |
| $u_1^2 + v_1^2 = d_0^2$ | $-y_0 + y_3 + v_4 = 0$ |
| $x_0 - x_1 + u_1 = 0$ | $u_1^2/2 + v_1^2/2 + u_4^2/2 + v_4^2/2 - w_1^2/2 - w_2^2/2 = o_1$ |
| $y_0 - y_1 + v_1 = 0$ | $-u_1 - u_4 + w_1 = 0$ |
| $u_2^2 + v_2^2 = d_1^2$ | $-v_1 - v_4 + w_2 = 0$ |
| $x_1 - x_2 + u_2 = 0$ | $u_1^2/2 + v_1^2/2 + u_2^2/2 + v_2^2/2 - w_3^2/2 - w_4^2/2 = o_2$ |
| $y_1 - y_2 + v_2 = 0$ | $-u_1 - u_2 + w_3 = 0$ |
| $u_3^2 + v_3^2 = d_2^2$ | $-v_1 - v_2 + w_4 = 0$ |
| $x_2 - x_3 + u_3 = 0$ | $-x_0 + x_1 = c$ |

Figure 10: Constraint equations of Figure 3 (b) in separable form

Table 1: Numerical results of the bracket example

| ***Variables*** | **Initial values** | **Final values (after 20 iterations)** | **Descriptions** |
|---|---|---|---|
| | *$X_0$ = [0, 0.25, 0.5]* | *$X_0$ = [0, 0, 0]* | *x* coordinate of $P_0$ |
| | *$Y_0$ = [0, 0.25, 0.5]* | *$Y_0$ = [0, 0, 0]* | *y* coordinate of $P_0$ |
| | *$X_1$ = [0.5, 0.75, 1]* | *$X_1$ = [0.5, 0.505012, 0.510024]* | *x* coordinate of $P_1$ |
| | *$Y_1$ = [0, 0.25, 0.5]* | *$Y_1$ = [0, 0, 0]* | *y* coordinate of $P_1$ |
| | *$X_2$ = [0.5, 0.75, 1]* | *$X_2$ = [0.5, 0.516686, 0.533372]* | *x* coordinate of $P_2$ |
| | *$Y_2$ = [0, 0.25, 0.5]* | *$Y_2$ = [0.23886, 0.249714, 0.260569]* | *y* coordinate of $P_2$ |
| | *$X_3$ = [0, 0.25, 0.5]* | *$X_3$ = [0, 0.0116355, 0.0232709]* | *x* coordinate of $P_3$ |
| | *$Y_3$ = [0, 0.25, 0.5]* | *$Y_3$ = [0.238869, 0.249677, 0.260485]* | *y* coordinate of $P_3$ |
| ***Parameters*** | *$A_0$ = [0, 0, 0]* | | fixed position of $P_0$ |
| | *$B_0$ = [0, 0, 0]* | | fixed position of $P_0$ |
| | *$D_0$ = [0.49, 0.50, 0.51]* | | distance $d_0$ |
| | *$D_1$ = [0.24, 0.25, 0.26]* | | distance $d_1$ |
| | *$D_2$ = [0.49, 0.50, 0.51]* | | distance $d_2$ |
| | *$D_3$ = [0.24, 0.25, 0.26]* | | distance $d_3$ |
| | *$O_1$ = [-0.001, 0, 0.001]* | | perpendicularity |
| | *$O_2$ = [-0.001, 0, 0.001]* | | perpendicularity |

Figure 11 shows the convergence speed when solving the nonlinear equations. After 15 iterations, the widths of intervals are stabilized.

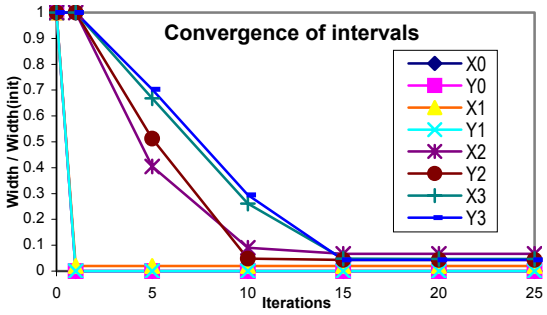



Figure 11: Convergence of Interval calculation in the bracket example

## 7. Conclusion

This paper presents a nominal interval constraint representation scheme based on nominal interval for CAD applications. It provides a generic numerical parameter scheme to represent inexactness at early design stages, uncertainty in detailed design, as well as the boundary information for design optimization. It relaxes the restriction of under-constrained and over-constrained situations for variational geometry. A generalized iterative nonlinear constraint solving method based on linear enclosure is developed for fast convergence. Inequalities are transformed into equations and can be solved uniformly. Interval subdivision and constraint re-specification methods are developed for design refinement. Active and inactive constraints are differentiated in sensitivity analysis.

## Acknowledgement

The author would like to thank Dr. G. William Walster and Mark Woodyard of Sun Microsystems and Nate Hayes of Sunfish Studio for discussions and comments, as well as the anonymous referees for the suggestions.

## References

1. Mudur, S.P. and Koparkar, P.A., "Interval Methods for Processing Geometric Objects", *IEEE Computer Graphics and Applications*, Vol.4, No.2 (February 1984), pp.7-17
2. Toth, D.L., "On Ray Tracing Parametric Surfaces", *Computer Graphics*, Vol.19, No.3 (July 1985), pp.171-179
3. Kalra, D. and Barr, A.H., "Guaranteed Ray Intersections with Implicit Surfaces", *Computer Graphics*, Vol.23, No.3 (July 1989), pp.297-304
4. Moore, M. and Wilhelms, J., "Collision Detection and Response for Computer Animation", *Computer Graphics*, Vol.22, No.4 (August 1988), pp.289-298
5. Von Herzen, B., Barr, A.H. and Zatz, H.R., "Geometric Collisions for Time-Dependent Parametric Surfaces", *Computer Graphics*, Vol.24, No.4 (August 1990), pp.39-48
6. Duff, T., "Interval Arithmetic and Recursive Subdivision for Implicit Functions and Constructive Solid Geometry", *Computer Graphics*, Vol.26, No.2 (July 1992), pp.131-138
7. Snyder, J., *Generative Modeling for Computer Graphics and CAD: Symbolic Shape Design Using Interval Analysis* (Cambridge, MA: Academic Press, 1992)

8. Snyder, J.M., Woodbury, A.R., Fleischer, K., Currin, B., and Barr, A.H., "Interval Methods for Multi-Point Collisions Between Time-Dependant Curved Surfaces", *ACM Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, , September 1993, New York, NY*, pp.321-334
9. Bliek, C., "Computer methods for design automation", *Ph.D. Thesis,* Massachusetts Institute of Technology, 1992
10. Rao, S.S. and Berke, L., "Analysis of uncertain structural systems using interval analysis", *AIAA Journal*, Vol.35, No.4, pp.727-735
11. Rao, S.S. and Cao, L. "Optimum design of mechanical systems involving interval parameters", *ASME Journal of Mechanical Design*, Vol.124, (September 2002), pp.465-472
12. Muhanna, R.L. and Mullen, R.L., "Formulation of Fuzzy Finite-Element Methods for Solid Mechanics Problems", *Computer-Aided Civil and Infrastructure Engineering*, Vol.14, (1999), pp. 107-117
13. Muhanna, R.L. and Mullen, R.L., "Uncertainty in mechanics problems --- interval-based approach", *ASCE Journal of Engineering Mechanics*, Vol.127, No.6 (June 2001), pp. 557-566
14. Finch, W.W. and Ward, A.C., "A set-based system for eliminating infeasible designs in engineering problems dominated by uncertainty", *ASME Proceedings of DETC97/dtm-3886*, Sept.14-17, 1997, Sacramento, CA, USA
15. Sederberg, T.W. and Farouki, R.T., "Approximation by Interval Bezier Curves", *IEEE Computer Graphics and Applications*, Vol.12, No.5 (September 1992), pp.87-95
16. Maekawa, T. and Patrikalakis, N.M., "Computation of Singularities and Intersections of Offsets of Planar Curves", *Computer Aided Geometric Design*, Vol.10, No.5 (Oct. 1993), pp.407-429
17. Maekawa, T. and Patrikalakis, N.M., "Interrogation of differential Geometry Properties for Design and Manufacture", *The Visual Computer*, Vol.10, No.4 (March 1994), pp.216-237
18. Hu, C.Y., Patrikalakis, N.M., and Ye, X., "Robust Interval Solid Modeling, Part II: Boundary Evaluation", *Computer-Aided Design*, Vol.28, No.10 (October, 1996), pp.819-830
19. Hu, C.-Y., Maekawa, T., Patrikalakis, N.M., and Ye, X., "Robust Interval Algorithm for Surface Intersections", *Computer-Aided Design*, Vol.29, No.9 (1997), pp.617-627
20. Abrams, S.L., Cho, W., Hu, C.Y., Maekawa, T., Patrikalakis, N.M., Sherbrooke, E.C., and Ye, X., "Efficient and Reliable Methods for Rounded-Interval Arithmetic", *Computer-Aided Design*, Vol.30, No.8 (July 1998), pp.657-665
21. Shen, G. and Patrikalakis, N.M., "Numerical and Geometric Properties of Interval B-Splines", *International Journal of Shape Modeling*, Vol.4 (1998), pp.31-62
22. Tuohy, S.T., Maekawa, T., Shen G., and Patrikalakis, N.M., "Approximation of Measured Data with Interval B-Splines", *Computer-Aided Design*, Vol.29, No.11 (1997), pp.791-799
23. Wallner, J., Krasauskas, R., and Pottmann, H., "Error Propagation in Geometric Constructions", *Computer-Aided Design*, Vol.32, No.11 (September, 2000), pp.631-641
24. Chen, F. and Lou, W., "Degree Reduction of Interval Bezier Curves", *Computer-Aided Design*, Vol.32, No.10 (Sept. 2000), pp.571-582
25. Lin, H., Liu, L., and Wang, G., "Boundary Evaluation for Interval Bezier Curve", *Computer-Aided Design*, Vol.34, No.9 (August 2002), pp.637-646
26. Ratschek, H. and Rokne, J., *New Computer Methods for Global Optimization* (New York: Ellis Horwood Limited, 1988), Ch.2, pp.28-29
27. Hansen, E., "An Overview of Global Optimization Using Interval Analysis", In: Moore, R.E., eds., *Reliability in Computing: The Role of Interval Methods in Scientific Computing* (Boston: Academic Press, 1988), , pp.289-305

28. Nickel, K., "How to Fight the Wrapping Effect", In: Nickel, K., eds., *Proceedings of the International Symposium on Interval Mathematics, September 23-26, 1985, Freiburg i. Br., Germany*, pp. 121-132
29. Alefeld, G. and Herzberger, J., *Introduction to Interval Computations* (New York: Academic Press, 1983)
30. Hansen, E. and Walster, G.W., *Global Optimization Using Interval Analysis* (New York: Marcel Dekker, 2004)
31. Kolev, L.V., "A New Method for Global Solution of Systems of Non-linear Equations", *Reliable Computing*, Vol.4, No.2 (May, 1998), pp.125-146
32. Kolev, L.V., "Automatic Computation of a Linear Interval Enclosure", *Reliable Computing*, Vol.7, No.1 (February, 2001), pp.17-28
33. Yamamura, K., "An Algorithm for Representing Functions of Many Variables by Superpositions of Functions of One Variable and Addition", *IEEE transactions on Circuits and Systems – I: Fundamental Theory and Application*, Vol.43, No.4 (April, 1996), pp.338-340
34. Collins, G.E. and Johnson, J.R., "Quantifier Elimination and the Sign Variation Method for Real Root Isolation", *Proceedings of the ACM-SIGSAM 1989 International Symposium on Symbolic and Algebraic Computation, July 17-19, 1989 Portland, Oregon*, pp.264-271
35. Collins, G.E. and Akritas, A.G., "Polynomial Real Root Isolation Using Descarte's Rule of Signs", *Proceedings of the Third ACM Symposium on Symbolic and Algebraic Computation, August 10-12, 1976, Yorktown Heights, New York*, pp.272-275

# Reliable Modeling and Optimization for Chemical Engineering Applications: Interval Analysis Approach

Youdong Lin, C. Ryan Gwaltney and Mark A. Stadtherr*
*Department of Chemical and Biomolecular Engineering, University of Notre Dame, Notre Dame, IN 46556, USA*

**Abstract.** In many applications of interest in chemical engineering it is necessary to deal with nonlinear models of complex physical phenomena, on scales ranging from the macroscopic to the molecular. Frequently these are problems that require solving a nonlinear equation system and/or finding the global optimum of a nonconvex function. Thus, the reliability with which these computations can be done often an important issue. Interval analysis provides tools with which these reliability issues can be addressed, allowing such problems to be solved with complete certainty. This presentation will focus on three types of applications: 1) Parameter estimation in modeling of phase equilibrium, including implications of using locally vs. globally optimal parameters in subsequent computations; 2) Nonlinear dynamics, in particular the location of equilibrium states and bifurcations of equilibria in ecosystem models used to assess the risk associated with the introduction of new chemicals into the environment; 3) Molecular modeling, with focus on transition state analysis of diffusion of a sorbate molecule in a zeolite.

## 1. Introduction

In many applications of interest in chemical engineering it is necessary to deal with nonlinear models of complex physical phenomena, on scales ranging from the macroscopic to the molecular. Frequently these are problems that require solving a nonlinear equation system and/or finding the global optimum of a nonconvex function. Thus, the reliability with which these computations can be done often an important issue. For example, if there are multiple solutions to the model, have all been located? If there are multiple local optima, has the global solution been found? Interval mathematics can provide the modeler with the tools needed to resolve these issues with mathematical and computational certainty, thus providing a degree of problem-solving reliability not available when using standard methods.

In recent years, it has been shown that strategies based on an interval-Newton approach can be used to reliably solve a wide variety of global optimization and nonlinear equation solving problems in chemical engineering, including computation of fluid phase equilibrium from activity coefficient models [35, 42, 45], cubic equation-of state (EOS) models [5, 19, 20, 44] and statistical associating fluid theory [50], calculation of critical points from cubic EOS models [43], location of azeotropes [32] and reactive azeotropes [33], computation of solid-fluid equilibrium [40, 51], parameter estimation using standard least squares [8] and error-in-variables (EIV) [9, 11, 10], and calculation of adsorption in nanoscale pores from a density function theory model [34]. In each case, the interval approach provides a mathematical and computational guarantee either that all solutions have been located in

* Author to whom all correspondence should be addressed. E-mail: markst@nd.edu

a nonlinear equation solving problem or that the global optimum has been found in an optimization problem.

In this paper, we will summarize recent work on three types of applications: 1) Parameter estimation in modeling of phase equilibrium, including implications of using locally vs. globally optimal parameters in subsequent computations; 2) Nonlinear dynamics, in particular the location of equilibrium states and bifurcations of equilibria in ecosystem models used to assess the risk associated with the introduction of new chemicals into the environment; and 3) Molecular modeling, with focus on transition state analysis of diffusion of a sorbate molecule in a zeolite. In the next section, we provide a brief outline of the interval-Newton methodology used for nonlinear equation solving and global optimization in the applications of interest.

## 2. Background

Several good introductions to interval computations are available [17, 22, 26, 37]. Of particular interest here is the interval-Newton method. Given an $n \times n$ nonlinear equation system $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ with a finite number of real roots in some initial interval, this technique provides the capability to find tight enclosures of *all* the roots of the system that lie within the given initial interval. For the unconstrained minimization of $\phi(\boldsymbol{x})$, a common approach is to seek stationary points, that is, to solve the nonlinear system $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\nabla}\phi(\boldsymbol{x}) = \boldsymbol{0}$. The global optimum will be one of roots of this nonlinear equation system, but there may be other roots as well, representing local optima and saddle points. To identify the global optimum, it is critical that none of the roots be missed, and such a guarantee can be provided by the interval-Newton approach. For a constrained optimization problem, the interval-Newton method can be applied to solve the KKT or Fritz-John conditions. In this section, we first summarize the interval-Newton methodology used, and then give a couple of simple examples that demonstrate the power of the approach.

### 2.1. Methodology

Given some initial interval $\boldsymbol{X}^{(0)}$, the interval-Newton algorithm is applied to a sequence of subintervals. For a subinterval $\boldsymbol{X}^{(k)}$ in the sequence, the first step is the *function range test*. An *interval extension* $\boldsymbol{F}(\boldsymbol{X}^{(k)})$ of the function $\boldsymbol{f}(\boldsymbol{x})$ is calculated. An interval extension provides upper and lower bounds on the range of values that a function may have in a given interval. It is often computed by substituting the given interval into the function and then evaluating the function using interval arithmetic. Thus the interval extension is often wider than the actual range of function values, but it always includes the actual range. If there is any component of the interval extension $\boldsymbol{F}(\boldsymbol{X}^{(k)})$ that does not include zero, then the interval can be discarded, since no solution of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ can exist in this interval. The next subinterval in the sequence may then be considered. Otherwise, testing of $\boldsymbol{X}^{(k)}$ continues. During this step, other interval-based techniques (e.g., constraint propagation) may also be applied to try to shrink $\boldsymbol{X}^{(k)}$ before proceeding.

For a global minimization problem, the next step is the *objective range test*. The interval extension $\Phi(\boldsymbol{X}^{(k)})$, containing the range of $\phi(\boldsymbol{x})$ over $\boldsymbol{X}^{(k)}$ is computed. If the lower bound of $\Phi(\boldsymbol{X}^{(k)})$ is greater than a known upper bound on the global minimum, then $\boldsymbol{X}^{(k)}$ can be discarded since it cannot contain the global minimum and need not be further tested. If it is known that $\boldsymbol{X}^{(k)}$ contains a point that can be used to update (reduce) the upper bound on the global minimum (i.e., if the upper bound of $\Phi(\boldsymbol{X}^{(k)})$ is less than the current upper bound on the global minimum), then this update is performed. This can be done in many different ways. A simple, cheap approach that we have used effectively is to evaluate $\phi(\boldsymbol{x})$ at the midpoint of $\boldsymbol{X}^{(k)}$ and use this to update the upper bound. Another approach is to use a local minimization routine starting at the midpoint of $\boldsymbol{X}^{(k)}$. For this purpose, we have used the simple, low-overhead direct search algorithm of Hooke and Jeeves [18, 25]. Use of the local minimizer involves additional computational overhead, but it most cases leads to a better upper bound on the global minimum. In cases when all the stationary points are desired rather than just the global minimum, this test step can be turned off.

The next step is the *interval-Newton test*. The linear interval equation system

$$\boldsymbol{F}'(\boldsymbol{X}^{(k)})(\boldsymbol{N}^{(k)} - \boldsymbol{x}^{(k)}) = -\boldsymbol{f}(\boldsymbol{x}^{(k)}), \tag{1}$$

is solved for a new interval $\boldsymbol{N}^{(k)}$, where $\boldsymbol{F}'(\boldsymbol{X}^{(k)})$ is an interval extension of the Jacobian of $\boldsymbol{f}(\boldsymbol{x})$, and $\boldsymbol{x}^{(k)}$ is an arbitrary point in $\boldsymbol{X}^{(k)}$. It has been shown [17, 26, 37] that any root contained in $\boldsymbol{X}^{(k)}$ is also contained in the *image* $\boldsymbol{N}^{(k)}$. This implies that if the intersection between $\boldsymbol{X}^{(k)}$ and $\boldsymbol{N}^{(k)}$ is empty, then no root exists in $\boldsymbol{X}^{(k)}$, and also suggests the iteration scheme $\boldsymbol{X}^{(k+1)} = \boldsymbol{X}^{(k)} \cap \boldsymbol{N}^{(k)}$. In addition, it has also been shown [17, 26, 37] that, if $\boldsymbol{N}^{(k)} \subset \boldsymbol{X}^{(k)}$, then there is a *unique* root contained in $\boldsymbol{X}^{(k)}$ and thus in $\boldsymbol{N}^{(k)}$. Thus, after computation of $\boldsymbol{N}^{(k)}$ from Eq. (1), there are three possibilities: (1) $\boldsymbol{X}^{(k)} \cap \boldsymbol{N}^{(k)} = \emptyset$, meaning there is no root in the current interval $\boldsymbol{X}^{(k)}$ and it can be discarded; (2) $\boldsymbol{N}^{(k)} \subset \boldsymbol{X}^{(k)}$, meaning that there is *exactly* one root in the current interval $\boldsymbol{X}^{(k)}$; (3) neither of the above, meaning that no conclusion can be drawn. In the last case, if $\boldsymbol{X}^{(k)} \cap \boldsymbol{N}^{(k)}$ is sufficiently smaller than $\boldsymbol{X}^{(k)}$, then the interval-Newton test can be reapplied to the resulting intersection, $\boldsymbol{X}^{(k+1)} = \boldsymbol{X}^{(k)} \cap \boldsymbol{N}^{(k)}$. Otherwise, the intersection $\boldsymbol{X}^{(k)} \cap \boldsymbol{N}^{(k)}$ is bisected, and the resulting two subintervals are added to the sequence (stack) of subintervals to be tested. If an interval containing a unique root has been identified, then this root can be tightly enclosed by continuing the interval-Newton iteration, which will converge quadratically to a desired tolerance (on the enclosure diameter).

This approach is referred to as an interval-Newton/generalized-bisection (IN/GB) method. At termination, when the subintervals in the sequence have all been tested, either all the real roots of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ have been tightly enclosed, or it is determined that no root exists. Applied to nonlinear equation solving problems, this can be regarded as a type of branch-and-prune scheme on a binary tree. Applied to global optimization problems, with the objective range test turned on, it can be regarded as a type of branch-and-bound scheme, again on a binary tree. It should be emphasized that the enclosure, existence, and uniqueness properties discussed above, which are the basis of the IN/GB method, can be derived without making any strong assumptions about the function $\boldsymbol{f}(\boldsymbol{x})$ for which roots

are sought. The function must have a *finite* number of roots over the search interval of interest; however, no special properties such as convexity or monotonicity are required, and $\boldsymbol{f}(\boldsymbol{x})$ may have transcendental terms.

Clearly, the solution of the linear interval system given by Eq. (1) is essential to this approach. To see the issues involved in solving such a system, consider the general linear interval system $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{B}$, where the matrix $\boldsymbol{A}$ and the right-hand-side vector $\boldsymbol{B}$ are interval-valued. The solution set $\mathbb{S}$ of this system is defined by $\mathbb{S} = \left\{ \boldsymbol{z} \,\middle|\, \tilde{A}\boldsymbol{z} = \boldsymbol{b}, \tilde{A} \in \boldsymbol{A}, \boldsymbol{b} \in \boldsymbol{B} \right\}$. However, in general this set is not an interval and may have a very complex, polygonal geometry. Thus to "solve" the linear interval system, one instead seeks an interval $\boldsymbol{Z}$ containing $\mathbb{S}$. Computing the interval hull (the tightest interval containing $\mathbb{S}$) is NP-hard [39], but there are several methods for determining an interval $\boldsymbol{Z}$ that contains but overestimates $\mathbb{S}$. Various interval-Newton methods differ in how they solve Eq. (1) for $\boldsymbol{N}^{(k)}$ and thus in the tightness with which the solution set is enclosed. By obtaining bounds that are as tight as possible, the overall performance of the interval-Newton approach can be improved, since with a smaller $\boldsymbol{N}^{(k)}$ the volume of $\boldsymbol{X}^{(k)} \cap \boldsymbol{N}^{(k)}$ is reduced, and it is also more likely that either $\boldsymbol{X}^{(k)} \cap \boldsymbol{N}^{(k)} = \emptyset$ or $\boldsymbol{N}^{(k)} \subset \boldsymbol{X}^{(k)}$ will be satisfied. Thus, intervals that may contain solutions of the nonlinear system are more quickly contracted, and intervals that contain no solution or that contain a unique solution may be more quickly identified, all of which leads to a likely reduction in the number of bisections needed.

Frequently, $\boldsymbol{N}^{(k)}$ is computed component-wise using an interval Gauss-Seidel approach, preconditioned with an inverse-midpoint matrix. Though the inverse-midpoint preconditioner is a good general-purpose preconditioner, it is not always the most effective approach [26]. Recently, a hybrid preconditioning approach (HP/RP), which combines a simple pivoting preconditioner with the standard inverse-midpoint scheme, has been described by Gau and Stadtherr [12] and shown to achieve substantially more efficient computational performance than the inverse-midpoint preconditioner alone, in some cases by multiple orders of magnitude. However, it still cannot yield the tightest enclosure of the solution set, which, as noted above, is in general an NP-hard problem. Lin and Stadtherr [29, 31] have recently suggested a strategy (LISS_LP) based on linear programming (LP) for solving the linear interval system, Eq. (1), arising in the context of interval-Newton methods. Using this approach, exact component-wise bounds on the solution set can be calculated, while avoiding exponential time complexity. In numerical experiments [29, 31], LISS_LP has been shown to achieve further computational performance improvements compared with HP/RP.

## 2.2. Examples

To provide some initial examples of the power of this methodology, we use two global optimization problems, both of which have a very large number of local minima.

### 2.2.1. *Trefethen Challenge Problem*

This is a global optimization problem given by Trefethen [46] as part of a set of challenge problems in which at least 10 digits of precision were required in the final results. The global

minimum of the function

$$f(x,y) = \exp(\sin(50x)) + \sin(60\exp(y)) + \sin(70\sin(x)) + \sin(\sin(80y)) - \sin(10(x+y)) + (x^2+y^2)/4 \tag{2}$$

is sought, where $x \in [-1,1]$ and $y \in [-1,1]$. On the unit square ($[0,1] \times [0,1]$) alone, the function has 667 local minima, as well as many other stationary points.

This global optimization problem was solved successfully, with more than 10 digits of precision, in only 0.16 seconds CPU time on a Sun Blade 1000 model 1600 workstation, using the LISS_LP approach. The results for the global optimum are

$$x \in [-0.02440307969437517, -0.02440307969437516],$$

$$y \in [0.2106124271553557, 0.2106124271553558],$$

and

$$f \in [-3.306868647475245, -3.306868647475232]$$

This proves to be a very easy problem to solve using the interval approach.

2.2.2. *Siirola's Problem*

This problem is to find the global minimum of the function

$$f(\boldsymbol{x}) = 100 \prod_{i=1}^{N} \sum_{j=1}^{5} \left( \frac{j^5}{4425} \cos(j + jx_i) \right) + \frac{1}{N} \sum_{i=1}^{N} (x_i - x_{0,i})^2, \tag{3}$$

where $x_i \in [x_{0,i} - 20, x_{0,i} + 20]$ and $x_{0,i} = 3$, $i = 1, ..., N$. This is used as a test problem by Siirola *et al.* [41]. There are 2048 local minima for the case $N = 2$ and on the order of a hundred million ($10^8$) local minima for the case $N = 5$. The problem also has multiple ($N$) global minimizer points. The problems were solved for the cases of $N = 2$ to $N = 6$ on a Dell workstation (1.7 GHz Intel Xeon processor running Linux) using LISS_LP with local minimizer.

Results are shown in Table I. For each value of $N$, there are $N$ global minimizer points, all of which have been found. The global minimizer points can all be expressed in terms of only two numbers, denoted in Table I as $x_i^*$ and $x_{j\neq i}^*$. The $i$-th global minimizer point will have the value $x_i^*$ for its $i$-th element, and the value $x_{j\neq i}^*$ for its other $N-1$ elements. Again this proves to be a relatively easy problem to solve using the interval methodology. The results also show the exponential complexity that may be associated with deterministic global optimization (in general, an NP-hard problem).

The subsequent sections will now focus on three types of actual applications in chemical engineering, involving parameter estimation, nonlinear dynamics, and molecular modeling.

Table I. Global solution of Siirola's problem.

| $N$ | Global Minimizer Points $x_i^*$ | $x_{j\neq i}^*$ | Global Minimum | CPU time (s) |
|---|---|---|---|---|
| 2 | 4.6198510288 | 5.2820519601 | -88.1046253312 | 0.07 |
| 3 | 4.6201099154 | 5.2824296177 | -87.6730486951 | 2.12 |
| 4 | 4.6202393815 | 5.2826184940 | -87.4572049443 | 33.95 |
| 5 | 4.6203170683 | 5.2827318347 | -87.3276809494 | 413.61 |
| 6 | 4.6203688625 | 5.2828074014 | -87.2413242244 | 4566.42 |

## 3. Parameter Estimation in VLE Modeling

Because of its importance in the design of separation systems such as distillation, much attention has been given to modeling the thermodynamics of phase equilibrium in fluid mixtures, especially the case of vapor-liquid equilibrium (VLE). Typically these models take the form of excess Gibbs energy models or equation of state models, with binary parameters in the models determined by parameter estimation from experimental data. As an example, we consider here the estimation from binary VLE data of the energy parameters in the Wilson equation for liquid phase activity coefficient.

### 3.1. Problem Formulation

Expressed in terms of the molar excess Gibbs energy $g^E$ for a binary system, and the liquid-phase mole fractions $x_1$ and $x_2$, the Wilson equation is

$$\frac{g^E}{RT} = -x_1 \ln(x_1 + \Lambda_{12}x_2) - x_2 \ln(x_2 + \Lambda_{21}x_1) \tag{4}$$

from which expressions for the activity coefficients are

$$\ln \gamma_1 = -\ln(x_1 + \Lambda_{12}x_2) + x_2 \left[\frac{\Lambda_{12}}{x_1 + \Lambda_{12}x_2} - \frac{\Lambda_{21}}{\Lambda_{21}x_1 + x_2}\right] \tag{5}$$

$$\ln \gamma_2 = -\ln(x_2 + \Lambda_{21}x_1) - x_1 \left[\frac{\Lambda_{12}}{x_1 + \Lambda_{12}x_2} - \frac{\Lambda_{21}}{\Lambda_{21}x_1 + x_2}\right]. \tag{6}$$

The binary parameters $\Lambda_{12}$ and $\Lambda_{21}$ are given by

$$\Lambda_{12} = \frac{v_2}{v_1} \exp\left[-\frac{\theta_1}{RT}\right] \tag{7}$$

$$\Lambda_{21} = \frac{v_1}{v_2} \exp\left[-\frac{\theta_2}{RT}\right], \tag{8}$$

where $v_1$ and $v_2$ are the pure component liquid molar volumes, $T$ is the system temperature, $R$ is the gas constant, and $\theta_1$ and $\theta_2$ are the energy parameters that need to be estimated.

Given VLE measurements and assuming an ideal vapor phase, experimental values $\gamma_{1,\text{exp}}$ and $\gamma_{2,\text{exp}}$ of the activity coefficients can be obtained from the relation

$$\gamma_{i,\text{exp}} = \frac{y_{i,\text{exp}} P_{\text{exp}}}{x_{i,\text{exp}} P_i^0}, \quad i = 1, 2, \tag{9}$$

where $x_{i,\text{exp}}$ and $y_{i,\text{exp}}$ are, respectively, the experimental liquid and vapor phase mole fractions of component $i$, $P_{\text{exp}}$ is the experimental pressure, and $P_i^0$ is the vapor pressure of pure component $i$ at the system temperature $T$. For the example problem here we follow Gmehling *et al.* [13] and use the relative least squares objective

$$\phi(\boldsymbol{\theta}) \equiv \sum_{j=1}^{n} \sum_{i=1}^{2} \left( \frac{\gamma_{ji,\text{exp}} - \gamma_{ji,\text{calc}}(\boldsymbol{\theta})}{\gamma_{ji,\text{exp}}} \right)^2, \tag{10}$$

where the $\gamma_{ji,\text{calc}}(\boldsymbol{\theta})$ are calculated from the Wilson equation at conditions (temperature, pressure and composition) coincident to those used when measuring $\gamma_{ji,\text{exp}}$, and $n$ is the number of data points.

### 3.2. Results and Discussion

This parameter estimation problem has been solved for a large number of systems, and results presented in the DECHEMA VLE Data Collection [13]. Gau *et al.* [8] applied an interval-Newton approach to a few systems to determine the globally optimal parameters, and found that, in several cases, the parameters reported in the DECHEMA collection were only locally optimal parameters. A particularly interesting problem is the system benzene(1) – hexafluorobenzene(2), for which there are ten data sets, both isothermal and isobaric, found in DECHEMA. As shown in Table II, using the interval-Newton methodology (IN/GB), new globally optimal parameter values are discovered in five of the ten cases. CPU times are on a Sun Ultra 2/1300 workstation.

While the globally optimal parameter values provide a somewhat better prediction of activity coefficients, as measured by the relative least squares objective $\phi$, it is not clear whether this better fit will actually result in more accurate calculations of vapor-liquid equilibrium from the activity coefficient model. To test this, for the five cases in which new globally optimal parameters were found, we used both the locally optimal parameters (DECHEMA) and the globally optimal parameters (IN/GB) to predict the presence and location of homogeneous azeotropes. A homogeneous azeotrope is an equilibrium state in which the vapor and liquid phases have the same composition. Knowledge of azeotropes is critical in the design of distillation operations. Since separation by distillation is based on the difference in composition between liquid and vapor phases, if there is a homogeneous azeotrope at some composition, it will create a bottleneck beyond which no further separation can occur. The method of Maier *et al.* [32], which employs an interval method and is guaranteed to find all homogeneous azeotropes, or determine with certainty that there are none, was used to do the computation of azeotropes.

Table II. Parameter estimation for benzene(1) – hexafluorobenzene(2) system.

| Data Set | Volume: Page[1] | Data points | $T(^oC)$ or P (mmHg) | DECHEMA | | | IN/GB | | | No. of Minima | CPU time(s) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $\theta_1$ | $\theta_2$ | $\phi(\boldsymbol{\theta})$ | $\theta_1$ | $\theta_2$ | $\phi(\boldsymbol{\theta})$ | | |
| 1* | 7:228 | 10 | $T$=30 | 437 | -437 | 0.0382 | **-468** | **1314** | **0.0118** | 2 | 19.2 |
| 2* | 7:229 | 10 | 40 | 405 | -405 | 0.0327 | **-459** | **1227** | **0.0079** | 2 | 17.6 |
| 3* | 7:230 | 10 | 50 | 374 | -374 | 0.0289 | **-449** | **1157** | **0.0058** | 2 | 15.8 |
| 4* | 7:233 | 11 | 50 | 342 | -342 | 0.0428 | **-424** | **984** | **0.0089** | 2 | 14.1 |
| 5 | 7:231 | 10 | 60 | -439 | 1096 | 0.0047 | -439 | 1094 | 0.0047 | 2 | 12.4 |
| 6 | 7:232 | 9 | 70 | -424 | 1035 | 0.0032 | -425 | 1036 | 0.0032 | 2 | 10.1 |
| 7* | 7:234 | 17 | $P$=300 | 344 | -347 | 0.0566 | **-432** | **993** | **0.0149** | 2 | 22.5 |
| 8 | 7:235 | 16 | 500 | -405 | 906 | 0.0083 | -407 | 912 | 0.0083 | 2 | 18.3 |
| 9 | 7:236 | 17 | 760 | -407 | 923 | 0.0057 | -399 | 908 | 0.0053 | 1 | 17.9 |
| 10 | 7:226 | 29 | 760 | -333 | 702 | 0.0146 | -335 | 705 | 0.0146 | 2 | 26.1 |

[1]Refers to volume and page numbers in DECHEMA VLE Data Collection [13].
*New globally optimal parameters found.

Results of the azeotrope calculations are shown in Table III, along with experimental data indicating that this system has two homogeneous azeotropes. However, when the locally optimal parameters reported in DECHEMA are used in azeotrope prediction, there are three cases in which no azeotrope is found, and in the remaining two cases only one azeotrope is found. Using the globally optimal parameters found using the interval method, two azeotropes are predicted in all cases. In this case, by finding the globally, as opposed to locally, optimal parameter values, it clearly makes the difference between predicting physical reality or not. If the DECHEMA parameters are used, one would conclude that the Wilson equation is a very poor model. However, when the globally optimal parameters values are used, it appears that the Wilson equation is actually a relatively good model, though a better prediction of the azeotrope compositions would be desirable.

The difference between the use of the globally and locally optimal parameters can also have an effect on many other types of calculations. For example, Ulas *et al.* [48] demonstrate how batch distillation optimal control profiles are affected by using the globally optimal parameter values predicted by IN/GB, versus the locally optimal parameters published in DECHEMA. Since batch distillation is a dynamic process, the uncertainties in model parameters are translated into time-dependent uncertainties. Two different time-dependent relative volatility profiles are obtained using global and local parameter values for the Wilson model. These profiles are statistically analyzed and represented by Ito processes. The batch distillation optimal control problem is then solved for three cases: the stochastic global case (relative volatility is represented by an Ito process, obtained from global parameters), the stochastic local case (relative volatility is represented by an Ito process, obtained from local parameters) and the deterministic case (relative volatility is taken as constant). The results of these case studies show that the stochastic global reflux ratio profile results in the highest product yield and the product purity is significantly closer to the specified purity for optimal control.

In addition to problems involving a simple least squares objective, such as discussed above, the interval methodology can also be applied to parameter estimation problems in which the error-in-variables (EIV) approach is used. For example, Gau and Stadtherr [9, 11, 10], consider EIV parameter estimation problems in the modeling of VLE, reaction kinetics, and heat exchange networks, and solve them using the HP/RP algorithm for the interval-Newton method. When the EIV approach is used, the dimensionality of the optimization problem becomes much larger. The largest problem solved was a heat exchanger network problem with 264 variables [11]. Parameter estimation problems that require solving a nonlinear and nonconvex optimization problem, and for which there is thus the potential for multiple local optima, occur in many areas of engineering and science. This is an area in which use of an interval approach to guarantee global optimality could have a significant impact.

Table III. Azeotrope prediction for benzene(1) – hexafluorobenzene(2) system.

| Data | $T(^oC)or$ | DECHEMA | | | IN/GB | | | Experiment | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Set | P (mmHg) | $x_1$ | $x_2$ | $P$ or $T$ | $x_1$ | $x_2$ | $P$ or $T$ | $x_1$ | $x_2$ | $P$ or $T$ |
| 1 | $T$=30 | 0.0660 | 0.9340 | $P$=107 | 0.0541 | 0.9459 | $P$=107 | 0.15 | 0.85 | $P$=107 |
| | | | | | 0.9342 | 0.0658 | 121 | 0.95 | 0.05 | 120 |
| 2 | 40 | 0.0315 | 0.9685 | 168 | 0.0761 | 0.9239 | 168 | 0.16 | 0.84 | 167 |
| | | | | | 0.9244 | 0.0756 | 185 | 0.93 | 0.07 | 183 |
| 3 | 50 | NONE | | | 0.0988 | 0.9012 | 255 | 0.17 | 0.83 | 254 |
| | | | | | 0.9114 | 0.0886 | 275 | 0.90 | 0.10 | 273 |
| 4 | 50 | NONE | | | 0.0588 | 0.9412 | 256 | 0.17 | 0.83 | 254 |
| | | | | | 0.9113 | 0.0887 | 274 | 0.90 | 0.10 | 273 |
| 7 | $P$=300 | NONE | | | 0.1612 | 0.8388 | $T$=54.13 | 0.20 | 0.80 | $T$=54.55 |
| | | | | | 0.9315 | 0.0685 | 52.49 | 0.89 | 0.11 | 52.50 |

## 4. Nonlinear Dynamics: Ecological Modeling

A problem of frequent interest in many fields of science and engineering is the study of nonlinear dynamics. Through the use of bifurcation diagrams, a large amount of information concerning the number and stability of equilibria in a nonlinear ODE model can be concisely represented. Bifurcations of equilibria are typically found by solving a nonlinear algebraic system consisting of the equilibrium (steady-state) conditions along with one or more augmenting functions. Typically this equation system is solved using some continuation-based tool (e.g., AUTO [6]). However, in general, these methods do not provide any guarantee that all bifurcations will be found, and are often initialization dependent. Thus, without some *a priori* knowledge of system behavior, one may not know with complete certainty if all bifurcation curves have been identified and explored. We demonstrate here the use of an interval-Newton methodology as a way to ensure that *all* equilibrium states and bifurcations of interest are found.

In particular, we are interested in locating equilibrium states and bifurcations in food chain models. These models are descriptive of a wide range of behaviors in the environment, and are useful as a tool to perform ecological risk assessments. Our interest in ecological modeling is motivated by its use as one tool in studying the impact on the environment of the industrial use of newly discovered materials. Clearly it is preferable to take a proactive, rather than reactive, approach when considering the safety and environmental consequences of using new compounds. Of particular interest is the potential use of room temperature ionic liquid (IL) solvents in place of traditional solvents [4]. IL solvents have no measurable vapor pressure and thus, from a safety and environmental viewpoint, have several potential advantages relative to the traditional volatile organic compounds (VOCs) used as solvents, including elimination of hazards due to inhalation, explosion and air pollution. However, ILs are, to varying degrees, soluble in water; thus, if they are used industrially on a large scale, their entry into the environment via aqueous waste streams is of concern. The effects of trace levels of ILs in the environment are today essentially unknown and thus must be studied. Single species toxicity information is very important as a basis for examining the effects that a contaminant will have on an environment. However, this information, when considered by itself, is insufficient to predict impacts on a food chain, food web, or an ecosystem. Ecological modeling provides a means for studying the impact of such perturbations on a localized environment by focusing not just on the impact on one species, but rather on the larger impacts on the food chain and ecosystem. Of course, ecological modeling is just one part of a much larger suite of tools, including toxicological [7, 21], hydrological and microbiological studies, that must be used in addressing this issue.

Food chain models are often simple, but display rich mathematical behavior, with varying numbers and stability of equilibria that depend on the model parameters (e.g., [14, 36]). Therefore, bifurcation analysis is quite useful in characterizing the mathematical behavior of predator/prey systems, as it allows for the concise representation of model behavior over a wide range of parameters. We will focus on one particular food chain model here, namely a tritrophic (prey, predator, superpredator) Rosenzweig-MacArthur model, as described in much more detail by Gwaltney *et al.* [16]

### 4.1. Problem Formulation

The Rosenzweig-MacArthur model features a logistic prey ($i = 1$), and hyperbolic (Holling Type II) predator ($i = 2$) and superpredator ($i = 3$) responses. In terms of the biomasses $x_1$, $x_2$ and $x_3$, the model is given by

$$\frac{dx_1}{dt} = x_1 \left[ r\left(1 - \frac{x_1}{K}\right) - \frac{a_2 x_2}{b_2 + x_1} \right] \tag{11}$$

$$\frac{dx_2}{dt} = x_2 \left[ e_2 \frac{a_2 x_1}{b_2 + x_1} - \frac{a_3 x_3}{b_3 + x_2} - d_2 \right] \tag{12}$$

$$\frac{dx_3}{dt} = x_3 \left[ e_3 \frac{a_3 x_2}{b_3 + x_2} - d_3 \right]. \tag{13}$$

Here $r$ is the prey growth rate constant, $K$ is the prey carrying capacity of the ecosystem, the $d_i$ are death rate constants, the $a_i$ represent maximum predation rates, the $b_i$ are half-saturation constants, and the $e_i$ are predation efficiencies.

The equilibrium (steady-state) condition is simply

$$d\mathbf{x}/dt = \mathbf{0}, \tag{14}$$

which in this case is subject to the feasibility condition $\mathbf{x} \geq \mathbf{0}$. Thus, once all the model parameters have been specified, there is a $3 \times 3$ system of nonlinear equations to be solved for the equilibrium states. The stability of these states can be determined from the eigenvalues of the Jacobian $J$ (of $d\mathbf{x}/dt$). According to linear stability analysis, for an equilibrium state to be stable, all of the eigenvalues of the Jacobian must have negative real parts. In addition to equilibrium states, we are also interested in computing bifurcations of equilibria. These include the appearance and disappearance of equilibrium states (fold or saddle node bifurcation), the exchange of stability of two equilibria (transcritical bifurcation), and the change of stability of an equilibrium point (Hopf bifurcation). Three types of codimension-1 bifurcations, namely fold, transcritical and Hopf, and two types of codimension-2 bifurcations, namely double-fold (or double-zero) and fold-Hopf are of particular interest. For codimension-1 bifurcations there is one free parameter and one additional augmenting condition that must be satisfied. For a fold or transcritical bifurcation the additional condition is that an eigenvalue of the Jacobian is zero, or equivalently

$$\det[J(\mathbf{x}, \alpha)] = 0, \tag{15}$$

where $\alpha$ is the free parameter. For a Hopf bifurcation the additional condition is that the Jacobian has a pair of complex conjugate eigenvalues whose real parts are zero. This condition can also be expressed [28] in terms of a bialternate product as

$$\det[2J(\mathbf{x}, \alpha) \odot I] = 0. \tag{16}$$

It can also be shown that to locate a double-fold or a fold-Hopf codimension-two bifurcation of equilibrium, the equilibrium condition can be augmented with the two additional

equations

$$\det[J(\mathbf{x}, \alpha, \beta)] = 0 \tag{17}$$

$$\det[2J(\mathbf{x}, \alpha, \beta) \odot I] = 0 \tag{18}$$

and two additional variables (free parameters) $\alpha$ and $\beta$.

Whether one is looking for equilibrium states, or the bifurcations of equilibria discussed above, there is a system of nonlinear equations to be solved that may have multiple solutions, or no solutions, and the number of solutions may be unknown *a priori*. For simple models, including the Rosenzweig-MacArthur model, it may be possible to solve for some of equilibrium states and bifurcations analytically, but for more complex models a computational method is needed that is capable of finding, with certainty, all the solutions of the nonlinear equation system.

### 4.2. Results and Discussion

Following Gragnani *et al.* [14], the parameters used were set to $a_2 = 5/3$, $b_2 = 1/3$, $e_2 = 1$, $d_2 = 0.4$, $a_3 = 0.05$, $b_3 = 0.5$, $e_3 = 1$, and $d_3 = 0.01$. A bifurcation diagram with the prey carrying capacity, $K$, and the prey growth rate constant, $r$, as the free parameters was then computed using the IN/GB methodology, with the result shown in Fig. 1. In an $r$ vs. $K$ bifurcation diagram the values of $r$ at which bifurcations occur are plotted as a function of $K$. Such a diagram was generated here by using the IN/GB method to repeatedly solve the augmented systems for $r$ and $\boldsymbol{x}$ for slightly different values of $K$, going from $K = 0$ to $K = 2$ in steps of $K = 0.005$. There may be some values of $K$ for which one of the augmented systems has an infinite number of solutions for $r$ (i.e., the vertical line in Fig. 1). This case cannot be handled directly by the IN/GB technique, or could be missed entirely by the stepping in $K$. Thus, to ensure that all of the bifurcations are found, it is necessary to also scan in the $r$ direction. That is, the IN/GB method was also used to repeatedly solve the augmented systems for $K$ and $\boldsymbol{x}$ for slightly different values of $r$, in this case going from $r = 0$ to $r = 2$ in steps of $r = 0.005$. To locate codimension-two bifurcations (double-fold and fold-Hopf), the IN/GB method was used to solve the doubly-augmented system given by Eqs. (14,17,18) for $K$, $r$ and $\boldsymbol{x}$. The average CPU time (1.7 GHz Intel Xeon processor running Linux) for each solution of Eqs. (14,15) for fold and transcritical bifurcations was about 0.6 seconds, and for each solution of Eqs. (14,16) for Hopf bifurcations was about 1.4 seconds. Solving Eqs. (14,17,18) for codimension-two bifurcations required about 39 seconds. The initial intervals used for the components of $\boldsymbol{x}$ were in all cases [0, 5000] and for the parameters $K$ and $r$ were [0, 2].

As shown in Fig. 1, fold and transcritical of equilibria curves were both found, and are labeled FE and TE respectively. Hopf bifurcation curves were also found, and are labeled H or $H_p$ (for planar Hopf). A planar Hopf bifurcation is one that occurs in a independent two-variable subset of state space. A single fold-Hopf bifurcation was located; this point is represented as an open diamond and labeled FH (no double-fold bifurcations were found). This bifurcation diagram corresponds exactly with the known $K$ vs. $r$ bifurcation diagram for this model, as reported by Gragnani *et al.* [14] This confirms the utility and accuracy of

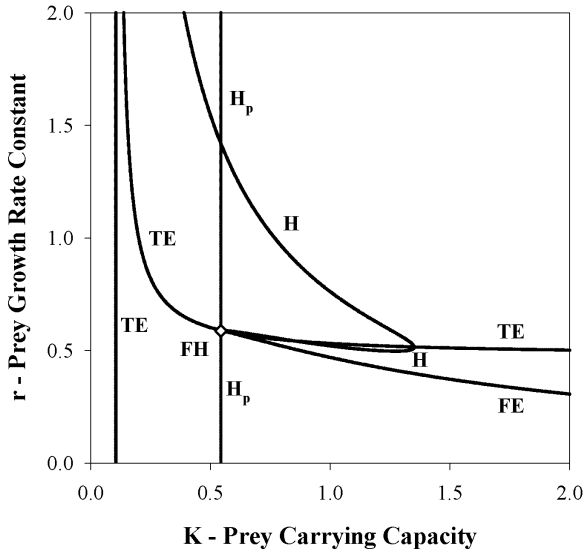



*Figure 1.* Bifurcation diagram of $r$ vs. $K$. TE: Transcritical of equilibrium; FE: Fold of equilibrium; H: Hopf; $H_p$: Planar Hopf; FH: Fold-Hopf codimension-2.

the IN/GB algorithm for computing bifurcation of equilibria diagrams. Bifurcation diagrams such as this can be very easily and automatically generated using the IN/GB methodology, with complete certainty that all bifurcation curves have been found.

Using the same procedure as described above, a $d_2$ vs. $K$ bifurcation diagram for the Rosenzweig-MacArthur model was also generated. The predator death rate constant $d_2$ is now a free parameter, and $r$ is now a fixed parameter set at $r = 1$. The resulting bifurcation diagram is shown in Fig. 2. This diagram illustrates that at a constant prey carrying capacity and growth rate constant ($r = 1$), increasing or decreasing the predator death rate will cause macroscopic changes (bifurcations) in system behavior. For relatively small values of $K$, there are two transcritical bifurcations that occur as $d_2$ is changed, and for larger values of $K$ there are also two Hopf bifurcations. No double-fold or fold-Hopf codimension-two bifurcations were found. In order to more closely observe these changes in behavior, solution branch diagrams showing the equilibrium states were generated by using IN/GB to solve Eq. (14) for the case of $K = 1$. Fig. 3 gives the solution branch diagrams for $\boldsymbol{x}$ as $d_2$ is varied from 0 to 2.

Based on the bifurcation diagram (Fig. 2) at $K = 1$, we would expect that as $d_2$ is increased from 0 to 2, there should be observed first a Hopf bifurcation (the planar Hopf is not observed in this case, due to the sign of the third eigenvalue) and then two transcritical bifurcations. This is what is in fact seen in Fig. 3. These diagrams illustrate that there is a minimum predator death rate constant $d_2$ that results in stable system behavior. At low predator death rates, the system is unstable and likely exhibits cycles of population booms and busts. As the predator death rate increases, enough predators are dying off at

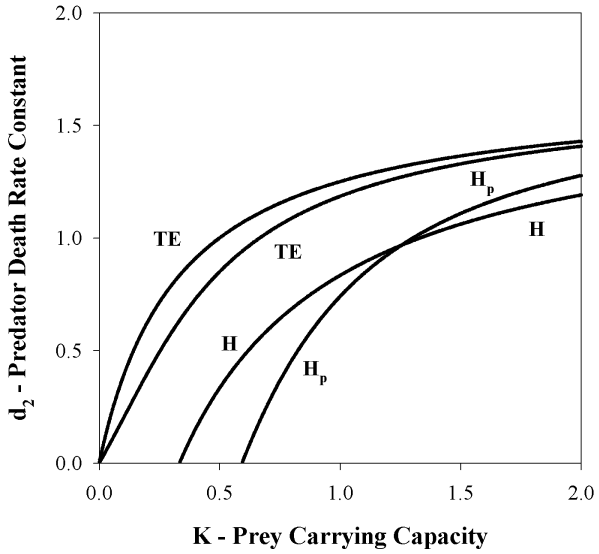



*Figure 2.* Bifurcation diagram of $d_2$ vs. $K$. TE: Transcritical of equilibrium; FE: Fold of equilibrium; H: Hopf; $H_p$: Planar Hopf.

any given time to prevent the cycles from occurring, and the cycles collapse to a stable steady-state in a Hopf bifurcation. These results also give a sense of the effects of releasing a toxin that specifically targets the predator trophic level, and increases the predator death rate constant. Prior to examining these diagrams, one would expect that such a release would have an impact on both the predator and the superpredator populations. The plot of $x_3$ in Fig. 3 shows that increasing the predator death rate constant causes a linear decrease in the stable superpredator biomass. However, according to the plot of $x_2$ in Fig. 3, the stable predator population is not affected until the superpredator population reaches zero. Though these results may seem somewhat counterintuitive, they are indicative of the complex interactions that may occur in food chains. An ecotoxin released at a very low concentration could affect organisms at different trophic levels to varying degrees. For the case considered here, one might observe an impact on the superpredator population and thus assume that the effect of the ecotoxin was at that level, even though the actual effect is on the predator level (death rate constant $d_2$). Using models such as this one can obtain insights into the impacts of an ecotoxin that might not otherwise be apparent.

The interval methodology has been applied successfully to several other ecological models by Gwaltney *et al.* [16] and Gwaltney and Stadtherr [15]. We anticipate that this methodology will also be useful for computing equilibrium states and bifurcations of equilibria in a wide variety of other problems in engineering and science in which nonlinear dynamical behavior is of interest.

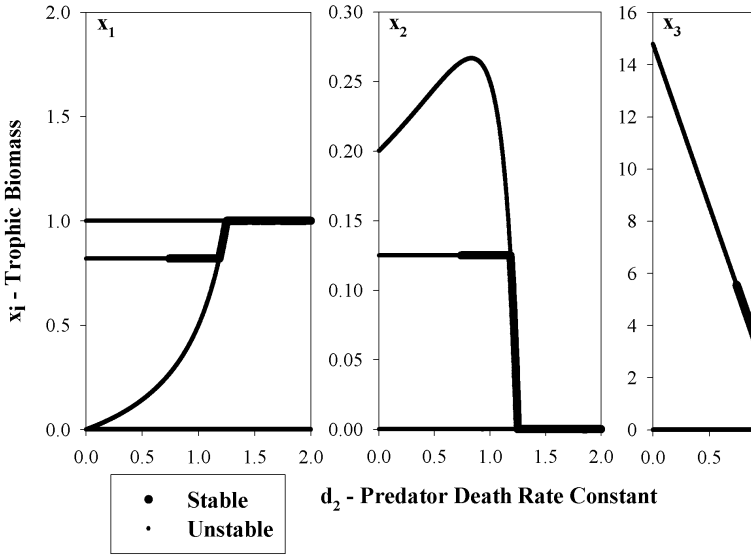



*Figure 3.* Solution branch diagram illustrating the change in equilibrium states (species biomass) with changes in $d_2$. From left to right: prey, predator, and superpredator biomasses. $K = 1$ and $r = 1$ for all three plots.

## 5. Molecular Modeling: Transition State Analysis

Transition-state theory is a well-established methodology which, by providing an approach for computing the kinetics of infrequent events, is useful in the study of numerous physical systems. Classically, it assumes that there exists a potential energy hypersurface which divides the space into a reactant region and a product region. Although the theory was originally for interpretation of chemical reaction rates, it can be amended for non-reacting systems, including desorption/adsorption and diffusion processes in which no chemical bonds are broken or made.

Of particular interest here is the problem of computing the diffusivity of a sorbate molecule in a zeolite. This can be done using the methodology of transition-state theory, as described by June *et al.* [23] It is assumed that diffusive motion of the sorbate molecules through the zeolite occurs by a series of uncorrelated hops between potential minima in the zeolite lattice. A sorption state or site is constructed around each minimum of the potential energy hypersurface. A first order rate constant, $k_{ij}$, is then associated with the rate of transition between a given pair of neighboring sites, $i$ and $j$. Any such pair of sites is then assumed to be separated by a dividing surface on which a saddle point of the potential energy hypersurface is located. The saddle point can be viewed as the transition state between sites, and a couple of steepest decent paths from the saddle point connect the minima associated with the $i$ and $j$ sites. After rate constants have been determined for all possible transitions between the sorption sites, a continuous-time/discrete-space Monte Carlo calculation can

then be used to determine the self-diffusivity of the sorbate molecules. Obviously, in this application, and in other applications of transition-state theory, finding all local minima and saddle points of the potential energy surface, $\mathcal{V}$, is critical. We demonstrate here, using a sorbate-zeolite system, the use of the interval-Newton methodology to find all stationary points of a potential energy surface.

Stationary points satisfy the condition $\boldsymbol{g} = \boldsymbol{\nabla}\mathcal{V} = \mathbf{0}$; that is, at a stationary point the gradient of the potential energy surface is zero. Using the eigenvalues of $H = \nabla^2\mathcal{V}$, the Hessian of the potential energy surface, stationary points can be classified into local minima, local maxima, and saddle points (of order determined by the number of negative eigenvalues). There are a number of methods for locating stationary points. A Newton or quasi-Newton method, applied to solve the nonlinear equation system $\boldsymbol{\nabla}\mathcal{V} = \mathbf{0}$, will yield a solution whenever the initial guess is sufficiently close to a stationary point. This method can be used in an exhaustive search, using many different initial guesses, to locate stationary points. The set of initial guesses to use might be determined by the user (intuitively or arbitrarily) or by some type of stochastic multistart approach. Another popular approach is the use of eigenmode-following methods, as done, for example, by Tsai and Jordan [47]. These methods can be regarded as variations of Newton's method. In an eigenmode-following algorithm, the Newton step is modified by shifting some of the eigenvalues of the Hessian (from positive to negative or vice versa). By selection of the shift parameters, one can effectively find the desired type of stationary points, e.g. minima and first-order saddles. There are also a number of other approaches, many involving some stochastic component, for finding stationary points.

In the context of sorbate-zeolite systems, June *et al.* [23] use an approach in which minima and saddle points are located separately. A three step process is employed in an exhaustive search for minima. First, the volume of the search space (one asymmetric unit) is discretized by a grid with a spacing of approximately 0.2Å, and the potential and gradient vector are tabulated on the grid. Second, each cube formed by a set of nearest-neighbor grid nodes is scanned, and the three components of the gradient vector on the eight vertices of the cube checked for changes in sign. Finally, if all three components are found to change sign on two or more vertices of the cube, a BFGS quasi-Newton minimization search algorithm is initiated to locate a local minimum, using the coordinates of the center of the cube as the initial guess. Two different algorithms are tried for determining the location of saddle points. One searches for global minimizers in the function $\boldsymbol{g}^{\mathrm{T}}\boldsymbol{g}$, i.e. the sum of the squares of the components of the gradient vector. The other algorithm, due to Baker [3], searches for saddle points directly from an initial point by maximizing the potential energy along the eigenvector direction associated with the smallest eigenvalue and by minimizing along directions associated with all other eigenvalues of the Hessian.

All the methods discussed above, however, have a major shortcoming, namely that they provide no guarantee that *all* local minima and first order saddle points will actually be found. One approach to resolving this difficulty is given by Westerberg and Floudas [49], who transform the equation-solving problem $\boldsymbol{\nabla}\mathcal{V} = \mathbf{0}$ into an equivalent optimization problem that has global minimizers corresponding to the solutions of the equation system (i.e., the stationary points of $\mathcal{V}$). A deterministic global optimization algorithm, based on a

branch-and-bound strategy with convex underestimators, is then used to find these global minimizers. Whether or not all stationary points are actually found depends on proper choice of a parameter (alpha) used in obtaining the convex underestimators, and Westerberg and Floudas do not use a method that guarantees a proper choice. However, there do exist techniques [1, 2], based on an interval representation of the Hessian, that in principle could be used to guarantee a proper value of alpha, though likely at considerable expense computationally. We demonstrate here an approach in which interval analysis is applied directly to the solution of $\boldsymbol{\nabla}\mathcal{V} = \mathbf{0}$ using an interval-Newton methodology. This provides a mathematical and computational *guarantee* that *all* stationary points of the potential energy surface will be found (or, more precisely, enclosed within an arbitrarily small interval).

### 5.1. Problem Formulation

Zeolites are materials in which $AlO_4$ and $SiO_4$ tetrahedra are the building blocks of a variety of complex porous structures characterized by interconnected cavities and channels of molecular dimensions [24]. Silicalite contains no aluminum and thus no cations; this has made it a common and convenient choice as a model zeolite system. The crystal structure of silicalite, well known from X-ray diffraction studies [38], forms a three-dimensional interconnected pore network through which a sorbate molecule can diffuse. In this work, the phase with orthorhombic symmetry is considered and a rigid lattice model, in which all silicon and oxygen atoms in the zeolite framework are occupying fixed positions and there is perfect crystallinity, is assumed. One spherical sorbate molecule (united atom) will be placed in the lattice, corresponding to infinitely dilute diffusion. The system is comprised of 27 unit cells, each of which is $20.07 \times 19.92 \times 13.42$Å with 96 silicon atoms and 192 oxygen atoms.

All interactions between the sorbate and the oxygen atoms of the lattice are treated atomistically with a truncated Lennard-Jones 6-12 potential. That is, for the interaction between the sorbate and oxygen atom $i$ the potential is given by

$$\mathcal{V}_i = \begin{cases} \frac{a}{r_i^{12}} - \frac{b}{r_i^6} & r_i < r_{\text{cut}} \\ 0 & r_i \geq r_{\text{cut}}, \end{cases} \tag{19}$$

where $a$ is a repulsion parameter, $b$ is an attraction parameter, $r_{\text{cut}}$ is the cutoff distance, and $r_i$ is the distance between the sorbate and oxygen atom $i$. This distance is given by

$$r_i^2 = (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2, \tag{20}$$

where $(x, y, z)$ are the Cartesian coordinates of the sorbate, and $(x_i, y_i, z_i), i = 1, \ldots, N$ are the Cartesian coordinates of the $N$ oxygen atoms. The silicon atoms, being recessed within the $SiO_4$ tetrahedra, are neglected in the potential function [27]. Therefore, the total potential energy, $\mathcal{V}$, of a single sorbate molecule in the absence of neighboring sorbate molecules is represented by a sum over all lattice oxygens,

$$\mathcal{V} = \sum_{i=1}^{N} \mathcal{V}_i. \tag{21}$$

The interval-Newton methodology will be applied to determine the sorbate locations $(x, y, z)$ that are stationary points on the potential energy surface $\mathcal{V}$ given by Eq. (21), that is, to solve the nonlinear equation system $\boldsymbol{\nabla}\mathcal{V} = \mathbf{0}$. To achieve tighter interval extensions of the potential function and its derivatives, and thus improve the performance of the interval-Newton method, the mathematical properties of the Lennard-Jones potential and its first- and second-order derivatives can be exploited, as described in detail by Lin and Stadtherr [30].

### 5.2. Results and Discussion

The interval-Newton methodology described above (LISS_LP) is now applied to find the stationary points of the potential energy surface $\mathcal{V}$ for the case of xenon as a sorbate in silicalite, as described by June *et al.* [23] Due to the orthorhombic symmetry of the silicalite lattice, the search space is only one asymmetric unit, $[0, 10.035] \times [0, 4.98] \times [0, 13.42]$Å, which is one-eighth of a unit cell. This defines the initial interval for the interval-Newton method, namely $X^{(0)} = [0, 10.035]$Å, $Y^{(0)} = [0, 4.98]$Å, and $Z^{(0)} = [0, 13.42]$Å. Following June *et al.* [23], stationary points with extremely high potential, such as $\mathcal{V} > 0$, will not be sought. To do this, we calculate the interval extension of $\mathcal{V}$ over the interval currently being tested, and if its lower bound is greater than zero, then the current interval is discarded. All computations were performed on a Dell workstation running a 1.7 GHz Intel Xeon processor under Linux.

Using the LISS_LP strategy for the interval-Newton method, a total of 15 stationary points were found in a computation time of 724 s. The locations of the stationary points, their energy value, and their type are listed in Table IV. Five local minima were found, along with 8 first-order saddle points and two second-order saddle points. June *et al.* [23] report the same five local minima, as well as 9 of the 10 saddle points. They do not report finding the lower energy second-order saddle point (saddle point #14 in Table IV).

For each first-order saddle point in Table IV, we followed June *et al.*'s method [23] to associate the saddle point with the transition state between two specific minima. The saddle point first was perturbed by $10^{-5}$Å in either direction along the eigenvector of the Hessian matrix associated with the negative eigenvalue. A steepest descent method using a step of 0.01Å was taken in the direction $-\boldsymbol{g}$. After 500 iterations, the steepest descent calculation was terminated and a Newton method was used to locate the minima connected through the saddle point. The results of these calculations are given in the rightmost column of Table IV. For example, the lowest energy saddle point (#6) can be viewed as connecting minima #1 and #3. In some cases the descent path from a saddle point led to a state outside the initial search box. Since the search box is one asymmetric unit, for each state found outside the search box, we can always find the equivalent state inside the search box through the symmetry operator and/or the periodic operator. In Table IV this is indicated by marking the state number with a prime. Thus, saddle point #7 connects minimum #2 with an equivalent point in a neighboring asymmetric unit. As expected, the results found for the states connected by the first-order saddle points is consistent with the analysis of June *et al.* [23]

Table IV. Stationary points of the potential energy surface of xenon in silicalite

| No. | Type | Energy(kcal/mol) | x(Å) | y(Å) | z(Å) | Connects |
|---|---|---|---|---|---|---|
| 1 | minimum | -5.9560 | 3.9956 | 4.9800 | 12.1340 | |
| 2 | minimum | -5.8763 | 0.3613 | 0.9260 | 6.1112 | |
| 3 | minimum | -5.8422 | 5.8529 | 4.9800 | 10.8790 | |
| 4 | minimum | -5.7455 | 1.4356 | 4.9800 | 11.5540 | |
| 5 | minimum | -5.1109 | 0.4642 | 4.9800 | 6.0635 | |
| 6 | 1st order | -5.7738 | 5.0486 | 4.9800 | 11.3210 | (1, 3) |
| 7 | 1st order | -5.6955 | 0.0000 | 0.0000 | 6.7100 | (2′, 2) |
| 8 | 1st order | -5.6060 | 2.3433 | 4.9800 | 11.4980 | (1, 4) |
| 9 | 1st order | -4.7494 | 0.1454 | 3.7957 | 6.4452 | (2, 5) |
| 10 | 1st order | -4.3057 | 9.2165 | 4.9800 | 11.0110 | (3, 4) |
| 11 | 1st order | -4.2380 | 0.0477 | 3.9147 | 8.3865 | (2, 4) |
| 12 | 1st order | -4.2261 | 8.6361 | 4.9800 | 12.8560 | (3, 5′) |
| 13 | 1st order | -4.1405 | 0.5925 | 4.9800 | 8.0122 | (4, 5) |
| 14 | 2nd order | -4.1404 | 0.5883 | 4.8777 | 8.0138 | (4,5),(4,4′) |
| 15 | 2nd order | -4.1027 | 9.1881 | 4.1629 | 11.8720 | (2,3),(4,5) |

A similar procedure was used on the two second-order saddle points, but using both negative eigenvalues. For example, in the case of saddle point #15, beginning with perturbations in either direction along the eigenvector associated with the most negative eigenvalue leads to a connection between minima #2 and #3. Repeating with the least negative eigenvalue leads to a connection between minima #4 and #5. Thus, this saddle point can be viewed as providing a crossconnection involving these four points. However, there are lower energy connections between all except #2 and #3. Though June *et al.* [23] do not identify this point as a second-order saddle, they do identify it as associating minima #2 and #3.

The second-order saddle point #14, not reported by June *et al.* [23], is very close to the first-order saddle point #13, and slightly lower in energy. Apparently neither of the two methods tried by June *et al.* [23] was able to locate this point. The first method they tried uses the same grid-based optimization scheme used to locate local minima in $\mathcal{V}$, but instead applied to minimize $\boldsymbol{g}^{\mathrm{T}}\boldsymbol{g}$. However, stationary points #13 and #14 are approximately 0.1Å apart, while the grid spacing they used was approximately 0.2Å. This illustrates the danger in using grid-based schemes for finding all solutions to a problem. By using the interval methodology described here, one never needs to be concerned about whether or not a grid spacing is fine enough to find all solutions. The second method they tried was Baker's algorithm [3], as described briefly above, but it is unclear how they initialized the algorithm. A key advantage of the interval method is that no point initialization is required. Only an initial *interval* must be supplied, here corresponding to one asymmetric unit, and this

is determined by the geometry of the zeolite lattice. Thus, in this context the interval methodology is initialization independent.

Lin and Stadtherr [30] have also studied two other sorbate-zeolite systems, and used the interval methodology to find all stationary points on the potential energy surfaces. While we have concentrated here on problems involving transition-state analysis of diffusion in zeolites, we anticipate that the methodology will be useful in many other types of problems in which transition-state theory is applied.

## 6. Concluding Remarks

We have demonstrated that the interval-Newton approach is a powerful, deterministic approach to the solution of a number of global optimization problems, as well as nonlinear equation solving problems, such as those that arise in chemical engineering and other areas of engineering and science. Problems with a very large number of local optima can be effectively solved, as can problems with a relatively large number of variables. Continuing improvements in methodology, together with advances in software and hardware will make this an increasingly attractive problem solving tool.

The validation provided by the interval approach comes at the expense of additional computation time. Essentially one has a choice between fast methods that may give the wrong answer, or a slower method that is guaranteed to give the correct answer. Thus, a modeler may need to consider the trade off between the additional computing time and the risk of getting the wrong answer to a problem. Certainly, for "mission critical" situations, the additional computing expense is well spent.

## Acknowledgements

This work has been supported in part by the National Science Foundation Grant EEC97-00537-CRCD, the Environmental Protection Agency Grant R826-734-01-0, the donors of The Petroleum Research Fund, administered by the ACS, under Grant 35979-AC9, and by the Indiana 21st Century Research & Technology Fund.

## References

1. Adjiman, C. S., I. P. Androulakis, and C. A. Floudas: 1998a, 'A Global Optimization Method, Alpha-BB, for General Twice-Differentiable Constrained NLPs – II. Implementation and Computational Results'. *Comput. Chem. Eng.* **22**, 1159.
2. Adjiman, C. S., S. Dallwig, C. A. Floudas, and A. Neumaier: 1998b, 'A Global Optimization Method, Alpha-BB, for General Twice-Differentiable Constrained NLPs – I. Theoretical Advances'. *Comput. Chem. Eng.* **22**, 1137.
3. Baker, J.: 1986, 'An Algorithm for The Location of Transition-States'. *J. Comput. Chem.* **7**, 385–395.

4. Brennecke, J. F. and E. J. Maginn: 2001, 'Ionic Liquids: Innovative Fluids for Chemical Processing'. *AIChE J.* **47**, 2384–2389.
5. Burgos-Solórzano, G. I., J. F. Brennecke, and M. A. Stadtherr: 2004, 'Validated Computing Approach for High-Pressure Chemical and Multiphase Equilibrium'. *Fluid Phase Equilib.* **219**, 245–255.
6. Doedel, E. J., A. R. Champneys, T. F. Fairgrieve, Y. A. Kuznetsov, B. Sandstede, and X. J. Wang: 1997, 'AUTO97: Continuation and Bifurcation Software for Ordinary Differential Equations'. Technical report, Department of Computer Science, Concordia Univeristy, Montreal, Canada.
7. Freemantle, M.: 2002, 'Meeting Briefs: Ionic Liquids Separated From Mixtures by $CO_2$'. *Chem. Eng. News* **80**(36), 44–45.
8. Gau, C.-Y., J. F. Brennecke, and M. A. Stadtherr: 2000, 'Reliable Parameter Estimation in VLE Modeling'. *Fluid Phase Equilib.* **168**, 1–18.
9. Gau, C.-Y. and M. A. Stadtherr: 2000, 'Reliable Nonlinear Parameter Estimation Using Interval Analysis Error-in-Variable Approach'. *Comput. Chem. Eng.* **24**, 631–638.
10. Gau, C.-Y. and M. A. Stadtherr: 2002a, 'Deterministic Global Optimization for Data Reconciliation and Parameter Estimation Using Error-in-Variables Approach'. In: R. Luus (ed.): *Optimization and Optimal Control in Chemical Engineering*. Trivandrum, India: Research Signpost.
11. Gau, C.-Y. and M. A. Stadtherr: 2002b, 'Deterministic Global Optimization for Error-in-Variables Parameter Estimation'. *AIChE J.* **48**, 1191–1197.
12. Gau, C.-Y. and M. A. Stadtherr: 2002c, 'New Interval Methodologies for Reliable Chemical Process Modeling'. *Comput. Chem. Eng.* **26**, 827–840.
13. Gmehling, J., U. Onken, and W. Arlt: 1977–1990, *Vapor-liquid Equilibrium Data Collection, Chemistry Data Series, Vol. I, Parts 1-8*. Frankfurt/Main, Germany: DECHEMA.
14. Gragnani, A., O. De Feo, and S. Rinaldi: 1998, 'Food Chains in the Chemostat: Relationships between Mean Yield and Complex Dynamics'. *B. Math. Biol.* **60**(4), 703–719.
15. Gwaltney, C. R. and M. A. Stadtherr: 2004, 'Reliable Computation of Equilibrium States and Bifurcations in Nonlinear Dynamics'. In: *Proceedings PARA'04 Workshop on State-of-the-art in Scientific Computing*. Lyngby, Denmark.
16. Gwaltney, C. R., M. P. Styczynski, and M. A. Stadtherr: 2004, 'Reliable Computation of Equilibrium States and Bifurcations in Food Chain Models'. *Comput. Chem. Eng.* in press.
17. Hansen, E. R.: 1992, *Global Optimization Using Interval Analysis*. New York: Marcel Dekker.
18. Hooke, R. and T. A. Jeeves: 1961, 'Direct Search Solution of Numerical and Statistical Problems'. *J. Assoc. Comput. Mach.* **8**, 212–229.
19. Hua, J. Z., J. F. Brennecke, and M. A. Stadtherr: 1998a, 'Enhanced Interval Analysis for Phase Stability Cubic Equation of State Models'. *Ind. Eng. Chem. Res.* **37**, 1519.
20. Hua, J. Z., J. F. Brennecke, and M. A. Stadtherr: 1998b, 'Reliable Computation of Phase Stability Using Interval Analysis Cubic Equation of State Models'. *Comput. Chem. Eng.* **22**, 1207.
21. Jastorff, B., R. St ormann, J. Ranke, K. M olter, F. Stock, and B. Oberheitmann: 2003, 'How Hazardous Are Ionic Liquids? Structure-Activity relationships and biological testing as important elements for sustainability evaluation'. *Green Chemistry* **5**, 136–142.
22. Jaulin, L., M. Kieffer, O. Didrit, and É Walter: 2001, *Applied Interval Analysis*. London: Springer-Verlag.
23. June, R. L., A. T. Bell, and D. N. Theodorou: 1991, 'Transition-State Studies of Xenon and $SF_6$ Diffusion in Silicalite'. *J. Phys. Chem.* **95**, 8866–8878.
24. Karger, J. and D. M. Ruthven: 1992, *Diffusion in Zeolites and Other Microporous Solids*. New York: Wiley.
25. Kaupe, A. F.: 1963, 'Algorithm 178 Direct Search'. *Commun. ACM* **6**, 313.
26. Kearfott, R. B.: 1996, *Rigorous Global Search: Continuous Problems*. Dordrect, The Netherlands: Kluwer Academic Publishers.
27. Kiselev, A. V., A. A. Lopatkin, and A. A. Shulga: 1985, 'Molecular Statistical Calculation of Gas Adsorption by Silicalite'. *Zeolites* **5**, 1508–1516.
28. Kuznetsov, Y. A.: 1998, *Elements of Applied Bifurcation Theory*. New York: Springer-Verlag.

29. Lin, Y. and M. A. Stadtherr: 2004a, 'Advances in Interval Methods for Deterministic Global Optimization in Chemical Engineering'. *J. Global Optim.* in press.
30. Lin, Y. and M. A. Stadtherr: 2004b, 'Locating Stationary Points of Sorbate-Zeolite Potential Energy Surfaces Using Interval Analysis'. *J. Chem. Phys.* submitted for publication.
31. Lin, Y. and M. A. Stadtherr: 2004c, 'LP Strategy for Interval-Newton Method in Deterministic Global Optimization'. *Ind. Eng. Chem. Res.* **43**, 3741–3749.
32. Maier, R. W., J. F. Brennecke, and M. A. Stadtherr: 1998, 'Reliable Computation of Homogeneous Azeotropes'. *AIChE J.* **44**, 1745.
33. Maier, R. W., J. F. Brennecke, and M. A. Stadtherr: 2000, 'Reliable Computation of Reactive Azeotropes'. *Comput. Chem. Eng.* **24**, 1851–1858.
34. Maier, R. W. and M. A. Stadtherr: 2001, 'Reliable Density-Functional-Theory Calculations of Adsorption in Nanoporous Materials'. *AIChE J.* **47**, 1874–1884.
35. McKinnon, K. I. M., C. G. Millar, and M. Mongeau: 1996, '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*. Dordrecht, The Netherlands: Kluwer Academic Publishers.
36. Moghadas, S. M. and A. B. Gumel: 2003, 'Dynamical and Numerical Analysis of a Generalized Food-chain Model'. *Appl. Math. Comput.* **142**(1), 35–49.
37. Neumaier, A.: 1990, *Interval Methods for Systems of Equations*. Cambridge, England: Cambridge University Press.
38. Olson, D. H., G. T. Kokotailo, S. L. Lawton, and W. M. Meier: 1981, 'Crystal Structure and Structure-Related Properties of ZSM-5'. *J. Phys. Chem.* **85**, 2238–2243.
39. Rohn, J. and V. Kreinovich: 1995, 'Computing Exact Componentwise Bounds on Solution of Linear Systems with Interval Data Is NP-Hard'. *SIAM J. Matrix. Anal.* **16**, 415–420.
40. Scurto, A. M., G. Xu, J. F. Brennecke, and M. A. Stadtherr: 2003, 'Phase Behavior and Reliable Computation of High-Pressure Solid-Fluid Equilibrium with Cosolvents'. *Ind. Eng. Chem. Res.* **42**, 6464–6475.
41. Siirola, J. D., S. Hauen, and A. W. Westerberg: 2002, 'Agent-based Strategies for Multiobjective Optimization'. AIChE Annual Meeting, Indianapolis, IN, Paper 265g.
42. Stadtherr, M. A., C. A. Schnepper, and J. F. Brennecke: 1995, 'Robust Phase Stability Analysis Using Interval Methods'. *AIChE Symp. Ser.* **91**(304), 356.
43. Stradi, B. A., J. F. Brennecke, J. P. Kohn, and M. A. Stadtherr: 2001, 'Reliable Computation of Mixture Critical Points'. *AIChE J.* **47**, 212–221.
44. Stradi, B. A., G. Xu, J. F. Brennecke, and M. A. Stadtherr: 2000, 'Modeling and Design of an Environmentally Benign Reaction Process'. *AIChE Symp. Ser.* **96**(323), 371–375.
45. Tessier, S. R., J. F. Brennecke, and M. A. Stadtherr: 2000, 'Reliable Phase Stability Analysis for Excess Gibbs Energy Models'. *Chem. Eng. Sci.* **55**, 1785.
46. Trefethen, N.: 2002, 'A Hundred-dollar Hundred-digit Challenge'. *SIAM News* **35**, 1.
47. Tsai, C. J. and K. D. Jordan: 1993, 'Use of An Eigenmode Method to Locate The Stationary-Points on The Potential-Energy Surfaces of Selected Argon And Water Clusters'. *J. Phys. Chem.* **97**, 11227–11237.
48. Ulas, S., U. M. Diwekar, and M. A. Stadtherr: 2004, 'Uncertainties in Parameter Estimation and Optimal Control in Batch Distillation'. *Comput. Chem. Eng.* submitted for publication.
49. Westerberg, K. M. and C. A. Floudas: 1999, 'Locating All Transition States and Studying the Reaction Pathways of Potential Energy Surfaces'. *J. Chem. Phys.* **110**, 9259–9295.
50. Xu, G., J. F. Brennecke, and M. A. Stadtherr: 2002, 'Reliable Computation of Phase Stability and Equilibrium from the SAFT Equation of State'. *Ind. Eng. Chem. Res.* **41**, 938–952.
51. Xu, G., A. M. Scurto, M. Castier, J. F. Brennecke, and M. A. Stadtherr: 2000, 'Reliable Computation of High Pressure Solid-Fluid Equilibrium'. *Ind. Eng. Chem. Res.* **39**, 1624–1636.

# A Computational Approach to Existence Verification and Construction of Robust QFT Controllers

P. S. V. Nataraj and Sachin Tharewal
*Systems and Control Engineering Group, Room 101, ACRE Building, Indian Institute of Technology Bombay, Mumbai 400076, India.*
*e-mail: Nataraj <nataraj@ee.iitb.ac.in>, Sachin <tharewal@iitb.ac.in>*

**Abstract.** Horowitz's quantitative feedback theory (QFT) (Horowitz, 1993) approach to robust control has been gaining popularity in the control literature for design of robust feedback systems. A central problem in QFT consists of proving the *existence* (or *non-existence*) of a QFT controller solution to a given design problem. In this paper, we propose a novel method based on interval analysis (Moore, 1979) to computationally verify the existence (or non-existence) of a controller solution, for a specified controller structure and an initial domain of controller parameter values. A feature of our proposed method is that it is a constructive existence method, i.e., if a solution of the specified structure exists for the given parameter domain, then all controller solutions lying in the domain are generated with our method. Essentially, the proposed method uses successive partitioning of the parameter domain and controller feasibility tests. We demonstrate the proposed method through a benchmark example.

**Keywords:** Control Synthesis, Interval Analysis, Quantitative Feeback Theory, Robust Control, Robust Synthesis.

## 1. Introduction

A versatile and practical engineering approach to the robust control problem is based on quantitative feedback theory (QFT) of Horowitz (Horowitz, 1991; Horowitz, 1993). The design is quantitative in the sense that the feedback is directly related to the amount of uncertainty and external disturbance. QFT has evolved with techniques to deal with single-input single-output (SISO) as well as multi-input multi-output (MIMO) cases, for linear and nonlinear, lumped and distributed parameter, time varying and time invariant systems. The QFT technique has been successfully applied to several practical problems with large plant uncertainty.

Consider a linear time invariant plant with parametric uncertainty given by $P(s,\lambda)$, where

$$\lambda = (\lambda_1, \ldots, \lambda_l)^T \in \mathbb{R}^l$$

is a plant parameter vector, which varies over a box $\boldsymbol{\lambda} = \left[\underline{\lambda}, \overline{\lambda}\right]$ consisting of two real column vectors $\underline{\lambda}$ and $\overline{\lambda}$ of length $l$ with $\underline{\lambda} \leq \overline{\lambda}$. This gives rise to the parametric plant family or set

$$\mathbb{P} \equiv \{P(s,\lambda) : \lambda \in \boldsymbol{\lambda}\}$$

with the nominal plant $P(s,\lambda_0)$ corresponding to an arbitrary nominal $\lambda_0 \in \boldsymbol{\lambda}$.

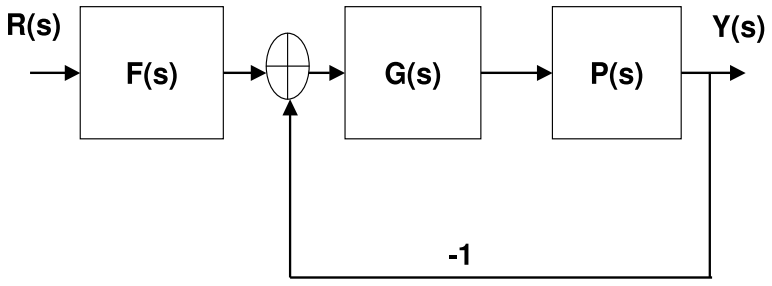



*Figure 1.* The two degree-of-freedom structure used in QFT

To achieve various specifications (specs), generally, the plant $P(s,\lambda)$ is embedded in the two degree-of-freedom feedback structure of QFT formulation as shown in Fig. 1, where $G(.)$ and $F(.)$ are transfer functions for the controller and prefilter, respectively. The controller $G(s,x)$ can be represented in the gain-pole-zero form as

$$G(s,x) = \frac{k_G \prod_{i=1}^{n_z} (s+\widetilde{z}_i) \prod_{i=1}^{n'_z} (s^2 + 2\zeta_i \upsilon_i s + \upsilon_i^2)}{\prod_{k=1}^{n_p} (s+p_k) \prod_{k=1}^{n'_p} \left(s^2 + 2\xi_k \vartheta_k s + \vartheta_k^2\right)} \tag{1}$$

where the controller parameter vector $x$ is

$$\begin{aligned} x = (\quad & k_G, \widetilde{z}_1, \ldots, \widetilde{z}_{n_z}, \zeta_1, \ldots, \zeta_{n'_z}, \upsilon_1, \ldots, \upsilon_{n'_z}, \\ & p_1, \ldots, p_{n_p}, \xi_1, \ldots, \xi_{n'_p}, \vartheta_1, \ldots, \vartheta_{n'_p}) \end{aligned} \tag{2}$$

The open loop transmission function is defined as

$$L(s,x,\lambda) = G(s,x)P(s,\lambda)$$

and the *nominal* open loop transmission function as

$$L_0(s,x) = G(s,x)P(s,\lambda_0)$$

The magnitude and angle functions of $L_0(s,x)$ are defined as

$$L_{0\_mag}(\omega,x) = |L_0(s=j\omega,x)|\,;\; L_{0\_ang}(\omega,x) = \angle L_0(s=j\omega,x) \tag{3}$$

Typically, following specifications are to be met for all $P(s,\lambda) \in \mathbb{P}$ and $\omega \in [0,\omega']$.

– Robust stability margin spec:

$$\left| \frac{L(j\omega,x,\lambda)}{1+L(j\omega,x,\lambda)} \right| \leq w_s \tag{4}$$

– Robust tracking performance spec:

$$|T_L(j\omega)| \leq \left| F(j\omega) \frac{L(j\omega, x, \lambda)}{1 + L(j\omega, x, \lambda)} \right| \leq |T_U(j\omega)| \tag{5}$$

– Robust input disturbance rejection performance spec:

$$\left| \frac{G(j\omega, x)}{1 + L(j\omega, x, \lambda))} \right| \leq w_{d_i}(\omega)$$

– Robust output disturbance rejection performance spec:

$$\left| \frac{1}{1 + L(j\omega, x, \lambda)} \right| \leq w_{d_o}(\omega)$$

The QFT design procedure begins with the generation of the plant *template*, which is nothing but the value set of plant at a design frequency $\omega$, given as

$$\mathcal{P}(\omega) = \{P(s = j\omega, \lambda) : \lambda \in \boldsymbol{\lambda}\}$$

This is followed by the QFT bound generation step. At each design frequency $\omega_i$, the plant template $\mathcal{P}(\omega_i)$ is used to translate the given performance and stability specs into regions in the Nichols chart where the nominal loop transmission $L_0(j\omega_i, x)$ is allowed to lie. The composition of all such bounds at $\omega_i$ is referred to as the *bound* on $L_0(j\omega_i, x)$ at $\omega_i$ and is denoted as $B(L_{0\_ang}(j\omega_i, x), \omega_i)$ or simply as $B(\omega_i)$. For example, the bounds at various $\omega_i$ are plotted in Fig. 4 along with the so-called universal high frequency bound (UHFB) valid for all $\omega \geq \omega_h$, where $\omega_h$ is some sufficiently "high" frequency. At any given $\omega_i$, the magnitude of the bound generally varies with the phase $L_{0\_ang}(j\omega_i, x)$; while some bounds are single-valued upper or lower bounds, the others are multiple-valued.

The objective of the QFT procedure is to synthesize $G(s, x)$ that satisfies the bounds $B(\omega_i)$ at all the design frequencies, and then synthesize a prefilter $F(s)$ which places the allowable variation in magnitude of the closed loop system, inside the respective tracking bounds. The details of the QFT design procedure can be found in (Horowitz, 1993).

A central problem in QFT consists of proving the *existence* of a controller solution to a given design problem. Any arbitrary design specs cannot be achieved by a specified controller transfer function structure, particularly for the plants with uncertainty. In certain cases, e.g., for non-minimum phase plants with uncertainty, one can analytically verify the non-existence of controller solution, as demonstrated by Horowitz (Horowitz, 1993). But this argument cannot be generalised, and a great deal of expertise would be required for commenting on the existence of the controller solution. This motivates us to propose a method to computationally verify the *existence* of a controller solution. In this paper, we propose a novel method based on interval analysis (Moore, 1979) to computationally verify the *existence* (or *non-existence*) of a controller solution, for a specified controller structure and an initial domain of controller parameter values.

## 2. Existence Verification

We propose an algorithm to computationally verify the existence of a controller solution for an uncertain plant transfer function, given certain performance and stability specs. The proposed existence verification method is constructive in its approach, i.e., if a solution of the specified structure exists for the given parameter domain, then *all* controller solutions lying in the domain are generated with the method.

Using the QFT formulation, the given specs are converted to the constraints satisfaction problem. The set of bounds $B(\omega_i)$ as described in section 1, gives rise to the set of constraints in the existence verification problem. The tracking and disturbance bounds at design frequency $\omega_i$ are to be respected, leading to the following type of constraints:

– single-valued upper bound :

$$h_i\left(x\right) = \left|B\left(L_{0\_ang}(j\omega_i, x), \omega_i\right)\right| - L_{0\_mag}(j\omega_i, x) \leq 0 \tag{6}$$

– single-valued lower bound :

$$h_i\left(x\right) = L_{0\_mag}(j\omega_i, x) - \left|B\left(L_{0\_ang}(j\omega_i, x), \omega_i\right)\right| \leq 0 \tag{7}$$

To ensure the nominal closed loop stability of the system, the nominal loop transmission is forced to lie on the right side of the respective stability bounds. Thus, the multiple valued stability bounds give rise to additional constraints of following type:

$$\begin{aligned} h_i^s\left(x\right) = \angle B(L_{0\_mag}(j\omega_i, x), \omega_i) - L_{0\_ang}(j\omega_i, x) \leq 0, \\ \text{for } L_{0\_mag}(j\omega_i, x) \in \left[\min\left|B(\omega_i)\right|, \max\left|B(\omega_i)\right|\right] \end{aligned} \tag{8}$$

With these bounds, the constraint satisfaction problem can be given as

$$\begin{aligned} \text{find all} \quad & x \in \mathbf{x} \\ \text{such that} \quad & H(x) \leq 0 \end{aligned} \tag{9}$$

where, $H\left(x\right) = \{h_i(x), h_i^s(x)\}$ is the set of bound constraints in (6,7) and (8), and $\mathbf{x}$ is the bound constrained controller parameter domain. The parameter domain is either user specified, or is constructed based on the given structure of the controller transfer function. To verify if the given constraints can be satisfied with a pre-specified controller structure, we suggest the following procedure to construct the controller parameter domain:

– The upper bound for the interval values of the corner frequency of the poles and zeros of the controller is set to $10^a\omega_h$, where, for instance, $a \approx 1$ or 2. This sets the cutoff frequency for the poles and zeros to a few decades beyond the high frequency $\omega_h$. The upper bound for the interval value of the high frequency gain of the controller can be set to a large value. To avoid the internal stability problem and the RHP pole/zero cancellation of the design, the lower bound for the interval values of the corner frequency of poles and zeros is set to zero.

Finding the solution over a bounded domain is as good as testing the feasibility of infinite combinations of controller parameter values. The most efficient and reliable techniques to do this are based on the rigorous search using interval analysis. Hence, we next present an algorithm based on interval analysis to solve this constraints satisfaction problem for existence verification of controller solution.

### 2.1. The Proposed Algorithm

The proposed algorithm uses successive partitioning of the given search domain, and the range inclosure property of the interval arithmetic

$$\text{range}\,(f, \mathbf{z}) \subseteq f\,(\mathbf{z})$$

where, $f\,(\mathbf{z})$ is the natural interval evaluation of a function $f$ over the box $\mathbf{z}$.

In this strategy of the proposed algorithm, at each iteration the controller parameter box $\mathbf{z}$, currently under process, is split into two subboxes, and tested for its feasibility. Any subbox which does not satisfy the constraints is discarded. The subbox which satisfy the constraints is added to a solution list $\mathcal{L}^{sol}$, and the remaining subbox(s) are added to a stack list $\mathcal{L}^{stack}$ for further processing. This process is recursively carried out till the stack list $\mathcal{L}^{stack}$ is exhausted (emptied), i.e., till the given search domain is completely processed.

The proposed algorithm essentially consists of five major components: a feasibility test, list handling, initialization, a termination criterion, and a bisection strategy.

The **Feasibility test** determines if a box $\mathbf{z}$ of controller parameter values satisfies the QFT bound constraints. Evaluation of the natural interval extensions of the nominal loop transmission magnitude and angle functions on $\mathbf{z}$ at some $\omega_i$ gives an angle-magnitude rectangle $\{L_{0\_ang}(j\omega_i, \mathbf{z}), L_{0\_mag}(j\omega_i, \mathbf{z})\}$, in the Nichols chart. This rectangle is called as the $L_0$ box at $\omega_i$. Based on the relative location of this rectangle w.r.t. the bound $B(\omega_i)$, the parameter box $\mathbf{z}$ is determined as *feasible*, *infeasible*, or *ambiguous* at $\omega_i$. The overall feasibility of box $\mathbf{z}$ is decided based on its feasibility at each of the design frequencies. A flag variable $flag_z$ represents the feasibility of box $\mathbf{z}$. The details for the feasibility test are given in sec. 2.1.1.

**List handling**: A stack list $\mathcal{L}^{stack}$ and a solution list $\mathcal{L}^{sol}$ is maintained to save the boxes generated during the partitioning process. The boxes which are determined as *feasible* are put into the solution list $\mathcal{L}^{sol}$, and the *ambiguous* boxes, which need further processing, are put into the stack list $\mathcal{L}^{stack}$. Since the whole stack list $\mathcal{L}^{stack}$ is to be processed, any box from this list can be picked up for further processing, but for convenience, the first box of the stack list $\mathcal{L}^{stack}$ is taken up as current box $\mathbf{y}$ for processing in the next iteration.

**Initialization** (step 1 in the algorithm): The current box under process denoted as $\mathbf{z}$ is set to the initial search box $\mathbf{z}^0$, and the feasibility test is done for $\mathbf{z}$. If $\mathbf{z}$ is *infeasible*, then by the inclusion property of interval analysis there is no feasible solution $\forall z \in \mathbf{z}$; hence, the algorithm exits, declaring that no feasible solution exists in the given initial search box. Else, a stack list $\mathcal{L}^{stack}$ is initialized with the box $\mathbf{z}$, and the solution list $\mathcal{L}^{sol}$ is initialized as an empty list.

**Termination** (step 2 in the algorithm): Since the objective is to find out all the solutions of the constraints satisfaction problem, the algorithm should terminate only after the given initial search domain is completely processed. The stack list $\mathcal{L}^{stack}$ holds the boxes (i.e., part of initial search domain) which are neither fully acceptable nor rejectable as controller solutions. Thus, the algorithm can terminate when all such boxes are completely processed or in other words when the stack list $\mathcal{L}^{stack}$ is emptied. The termination condition is given as

$$\mathcal{L}^{stack} = \varnothing \tag{10}$$

**Bisection** (step 4 in the algorithm): If the above termination condition is not met, i.e., box $\mathbf{z}$ is *ambiguous*, then $\mathbf{z}$ is split along the maximum width direction into two subboxes $\mathbf{v}^1$ and $\mathbf{v}^2$.

**Feasibility check for new subboxes** (step 5 in the algorithm): The feasibility check is performed on each of these two subboxes, any infeasible subbox(s) which does not satisfy the constraints is discarded and the feasible one is added to the solution list $\mathcal{L}^{sol}$.

The algorithm for existence verification of a controller solution based on the above described strategy is now presented.

**Inputs:** Numerical bound set, the design frequency set $\{\omega_i : i = 1, \cdots, n\}$, expressions for natural interval extensions $L_{0\_mag}(\omega, \mathbf{z})$ , $L_{0\_ang}(\omega, \mathbf{z})$ of the nominal loop transmission magnitude and angle functions in (3), and the initial search box $\mathbf{z}^0$.

**Output:** List of feasible controller parameters or a message "No feasible solution exists in the given initial search domain".

**BEGIN Algorithm**

1. Checking the feasibility of initial search box.

   a) Set the current box to the initial search box, i.e., set $\mathbf{z} = \mathbf{z}^0$.

   b) Call *Feasibility* Subroutine to determine if the current box $\mathbf{z}$ is completely infeasible, completely feasible, or an ambiguous case. The feasibility test returns a value for the variable $flag_z$.

   c) Initialization

      *i)* **IF** $flag_z$ = *infeasible* **THEN** print the message "No feasible solution exists in the given initial search domain", and **Exit** the algorithm.

      *ii)* **ELSE IF** $flag_z$ = *feasible* **THEN** print the message "The complete initial search domain is a feasible set of solution", and **Exit** the algorithm.

      *iii)* **ELSE** initialize the stack list $\mathcal{L}^{stack} \Leftarrow \{\mathbf{z}\}$ and initialize the solution $\mathcal{L}^{sol} \Leftarrow \{\}$. **END IF**

2. Choose the first box from the stack list $\mathcal{L}^{stack}$ as current box $\mathbf{z}$, and delete its entry from the stack list $\mathcal{L}^{stack}$.

3. Split the current box $\mathbf{z}$ in the maximum width direction to get two new subboxes $\mathbf{v}^1$ and $\mathbf{v}^2$, such that $\mathbf{z} = \mathbf{v}^1 \bigcup \mathbf{v}^2$ .

4. Call *Feasibility* Subroutine to determine the feasibility of each new subbox, and get the value of $flag_{v^1}$ and $flag_{v^2}$.

5. **DO** for $i = 1, 2$,

   a) **IF** $flag_{v^i} = infeasible$, **THEN** discard the subbox $\mathbf{v}^i$

   b) **ELSE IF** $flag_{v^i} = feasible$, **THEN** add the subbox $\mathbf{v}^i$ to the solution list $\mathcal{L}^{sol}$

   c) **ELSE** add $\mathbf{v}^i$ to the stack list $\mathcal{L}^{stack}$.
   **END IF**

   **END DO**

6. **IF** the termination condition given in (10) holds, **THEN**

   a) **IF** $\mathcal{L}^{sol} = \varnothing$, **THEN** print the message "No feasible solution exists in the given initial search domain" and **Exit** the algorithm.

   b) **ELSE IF** $\mathcal{L}^{sol} \neq \varnothing$, **THEN** print the message "The feasible solutions are:" $\mathcal{L}^{sol}$, and **Exit** the algorithm.
   **END IF**

   **END IF**

7. **Go to** step **6**.

**END Algorithm**

2.1.1. *Feasibility Subroutine*
This subroutine finds the feasibility of the controller parameter box $\mathbf{z}$, and returns the value of $flag_z$, which represents its feasibility.

**Inputs:** Numerical bound set, the design frequency set $\{\omega_i : i = 1, \cdots, n\}$, expressions for natural interval extensions $L_{0\_mag}(\omega, \mathbf{z})$ , $L_{0\_ang}(\omega, \mathbf{z})$ of the nominal loop transmission magnitude and angle functions in (3), and the parameter box $\mathbf{z}$.

**Output:** Value of $flag_z$.

**BEGIN Subroutine**

1. At every design frequencies $\omega_i$, $i = 1, \cdots, n$, do the following:

   a) Evaluate $L_{0\_mag}(\omega_i, \mathbf{z})$ and $L_{0\_ang}(\omega_i, \mathbf{z})$.

   b) For single valued upper bounds: Over the phase interval $L_{0\_ang}(\omega_i, \mathbf{z})$, find out the maximum and minimum magnitude value of the bound $B(\omega_i)$, and denote it as $B^{\max}_{mag}(\omega_i, \mathbf{z})$ and $B^{\min}_{mag}(\omega_i, \mathbf{z})$, respectively (Fig. 2 explains this notation).

   c) For multiple valued stability bounds:

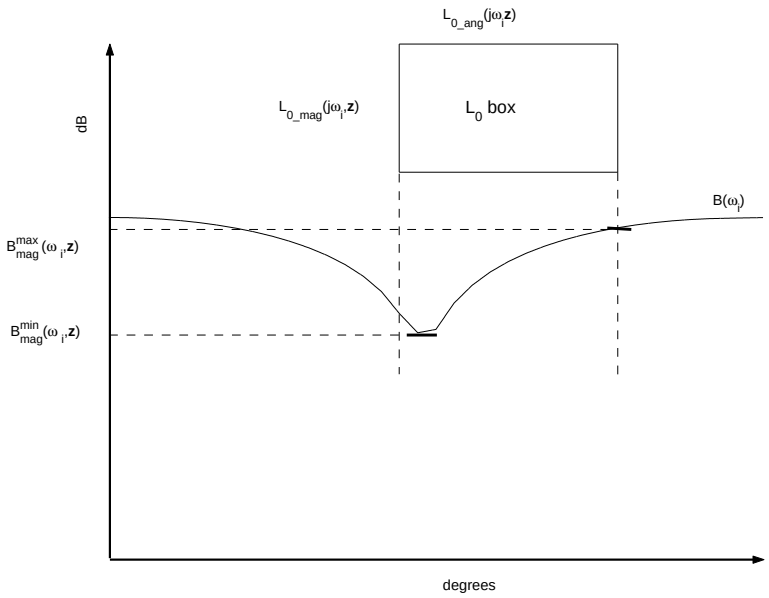



*Figure 2.* Definitions of $B^{\max}_{mag}(\omega_i, \mathbf{z})$ and $B^{\min}_{mag}(\omega_i, \mathbf{z})$.

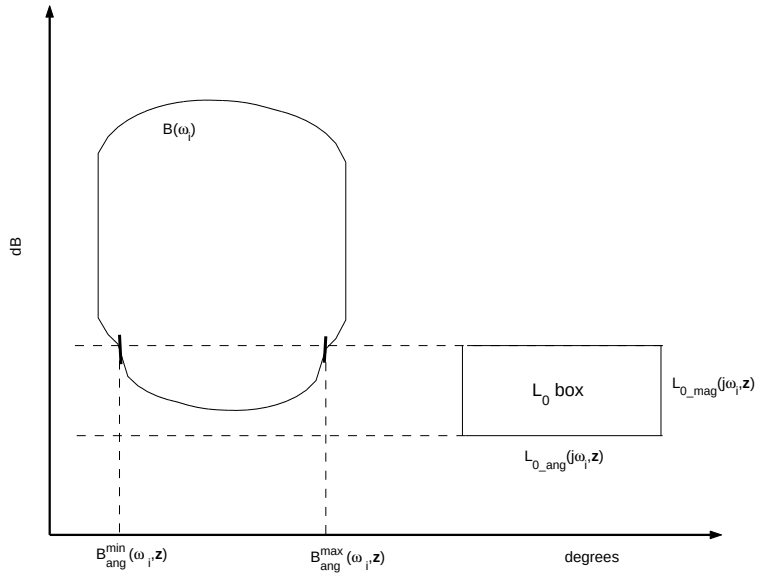



*Figure 3.* Definitions of $B^{\max}_{ang}(\omega_i, \mathbf{z})$ and $B^{\min}_{ang}(\omega_i, \mathbf{z})$

*i)* **IF** $L_{0\_mag}\left(j\omega_i, \mathbf{z}\right) \bigcap \left[\min\left|B(\omega_i)\right|, \max\left|B(\omega_i)\right|\right] \neq \varnothing$ **THEN**

A) Over the magnitude interval $L_{0\_mag}\left(\omega_i, \mathbf{z}\right)$, find out the maximum and minimum phase value of the bound $B(\omega_i)$, and denote it as $B^{\max}_{ang}(\omega_i, \mathbf{z})$ and $B^{\min}_{ang}(\omega_i, \mathbf{z})$, respectively (Fig. 3 explains this notation).

**END IF**

2. Set the feasibility flag as follows:

a) **IF** for all $\omega_i$, $i = 1, \cdots, n$,

$$\inf\{L_{0\_mag}\left(\omega_i, \mathbf{z}\right)\} \geqslant B^{\max}_{mag}(\omega_i, \mathbf{z})$$

**AND**

$$\begin{aligned}\Big\{L_{0\_mag}\left(j\omega_i, \mathbf{z}\right) \bigcap \quad & \left[\min\left|B(\omega_i)\right|, \max\left|B(\omega_i)\right|\right] = \varnothing \\ & \textbf{OR} \\ L_{0\_ang}\left(\omega_i, \mathbf{z}\right) \quad & \geqslant B^{\max}_{ang}(\omega_i, \mathbf{z})\Big\}\end{aligned}$$

**THEN** set the $flag_z$ = *feasible* and **RETURN**.

b) **ELSE IF** for any $\omega_i$, $i = 1, \cdots, n$,

$$\sup\{L_{0\_mag}\left(\omega_i, \mathbf{z}\right)\} \leqslant B^{\min}_{mag}(\omega_i, \mathbf{z})$$

**OR**

$$\begin{aligned}\Big\{L_{0\_mag}\left(j\omega_i, \mathbf{z}\right) \bigcap \quad & \left[\min\left|B(\omega_i)\right|, \max\left|B(\omega_i)\right|\right] \neq \varnothing \\ & \textbf{AND} \\ L_{0\_ang}\left(\omega_i, \mathbf{z}\right) \quad & \leqslant B^{\min}_{ang}(\omega_i, \mathbf{z})\Big\}\end{aligned}$$

**THEN** set the $flag_z$ = *infeasible* and **RETURN**.

c) **ELSE** set the $flag_z$ = *ambiguous* and **RETURN**.
**END IF**

**END Subroutine**

Thus, the feasibility subroutine returns $flag_z$ = *infeasible*, *feasible*, or *ambiguous*, depending on whether the parameter box $\mathbf{z}$ is completely infeasible, completely feasible, or an ambiguous case, respectively, w.r.t. the bound constraints.

REMARK 1. ***Convergence and Reliability:*** *The convergence of the proposed algorithm can be easily proved on the lines of interval branch and bound algorithms (*Ratschek and Rokne, 1988*) and the reliability of the algorithm immediately follows from the interval analysis techniques.*

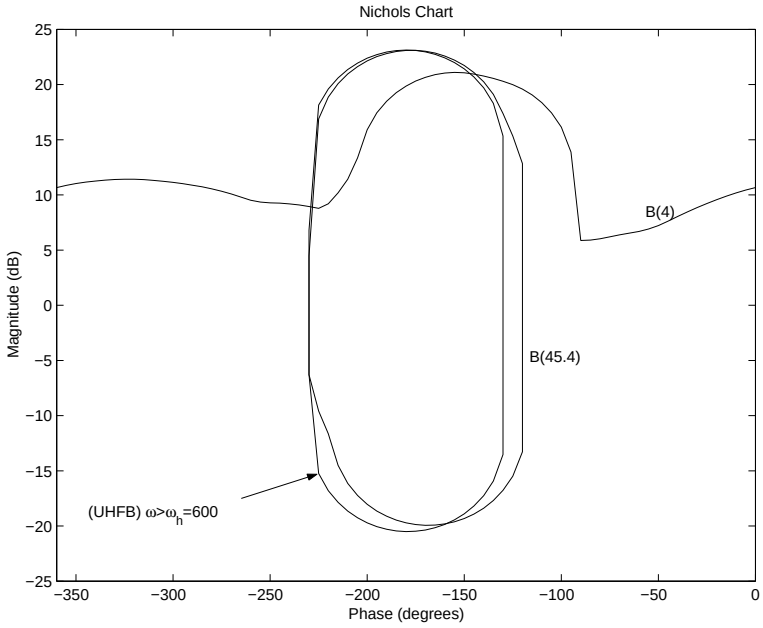



*Figure 4.* Bounds on $L_0$

## 3. Design Example

The proposed algorithm was tested on a QFT benchmark example, so that the results can be compared with that of an existing method. The example chosen is the design of robust controller for a non-minimum phase plant with uncertainty, given by Horowitz (Horowitz, 1993).

The uncertain plant transfer function is given as

$$P(s,\lambda) = \frac{k(1-\tau s)}{s(1+\beta s)} : k \in [1,3], \beta \in [0.3,1], \tau \in [0.05,0.1]. \tag{11}$$

The specs are:

- Robust stability margin spec (4): $w_s = 1.3032$
- Tracking spec (5): $|T_U(j4)| = 0.5dB$ and $|T_L(j4)| = -3.5dB$.

With just two design frequencies 4 and 45.4 rad/sec, using the Bode gain-phase relationship, Horowitz (Horowitz, 1993) showed analytically that, for the uncertain plant transfer (11), no controller solution exists for the above given specs.

We use the proposed algorithm to computationally verify the above finding of Horowitz, i.e., of the non-existence of a controller of first and second order for the above specs. We

choose the non-minimum phase plant

$$P_0(s) = \frac{(1 - 0.05s)}{s(1 + 0.3s)}$$

as the nominal case. The stability and tracking bounds for these specs are shown in Fig. 4. These bounds $B(\omega_i)$ at each design frequency $\omega_i$ are generated using the QFT toolbox (Borghesani et al., 1995). From the nature of the generated QFT templates, we find that the UHFB frequency $\omega_h \approx 600$ rad/sec. Based on this value of $\omega_h$, the upper bound on the pole/zero can be fixed as 6000 rad/sec. (see sec. 2), but instead we choose an arbitrarily large value of $10^4$ rad/sec. Moreover, using an arbitrarily large upper bound for the gain, the initial search box $\mathbf{z}^0$ is constructed as follows:

- For the first order controller, the parameter vector $z = \{k, \widetilde{z}_1, p_1\}$ is
$$\mathbf{z}^0 = (0, 10^8], (0, 10^4], (0, 10^4]$$
- For the second order controller, the parameter vector $z = \{k, \widetilde{z}_1, \widetilde{z}_2, p_1, p_2\}$ is
$$\mathbf{z}^0 = (0, 10^8], (0, 10^4], (0, 10^4], (0, 10^4], (0, 10^4]$$

For the aforementioned structures and the initial search domains, the proposed algorithm terminated with the message: "No feasible solution exists in the given initial search domain". Thus, this finding is in agreement with the analytically found '*non-existence*' of Horowitz mentioned above.

## 4. Conclusions

An algorithm has been proposed in this paper to computationally verify the *existence* (or *non-existence*) of a QFT controller solution, for a specified controller structure and an initial domain of controller parameter values. The proposed algorithm has been tested successfully on a QFT benchmark example for cross validating the results.

## Acknowledgment

The financial assistance provided by Council of Scientific and Industrial Research(CSIR), India to Sachin Tharewal for carrying out the research work as SRF is gratefully acknowledged.

## References

Borghesani, C., Y. Chait, and O. Yaniv: 1995, 'The QFT Control Design MATLAB Toolbox'. The MathWorks, Inc., MA, USA.

Horowitz, I. M.: 1991, 'Survey of quantitative feedback theory (QFT)'. *International Journal of Control* **53**(2), 255–291.
Horowitz, I. M.: 1993, *Quantitative feedback design theory (QFT)*. Boulder, Colorado: QFT Publications.
Moore, R. E.: 1979, *Methods and applications of interval analysis*. Philadelphia: SIAM.
Ratschek, H. and J. Rokne: 1988, *New computer methods for global optimization*. New York: Wiley.

# Efficient Method of Solution of Large Scale Engineering Problems with Interval Parameters

ANDRZEJ POWNUK
*Department of Civil Engineering, Silesian University of Technology,Gliwice, Poland,*
*pownuk@zeus.polsl.gliwice.pl, http://zeus.polsl.gliwice.pl/∼pownuk*

**Abstract.** In this paper sensitivity analysis method [4] and first order Taylor expansion method will be applied to solution of finite element equations of truss structures and non-stationary diffusion equation with interval parameters. Only linear-elastic model of truss structures is considered.

In order to calculate the interval solution (i.e. displacement vector $\mathbf{u}$ ) it is necessary to calculate derivative $\frac{\partial \mathbf{u}}{\partial \mathbf{h}}$. According to many numerical experiments and some theoretical results it is convenient to assume that in some engineering applications the function **u=u(h)** is monotone. Under such assumption it is possible to predict how to calculate the extreme solutions. Presented method gives quite accurate, however only approximate results.

Monotonicity assumption is not always true. Because of that the results are not always exact. The function **u=u(h)** is highly nonlinear, because of that presented algorithm is better than first order Taylor expansion. On the following web page [1] it is possible to compare presented algorithm, the exact results and the first order Taylor expansion using appropriate web applications.

**Keywords:** uncertainty, interval equations, truss structures

## 1. Introduction to interval FEM

Many engineering problems can be described by parameter dependent system of equations in the following form [5]:

$$\mathbf{K}(\mathbf{h})\,\mathbf{u} = \mathbf{Q}(\mathbf{h}) \tag{1}$$

where $\mathbf{K} \in R^{n\times n}$, $\mathbf{Q} \in R^n$, $\mathbf{u} \in R^n$, $\mathbf{h} \in R^m$. $\mathbf{h}$ is a vector of parameters of the structures (i.e. material characteristics, geometric characteristics, loads and other externals fields such as temperature. Very often we do not know the exact values of the parameters of the structure. Usually, if we do not know the exact values of the parameter $h_i$ it is possible to estimate an upper and lower bound such that:

$$h_i^- \le h_i \le h_i^+ \quad \text{for} \quad i = 1, \ldots, m \tag{2}$$

in general we can write:

$$\mathbf{h} \in \hat{\mathbf{h}} \subset R^m \tag{3}$$

where $\hat{\mathbf{h}} = \left[h_1^-, h_1^+\right] \times \left[h_2^-, h_2^+\right] \times ... \times [h_m^-, h_m^+]$. Presented method can be applied, when it is not possible to obtain probabilistic characteristic of the structure.

Exact solution set of the equation (1) is very complicated and can be defined in the following way:

$$\mathbf{u}\left(\hat{\mathbf{h}}\right) = \{\mathbf{h} : \mathbf{K}\left(\mathbf{h}\right)\mathbf{u} = \mathbf{Q}(\mathbf{h}), \mathbf{h} \in \hat{\mathbf{h}}\} \tag{4}$$

Due to high complexity of the set $\mathbf{u}\left(\hat{\mathbf{h}}\right)$ in applications we can only find the smallest interval $\hat{\mathbf{u}}\left(\hat{\mathbf{h}}\right)$ which contains the set $\mathbf{u}\left(\hat{\mathbf{h}}\right)$ i.e.

$$\hat{\mathbf{u}}\left(\hat{\mathbf{h}}\right) = hull\ \mathbf{u}\left(\hat{\mathbf{h}}\right) \tag{5}$$

One can call the set $\hat{\mathbf{u}}\left(\hat{\mathbf{h}}\right)$ the interval solution.

Now some selected methods of finding the interval solution will be presented.

## 2. Endpoints combinations method

According to many numerical examples [4, 2] very the endpoint combination method

$$u_i^- = min\{\ u_i\left(h_1^{\alpha_1}, h_2^{\alpha_2}, ..., h_m^{\alpha_m}\right) : \alpha_1, ..., \alpha_m \in \{-, +\}\} \tag{6}$$

$$u_i^+ = max\{\ u_i\left(h_1^{\alpha_1}, h_2^{\alpha_2}, ..., h_m^{\alpha_m}\right) : \alpha_1, ..., \alpha_m \in \{-, +\}\} \tag{7}$$

give very good approximation of the solution, particularly when the intervals are relatively narrow. In some cases the results are exact (for example in the case of system of linear interval equations).

## 3. First order Taylor expansion method

We can approximate the value of the nonlinear function $\mathbf{u} = \mathbf{u}\left(\mathbf{h}\right)$ by using first order Taylor expansion method:

$$\mathbf{u}\left(\mathbf{h}\right) \approx \mathbf{u}_L\left(\mathbf{h}\right) = \mathbf{u}\left(\mathbf{h}_0\right) + \frac{\partial \mathbf{u}\left(\mathbf{h}_0\right)}{\partial \mathbf{h}}\left(\mathbf{h} - \mathbf{h}_0\right) \tag{8}$$

$\mathbf{h}_0$is a mid point of the interval vector $\hat{\mathbf{h}}$ (i.e. $\mathbf{h}_0 = mid\left(\hat{\mathbf{h}}\right)$).

The vector $\mathbf{u}\left(\mathbf{h}_0\right)$ is a solution of the following system of linear equations:

$$\mathbf{K}\left(\mathbf{h}_0\right)\mathbf{u}\left(\mathbf{h}_0\right) = \mathbf{Q}\left(\mathbf{h}_0\right) \tag{9}$$

If one calculate derivative of the equation (9) it is possible to get the matrix $\frac{\partial \mathbf{u}(\mathbf{h}_0)}{\partial \mathbf{h}}$

$$\mathbf{K}\left(\mathbf{h}_0\right)\frac{\partial \mathbf{u}\left(\mathbf{h}_0\right)}{\partial h_i} = \frac{\partial \mathbf{Q}\left(\mathbf{h}_0\right)}{\partial h_i} - \frac{\partial \mathbf{K}\left(\mathbf{h}_0\right)}{\partial h_i}\mathbf{u}\left(\mathbf{h}_0\right), \quad i = 1, \ldots, m, \tag{10}$$

when parameters $\mathbf{h}$ belong to the interval vector$\hat{\mathbf{h}}$, then the extreme values of the function $\mathbf{u}_L(\mathbf{h})$ can be calculated by using natural interval extension [3]

$$\hat{\mathbf{u}}_L\left(\hat{\mathbf{h}}\right) = \mathbf{u}\left(\mathbf{h}_0\right) + \frac{\partial \mathbf{u}\left(\mathbf{h}_0\right)}{\partial \mathbf{h}}\left(\hat{\mathbf{h}} - \mathbf{h}_0\right). \tag{11}$$

In calculation we can calculate upper and lower bounds in the following way:

$$u_{i,L}^{-}\left(\hat{\mathbf{h}}\right) = u_i\left(\mathbf{h}_0\right) - \sum_{\alpha=1}^{m}\left|\frac{\partial u_i\left(\mathbf{h}_0\right)}{\partial h_\alpha}\right|\left(h_\alpha^{+} - h_{\alpha,0}\right), \tag{12}$$

$$u_{i,L}^{+}\left(\hat{\mathbf{h}}\right) = u_i\left(\mathbf{h}_0\right) + \sum_{\alpha=1}^{m}\left|\frac{\partial u_i\left(\mathbf{h}_0\right)}{\partial h_\alpha}\right|\left(h_\alpha^{+} - h_{\alpha,0}\right). \tag{13}$$

## 4. High order Taylor expansion method

In order to find better approximation of the exact solution one can apply the following algorithm.

**Algorithm 1**

1) Approximate the value of the function $\mathbf{u}$=$\mathbf{u}(\mathbf{h})$ by high order Taylor expansion $\mathbf{u} = \mathbf{u}_{ap}\left(\mathbf{h}\right)$.

2) Find the points $\mathbf{h}_1^*, \mathbf{h}_2^*, ..., \mathbf{h}_p^*$ where the approximate function $\mathbf{u}_{ap}\left(\mathbf{h}\right)$ has extreme values.

3) Calculate the values of the function $\mathbf{u}_{ap}\left(\mathbf{h}\right)$ in the points $\mathbf{h}_1^*, \mathbf{h}_2^*, ..., \mathbf{h}_p^*$.

4) Calculate the interval solution.

$$u_i^{-} = min\left\{\ u_{ap,i}\left(\mathbf{h}_1^*\right), u_{ap,i}\left(\mathbf{h}_2^*\right), ..., u_{ap,i}\left(\mathbf{h}_p^*\right)\right\}, i = 1, \ldots, n, \tag{14}$$

$$u_i^{+} = max\left\{\ u_{ap,i}\left(\mathbf{h}_1^*\right), u_{ap,i}\left(\mathbf{h}_2^*\right), ..., u_{ap,i}\left(\mathbf{h}_p^*\right)\right\}, i = 1, \ldots, n. \tag{15}$$

In general, instead of Taylor expansion it is possible to apply any approximation method.

## 5. Monotonicity assumption and first order sensitivity analysis method

In engineering applications the function $\mathbf{u}$=$\mathbf{u}(\mathbf{h})$ is very often monotone (usually for narrow intervals $\hat{h}_1, \hat{h}_2, ..., \hat{h}_m$). In that case it is possible to calculate the extreme values of the displacements $u_i$ by using appropriate endpoints of the intervals $\hat{h}_1, \hat{h}_2, ..., \hat{h}_m$. The algorithm of calculation is the following:

**Algorithm 2**

1) Calculate the mid point solution of the equation (9).
2) Calculate the derivatives of the solution in the mid point from the equation (4).
3) Calculate the sign vectors $\mathbf{S}_1, ..., \mathbf{S}_n$.

$$\mathbf{S}_i = \left[ \; sign\left(\frac{\partial u_i(\mathbf{h}_0)}{\partial h_1}\right) \;\; sign\left(\frac{\partial u_i(\mathbf{h}_0)}{\partial h_2}\right) \;\; ... \;\; sign\left(\frac{\partial u_i(\mathbf{h}_0)}{\partial h_m}\right) \; \right]. \tag{16}$$

4) Calculate the independent sign vectors.
Two sign vectors $\mathbf{S}_i, \mathbf{S}_j$ are independent if

$$\mathbf{S}_i \neq \mathbf{S}_j, \quad \mathbf{S}_i \neq (-1) \cdot \mathbf{S}_j, \tag{17}$$

Independent sign vectors will be denoted as $\mathbf{S}_1^*, ..., \mathbf{S}_{n^*}^*$, where $n^*$ is a number of independent sign vectors.
5) Calculate the extreme values of the solution for all independent sign vectors.

$$\mathbf{u}_i^{-,*} = \mathbf{u}\left(\mathbf{H}^-\left(\mathbf{S}_i^*, \hat{\mathbf{h}}\right)\right), \quad \mathbf{u}_i^{+,*} = \mathbf{u}\left(\mathbf{H}^+\left(\mathbf{S}_i^*, \hat{\mathbf{h}}\right)\right), \tag{18}$$

where

$$H_j^-\left(\mathbf{S}_i^*, \hat{\mathbf{h}}\right) = \begin{cases} h_j^-, \text{ if } S_{i,j}^* \geq 0 \\ h_j^+, \text{ if } S_{i,j}^* < 0 \end{cases}, \quad i = 1, \ldots, n, \tag{19}$$

$$H_j^+\left(\mathbf{S}_i^*, \hat{\mathbf{h}}\right) = \begin{cases} h_j^+, \text{ if } S_{i,j}^* \geq 0 \\ h_j^-, \text{ if } S_{i,j}^* < 0 \end{cases}, \quad i = 1, \ldots, n. \tag{20}$$

6) Calculate the interval solution $\hat{\mathbf{u}}$.

$$u_i^- = min\left\{ \; u_i\left(\mathbf{h}_1^-\right), ..., u_i\left(\mathbf{h}_{n^*}^-\right), u_i\left(\mathbf{h}_1^+\right), ..., u_i\left(\mathbf{h}_{n^*}^+\right) \right\}, \tag{21}$$

$$u_i^+ = max\left\{ \; u_i\left(\mathbf{h}_1^-\right), ..., u_i\left(\mathbf{h}_{n^*}^-\right), u_i\left(\mathbf{h}_1^+\right), ..., u_i\left(\mathbf{h}_{n^*}^+\right) \right\}, \tag{22}$$

where

$$\mathbf{h}_i^- = \mathbf{H}^-\left(\mathbf{S}_i^*, \hat{\mathbf{h}}\right), \quad \mathbf{h}_i^+ = \mathbf{H}^+\left(\mathbf{S}_i^*, \hat{\mathbf{h}}\right). \tag{23}$$

## 6. High order sensitivity analysis method

The first order sensitivity analysis method is based on the following general algorithm.

**Algorithm 3**

1) Approximate the value of the function by first order Taylor expansion.
2) Find the points $\mathbf{h}_1^*, \mathbf{h}_2^*, ..., \mathbf{h}_p^*$ where the Taylor expansion has extreme values.
3) Calculate the values of the exact solution in the points $\mathbf{h}_1^*, \mathbf{h}_2^*, ..., \mathbf{h}_p^*$.

4) Calculate the interval solution.

$$u_i^- = min\left\{ u_i\left(\mathbf{h}_1^*\right), u_i\left(\mathbf{h}_2^*\right), ..., u_i\left(\mathbf{h}_p^*\right)\right\}, \quad i = 1, \ldots, n, \tag{24}$$

$$u_i^+ = max\left\{ u_i\left(\mathbf{h}_1^*\right), u_i\left(\mathbf{h}_2^*\right), ..., u_i\left(\mathbf{h}_p^*\right)\right\}, \quad i = 1, \ldots, n. \tag{25}$$

In order to get better approximation of the exact solution one can approximate the function by high order Taylor expansion (or other approximation method).

**Algorithm 4**

1) Approximate the value of the function $\mathbf{u}$=$\mathbf{u}(\mathbf{h})$ by high order Taylor expansion.

2) Find the points $\mathbf{h}_1^*, \mathbf{h}_2^*, ..., \mathbf{h}_p^*$ where the approximate function (i.e. Taylor expansion) has extreme values.

3) Calculate the values of the exact solution in the points $\mathbf{h}_1^*, \mathbf{h}_2^*, ..., \mathbf{h}_p^*$.

4) Calculate the interval solution.

$$u_i^- = min\left\{ u_i\left(\mathbf{h}_1^*\right), u_i\left(\mathbf{h}_2^*\right), ..., u_i\left(\mathbf{h}_p^*\right)\right\}, \tag{26}$$

$$u_i^+ = max\left\{ u_i\left(\mathbf{h}_1^*\right), u_i\left(\mathbf{h}_2^*\right), ..., u_i\left(\mathbf{h}_p^*\right)\right\}. \tag{27}$$

In general, instead of Taylor expansion it is possible to apply any approximation method.

## 7. Comparison of First order Taylor expansion method and second order monotonicity test.

Let us consider a truss structure which is shown on Fig. 1.

In calculation the following numerical data was assumed P = 10 [kN], L = 1 [m], E = 210 [GPa] (Young modulus), A = 0.0025 $[m^2]$ (area of cross-section).

Accuracy of the first order sensitivity analysis method and first order Taylor expansion method is expressed by using the following numbers:

$$du_{\mathrm{i}}^- = \frac{\left(u_i^{mid} - u_i^-\right) - \left(u_i^{mid} - u_{i,exact}^-\right)}{u_i^{mid} - u_{i,exact}^-} \cdot 100\% \tag{28}$$

$$du_{\mathrm{i}}^+ = \frac{\left(u_i^+ - u_i^{mid}\right) - \left(u_{i,exact}^+ - u_i^{mid}\right)}{u_{i,exact}^+ - u_i^{mid}} \cdot 100\% \tag{29}$$

where $u_i^{mid}$ is a mid point solution, $u_{i,exact}^+$ is an exact value of upper bound, $u_{i,exact}^-$ is an exact value of upper bound.

Time of calculations is shown below. Calculation was done using computer with AMD Athlon XP 2600 with 512 MB RAM under RedHat Linux 9.0.

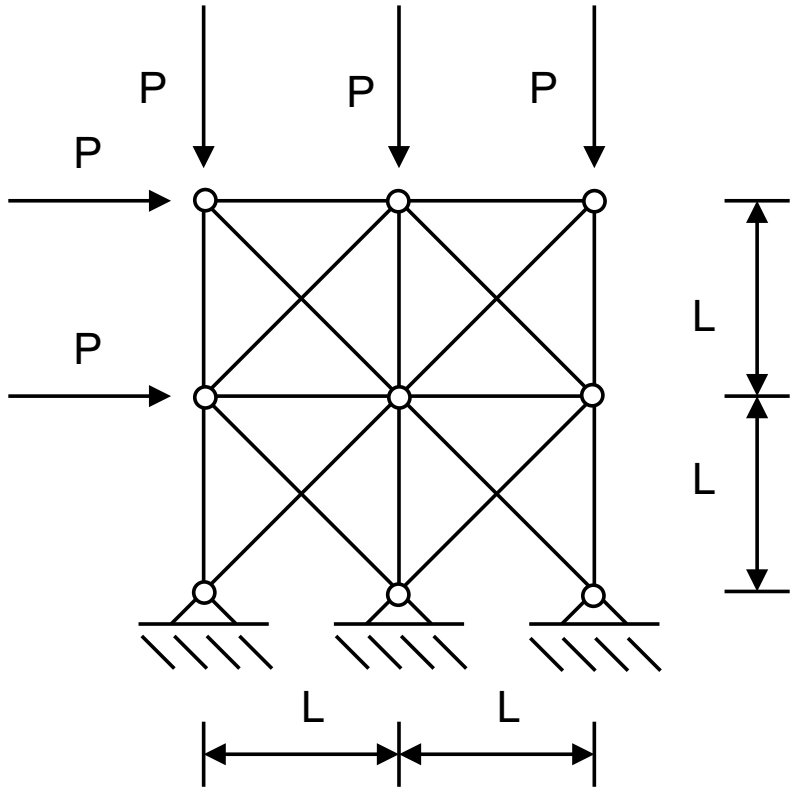



*Figure 1.* Truss structure

## 8. Interval solution of non-stationary diffusion equation

Let us consider non-stationary diffusion equation with interval parameters

$$\frac{\partial}{\partial x}\left(\beta\frac{k_x A_x}{\mu B}\frac{\partial p}{\partial x}\right)\Delta x+\frac{\partial}{\partial y}\left(\beta\frac{k_y A_y}{\mu B}\frac{\partial p}{\partial y}\right)\Delta y=\frac{V_b}{\alpha_c}\frac{\partial}{\partial t}\left(\frac{\phi}{B}\right) \tag{30}$$

where $k_x, A_x, k_y, A_y, \beta, \mu, B, \phi$ are some (interval) parameters, $p$ is the pressure of oil, $t$ is the time.

In order to get interval valued pressure of oil first order sensitivity analysis method was applied. The algorithm was implemented in C++ language using Borland C++ Builder compiler.

The program is able to take into account dependency of the parameters in different regions. The examples of the regions with different independent parameters are shown on Fig. (2).

Graphical representation of the interval solution (interval peruse of oil) for 7-th time step is shown in the Fig. (3).

Table I. Uncertainty of E and A 5%

| Sensitivity | | Taylor | |
|---|---|---|---|
| $du_i^-$ | $du_i^+$ | $du_i^-$ | $du_i^+$ |
| 0.00 | -0.50 | 5.34 | -5.50 |
| 0.28 | -0.37 | 5.03 | -5.14 |
| -1.31 | -0.07 | 2.64 | -4.36 |
| -2.72 | -2.82 | -6.49 | 0.76 |
| 1.45 | 0.67 | 6.36 | -4.31 |
| 1.11 | -1.43 | -3.42 | 3.10 |
| -1.12 | -0.34 | 3.15 | -4.72 |
| -0.19 | 0.00 | -0.60 | -0.47 |
| -0.43 | 0.00 | 3.41 | -4.52 |
| 0.21 | 0.00 | -3.95 | 3.95 |
| 1.36 | 0.00 | 5.78 | -4.61 |
| -1.12 | 0.00 | -5.74 | 4.57 |

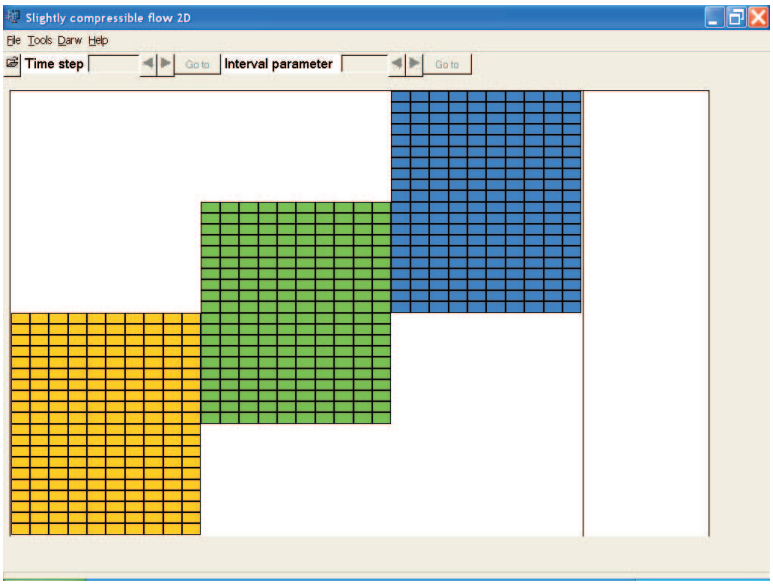



*Figure 2.* Graphical representation of dependences

Table II. Uncertainty of E and A 20%

| Sensitivity | | Taylor | |
|---|---|---|---|
| $du_i^-$ | $du_i^+$ | $du_i^-$ | $du_i^+$ |
| 0.00 | -1.37 | 18.67 | -20.32 |
| 0.00 | 0.00 | 18.35 | -19.30 |
| -0.39 | -0.08 | 14.88 | -17.56 |
| 0.00 | -0.03 | -16.16 | 13.61 |
| -0.18 | -0.18 | 18.84 | -19.80 |
| 0.00 | 0.00 | -18.48 | 17.32 |
| 0.00 | -0.76 | 16.06 | -18.63 |
| -0.34 | -5.24 | -4.57 | -7.82 |
| -1.66 | -1.13 | 14.22 | -19.03 |
| -0.06 | -1.94 | -17.27 | 12.69 |
| -0.31 | -0.93 | 16.35 | -19.22 |
| -0.48 | 0.00 | -19.07 | 17.91 |

In presented numerical example there were 10 interval parameters and 10 time steps. In presented example the system of parameter dependent equations (1) was generated by using finite difference method.

## 9. Conclusions

For truss structure:

1) Endpoint combination method gives exact results.

2) Endpoint combination method is very inefficient.

3) First order sensitivity analysis method is much more accurate than first order Taylor expansion method, particularly for large intervals.

4) Taylor expansion method is much more efficient than sensitivity analysis method.

5) The results of Taylor expansion method are acceptable for small intervals.

For diffusion equation example:

1) Sensitivity analysis method can be applied to solution of complex engineering problem.

Table III. Uncertainty of E and A 50%

| Sensitivity | | Taylor | |
|---|---|---|---|
| $du_i^-$ | $du_i^+$ | $du_i^-$ | $du_i^+$ |
| -0.03 | -1.19 | 43.01 | -48.34 |
| -37.10 | -0.39 | -11.27 | -46.95 |
| -1.53 | -0.24 | 28.41 | -44.04 |
| -0.25 | -4.30 | -41.91 | 21.75 |
| -0.29 | -0.28 | 43.11 | -47.35 |
| -0.33 | -0.04 | -45.43 | 38.26 |
| 0.00 | -1.97 | 31.88 | -45.78 |
| -13.59 | -15.68 | -32.33 | -30.86 |
| 0.00 | -1.21 | 25.84 | -46.34 |
| -0.34 | -7.88 | -43.72 | 19.25 |
| -1.68 | -2.03 | 29.28 | -46.56 |
| -1.70 | 0.00 | -46.31 | 40.78 |

Table IV. Endpoint combination method

| Number of interval parameters | Time of calculations in seconds |
|---|---|
| 10 | 0.02 |
| 15 | 1.86 |
| 18 | 17.18 |
| 20 | 124.69 |

2) It is using first order sensitivity analysis method possible to take into account different dependency of the parameters.

3) Parameter dependent system of equations (1) can be created using finite difference method.

Table V. First order sensitivity analysis

| Number of interval parameters | Time of calculations in seconds |
|---|---|
| 105 | 2 |
| 410 | 452 |
| 915 | 15208 |
| 1620 | 149554 |
| 2525 | 833782 |

Table VI. First order Taylor expansion

| Number of interval parameters | Time of calculations in seconds |
|---|---|
| 68 | 0.01 |
| 105 | 0.02 |
| 410 | 1.22 |
| 915 | 16.64 |
| 1314 | 50.04 |

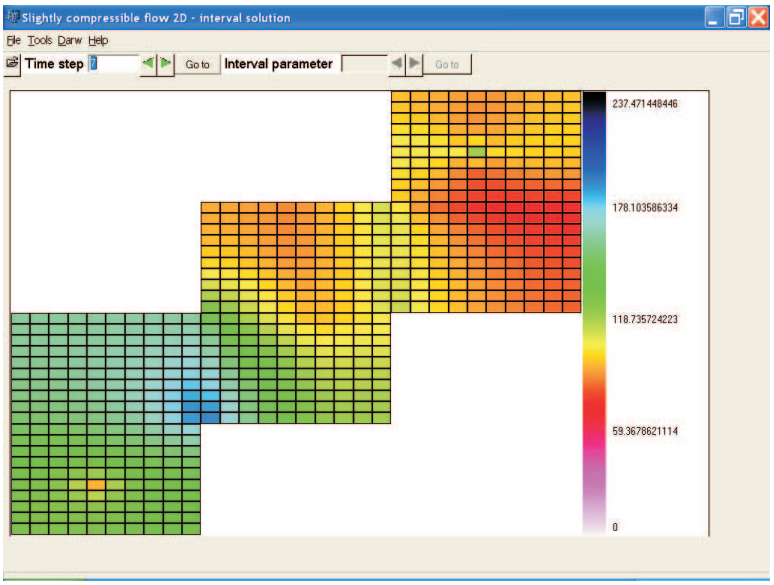



*Figure 3.* Interval pressure

## References


1. http://zeus.polsl.gliwice.pl/∼pownuk/interval_truss.htm.
2. McWilliam, S.: Anti-optimisation of uncertain structures using interval analysis, Computers and Structures, 79 (2001) pp. 421-430.
3. Neumaier, A.: *Interval methods for systems of equations*, Cambridge University Press, New York, 1990.
4. Pownuk, A.: *Numerical solutions of fuzzy partial differential equation and its application in computational mechanics*, in Fuzzy Partial Differential Equations and Relational Equations: Reservoir Characterization and Modeling (M. Nikravesh, L. Zadeh and V. Korotkikh, eds.), Studies in Fuzziness and Soft Computing, Physica-Verlag, 2004, pp. 308-347.
5. Zienkiewicz, O.C. and Taylor, R.L.: *The Finite Element Method Fifth edition, Volume 1: The Basis*, Butterworth-Heinemann, Oxford 2000.

# Buckling Analysis of Structures with Uncertain Properties and Loads Using an Interval Finite Element Method

Mehdi Modares
*Department of Civil Engineering, Case Western Reserve University, Cleveland, Ohio 44106*

Robert Mullen
*Department of Civil Engineering, Case Western Reserve University, Cleveland, Ohio 44106*
*e-mail*: robert.mullen@cwru.edu

Rafi L. Muhanna
*Department of Civil Engineering, Georgia Institute of Technology, Savannah, Georgia 31407*
*e-mail*: rafi.muhanna@gtsav.gatech.edu

Hao Zhang
*Department of Civil Engineering, Georgia Institute of Technology, Savannah, Georgia 31407*
*e-mail*: hao.zhang@ce.gatech.edu

**Abstract**. In order to ensure the safety of a structure, one must provide for adequate strength of structural elements. In addition, one must prevent large unstable deformations such as buckling. In most analyses of buckling, structural properties and applied loads are considered certain. This approach ignores the fact that imperfections and unknown changes in properties, albeit small, are required for onset of buckling. In this paper, we extend the interval finite element methods developed by the authors to solve for the possible values of loads that will result in a structural stability failure. The analysis requires that interval axial element forces in each frame element in a structure be calculated. These values are calculated from a linear system of interval equations resulting from the static structural analysis. Using the calculated axial loads, a subsequent interval eigenvalue problem is solved for the buckling loads. For both solutions of the linear system of equations and the eigenvalue problem, the unique properties of the finite element methods result in sharp solutions. Several structural problems are presented as exemplars. The sharpness of the solution is demonstrated by comparing to combinatorial solutions.

## 1. Introduction

In order to ensure the safety of a structure, one must provide for adequate strength of structural elements. In addition, one must prevent large unstable deformations known as buckling. In determining adequate strength as well as adequate stability, the finite element method has become the standard of practice for predicting a structure's behavior.
In current practice, uncertainty in system parameters is not considered during the analysis. Uncertainty is accounted for in a design by a combination of load amplification and strength reduction factor.

Such factors are based on probabilistic models of historic data. Thus, consideration of the impact of changing uncertainty on a design has been removed from current analysis tools.

In order to mitigate this problem, the authors among others (REF IFEM) have developed an interval based finite element method (IFEM). IFEM allows a structural analyst to calculate the impact of uncertainty in parameters on the structure's predicted behavior. To our knowledge, IFEM has only been applied to analysis addressing the strength criterion. In this paper, we extend IFEM to linear stability analysis of structures.

The method presented in this paper requires that interval internal element axial forces in each element in a structure be calculated. These values are calculated from a linear system of interval equations resulting from the static structural analysis. Using the calculated internal forces, a subsequent interval eigenvalue problem is formulated. The solution of the interval eigenvalue problem is then used to calculate the bounds on the critical buckling load. For both the solution of the linear system of equations and the eigenvalue problem, the unique properties of the finite element method are employed to achieve sharp results.

In the following, a brief review of IFEM for calculation of internal element forces is presented. Section 3 describes the formulation of the interval linear stability problem. In section 4, a method for calculation of exact bounds on the resulting eigenvalue problem is then given. An example problem is presented in section 5. Observations and conclusions are given in section 6.

## 2. Review of Static Interval Finite Element Methods

The linear stability analysis of structures requires the element forces to be determined as the first step in the analysis. For problems with interval values for the stiffness or loads, one needs an interval solution to the underlying statics problem. For the solution of interval finite element (IFEM) problems, Muhanna and Mullen (2001) introduced an Element-by-Element interval finite element formulation, in which a guaranteed enclosure for the solution of interval linear systems of equations was achieved. This method accounts for the parametric representation of element properties and a very sharp enclosure for the solution set due to loading, material and geometric uncertainty in solid mechanics problems. Element matrices were formulated based on the physics, and Lagrange multiplier or penalty methods were applied to impose the necessary constraints for compatibility and equilibrium.

For example, a two-element finite element construct is shown in figure (1). In this example, (E) is Young's modulus, (A) the cross-sectional area, and (L) the length of each element. Subscripts here indicate element number. Nodal loads are denoted by (P), and nodal displacements are denoted by (u).

The conventional finite element formulation results in a global stiffness matrix as given in Eq. (1)

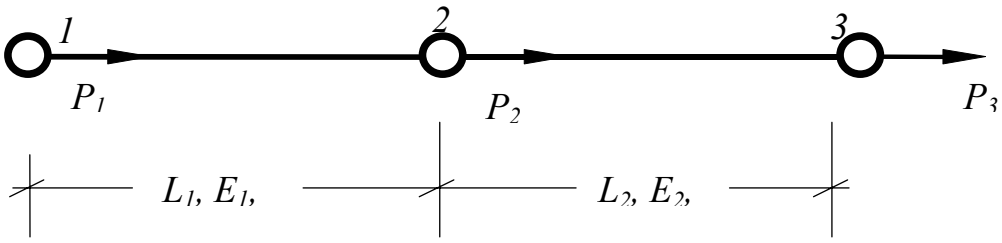



Figure (1) : Two connected linear truss elements.

$$\begin{pmatrix} \frac{E_1A_1}{L_1} & -\frac{E_1A_1}{L_1} & 0 \\ -\frac{E_1A_1}{L_1} & \frac{E_1A_1}{L_1}{}_1 + \frac{E_2A_2}{L_2} & -\frac{E_2A_2}{L_2} \\ 0 & -\frac{E_2A_2}{L_2} & \frac{E_2A_2}{L_2} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} \tag{1}$$

When the parameters E, A, or L are interval quantities, the resulting interval matrix allows for independent interval values for elements of this matrix which is not physically possible. The element-by-element method generates a global stiffness matrix in the form shown in Eq. (2).

$$K = \mathrm{DS} = \begin{pmatrix} \chi_1 & 0 & 0 & 0 \\ 0 & \chi_1 & 0 & 0 \\ 0 & 0 & \chi_2 & 0 \\ 0 & 0 & 0 & \chi_2 \end{pmatrix} \begin{pmatrix} \frac{E_1A_1}{L_1} & -\frac{E_1A_1}{L_1} & 0 & 0 \\ -\frac{E_1A_1}{L_1} & \frac{E_1A_1}{L_1} & 0 & 0 \\ 0 & 0 & \frac{E_2A_2}{L_2} & -\frac{E_2A_2}{L_2} \\ 0 & 0 & -\frac{E_2A_2}{L_2} & \frac{E_2A_2}{L_2} \end{pmatrix} \tag{2}$$

where $\chi_i$ is the interval multiplier of the $i$th finite element obtained due to uncertainty in $E_i$, $A_i$, and $l_i$. Such a form (i.e.***DS***) allows for factoring out the interval multiplier, resulting with an exact inverse for (***DS***).

To ensure compatibility (unique displacements for all elements connecting to a node), one adds constraint conditions in the form of Eq. (3). The resulting system of linear interval equations becomes Eq. (3) and (4)

Equation (1) can be introduced in the following equivalent form:

$$\widetilde{C}U = 0 \tag{3}$$

$$KU + \widetilde{C}^T\lambda = P \tag{4}$$

If we express $\boldsymbol{K}$ ( $n \times n$) in the form $\boldsymbol{D}\widetilde{\boldsymbol{S}}$ and substitute in equation (4):

$$D\widetilde{S}U = P - \widetilde{C}^T\lambda \tag{5}$$

where $\boldsymbol{D}(n \times n)$ is interval diagonal matrix, where its diagonal entries are the positive interval multipliers associated with each element, and $n$ is the multiplication of degrees of freedom per element and the number of elements in the structure. $\widetilde{\boldsymbol{S}}(n \times n)$ is a deterministic singular matrix (fixed point matrix). If we multiply equation (3) by $\boldsymbol{D}\widetilde{\boldsymbol{C}}^{\boldsymbol{T}}$ and add the result to equation (5), we get:

$$D(\widetilde{S}U + \widetilde{C}^T\widetilde{C}U) = (P - \widetilde{C}^T\lambda) \tag{6}$$

or:

$$D(\widetilde{S}U + \widetilde{Q}U) = (P - \widetilde{C}^T\lambda)$$
$$D(\widetilde{S} + \widetilde{Q})U = (P - \widetilde{C}^T\lambda)$$
$$D\widetilde{R}U = (P - \widetilde{C}^T\lambda) \quad (7)$$

where $(\widetilde{R})$ is a deterministic positive definite matrix, and the displacement vector $\boldsymbol{U}$ can be obtained from equation (7) in the following form:

$$U = \widetilde{R}^{-1}D^{-1}(P - \widetilde{C}^T\lambda) \quad (8)$$

where $(\widetilde{R}^{-1}D^{-1})$ is an exact inverse of the interval matrix $(D\widetilde{R})$. Equation (8) can be presented in the form:

$$U = \widetilde{R}^{-1}M\ \ \delta \quad (9)$$

Matrix (***M***) has the dimensions ($n$ × number of elements), and its derivation has been discussed in the previous works of the authors (Mullen and Muhanna 1999, Muhanna and Mullen 1999). The vector $\boldsymbol{\delta}$ is an interval vector that has the dimension of (number of elements × *1*) and its elements are the diagonal entries of $\boldsymbol{D}^{-1}$ with the difference that every interval value associated with an element is occurring only once.

If the interval vector $\boldsymbol{\lambda}$ can be determined exactly, the solution of Eq. (8) will represent an exact hull for the solution set of the general interval FE equilibrium equation.

More details on optimal implementation of the above concepts for static finite element solutions is presented in another paper in this proceedings. (Muhanna, Mullen and Zhang 2004).

## 3. Problem Definition

Deterministic Buckling Analysis*:*
As discussed in the previous section, the buckling analysis using the linear finite element method is carried out in two main steps. First, a parametric static analysis is performed using an arbitrary ordinate of applied load.

$$[Ke]\{u\} = \{P\} \quad (10)$$

The solution output includes the internal axial forces in terms of the load ordinate. Using these results, the geometric stiffness of the structure is developed which represents the pre-compression load's effects on the total stiffness of the structure (McGuire, Gallagher and Ziemian 2000).

Second, a generalized eigenvalue problem is performed between the elastic and geometric stiffness matrices of the structure in order to find the critical buckling loads in terms of the geometric and material characteristics of the structure.

$$([Ke] - \lambda[Kg])\{u\} = \{0\} \quad (11)$$

Buckling Analysis for Structures with Bounded Uncertainty*:*
For structures with bounded uncertainty present in the stiffness characteristics, the buckling analysis procedure requires the modifications on the following: First, the representation of stiffness characteristics must consider the presence of uncertainty using interval numbers. Second, the static analysis must be performed using the obtained interval stiffness matrix; hence, the calculated element axial forces are interval values. Third, using the obtained element interval axial forces, the interval geometric stiffness matrix can be established. Fourth, the interval eigenvalue problem must be solved in order to obtain the bounds on the critical buckling loads.

## I. Interval Representation of Uncertainty

Interval Number*:*
A real interval is a closed set of the form:

$$\widetilde{Z} = [z^l, z^u] = \{z \in \Re \mid z^l \le z \le z^u\} \tag{12}$$

In this work, the symbol (~) represents an interval quantity.

Interval Formulation:
The structure's global stiffness can be viewed as a summation of the element contributions to the global stiffness matrix:

$$[Ke] = \sum_{i=1}^{n} [L_i][Ke_i][L_i]^T \tag{13}$$

where $[L_i]$ is the element Boolean connectivity matrix and $[K_i]$ is the element stiffness matrix in the global coordinate system. Considering the presence of uncertainty in the stiffness properties, the non-deterministic element stiffness matrix is expressed as:

$$[\widetilde{Ke}_i] = ([l_i, u_i])[Ke_i] \tag{14}$$

in which $[l_i, u_i]$ is an interval number that pre-multiplies the deterministic element stiffness matrix. Therefore, the structure's global stiffness matrix in the presence of any uncertainty is the linear summation of the contributions of non-deterministic interval element stiffness matrices:

$$[\widetilde{Ke}] = \sum_{i=1}^{n} ([l_i, u_i])[L_i][Ke_i][L_i]^T = \sum_{i=1}^{n} ([l_i, u_i])[\overline{Ke}_i] \tag{15}$$

in which, $[\overline{Ke}_i]$ is the deterministic element stiffness contribution to the global stiffness matrix.

## II. Interval Geometric Stiffness Matrix

Using the obtained interval axial forces by IFEM (Section 2), the interval geometric stiffness matrix can be set up. The structure geometric stiffness can be viewed as a summation of the element contributions to the global geometric stiffness matrix:

$$[Kg] = \sum_{i=1}^{n} [L_i]((f_i)[\hat{K}g_i])[L_i]^T \tag{16}$$

where, ( $f_i$ ) is the element axial force and $[\hat{K}g_i]$ is the force independent matrix of geometric stiffness. Considering the axial force as an interval quantity, the interval structure's geometric stiffness matrix can be established as:

$$[\widetilde{K}g] = \sum_{i=1}^{n} (\widetilde{f}_i)[L_i][Kg_i][L_i]^T = \sum_{i=1}^{n} (\widetilde{f}_i)[\overline{K}g_i] \tag{17}$$

where $[\overline{K}g_i] = [L_i][\hat{K}g_i][L_i]^T$ and ( $\widetilde{f}_i = [f\min_i, f\max_i]$ ) is the element interval axial load.

## III. Interval Eigenvalue Problem for Buckling Analysis

Hollot and Barlett (1987) studied the spectra of eigenvalues of an interval matrix family which are found to depend on the spectrum of its extreme sets. Dief (1991) presented a method for computing interval eigenvalues of an interval matrix based on an assumption of invariance properties of eigenvectors.

The concept of interval eigenvalue problem has been used in structures with interval uncertainty. Modares and Mullen (2004) have introduced a method for the solution of the interval eigenvalue problem which determines the exact bounds of the natural frequencies of a structure using IFEM formulation.

In order to obtain the bounds on the critical buckling loads, a generalized interval eigenvalue problem must be performed between the interval elastic and interval geometric stiffness matrices as:

$$(\sum_{i=1}^{n} ([l_i, u_i])[\overline{K}e_i])\{u\} = (\widetilde{\lambda})(\sum_{i=1}^{n} (\widetilde{f}_i)[\overline{K}g_i])\{u\} \tag{18}$$

Interval Eigenvalue Problem Definition:
The eigenvalue problems for matrices containing interval values are known as the interval eigenvalue problems. If $[\widetilde{A}]$ is an interval matrix $(\widetilde{A} \in IR^{n\times n})$ and $[A]$ is a member of the interval matrix $(A \in \widetilde{A})$, the interval eigenvalue problem is shown as:

$$([A] - \lambda[I])\{u\} = 0, (A \in \widetilde{A}) \tag{19}$$

The solution of interest to the real interval eigenvalue problem is defined as an inclusive set of real values $(\widetilde{\lambda})$ such that for any member of the interval matrix, the solution to its eigenvalue problem is a member of the solution set shown as:

$$\{\lambda \in \widetilde{\lambda} = [\lambda^l, \lambda^u] \mid \forall A \in \widetilde{A} : ([A] - \lambda[I])\{u\} = 0\} \tag{20}$$

which is the enclosure of all possible solutions. A sharp enclosure is defined as the solution with the smallest radius as:

$$\{\exists \lambda \in \widetilde{\lambda} = [\lambda^l, \lambda^u] \,|\, \forall A \in \widetilde{A} : ([A] - \lambda[I])\{u\} = 0\} \tag{21}$$

## 4. Solution for Interval Eigenvalue Problem

The following concepts must be considered in order to bound the non-deterministic interval eigenvalue problem, Eq.(18).

The classical linear eigenpair problem for a symmetric matrix is:

$$Ax = \lambda x \tag{22}$$

with the solution of real eigenvalues ($\lambda_1 \le \lambda_2 \le ... \le \lambda_n$) and corresponding eigenvectors ($x_1, x_2, ..., x_n$). This equation can be transformed into a ratio of quadratics known as the *Rayleigh quotient*:

$$R(x) = \frac{x^T Ax}{x^T x} \tag{23}$$

The Rayleigh quotient for a symmetric matrix is bounded between the smallest and the largest eigenvalues (Bellman 1960 and Strang 1976).

$$\lambda_1 \le R(x) = \frac{x^T Ax}{x^T x} \le \lambda_n \tag{24}$$

Thus, the first eigenvalue ($\lambda_1$) can be obtained by performing an *unconstrained minimization* on the scalar-valued function of Rayleigh quotient:

$$\min_{x \in R^n} R(x) = \min_{x \in R^n} \left(\frac{x^T Ax}{x^T x}\right) = \lambda_1 \tag{25}$$

For finding the next eigenvalues, the concept of maximin characterization can be used. This concept obtains the k$^{th}$ eigenvalue by imposing (*k-1*) constraints on the minimization of the Rayleigh quotient:

$$\lambda_k = \max[\min R(x)] \tag{26}$$
$$\text{(subject to constrains } (x^T z_i = 0), i = 1, ...k-1, k \ge 2)$$

Bounding the Critical Buckling Loads:

Using the concepts of minimum and maximin characterizations of eigenvalues for symmetric matrices, the solution to the generalized interval eigenvalue problem for the critical buckling loads of a structure with uncertainty in the stiffness characteristics (Eq.(18)) for the first eigenvalue can be shown as:

$$\tilde{\lambda}_1 = \min_{x \in R^n}\left(\frac{x^T[\tilde{K}e]x}{x^T[\tilde{K}g]x}\right) = \min_{x \in R^n}\left(\frac{x^T\left(\sum_{i=1}^{n}([l_i, u_i])[\overline{K}e_i]\right)x}{x^T\left(\sum_{i=1}^{n}([f\min_i, f\max_i])[\overline{K}g_i]\right)x}\right) \tag{27}$$

for the next eigenvalues:

$$\tilde{\lambda}_k = \max\left[\min_{x.z_i=0, i=1,\ldots,k-1}\frac{x^T[\tilde{K}e]x}{x^T[\tilde{K}g]x}\right] = \max\left[\min_{x.z_i=0, i=1,\ldots,k-1}\left(\frac{x^T\left(\sum_{i=1}^{n}([l_i, u_i])[\overline{K}e_i]\right)x}{x^T\left(\sum_{i=1}^{n}([f\min_i, f\max_i])[\overline{K}g_i]\right)x}\right)\right] \tag{28}$$

Bounding Deterministic Eigenvalue Problems for the Critical Buckling Loads*:*

Since the matrices $[\overline{K}e_i]$ and $[\overline{K}g_i]$ are non-negative definite, the terms $(x^T(\overline{K}e_i)x)$ and $(x^T(\overline{K}g_i)x)$ are non-negative. Therefore, the upper bounds on the eigenvalues in Eqs.(18) and (19) are obtained by considering maximum values of interval coefficients of uncertainty for all elements in the elastic stiffness matrix and the lower values of axial force in the geometric stiffness matrix . Similarly, the lower bounds on the eigenvalues are obtained by considering minimum values of interval coefficients of uncertainty for all elements in the elastic stiffness matrix and the upper values of axial force in the geometric stiffness matrix.

Also, it can be observed that any other element stiffness selected from the interval sets will yield eigenvalues between the upper and lower bounds. Using these concepts, the deterministic eigenvalue problems corresponding to the maximum and minimum critical buckling loads are obtained as:

$$\left(\sum_{i=1}^{n}(u_i)[\overline{K}_i]\right)\{u\} = (\lambda_{\max})\sum_{i=1}^{n}(f\min_i)[\overline{K}g_i]\{u\} \tag{29}$$

$$\left(\sum_{i=1}^{n}(l_i)[\overline{K}_i]\right)\{u\} = (\lambda_{\min})\sum_{i=1}^{n}(f\max_i)[\overline{K}g_i]\{u\} \tag{30}$$

## 5. Example

The bounds on the critical buckling load for a 2D statically indeterminate truss with interval uncertainty present in the modulus of elasticity of each element are determined (Figure (2)).

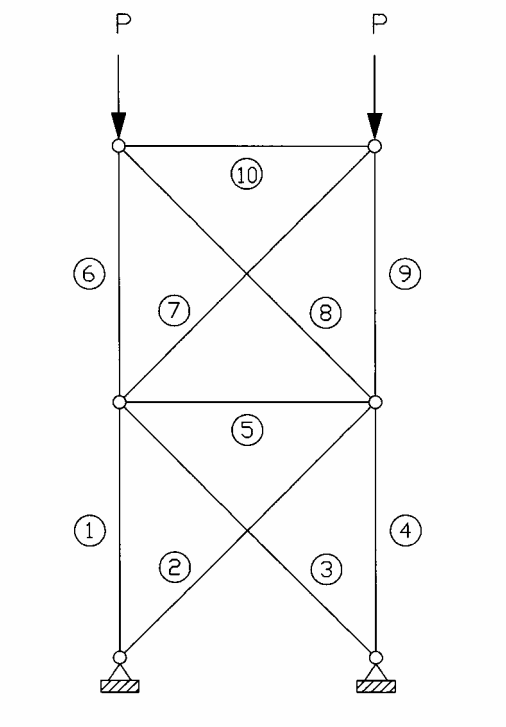



Figure (2): The structure of 2-D truss

The cross-sectional area $A$, the length for horizontal and vertical members $L$, the Young's moduli $E$ for all elements are $\widetilde{E} = ([0.99,1.01])E$.

The problem is solved using the method presented in this work. First, a static analysis on the structure with uncertainty is performed using IFEM, and the bounds on obtained element axial forces are obtained. Second, two deterministic eigenvalue problems are performed to obtain the bounds on the critical buckling load.

For comparison, a combinatorial analysis has performed which considers lower and upper values of uncertainty for each element i.e. solving ( $2^n = 2^{10} = 1024$ ) deterministic problems.

The static analysis results obtained by IFEM and the brute force combination solution are summarized in Table (1).

| | Lower Bound IFEM | Upper Bound IFEM | Lower Bound Combination Method | Upper Bound Combination Method |
|---|---|---|---|---|
| $f_1$ | -0.7943 | -0.7863 | -0.7945 | -0.7862 |
| $f_2$ | -0.3021 | -0.2908 | -0.3023 | -0.2905 |
| $f_3$ | -0.3021 | -0.2908 | -0.3023 | -0.2905 |
| $f_4$ | -0.7943 | -0.7863 | -0.7945 | -0.7862 |
| $f_5$ | 0.3887 | 0.4013 | 0.3882 | 0.4018 |
| $f_6$ | -0.8182 | -0.8108 | -0.8187 | -0.8104 |
| $f_7$ | -0.2674 | -0.2569 | -0.2679 | -0.2563 |
| $f_8$ | -0.2674 | -0.2569 | -0.2679 | -0.2563 |
| $f_9$ | -0.8182 | -0.8108 | -0.8187 | -0.8104 |
| $f_{10}$ | 0.1817 | 0.1891 | 0.1812 | 0.1895 |

Table (1): Static Analysis of the example problem using IFEM and combination method

Second, a buckling analysis is performed using the method presented in this work. Also, the solution to a combinatorial buckling analysis is obtained, and the results for the fundamental critical buckling load is summarized in Table (2).

| | Lower Bound Present Method | Lower Bound *with* Exact Forces | Lower Bound Combination Method | Relative Uncertainty Present Method | Relative Uncertainty Combination Method |
|---|---|---|---|---|---|
| $P_{cr1} / AE$ | 0.1080 | 0.1081 | 0.1093 | 0.021 | 0.010 |

Table (2): Buckling of the example problem using the present method and comparison with the combinatorial analysis results

In practice, the lowest buckling load is the only value of interest. As such, we have compared only the lower bound in Table 2. The example problem shows an overestimation of the width of the interval results of the proposed method compared to a combinatorial solution. The overestimation could be attributed to three possible sources: overestimation in the interval values

of the static solution of internal forces, overestimation in the eigenvalue solution or overestimation from uncoupling of the element forces Eq. (10) and the critical load Eq. (11). The internal forces calculated by the interval method and the exact combinatorial results are correct in the first three digits. The eigen solution has been proved to be sharp. Therefore the uncoupling of the static solution from Eq. (10) and the stability equation (11), is the most likely cause of the overestimation of the width seen in the solution. This can be seen by examining a solution where the exact combinatorial internal forces are used in Eq. (11) to find the critical buckling load. In this calculation, the critical load in (0.1081), just slightly above the results from the proposed method of 0.1080 (See Table 2).

## 6. Discussion

In this paper, we have introduced a method for linear stability analysis of a structure with stiffness properties expressed as an interval quantity. To our knowledge, this is the first treatment of interval methods for structural stability. The conventional two step method consisting of solving the linear static problem for internal forces and subsequent solution of an eigen problem for the critical buckling load has been adapted from the non-interval approach. The method presented provides a lower bound for the minimum buckling load. Dependency of the interval internal forces and interval stiffness parameters have not been included in the method; this is the expected cause of loss of sharpness in the interval results.

## References


1. Bellman, Richard, 1960, *Introduction to Matrix Analysis*, McGrawHill.
2. Dief, A.,1991, *Advanced Matrix theory for Scientists and Engineers*,pp.262-281. Abacus Press (1991)
3. Hollot, C. and Bartlett, A., " On the eigenvalues of interval matrices", Technical Report, Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA (1987).
4. McGuire, W. , Gallagher, R. and Ziemian ,R. 2000. "Matrix Structural Analysis", John Wiley & Sons.
5. Muhanna, Rafi L. & Mullen, Robert L., 2001. "Uncertainty in Mechanics Problems-Interval-Based Approach". Journal of Engineering Mechanics June-2001, pp.557-566.
6. Modares, Mehdi and Mullen, Robert L., 2004. " Free Vibration of Structures with Interval Uncertainty ". 9th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability.
7. Muhanna, Rafi L., Mullen, Robert L., Zhang, Hao, 2004 "Interval Finite Elements as a Basis for Generalized Models of Uncertainty in Engineering Mechanics". NSF workshop on Reliable Engineering Computing
8. Strang, Gilbert, 1976." Linear Algebra and its Applications ", Massachusetts Institute of Technology

# Experiments with Range Computations using Extrapolation

P. S. V. Nataraj and Shanta Sondur
*Systems and Control Engineering Group, Room No. 101, ACRE Building, Indian Institute of Technology, Bombay, MUMBAI 400 0076, INDIA.*
*e-mail: Nataraj<nataraj@sc.iitb.ac.in>,Shanta<shanta@ee.iitb.ac.in>*

**Abstract.** The *natural interval extension* (NIE) used widely in interval analysis has the first order convergence property, i.e., the excess width of the range enclosures obtained with the NIE goes down at least *linearly* with the domain width. Here, we show how range approximations of *higher convergence orders* can be obtained from the sequence of range enclosures generated with the NIE and uniform subdivision. We combine the well-known *Richardson Extrapolation Process* (Sidi, 2003) with *Brezenski's error control method* (Brezenski, 1983) to generate non-validated range approximations to the true range. We demonstrate the proposed method for accelerating the convergence orders on several multidimensional examples, varying from one to six dimensions. These numerical experiments also show that considerable computational savings can be obtained with the proposed procedure. However, the theoretical basis of the proposed method remains to be investigated.

**Keywords:** Extrapolation, NIE, REP, Acceleration of Sequences.

## 1. Introduction

A major focus of interval analysis (Moore, 1979) is developing interval algorithms which produce sharp bounds on the solutions. The *natural interval extension* (NIE) is the simplest tool that is widely used in interval analysis to compute the range enclosures of functions. The range enclosures obtained using NIE can be tightened further with the help of the uniform subdivision method (Moore, 1979). These range enclosures possesses the property of first order convergence, i.e., the excess width of the computed range enclosures goes down at least linearly with the domain width.

In this paper, we propose a new method to accelerate the convergence rate of the range enclosures, obtained with the NIE and uniform subdivision, using an *extrapolation* process, such as the Richardson extrapolation process (REP). In the proposed method, we first obtain the range enclosures with the NIE and uniform subdivision, for a sequence of geometrically increasing subdivision factors. Then, we construct two separate sequences of lower and upper bounds from the obtained range enclosures. Next, we extrapolate these sequences to their respective limits (which are the range infimum and range supremum) using the REP. This produces the Romberg Tables for the range infimum and supremum. To these Romberg Tables, we next apply Brezenski's error control criterion and generate the so-called Brezenski's tables of intervals containing the range infimum and supremum. Finally, from the Brezenski's tables, we construct a table of intervals approximating the range. The sequences of the range approximating intervals in this table converges columnwise increasingly faster than the sequence of range enclosures obtained with the existing method of NIE and uniform subdivision.

The outline of the paper is as follows. In section 2, we discuss the basics of the sequence transformation, the REP and Brezenski's error control criterion. In section 3, we review the NIE and uniform subdivision. In section 4, we present the proposed algorithms. In section 5, we demonstrate the effectiveness of the proposed method on several multidimensional examples, varying from one to six dimensions. Finally, in section 6 we draw the conclusions of the work.

## 2. Extrapolation Process - Sequence Transformation

Extrapolation methods (equivalently, *convergence acceleration methods* or *sequence transformations*) are popularly used for accelerating the convergence process of sequences (Brezenski and Zaglia, 2002). Extrapolation methods basically transform the original sequence into another one which converges to the limit more quickly (when the limit exists).

Let $(S_n)$ be a sequence of (real or complex) numbers which converges to the limit $S$ and $(T_n)$ be another sequence obtained by transforming the sequence $(S_n)$ using some suitable transformation method $T$.

In order to obtain a higher rate of convergence, the new transformed sequence $(T_n)$ must exhibit the following properties:

1. $(T_n)$ must converge.
2. $(T_n)$ must converge to the same limit as $(S_n)$.
3. $(T_n)$ must converge to $S$ faster than $(S_n)$, that is

$$\lim_{n\to\infty} (T_n - S) / (S_n - S) = 0$$

If the new sequence $(T_n)$ possesses property (3), we say that the transformation $T$ *accelerates the convergence* of the sequence $(S_n)$ or that the sequence $(T_n)$ *converges faster* than $(S_n)$.

These properties, in general, do not hold for all converging sequences $(S_n)$. We can obtain the new transformed sequence $(T_n)$ possessing the above mentioned properties only if the sequence $(S_n)$ to be accelerated belongs to the kernel $\mathfrak{K}_T$ of the transformation used (the kernel $\mathfrak{K}_T$ is the set of sequences for which there exists an $S$ such that $\forall n \geq N$, $T_n = S$, cf. (Brezenski and Zaglia, 2002)).

For instance, amongst the wide range of transformation methods available, the well-known Aitken's $\Delta^2$ transformation process is given by

$$T_n = S_n - \frac{(S_{n+1} - S_n)^2}{S_{n+2} - 2S_{n+1} + S_n},\ n = 0, 1, \ldots \tag{1}$$

For the Aitken's process, the kernel $\mathfrak{K}_T$ is the set of sequences of the form

$$S_n = S + a\lambda^n \tag{2}$$

where, $a$ and $\lambda$ are scalars with $a \neq 0$ and $\lambda \neq 1$. Usually, $S$ is the limit of the sequence $(S_n)$, but this is not always the case. In the Aitken's process, $S$ is the *limit* of $(S_n)$ if $|\lambda| < 1$, and is called the *anti-limit* if $|\lambda| > 1$. It can be shown (Brezenski and Zaglia, 2002) that the Aitken's process accelerates the convergence of all sequences for which there exists a $\lambda \in [-1\ , +1)$ such that

$$\lim_{n\to\infty} \frac{(S_{n+1} - S)}{(S_n - S)} = \lambda$$

A sequence transformation $T : (S_n) \to (T_n)$ is said to be an extrapolation method if it is such that $\forall n \geq N$, $T_n = S$ if and only if $(S_n) \in \mathfrak{K}_T$. Thus, any sequence transformation can be viewed as an extrapolation method.

Amongst the various extrapolation methods (Sidi, 2003), perhaps the most popular and widely used method is the REP. Let $K \in \mathbb{N}$, $\rho \geq 2$, and $\{S_n\}$ be the sequence to be accelerated. The REP can be given as

$$T_0^{(j)} = S_j,\ j = 0, 1, \ldots, K \tag{3}$$

$$T_k^{(j)} = T_{k-1}^{(j)} + \frac{\left(T_{k-1}^{(j)} - T_{k-1}^{(j-1)}\right)}{(\rho^k - 1)},\ \begin{cases} k = 1, 2, \ldots, K, \\ j = k, \ldots, K. \end{cases} \tag{4}$$

which is similar to the Aitken's $\Delta^2$ process for the first extrapolated column $k = 1$ as given in (1).

The sequences $\left\{T_k^{(j)}\right\}$ computed using (4) can be arranged in a two-dimensional array called the Romberg Table, denoted $[T]^k$, cf. Table 1. The arrows in the table show the flow of computations. The $k^{th}$ column of the Romberg Table is referred to as the $(k-1)^{th}$ *extrapolated* column. Details of the REP are in (Sidi, 2003).

Table 1. The Romberg Table, $[T]^K$ with $K = 5$ (i.e., with 5 extrapolated columns)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $T_0^{(0)}$ | | | | | | | | | |
| | $\searrow$ | | | | | | | | |
| $T_0^{(1)}$ | $\rightarrow$ | $T_1^{(1)}$ | | | | | | | |
| | $\searrow$ | | $\searrow$ | | | | | | |
| $T_0^{(2)}$ | $\rightarrow$ | $T_1^{(2)}$ | $\rightarrow$ | $T_2^{(2)}$ | | | | | |
| | $\searrow$ | | $\searrow$ | | $\searrow$ | | | | |
| $T_0^{(3)}$ | $\rightarrow$ | $T_1^{(3)}$ | $\rightarrow$ | $T_2^{(3)}$ | $\rightarrow$ | $T_3^{(3)}$ | | | |
| | $\searrow$ | | $\searrow$ | | $\searrow$ | | $\searrow$ | | |
| $T_0^{(4)}$ | $\rightarrow$ | $T_1^{(4)}$ | $\rightarrow$ | $T_2^{(4)}$ | $\rightarrow$ | $T_3^{(4)}$ | $\rightarrow$ | $T_4^{(4)}$ | |
| | $\searrow$ | | $\searrow$ | | $\searrow$ | | $\searrow$ | | $\searrow$ |
| $T_0^{(5)}$ | $\rightarrow$ | $T_1^{(5)}$ | $\rightarrow$ | $T_2^{(5)}$ | $\rightarrow$ | $T_3^{(5)}$ | $\rightarrow$ | $T_4^{(5)}$ | $\rightarrow$ $T_5^{(5)}$ |

## 2.1. Brezenski's Error Control criterion

Brezenski's theorem on error control (Brezenski, 1983) explains how to construct a sequence of intervals containing the unknown limit of the sequence under consideration.

Let $\{S_n\}$ be the sequence under consideration. Let $S$ be the limit of the sequence $\{S_n\}$. Let $\{T_n\}$ and $\{V_n\}$ be two other sequences obtained by applying REP to $\{S_n\}$. Suppose the sequence $\{T_n\}$ converges faster than $\{S_n\}$, and $\{V_n\}$ converges faster than $\{T_n\}$, both to the same limit $S$. Thus, $\{S_n\}$, $\{T_n\}$, and $\{V_n\}$ can be successive columns of the Romberg Table 1.

Let $b \in R$ (called as the Brezenski's factor). Define

$$V_n(b) = V_n - b(V_n - T_n), \quad n \in N$$

and construct the interval

$$J_n(b) = [\min(V_n(b),\ V_n(-b)), \max(V_n(b),\ V_n(-b))] \tag{5}$$

THEOREM 1. *(*Brezenski, 1983*) If* $T_n - S = o(S_n - S)$ *and if* $V_n - S = o(T_n - S)$ *then* $\forall b \neq 0, \exists N : \forall n \geq N,\ S \in J_n(b)$. *Moreover* $V_n(\pm b) - S = o(S_n - S)$.

REMARK 2.1. *Brezenski has pointed out a fundamental practical point in (*Brezenski, 1983*): "Under some assumptions, the theorem given above says that for all n greater than N, S belongs to some interval. However, such a N is not known without adding supplementary assumptions. Such an N has been attained if the interval at the step n + 1 is contained in the interval obtained at the step n, whatever n ≥ N may be. This is a good test for having attained this N".*

REMARK 2.2. *As pointed out in Theorem 1, the Brezenski's sequence of intervals* $V_n(\pm b)$ *(so, also* $J_n(b)$*) can have a rate of convergence faster than* $\{S_n\}$, *at the most of* $\{T_n\}$, *but not faster than* $\{T_n\}$. *Hence, we lose the benefit of extrapolation by one column.*

REMARK 2.3. *The value of Brezenski's factor b decides the two factors in constructing the Brezenski's sequence of intervals* $J_n(b)$ *in (5). One is the width of the sequence of intervals* $J_n(b)$, *and the other is the value of N referred to in Theorem 1. Larger the value of b, wider is the interval* $J_n(b)$, *but smaller is N. Whereas, smaller the value of b, tighter is the interval* $J_n(b)$, *but larger is N. In general, the suggested value of b is between* 0 *and* 1, *cf. (*Brezenski, 1983*).*

## 3. THE NIE AND UNIFORM SUBDIVISION

Consider the interval vector (also called as a box) $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_l)^T$ with components $\mathbf{x}_j = \left[\underline{x}_j, \overline{x}_j\right]$. Denote the range of a function $f : \mathbb{R}^n \to \mathbb{R}$ over the box $\mathbf{x}$ as

$$f^{range}(\mathbf{x}) = \{f(x) \,|x \in \mathbf{x}\}$$

Let $f(\mathbf{x})$ denote the natural interval extension (NIE) of $f$, and $e(\mathbf{x})$ be the error interval associated with the range enclosure obtained with $f(\mathbf{x})$. Then, we can express $f(\mathbf{x})$ as

$$f(\mathbf{x}) = \left[\underline{f(\mathbf{x})},\ \overline{f(\mathbf{x})}\right] = f^{range}(\mathbf{x}) + e(\mathbf{x}) \tag{6}$$

Suppose we uniformly subdivide the interval vector $\mathbf{x}$ using the subdivision factor $N$, as follows (wid $\mathbf{x}$ denotes the width of the box $\mathbf{x}$):

$$\mathbf{x}_{i,j} = [\underline{\mathbf{x}}_i + (j-1)\ \text{wid}\ \mathbf{x}_i/N,\ \underline{\mathbf{x}}_i + j\ \text{wid}\ \mathbf{x}_i/N],\ j = 1, 2, \ldots, N \tag{7}$$

$$\mathbf{x}_i = \bigcup_{j=1}^{N} \mathbf{x}_{i,j} \tag{8}$$

$$\mathbf{x} = \bigcup_{j_i=1}^{N} \left(\mathbf{x}_{1,j_1},\ \mathbf{x}_{2,j_2}, \ldots, \mathbf{x}_{l,j_l}\right) \tag{9}$$

Let $e_{(N)}(\mathbf{x})$ be the error interval associated with $N$ partitions of the interval vector $\mathbf{x}$, expressed as

$$e_{(N)}(\mathbf{x}) = \bigcup_{j_i=1}^{N} e\left(\mathbf{x}_{1,j_1},\ \mathbf{x}_{2,j_2}, \ldots, \mathbf{x}_{l,j_l}\right) \tag{10}$$

Define $f_{(N)}(\mathbf{x})$ as

$$f_{(N)}(\mathbf{x}) = \bigcup_{j_i=1}^{N} f\left(\mathbf{x}_{1,j_1},\ \mathbf{x}_{2,j_2}, \ldots, \mathbf{x}_{l,j_l}\right) = f^{range}(\mathbf{x}) + e_{(N)}(\mathbf{x}) \tag{11}$$

Then, Moore (Moore, 1979) has shown that there exists a constant $\sigma$ such that the *excess width* is given by

$$\text{wid}\ e_{(N)}(\mathbf{x}) \leq \frac{\sigma}{N}\ \text{wid}\ \mathbf{x} \tag{12}$$

or

$$\text{wid}\ e_{(N)}(\mathbf{x}) = \frac{\sigma}{N}\ \text{wid}\ \mathbf{x} + O\left(\text{wid}\ \mathbf{x}^2\right) \tag{13}$$

From (11) and (13),

$$\underline{f_{(N)}(\mathbf{x})} = \underline{f^{range}(\mathbf{x})} + \frac{\sigma}{N}\ \text{wid}\ \mathbf{x} + O\left(\text{wid}\ \mathbf{x}^2\right) \tag{14}$$

$$\overline{f_{(N)}(\mathbf{x})} = \overline{f^{range}(\mathbf{x})} + \frac{\sigma}{N} \text{ wid } \mathbf{x} + O\left(\text{wid } \mathbf{x}^2\right) \tag{15}$$

Comparing (14) with (2), for the infimum of the range we have

$$S_n \leftrightarrow \underline{f_{(N)}(\mathbf{x})}, \quad S \leftrightarrow \underline{f^{range}(\mathbf{x})} \quad \lambda^n \leftrightarrow \frac{1}{N^n} \quad a \leftrightarrow \sigma \tag{16}$$

Similarly, from (2) and (15), for the supremum of the range we have

$$S_n \leftrightarrow \overline{f_{(N)}(\mathbf{x})}, \quad S \leftrightarrow \overline{f^{range}(\mathbf{x})} \quad \lambda^n \leftrightarrow \frac{1}{N^n} \quad a \leftrightarrow \sigma \tag{17}$$

REMARK 3.1. *Using NIE and different subdivision factors $N$, we can thus construct two separate sequences converging to two different limits. One is the sequence of lower bounds on the range enclosures converging to the range infimum, and the other is the sequence of upper bounds on the range enclosures converging to the range supremum. In our work, we shall construct these two separate sequences of lower and upper bounds of the range enclosure and extrapolate them to their respective limits (we do not directly apply extrapolation to the sequence of intervals enclosing the range).*

## 4. The Proposed Method

Based on Remark 3.1, we first construct two separate sequences of lower and upper bounds on the infimums and supremums of the range enclosures, and then obtain the Romberg tables for the infimum and supremum by extrapolating these sequences separately to their respective limits.

The algorithm *Sequence_infsup* accepts as inputs the initial box $\mathbf{x}$, the function $f$, and number $K$ of extrapolated columns required in the Romberg Table. It returns the sequences of infimums $\left\{A_0^{(j)}\right\}_{j=0}^{K}$ and the sequences of supremums $\left\{B_0^{(j)}\right\}_{j=0}^{K}$. The sequences of lower and upper bounds are generated for a geometrically increasing uniform subdivision factor $N_j = 2^j, \quad j = 0, 1, \ldots, K$.

**ALGORITHM** SEQUENCES OF LOWER AND UPPER BOUNDS:
$\left[\left\{A_0^{(j)}\right\}_{j=0}^{K}, \ \left\{B_0^{(j)}\right\}_{j=0}^{K}\right]$ = Sequence_infsup($\mathbf{x}$, $f$, $K$)

**Inputs:** Initial box $\mathbf{x}$, function $f$, the number $K$ of extrapolated columns in Romberg table.

**Output:** The sequence of infimums $\left\{A_0^{(j)}\right\}_{j=0}^{K}$

and supremums $\left\{B_0^{(j)}\right\}_{j=0}^{K}$.

**BEGIN Algorithm**

1. Set
$$A_0^{(0)} = \underline{f\,(\mathbf{x})}, \quad B_0^{(0)} = \overline{f\,(\mathbf{x})}$$
2. FOR $j = 1, 2, \ldots, K$

   a) Compute the number of elements in the uniform subdivision partition as $N_j = 2^j$.
   b) Using $N_j$, partition the initial box $\mathbf{x}$ uniformly as per (7), (8), and (9)
   c) In this subdivision partition of $\mathbf{x}$, obtain the range enclosure $f_{(N_j)}(\mathbf{x})$ as per (11)
$$f_{(N_j)}(\mathbf{x}) = \left[\underline{f_{(N_j)}(\mathbf{x})},\ \overline{f_{(N_j)}(\mathbf{x})}\right] = \bigcup_{j_i=1}^{N_j} f\left(\mathbf{x}_{1,j_1},\ \mathbf{x}_{2,j_2}, \ldots, \mathbf{x}_{l,j_l}\right)$$
   d) Set
$$A_0^{(j)} \longleftarrow \underline{f_{(N_j)}(\mathbf{x})},\ B_0^{(j)} \longleftarrow \overline{f_{(N_j)}(\mathbf{x})}$$

3. RETURN $\left\{A_0^{(j)}\right\}_{j=0}^{K}$ and $\left\{B_0^{(j)}\right\}_{j=0}^{K}$.

**END Algorithm**

### 4.1. Romberg Table for the infimum and supremum with the REP

Having constructed the sequences of lower and upper bounds on the range enclosure, we can now apply the REP and obtain the respective Romberg tables by executing the algorithm *Romberg_inf* and *Romberg_sup.*

**ALGORITHM** ROMBERG TABLE FOR THE INFIMUM:

$$[A]^K = \text{Romberg\_inf}\left(\left\{A_0^{(j)}\right\}_{j=0}^{K},\ K\right)$$

**Inputs:** The sequence of lower bounds $\left\{A_0^{(j)}\right\}_{j=0}^{K}$, and the number $K$ of columns required in the Romberg Table.

**Output:** The Romberg Table $[A]^K$ containing the extrapolated sequences.

**BEGIN Algorithm**

1. Set $T_0^{(j)} = A_0^{(j)},\ j = 0, 1, \ldots, K.$

2. Construct the Romberg Table for the range infimum, using the REP in (4):

$$A_k^{(j)} = A_{k-1}^{(j)} + \frac{A_{k-1}^{(j)} - A_{k-1}^{(j-1)}}{(2^k - 1)},\ \begin{cases} k = 1, 2, \ldots, K, \\ j = k, \ldots, K. \end{cases}$$

$$[A]^K = \left\{ A_k^{(j)},\ k = 0, 1, \ldots, K,\ j = k, \ldots, K \right\}$$

3. RETURN the Romberg Table $[A]^K$

**END Algorithm**

We can have a similar algorithm *Romberg_sup* based on $\left\{ B_0^{(j)} \right\}_{j=0}^{K}$ to generate the Romberg Table $[B]^K$ of extrapolated sequences for the range supremum (the description is omitted here).

### 4.2. Brezenski's Error control and approximated bounds

As there is no guarantee that the extrapolation on the lower bound sequence will again result in a lower bound on the range infimum, it is necessary to have an error estimate for the entries in the Romberg Table (the same also holds true for the upper bound sequence). Among the many error estimation methods (Brezenski and Zaglia, 2002; Sidi, 2003; Walz, 1996), we adopt the error control criterion proposed by Brezenski (Brezenski and Zaglia, 2002) to generate intervals which asymptotically contain the true range.

Based on Theorem 1 and Remark 2.1, we can have an algorithm to construct the so-called Brezenski's Table of intervals for the range infimum and Brezenski's Table of intervals for the range supremum, and from these, the final Table range approximations with higher order convergence rate.

**ALGORITHM** RANGE APPROXIMATOR:

$[Range\_approx]^K$ =Range_Approx$\left( [A]^K \text{ and } [B]^K \right)$

**Input:** The Romberg Tables $[A]^K$ and $[B]^K$, and a value for Brezenski's factor $b \in \mathbb{R}$.

**Output:** The Table $[Range\_approx]^K$ containing the range approximating intervals.

**BEGIN Algorithm**

1. Set $k = 0$.

2. From the Romberg Table $[A]^K$, construct Brezenski's table $[C]^K$ of intervals for the infimum as follows (cf. equation 5):

$$V_{k+2}^{(j)}(b) = A_{k+2}^{(j)} - b\left(A_{k+2}^{(j)} - A_{k+1}^{(j)}\right), \quad \begin{cases} k = 0, 1, \ldots, K-2, \\ j = k+2, \ldots, K. \end{cases}$$

$$C_{k+2}^{(j)} = \left[\min\left(V_{k+2}^{(j)}(+b),\ V_{k+2}^{(j)}(-b)\right),\ \max\left(V_{k+2}^{(j)}(+b),\ V_{k+2}^{(j)}(-b)\right)\right],$$

$$k = 0, 1, \ldots, K-2,\ j = k+2, \ldots, K.$$

3. Similarly, from the Romberg Table $[B]^K$, construct Brezenski's Table $[D]^K$ of intervals for supremums.

4. Check for nestedness[1] of the intervals in Table $[C]^K$. For each nested interval, find its infimum. Form the Table $[C_L]^K$ with these infimums as the corresponding entries. Do likewise for the intervals in $[D]^K$, using the supremum of each nested interval to form the Table $[D_U]^K$.

5. Construct intervals whose lower and upper endpoints are the corresponding entries of $[C_L]^K$ and $[D_U]^K$, respectively. Construct a Table of range approximations $[Range\_approx]^K$ based on these intervals.

6. RETURN $[Range\_approx]^K$.

**END Algorithm**

## 5. Numerical Experiments

We test and compare the performance of the proposed technique on several multidimensional examples. The examples considered and the test results are listed in the Appendix.

The range overestimation of the intervals in $[Range\_approx]^K$, and the order of convergence for the same are shown in Tables 2 to 12. The range overestimation of the intervals are computed as

$$\text{Range Overestimation} = \text{wid}\ [Range\_approx]^K\ - \text{wid}\ f^{range}(\mathbf{x}).$$

In Tables 2 to 12, the '$a$' part of the table shows the range overestimation of the intervals in $[Range\_approx]^K$. The first column ($k = 0$) shows the range overestimation of the range enclosures obtained with the NIE and uniform subdivision, whereas the second column ($k = 1$), gives the range overestimation for the first extrapolated column. The subsequent columns in the tables show the range overestimations for extrapolated columns $k = 2, \ldots, K$.

[1] Nestedness is checked columnwise, for consecutive intervals in each column.

The '$b$' part of the table shows the order of convergence of the same intervals. A star entry in the tables signifies that the computed quotient value is erratic, because the numerical zero (i.e. zero within machine precision) is reached for the corresponding range overestimations.

The comparison of the number of subdivisions required and the number of boxes generated to achieve the desired accuracy $\varepsilon = 1e-11$, with the proposed method and with the existing NIE and uniform subdivision method are shown in Tables 13 and 14, respectively[2].

### 5.1. Discussion

Based on the results in Tables 2 to 12, we make the following observations.

- From the quotient entries in the '$b$' part of Tables 2 to 12, we observe that sequences converge columnwise with the order $O\left(\frac{1}{N^K}\right)$. Thus, it seems beneficial to apply extrapolation and accelerate the rate of convergence of the range enclosures obtained with the NIE and the uniform subdivision.
- With the proposed technique, the number of subdivisions required to achieve the desired accuracy are significantly reduced compared to the existing method.
- In all the examples, the intervals in Table $[Range\_approx]^K$ enclose the true function range (these Tables are omitted here for want of space, but are available from the authors).

## 6. CONCLUSIONS

Summarizing the results of the numerical tests, we see that the proposed technique based on extrapolation works well, and generates range approximating intervals of high accuracy. We see that the number of uniform subdivisions required by the proposed method is significantly less compared to the existing NIE and uniform subdivision method. We also obtain, the significant reductions in the number of generated boxes to achieve the desired accuracy.

However, it should be pointed out that we have also come across several examples where the REP was unsuccessful. For instance, this happened in the example

$$f(x) = 1 - 5x + \frac{1}{3}x^3,\ x \in [2,3]\,.$$

The reason for the same is not yet clear. The range approximating intervals generated by this technique are non-validated intervals, and a technique to rigorously validate the same remains to be found. The theoretical proof for the proposed method is also to be investigated.

[2] In some examples, we have subdivided more than necessary, just to further illustrate that a much higher accuracy can usually be achieved with just one or two additional elements in the Romberg Tables.

## References


Asaithambi, N. S., S. Z. and R. E. Moore, 'On computing the range of values'. *Computing* **28**.

Brezenski, C.: 1983, 'Error Control in Convergence Acceleration Processes'. *IMA J. Nunerical Analysis* **3**, 65–80.

Brezenski, C. and M. R. Zaglia: 2002, *Extrapolation Methods, Theory and Practice*. North-Holland, Amsterdam, second edition.

Cornelius, H. and R. Lohner: 1984, 'Computing the Range Values of Real Functions with Accuracy Higher than Second Order'. *Computing* **33**, 331–347.

Costabile, F., G. M. I. and S. Serra: 1996, 'Asymptotic Expansion and Extrapolation for Bernstein Polynomials with Applications'. *BIT* pp. 676–687.

Horowitz, I. M.: 1993, *Quantitative Feedback Design Theory (QFT),*. QFT Publications, Boulder.

Makino, K.: 1998, *Rigorous Analysis of Nonlinear Motion in Particle Accelerators*. PhD thesis, Department of Physics and Astronomy, Michigan State University.

Makino, K. and M. Berz: 2003, 'Taylor Models and Other Validated Functional Inclusion Methods'. *International Journal of Pure and Applied Mathematics* **4**(4), 379–456.

Moore, R. E.: 1979, *Methods And Applications of Interval Analysis*. SIAM Philadelphia.

More, J. J., G. B. S. and K. E. Hillstrom, 'Testing unconstrained optimization software'. *ACM Trans. Mathematical Software* **7**.

Ratz, D. and T. Csendes: 1995, 'On the selection of subdivision directions in interval branch-and-bound methods for global optimization'. *Journal of Global Optimization* **7**, 183–207.

Sidi, A.: 2003, *Practical Extrapolation Methods Theory and Applications*. Cambridge University Press.

Walz, G.: 1996, *Asymptotics Extrapolation*. Akademie Verlag.

# Appendix

EXAMPLE 1. *The 1-dimensional example of Makino and Berz (*Makino, 1998*, Example 1).*

$$f(x) = 1/x + x, \qquad x \in [1.9,\ 2.1]\,.$$

Table 2.

Table 2a

Range overestimation for the function given in Example 1

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ |
|---|---|---|---|---|---|---|
| 2 | 0.050 | | | | | |
| 4 | 0.025 | | | | | |
| 8 | 0.013 | 0.013 | | | | |
| 16 | $6.29E-3$ | $6.29E-3$ | $8.07E-5$ | | | |
| 32 | $3.14E-3$ | $3.14E-3$ | $1.99E-5$ | $2.49E-7$ | | |
| 64 | $1.67E-3$ | $1.67E-3$ | $4.98E-6$ | $3.12E-8$ | $4.01E-10$ | |
| 128 | $7.87E-4$ | $7.87E-4$ | $1.24E-6$ | $3.91E-9$ | $2.49E-11$ | $3.10E-13$ |
| 256 | $3.94E-4$ | $3.94E-4$ | $3.10E-7$ | $4.89E-10$ | $1.55E-12$ | $9.77E-15$ |

Table 2b

Quotients of the above entries of Table 2a

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ |
|---|---|---|---|---|---|---|
| 2 | 1.9963 | | | | | |
| 4 | 1.9972 | | | | | |
| 8 | 1.9983 | 1.9983 | | | | |
| 16 | 1.9991 | 1.9991 | 4.036 | | | |
| 32 | 1.9995 | 1.9995 | 4.017 | 7.980 | | |
| 64 | 1.9997 | 1.9997 | 4.008 | 7.990 | 16.104 | |
| 128 | 1.9998 | 1.9998 | 4.004 | 7.995 | 16.045 | 31.773 |
| | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N^2}\right)$ | $O\left(\frac{1}{N^3}\right)$ | $O\left(\frac{1}{N^4}\right)$ | $O\left(\frac{1}{N^5}\right)$ |

**Comments:** In the above Table 2a, for the uniform subdivision factor $N = 256$, the second extrapolated column ($k = 2$) gives a reduction in the overestimation by 1271 times (from $3.94E-4$ to $3.10E-7$), whereas in the $5^{th}$ extrapolated column ($k = 5$) the reduction is $4.03e+10$ times (from $3.94E-4$ to $9.77E-15$). The rate of convergence of excess width is given in Table 2b. Here, we see that the excess width obtained with the NIE (given in column 2) goes down linearly with $O\left(\frac{1}{N}\right)$. The rate of convergence is accelerated in the subsequent extrapolated columns from $O\left(\frac{1}{N}\right)$ in column 3 to $O\left(\frac{1}{N^5}\right)$ in column 7.

EXAMPLE 2. *The 1-dimensional example of Cornelius and Lohner (*Cornelius and Lohner, 1984*, Example 2).*

$$f(x) = \frac{x+2}{\sqrt{x}}, \qquad x \in [1,3]\,.$$

Table 3.

Table 3a

Range overestimation for the function given in Example 2

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ | $k=6$ |
|---|---|---|---|---|---|---|---|
| 4 | 0.88 | | | | | | |
| 8 | 0.43 | 0.43 | | | | | |
| 16 | 0.22 | 0.22 | | | | | |
| 32 | 0.11 | 0.11 | $8.24E-4$ | | | | |
| 64 | 0.05 | 0.05 | $1.88E-4$ | $5.78E-6$ | | | |
| 128 | 0.03 | 0.03 | $4.51E-5$ | $6.98E-7$ | $3.42E-8$ | | |
| 256 | 0.01 | 0.01 | $1.10E-5$ | $8.58E-8$ | $1.87E-9$ | $7.90E-11$ | |
| 512 | $6.67E-3$ | $6.67E-3$ | $2.73E-6$ | $1.06E-8$ | $1.10E-10$ | $2.36E-12$ | $1.43E-13$ |
| 1024 | $3.33E-3$ | $3.33E-3$ | $6.78E-7$ | $1.32E-9$ | $6.64E-12$ | $7.19E-14$ | $2.22E-15$ |

Table 3b

Quotients of the above entries of Table 3a

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ | $k=6$ |
|---|---|---|---|---|---|---|---|
| 4 | 2.031 | | | | | | |
| 8 | 2.014 | 2.014 | | | | | |
| 16 | 2.006 | 2.006 | | | | | |
| 32 | 2.003 | 2.003 | 4.38 | | | | |
| 64 | 2.001 | 2.001 | 4.18 | 8.29 | | | |
| 128 | 2.0008 | 2.0008 | 4.09 | 8.14 | 18.25 | | |
| 256 | 2.0004 | 2.0004 | 4.04 | 8.07 | 17.07 | 33.52 | |
| 512 | 2.0002 | 2.0002 | 4.02 | 8.03 | 16.52 | 32.77 | 64 |
| | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N^2}\right)$ | $O\left(\frac{1}{N^3}\right)$ | $O\left(\frac{1}{N^4}\right)$ | $O\left(\frac{1}{N^5}\right)$ | $O\left(\frac{1}{N^6}\right)$ |

**Comments:** In the above Table 3*a*, for the uniform subdivision factor $N = 1024$, the second extrapolated column ($k = 2$) gives a reduction in the overestimation by 4912 times (from $3.33E-3$ to $6.78E-7$), whereas in the $6^{th}$ extrapolated column ($k = 6$) the reduction is $1.50e+12$ times (from $3.33E-3$ to $2.22E-15$). The rate of convergence of excess width is given in Table 3*b*. Here, we see that the excess width obtained with the NIE (given in column 2) goes down linearly with $O\left(\frac{1}{N}\right)$. The rate of convergence is accelerated in the subsequent extrapolated columns from $O\left(\frac{1}{N}\right)$ in column 3 to $O\left(\frac{1}{N^6}\right)$ in column 8.

EXAMPLE 3. *The 1-dimensional example of Costabile et. al. (*Costabile and Serra, 1996*, Example 1).*

$$f(x) = \sin(x)\cos(x), \qquad x \in [0, 0.5]\,.$$

Table 4.

Table 4a

Range overestimation for the function given in Example 3

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ | $k=6$ |
|---|---|---|---|---|---|---|---|
| 2 | 0.044 | 0.044 | | | | | |
| 4 | 0.025 | 0.025 | | | | | |
| 8 | 0.014 | 0.014 | | | | | |
| 16 | $6.98E-3$ | $6.98E-3$ | $4.18E-4$ | | | | |
| 32 | $3.54E-3$ | $3.54E-3$ | $1.04E-4$ | $9.73E-7$ | | | |
| 64 | $1.78E-3$ | $1.78E-3$ | $2.58E-5$ | $1.34E-7$ | $4.34E-9$ | | |
| 128 | $8.95E-4$ | $8.95E-4$ | $6.43E-6$ | $1.75E-8$ | $2.66E-10$ | $1.72E-12$ | |
| 256 | $4.48E-4$ | $4.48E-4$ | $1.61E-6$ | $2.24E-9$ | $1.65E-11$ | $5.47E-14$ | $1.06E-15$ |

Table 4b

Quotients of the above entries of Table 4a

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ | $k=6$ |
|---|---|---|---|---|---|---|---|
| 2 | 1.73 | | | | | | |
| 4 | 1.87 | | | | | | |
| 8 | 1.94 | 1.94 | | | | | |
| 16 | 1.97 | 1.97 | 4.032 | | | | |
| 32 | 1.99 | 1.99 | 4.017 | 7.26 | | | |
| 64 | 1.99 | 1.99 | 4.008 | 7.65 | 16.29 | | |
| 128 | 2.00 | 2.00 | 4.004 | 7.83 | 16.15 | 31.42 | – |
| | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N^2}\right)$ | $O\left(\frac{1}{N^3}\right)$ | $O\left(\frac{1}{N^4}\right)$ | $O\left(\frac{1}{N^5}\right)$ | |

**Comments:** In the above Table 4*a*, for the uniform subdivision factor$N = 256$, the second extrapolated column ($k = 2$) gives a reduction in the overestimation by 278 times (from $4.48E-4$ to $1.61E-6$), whereas in the $6^{th}$ extrapolated column ($k = 6$) the reduction is $4.23e+11$ times (from $4.48E-4$ to $1.06E-15$). The rate of convergence of excess width is given in Table 4*b*. Here, we see that the excess width obtained with the NIE (given in column 2) goes down linearly with $O\left(\frac{1}{N}\right)$. The rate of convergence is accelerated in the subsequent extrapolated columns from $O\left(\frac{1}{N}\right)$ in column 3 to $O\left(\frac{1}{N^5}\right)$ in column 7.

EXAMPLE 4. *The 2-dimensional example of Asaithambi et al. (*Asaithambi and Moore, *, Example 2).*

$$f(x) = x_1\,(1 - x_1)\left(1 - \frac{5}{8}x_2 + \frac{3}{2}x_2^2 - x_2^3\right), \quad x_1 \in [-1, 1]\,,\ x_2 \in [0, 1]\,.$$

Table 5.

Table 5a

Range overestimation for the function given in Example 4

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ |
|---|---|---|---|---|---|
| 16 | 0.30 | 0.30 | | | |
| 32 | 0.09 | 0.09 | | | |
| 64 | 0.016 | 0.016 | | | |
| 128 | $8.02E-3$ | $8.02E-3$ | | | |
| 256 | $3.96E-3$ | $3.96E-3$ | $1.05E-4$ | | |
| 512 | $1.97E-3$ | $1.97E-3$ | $2.60E-5$ | $8.94E-8$ | |
| 1024 | $9.80E-4$ | $9.80E-4$ | $6.47E-6$ | $1.12E-8$ | 0.0 |
| 2048 | $4.89E-4$ | $4.89E-4$ | $1.61E-6$ | $1.40E-9$ | 0.0 |
| 4096 | $2.44E-4$ | $2.44E-4$ | $4.03E-7$ | $1.75E-10$ | 0.0 |

Table 5b

Quotients of the above entries of Table 5a

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ |
|---|---|---|---|---|---|
| 16 | 3.428 | 3.428 | | | |
| 32 | 5.381 | 5.381 | | | |
| 64 | 2.052 | 2.052 | | | |
| 128 | 2.026 | 2.026 | | | |
| 256 | 2.013 | 2.013 | 4.038 | | |
| 512 | 2.007 | 2.007 | 4.019 | 8.0 | $\infty$ |
| 1024 | 2.003 | 2.003 | 4.010 | 8.0 | $\infty$ |
| 2048 | 2.002 | 2.002 | 4.005 | 8.0 | $\infty$ |
| | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N^2}\right)$ | $O\left(\frac{1}{N^3}\right)$ | |

**Comments:** In the above Table 5a, for the uniform subdivision factor $N = 4096$, the second extrapolated column ($k = 2$) gives a reduction in the overestimation by 606 times (from $2.44E-4$ to $4.03E-7$), whereas in the $4^{th}$ extrapolated column ($k = 4$) we obtain the true function range. The rate of convergence of excess width is given in Table 5b. Here, we see that the excess width obtained with the NIE (given in column 2) goes down linearly with $O\left(\frac{1}{N}\right)$. The rate of convergence is accelerated in the subsequent extrapolated columns from $O\left(\frac{1}{N}\right)$ in column 3 to $O\left(\frac{1}{N^3}\right)$ in column 5.

EXAMPLE 5. *The 2-dimensional three-hump camel back function example of Asaithambi et al. (*Asaithambi and Moore, *, Example 4).*

$$f(x) = 2x_1^2 - 1.05x_1^4 + \frac{1}{6}x_1^6 - x_1x_2 + x_2^2, \qquad x_1 \in [-2, 4]\,,\ x_2 \in [-2, 4]\,.$$

Table 6.

Table 6a

Range overestimation for the function given in Example 5

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ |
|---|---|---|---|---|---|
| 128 | 13.45 | 13.45 | | | |
| 256 | 6.581 | 6.581 | | | |
| 512 | 3.143 | 3.143 | | | |
| 1024 | 1.572 | 1.572 | | | |
| 2048 | 0.787 | 0.787 | $6.19E-3$ | | |
| 4096 | 0.394 | 0.394 | $4.39E-4$ | $4.24E-7$ | |
| 8192 | 0.197 | 0.197 | $1.10E-4$ | $5.29E-8$ | $4.23E-5$ |
| 16384 | 0.098 | 0.098 | $2.74E-5$ | $6.61E-9$ | $1.06E-11$ |
| 32768 | 0.049 | 0.049 | $6.86E-6$ | $8.26E-10$ | $6.97E-13$ |

Table 6b

Quotients of the above entries of Table 6a

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ |
|---|---|---|---|---|---|
| 128 | 2.043 | 2.043 | | | |
| 256 | 2.094 | 2.094 | | | |
| 512 | 1.999 | 1.999 | | | |
| 1024 | 1.997 | 1.997 | | | |
| 2048 | 1.998 | 1.998 | 14.13 | | |
| 4096 | 1.999 | 1.999 | 3.998 | 8.017 | |
| 8192 | 1.999 | 1.999 | 3.999 | 8.009 | 3998812 |
| 16384 | 1.999 | 1.999 | 3.999 | 8.004 | 15.18 |
| | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N^2}\right)$ | $O\left(\frac{1}{N^3}\right)$ | $O\left(\frac{1}{N^4}\right)$ |

**Comments:** In the above Table 6a, for the uniform subdivision factor $N = 32768$, the second extrapolated column ($k = 2$) gives a reduction in the overestimation by 7143 times (from 0.049 to $6.86E-6$), whereas in the $4^{th}$ extrapolated column ($k = 4$) the reduction is $7.03e+10$ times (from 0.049 to $6.97E-13$). The rate of convergence of excess width is given in Table 6b. Here, we see that the excess width obtained with the NIE (given in column 2) goes down linearly with $O\left(\frac{1}{N}\right)$. The rate of convergence is accelerated in the subsequent extrapolated columns from $O\left(\frac{1}{N}\right)$ in column 3 to $O\left(\frac{1}{N^4}\right)$ in column 6.

EXAMPLE 6. *The 2-dimensional exponential function of Moore (*Moore, 1979*, pp. 45).*

$$f(x) = x_1 \exp\left(x_1 + x_1^2\right) - x_2^2, \qquad x_1 \in [1,2]\,,\ x_2 \in [0,1]\,.$$

Table 7.

Table7a

Range overestimation for the function given in Example 6

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ | $k=6$ |
|---|---|---|---|---|---|---|---|
| 4 | 0.51 | | | | | | |
| 8 | 0.25 | 0.25 | | | | | |
| 16 | 0.13 | 0.13 | 0.015 | | | | |
| 32 | 0.06 | 0.06 | $4.29E-3$ | | | | |
| 64 | 0.03 | 0.03 | $1.10E-3$ | $1.50E-5$ | | | |
| 128 | 0.02 | 0.02 | $2.78E-4$ | $9.34E-7$ | $3.25E-7$ | | |
| 256 | $7.81E-3$ | $7.81E-3$ | $6.96E-5$ | $5.83E-8$ | $2.03E-8$ | $1.58E-10$ | |
| 512 | $3.91E-3$ | $3.91E-3$ | $1.74E-5$ | $3.64E-9$ | $1.27E-9$ | $2.46E-12$ | $8.34E-13$ |
| 1024 | $1.95E-3$ | $1.95E-3$ | $4.35E-6$ | $2.27E-10$ | $7.91E-11$ | $4.44E-14$ | $1.91E-14$ |

Table 7b

Quotients of the above entries of Table 7a

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ | $k=6$ |
|---|---|---|---|---|---|---|---|
| 4 | 2.0278 | | | | | | |
| 8 | 2.0151 | 2.0151 | | | | | |
| 16 | 2.0085 | 2.0085 | 3.528 | | | | |
| 32 | 2.0044 | 2.0044 | 3.888 | | | | |
| 64 | 2.0022 | 2.0022 | 3.972 | 16.112 | | | |
| 128 | 2.0011 | 2.0011 | 3.993 | 16.028 | 16.037 | | |
| 256 | 2.0005 | 2.0005 | 3.998 | 16.007 | 16.009 | 64.14 | |
| 512 | 2.0002 | 2.0002 | 3.999 | 16.001 | 16.001 | 55.37 | $*$ |
| | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N^2}\right)$ | $O\left(\frac{1}{N^3}\right)$ | $O\left(\frac{1}{N^4}\right)$ | $O\left(\frac{1}{N^5}\right)$ | |

**Comments:** In the above Table 7a, for the uniform subdivision factor $N = 1024$, the second extrapolated column ($k = 2$) gives a reduction in the overestimation by 448 times (from $1.95E-3$ to $4.35E-6$), whereas in the $6^{th}$ extrapolated column ($k = 6$) the reduction is$1.02e+11$ times (from $1.95E-3$ to $1.91e-14$). The rate of convergence of excess width is given in Table 7b. Here, we see that the excess width obtained with the NIE (given in column 2) goes down linearly with $O\left(\frac{1}{N}\right)$. The rate of convergence is accelerated in the subsequent extrapolated columns from $O\left(\frac{1}{N}\right)$ in column 3 to $O\left(\frac{1}{N^5}\right)$ in column 7.

EXAMPLE 7. *The 3-dimensional function of Makino (*Makino and Berz, 2003*, pp. 403).*

$$f(x,y,z) = \frac{4\tan(3y)}{3x + x\sqrt{\frac{6x}{-7(x-8)}}} - 120 - 2x - 7z(1+2y) - \sinh(0.5 + \frac{6y}{8y+7}) + \frac{(3y+13)^2}{3z}$$
$$-20z(2z-5) + \frac{5x\tanh(0.9z)}{\sqrt{5y}} - 20y\sin(3z),$$
$$x_1 \in [1.75, 2.25]\,,\ x_2 \in [0.75, 1.25]\,,\ x_3 \in [0.75, 1.25]\,.$$

Table 8.

Table 8a

Range overestimation for the function given in Example 7

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ | $k=6$ |
|---|---|---|---|---|---|---|---|
| 8 | 16.5 | 16.5 | | | | | |
| 16 | 8.16 | 8.16 | | | | | |
| 32 | 4.06 | 4.06 | | | | | |
| 64 | 2.03 | 2.03 | $8.71e-3$ | $7.54e-4$ | | | |
| 128 | 1.01 | 1.01 | $2.43e-3$ | $9.32e-5$ | $5.68e-7$ | $1.06e-7$ | |
| 256 | 0.51 | 0.51 | $6.39e-4$ | $1.16e-5$ | $3.63e-8$ | $3.32e-9$ | $2.07e-11$ |
| 512 | 0.25 | 0.25 | $1.64e-4$ | $1.44e-6$ | $2.29e-9$ | $1.04e-10$ | $4.67e-13$ |

Table 8b

Quotients of the above entries of Table 8a

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ | $k=6$ |
|---|---|---|---|---|---|---|---|
| 8 | 2.035 | | | | | | |
| 16 | 2.024 | | | | | | |
| 32 | 2.017 | 2.017 | | | | | |
| 64 | 2.009 | 2.009 | | | | | |
| 128 | 2.005 | 2.005 | 3.580 | 8.092 | | | |
| 256 | 2.003 | 2.003 | 3.802 | 8.061 | 15.647 | 31.9162 | |
| 512 | 2.0007 | 2.0007 | 3.903 | 8.034 | 15.851 | 31.9161 | 44.202 |
| | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N^2}\right)$ | $O\left(\frac{1}{N^3}\right)$ | $O\left(\frac{1}{N^4}\right)$ | $O\left(\frac{1}{N^5}\right)$ | $O\left(\frac{1}{N^6}\right)$ |

**Comments:** In the above Table 8a, for the uniform subdivision factor $N = 512$, the second extrapolated column ($k = 2$) gives a reduction in the overestimation by 1524.4 times (from 0.25 to $1.64e-4$), whereas in the $6^{th}$ extrapolated column ($k = 6$) the reduction is $5.35e+11$ times (from 0.25 to $4.67e-13$). The rate of convergence of excess width is given in Table 8b. Here, we see that the excess width obtained with the NIE (given in column 2) goes down linearly with $O\left(\frac{1}{N}\right)$. The rate of convergence is accelerated in the subsequent extrapolated columns from $O\left(\frac{1}{N}\right)$ in column 3 to $O\left(\frac{1}{N^6}\right)$ in column 8.

EXAMPLE 8. *The 4-dimensional trigonometric function of More et al. (*More and Hillstrom, *, Example 26).*

$$f(x)=\sum_{i=1}^{4} f_i\left(x\right)^2\,,\;\; f_i\left(x\right)=4-\sum_{j=1}^{4}\cos x_j+i\left(1-\cos x_i\right)-\sin x_i,\qquad x_i\in[0.75,2.75]^4\,.$$

Table 9.

Table 9a

Range overestimation for the function given in Example 8

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ |
|---|---|---|---|---|---|---|
| 16 | 0.70 | | 0.15 | | | |
| 32 | 0.37 | 0.37 | 0.04 | $2.17E-3$ | | |
| 64 | 0.19 | 0.19 | $9.66E-3$ | $2.56E-4$ | | |
| 128 | 0.10 | 0.10 | $2.44E-3$ | $3.08E-5$ | | |
| 256 | 0.05 | 0.05 | $6.12E-4$ | $3.76E-6$ | $7.75E-8$ | $1.35E-8$ |
| 512 | 0.02 | 0.02 | $1.53E-4$ | $4.63E-7$ | $6.08E-9$ | $4.22E-10$ |
| 1024 | 0.01 | 0.01 | $3.84E-5$ | $5.75E-8$ | $4.19E-10$ | $1.34E-11$ |
| 2048 | $6.07E-3$ | $6.07E-3$ | $9.60E-6$ | $7.17E-9$ | $2.76E-11$ | $6.42E-13$ |
| 4096 | $3.04E-3$ | $3.04E-3$ | $2.40E-6$ | $8.95E-10$ | $1.98E-12$ | $2.50E-13$ |

Table 9b

Quotients of the above entries of Table 9a

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ |
|---|---|---|---|---|---|---|
| 16 | 1.897 | | 3.827 | | | |
| 32 | 1.949 | 1.949 | 3.920 | 8.465 | | |
| 64 | 1.974 | 1.974 | 3.962 | 8.334 | | |
| 128 | 1.987 | 1.987 | 3.981 | 8.195 | | |
| 256 | 1.993 | 1.993 | 3.990 | 8.105 | 12.75 | 31.92 |
| 512 | 1.996 | 1.996 | 3.995 | 8.054 | 14.52 | 31.59 |
| 1024 | 1.998 | 1.998 | 3.997 | 8.027 | 15.18 | * |
| 2048 | 1.999 | 1.999 | 3.998 | 8.012 | 13.91 | * |
| | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N^2}\right)$ | $O\left(\frac{1}{N^3}\right)$ | $O\left(\frac{1}{N^4}\right)$ | $O\left(\frac{1}{N^5}\right)$ |

**Comments:** In the above Table 9a, for the uniform subdivision factor $N=4096$, the second extrapolated column ($k=2$) gives a reduction in the overestimation by 1266 times (from $3.04E-3$ to $2.40E-6$), whereas in the $5^{th}$ extrapolated column ($k=5$) the reduction is $1.22e+10$ times (from $3.04E-3$ to $2.50E-13$). The rate of convergence of excess width is given in Table 9b. Here, we see that the excess width obtained with the NIE (given in column 2) goes down linearly with $O\left(\frac{1}{N}\right)$. The rate of convergence is accelerated in the subsequent extrapolated columns from $O\left(\frac{1}{N}\right)$ in column 3 to $O\left(\frac{1}{N^5}\right)$ in column 7.

EXAMPLE 9. *The 5-dimensional Griewank function of Ratz and Csendes (*Ratz and Csendes, 1995*, pp. 205).*

$$f(x) = \sum_{i=1}^{5} \frac{x_i^2}{400} - \prod_{i=1}^{5} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \qquad x_i \in [-601, -599]^5 .$$

Table 10.

Table 10a

Range overestimation for the function given in Example 9

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ |
|---|---|---|---|---|---|
| 4 | 0.14 | | | | |
| 8 | 0.08 | | | | |
| 16 | 0.04 | 0.04 | | | |
| 32 | 0.02 | 0.02 | | | |
| 64 | $9.72E-3$ | $9.72E-3$ | $4.31E-5$ | | |
| 128 | $4.86E-3$ | $4.86E-3$ | $7.68E-6$ | $4.103E-6$ | $5.06E-8$ |
| 256 | $2.43E-3$ | $2.43E-3$ | $1.53E-6$ | $5.16E-7$ | $1.47E-9$ |
| 512 | $1.22E-3$ | $1.22E-3$ | $3.35E-7$ | $6.46E-8$ | $4.64E-11$ |

Table 10b

Quotients of the above entries of Table 10a

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ |
|---|---|---|---|---|---|
| 4 | 1.926 | | | | |
| 8 | 1.952 | | | | |
| 16 | 1.986 | 1.986 | | | |
| 32 | 1.995 | 1.995 | | | |
| 64 | 1.998 | 1.998 | 5.61 | | |
| 128 | 1.999 | 1.999 | 5.01 | 7.95 | 34.33 |
| 256 | 1.999 | 1.999 | 4.58 | 7.99 | 31.76 |
| | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N^2}\right)$ | $O\left(\frac{1}{N^3}\right)$ | $O\left(\frac{1}{N^4}\right)$ |

**Comments:** In the above Table 10a, for the uniform subdivision factor $N = 512$, the second extrapolated column ($k = 2$) gives a reduction in the overestimation by 3642 times (from $1.22E-3$ to $3.35E-7$), whereas in the $4^{th}$ extrapolated column ($k = 4$) he reduction is $2.63e+7$ times (from $1.22E-3$ to $4.64E-11$). The rate of convergence of excess width is given in Table 10b. Here, we see that the excess width obtained with the NIE (given in column 2) goes down linearly with $O\left(\frac{1}{N}\right)$. The rate of convergence is accelerated in the subsequent extrapolated columns from $O\left(\frac{1}{N}\right)$ in column 3 to $O\left(\frac{1}{N^4}\right)$ in column 6.

EXAMPLE 10. *The 6-dimensional trigonometric function of More et al. (*More and Hillstrom, , *Example 26).*

$$f(x)=\sum_{i=1}^{6} f_i(x)^2\,,\ f_i(x)=6-\sum_{j=1}^{6}\cos x_j+i\,(1-\cos x_i)-\sin x_i,\qquad x_i\in[0.75,2.75]^6\,.$$

Table 11.

Table 11a

Range overestimation for the function given in Example 10

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ |
|---|---|---|---|---|---|---|
| 4 | 5.56 | | | | | |
| 8 | 3.43 | | | | | |
| 16 | 1.88 | | 0.33 | | | |
| 32 | 0.98 | 0.98 | 0.08 | $1.21E-3$ | | |
| 64 | 0.50 | 0.50 | 0.02 | $1.12E-4$ | $6.53E-6$ | |
| 128 | 0.25 | 0.25 | $5.29E-3$ | $1.11E-5$ | $2.11E-6$ | |
| 256 | 0.13 | 0.13 | $1.32E-3$ | $1.20E-6$ | $1.87E-7$ | $1.87E-8$ |
| 512 | 0.06 | 0.06 | $3.31E-4$ | $1.37E-7$ | $1.34E-8$ | $5.82E-10$ |
| 1024 | 0.03 | 0.03 | $8.28E-5$ | $1.63E-8$ | $8.90E-10$ | $1.92E-11$ |

Table 11b

Quotients of the above entries of Table 11a

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ |
|---|---|---|---|---|---|---|
| 4 | 1.622 | | | | | |
| 8 | 1.822 | | | | | |
| 16 | 1.914 | | 3.957 | | | |
| 32 | 1.957 | 1.957 | 3.984 | 10.74 | | |
| 64 | 1.979 | 1.979 | 3.994 | 10.07 | 03.09 | |
| 128 | 1.989 | 1.989 | 3.997 | 09.31 | 11.32 | |
| 256 | 1.994 | 1.994 | 3.999 | 08.74 | 13.96 | 32.06 |
| 512 | 1.997 | 1.997 | 3.999 | 08.40 | 15.03 | 30.32 |
| | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N^2}\right)$ | $O\left(\frac{1}{N^3}\right)$ | $O\left(\frac{1}{N^4}\right)$ | $O\left(\frac{1}{N^5}\right)$ |

**Comments:** In the above Table 11a, for the uniform subdivision factor $N=1024$, the second extrapolated column ($k=2$) gives a reduction in the overestimation by 362 times (from 0.03 to $8.28E-5$), whereas in the $5^{th}$ extrapolated column ($k=5$) the reduction is $1.56e+9$ times (from 0.03 to $1.92E-11$). The rate of convergence of excess width is given in Table 11b. Here, we see that the excess width obtained with the NIE (given in column 2) goes down linearly with $O\left(\frac{1}{N}\right)$. The rate of convergence is accelerated in the subsequent extrapolated columns from $O\left(\frac{1}{N}\right)$ in column 3 to $O\left(\frac{1}{N^5}\right)$ in column 7.

EXAMPLE 11. *The 3-dimensional non-rational example of Horowitz (*Horowitz, 1993*, pp. 129). The magnitude function for the non-rational system is*

$$f(x) = -10\log_{10}\left\{1 + x_2\left(x_2 + 2\cos 2x_1\right)\right\}, \quad x_1 \in [1,2]\,,\ x_2 \in [0.4, 0.6]\,,\ x_3 \in [0.01, 0.02]\,.$$

Table 12.

Table 12a

Range overestimation for the function given in Example 11

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ |
|---|---|---|---|---|---|
| 4 | 0.959 | | | | |
| 8 | 0.457 | 0.457 | | | |
| 16 | 0.223 | 0.223 | 0.014 | | |
| 32 | 0.110 | 0.110 | $2.85E-3$ | $1.69E-4$ | |
| 64 | 0.055 | 0.055 | $6.53E-4$ | $1.98E-5$ | $2.06E-6$ |
| 128 | 0.027 | 0.027 | $1.56E-4$ | $2.40E-6$ | $1.02E-7$ |
| 256 | 0.013 | 0.013 | $3.83E-5$ | $2.96E-7$ | $5.72E-9$ |
| 512 | 6.824 | 6.824 | $9.47E-6$ | $3.67E-8$ | $3.38E-10$ |
| 1024 | $3.41E-3$ | $3.41E-3$ | $2.35E-6$ | $4.57E-9$ | $2.08E-11$ |

Table 12b

Quotients of the above entries of Table 12a

| $N$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ |
|---|---|---|---|---|---|
| 4 | 2.1019 | | | | |
| 8 | 2.0469 | 2.0469 | | | |
| 16 | 2.0226 | 2.0226 | 4.792 | | |
| 32 | 2.0111 | 2.0111 | 4.366 | 8.534 | |
| 64 | 2.0055 | 2.0055 | 4.177 | 8.256 | 20.11 |
| 128 | 2.0027 | 2.0027 | 4.087 | 8.125 | 17.89 |
| 256 | 2.0013 | 2.0013 | 4.043 | 8.062 | 16.90 |
| 512 | 2.0006 | 2.0006 | 4.021 | 8.031 | 16.27 |
| | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N^2}\right)$ | $O\left(\frac{1}{N^3}\right)$ | $O\left(\frac{1}{N^4}\right)$ |

**Comments:** In the above Table 12a, for the uniform subdivision factor $N = 1024$, the second extrapolated column ($k = 2$) gives a reduction in the overestimation by 1451 times (from $3.41E-3$ to $2.35E-6$), whereas in the $4^{th}$ extrapolated column ($k = 4$) the reduction is $1.64e+8$ times (from $3.41E-3$ to $2.08E-11$). The rate of convergence of excess width is given in Table 12b. Here, we see that the excess width obtained with the NIE (given in column 2) goes down linearly with $O\left(\frac{1}{N}\right)$. The rate of convergence is accelerated in the subsequent extrapolated columns from $O\left(\frac{1}{N}\right)$ in column 3 to $O\left(\frac{1}{N^4}\right)$ in column 6.

Table 13. Comparison of the number of uniform subdivisions required to achieve a range accuracy of $1e-11$ with the existing and proposed methods

| Example Named | $l$ | No. of Subdivisions Required | |
|---|---|---|---|
| | | Existing method | Proposed method |
| 1 | 1 | 37 | 7 |
| 2 | 1 | 37 | 8 |
| 3 | 1 | 37 | 7 |
| 4 | 2 | 37 | 10 |
| 5 | 2 | 51 | 14 |
| 6 | 2 | 41 | 9 |
| 7 | 3 | 46 | 8 |
| 8 | 4 | 44 | 10 |
| 9 | 5 | 41 | 9 |
| 10 | 6 | 51 | 10 |
| 11 | 3 | 41 | 10 |

Table 14. Comparison of the number of boxes processed to achieve a range accuracy of $1e-11$ with the existing and proposed methods

| Example Named | $l$ | No. of Subboxes Generated | |
|---|---|---|---|
| | | Existing method | Proposed method |
| 1 | 1 | $1.37e+11$ | 128 |
| 2 | 1 | $1.37e+11$ | 256 |
| 3 | 1 | $1.37e+11$ | 128 |
| 4 | 2 | $2.75e+11$ | 2048 |
| 5 | 2 | $4.50e+15$ | 32768 |
| 6 | 2 | $4.40e+12$ | 1024 |
| 7 | 3 | $2.11e+14$ | 768 |
| 8 | 4 | $7.04e+13$ | 4096 |
| 9 | 5 | $1.10e+13$ | 2560 |
| 10 | 6 | $1.35e+16$ | 6144 |
| 11 | 3 | $6.60e+12$ | 3072 |

# Interval Finite Element as a Basis for Generalized Models of Uncertainty in Engineering Mechanics

Rafi L. Muhanna
*School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, e-mail: rafi.muhanna@gtrep.gatech.edu*

Robert L. Mullen
*Department of Civil and Environmental Engineering, Case Western Reserve University, Cleveland, OH 44106, USA, e-mail: rlm@po.cwru.edu*

Hao Zhang
*School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, e-mail: hao.zhang@ce.gatech.edu*

**Abstract.** Latest scientific and engineering advances have started to recognize the need of defining multiple types of uncertainty. Probabilistic modeling cannot handle situations with incomplete or little information on which to evaluate a probability, or when that information is nonspecific, ambiguous, or conflicting [46, 11, 43]. Many interval-based models of uncertainty have been developed to treat such situations.

This paper presents an interval approach for the treatment of parameter uncertainty for linear static problems of mechanics. Uncertain parameters are introduced in the form of unknown but bounded quantities (intervals). Interval analysis is applied to Finite Element Method to analyze the system response due to uncertain stiffness and loading. To avoid overestimation, the formulation is based on an element-by-element (EBE) technique. Element matrices are formulated, based on the physics of materials, and the Lagrange multiplier method is applied to impose the necessary constraints for compatibility and equilibrium. Earlier EBE formulation provided sharp bounds only on displacements [29]. Based on the developed formulation, the bounds on the system's displacement and forces are obtained simultaneously and have the same level of accuracy. Very sharp enclosures for the exact system responses are obtained. A number of numerical examples are introduced and scalability is illustrated.

## 1. Introduction

An important issue faced by real life engineering practice is how to deal with variables and parameters of uncertain values. For a proper performance assessment, these uncertainties must be accounted for appropriately. There are various ways in which the types of uncertainty might be classified. One is distinguish between "aleatory" (or stochastic) uncertainty and "epistemic" uncertainty. The first refers to underlying, intrinsic variabilities of physical quantities and the latter refers to uncertainty which might be reduced with additional data or information, or better modeling and better parameter estimation [23]. Probability theory is the traditional approach to handle uncertainty. This approach requires sufficient statistical data to justify the assumed statistical distributions. Analysts agree that, given sufficient statistical data, the probability theory describes the stochastic uncertainty well. However, probabilistic modeling cannot handle situations with incomplete or little information on which to evaluate a probability, or when that information is nonspecific, ambiguous, or

conflicting [46, 11, 43]. Many generalized models of uncertainty have been developed to treat such situations, which includes imprecise probabilities [46], Dempster-Shafer theory of evidence [9, 44] and random set [19], fuzzy sets [47], possibility theory [8], probability bounds [12], convex model [5], and others.

These set-based uncertainty models have a variety of mathematical descriptions, however, they are all closely connected with interval arithmetic [25]. For example, a fuzzy number [47] can be viewed as a set of valued intervals with different confidence of given level of presumptions ($\alpha$ cuts). Thus fuzzy arithmetic can be performed as interval arithmetic on $\alpha$ cuts. A Dempster-Shafer structure [9, 44] with interval focal elements can be viewed as a set of intervals with probability mass assignments, where the computation is carried out using the interval focal sets. Probability bounds analysis [12] is a combination of the methods of standard interval analysis and probability theory. Uncertain variables are decomposed into a list of pairs of the form (interval, probability). In this sense, interval arithmetic serves as the calculation tool for the generalized models of uncertainty.

Recently, various generalized models of uncertainty have been applied to finite element method (FEM) to solve a partial differential equation with uncertain parameters. Regardless what model is adopted, the proper interval solution will represent the first requirement for any further rigorous formulation. Finite element method with interval valued parameters results in Interval Finite Element Method (IFEM). The numerical solution of IFEM is the focus of this paper. Different formulations of IFEM have been developed. However, the used solution techniques can be reduced to two main approaches; optimization-based and anti-optimization. In the optimization approaches [20, 38, 1, 24], optimization algorithm is employed to search for the extrema (max/min) of the system response in the interval parameter domain. This optimization approach often encounters practical difficulties. Firstly it requires sophisticated optimization algorithm, where the objective function is implicit and complicated in most structural engineering problems, thus often only approximate solution is achievable. Secondly, this approach is computationally expensive. For each response quantity, two optimization problems must be solved to find the extreme lower and the extreme upper bounds. This will be a huge computational effort, especially in the case of practical engineering problems.

More recently, anti-optimization approaches for the interval finite element analysis have been developed in a number of works. For linear elastic problems, this approach leads to a system of linear interval equations, then the solution is sought using interval methods developed for this purpose. The major difficulty associated with this approach is the so-called "dependency problem" [26, 34, 17, 29]. The dependency in interval arithmetic leads to an overestimation of the system response. A straightforward replacement of the system parameters with interval ones without taking care of the dependency problem is known as a naïve application of interval arithmetic in finite element method (naïve interval FEM), and usually such a use results in meaningless wide and even catastrophic results [29].

In the anti-optimization category, a number of developments can be presented. A combinatorial approach (based on an exhaustive combination of the extreme values of the interval parameters) was used in [37]. This approach gives exact solution in linear elastic problems. However, it is computationally tedious and expensive, and is limited to the solutions of

small-scale problems only. Convex modeling and superposition approach was proposed to analyze load uncertainty in [35], and exact solution was obtained. However, the superposition is only applicable to load uncertainty. Combinatorial approach was used in [14] to treat interval modulus of elasticity. Chen et al. [6] have developed static displacement bounds analysis using matrix perturbation theory. The first-order perturbation was used and second-order term had been neglected. The result is approximate and not guaranteed to contain the exact bounds. McWilliam [22] proposed two methods for determining the static displacement bounds of structures with interval parameters. The first method is a modified version of perturbation analysis. The second method is based on the assumption that the displacement surface is monotonic. However, for the general case, the validity of monotonicity is difficult to verify. Dessombz [10] have introduced an interval FEM in which the interval parameters were factorized during the assemblage process of the stiffness matrix, then Rump's iterative algorithm [40] was employed for solving the linear interval equation. In this work, the overestimation control becomes more difficult with the increase of the number of the interval parameters, which does not lead to useful results for practical problems. In the works of Muhanna and Mullen [27], Mullen and Muhanna [30, 31], an interval-based fuzzy finite element has been developed for treating uncertain loads in static structural problems. Load dependency was eliminated and the exact solution was obtained. Also, Muhanna and Mullen [29] have developed an interval finite element method based on element-by-element technique and Lagrange multiplier. Uncertain modulus of elasticity was considered. Most sources of overestimation were eliminated, and a sharp result for displacement was obtained.

In this paper a new formulation for interval finite element analysis of linear elastic structures will be introduced. Material and load uncertainties are handled simultaneously and sharp enclosures on the system's displacement and forces are obtained efficiently. A brief review of interval arithmetic is presented, the formulation is described, and numerical examples are given.

## 2. Short review of interval arithmetic

For simplicity and better clarity, all interval quantities will be introduced in bold face. Detailed information about interval arithmetic can be found in series of books and publications such as [16, 25, 2, 34, 42, 45].

### 2.1. Basic Definition

An interval number is a closed set in $\mathbb{R}$ that includes the possible range of an unknown real number, where $\mathbb{R}$ denotes the set of real numbers. Therefore, a real interval is a set of the form

$$\mathbf{x} = [\underline{x},\ \overline{x}] \tag{1}$$

where $\underline{x}$ and $\overline{x}$ are the lower and upper bounds (endpoints) of the interval number $\mathbf{x}$ respectively. The midpoint $\check{x}$ of $\mathbf{x}$ is introduced as

$$\check{x} \equiv \text{mid}(\mathbf{x}) := \frac{\underline{x} + \overline{x}}{2} \tag{2}$$

Sometimes it is convenient to write the interval in the midpoint form

$$\mathbf{x} = \check{x}(1 + \boldsymbol{\alpha}) \tag{3}$$

in which $\boldsymbol{\alpha}$ is a 0-midpoint interval. For example, when we say $\mathbf{x}$ has 4% uncertainty, it means $\boldsymbol{\alpha} = [-0.02, 0.02]$, and $\mathbf{x} = \check{x}(1 + [-0.02, 0.02])$.

The set of real intervals will be denoted by $\mathbb{IR}$. Operations with at least one interval operand are by definition interval operations. It is easy to see that the set of all possible results for $x \in \mathbf{x}$ and $y \in \mathbf{y}$ forms a closed interval (for 0 not in a denominator interval), and the endpoints can be calculated by

$$\mathbf{x} \circ \mathbf{y} = [\min\,(x_i \circ y_i),\ \max\,(x_i \circ y_i)] \quad \text{for } \circ \in \{+, -, \cdot, /\} \tag{4}$$

## 2.2. Dependency Problem in Interval Arithmetic

The interval-system quality is measured by the width of the interval results, and a sharp enclosure for the exact solution is desirable. However, the width of results may be unnecessarily wide in some occasions due to dependency effect. For example, if the interval function $f(\mathbf{x}) = \mathbf{x} - \mathbf{x}$ is evaluated with $\mathbf{x} = [a, b] = [1, 2]$, the interval subtraction rule (Appendix A) gives the result: $f(\mathbf{x}) = [a - b, b - a] = [-1, 1]$, which is containing the exact solution $[0, 0]$, but much wider. The interval arithmetic implicitly made the assumption that all intervals are independent, namely it treats $\mathbf{x} - \mathbf{x}$ as if evaluating the intervals $\mathbf{x} - \mathbf{y}$, and $\mathbf{x}, \mathbf{y}$ are two independent interval quantities that happen to have the same bounds. This phenomenon is referred as overestimation due to "dependency" of the variables [26, 34, 17, 29]. Reducing the overestimation is a central issue to a successful interval analysis. In general, sharp results are obtained with the proper understanding of the physical nature of the problem and reduction of the dependence. In the above example, the exact solution could be achieved in evaluating $\mathbf{x} - \mathbf{x}$ as $\mathbf{x}(1 - 1) = 0$.

## 2.3. Interval Vectors and Matrices

An interval matrix $\mathbf{A} \in \mathbb{IR}^{n \times k}$ is interpreted as a set of real $n \times k$ matrices by the convention $\mathbf{A} = \{A \in \mathbb{R}^{n \times k} \mid A_{ij} \in \mathbf{A_{ij}} \quad \text{for } i = 1, \ldots, n;\ j = 1, \ldots, k\}$. The set of $n \times k$ interval matrices is denoted by $\mathbb{IR}^{n \times k}$. An $n \times 1$ interval matrix is an interval vector, denoted by $\mathbb{IR}^n$. Operations on interval matrices are extended naturally from the corresponding deterministic matrices operations. Algebraic properties of interval matrix operations are provided in [34, 3, 21].

## 2.4. Linear Interval Equations

A linear interval equation with coefficient matrix $\mathbf{A} \in \mathbb{IR}^{n\times n}$ and right-hand side $\mathbf{b} \in \mathbb{IR}^n$ is defined as the family of linear equations

$$Ax = b \quad (A \in \mathbf{A},\ b \in \mathbf{b}) \tag{5}$$

Therefore, a linear interval equation represents systems of equations in which the coefficients are unknown numbers ranging in certain intervals. The solution set of (5) is given by:

$$\Sigma(\mathbf{A},\, \mathbf{b}) = \{x \in \mathbb{R}^n \mid \exists A \in \mathbf{A},\ \exists b \in \mathbf{b} : Ax = b\} \tag{6}$$

The solution set $\Sigma(\mathbf{A},\, \mathbf{b})$ usually is not an interval vector, and does not need even to be convex; in general, $\Sigma(\mathbf{A},\, \mathbf{b})$ has a very complicated structure. In order to guarantee that the solution set $\Sigma(\mathbf{A},\, \mathbf{b})$ is bounded, it is required that the matrix $\mathbf{A}$ be regular, i.e. that every matrix $A \in \mathbf{A}$ is nonsingular. The hull of the solution set $\Sigma(\mathbf{A},\, \mathbf{b})$ is an interval vector which has the narrowest possible interval components, denoted as

$$\mathbf{A}^H\mathbf{b} := \diamondsuit\Sigma(\mathbf{A}, \mathbf{b}) \tag{7}$$

where

$$\mathbf{A}^H\mathbf{b} = \diamondsuit\{A^{-1}b | A \in \mathbf{A}, b \in \mathbf{b}\} \quad \text{for } \mathbf{b} \in \mathbb{IR}^n \tag{8}$$

In fact, computing the hull of the solution set for the general case is NP-Hard problem [39]. The solution of interest is seeking an enclosure, i.e., an interval vector $\mathbf{x}$ containing $\mathbf{A}^H\mathbf{b}$, while narrow enough to be practically useful:

$$\mathbf{A}^H\mathbf{b} \ \subseteq \mathbf{x} \tag{9}$$

A number of methods have been developed to find $\mathbf{x}$ for the general linear interval equations such as Interval Gauss elimination, Interval Gauss-Seidel iteration, Krawczyk's iteration, and fixed-point iteration [15, 32, 34, 18, 40, 41]. These algorithms usually involve a preconditioning of the coefficient matrix, and then iterations are performed to get the enclosure. The present work is using Brouwer's fixed point theorem and Krawczyk's operator. This method has been discussed in the works of [15, 32, 33, 18, 40, 41].

One typical approach to find the solution of a linear system $Ax = b$, is to transform it into a fixed point equation $g(x) = x$, in which

$$g(x) = x - R(Ax - b) = Rb + (I - RA)x \tag{10}$$

and $R$ is a nonsingular matrix. From Brouwer's fixed point theorem, it follows that for some interval vector $\mathbf{x} \in \mathbb{IR}^n$

$$Rb + (I - RA)x \in \mathbf{x} \quad \forall x \in \mathbf{x} \tag{11}$$

implies

$$\exists x \in \mathbf{x} : Ax = b \tag{12}$$

To verify condition (11) is a range determination problem, and can be reduced to interval arithmetic:

$$Rb + (I - RA)\mathbf{x} \subseteq \mathbf{x} \tag{13}$$

If an interval vector $\mathbf{x}$ satisfying (13) can be found, then $\mathbf{x}$ contains the solution of $Ax = b$. The result can be extended to find the enclosure of the solution set of linear interval equation $\mathbf{Ax} = \mathbf{b}$ [34, 42]. The following theorem can be presented:

THEOREM 1 (Rump 2001). *Let* $\mathbf{A} \in \mathbb{IR}^{n\times n}$, $R \in \mathbb{R}^{n\times n}, \mathbf{b}, \mathbf{x} \in \mathbb{IR}^n$ *be given, if*

$$R\mathbf{b} + (I - R\mathbf{A})\mathbf{x} \subseteq \text{int}(\mathbf{x}) \tag{14}$$

*then* $R$ *and every matrix* $A \in \mathbf{A}$ *is nonsingular, and*

$$\Sigma(\mathbf{A}, \mathbf{b}) = \{x \in \mathbb{R}^n \mid \exists A \in \mathbf{A},\ \exists b \in \mathbf{b} : Ax = b\} \subseteq \mathbf{x} \tag{15}$$

where $\text{int}(\mathbf{x})$ denotes the interior of $\mathbf{x}$. Expression (15) provides a guaranteed enclosure to the solution set of the linear interval equation $\mathbf{Ax} = \mathbf{b}$. The residual form of (14) can be given in the form [34]:

$$R\mathbf{b} - R\mathbf{A}x_0 + (I - R\mathbf{A})\mathbf{x}^* \subseteq \text{int}(\mathbf{x}^*) \tag{16}$$

where $\mathbf{x} = x_0 + \mathbf{x}^*$ and $x_0$ is a deterministic vector, in particular, $\check{A}^{-1}$ is a good choice for $R$, and $x_0 = R\check{b}$. Assigning $\mathbf{z} = R\mathbf{b} - R\mathbf{A}x_0$, $\mathbf{C} = (I - R\mathbf{A})$, iteration could be constructed [40] in the following form

$$\mathbf{x}^{*n+1} = \mathbf{z} + \mathbf{C}(\boldsymbol{\varepsilon}\mathbf{x}^{*n}) \qquad (\text{for } n = 0, 1, 2, \ldots) \tag{17}$$

and the stopping criteria (16) becomes

$$\mathbf{x}^{*n+1} \subseteq \text{int}(\mathbf{x}^{*n}) \tag{18}$$

In Eq. (17) $\boldsymbol{\varepsilon}$ is a constant interval number, and it serves as an "inflation parameter" to enforce finite termination of the algorithm. If the condition (18) is satisfied after $n$ iterations, then $\mathbf{x}^{*n+1} + x_0$ is an enclosure of the solution set of $\mathbf{Ax} = \mathbf{b}$. The quality (how sharp the enclosure is) of the enclosure provided in (17) depends mainly on the width of the iterative matrix $\mathbf{C}$ and is crucial for the solution convergence the condition that the spectral radius $\rho(|\mathbf{C}|) < 1$ [41].

It is noticeable, however, that the above algorithm is designed for the non-parametric linear interval equations, i.e., the coefficients in the system are assumed to vary independently between their bounds. For many engineering problems, the coefficients have complex dependency relations. For example, the stiffness matrix in FEM is symmetric and positive definite. To account for the dependency effect, one approach is to adapt the solver for non-parametric interval equation. This approach usually involves reformulation of the coefficient matrix and right hand side vector. It has been shown a sharp or even exact enclosure could be obtained in some cases [28, 29, 10].

## 3. Interval finite element analysis

### 3.1. Overestimation in IFEM

A naïve use of interval arithmetic in FEM (naïve IFEM), i.e., replacing deterministic numbers in conventional FEM with interval numbers and solving the system as non-parametric interval equation will result in meaningless wide results [29, 10]. Let us consider the two step bar shown in Fig. 1. The structure is subjected to a unit load at node 3. The conventional FEM gives the equilibrium equations:

$$Ku = p \tag{19}$$

or

$$\begin{pmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{20}$$

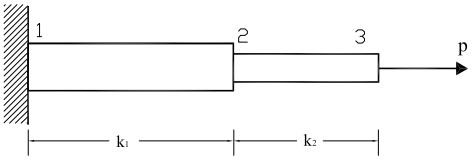



*Figure 1.* Original two-step bar

If the stiffness terms $k_1$ and $k_2$ are introduced as the interval parameters $\mathbf{k_1}$ and $\mathbf{k_2}$, and the interval numbers of $[0.99, 1.01]$ and $[1.98, 2.02]$ are assigned for $\mathbf{k_1}$ and $\mathbf{k_2}$ respectively, the naïve IFEM takes the following form:

$$\begin{pmatrix} [2.97, 3.03] & [-2.02, -1.98] \\ [-2.02, -1.98] & [1.98, 2.02] \end{pmatrix} \begin{pmatrix} \mathbf{u_1} \\ \mathbf{u_2} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{21}$$

Solving (21) using theorem 1, the value of $\mathbf{u_1}$ and $\mathbf{u_2}$ are obtained as:

$$\mathbf{u_1} = [0.876, 1.123]$$

and

$$\mathbf{u_2} = [1.349, 1.651] \tag{22}$$

On the other hand, the exact solution can be achieved by solving (20) symbolically

$$\mathbf{u_1} = \frac{1}{\mathbf{k_1}} = \frac{1}{[0.99, 1.01]} = [0.990, 1.010]$$

and

$$\mathbf{u_2} = \frac{\mathbf{k_1} + \mathbf{k_2}}{\mathbf{k_1 k_2}} = \frac{1}{\mathbf{k_1}} + \frac{1}{\mathbf{k_2}} = \frac{1}{[0.99, 1.01]} + \frac{1}{[1.98, 2.02]} = [1.485, 1.515] \tag{23}$$

The above-presented results for the interval solution of a simple two-step bar problem provide an insight about some aspects of the interval finite element formulation and reveal the most important sources of overestimation. The main two factors that lead for overestimation are the element coupling and multiple occurrences of the interval variables. The four parametric coefficients $\mathbf{k_2}$ in (20) represent the same physical quantity. In the computational process, interval arithmetic treats this physical quantity as four independent interval variables of equal endpoints. Evidently, the same physical quantity cannot have two different values at the same time. It is critical to the formulation of interval finite element analysis, the way the sources of overestimation are handled.

### 3.2. Present Formulation

In order to reduce the overestimation in the interval finite element solutions, the issues of coupling and multiple occurrences of interval variables have to be handled properly.

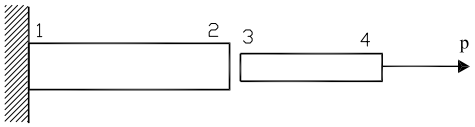



*Figure 2.* EBE two-step bar model

In this work, an element-by-element technique (EBE) is used to circumvent the element coupling problem [29]. The EBE technique can be illustrated by the two-step bar problem in Fig. 1. The elements are disjointed as shown in Fig. 2, thus the system stiffness matrix $K$ takes a block-diagonal structure with dimension of $a \times a$, and $a =$ degrees of freedom per element $\times$ number of elements in the structure. EBE approach adds to the number of degree of freedom (DOF) in the system but avoids the element coupling. The system stiffness matrix $K$ in EBE approach is singular, and Lagrange multiplier method will be used to ensure the compatibility conditions and eliminate the singularity of $K$.

In steady-state analysis, the variational formulation for a deterministic case of a discrete structural model is given in the following form [13, 4]

$$\Pi = \frac{1}{2}u^T K u - u^T p \tag{24}$$

with the conditions

$$\frac{\partial \Pi}{\partial u_i} = 0 \quad \text{for all } i \tag{25}$$

where $\Pi, K, u$, and $p$ are total potential energy, stiffness matrix, displacement vector, and load vector respectively. Assume that we want to impose onto the solution the $m$ linearly independent discrete constraints $Cu - t = 0$ where $C$ and $t$ contain constants. To impose constraints by Lagrange multipliers, we premultiply $Cu - t$ by a row vector $\lambda$ that contains as many Lagrange multipliers $\lambda_i$ as there are constraint equations, and add this to the potential energy (24) [7]. Thus

$$\Pi^* = \frac{1}{2}u^T K u - u^T p + \lambda^T (Cu - t) \tag{26}$$

Invoking the stationarity of $\Pi^*$, i.e., $\partial\Pi^*/\partial u = 0$ and $\partial\Pi^*/\partial\lambda = 0$ we obtain

$$\begin{pmatrix} K & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{pmatrix} p \\ t \end{pmatrix} \tag{27}$$

Considering the compatibility conditions in the present case takes the form $Cu = t = 0$, (27) reduces to

$$\begin{pmatrix} K & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{pmatrix} p \\ 0 \end{pmatrix} \tag{28}$$

Equation (28) stands for the deterministic FEM formulation. In the interval case, where the material and the load are considered to be interval numbers, the deterministic linear equation (28) becomes the interval linear equation

$$\begin{pmatrix} \mathbf{K} & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{p} \\ 0 \end{pmatrix} \tag{29}$$

The coefficient matrix in (29) represents the combination of two parts: the interval element-by-element stiffness matrix $\mathbf{K}$ and the constant deterministic Lagrange multipliers matrix $C$.

The linear interval equation (29) can be solved by theorem 1. However, theorem 1 is used with the implicit assumption that the coefficients of $\mathbf{A}$ are independent among themselves and as well as the components of $\mathbf{b}$ vary independently. Special treatment has to be applied to reduce the dependency effect.

For an element with interval parameters of modulus of elasticity $\mathbf{E}$, the interval parameter could be factorized out from the element stiffness matrix. Consider the $i$th finite element in the structure, assume the uncertainty in the modulus of elasticity is $\boldsymbol{\alpha}_i$, i.e., $\mathbf{E}_i = \check{E}_i(1+\boldsymbol{\alpha}_i)$, the element stiffness matrix $\mathbf{K}_i$ can be expressed in the form $\mathbf{K}_i = \check{K}_i(I + \mathbf{d}_i)$. $\check{K}_i$ is the midpoint of $\mathbf{K}_i$, $I$ is identity matrix, and $\mathbf{d}_i$ is an interval diagonal matrix containing the interval quantity $\boldsymbol{\alpha}_i$. Let us take a truss element for example, its element stiffness matrix can be written as

$$\begin{pmatrix} \frac{\check{E}A}{L} & -\frac{\check{E}A}{L} \\ -\frac{\check{E}A}{L} & \frac{\check{E}A}{L} \end{pmatrix} \left( \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} \boldsymbol{\alpha} & 0 \\ 0 & \boldsymbol{\alpha} \end{pmatrix} \right) \tag{30}$$

Later in the formulation, care will be taken of the multiple occurrence of $\boldsymbol{\alpha}$ in (30).

Following the same procedure for each element, the system stiffness matrix $\mathbf{K}$ constructed by EBE model can be expressed as:

$$\mathbf{K} = \check{K}(I + \mathbf{d}) \tag{31}$$

$\check{K}$ is the midpoint of $\mathbf{K}$, and $\mathbf{d}$ is an interval diagonal matrix; their submatrices are $\check{K}_i$ and $\mathbf{d}_i$, respectively, $i = 1, 2, \ldots, m$, where $m$ is the number of elements in the structure.

Applying this factorization, the system equation (29) can be written as

$$\left( \begin{pmatrix} \check{K} & C^T \\ C & 0 \end{pmatrix} + \begin{pmatrix} \check{K}\mathbf{d} & 0 \\ 0 & 0 \end{pmatrix} \right) \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{p} \\ 0 \end{pmatrix} \tag{32}$$

To utilize the theorem 1 in the present formulation, (32) is introduced as

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{33}$$

with

$$\mathbf{A} = \left( \begin{pmatrix} \check{K} & C^T \\ C & 0 \end{pmatrix} + \begin{pmatrix} \check{K}\mathbf{d} & 0 \\ 0 & 0 \end{pmatrix} \right), \quad \mathbf{x} = \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \mathbf{p} \\ 0 \end{pmatrix} \tag{34}$$

$\mathbf{A}$ can be decomposed furthermore

$$\begin{aligned} \mathbf{A} &= \begin{pmatrix} \check{K} & C^T \\ C & 0 \end{pmatrix} + \begin{pmatrix} \check{K} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{d} & 0 \\ 0 & 0 \end{pmatrix} \\ \mathbf{A} &= \check{A} + S\mathbf{D} \end{aligned} \tag{35}$$

Using the residual form (16) to construct fixed point iteration (17)

$$\mathbf{x}^{*n+1} = \mathbf{z} + \mathbf{C}(\boldsymbol{\varepsilon}\mathbf{x}^{*n}) \qquad (\text{for } n = 0, 1, 2, \ldots) \tag{36}$$

in which $\mathbf{z} = R\mathbf{b} - R\mathbf{A}x_0$, $\mathbf{C} = (I - R\mathbf{A})$, $R = \check{A}^{-1}$, $x_0 = R\check{b}$. By substituting $\mathbf{z}$ and $\mathbf{C}$, the iteration (36) becomes

$$\begin{aligned} \mathbf{x}^{*n+1} &= (R\mathbf{b} - R(\check{A} + S\mathbf{D})x_0) + (I - R(\check{A} + S\mathbf{D}))(\boldsymbol{\varepsilon}\mathbf{x}^{*n}) \\ \mathbf{x}^{*n+1} &= R\mathbf{b} - x_0 - RS\mathbf{D}x_0 - RS\mathbf{D}(\boldsymbol{\varepsilon}\mathbf{x}^{*n}) \\ \mathbf{x}^{*n+1} &= R\mathbf{b} - x_0 - RS\mathbf{D}(x_0 + \boldsymbol{\varepsilon}\mathbf{x}^{*n}) \\ \mathbf{x}^{*n+1} &= R\mathbf{b} - x_0 - RS\mathbf{M}^n\boldsymbol{\delta} \end{aligned} \tag{37}$$

In the problems with deterministic right hand side, we have $\mathbf{b} = \check{b}$, and (37) reduces to a even simpler form

$$\mathbf{x}^{*n+1} = -RS\mathbf{M}^n\boldsymbol{\delta} \tag{38}$$

A key point in the formulation (37) is that $\mathbf{D}(x_0 + \boldsymbol{\varepsilon}\mathbf{x}^{*n})$ has been introduced as $\mathbf{M}^n\boldsymbol{\delta}$ using the $\mathbf{M}$ matrix concept [31, 29] to handle the dependency problem in $\mathbf{D}(x_0 + \boldsymbol{\varepsilon}\mathbf{x}^{*n})$. $\mathbf{M}$ is an interval matrix with the dimensions $(n \times m)$, and $n =$ dimensions of the system. It contains the components from $(x_0 + \boldsymbol{\varepsilon}\mathbf{x}^{*n})$, it will be update with each iteration. $\boldsymbol{\delta}$ is an constant interval vector with the dimensions of $m$, and the components are the uncertainties $\boldsymbol{\alpha}_i$ of the modulus of elasticity of each element, $i = 1, \ldots, m$. Every interval parameter $\boldsymbol{\alpha}_i$ associated with element $i$ occurs only once in $\boldsymbol{\delta}$. The following example shows how generally $\mathbf{D}\mathbf{x}$ could be rewritten as $\mathbf{M}\boldsymbol{\delta}$. Suppose there are two interval parameters $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$

$$\begin{pmatrix} \boldsymbol{\alpha_1} & 0 & 0 & 0 \\ 0 & \boldsymbol{\alpha_1} & 0 & 0 \\ 0 & 0 & \boldsymbol{\alpha_2} & 0 \\ 0 & 0 & 0 & \boldsymbol{\alpha_2} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1 & 0 \\ \mathbf{x}_2 & 0 \\ 0 & \mathbf{x}_3 \\ 0 & \mathbf{x}_4 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha_1} \\ \boldsymbol{\alpha_2} \end{pmatrix} \tag{39}$$

This treatment eliminates the multiple occurrences of $\boldsymbol{\alpha}_i$ in $\mathbf{D}$, thus reduces the overestimation due to dependency problem. If the condition (18) is satisfied after $n$ iterations, the enclosure $\mathbf{x}$ is given by

$$\mathbf{x} = \mathbf{x}^{*n+1} + x_0 \tag{40}$$

The obtained interval vector $\mathbf{x}$ contains two parts: $\mathbf{x} = [\mathbf{u} \quad \boldsymbol{\lambda}]$. The first part, $\mathbf{u}$, is the enclosure for the system's displacement response.

In conventional deterministic FEM, element forces in global coordinate can be calculated by

$$F_i = K_i u_i \tag{41}$$

in which $K_i$, $u_i$ are element stiffness matrix and element nodal displacement in global coordinate. The element forces in local coordinate can be obtained by premultiply a transformation matrix $T_i$. In the interval FEM, however, following the same procedure to calculate element force will bring in overestimation, making the bounds of the element forces unnecessarily wide. The reason is that both $\mathbf{K}_i$ and $\mathbf{u}_i$ are functions of the same interval parameter $\boldsymbol{\alpha}_i$, this multiple occurrences of $\boldsymbol{\alpha}_i$ should be eliminated. In the present IFEM formulation, element forces are calculated from Lagrange multipliers. From (29), it follows

$$\mathbf{K}\mathbf{u} = \mathbf{p} - C^T\boldsymbol{\lambda} \tag{42}$$

Because of its element-by-element structure, (42) produces the element forces directly (in global coordinate). Instead of calculating the left hand side of (42), we will calculate its right hand side to handle dependence problem. Suppose the enclosure $\mathbf{x}$ has been achieved after $n$ iterations, then $\boldsymbol{\lambda}$ can be obtained from $\mathbf{x}$ by a boolean matrix $L$, i.e.,

$$\boldsymbol{\lambda} \;=\; L\mathbf{x} \tag{43}$$

The interval load $\mathbf{p}$ can be rewritten as

$$\mathbf{p} = N\mathbf{b} \tag{44}$$

in which $N$ is a boolean matrix for $\mathbf{p}$. Substitute (37), (43) and (44) into $\mathbf{p} - C^T\boldsymbol{\lambda}$

$$\begin{aligned}
\mathbf{p} - C^T\boldsymbol{\lambda} &= \mathbf{p} - C^T L(\mathbf{x}^{*n+1} + x_0) \\
\mathbf{p} - C^T\boldsymbol{\lambda} &= N\mathbf{b} - C^T L(R\mathbf{b} - RS\mathbf{M}^n\boldsymbol{\delta}) \\
\mathbf{p} - C^T\boldsymbol{\lambda} &= (N - C^T LR)\mathbf{b} + C^T LRS\mathbf{M}^n\boldsymbol{\delta}
\end{aligned} \tag{45}$$

Equation (45) may be premultiplied by a transformation matrix $T$ to get the element forces in local coordinate, i.e.,

$$\mathbf{F} = T(\mathbf{p} - C^T\boldsymbol{\lambda}) = T(N - C^T LR)\mathbf{b} + TC^T LRS\mathbf{M}^n\boldsymbol{\delta} \tag{46}$$

In (46), the multiple occurrences of the interval load $\mathbf{b}$ and interval material parameter $\boldsymbol{\delta}$ has been minimized, and a very sharp results for element force response are obtained.

## 4. Examples

The present interval-based finite element method is illustrated by numerical solutions for three problems with stiffness and load uncertainty.

Table I. Solutions for displacements of two-bay truss

| | $\underline{\mathbf{v}}_2$(m) | $\overline{\mathbf{v}}_2$(m) | $\underline{\mathbf{u}}_4$(m) | $\overline{\mathbf{u}}_4$(m) | $\underline{\mathbf{v}}_4$(m) | $\overline{\mathbf{v}}_4$(m) |
|---|---|---|---|---|---|---|
| Comb$\times 10^{-5}$ | −21.0342 | −18.8416 | 3.7029 | 4.2043 | −1.04833 | −0.92828 |
| Present$\times 10^{-5}$ | −21.0429 | −18.822 | 3.6942 | 4.2075 | −1.04886 | −0.92657 |
| Naïve$\times 10^{-5}$ | −22.7616 | −17.1033 | 3.2221 | 4.6796 | −1.16246 | −0.81297 |
| Present error | 0.04% | 0.10% | 0.23% | 0.08% | 0.05% | 0.18% |
| Naïve error | 8.21% | 9.23% | 12.98% | 11.30% | 10.89% | 12.42% |

Table II. Solutions for axial forces of two-bay truss [compression(−)]

| | $\underline{\mathbf{N}}_2$(kN) | $\overline{\mathbf{N}}_2$(kN) | $\underline{\mathbf{N}}_4$(kN) | $\overline{\mathbf{N}}_4$(kN) | $\underline{\mathbf{N}}_4$(kN) | $\overline{\mathbf{N}}_4$(kN) |
|---|---|---|---|---|---|---|
| Comb | −8.3470 | −7.4613 | 11.4479 | 12.7533 | −14.2587 | −12.7992 |
| Present | −8.3513 | −7.4522 | 11.4390 | 12.7576 | −14.2635 | −12.7891 |
| Naïve | −9.691 | −6.127 | −10.336 | 34.542 | −15.910 | −11.164 |
| Present error | 0.05% | 0.12% | 0.08% | 0.03% | 0.03% | 0.08% |
| Naïve error | 16.10% | 17.88% | 190.29% | 170.85% | 11.58% | 12.78% |

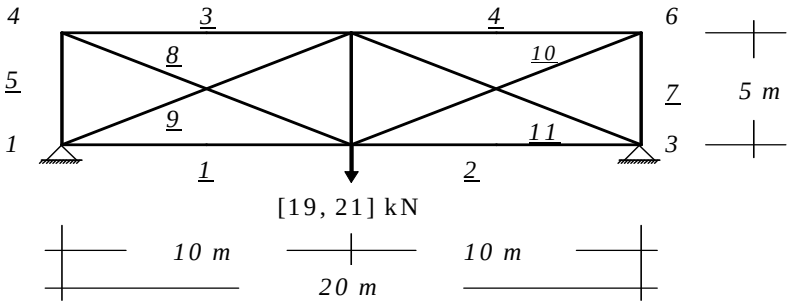



*Figure 3.* Two-bay truss

The first example is a two-bay truss as shown in Fig. 3. The truss is subjected to a concentrated load, applied at the middle lower joint. The variation in the loading is 10% of the midpoint value, and the used interval value is [19, 21]kN. Each element has a cross-sectional area $A_i = 0.01$ m$^2$, and an uncertain modulus of elasticity $\mathbf{E}_i = [199, 201]$ GPa, $i = 1, \ldots, 11$. The modulus of elasticity of each element are assumed to be varied independently.

The results for displacements and element forces are given in Table I and Table II, respectively. The present approach captured the bounds of the system response with errors within a range of 0.03% to 0.23%. However, the naïve IFEM overestimated the bounds of displacements by a range of 8.21% to 12.98%, and the errors escalated to as big as 190% in element force calculation.

Table III. Solutions for displacements of two-bay two-floor frame

| | $\underline{\mathbf{v}}_4$(m) | $\overline{\mathbf{v}}_4$(m) | $\underline{\mathbf{v}}_9$(m) | $\overline{\mathbf{v}}_9$(m) | $\underline{\boldsymbol{\theta}}_9$(rad) | $\overline{\boldsymbol{\theta}}_9$(rad) |
|---|---|---|---|---|---|---|
| Comb$\times 10^{-6}$ | $-6.7640$ | $-6.1548$ | $-13.0697$ | $-11.9207$ | 5.6331 | 6.2691 |
| Present$\times 10^{-6}$ | $-6.7660$ | $-6.1485$ | $-13.0760$ | $-11.9076$ | 5.6219 | 6.2767 |
| Naïve$\times 10^{-6}$ | $-8.7042$ | $-4.2104$ | $-15.3237$ | $-9.6599$ | 3.7305 | 8.1681 |
| Present error | 0.03% | 0.10% | 0.05% | 0.11% | 0.20% | 0.12% |
| Naïve error | 28.68% | 31.59% | 17.25% | 18.97% | 33.78% | 30.29% |

Table IV. Solutions for axial forces ($\mathbf{N}$), shear forces ($\mathbf{V}$) and bending moment ($\mathbf{M}$) of column 1 in two-bay two-floor frame

| | $\underline{\mathbf{N}}_1$(kN) | $\overline{\mathbf{N}}_1$(kN) | $\underline{\mathbf{V}}_1$(kN) | $\overline{\mathbf{V}}_1$(kN) | $\underline{\mathbf{M}}_1$(kN·m) | $\overline{\mathbf{M}}_1$(kN·m) |
|---|---|---|---|---|---|---|
| Comb | $-149.676$ | $-137.3503$ | 5.2608 | 5.8790 | $-14.1977$ | $-12.5250$ |
| Present | $-149.721$ | $-137.2694$ | 5.2408 | 5.8941 | $-14.2345$ | $-12.4775$ |
| Naïve | $-194.393$ | $-93.097$ | $-30.381$ | 41.572 | $-83.892$ | 57.047 |
| Present error | 0.03% | 0.06% | 0.38% | 0.26% | 0.26% | 0.38% |
| Naïve error | 29.88% | 32.22% | 677.50% | 607.13% | 490.89% | 555.47% |

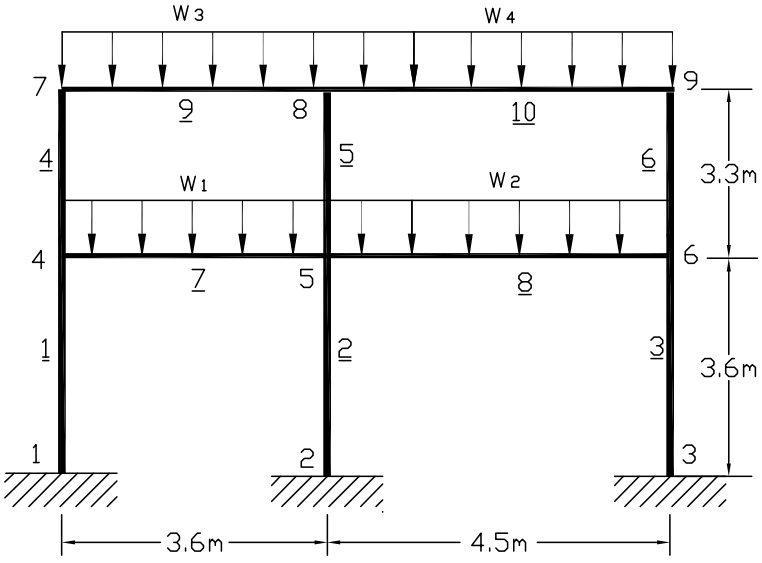



*Figure 4.* Two-bay two-floor frame

The second example is a two-bay two-floor frame as shown in Fig. 4. The columns have cross-sectional area $A_i = 0.4\text{m}^2$, moment of inertia $I_i = 0.036\text{m}^4$, interval modulus

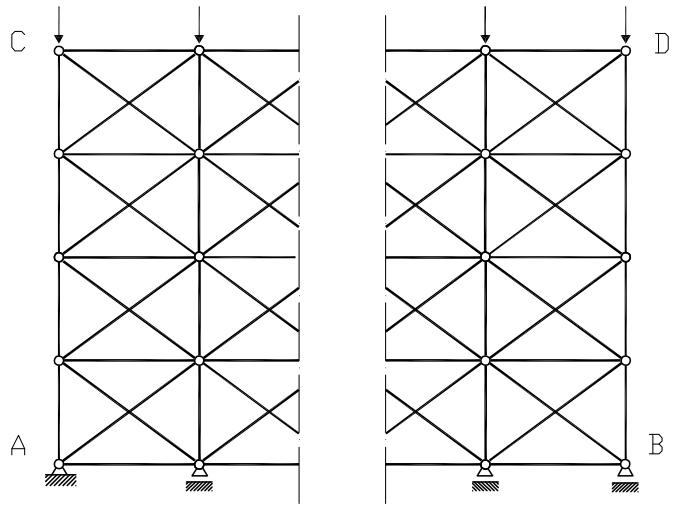



*Figure 5.* Large scale truss

Table V. Solutions for displacement (corner D) of twenty-bay truss

| | $\underline{\mathbf{u}}$(m) | $\overline{\mathbf{u}}$(m) | $\underline{\mathbf{v}}$(m) | $\overline{\mathbf{v}}$(m) |
|---|---|---|---|---|
| Pownuk solution$\times 10^{-5}$ | 7.54868 | 7.84538 | -5.82393 | -5.65726 |
| Present solution$\times 10^{-5}$ | 7.50621 | 7.88574 | -5.84312 | -5.63686 |

of elasticity $\mathbf{E}_i = [199, 201]$GPa, $i = 1, \ldots, 6$. The beams have cross-sectional area $A_i = 0.6\text{m}^2$, moment of inertia $I_i = 0.08\text{m}^4$, interval modulus of elasticity $\mathbf{E}_i = [199, 201]$GPa, $i = 7, \ldots, 10$. The frame is loaded by uniform loads $\mathbf{w}_i$ $(i = 1, 2, 3, 4)$. Each load has 8% uncertainty, and the following data were used: $\mathbf{w}_1 = [24,\ 26]$kN/m, $\mathbf{w}_2 = [24,\ 26]$kN/m, $\mathbf{w}_3 = [48,\ 52]$kN/m, $\mathbf{w}_4 = [48,\ 52]$ kN/m. All the uncertain quantities are varied independently.

The results for displacements of selected nodes are given in Table III. The shear force, axial force and bending moment (at node 4) of column 1 is listed in Table IV. The present algorithm leads to sharp bounds of the exact solution of displacements and element forces, with errors within a range of 0.03% to 0.38%. Whereas, the naïve IFEM solution overestimates the bounds of element forces by 30% to 677%, it could not even get the correct sign for some terms.

To investigate problem size effect on the present formulation, a series of large-scale truss problems were analyzed. The configuration of the structures is shown in Fig. 5. Each element has 1% uncertain modulus of elasticity $\mathbf{E}_i = [2.0895, 2.1105]$GPa, and 1% uncertain cross-sectional area $\mathbf{A}_i = [0.0024875, 0.0025125]\text{m}^2$. Assume all interval parameters are varied independently. Table V lists the displacement results for a 20 bay truss (648 interval parameters). In this example the naïve method failed to converge and the combinatorial method is computationally prohibitive due to the large number of interval parameters.

Table VI. Truss problems with interval parameters

| number of interval parameters | iteration number | iteration time (sec) | total computation time (sec) | variation in typical displacement* |
|---|---|---|---|---|
| 246 | 5 | 0.172 | 1.04 | 2.24% |
| 392 | 5 | 0.453 | 3.97 | 2.47% |
| 648 | 6 | 1.484 | 15.05 | 2.67% |
| 890 | 7 | 3.704 | 40.69 | 2.92% |
| 1192 | 7 | 8.031 | 95.8 | 3.23% |
| 1452 | 8 | 14.329 | 171.7 | 3.38% |
| 1932 | 8 | 26.078 | 381.5 | 3.79% |

*defined as ratio of radius to midpoint value
(horizontal displacement at corner D of the truss)

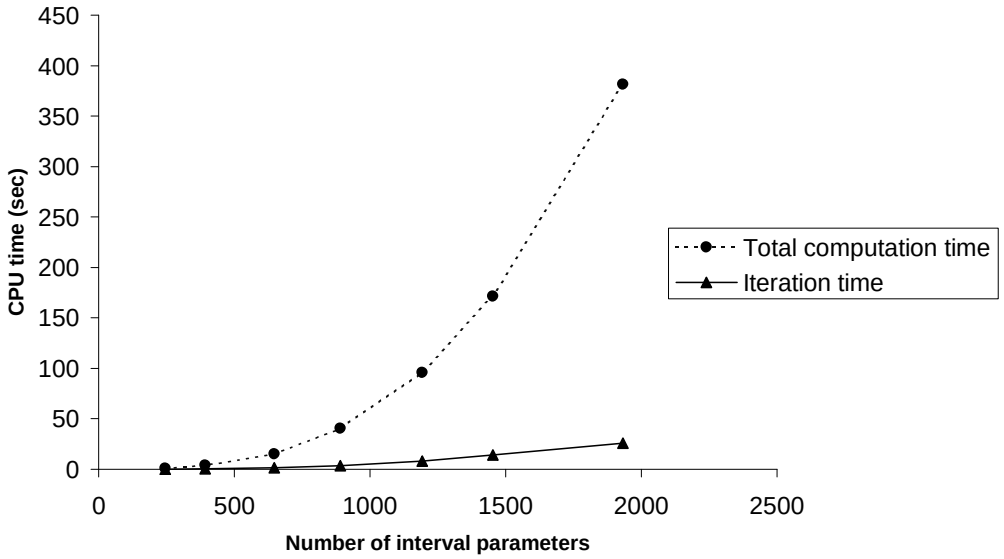



*Figure 6.* Computation time vs problem scale

Pownuk sensitivity analysis method [36] was used as an approximate solution to compare our results with. However, this sensitivity approach is based on the monotonicity assumption and does not provide a solution enclosure, but a good narrower estimate can be obtained when uncertainty is small enough.

Table VI lists the problem size, required number of iterations, iteration CPU time and total computational CPU time. The ratio of the radius of a typical displacement (the horizontal displacement at corner D) to its midpoint value is also listed in Table VI.

The computations were carried out on a PC with Intel Pentium4 2.4GHz CPU with 1GB RAM. The calculations show that the sharpness of the results maintains the same level despite of the increase of the problem size. Fig. 6 shows the relationship of problem size vs iteration CPU time, and problem size vs total computational CPU time. It can be seen the computational time does not increase exponentially with the increase of the problem size. In the current stage, most computation time is spent on calculating the preconditioning matrix $R$. It is important to note that system (29) is very sparse, and we expect a major computation time reduction when the sparsity is fully exploited. This will be a future work.

## 5. Conclusion

In this paper a new interval finite element formulation is presented. Uncertain loads and stiffness are introduced as interval numbers. The major difficulty associated with the IFEM is the overestimation due to dependency effect: the computed range of the response is much wider than the actual range. For engineering application, the physical nature of the problem must be considered to control the overestimation. In the present approach an element-by-element technique is used and the compatibility conditions are ensured by the Lagrange multiplier method. The resulting linear interval equation is solved using the Brouwer's fixed point theory with Krawczyk's operator and a newly developed overestimation control. The numerical examples show the naïve interval FEM produces meaningless wide results. The present approach, however, eliminates most sources of overestimation and a very sharp enclosure for the system's displacement and forces are obtained simultaneously and have the same level of accuracy. The numerical examples also illustrated the present formulation's scalability.

## References


1. Akpan, U. O., T. S. Koko, I. R. Orisamolu, and B. K. Gallant: 2001, 'Practical fuzzy finite element analysis of structures'. *Finite Elem. Anal Des* **38**, 93–111.
2. Alefeld, G. and J. Herzberger: 1983, *Introduction to interval computations.* New York: Academic Press.
3. Apostolatos, N. and U. Kulisch: 1968, 'Grundzüge einer Intervallrechtung für Matrizen und einige Anwwendungen'. *Elektron. Rechenanlagen* **10**, 73–83. (in German).
4. Bathe, K.: 1996, *Finite element procedures.* Upper Saddle River, NJ: Prentice-Hall.
5. Ben-Haim, Y. and F. Elishakoff: 1990, *Convex models of uncertainty in applied mechanics.* Amsterdam: Elsevier Science.
6. Chen, S. H., H. D. Lian, and X. W. Yang: 2002, 'Interval static displacement analysis for structures with interval parameters'. *Int. J Numer. Methods Engrg.* **53**, 393–407.
7. Cook, R. D., D. S. Malkus, and M. E. Plesha: 1989, *Concepts and applications of finite element analysis.* John Wiley & Sons.
8. de Cooman, G., D. Ruan, and E. E. Kerre (eds.): 1995, *Foundations and applications of possibility theory.* Singapore: World Scientific.
9. Dempster, A. P.: 1967, 'Upper and lower probabilities induced by a multi-valued mapping'. *Ann. Mat. Stat.* **38**, 325–339.

10. Dessombz, O., F. Thouverez, J.-P. Laîné, and L. Jézéquel: 2001, 'Analysis of mechanical systems using interval computations applied to finite elements methods'. *J. Sound. Vib.* **238**(5), 949–968.
11. Ferson, S. and L. R. Ginzburg: 1996, 'Different methods are needed to propagate ignorance and variability'. *Reliab. Engng. Syst. Saf.* **54**, 133–144.
12. Ferson, S., V. kreinovich, L. Ginzburg, D. S. Myers, and K. Sentz: 2003, 'Constructing probability boxes and Dempster-Shafer structures'. Technical Report SAND2002-4015, Sandia National Laboratories.
13. Gallagher, R. H.: 1975, *Finite element analysis fundamentals*. Englewood Cliffs, N.J.: Printice Hall.
14. Ganzerli, S. and C. P. Pantelides: 1999, 'Load and resistance convex models for optimum design'. *struct. Optim.* **17**, 259–268.
15. Gay, D. M.: 1982, 'Solving interval linear equations'. *SIAM J. Numer. Anal.* **19**(4), 858–870.
16. Hansen, E.: 1965, 'Interval arithmetic in matrix computation'. *SIAM J. Numer. Anal.* **I**(2), 308–320.
17. Hansen, E.: 1992, *Global optimization using interval analysis*. New York: Marcel Dekker.
18. Jansson, C.: 1991, 'Interval linear system with symmetric matrices, skew-symmetric matrices, and dependencies in the right hand side'. *Computing* **46**, 265–274.
19. Kendall, D. G.: 1974, *Foundations of a theory of random sets*, pp. 322–376. New York: Wiley.
20. Koyluoglu, U., S. Cakmak, N. Ahmet, and R. K. Soren: 1995, 'Interval algebra to deal with pattern loading and structural uncertainty'. *J. Engrg. Mech.* **121**(11), 1149–1157.
21. Mayer, O.: 1970, 'Algebraische und metrische Strukturen in der Intervallrechung und eingine Anwendungen'. *Computing* **5**, 144–162. (in German).
22. McWilliam, S.: 2000, 'Anti-optimisation of uncertain structures using interval analysis'. *Comput. Struct.* **79**, 421–430.
23. Melchers, R. E.: 1999, *Structural reliability analysis and prediction*. West Sussex, England: John Wiley & Sons, 2 edition.
24. Möller, B., W. Graf, and M. Beer: 2000, 'Fuzzy structural analysis using level-optimization'. *Comput. Mech.* **26**(6), 547–565.
25. Moore, R. E.: 1966, *Interval Analysis*. Englewood Cliffs, N. J.: Prentice-Hall, Inc.
26. Moore, R. E.: 1979, *Methods and applications of interval analysis*. Philadelphia: SIAM.
27. Muhanna, R. L. and R. L. Mullen: 1995, 'Development of interval based methods for fuzziness in continuum mechanics'. In: *Proc. ISUMA-NAFIPS'95*. pp. 23–45.
28. Muhanna, R. L. and R. L. Mullen: 1999, 'Formulation of fuzzy finite element methods for mechanics problems'. *Compu.-Aided Civ. Infrastruct. Engrg.* **14**, 107–117.
29. Muhanna, R. L. and R. L. Mullen: 2001, 'Uncertainty in mechanics problems-interval-based approach'. *J. Engrg. Mech.* **127**(6), 557–566.
30. Mullen, R. L. and R. L. Muhanna: 1996, 'Structural analysis with fuzzy-based load uncertainty'. In: *Proc. 7th ASCE EMD/STD Joint Spec. Conf. on Probabilistic Mech. and Struct. Reliability*. Mass., pp. 310–313.
31. Mullen, R. L. and R. L. Muhanna: 1999, 'Bounds of structural response for all possible loadings'. *J. Struct. Engrg., ASCE* **125**(1), 98–106.
32. Neumaier, A.: 1987, 'Overestimation in linear interval equations'. *SIAM J. Numer. Anal.* **24**(1), 207–214.
33. Neumaier, A.: 1989, 'Rigorous sensitivity analysis for parameter-dependent systems of equations'. *J. Math. Anal. Appl.* **144**, 14–25.
34. Neumaier, A.: 1990, *Interval methods for systems of equations*. Cambridge University Press.
35. Pantelides, C. P. and S. Ganzerli: 2001, 'Comparison of fuzzy set and convex model theories in structural design'. *Mech. Systems Signal Process.* **15**(3), 499–511.
36. Pownuk: 2004, 'Calculation of the extreme values of displacements in truss structures with interval parameters'. http://s212.bud.polsl.gliwice.pl/ andrzej/php/apdl2interval/apdl2interval_ init.php.
37. Rao, S. S. and L. Berke: 1997, 'Analysis of uncertain structural systems using interval analysis'. *AIAA J.* **35**(4), 727–735.
38. Rao, S. S. and L. Chen: 1998, 'Numerical solution of fuzzy linear equations in engineering analysis'. *Int. J. Numer. Meth. Engng.* **43**, 391–408.

39. Rohn, J.: 1995, 'Linear interval equations: computing sufficiently accurate enclosures is NP-Hard'. Technical Report 621, Institute of computer science, Academy of Sciences of the Czech Republic.
40. Rump, S. M.: 1983, 'Solving algebraic problems with high accuracy'. In: U. Kulisch and W. Miranker (eds.): *A new approach to scientific computation.* New York: Academic Press.
41. Rump, S. M.: 1992, 'On the solution of interval linear systems'. *Computing* **47**, 337–353.
42. Rump, S. M.: 2001, 'Self-validating methods'. *Linear Algebra Appl.* **324**, 3–13.
43. Sentz, K. and S. Ferson: 2002, 'combination of evidence in Dempster-Shafer theory'. Technical Report SAND2002-0835, Sandia National Laboratories.
44. Shafer, G.: 1976, *A mathematical theory of evidence.* Princeton,N.J.: Princeton University Press.
45. Sun microsystems: 2002, 'Interval arithmetic in high performance technical computing'. Sun microsystems. (A White Paper).
46. Walley, P.: 1991, *statistical reasoning with imprecise probabilities.* London: Chapman and Hall.
47. Zadeh, L. A.: 1978, 'Fuzzy Sets as a Basis for a Theory of Possibility'. *Fuzzy Sets and Systems* **1**, 3–28.

# Uncertainty in Thermal Basin Modeling: An Interval Finite Element Approach

Sebastião C. Pereira
*PETROBRAS R&D Center, Rio de Janeiro, RJ, Brazil*

Ulisses T. Mello
*IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, U.S.A.*

Nelson M. A. D. Ebecken
*COPPE, Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil*

Rafi L. Muhanna
*Georgia Institute of Technology, USA*

**Abstract.** Uncertainty assessment in basin modeling and reservoir characterization is traditionally treated by geostatistical methods which are normally based on stochastic probabilistic approaches. In this talk, an alternative interval-based approach will be present. A solution for the transient heat conduction in sedimentary basins will be introduced using an interval finite element approach. For this purpose, a novel formulation is developed to deal with both the special interval arithmetic properties and the transient term in the differential equation governing heat transfer. In this formulation, the "stiffness" matrix resulting from the discretization of the heat conduction equation is assembled using an element-by-element technique in which the finite elements are globally independent and lagrange multipliers are used to enforce continuity. This formulation is suggested as an alternative to traditional Monte Carlo method, where repetitive simulations are required to handle uncertainty and worst case system response is underestimated. The newly developed technique is applied to a one-dimensional thermal basin simulation to assess its potential and limitations. Numerical results will be introduced and their quality assessed.

## 1 - Introduction

Determinist numerical simulations are normally used to predict the behavior of geological systems. The predicted behavior or performance is normally used for risk assessment. A well known limitation of determinist simulations is that they provide a single set of results that do not convey information about the uncertainty associated with input parameters or coefficients. To overcome this limitation, uncertainty assessment is typically coupled with stochastic probabilistic approaches in which the input simulation parameter set is stochastically defined and multiple simulations are executed to estimate the uncertainty associated with a probabilistic distribution of the input parameter set space. While effective, this approach is rather expensive computationally. In addition, in deterministic numerical simulations, such as the traditional finite element approach, all the parameters are assumed to be precisely known. However, frequently in basin modeling this is not the case, since imprecise or fuzzy information may be present in the geometry, age, and material properties of the basin. Stochastic or probabilistic approaches have been developed to account for this kind of uncertainties. However, in these approaches material properties are normally treated as

random variables despite the fact that geological processes are not controlled by random phenomena. These considerations have led us to the consideration of *possibility* rather than *probability* and in this work, we present an alternative to the stochastic probabilistic approaches which is based on the interval mathematics to assess uncertainty. Here, we have applied the newly developed techniques to a one-dimensional thermal basin simulation to assess their potential and limitations.

Interval mathematics is a generalization in which interval numbers replace real, or crisp, numbers, interval arithmetic replaces real arithmetic and interval analysis replace real analysis (Hansen, 2000). Fuzzy numbers can be represented by confidence intervals and its calculations can be performed through interval mathematics. For a introduction to interval and fuzzy arithmetic we recommend reading Moore (1962), Neumaier (1990) and Kaufman & Gupta (1991). The Appendix A summarizes the most common interval operations and its properties. The numerical solution of partial differential equations (PDEs), governing the heat and fluid transfer in porous media, to be unconditionally stable normally requires that its time discretization is implicit. This stable solution leads to a set of simultaneous linear system of equations. When fuzzy numbers are used to represent material properties such as thermal conductivity, the description of the resulting linear system is no longer crisp, but is ambiguous or imprecise. This requires the linear system to be solved using a non-classic approach and unfortunately, there are few and efficient methods described in the literature to solve these systems. It is noteworthy that the solution of interval linear system is combinatorial in nature and the result is a convex hull bounding all possible solutions for the imprecise input system. The combinatorial method is the most accurate method for solving interval systems of equations. In this method, the system of Equation is solved for all combinations of interval numbers using their upper and lower bounds. Needless to say, that the combinatorial method is extremely expensive and cannot solve large systems. However, it can be used to evaluate the accuracy of the other methods for small systems. A method is said to overestimate the results when they are larger than hull estimated by the combinatorial method.

The classic approaches to solve interval linear systems, such as Gaussian elimination, fails to solve interval systems because of some special properties of interval arithmetic, such as the subcancel property, in which one number minus itself is not zero, and one number divided by itself is not the unit. These properties require extensive modifications in the algorithms to solve the systems. In addition, the overestimation induced by variable repetition in a mathematical expression makes especially difficult to deal with interval numbers. In the literature, several papers have described methods to solve interval linear systems of Equations. For example, Neumaier (1990) proposed a preconditioning technique to Gauss Elimination and Gauss Seidel Method. Unfortunately, these techniques generate overestimation in the solution and they also fail to solve both large problems and systems with large interval widths. Rao (1995) discussed an optimization method using the Powell algorithm to solve interval linear systems. This method is very expensive and frequently generates results in which the range is too tight, underestimating the results. Recently Muhanna (2001) presented an interval-based finite element formulation which makes use of an Element-by-Element (EBE) technique to calculate the solution of steady-state problems in mechanics which avoids most sources of overestimations and computes a very sharp solution hull. In this paper, we extend the EBE technique to handle time dependent problems. In addition, we briefly describe a C++ library that we implemented with the combinatorial, preconditioned Gauss elimination, Gauss Seidel, and the Powel methods to solve interval linear systems of Equations.

## 2 - Heat transfer in sedimentary basins

To illustrate the interval-based uncertainty analysis to basin modeling, we discuss the effect of uncertainty in thermal conductivity of rocks on the predicted temperature evolution of a given basin.

The thermal conductivity of sedimentary rocks is a material property which is well know to vary largely in nature and the impact of its variation can be evaluated with interval mathematics. The thermal conductivity is key parameter in the partial differential equation that governs the heat transfer in compacting porous sediments (Mello, 1994):

$$\left[\frac{\phi}{(1-\phi)}\rho_f c_f + \rho_s c_s\right]\frac{\partial T}{\partial t} = \frac{\partial}{\partial Z}\left[k(1-\phi)\frac{\partial T}{\partial Z}\right] - \rho_f c_f q_f \frac{\partial T}{\partial Z} + \frac{Q_h}{(1-\phi)} \quad , \tag{2.1}$$

where:

$\phi$ = Sediment porosity;

$\rho_f$ = Pore fluid density, $(kg/m^3)$;

$c_f$ = Pore fluid specific heat, $(J/kg)$;

$\rho_s$ = Solid grain density, $(kg/m^3)$;

$c_s$ = Solid grain specific heat, $(J/kg)$;

$T$ = Temperature, $(^oC)$;

$t$ = Time, $(m)$;

$k$ = Bulk thermal conductivity of sediments, $(W/m\ ^oC)$;

$Z = \int_{z1}^{z2} (1-\phi)dz$ = Fully compacted depth (m);

$q_f$ = Darcian fluid velocity, $(m/s)$ ;

$Q_h$ = Heat source/sink, $(J/m)$.

Equation (2.1) describes the transfer of heat within the sediments via diffusion and advection processes. The bracketed term in the left-hand side of Equation (2.1) is the sediment bulk heat capacity. The first term in the right hand side describes the conduction of heat, the second term represents the advection due to fluid carriage of heat and the final term accounts for the heat gained or lost sources or sinks. The respective essential and natural boundary conditions used to solve Equation (2.1) are:

$$T(S_z) = T_{surf}(t) \quad , \tag{2.2}$$

$$k(1-\phi)\left.\frac{\partial T}{\partial Z}\right|_{z=0} = Q(t) \quad , \tag{2.3}$$

where $T_{surf}$ is the temperature at interface water-sediment ($S_z$) and $Q(t)$ is the basal heat flux entering the basin. This heat flux can be calculated by the degree of crustal and lithospheric mantle thinning as described by McKenzie (1978):

$$Q(t) = \frac{k_a T_m}{a}\left[1 + \frac{\beta}{\pi}\sum_{n=1}^{\infty} sin\left(\frac{n\pi}{\beta}\right) exp\left(\frac{-n^2\pi^2 kt}{a^2}\right)\right] \quad , \tag{2.4}$$

where:

$k_a$ = Lithosphere thermal conductivity (W/m°C);

$T_m$ = Mantle temperature (°C);

$a$ = Lithosphere thickness (m);

$k = (k_a / \rho c)$ =Thermal diffusivity ($m^2$/s);

$\tau = \left(a^2 / \pi^2 k\right)$ =Thermal decline (s);

$\beta =$ Lithospheric extension factor.

For deterministic solutions, we have used the data displayed in Table 1 and 2. In Table 1 shows the physical properties for typical sediments are listed and Table 2 presents typical lithospheric parameters.

| Lithology | $\phi_0$ | $b$ | $\rho$ | $k$ | $C$ |
|---|---|---|---|---|---|
| | | | | | |
| Shale | 63.0 | 0.58 | 2.68 | 1.5 | 950.0 |
| Silt | 56.0 | 0.39 | 2.68 | 2.0 | 860.0 |
| Sandstone | 50.0 | 0.50 | 2.65 | 3.0 | 750.0 |
| Limestone | 60.0 | 0.44 | 2.72 | 2.5 | 860.0 |
| Chalk | 70.0 | 0.71 | 2.67 | 3.5 | 800.0 |
| Salt | 0.05 | 0.005 | 2.20 | 5.5 | 854.0 |
| Basalt | 5.00 | 0.0 | 2.85 | 2.0 | 775.0 |

Table 1 - Physical properties of selected lithologies.

where:

$\phi_0 =$ Surface porosity;

$b =$ Porosity decay coefficient (1/km);

$\rho =$ Density (g/$cm^3$);

$k =$ Thermal conductivity (W/m°C);

$C =$ Heat capacity (J/kg°C).

| | |
|---|---|
| Thermal diffusivity | 0.008 ($m^2$/s) |
| Thermal expansion coefficient | 0.000034 (1/°C) |
| Crustal thickness | 31200.0 (m) |
| Lithosphere thickness | 125000.0 (m) |
| Temperature at the lithosphere base | 1333.0 (°C) |
| Mantle density | 3330 (kg/$m^3$) |
| Crustal density | 2800 (kg/$m^3$) |
| Steady-state heat flow | 41.84 (mW/$m^2$) |

Table 2 – Lithosphere properties

### 3 – Traditional finite element formulation for the transient heat conduction

Before we discuss the interval finite element formulation, we briefly review the traditional deterministic Galerkin finite element formulation for solving Equation (2.1). By neglecting the advection term in (2.1), this equation then describes a general diffusion problem which can be stated as an initial-boundary value problem (IBV). It can be expressed in a general form by the following differential equation (Burnett, 1987):

$$\mu(x)\frac{\partial U(x,t)}{\partial t} - \frac{\partial}{\partial x}\left(\alpha(x)\frac{\partial U(x,t)}{\partial x}\right) = f(x,t). \tag{3.1}$$

When Equation (3.1) is applied to heat conduction, its symbols represent:

$U(x,t) = T(x,t)$ = Temperature;
$\mu(x) = \rho(x)c(x)$ = Heat storage;
$\alpha(x) = k(x)$ = Thermal conductivity;
$f(x,t) = Q(x,t)$ = Heat source;

$\rho(x) =$ Density;

$c(x) =$ Specific heat.

IBV problems consists of finding $U = U(x)$ satisfying (3.1) $\forall x \in \Omega$ and the prescribed boundary conditions (BCs) which are assumed take the form:

$$U(x) = g(x) \qquad \forall x \in \Gamma_g \tag{3.2}$$

$$\alpha\frac{\partial U}{\partial x} = h(x) \qquad \forall x \in \Gamma_h \tag{3.3}$$

where $\Omega$ is the domain and $\Gamma$ the boundary, $g$ and $h$ are given functions, $g$ is the essential, or Dirichlet BC, and $h$ is the natural, or Newman BC.

Approximating $U = Na$ and solving (3.1) by the Galerkin method using the implicit time method results into the following element equations:

$$\left[k_{eff}\right]\{a\}_n = \{f_{eff}\} \tag{3.4}$$

where:

$$\left[k_{eff}\right] = \frac{1}{\Delta t_n}[c] + [k]; \tag{3.5}$$

$$\{f_{eff}\} = +\{f\}_n + \left(\frac{1}{\Delta t_n}[c]\right)\{a\}_{n-1}; \tag{3.6}$$

$\Delta t_n$ = time step n;
$\{a\}_{n-1} =$ solution for the time-step n-1;

$N =$ shape functions.

The coefficients of the element "stiffness" ( $k$ ) and accumulation ( $c$ ) matrices are given by:

$$k = \begin{bmatrix} \frac{\alpha}{L} & -\frac{\alpha}{L} \\ -\frac{\alpha}{L} & \frac{\alpha}{L} \end{bmatrix} \qquad c = \begin{bmatrix} \frac{1}{3}\mu L & \frac{1}{6}\mu L \\ \frac{1}{6}\mu L & \frac{1}{3}\mu L \end{bmatrix} \tag{3.7}$$

where $L$ is the length of 1D finite element. In the steady-state heat conduction problem, the element 'stiffness" is only composed by the $k$ matrix. In a transient heat conduction problem, an additional accumulation matrix $c$ is added to form the effective element "stiffness" $k_{eff}$. The accumulation matrix represents the element heat storage capacity, $\mu$. For the global solution of the IBV problem, the elements are assembled in a global stiffness $K_{eff}$ and in a global $F_{eff}$ vector. Then a linear system of equations is solved for the primary variable $a$:

$$\left[K_{eff}\right]\{a\}_n = \left\{F_{eff}\right\} \quad , \tag{3.8}$$

$$\left[K_{eff}\right] = \underset{e=1}{\overset{nel}{A}} k_{eff} \; , \; \left\{F_{eff}\right\} = \underset{e=1}{\overset{nel}{A}} f_{eff} \; ,$$

symbol $A$ represents the assembly operation.

## 4 – The Element-by-Element (EBE) Formulation with Element Overlap

A straightforward way to transform the traditional finite element formulation into a interval finite element formulation is to replace the real "stiffness" matrix by an interval, or fuzzy, stiffness matrix and solve the resulting interval linear system. Unfortunately, the direct solution of this linear system of Equations can produce overestimated results and arithmetic operations problems (Kulpa, 1998). This occurs due to the large number of arithmetic operations and the width of the intervals numbers during the solution process. Muhanna (2001) proposed an EBE finite element method for steady-state problems that avoids a great number of these operations. In his method, the assembly operation is modified by keeping the elements effectively disconnected and enforcing continuity in the mesh by using Lagrangian constraints. Using this approach, the stiffness matrix can be factored in two matrices: one interval diagonal matrix and another real banded matrix. The inversion of these matrices is done separately and involves very few interval arithmetic operations because the inversion of the diagonal matrix requires a single interval division per row. In spite of this advance, this method is not directly applied to transient problems in which the stiffness matrix has the contribution of the heat capacity term (C matrix). In this case, the resulting stiffness matrix cannot be factored using the same algorithm that was described for steady-state case.

In the sequence, we discuss a new formulation to extent the EBE formulation to solve interval transient heat conduction problems. The goal of this formulation is the same of the steady-state formulation described previously, that is, to factor the global stiffness matrix into two matrices, one interval diagonal and another banded real to reduce the number of arithmetic operations involving interval numbers during the solution of the linear system.

Figure 1 illustrates the EBE formulation using a one-dimensional thermal basin modeling example. In this example, there are three stratigraphic horizons ( $H_i, i = i \cdots 3$ ) and two layers

($C_i, i = \cdots 2$) represented by the traditional finite element mesh with three nodes (horizons) and two elements (layers). The heat flux is specified (natural condition) at bottom ($Q(t)$) and the surface temperature specified (essential boundary) at the top ($T_{surf}$). Each layer $i$, has its physical properties: conductivity ($\alpha_i$), heat capacity ($\mu_i$) and its thickness ($L_i$).

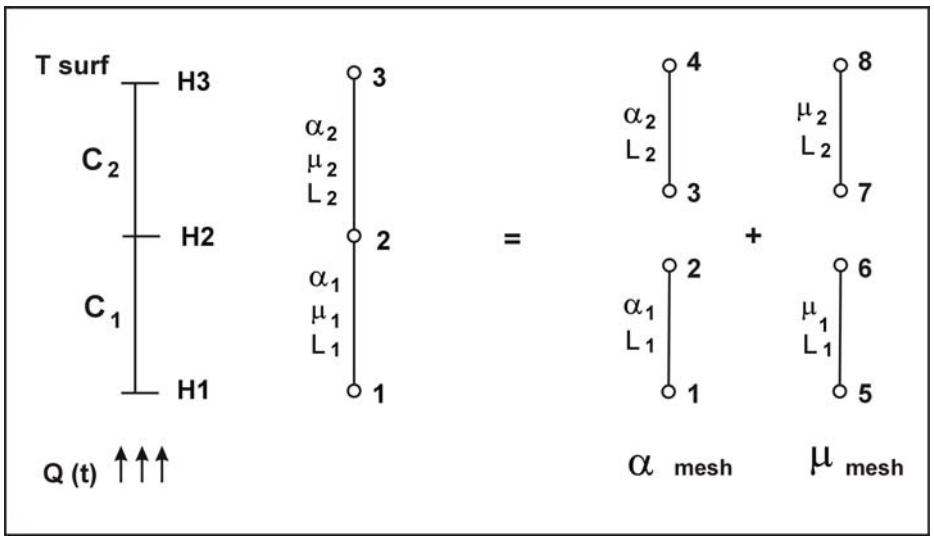



Figure 1 – EBE scheme for a mesh with 2 elements. The nodes are split and renumbered producing 4 overlaps elements.

The overlapping elements are in the right (Fig. 1), there is essentially one mesh for the conductivity ($\alpha$ mesh) overlapping with another one for the heat capacity ($\mu$ mesh). The thermal conductivity and the heat capacity are intervals numbers represented by $\alpha = [\underline{\alpha}, \overline{\alpha}]$ and the $\mu = [\underline{\mu}, \overline{\mu}]$. In the interval notation, the underscore and overscore symbols represent the interval lower bound and upper bound respectively. After the mesh split, the nodes are duplicated, node 1 becomes 1 and 5; node 2 becomes 2, 3, 6, 7; and node 3 becomes 4, 8. The mesh compatibility, or the result continuity, has to be enforced and thus constraining equations must be satisfied: $T_1 = T_5$; $T_2 = T_3 = T_6 = T_7$; $T_4 = T_8$, where $T$ is temperature.

The global linear system of equations with the node compatibilities using the EBE formulation with overlap for this mesh is:

$$
\left[\begin{array}{cccccccc|ccccc}
\frac{\alpha_1}{L_1} & -\frac{\alpha_1}{L_1} & & & & & & & 1 & 0 & 0 & 0 & 0 \\
-\frac{\alpha_1}{L_1} & \frac{\alpha_1}{L_1} & & & & & & & 0 & 1 & 1 & 1 & 0 \\
 & & \frac{\alpha_2}{L_2} & -\frac{\alpha_2}{L_2} & & 0 & & & 0 & -1 & 0 & 0 & 0 \\
 & & -\frac{\alpha_2}{L_2} & \frac{\alpha_2}{L_2} & & & & & 0 & 0 & 0 & 0 & 1 \\
 & & & & \frac{\mu_1 L_1}{3\Delta t} & \frac{\mu_1 L_1}{6\Delta t} & & & -1 & 0 & 0 & 0 & 0 \\
 & & 0 & & \frac{\mu_1 L_1}{6\Delta t} & \frac{\mu_1 L_1}{3\Delta t} & & & 0 & 0 & -1 & 0 & 0 \\
 & & & & & & \frac{\mu_2 L_2}{3\Delta t} & \frac{\mu_2 L_2}{6\Delta t} & 0 & 0 & 0 & -1 & 0 \\
 & & & & & & \frac{\mu_2 L_2}{6\Delta t} & \frac{\mu_2 L_2}{3\Delta t} & 0 & 0 & 0 & 0 & -1 \\
\hline
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\cdot
\left\{\begin{array}{c} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \\ \hline \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{array}\right\}
=
\left\{\begin{array}{c} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \\ P_8 \\ \hline 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}\right\}
\tag{4.1}
$$

or more compactly:

$$
\begin{bmatrix} \boldsymbol{K} & \boldsymbol{C}^T \\ \boldsymbol{C} & \boldsymbol{0} \end{bmatrix}
\begin{Bmatrix} \boldsymbol{T} \\ \boldsymbol{\lambda} \end{Bmatrix}
=
\begin{Bmatrix} \boldsymbol{P} \\ \boldsymbol{0} \end{Bmatrix}
\tag{4.2}
$$

where $\boldsymbol{K}$ is the stiffness matrix , $\boldsymbol{C}$ is the compatibility node matrix, $\boldsymbol{T}$ the unknown temperature $\boldsymbol{\lambda}$ the vector of Lagrange multipliers, and $\boldsymbol{P}$ is the heat source. The first two blocks in the diagonal of the stiffness matrix in Equation (4.1) are from to the thermal conductivity mesh. Third and fourth bocks are from to heat capacity elements. The $\boldsymbol{0}$, $\boldsymbol{1}$ and $\boldsymbol{-1}$ coefficients are from the constraining equations used to enforce the nodes compatibilities. The linear system of equation above can be written as:

$$
\boldsymbol{KT} + \boldsymbol{C}^T \boldsymbol{\lambda} = \boldsymbol{F} \tag{4.3}
$$

$$
\boldsymbol{CT} = \boldsymbol{0} \tag{4.4}
$$

The stiffness matrix ( $\boldsymbol{K}$ ) can be written as the product of two matrices:

$$
\boldsymbol{K} = \boldsymbol{DS} \quad , \tag{4.5}
$$

where $\boldsymbol{D}$ is an interval diagonal matrix and $\boldsymbol{S}$ is a banded real matrix:

$$D = \begin{pmatrix} \frac{\alpha_1}{L_1} & & & & & & & \\ & \frac{\alpha_1}{L_1} & & & & & & \\ & & \frac{\alpha_2}{L_2} & & & 0 & & \\ & & & \frac{\alpha_2}{L_2} & & & & \\ & & & & \frac{\mu_1 L_1}{\Delta t} & & & \\ & & 0 & & & \frac{\mu_1 L_1}{\Delta t} & & \\ & & & & & & \frac{\mu_2 L_2}{\Delta t} & \\ & & & & & & & \frac{\mu_2 L_2}{\Delta t} \end{pmatrix} \tag{4.6}$$

$$S = \begin{pmatrix} 1 & -1 & & & & & & \\ -1 & 1 & & & & 0 & & \\ & & 1 & -1 & & & & \\ & & -1 & 1 & & & & \\ & & & & \frac{1}{3} & \frac{1}{6} & & \\ & & & & \frac{1}{6} & \frac{1}{3} & & \\ & 0 & & & & & \frac{1}{3} & \frac{1}{6} \\ & & & & & & \frac{1}{6} & \frac{1}{3} \end{pmatrix} \tag{4.7}$$

Since the diagonal matrix $D$ has the interval numbers ($\alpha$ and $\mu$), its inverse is obtained trivially. The $S$ matrix is block diagonal and singular (the second line equals to the first multiplied by $-1$) and, thus, cannot be directly inverted.

Substituting (4.5) in (4.3) results into:

$$DST = P - C^T\lambda \tag{4.8}$$

Multiplying (4.4) by $DC^T$, and adding the result to (4.8), after some algebraic operations we obtain:

$$D\left(ST + C^T CT\right) = P - C^T\lambda \tag{4.9}$$

If we define $Q = C^T C$ and $R = S + Q$ we have:

$$\boldsymbol{DRT = P - C^T \lambda} \tag{4.10}$$

Finally, the temperature solution vector can be obtained from (4.10) by:

$$\boldsymbol{T = R^{-1} D^{-1} \left( P - C^T \lambda \right)} \tag{4.11}$$

This Equation can be further simplified by defining the vector:

$$\boldsymbol{p = ( P - C^T \lambda ) = \left\{ p_1,\ p_2,\ p_3, p_4,\ p_5, p_6,\ p_7, p_8 \right\}^T}, \tag{4.12}$$

then Equation (4.11) becomes:

$$\boldsymbol{T = R^{-1} M \delta}\ , \tag{4.13}$$

where:

$$\boldsymbol{M} = \begin{bmatrix} p_1 & 0 & 0 & 0 \\ p_2 & 0 & 0 & 0 \\ 0 & p_3 & 0 & 0 \\ 0 & p_4 & 0 & 0 \\ 0 & 0 & p_5 & 0 \\ 0 & 0 & p_6 & 0 \\ 0 & 0 & 0 & p_7 \\ 0 & 0 & 0 & p_8 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\delta} = \begin{Bmatrix} \frac{L_1}{\alpha_1} \\ \frac{L_2}{\alpha_2} \\ \frac{\Delta t}{\mu_1 L_1} \\ \frac{\Delta t}{\mu_2 L_2} \end{Bmatrix} \tag{4.14}$$

The matrix $\boldsymbol{M}$ has dimensions ($4 \times$ *number of elements*) $\times$ ($2 \times$ *number of elements*) and the vector $\boldsymbol{\delta}$ has dimensions $2 \times$ *number of elements*. The rows of the vector $\boldsymbol{\delta}$ are essentially the diagonal values of $\boldsymbol{D^{-1}}$. Because the interval numbers occur only once in this vector, we avoid interval operation repetition with the same number. The matrix sizes are twice as large as the EBE for steady-state formulation because of the mesh duplication and this is the cost to reduce the number of interval operations.

## 5 - Details of the Implementation

Because no library was readily available to us to solve interval linear systems of equations, we have implemented one using Object-Oriented (OO) technologies in C++. By making use o templates and traits techniques we have been able to develop a library that can solve a linear systems of equations for distinct types of numbers such as real, interval and fuzzy using the same implementation. This library also can handle different matrix storage structures such as dense, banded, and tridiagonal. In this library, we have implemented the following methods: (1) preconditioned Gauss-Seidel (Kulpa, 1998); (2) Preconditioned Gaussian elimination (Kulpa, 1998); (3) Optimization using Powell's method (Rao, 1998) ; and (4) the combinatorial method (Muhanna,2000). Figure 2, a graphical

representation of a fuzzy linear system of equations with triangle numbers, shows that a fuzzy system can be solved by $\alpha$ -cuts planes of interval linear system of equations.

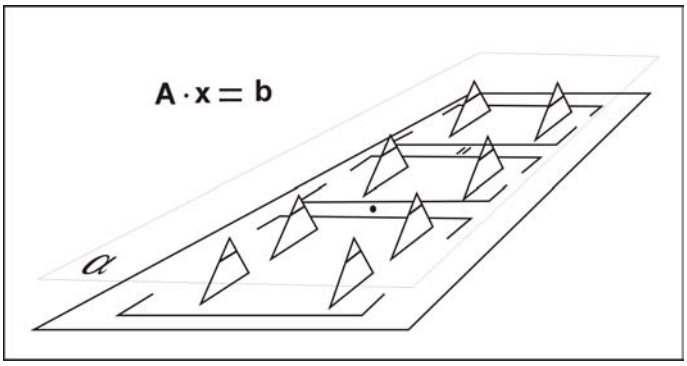



Figure 2 – Fuzzy linear system of equations.

In Figure 3, we display the class derivation scheme for numbers and matrices, as well as the main algorithm classes for solution of linear systems. Following the OO approach, each class number is responsible by its operations (addition, subtraction, multiplication, division, absolute value, etc). This data encapsulation approach is also applied to each matrix class which is responsible by its own matrix operations (LU decomposition, multiply by vector, inverse, determinant, rank etc). We used the classes implemented by Deodato (1995) for interval and fuzzy number operations. In his implementation, the fuzzy number is subdivided in $\alpha$ -cuts confidence intervals. We made extensive use of operator overload, inlining, and templates in our implementation.

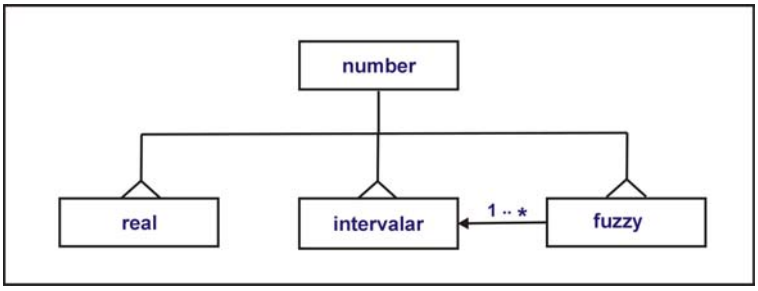



Figura 3 – Derivation class for number types.

Using templates and the generic programming approach (Mello & Khabibrakhmanov, 2003), there is a single implementation for the solution of the system ( $\boldsymbol{Ax = y}$ ), independently of the number type and matrix structure. The matrix and number types are template parameters implemented by the template specialization techniques.

This is a definitive advantage of generic programming, the algorithm does not need to know the details of the data structures used as long a common interface is provided for each number or matrix type. Then, the numbers and matrices operations are responsibility of the numbers and the matrix classes respectively. Consequently, the same algorithm works regardless the type of the matrix or number provided.

In order to apply the concepts discussed so far in this paper, we modified a one-dimensional basin modeling software denominated GEOFEM - *Geological applications Of the Finite Element Method* – (Mello, 1994) which was originally written in C language, to C++ (GEOFEM++) to perform interval and fuzzy operations. A specific EBE assembly operation was added to this software to obtain the global stiffness matrix. We have adapted GEOFEM++ to perform Monte Carlo simulations with uniform, triangular, normal, and exponential density distributions.

## 6 - Application and Discussion

In this section, we apply the techniques discussed in this paper and we compare its performance with more traditional methods to assess uncertainty. For this discussion, we have used a synthetic well, in which we consider the thermal conductivity of shale and sandstone uncertain as described in Table 3. The lower and upper bounds of the interval value are obtained by the medium subtracting and adding the standard deviation respectively.

| Lithology | Minimum | Maximum | Medium | Standard Deviation | Interval Value |
|---|---|---|---|---|---|
| Shale | 1.3 | 1.7 | 1.50 | 0.05 | [1.45, 1.55] |
| Sandstone | 2.90 | 3.10 | 3.00 | 0.10 | [2.90, 3.10] |

Table 3 – Thermal conductivity values for shale and sandstone (W/mK)

For the sake of simplicity, we discuss initially a very simple example to evaluate the algorithm efficiency in relation to the size of the mesh and the width of the interval numbers. In Figure 4, we display a one-dimensional mesh representing the present-day column of sediments with 4 elements (layer) and 5 nodes (horizons).

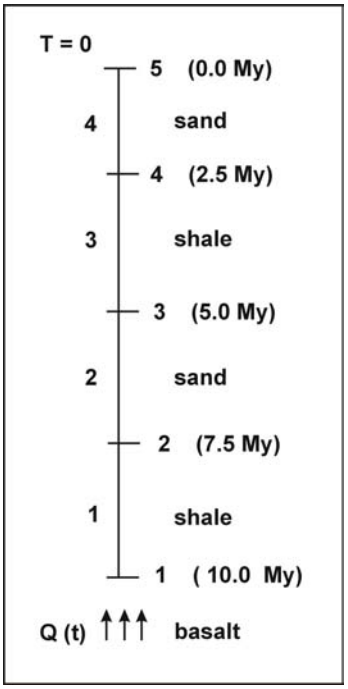



Figure 4 – The elements in the mesh represent geological layers and the nodes the horizons. The respective age, million of years (My), of each horizon is shown on the right of the node numbers.

We calculated the evolution of this mesh over the time, the results for a single node of the mesh at present-day time are shown in Table 4. The value (15.86 $^{o}$C) is the result of a traditional deterministic simulation using real numbers (Crisp solution) for the thermal conductivity (center value in Table 3). In the sequence, the Monte Carlo (MC) method was applied with 1000 experiments using a uniform distribution, and the results provide a range of uncertainty to the temperature between 14.95 to 16.85 $^{o}$C. For the uniform distribution, the average value is the same as the crisp solution. Note that an uncertainty of 13 to 17% in the thermal conductivity of the sediments induced a lower than 7% uncertainty in the temperature. The combinatorial methods, which provides an accurate hull for the solution, resulted in a range, [15.43 16.50], close to the MC method but with a little tighter interval. The Preconditioned Gaussian elimination and Gauss-Seidel results are clearly overestimated, whereas the Powell method generated a much tighter solution. Clearly, a poor selection of the interval solution method can provide inaccurate solutions. The EBE method provided a good solution close to the combinatorial method. This shows the advantages and limitations of the interval methods. In our opinion, the EBE method and MC method were able to assess the uncertainty accurately in this simple case.

| **Method** | **Temperature** |
|---|---|
| | |
| Crisp | 15.86 |
| Monte Carlo (MC) | [14.95, 15.86, 16.85] |
| Preconditioned Gauss Elimination | [12.38, 15.86, 21.80] |
| Preconditioned Gauss-Seidel | [10.77, 15.86, 27.35] |
| Combinatorial | [15.43, 15.86, 16.50] |
| Powell | [15.83, 15.86, 16.01] |
| EBE | [15.58, 15.86, 16.30] |

Table 4 – Comparison among different solution methods for node 3 for the mesh displayed in Figure 4.

In the next experiment, we doubled the number of elements in the mesh as shown in Figure 5, the results for the node 5 are presented in Table 5.

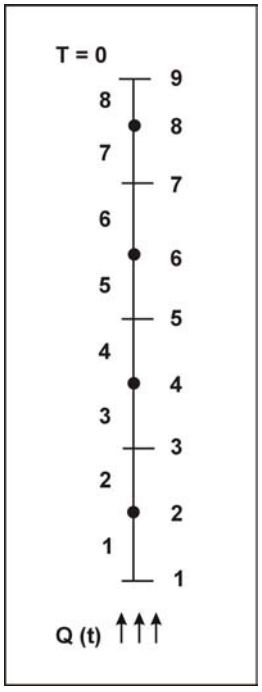


Figure 5 – Mesh with 8 elements.

| Method | Temperature ($^{o}$C) |
|---|---|
| | |
| Crisp | 15.66 |
| Monte Carlo (MC) | [14.83, 15.66, 16.73] |
| Preconditioned Gauss Elimination | *Failed* |
| Preconditioned Gauss-Seidel | [7.07, 15.66, 60.33] |
| Combinatorial | [15.22, 15.66, 16.34] |
| Powell | [15.66, 15.66, 15.73] |
| EBE | [15.14, 15.66, 16.22] |

Table 5 – Results for the node 5 of the mesh shown in Figure 5.

When the number of the elements increases, the number of operations to invert the interval global stiffness matrix also increases, leading to potential problems such as excessive overestimation. This can be verified in Table 5, especially for the Preconditioned Gauss-Seidel method. For this case, the preconditioned Gauss elimination failed. This failure occurred due the large number of operations with interval number during the pivoting phase of the Gaussian elimination. The large number of operation tends to increase the width of interval number. This width may include zero in the range, making the interval division underdetermined. Similarly to the previous case, the Powell method results were again too tight, and the MC and EBE methods provided good results.

To evaluate the effect o wider range of uncertainty, we multiplied the standard deviation of the thermal conductivity by two (Table 6). The results of the application of the selected methods are shown in Table 7 for this experiment. We used the same four-element mesh as displayed in Figure 4.

| Lithology | Minimum | Maximum | Medium | Standard Deviation | Interval Value |
|---|---|---|---|---|---|
| Shale | 1.30 | 1.70 | 1.50 | 0.10 | [1.40, 1.60] |
| Sandstone | 2.90 | 3.10 | 3.00 | 0.20 | [2.80, 3.20] |

Table 6 – Thermal conductivity for shale and sand (W/mK).

| Method | Temperature ($^{o}$C) |
|---|---|
| | |
| Crisp | 15.66 |
| Preconditioned Gauss Elimination | *Failed* |
| Preconditioned Gauss-Seidel | *Failed* |
| Combinatorial | [14.58, 15.66, 17.34] |
| Powell | *Failed* |
| Monte Carlo (MC) | [14.95, 15.90, 16.92] |
| EBE | [15.01, 15.66, 16.87] |

Table 7 – Results increasing the thermal conductivity width of the shale and sand as shown in Table 6.

The increase in the range of uncertainty caused further deterioration in the quality of some solution methods. In this case, the preconditioned Gauss Elimination and Gauss-Seidel failed to converge. The Powell method converged to an incorrect result. The EBE is the only interval method that provided results with good quality. It is interesting to note that the MC method resulted in slight tighter result when compared with the solution hull. This result could be even tighter if one included a density distribution with a shape different from the box distribution we selected.

As the last evaluation case, we applied the EBE, MC and Combinatorial methods to a well with real data. The stratigraphy is described in Table 8. In the first line, sandstone is the lithology between the horizons 1 and 2. Table 9 shows the lithology conductivity uncertainty used in this case.

| Horizon | Age (My) | Depth (m) | Lithology |
|---|---|---|---|
| 1 | 0.0 | 0.0 | Sandstone |
| 2 | 14.4 | 438.0 | Silt |
| 3 | 25.0 | 973.0 | Sandstone |
| 4 | 32.5 | 1698.0 | Shale |
| 5 | 42.0 | 3018.0 | Limestone |
| 6 | 64.70 | 3681.0 | Shale |
| 7 | 83.0 | 4050.0 | Sandstone |
| 8 | 98.8 | 4243.0 | Silt |
| 9 | 110.8 | 4719.0 | Shale |
| 10 | 114.0 | 4873.0 | Silt |
| 11 | 120.0 | 6448.0 | ------- |

Table 8 – Stratigraphy for the well used in our analysis.

| Lithology | Minimum | Maximum | Medium | Standard Deviation | Interval Value |
|---|---|---|---|---|---|
| Shale | 1.30 | 1.70 | 1.50 | 0.20 | [1.30, 1.50] |
| Sandstone | 2.50 | 3.50 | 3.00 | 0.50 | [2.50, 3.50] |
| Limestone | 2.00 | 2.50 | 3.00 | 0.50 | [2.00, 3.00] |
| Silt | 1.50 | 2.00 | 2.50 | 0.50 | [1.50, 2.50] |

Table 9 – Conductivities used in the 22 elements mesh (W/mK).

For this analysis we assumed no variation in the paleobathymetry and sea-level over time. We built a finite element mesh with 22 elements to represent the stratigraphy listed in Table 8, with two elements for each layer. For this modeling, we used the approach described in Mello et al. (1994a) and Mello & Karner (1996), which make use of a fully compacted coordinate system, to eliminate mesh deformation over time. Figure 6 summarizes the temperature calculation at present-day time.

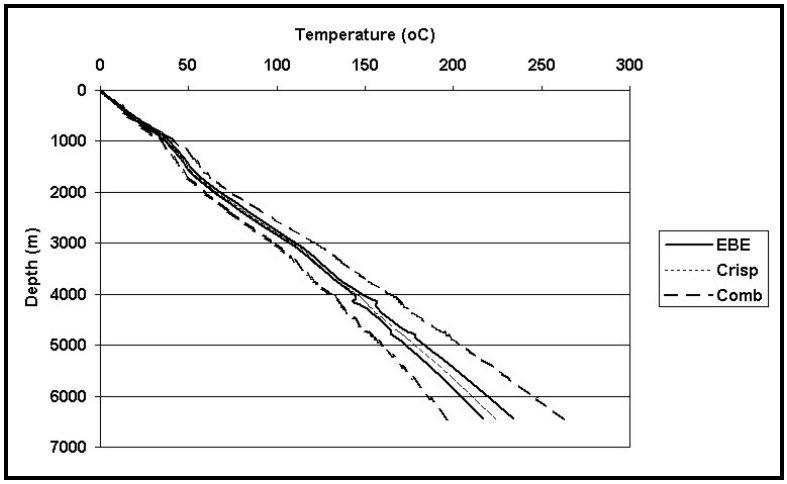



Figure 6 – Temperature versus depth for the data of Table 8.

In this figure, we display the results for Crisp, Combinatorial and EBE methods. The other methods failed for this case. The uncertainty increases with the depth due to the transient numerical solution. Note that, the EBE results were confined within the hull of the solution as defined by the combinatorial method. Although the MC result is not displayed in Figure 6, it was close to the EBE result (Table 10). This table shows the comparison of the MC method with the EBE and Combinatorial methods at the selected depth of 4146 m. For the MC simulations, it was used a uniform distribution of the thermal conductivity, 1000 events, 15 histogram classes, and a cut-off of 5%.

| Method | Temperature ($^{o}$C) |
|---|---|
| Crisp | 150 |
| Combinatorial | [134, 169] |
| Monte Carlo | [143, 156] |
| EBE | [143, 157] |

Tabela 10 – Temperature estimation at the depth 4146 m in Figure 6.

The EBE simulation took approximately 6 seconds and the MC took 35 minutes in a Intel P4 machine with 1,7GHZ and 1GB of RAM. The good quality of the EBE result was only achievable due to the EBE formulation which reduce the number of interval operations significantly.

## 7- Conclusions

In this work we evaluated the potential and limitations of an interval possibilistic approach to assess uncertainty in basin modeling. The interval arithmetic approach is an alternative to traditional probabilistic stochastic mehodology. We extended the interval finite element EBE formulation to transient heat transport Equation. This formulation proved to present good results within the hull of possible solutions with a quality similar to the Monte Carlo method. However, the EBE formulation has the advantage to perform the uncertainty analysis with a single simulation, requiring much less computational resources.

Here we compared the EBE formulation with more traditional solutions for interval systems of Equations and the EBE has proved to be the most robust. The Preconditioned Gaussian elimination and Gauss-Seidel methods did not performed well with large meshes by either being excessively overestimated or failing to converge. In addition, these methods also had problems to deal relatively wide numbers.

The optimization Powel method algorithm may produce incorrect or too tight results in basin modeling. The Combinatorial method, which provides the exact convex hull of the possible solutions, cannot be used in practical problems since it requires $2^n$ operation and becomes rapidly unviable when the number of interval variables grows.

The Monte Carlo method gives adequate results to uncertainty analysis when there is sufficient statistical information about the variables. The MC method has the advantage of allow analysis of multiple uncertain variables simultaneously without the need to change simulation applications. The major problem with MC method is the computational cost and the randomness assumption for geological processes that are in general not random.

The EBE with element overlap seems to be a viable alternative for one-dimensional basin modeling due to the quality of its results. However more studies are necessary to analyze its possible applications to multidimensional basin modeling. In higher dimensions, the size of the EBE global matrices may become too large to be solved efficiently.

## 8 – Acknowledgments

This paper has benefited greatly by the critically reviewing made by Andrew Conn.

## References


1. Burnett, D. S., 1987, *Finite Element Analysis from Concepts to Applications*. Addison-Wesley Publishing Company, Whippany NJ, USA, and ISBN: 0-201-10806-2.
2. Deodato, S., Anile, A. M., Privitera, G., 1995, "Implementing fuzzy arithmetic", *Fuzzy Sets and Systems*, v. 72, pp. 239-250.
3. Hansen E. R., 2000, "The Hull of Preconditioned Interval Linear Equations *Reliable Computing*", v.6, n.2, pp. 95-103.
4. Hanss, M., 1999, "On the Implementation of Fuzzy Arithmetical Operations for Engineering Problems", *Proceedings of the 8th International Conference of the North American Fuzzy Information Processing Society* - NAFIPS 99, New York, NY, pp. 462-466.
5. Kulpa Z., Pownuk A., Skalna I., 1998, "Analysis of Linear Mechanical Structures with Uncertainties by Means of Interval Methods", *Computer Assisted Mechanics and Engineering Sciences*, v. 5, pp. 443-477.
6. McKenzie D., 1978, "Some remarks on the development of sedimentary basins", *Earth Planet. Sci. Lett*., v. 40, pp. 25-32.
7. Mello, U. T. and Karner, G. D. -1996- Development of sediment overpressure and its effect on thermal maturation: Application to the Gulf of mexico basin. American Association of Petroleum Geologists Bulletin, v.80(9):1367-1396.
8. Mello, U. T. and Khabibrakhmanov, I., 2003, On the Reusability and Numeric Efficiency of C++ Packages in Scientific Computing. Proceeding of the ClusterWorld Conference and Expo, June 23-26, 2003 in San Jose, California.
9. Mello, U. T., 1994, *Thermal and Mechanical History of Sediments in Extensional Basins*. Ph.D. dissertation, Columbia University, USA.
10. Mello, U. T.; Karner, G. D. and Anderson, R. N. -1994a- A physical explanation for the positioning of the depth to the top of overpressure in shale-dominated sequences in the Gulf Coast basin, United States. Journal of Geophysical Research, v.99, p. 2775-2789.
11. Moore, R. E., 1962, *Interval Arithmetic and Automatic Error Analysis in Digital Computing*. PhD dissertation, Stanford University, USA.
12. Muhanna, R. L., Mullen, R. L., 1995, "Development of Interval Based Methods for Fuzziness in Continuum Mechanics", *Proc*., *ISUMA-NAFIPS'95*, September 17-20, pp. 145-150.
13. Muhanna, R. L., Mullen, R. L., 1999, "Formulation of Fuzzy Finite Element Methods for Mechanics Problems", *Computer-Aided Civil and Infrastructure Engineering* (*previously Microcomputers in Civil Engineering*), v. 14, pp. 107-117.
14. Muhanna, R. L., Mullen, R. L., 2000, "Sharp Enclosure for Material Uncertainty in Solid and Structural Mechanics-Interval Based Approach, Part I, *"Proceedings of the Eighth ASCE Joint Specialty Conference on Probabilistic Mechanics and Structural Reliability,* University of Notre Dame, Notre Dame, Indiana, pp. 24-26.
15. Muhanna, R. L., Mullen, R. L., 2001, " Treatment of Geometric Tolerances In Finite Element Analysis," *International Journal of Advanced Manufacturing Systems,* v. 4, n. 1, pp. 143-162.
16. Muhanna, R. L., Mullen, R. L., 2001, " Uncertainty in Mechanics Problems - Interval-Based Approach", *Journal of Engineering Mechanics, ASCE,* v. 127, n. 6, pp. 557-566.
17. Muhanna, R., Ballarini, R., R. Mullen, 1998, "Calculation of Stress Intensity Factors with Fuzzy Loading, "*Proceedings of the 4-th International Conference on Stochastic Structural Dynamics*, University of Notre Dame, Notre Dame, IN, USA, pp. 6-8.
18. Mullen, R. L., Muhanna, R. L. 1999, "Bounds of Structural Response for All Possible Loadings ", *Journal of Structural Engineering, ASCE*, v. 125, n. 1, pp 98-106.

19. Mullen, R. L., Muhanna, R. L., 1996, "Structural Analysis with Fuzzy-Based Load Uncertainty", *Proc, 7th ASCE EMD/STD Joint Specialty Conference on Probabilistic Mechanics and Structural Reliability, WPI, MA*, pp. 310-313.
20. Mullen, R. L., Muhanna, R. L., 1998, "Interval Based Finite Element Methods ", *Proceedings of the 4-th International Conference, NMA'98, Recent Advances in Numerical Methods and Applications II*, Aug. 19-23, editors, Iliev, O. P., Kaschiev, M. S., etc., World Scientific, pp. 362-371.
21. Mullen, R. L., Muhanna, R., L., 1999, "Interval-Based Geometric and Material Uncertainty for Mechanics Problems," *Proceedings of the 13th ASCE Engineering Mechanics Division Conference.* The Johns Hopkins University, Baltimore, MD, pp. 13-16.
22. Mullen, R. L., Muhanna, R., L., 2000, "Sharp Interval Estimates for Finite Element Solutions with Fuzzy Material Properties, "*Proceedings of the 14th ASCE Engineering Mechanics Conference*, The University of Texas at Austin, pp. 21-24.
23. Nakagiri, S., Suzuki, K., 1999, "Finite Element Interval Analysis of External Loads Identified by Displacement Input Uncertainty", *Comput. Methods Appl. Mech. Engrg*. 168, pp. 63-72.
24. Neumaier, A., 1990, *Interval methods for systems of Equations*, Cambridge University Press, New York.
25. Neumaier. A., 1987,"Overestimation in Linear Interval Equations", *SIAM J. Numerical Analysis*, v. 24, n. 1, pp. 207-214.
26. Rao, S. S., Berke, L., 1997, "Analysis of Uncertain Structural Systems Using Interval Analysis", AIAA *Journal,* v. 35, n. 4, pp. 727-735.
27. Rao, S. S., Li Chen, 1998, "Numerical Solution of Fuzzy Linear Equations In Engineering Analysis, " *Int. J. Numer. Meth. Eng*. v. 43, pp. 391-408.
28. Rao, S. S., Sawyer, P., 1995, "Fuzzy Finite Element Approach for Analysis of Imprecisely Defined Systems", *AIAA Journal,* v. 33, n. 12, pp. 2364-2370.
29. Rump S. M., 1990, "Rigorous sensitive analysis for systems of linear and nonlinear Equations", Math. Of Computations, v. 54, n. 90, pp. 721-736.

## APPENDIX - INTERVAL ARITHMETIC OPERATIONS

An interval number $\boldsymbol{x}$ is represented by:

$$\boldsymbol{x} = \left[\underline{x}, \overline{x}\right]$$

where:

$\underline{\boldsymbol{x}}$ = lower bound

$\overline{\boldsymbol{x}}$ = upper bound

$\widetilde{\boldsymbol{x}} =$ is a real number that belongs to the interval number $\boldsymbol{x} = \left[\underline{x}, \overline{x}\right]$

$\breve{\boldsymbol{x}} =$ is the real number midpoint of $\boldsymbol{x}$

$\boldsymbol{wid}\ \boldsymbol{x} = \overline{x} - \underline{x}$ , is the width of interval number $\boldsymbol{x}$

**- Operations:**

$$\boldsymbol{x} + \boldsymbol{y} = \left[\underline{x} + \underline{y}, \overline{x} + \overline{y}\right] = \left[\underline{x} + \underline{y}, \overline{x} + \overline{y}\right]$$

$$\boldsymbol{x} - \boldsymbol{y} = \left[\underline{x} - \overline{y}, \overline{x} - \underline{y}\right]$$

$$\boldsymbol{x} \times \boldsymbol{y} = \left[\boldsymbol{min}\left\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\right\}, \boldsymbol{max}\left\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\right\}\right]$$

$$\frac{1}{x} = \left[\frac{1}{\overline{x}}, \frac{1}{\underline{x}}\right] \quad , \quad if \quad \underline{x} \times \overline{x} > 0$$

**- Properties:**

$x(y \pm z) \subseteq xy \pm xz, \quad para\ x, y, z \in IR$

$x - y \subseteq (x + z) - (y + z)$

$x / y \subseteq (xz) / yz$

$x - x \neq 0, \quad 0 \in (x - x)$

$x / x \neq 1, \quad 1 \in x / x$

Note that when there is a repetition of a variable in a mathematical expression we get an overestimated result.

# Using data bases to test methods for decisions under uncertainty

Raphael T. Haftka and Raluca I. Rosca
*Mechanical and Aerospace Engineering Department, University of Florida, Gainesville, FL 32611-6250*

Efstratios Nikolaidis
*Mechanical, Industrial and Manufacturing Engineering, University of Toledo, Toledo, OH 43606*

**Abstract.** To address the need for efficient and unbiased experimental testing of methods for decision under uncertainty, we devise an approach for probing weaknesses of these methods by running numerical experiments on readily available or easily obtainable databases of real life data. Since the approach uses real life data, it allows us to study the effect of modeling error on the performance of a method. For illustration, we apply probabilistic and possibilistic approaches to a database of results of a domino tower competition. The experiments yielded several surprising results. First, even though a probabilistic metric of success was used, there was no significant difference between the rates of success of the probabilistic and possibilistic models. Second, the common practice of inflating uncertainty when there is little data about the uncertain variables shifted the decision differently for the probabilistic and possibilistic models, with the latter being counter-intuitive. Finally, inflation of uncertainty proved detrimental even when very little data was available.

## 1. Introduction

Engineering design decisions commonly involve mathematical models, such as models for calculation of stresses in structural design, to help decision makers predict the outcomes of alternative courses of action. Errors in models are usually investigated experimentally, such as in aircraft certification tests, and occasionally such tests reveal weaknesses in the underlying models. An example of such weakness is sensitivity of failure loads to inevitable small imperfections in geometric shape.

Uncertainty affects the ability of a decision maker to make good decisions. Increasingly, uncertainty is taken into account in design decisions using models, such as probability distributions. Again, design decisions that are sensitive to errors in models of uncertainty may look good on paper but may be very poor in reality. However, there has been little work on using experiments to probe for such sensitivity or other weaknesses in methods or practices for building models of uncertainty.

Examples include the Dartboard Contest, conducted by *The Wall Street Journal* (*WSJ*), (see Greene and Smart, 1999), which compared active and passive investing. In the contest, experts (analysts or fund managers) competed with the *WSJ* staff, which selected stocks by throwing darts at a printout of the *WSJ* stock tables. *WSJ* reported that experts won 61 percent of 140 contests. Baer and Gensler (2002) re-analyzed the study, accounting for additional factors including dividends and risk (experts favored high-risk stocks). With these factors included, passive investing (throwing darts) turned out to be as good as active investing.

Walley (1991, pp. 632-638) conducted an experiment using data from the 1982 Soccer World Cup to compare Bayesian and imprecise (upper and lower) probabilities in making decisions about gambles on games. Of 17 participants, those who used upper and lower probabilities did better than one participant who used Bayesian probability. Participants whose probabilities of the three outcomes (win, lose, tie) of each game were uniform did better than those participants whose probabilities were far apart.

Winkler (1971) and de Finetti (1972) performed experiments to investigate how people assess precise probabilities. Like Walley, Winkler observed that more uniform probabilities tend to improve the degree of success. Walley's and Winkler's studies suggest that if one has low confidence in the probabilities of the outcomes of an uncertain event, one should select a probability distribution with large variance, and consequently large Shannon's entropy. This is consistent with the practice of using the maximum entropy principle (Kapur and Kevasan, 1992) to model uncertainty.

Unlike the game and investing examples, it is difficult to carry certification tests for engineering design decisions to probe models of uncertainty, because products are usually designed for low probabilities of failure, and many thousands of tests may be required to reveal weaknesses. Occasionally, disastrous failures reveal inadequacy of probability of failure estimates, as happened with the space shuttle. Instead of waiting for disasters, we can also use ingenuity to test methods for making decisions under uncertainty. This involves inventing decision problems for data already available in existing databases.

Gigerenzer and Todd (2000, pp. 97-118) pioneered this approach, pitting a complex decision-making method against a simpler, heuristic one. If the simpler method wins or draws, it reveals possible weakness in the more complex method. They used 20 existing available databases to compare methods for making binary decisions (e.g., find which of two professors has a higher salary, given cues such as each professor's rank and gender). They found that a heuristic method that takes into account only a single dominant cue bested the standard (and more complex) regression approach that takes all the cues into account.

We generalize Gigerenzer's testing procedure to compare methods for making decisions under uncertainty that require choice of optimum values of design (decision) variables. We have two objectives. First, we want to demonstrate that it is easy to take a database and invent scenarios calling for a decision (in short, decision scenarios) that lead to meaningful tests of the effectiveness of decision-making methods. Our testing procedure allows us to study the effect of modeling error because it uses real life data. Second, we wish to demonstrate with a simple example that such tests can raise concerns about aspects of methods that may not be readily apparent by examining the theoretical foundations of the methods.

As an example, we use a database (Table 1) of experiments in which one of us (Rosca), as well as a group of students engaged in a competition (Rosca, 2001), stacked domino blocks until they toppled (Fig. 1). We invent a decision scenario for a decision maker to guarantee a height for a domino tower that she will build so as to best a competitor by selecting a guaranteed height that is both attainable and competitive (it is unlikely that Competitor's tower will be taller by a given margin). This scenario is similar to a class of decision problems where a decision maker guarantees a performance level, and wins if she delivers it and the competitor fails to do so. This example allows us to compare the use of probability and possibility for making decisions.

Section 2 presents the approach for probing such methods for design under uncertainty using existing data. Sections 3-4 present the example with the domino towers, the results, and the lessons learned. Section 5 summarizes the conclusions of the study.

## 2. Testing approach

Figure 2 is an influence diagram of a decision with imperfect information. Elements of the decision are a *decision maker(s)*, *alternative courses of action* (in short*, actions*), *uncertain event(s)* and their *outcomes*, *consequences* of actions and *information* about the likelihood of the outcomes of the uncertain events. The consequence of an action depends on the outcomes of the uncertain events. The decision maker wants to select the action with the most desirable consequence. The decision maker has imperfect information about the likelihood (e.g., the probabilities) of the possible outcomes of the uncertain events. In this paper, we consider uncertain events whose out-

comes are characterized by variables (e.g., the uncertain collapse height of a tower of domino blocks is a variable).

Our approach for testing methods for decision making under uncertainty is to select a database with samples of some variables, construct a decision scenario in which these variables represent the uncertainties that the decision maker faces, make repeated decisions and evaluate the consequences of these decisions using the database. The four steps of the testing approach are explained in detail in the following (Fig.3). Note that the variables in the data base do not have to be random; rather, the decision maker is uncertain about the values that they assume.

Table 1. Domino competition database: maximum built height (in domino units)

| Height | Number of towers of given height | | Height | Number of towers of given height | | Height | Number of towers of given height | |
|---|---|---|---|---|---|---|---|---|
| | Rosca | Competitors | | Rosca | Competitors | | Rosca | Competitors |
| **20** | 1 | 0 | **32** | 3 | 7 | **44** | 0 | 1 |
| **21** | 1 | 0 | **33** | 4 | 7 | **45** | 2 | 2 |
| **22** | 0 | 1 | **34** | 4 | 4 | **46** | 2 | 4 |
| **23** | 2 | 0 | **35** | 1 | 7 | **47** | 0 | 3 |
| **24** | 0 | 0 | **36** | 3 | 3 | **48** | 0 | 0 |
| **25** | 2 | 1 | **37** | 5 | 9 | **49** | 0 | 0 |
| **26** | 1 | 0 | **38** | 1 | 2 | **50** | 0 | 0 |
| **27** | 1 | 9 | **39** | 2 | 3 | **51** | 0 | 0 |
| **28** | 3 | 2 | **40** | 2 | 1 | **52** | 0 | 0 |
| **29** | 3 | 4 | **41** | 1 | 3 | **53** | 0 | 0 |
| **30** | 3 | 6 | **42** | 0 | 2 | **54** | 0 | 0 |
| **31** | 3 | 5 | **43** | 0 | 3 | **55** | 0 | 1 |

Step A. Select a database. We can start with almost any database with samples of a reasonable size (e.g., greater than or equal to 30).

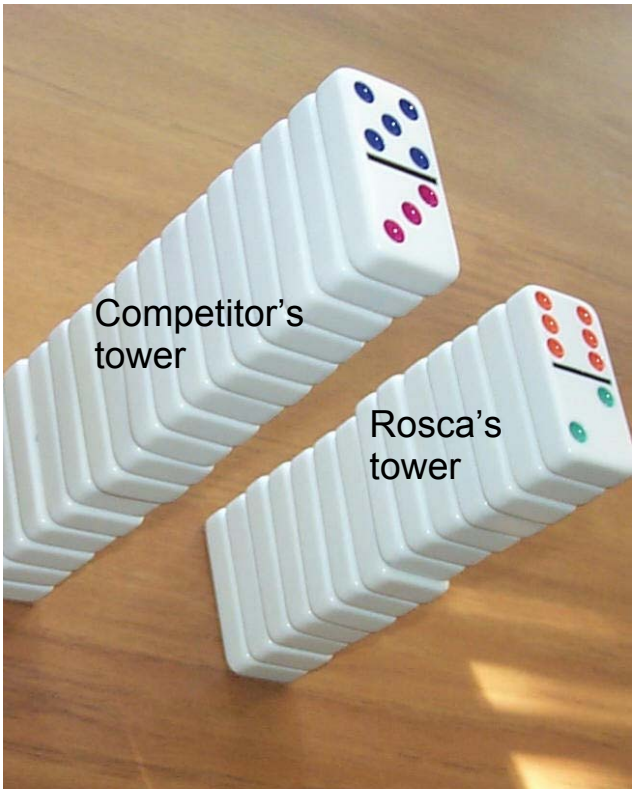



Figure 1: Domino towers in a competition

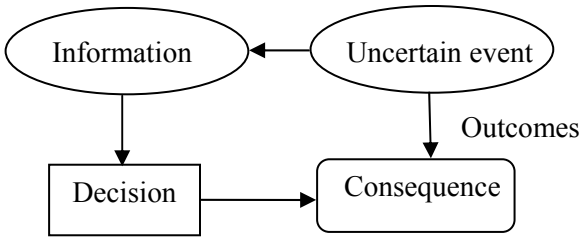



Figure 2. Decision under uncertainty. Arrows show relationships between the elements of a decision.

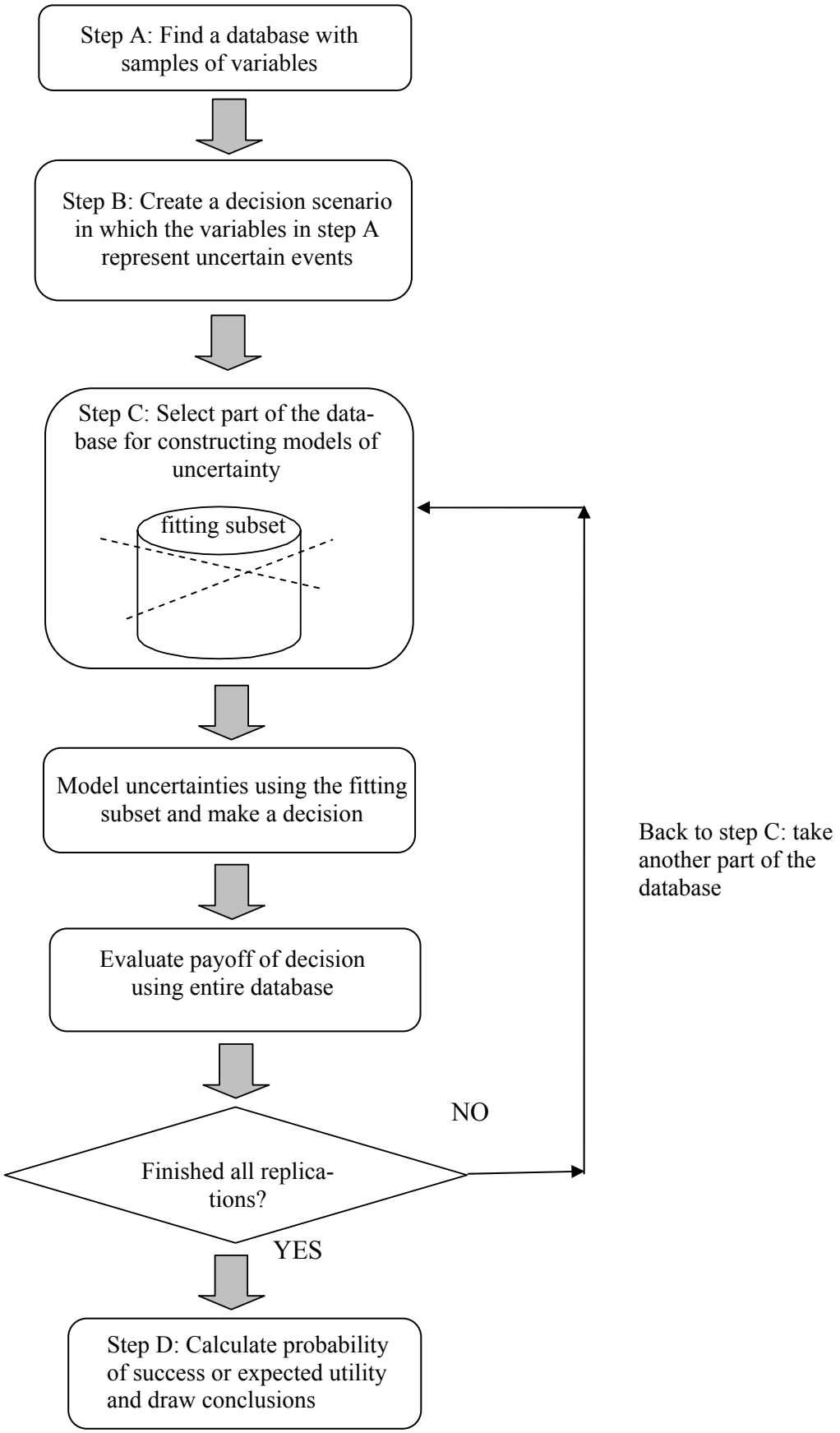



Figure 3: Approach for testing a method for decision under uncertainty

In the example of Table 1, the database contains heights of domino towers (just before they toppled) built by one of the authors (Rosca) in 50 trials and by 16 competitors in 90 trials (Fig. 1). The maximum height of a stable tower built by stacking domino blocks until the tower topples will be called *maximum built height* in this paper. This data gives us some statistical information (summarized in the histograms of Figure 4) on the height of the domino towers that Rosca and the Competitor can build.

Step B. Create a decision scenario given the variables in the database selected in step A. This is an unusual way to construct a decision scenario; instead of identifying the uncertainties in a given decision scenario, we invent a decision in which the variables in the database represent the uncertainties. A decision scenario is defined in terms of the following:

- the decision maker(s)
- the decision maker's objective
- the alternative courses of action (or choices)
- the possible consequences of an action
- the variables that affect the consequences of an action
- an algorithm for determining the consequence of an action given the values of the variables
- the information available for modeling the uncertainty associated with the variables.

For the domino-tower competition, the decision scenario we created is for Rosca (decision maker) to compete with a randomly chosen competitor (called Competitor) and guarantee a minimum height that she would build. Rosca loses if she did not meet her guarantee even if her tower was taller than that of Competitor's. To compensate for this disadvantage of Rosca, the rules of the competition stipulate that Competitor wins only if his tower height exceeds that of Rosca's guarantee plus a handicap. Both Rosca and Competitor are to build towers until they topple, and the maximum built height counts. If Rosca makes a high guarantee she risks not meeting it. If she makes a low guarantee she risks Competitor beating her guarantee plus the handicap. Figure 5 shows the decision tree, while Table 2 shows the elements of the decision scenario.

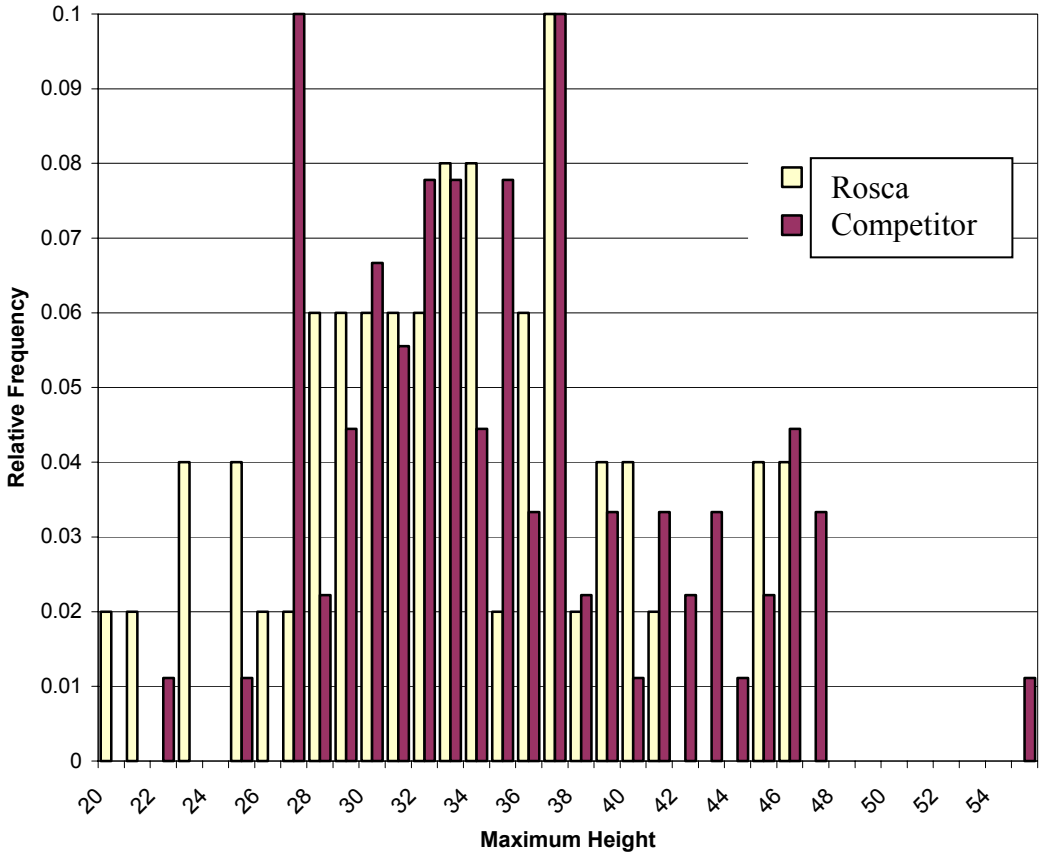



Figure 4. Histograms of maximum built heights of domino towers

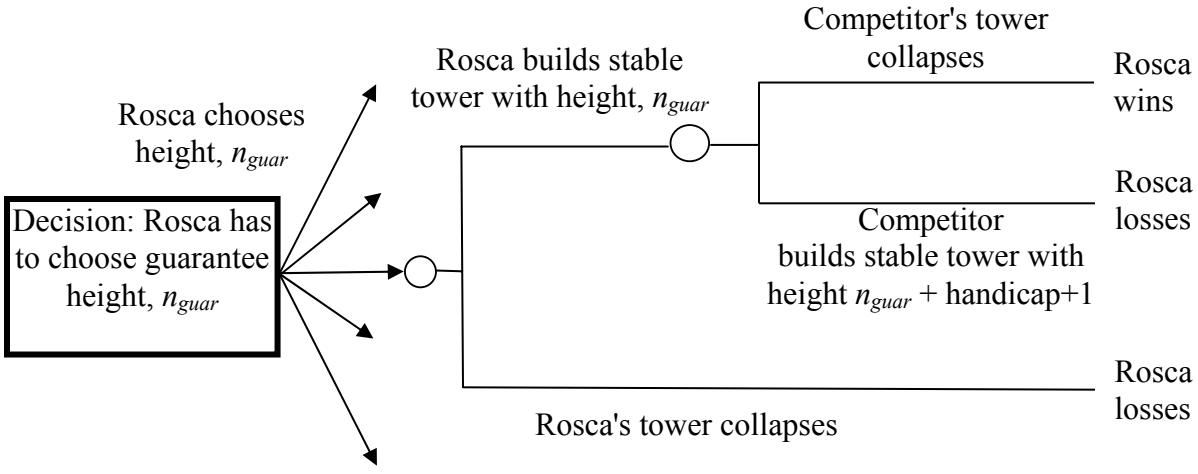



Figure 5: Domino competition: decision/event tree

Table 2. Elements of decision scenario in example

| Element | Description |
|---|---|
| Decision maker | Rosca |
| Objective | Win contest |
| Alternative courses of action | Guarantee different tower heights, $n_{guar}$ |
| Possible consequences of an action | Rosca loses or wins |
| Variables that affect the consequences of an action | Maximum built heights of Rosca's tower ($n_{del}$) and Competitor's tower ($n_{comp}$) |
| Algorithm for determining the consequence of an action given the values of the variables in the database | 3 Rosca's tower collapses below guarantee, ($n_{del} < n_{guar}$) (Rosca loses)<br>3 Rosca builds a stable tower with guaranteed height, ($n_{del} \geq n_{guar}$) and Competitor builds stable tower with height greater than the guaranteed height plus the handicap, $n_{comp} > n_{guar} + n_{hand}$ (Rosca loses)<br>3 Otherwise Rosca wins |
| Information for modeling uncertainty | Data on maximum built heights of Rosca's and Competitor's towers in database |

Step C. Make decisions using the method(s) we want to test using part of the database. We select a part of the database (called fitting dataset) to construct models of the uncertainties, which are used to make a decision. This adds statistical uncertainty (uncertainty in estimating the statistics of the population in the database from the fitting dataset) to the uncertainty due to variability. It is important to investigate the effect of statistical uncertainty since it is usually present in design decisions.

Using a part of the database to construct models of the variables allows us to test the method on many decisions; each obtained using a different fitting part. We reduce the element of chance in the choice of the fitting part by selecting it randomly, and repeating the process many times. Thus we obtain a large number of decisions whose payoffs can be evaluated. This concept of testing a model using multiple random fitting datasets is commonly used in validating response surface approximations, such as neural networks (e.g., Hush and Horne 1993).

For the domino-tower problem, we employ probabilistic and possibilistic methods to decide what height to guarantee. We provide Rosca with a small random sample (the size is five for the results presented in this paper) of her own past performance as well as a similar sample of Competitor's past performance. We could select $\binom{50}{5}$ different parts of the database with values of the collapse heights of Rosca's towers, where the notation $\binom{50}{5}$ indicates the number of all different 5-tuples taken from a population of 50 objects.

Step D. Evaluate the payoff of a decision by using the database as the entire universe of possible outcomes.

For a binary consequence (success or failure), we measure the payoff of a decision by its probability of success evaluated from the entire universe of all possible outcomes. If success is a

matter of degree, then we use the expected utility (Marston and Mistree, 1998, Hazelrigg, 1997, chapter 7) instead.

In the domino-tower problem we have 90×50=4,500 possible combinations of maximum built heights of Rosca and Competitor, and we can readily calculate the ratio of successful decisions out of the total.

**Student grade data base**

To demonstrate the generality of the testing approach, we present a database with very different characteristics from the domino database. As faculty members we regularly create student grade files, such as the one shown in the table below.

| student | quiz1 | quiz-2 | quiz-3 | Exam-1 | quiz-4 | … | Course average | Grade |
|---|---|---|---|---|---|---|---|---|
| 1 | 30 | 26 | 0 | 73 | 22 | … | 80.91 | B |
| 2 | 18 | 25 | 28 | 99 | 11 | … | 95.05 | A |
| : | : | : | : | : | : | : | : | : |
| 44 | 25 | 31 | 30 | 62.5 | 24 | … | 48.53 | F |
| 45 | 23 | 21 | 10 | 68 | 11 | … | 86.13 | B+ |

Using the student-grade database we can create the following decision scenario. A professor wants to identify students who are likely to get D or below (considered failure here) in order to call them for consultation. It is desirable to make the decision process simple and transparent. Therefore, the process is that if a student's course average is below a cutoff value, $a_c$, at the end of week $T$ of the semester, the student will be called. To aid the professor identify which students to call for consultation, a teaching assistant (decision maker) wants to develop a model predicting if a student, whose course average at the end of week $T$ is known, will fail. The construction of a predictive model can be viewed as a decision in which the teaching assistant decides on the consultation time and cutoff grade (decision variables). The teaching assistant's objectives are to maximize the model accuracy and minimize the waiting time to issue a warning.

The consequences of a choice of the consultation time and the cutoff grade include the number $P$ of false positives (students called for consultation who would have passed), and the number $N$ of false negatives (students who failed but were not called for consultation). It is desirable to minimize $P$ and $N$, which would call for waiting as long as possible. On the other hand, the longer the professor waits, the smaller is the chance that the student can improve much. We therefore define the following loss function to be minimized by the decision maker:

$$L = k_P P + k_N N + k_T T$$

The coefficient $k_P$ is the weight assigned to wasted time with students who do not need the consultation, $k_N$ is the weight assigned to missing students who need it, and $k_T$ is the weight assigned to the loss of time available to the student for corrective action.

In this problem the decision maker is uncertain if a given student will pass or fail. The grades in the database provide information that the teaching assistant can use to model this uncertainty. Figure 6 explains the decision of the teaching assistant. Using the information in the database the teaching assistant builds a model of the random variables. Then he/she chooses the waiting time and the cutoff grade so as to minimize the loss function of the model for predicting if a student will fail.

This problem may be particularly useful for testing Bayesian approaches against more traditional probabilistic approaches, because data from other courses and previous experience with the same course may be used to create prior probability distributions of failure as functions of the two decision variables. We can update the prior probability distributions using the grades in the database.

The testing procedure involves providing information on the entire semester history for a few students, using it to select the two decision variables, and then testing the consequences on the remaining students. That is, once the week $T$ and cutoff $a_c$ have been chosen, we can use the final grade information to obtain the number of false positive $P$ and number of false negative $N$, and calculate the loss function. The procedure can be repeated for many random subsets of the students.

The loss function implicitly assumes that a consultation with the student will increase the chances of the student to pass. While this is not obvious, we note that this issue does not lessen the value of this example for testing methods for making decisions, as the loss function involves diagnosis rather than corrective effects.

## 3. Comparing Possibilistic and Probabilistic Formulations for Domino Problem

A common option for modeling the uncertainty in the maximum built height in the domino decision problem of Table 2 is to fit probability distributions to data on past performance (data from Table 1 or Figure 4). Our previous investigation into the mechanics of the domino problem revealed that the probability distribution of stack heights for a single builder or for a group of builders can be approximated well by a shifted Gamma distribution but is approximated almost as well by a normal distribution (Rosca, 2001).

In order to demonstrate the utility of our testing approach we compare a probabilistic and a possibilistic method for making the decision. The latter is based on a simpler representation of the uncertainty via a triangular possibility distribution function and may be therefore less sensitive to the lack of available data. Possibility theory is presented in several books and papers including Dubois and Prade (1988), Joslyn (1994, 1995), and Nikolaidis et al. (2004).

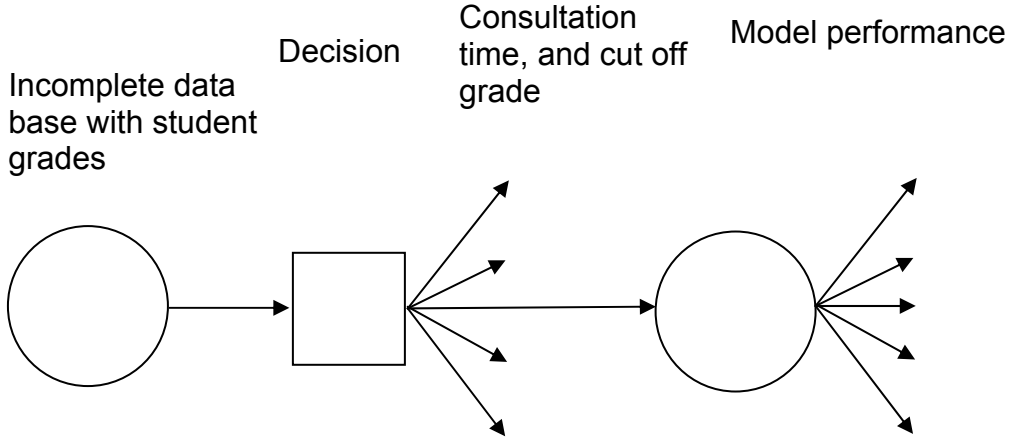



Figure 6. Decision about consultation time and cutoff grade when constructing a model for predicting a student's performance

**Problem formulation**

In possibility theory, the possibility of an event and the possibility of its complement do not necessarily add up to 1 (as is the case for probability theory). Therefore, we can maximize the possibility of success or minimize the possibility of failure without necessarily obtaining the same guarantee. We assume that the maximum built heights of Rosca's and Competitor's towers are independent. Then the possibility of Rosca winning is equal to the minimum of the possibility of Rosca building a stable tower (*Rosca delivers*) with guaranteed height $n$ and the possibility of Competitor failing to build a tower taller than the guaranteed height plus the handicap (*Competitor fails*):

$Pos\ (winning\ (n_{guar})) = Pos\ (\text{Rosca delivers and Competitor fails}) =$

$$min\ [Pos\ (n_{del} \geq n_{guar}),\ Pos\ (n_{comp} < n_{guar} + n_{hand} + 1)] \quad (1a)$$

Similarly, the possibility of Rosca losing is:

$$Pos\ (losing\ (n_{guar})) = max\ [Pos\ (n_{del} < n_{guar}),\ Pos\ (n_{com} \geq n_{guar} + n_{hand} + 1)]. \quad (1b)$$

Both formulations can provide multiple optima. For our data, the sets of optima given by these two possibilistic approaches were not disjoint. We call the intersection of these two sets the *possibilistic optimum*. There might be cases where the intersection contains more than one element.

Since Rosca built her towers without interaction with the other builders and at a different time, Rosca's and Competitor's maximum built heights are assumed statistically independent. Therefore, the probability of Rosca winning a contest when she guaranteed a stack of height $n=n_{guar}$ is equal to the product of probability that Rosca delivers and the Competitor fails:

$$Pro\ (winning\ (n_{guar})) = Pro(n_{del} \geq n_{guar}) \cdot Pro(n_{comp} < n_{guar} + n_{hand} + 1) = [1 - F_{Rosca}(n_{guar})]\ F_{Comp}(n_{guar} + n_{hand} + 1) \quad (2)$$

where $F_{Rosca}(n)$ and $F_{Comp}(n)$ denote the cumulative distribution functions of the maximum built heights of Rosca's and Competitor's towers, respectively. The probability that Rosca delivers decreases with $n_{guar}$, whereas the probability of Competitor's failure increases with $n_{guar}$. In the probabilistic formulation, we want to find the guaranteed stack height, $n_{guar}$, that maximizes the probability of winning.

We will compare the optima obtained by the two formulations when using data from the domino experiments to model uncertainty. We analyze two cases: (1) all data are used to find the optimum, and (2) only a sample of the data is used. When little data is available a designer can use a standard probability distribution that it is known to describe the uncertain variable (in our case the maximum built height) or employ the maximum entropy principle if such a standard distribution is not known. In this study, we consider that both the probabilistic and possibilistic designers know that the Gamma and the normal distributions fit well the maximum built height of a tower. In case (1), there is only uncertainty due to the error in approximating the actual discrete distribution as Gamma or normal probability distributions (called here fitting error). The true values of the parameters of these distributions are computed from the entire database of the maximum built heights, which in this study is considered the universe of the values of the maximum built heights. In case (2), there is additional uncertainty in the values of the parameters of the distributions besides the uncertainty due to fitting error.

In the decision scenario considered in this study, the probabilistic approach has two advantages compared to the possibilistic approach: a) the probabilistic approach seeks to maximize the

right objective (the probability of Rosca winning), and b) even when a small sample of data is available, Rosca knows that the Gamma and the normal probability distributions fit well the data for the maximum built heights. The information on the type of probability distribution is one that the possibilistic designer cannot directly utilize, so the probabilistic designer has an advantage.

**Definition and evaluation of the likelihood of winning**

Generally, for a given sample, the possibilistic and probabilistic formulations yield different optima, because they maximize different objective functions. We compare the two optima in terms of their relative frequency of winning (also called likelihood of winning), considering all possible Rosca-Competitor competitions obtained by combining all the data for the collapse heights of the towers built by Rosca and Competitor. With 50 experiments available for Rosca and 90 experiments available for Competitor, the likelihood is calculated by counting the number of pairs for which Rosca won as a fraction of the universe of possible pairs of Rosca and Competitor data, that is, 4,500 pairs. Consider a competition in which the maximum built height of Rosca is $n_{del}=N_1$ blocks and the maximum built height of the competitor's tower is $n_{comp}=N_2$ blocks. Rosca won if

$$N_1 \geq n_{guar} \text{ and } N_2 < n_{guar} + n_{hand} + 1.$$

The likelihood of winning of $n_{guar}$ is the total number of pairs ($N_1$, $N_2$) for which Rosca won, divided by 4,500. This likelihood of winning may be viewed as an approximation to the actual probability based on the limited database. We prefer to view it as an exact calculation for a problem with a limited discrete universe.

Using the likelihood of winning as a metric of the quality of a decision and with all of the data and no fitting errors, the probabilistic formulation should be superior. The possibilistic approach can prevail only if the fitting errors and the errors due to incomplete data overcome the natural advantage of the probabilistic approach.

**Splitting the data into fitting and testing sets**

If we use all the data for selecting the optimal $n_{guar}$, we have a single example from which it is difficult to draw conclusions. However, the relatively large amount of data allows us to use subsets for making the decision and evaluating the payoff, and then to repeat the process for different subsets. This reduces the element of chance in the results. Here, we perform the comparison for 80 randomly chosen subsets.

We draw samples of size $n_{sample}$ from both the data sets of Rosca and Competitor. Based on these samples, we fit a shifted Gamma or a normal probability density and a possibility distribution. The fitting processes for the probability and possibility distributions of the collapse heights are described in Appendix 1. Based on the fitted functions and using a probabilistic or a possibilistic formulation, we solve the guaranteed height problem, obtaining one (or more) optimum guarantees.

Step D of our testing approach calls for evaluating the payoffs of the decisions. We compare the guarantees selected by each method in terms of their likelihood of Rosca winning on all possible combinations of the available data.

## 4. Results

We studied the likelihood of winning of the two methods first when all measurements in the database are known, and then when only five measurements are given. We also investigated the effectiveness of the common practice of inflating the variance of a variable to account for statistical uncertainty (uncertainty in estimating the statistics of a population from those of a sample). This

second study revealed an unexpected difference between probability- and possibility-based designs.

**All data known**

In this case, we do not have multiple samples, and we can make a single decision. However, we vary the handicap through the set of values {2, 5, 8, 11, 15}. Figure 7 shows the likelihood of Rosca winning for the probabilistic (with Gamma distribution) and possibilistic designs versus the handicap. As expected, the likelihood of winning increases with the handicap. One cannot tell which method does better from Fig. 7, as the probabilistic design wins for a handicap of 2, 5 and 15, while the possibilistic design wins for a handicap of 8 or 11. Figure 7 also shows the maximum achievable likelihood of winning in the ideal case where the probability distributions of the populations of the maximum built heights of the two players are known. These are the true probability distributions of the maximum built heights and they are equal to those in the histograms in Fig. 4. The difference between the maximum achievable probability of winning and the likelihood of winning of the probabilistic approach is due to the fitting error of the Gamma distribution to the data. It is observed that the effect of the fitting error is small.

Table 3 shows the optimum guarantee selected by the two probabilistic models and the possibilistic approach. The optimum guarantee decreases with the handicap increasing. This is because the increased handicap makes it harder for Competitor to build a tall enough stable tower and a low guarantee will reduce the risk of Rosca's failure to deliver the guarantee. The average likelihoods of winning of the three methods over the five values of the handicap are: 0.5289 for the probabilistic design method using the Gamma distribution, 0.5258 for the probabilistic design method using the normal distribution and 0.5234 for the possibilistic design approach, which are very close. These results may indicate that when all the data is available to the decision maker, the errors incurred by fitting the data to a probability distribution offset the advantage of the probabilistic approach over the possibilistic one (that it maximizes the same objective as the one used to score the results).

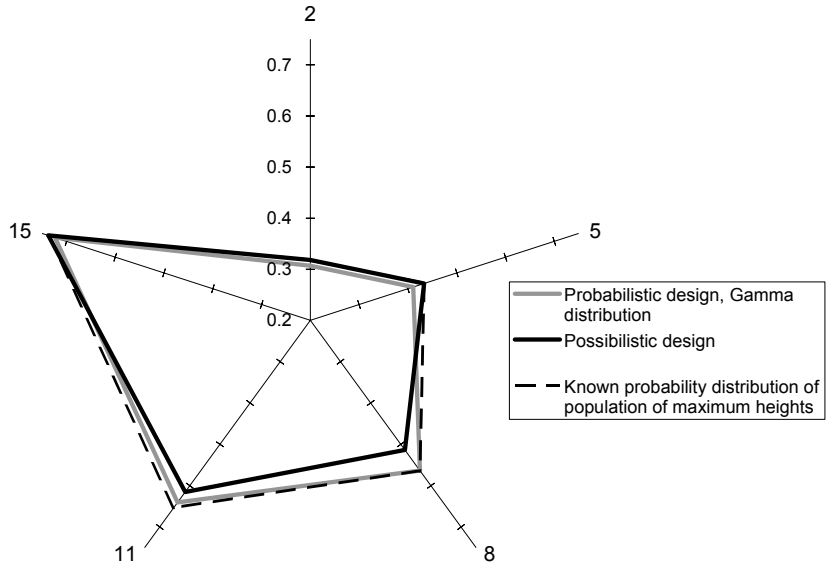



Figure 7. Comparison of the likelihood of success of probabilistic and possibilistic designs versus the handicap for the case where the decision maker has all the data in the database and the case where the decision maker knows the true probability distribution of the population of maximum heights

Table 3: Variation of optimum guarantee and its likelihood of winning with handicap values when all data are known; cases where the optimum guarantee was found are marked in bold. The optimum guarantee for a handicap of 11 is 28, corresponding to a likelihood of winning of 0.6533.

| Handicap | Probabilistic optimum (shifted Gamma fit) | | Probabilistic optimum (normal fit) | | Possibilistic optimum (triangular fit) | |
|---|---|---|---|---|---|---|
| | Optimum | Likelihood of winning | Optimum | Likelihood of winning | Optimum | Likelihood of winning |
| 2 | 32 | 0.3067 | **33** | 0.3180 | **33** | 0.3180 |
| 5 | 31 | 0.4107 | **32** | 0.4333 | **32** | 0.4333 |
| 8 | **29** | 0.5633 | 30 | 0.5360 | 31 | 0.5133 |
| 11 | 27 | 0.6402 | 29 | 0.6153 | 29 | 0.6153 |
| 15 | 26 | 0.7236 | 27 | 0.7262 | **28** | 0.7373 |

When only few experimental data are available to fit a probability distribution, a standard practice (Fox and Safie, 1992) is to inflate the variance of a distribution, keeping the mean value the same. Considering the parameters of a probability distribution to be random variables in order to account for statistical uncertainty also increases the variance of the distribution. We inflate the variance by adding to it an inflation factor multiplied by the standard deviation of the variance (see Appendix 2). When all the data is known, the effect of inflation is small because the standard deviation of the variance is small (see for example Table A1). Therefore, in order to understand the effect of inflation, we consider also the extreme case of an inflation factor of 15.

For a possibility distribution, there is no standard way to inflate the uncertainty. We use the simple approach of keeping fixed the mode of the distribution, which is equal to the sample mean, and inflating the support by the inflation factor. That is, if the mean is 32, and the support of the possibility distribution function is the interval (30, 35), then an inflation factor of 1 will inflate the interval to (28, 38), and an inflation factor of 2 to (26, 41). Here we use an inflation factor of 2, which corresponds to extreme inflation, similar in magnitude to an inflation factor of 15 for the probabilistic data.

From Table 4 we observe that when the uncertainty in Competitor's performance increases (inflation of 15), the probabilistic optimum guarantee decreases. On the other hand, when the uncertainty in Rosca's performance increases, the probabilistic optimum guarantee increases. The possibilistic optimum guarantee exhibits the opposite trend.

Table 4: Effect of inflating the uncertainty Rosca's and the Competitor's performance on the optimum guarantee and its likelihood of winning; handicap value, $n_{hand}$ *is 5*, all-data case. The true optimum height is 32. The probabilistic optimum decreases when the variability in Competitor's performance increases, and increases when the variability in Rosca's performance increases; the possibilistic optimum exhibits the opposite trend.

| Rosca | Competitor | Probabilistic optimum (shifted Gamma fit) | | Probabilistic optimum (normal fit) | | Possibilistic optimum (triangular fit) | |
|---|---|---|---|---|---|---|---|
| Inflation factor | | Optimum | Likelihood of winning | Optimum | Likelihood of winning | Optimum | Likelihood of winning |
| 0 | 0 | 31 | 0.4107 | 32 | 0.4333 | 32 | 0.4333 |
| 0 | 15 | 28 | 0.3920 | 29 | 0.3987 | 33 | 0.4020 |
| 15 | 0 | 33 | 0.4020 | 34 | 0.3578 | 31 | 0.4107 |
| 15 | 15 | 30 | 0.4240 | 32 | 0.4333 | 32 | 0.4333 |

When the variance of the Competitor's performance is inflated by a very large amount then the probabilistic optimum guaranteed height always decreases. The reason is that when the variance becomes very large, the probability of the Competitors' failure becomes insensitive to the guaranteed height. Therefore, for increasing the probability of winning given by (Eq. 2) *Pro (winning (n)) = [ 1-$F_{Rosca}(n)$] $F_{Comp}(n+n_{hand}+1)$* it is more important to increase the probability of Rosca's delivering the guarantee than to increase the probability of Competitor's failure.

The effect of inflating uncertainty on the optimum can be understood by examining the condition that the optimum must satisfy. At the optimum, the derivative of the logarithm of the probability of success is zero,

$$\frac{\partial \log(P(winning))}{\partial n} = \frac{-f_{Rosca}(n)}{(1-F_{Rosca}(n))} + \frac{f_{Comp}(n+n_{hand}+1)}{F_{Comp}(n+n_{hand}+1)} = 0 \qquad (3)$$

The two terms on the right hand side of the above equation are the sensitivities of the logarithms of the probabilities that Rosca delivers and Competitor fails. Extreme inflation of the uncertainty in the Competitor's performance makes the sensitivity of the derivative of the probability that the Competitor fails almost zero. The optimum guaranteed height decreases in order to maintain equality of the two terms in Eq. (3) (Fig. 8). Rosca (2001) provided a similar explanation as to why the optimum guarantee increases when the probability density function of the decision maker (Rosca) is inflated.

So in this probabilistic guarantee-setting problem, when the decision maker is highly uncertain about the capability of the competition, she should set conservative goals. On the other hand, when the decision maker is very uncertain in her own capability, she should set aggressive goals because it is more important to prevent the Competition from succeeding than to help the decision maker deliver the guaranteed performance. This fits the common sense notion that given two dangers, one should pay more attention to the danger which one can manage more easily.

In possibility, we can minimize the possibility of Rosca losing the contest or maximize the possibility of her winning. We minimize the possibility of losing because the possibility of winning is equal to one for heights between 30 and 33. The height for which the possibilities that Rosca delivers and the Competitor fails become equal ($n_{opt}$ in Fig. 9), minimizes the possibility of losing. Indeed, any deviation from $n_{opt}$ increases the possibility of Rosca losing. Smaller heights than $n_{opt}$ have higher possibility of Competitor success, while larger heights have higher possibility of Rosca failure. In both cases, the possibility of Rosca losing the contest (Eq. 1b) is higher than that for $n_{opt}$. In Table 4, the possibilistic optimum displays the opposite trend than the probabilistic optimum, increasing when we inflate Competitor's possibility distribution and decreasing when we inflate Rosca's possibility distribution. This can be explained by observing Fig. 9; inflating the Competitor's possibility distribution will increase the optimum (which is the intersection of the possibility distributions of the two players). Thus, in contrast to probabilistic design, inflation increases the importance of a failure mode in the possibilistic approach.

The philosophies of the probability and possibility can be further understood by examining a scenario that accentuates the difference of the optima of the two approaches. Consider the extreme case where the uncertainty in the Competitor's performance is very large and the uncertainty in Rosca's performance very small (that is, Rosca predicts quite accurately the maximum height of a tower that she can build but she does not know much about Competitor's performance). Figure 10 shows the probability densities and possibility distributions of the maximum built heights of the two players. The optimum guarantee that maximizes the probability of success is on the left tail of Rosca's probability density function where there is a small probability that she will not deliver. Then, the probability of winning is approximately equal to the probability of Competitor's failure for this height, which is 0.5. On the other hand, the possibilistic opti-

mum is very close to the mean value of the Rosca's distribution and has a probability of success 0.256, which is approximately equal to one half of the probability of success of the probabilistic optimum. It is also interesting that the possibilistic optimum is less robust to errors in the mean value of Rosca's maximum built height than the probabilistic optimum. For example, even a small reduction in the true mean value of Rosca's probability distribution will reduce greatly the probability of success of the possibilistic optimum. *Even though we have been comparing probability and possibility for the past few years, we needed this experimental result to discern the important differences between probability and possibility identified in this study*.

It is not difficult to check that the effect of inflation of the possibilistic optimum depends on the relative positions of the peaks for the two possibility distributions. That is, when the two are reversed, probability and possibility will behave in the same way. However, for probability distributions, the relative positions of the peaks do not affect the result that the inflated mode loses importance.

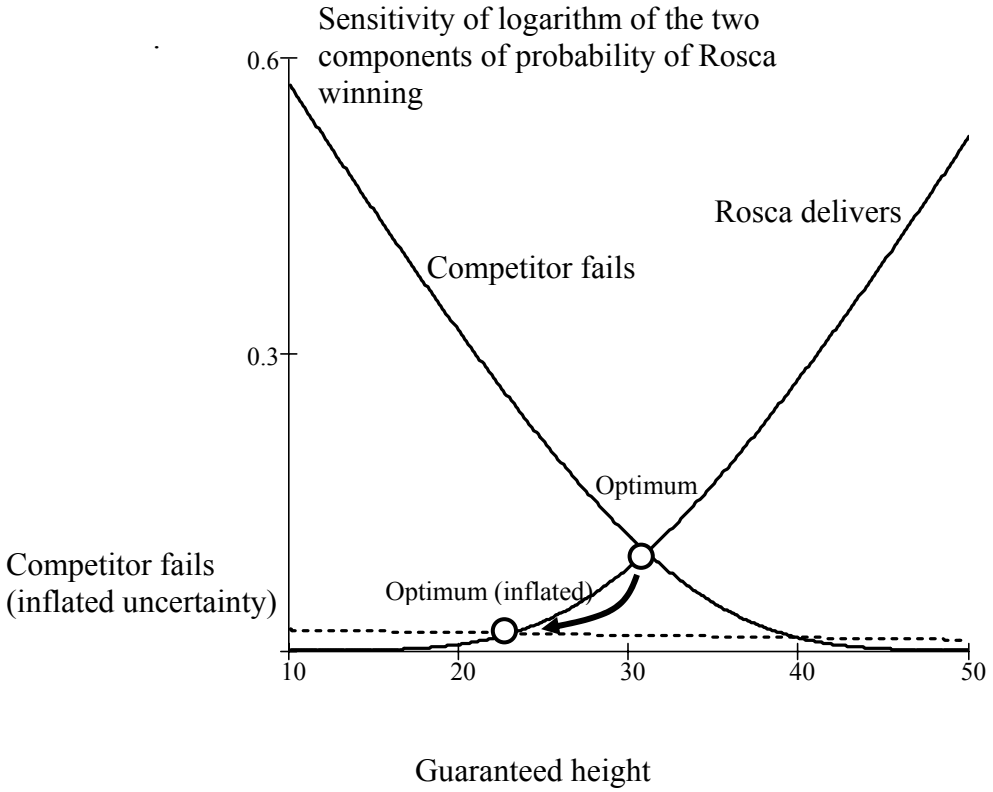



Figure 8. Extreme inflation of the uncertainty in Competitor's performance reduces the sensitivity of the probability of Competitor's failure to the guaranteed height, thereby reducing the optimum guaranteed height.

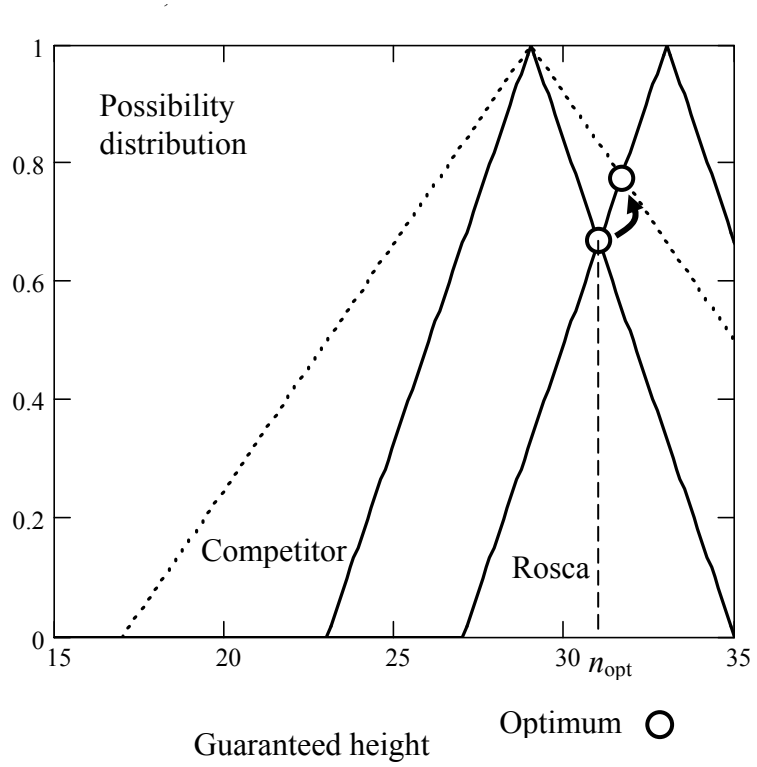



Figure 9. Inflation of the uncertainty in Competitor's performance increases importance of competitor's failure, thereby increasing the optimum guaranteed height.

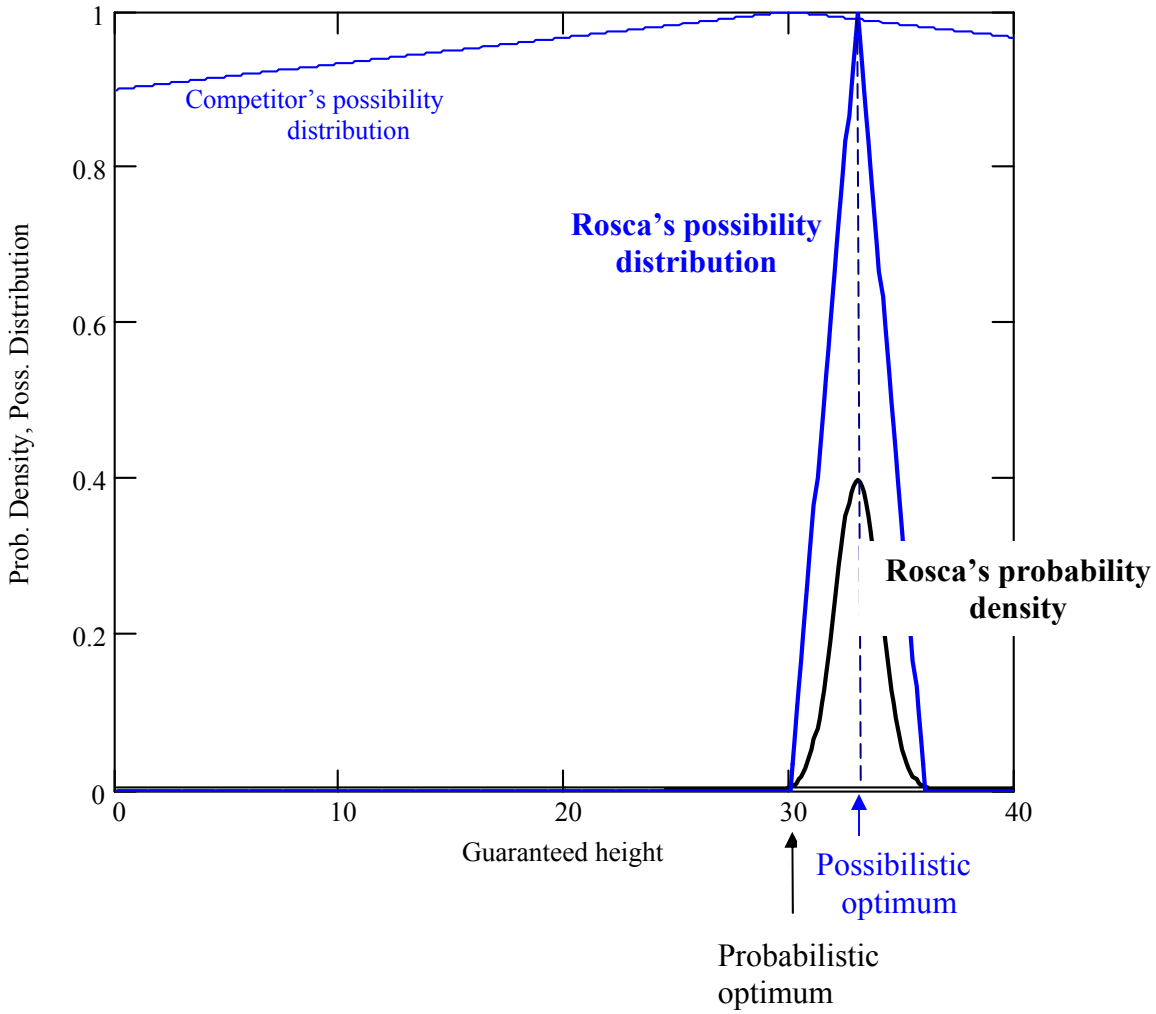



Figure 10. Probabilistic and possibilistic optima when there is high uncertainty in Competitor's performance and little uncertainty in Rosca's performance. The difference between the ways probabilistic and possibilistic design find the optimum guaranteed height is accentuated in this case. The probability density of Competitor's maximum built height is almost zero over the entire range of heights in the figure.

The contrast between the effect of inflation on probability and possibility is clearly due to the non-additivity of possibilities. Consider in Fig. 8 the optimum guarantee when the probability distributions of the players are not inflated. Reducing the optimum guarantee by 1 increases the chance that Rosca delivers more than reducing the chance that the Competitor fails. In probability, inflating the probability distribution of Competitor automatically reduces the probability of any one outcome. Thus, it allows us to reduce the guarantee by 1 with a smaller reduction of the chance of Competitor's failure. In the possibility model we can increase the possibility of one event without reducing the possibility of another. Thus inflating Competitor's possibility distribution increases the possibility of all the outcomes.

Finally, Table 4 shows that inflating only one of the uncertain variables reduces the likelihood of success of both methods. The reason is that increasing the uncertainty in the performance of a player introduces bias in the probabilistic model, which reduces the quality of a decision. Also, inflation affects probabilistic design more that the possibilistic design.

**Scarce data – small sample size**

For the scarce data case, we use only a randomly selected small subset of the data for fitting a distribution and selecting a guarantee. The process is repeated 80 times to average out the effect of chance in the selection of the sample. Rather than presenting all 80 examples of optima, we present their average (over the 80 samples) likelihood of success. We tested the models of uncertainty for five handicap values. Thus, we were able to test the models on $5 \times 80 = 400$ different decisions using the same pair of datasets for Rosca's and Competitor's collapse heights. Each decision was evaluated using 4,500 pairs of maximum built heights.

Figure 11 shows the average likelihoods of success of the probabilistic method that uses the Gamma distribution to model uncertainty in the maximum built heights of the towers and the possibilistic design for different handicaps. Since only five data points are available to the decision maker instead of 50 or 90, the likelihood of success of the optimum guarantee deteriorates compared to the all-data case. The likelihood of success of the probabilistic design is slightly higher than that of the possibilistic design but the difference is small. Table 5 presents the average and standard deviation of the likelihood of success when a sample of five values is used. When a Gamma probability distribution is fitted to the data, the reduction of the likelihood of success ranges from 2% to 6%, compared to the all-data case. The reduction in the likelihood of success ranges between 2% to 4% when a normal distribution is fitted to the data. Finally, when a possibilistic approach is used, the reduction in the likelihood of success ranges between 2% and 4%.

For both possibilistic and probabilistic methods, increasing the handicap value increases the mean of the likelihood of success. The average likelihoods of winning of the three methods over the 400 cases are 0.4947 for the probabilistic design method using the Gamma distribution, 0.4968 for the probabilistic design method using the normal distribution and 0.4909 for the possibilistic design method, which are very close, with a small advantage to the probabilistic models. This result surprised us, because, generally, we expected the possibilistic approach to do better relative to the probabilistic approach for the scarce data than for the full data case. But in the decision problem considered, probabilistic design has the advantage over the possibilistic design that the type of the probability distribution of the maximum built height is known even in the scarce data-case. Possibility does not permit the designer to account directly for this information even if she knows the type of the possibility distribution.

The poorer results of the possibilistic approach could also be due to the way we constructed a possibility distribution function based on the available data or the inability of the approach to properly account for the independence of the built heights of the towers of the two players. The possibility of the intersection of two events is equal to the minimum of the possibilities of these events (Eq. 1). This yields counterintuitive results when the events are known to be statistically independent. For example, the possibility of Rosca building a tower with height that has high

possibility (e.g., 30) and the competitor building a very tall tower (e.g. 55) is equal to the possibility of both players building very tall towers as long as Rosca's tower has higher possibility than Competitor's tower. This is clearly wrong because it is very unlikely that both players will build high towers simultaneously (Eq. 2). A hybrid probabilistic/possibilistic approach, that characterizes uncertainty in the maximum built height using probability distributions and uncertainty in the distribution parameters using a possibility distribution would avoid the above pitfalls and could do better than the approach that we considered in this study.

Table 5: Mean and standard deviation (computed over the 80 cases) of the likelihood of success for probabilistic optimum (shifted Gamma and normal fit) and possibilistic optimum (triangular fit); sample size of 5.

| sample size=5 | Likelihood of success for probabilistic optimum (shifted Gamma fit) | | Likelihood of success for probabilistic optimum (normal fit) | | Likelihood of success for possibilistic optimum (triangular fit) | |
|---|---|---|---|---|---|---|
| $n_{hand}$ | Mean (of 80 runs) | Standard deviation | Mean (of 80 runs) | Standard deviation | Mean (of 80 runs) | Standard deviation |
| 2 | 0.2850 | 0.0361 | 0.2822 | 0.0398 | 0.2896 | 0.0290 |
| 5 | 0.3924 | 0.0441 | 0.3924 | 0.0451 | 0.3917 | 0.0478 |
| 8 | 0.4995 | 0.0552 | 0.5031 | 0.0496 | 0.4921 | 0.0576 |
| 11 | 0.5967 | 0.0622 | 0.5993 | 0.0513 | 0.5875 | 0.0656 |
| 15 | 0.6997 | 0.0559 | 0.7069 | 0.0412 | 0.6937 | 0.0586 |

We repeated the fitting and optimization procedure for the case of the sample size of 5, but this time we inflated the standard deviation of the maximum built height and the support of the possibility distribution of this height. We present in Table 6 only the results for symmetric inflation.

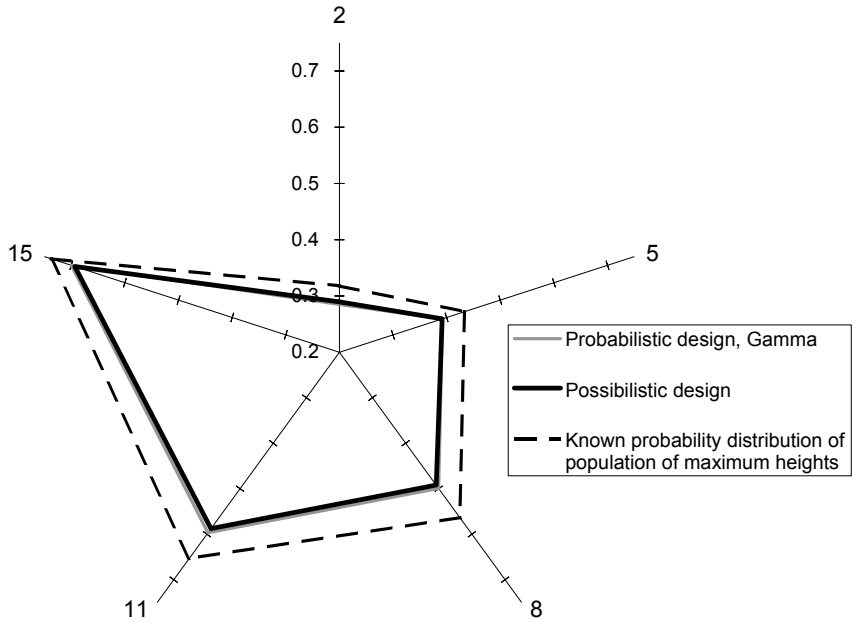



Figure 11. Comparison of the likelihood of success of probabilistic and possibilistic designs versus the handicap for the case where the decision maker has five data points and the case where the decision maker knows the true probability distribution of the population of maximum heights

Table 6: Mean and standard deviation (computed over the 80 cases) of the likelihood of success for probabilistic optimum (shifted Gamma and normal fit) and possibilistic optimum (triangular fit); sample size is 5. Both Rosca's and Competitor's inflation factors are 2.

| sample size=5 | Likelihood of success for probabilistic optimum (shifted Gamma fit) | | Likelihood of success for probabilistic optimum (normal fit) | | Likelihood of success for possibilistic optimum (triangular fit) | |
|---|---|---|---|---|---|---|
| $n_{hand}$ | Mean (of 80 runs) | Standard deviation | Mean (of 80 runs) | Standard deviation | Mean (of 80 runs) | Standard deviation |
| 2 | 0.2687 | 0.0520 | 0.2797 | 0.0466 | 0.2896 | 0.0290 |
| 5 | 0.3662 | 0.0591 | 0.3899 | 0.0496 | 0.3917 | 0.0478 |
| 8 | 0.4732 | 0.0758 | 0.4980 | 0.0554 | 0.4917 | 0.0575 |
| 11 | 0.5717 | 0.0849 | 0.6017 | 0.0481 | 0.5879 | 0.0645 |
| 15 | 0.6922 | 0.0637 | 0.7075 | 0.0360 | 0.6920 | 0.0606 |

Comparing Tables 5 and 6, we see that inflation had a detrimental effect on the probabilistic optimum. Indeed, for all but the handicap value of 2, the mean likelihood of success of the optimum given by the inflated shifted Gamma distribution is smaller than the corresponding non-inflated one. The same effect is observed for the normal distribution for all but the handicap of 11. For symmetric inflation, little or no effect is observed on the likelihood of success of the possibilistic optimum, because the possibilistic optimum does not change with symmetrical inflation. We also repeated the study with sample sizes of 3 and 10 and obtained similar results (Rosca, 2001).

The observation that inflation of uncertainty is counterproductive is at odds with Walley's observation in his World Cup experiment where those participants whose probabilities of the outcomes of a game were uniform made more money than those whose probabilities differed a lot. We think that Walley's experiment does not necessarily show that inflating uncertainty is an effective practice; it possibly shows that those participants who estimated uniform probabilities because they were aware of their ignorance did better than overconfident participants whose probabilities were asymmetric.

## 5. Conclusions

An approach for using existing data for probing weaknesses in models for making decisions under uncertainty has been developed. The approach may expose problems associated with errors in predictive models or in models of uncertainty because it uses real-life data. The approach requires two sets of data on one property (here, domino tower height) for two groups. It then creates a decision problem that involves finding an optimum in terms of one or more decision variables. The same dataset can be used to test methods on hundreds or thousands of different decisions within a short period at low cost. An example employing data on a domino tower competition was used for demonstration.

The utility of the experimental testing of methodologies for decisions under uncertainty was evidenced by several results that surprised us, even though we have been exploring the methods we evaluated for several years. These include the following:

1. Small fitting errors in the probability distributions were sufficient to give an advantage to possibilistic decision-making, even though the metric of success was probabilistic. This may indicate that these fitting errors deserve further study.
2. In contrast, the probabilistic approach suffered less than the possibilistic approach from small sample size. This may indicate that a better way of selecting possibility distribution

functions based on small samples may be needed, or that a hybrid probabilistic/possibilistic approach should be used in which possibility is only used only for those uncertainties for which it is difficult to estimate probabilistic models.

3. The process of magnification of the standard deviation of a probability distribution, which is commonly used when data are scarce, proved to be counterproductive.
4. The effect of magnifying uncertainty had an opposite effect on probability and possibility. Inflating uncertainty reduced the effect of a failure mode on the probabilistic decision, and it increased the effect of the mode on the possibilistic decision. This result was shown to be due to the fact that probability, unlike possibility, is additive. In the extreme case where uncertainty in one player's performance is much greater than in the other player's, the difference between the optimum decisions of the two methods is very large. In this case, possibility yielded a decision with much poorer performance than probability.

We note that there is a wealth of other data readily available for testing methods using the proposed approach, including records of student projects, insurance claims, stock market prices, and medical tests. As educators, we can readily see that universities have useful student databases. For example, records of the performance characteristics (e.g. the stroke and time ratio) of slider-crank mechanisms constructed in a class on design and analysis of mechanical systems can be used instead of domino heights.

## References


Baer, G., and Gesnsler, G., *The great mutual fund trap*, Broadway Books, Chapter 10, pp. 148-154, (2002).

de Finetti, B., *Probability, induction and statistics,* Wiley, London Chapters 1 and 2, (1972),.

Dubois, D., and Prade, H., *Possibility theory*, Plenum Press, New York (1988).

Fox, E. P., Safie F., Statistical characterization of life drivers for a probabilistic analy, *AIAA/SAE/ASME/ASEE, 28th Joint Propulsion Conference and Exhibit, Nashville, Tennessee*, AIAA-92-3414, (1992).

Freund, J.E., and Williams, F.J., *Dictionary/outline of basic statistics*, McGraw-Hill, New York, pp. 151, (1966).

Gigerenzer, G., and Todd, P. M., *Simple heuristics that make us smart*, Oxford University Press, New York, (1999).

Greene, J. and Smart, S., Liquidity provision and noise trading: Evidence from the 'investment dartboard' column. *Journal of Finance,* **54,** October, (1999).

R.T. Haftka, E.P. Scott and J.R. Cruz, Optimization and experiments: A survey. *Applied Mechanics Reviews,* **51,** 7, pp. 435-448, (1998).

G.A. Hazelrigg, *Systems engineering: An approach to information-based design*, Prentice Hall (1996).

Hush, D. R. and Horne, B. G., Progress in supervised neural networks. *IEEE Signal Processing Magazine,* pp. 8-38, (1993).

Joslyn, C., Possibilistic processes for complex systems modeling, Ph.D. dissertation, *Department of Systems Science, SUNY Binghamton,* Binghamton, New York, 1994.

Joslyn, C., In support of an independent possibility theory, in: de Cooman, Ruan, Kerre,, editors, *Foundations and applications of possibility theory*, World Scientific, Singapore, pp. 152-164, (1995).

Kapur, J.N., and Kevasan, H.K., *Entropy optimization principles with applications*, Academic Press, New York, (1992).

Marston M, Mistree F., An implementation of expected utility theory in decision based design. Proceedings of the *1998 DETC, 10th International Conference on Design Theory and Methodology, ASME* (1998).

Nikolaidis, E., Chen, Q., Cudney, H., Haftka, R. T., and Rosca, R., Comparison of Probability and Possibility for Design Against Catastrophic Failure Under Uncertainty, *Journal of Mechanical Design*, *ASME,* Vo. 126, Issue 3, pp. 386-394, (2004).

Rosca, R., Use of experimental data in testing methods for design against uncertainty. Ph.D. Dissertation, *Aerospace Engineering, Mechanics and Engineering Science Department*, University of Florida, Gainesville, (2001).

P. Walley *Statistical reasoning with imprecise probabilities*, Chapman and Hall, (1991).

R.L. Winkler, Probabilistic prediction: Some experimental results. *Journal of American Statistical Association,* pp. 675-685, (1966).

## Appendix 1: Description of the fitting process (fit of probability/possibility distribution functions)

In the possibilistic formulation, to each sample we fit an asymmetric triangular membership function, such that the mean of the sample corresponds to the peak of the membership function. The minimum and the maximum values in the sample are the minimum and the maximum values of the support of the triangular membership function. An example is shown in Fig. A1.

In the probabilistic formulation, we fit a probability density function (PDF, rather than CDF) to each sample. When all data are available, the best PDF fit is given by normal and shifted Gamma density functions. Therefore, even for small data samples (3, 5), we use the normal PDF and the shifted Gamma PDF to fit the data.

To find the shifted Gamma function, we choose the scale and shape parameters such that the mean and standard deviation are the same for the sample and the fitted PDF. We choose the third parameter (shift) as an integer that minimizes the sum of the squares of the differences between sample points and fit at the points of the sample. We choose the two parameters (mean and standard deviation) of the fitted normal PDF to be the mean and standard deviation for the sample.

Figure A2 shows the CDF of experimental data and of the fit for the same Competitor sample as in Fig. A1. Like for scarce data, the comparison of CDFs is more meaningful than the comparison of PDFs.

## Appendix 2: Definition of inflation factor

Consider $\{x_1,\ldots x_n\}$, a sample of values of a random variable $X$. Use of small sample sizes (say 5) for estimating the variance of $X$, may lead to large statistical errors. It is important to estimate the error in the variance and adjust the variance to account for the error.

If the mean value of the population is unknown, then an unbiased estimator of the variance of the variable is:

$$s^2 = \frac{1}{n-1}\sum_{1}^{n}(x_i - \bar{x})^2 \qquad (1)$$

where $\bar{x}$ is the sample mean $= \frac{1}{n}\sum_{1}^{n} x_i$ .

The variance of the above estimator is (see Freund and Williams, 1966, pp. 151 (F.7a)):

$$\sigma^2_{s^2} = \frac{\mu_4}{n} - \frac{(n-3)\sigma^4}{n(n-1)}$$

where $\mu_4$ is the fourth moment of the population about the mean $= \frac{1}{n}\sum_{1}^{n}(x_i - \mu)^4$, and $\sigma^4$ is the square of the variance of the population.

The following equation is used to inflate the unbiased estimate of the variance obtained from equation (1):

$$s'^2 = s^2 + r \cdot \sigma_{s^2} = s^2 + r \cdot \sqrt{\frac{\mu_4}{n} - \frac{(n-3)\sigma^4}{n(n-1)}} \qquad (2)$$

where *r* is called the *inflation factor*.

When both the mean and standard deviation of the population are unknown, we use the corresponding estimates of these values in Eq. (2). Then the variance of the estimated variance becomes:

$$\bar{\sigma}^2_{s^2} = \frac{1}{n^2}\sum_{1}^{n}(x_i - \bar{x})^4 - \frac{(n-3)}{n(n-1)^3}\left(\sum_{1}^{n}(x_i - \bar{x})^2\right)^2 .$$

The inflated estimate of the variance becomes:

$$s'^2 = s^2 + r \cdot \bar{\sigma}_{s^2} .$$

Table A1 presents the standard deviation of the data to be fitted, before and after we inflate the standard deviation. An inflation factor of 0 corresponds to no inflation. In Table A1, the increase in inflated standard deviation does not vary linearly with the inflation factor, but the increase in inflated variance does.

Table A1: Inflated standard deviation for Rosca's and Competitor's data; the mean of Rosca's data is 33.10 while the mean for Competitor's data is 35.08.

| Inflation factor | Inflated standard deviation | |
|---|---|---|
| | Rosca Data | Competitor Data |
| 0 | 6.21 | 6.30 |
| 1 | 6.76 | 6.75 |
| 2 | 7.26 | 7.17 |
| 15 | 12.02 | 11.30 |

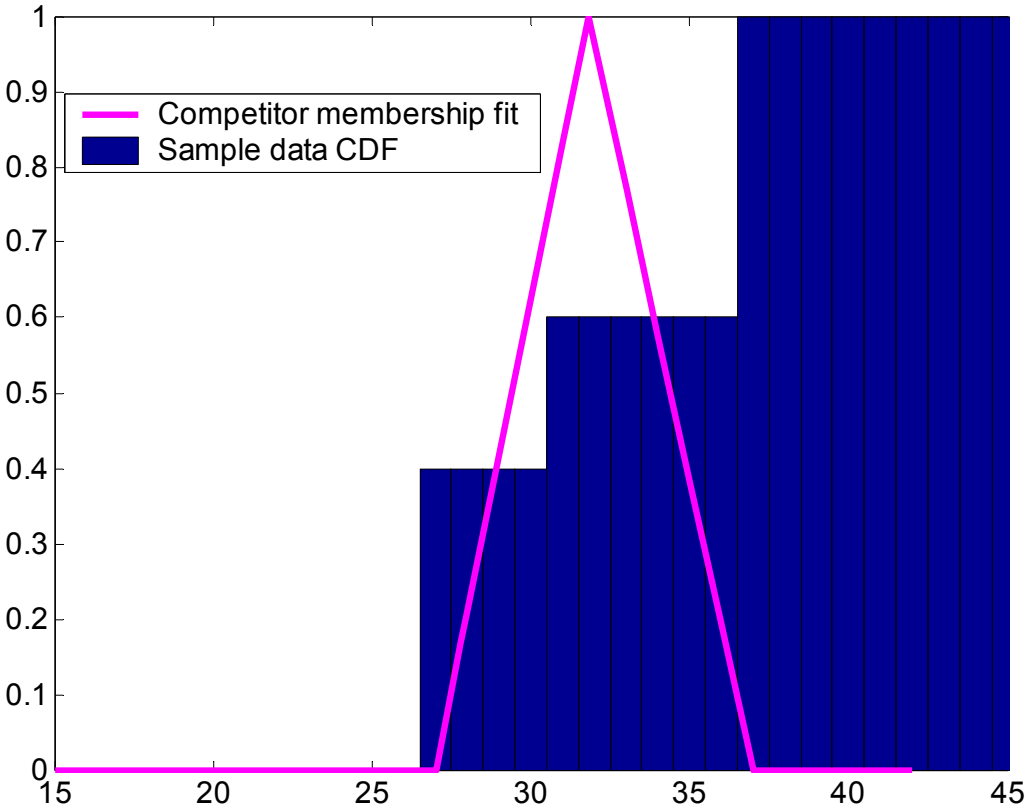



Figure A1: Triangular membership function (solid line) fitted to the sample of 5 from the Competitor's experiments [ 27  37  37  27  31] and sample cumulative histogram.

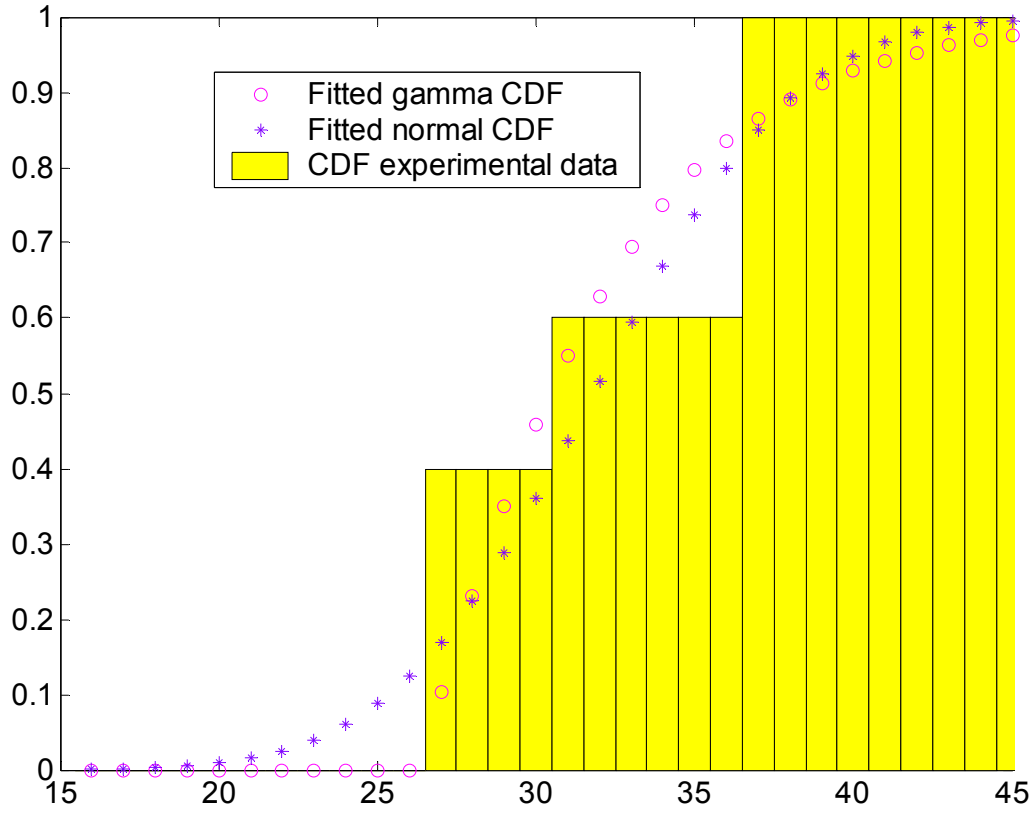



Figure A2: Experimental data CDF (bars), fitted shifted gamma CDF (circles) and fitted normal CDF (asterisks) for the same data as in Fig. A1.