

**PROCEEDINGS OF
THE NSF WORKSHOP ON
RELIABLE ENGINEERING COMPUTING
MODELING ERRORS AND UNCERTAINTY
IN ENGINEERING COMPUTATIONS**

FEBRUARY 22-24, 2006 | SAVANNAH, GEORGIA USA

EDITORS

Rafi L. Muhanna
Robert L. Mullen

■ NSF WORKSHOP ON RELIABLE ENGINEERING COMPUTING ■ MODELING ERRORS AND UNCERTAINTY IN ENGINEERING COMPUTATIONS

PROCEEDINGS: REC 2006 ORGANIZATION

EDITORS

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

WORKSHOP SPONSORS

National Science Foundation
Sun Microsystems
Georgia Institute of Technology

HONORARY WORKSHOP CO-CHAIRS

Ramon E. Moore
Eldon R. Hansen
Ivo Babuška

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
George Corliss, Marquette University
William Edmonson, North Carolina State University
Michael Eldred, Sandia National Labs
Scott Ferson, Applied Biomathematics
Baker Kearfott, University of Louisiana at Lafayette
Vladik Kreinovich, University of Texas at El Paso
Bernd Möller, Dresden University of Technology
Zissimos Mourelatos, Oakland University
Arnold Neumaier, University of Vienna, Austria
Efstratios Nikolaidis, University of Toledo
Andrzej Pownuk, Silesian University of Technology
Sigfried Rump, Technical University of Hamburg
Pol Spanos, Rice University
Mark Stadtherr, University of Notre Dame
William Walster, Sun Microsystems

LOCAL WORKSHOP ORGANIZING COMMITTEE

Julie Clayton
Natalie Cosner
Kimberly Gaither
Jillison Parks
Ashlee Peck
Michael Potter
Patricia Potter

■ NSF WORKSHOP ON RELIABLE ENGINEERING COMPUTING ■
MODELING ERRORS AND UNCERTAINTY IN ENGINEERING COMPUTATIONS

PROCEEDINGS: TABLE OF CONTENTS

- ii. **Workshop Organization**
- iii. **Table of Contents**
- v. **Preface**

- 1. **Discrete Mechanics on Interval Algebra**
F. Tonon

- 25. **Checking Computation of Numerical Functions by the Use of Functional Equations**
F. Vainstein and C. Jones

- 39. **A Library to Taylor Models for PVS Automatic Proof Checker**
Francisco Chaves and Marc Daumas

- 53. **Reliability of Structural Reliability Estimation**
I. Elishakoff and R. Santoro

- 65. **A Computational Environment for Decision Making Systems with Interval Matrices**
M. Nooner and Chenyi Hu

- 75. **Outlier Detection in Geodetic Applications with Respect to Observation Imprecision**
I. Neumann, H. Kutterer and S. Schön

- 91. **Modeling Hysteris in CLIP – The Two Tanks Problem**
D.K. Wittenberg and T.J. Hickey

- 113. **Worst Case Bounds in the Presence of Correlateds Uncertainty**
A. Neumaier

- 115. **Modeling Correlation and Dependence among Intervals**
S. Ferson and V. Kreinovich

- 127. **How to Take into Account Dependence Between the Inputs: From Interval Computations to Constraint-Related Set Computations, With Potential Applications to Nuclear Safety, Bio- and Geosciences**
M.Ceberio, S. Ferson, V. Kreinovich, S. Chopra, G. Xiang

- 155. **Validated Solution of Initial Value Problems for ODEs with Interval Parameters**
Y. Lin and M.A. Stadtherr

- 169. **Online Implementation of a Robust Controller using Hybrid Global Optimization Techniques**
Nataraj S.V. Paluri and N. Kubal

- 179. **Reduction in Space Complexity and Error Detection/Correction of Fuzzy Controller**
F. Vainstein, V. Osorio, E. Marte and R. Romero

- 189. **Interval Arithmetic Logic Unit for Signal Processing and Control Application**
W.W. Edmonson, R. Gupte, J. Gianchandani, S. Ocloo, and W. E. Alexander

- 197. **Interval-based Robust Statistical Techniques for Non-Negative Convex Functions, with Application to Timing Analysis of Computer Chips**
M. Orshansky, W. Wang, G. Xiang, V. Kreinovich

■ NSF WORKSHOP ON RELIABLE ENGINEERING COMPUTING ■
MODELING ERRORS AND UNCERTAINTY IN ENGINEERING COMPUTATIONS

PROCEEDINGS: TABLE OF CONTENTS

- 213. On Reliability of Higher-Order FEM in Fluid-Structure Interaction Problems**
P. Svacek
- 229. Interval Finite Element Methods: New Directions**
Muhanna, Solin, Kreinovich J. Chessa, R. Araiza, G.Xiang
- 245. Bounding the Response of Mechanical Structures with Uncertainties in all the Parameters**
E.D. Popova, R. Yankov, Z. Bonev
- 267. Overview of Reliability Analysis and Design Capabilities in DAKOTA**
M.S. Eldred, B.J. Bichon, B.M. Adams
- 293. Semantic Tolerance Modeling Based on Modal Interval Analysis**
Y. Wang
- 319. Why are Intervals and Imprecision Important in Engineering Design?**
J.M. Aughenbaugh and C. J. J. Paredis
- 341. Computational Methods for Decision Making Based on Imprecise Information**
M. Bruns, C.J.J. Paredis, and S. Ferson
- 369. Sampling Without Probabilistic Model**
M. Beer
- 391. Non-Probabilistic Design Optimizations with Insufficient Data**
Z.P. Mourelatos and J. Zhou
- 419. Prediction of Uncertain Structural Responses with Fuzzy Time Series**
B. Moeller and Uwe Reuter
- 439. Boundary Element Analysis of Systems Using Interval Methods**
B.F. Zalewski, R.L. Mullen, R.L. Muhanna
- 457. Reliable Dynamic Analysis of Transportation Systems**
M. Modares, R.L. Mullen and D.A. Gasparini
- 469. Geometric Uncertainty in Truss Systems: An Interval Approach**
R.L. Muhanna, A. Erdolen and R.L. Mullen
- 477. Prediction of Deflection for Prestressed Concrete Girders Using a Bayesian Approach**
X.J. Chen, C.W. Shen and L.J. Jacobs
- A. Author Index**

■ NSF WORKSHOP ON RELIABLE ENGINEERING COMPUTING ■ MODELING ERRORS AND UNCERTAINTY IN ENGINEERING COMPUTATIONS

PROCEEDINGS: PREFACE

Across all branches of engineering, computational methods share the need for Reliable results. Reliability can be achieved only if all sources of errors, approximations, and uncertainty are accounted for.

These proceedings embody the papers presented at the NSF workshop on Modeling Errors and Uncertainty in Engineering Computations hosted by The Center for Reliable Engineering Computing at the Georgia Institute of Technology. This NSF workshop focuses on the integration of the treatment of modeling errors and uncertainty into engineering computations. 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.

The objective of the workshop is to promote cross-disciplinary research in the area of treatment of modeling errors and parameter uncertainty and their mitigation in engineering software. The participants represent a truly interdisciplinary group: mathematicians, computer scientists, risk analysts, 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 main topics of the workshop include:

1. Errors in algorithms and computations.
2. Mitigation of various sources of errors in engineering calculations.
3. Integrating the treatment of modeling errors and uncertainty into design.

The work presented represent significant step towards achieving the goal of true reliability in engineering calculations.

The sponsors of this workshop are:

- National Science Foundation
- 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



Using Extended Interval Algebra in Discrete Mechanics

Fulvio Tonon

Department of Civil Engineering, University of Texas, Austin (USA), tonon@mail.utexas.edu

Abstract: Discrete mechanics deals with discrete mechanical systems, such as cellular automata, in which time proceeds in integer steps and the configuration space is discrete. Directly modeling discrete mechanical systems is a well known alternative to starting from a continuous setting, discretizing the model, and finally force the model to the finite alphabet of a computer. The time evolution of discrete dynamical systems, however, can be calculated exactly. In order to take into account imprecision in the input data and the need to accommodate a finite alphabet, extended interval analysis is introduced in the discrete mechanical systems formulation developed by Baez and Gilliam. It is shown how the Euler-Lagrange equation must be modified when working with interval input.

Keywords: Discrete mechanical systems, cellular automata, interval analysis.

1. Introduction

Baez and Gilliam (1994) and Gilliam (1996) developed an algebraic approach to the mechanics of discrete mechanical systems, that is, systems such as cellular automata (CA) (von Neumann, 1951), in which time evolution proceeds in integer steps and the state space is a finite set. By substituting algebraic geometry concepts for differential geometry concepts, the authors derived an analog for the Euler-Lagrange equation, a version of Noether's theorem, and symplectic techniques applicable to this context. They also gave a definition of complete integrability for a smooth mechanical system on a smooth real affine algebraic variety, and gave a criterion for the complete integrability of such systems. Additionally, they showed that, as the time steps of a discrete system decrease to zero, a solution of the discrete system converges uniformly to a solution of the corresponding continuous system. These Lagrangian and symplectic techniques allow one to use computers for *exactly* simulating discrete mechanical systems that take values in a commutative ring, k , as opposed to approximately simulating physical systems by numerically solving differential equations: let us expand on this crucial point.

One of the first uses of digital computers was to approximately simulate physical systems by numerically solving differential equations. This approach leads to numerical computation that is at least three levels removed from the physical world represented by those differential equations:

- 1) As a first step, one models a physical phenomenon using a differential equation (or a system of differential equations) or a variational principle.

© 2006 by authors. Printed in USA.

- 2) Then, one obtains the algebraic forms of the differential equation(s) or variational principle by forcing them into the mold of discrete time and space; and
- 3) Finally, in order to commit those algebraic forms to algorithms, one projects real-valued variables onto finite computer words, thus introducing round-off during computation and truncation.

Since at one end of the chain is the original physical system and at the other end is another physical system (a computer), physicists wondered whether there was a less roundabout approach to modeling physics (Toffoli, 1984; Toffoli and Margolus, 1987). Indeed, the moment one gives up symbolic manipulation as a major motive for using differential equations, one starts wondering whether one should keep them as a starting point for numerical modeling altogether. Adopting a totally different approach, CA have been proposed as a modeling tool that is isomorphic to the available and foreseeable computational resources (e.g., Toffoli and Margolus, 1987) and that is prototypical for complex interacting systems. Because of the intrinsic discreteness of CA, numerical integration is an *exact* process (there are no truncation or round-off errors), and thus the results that one obtains have the force of theorems. In other words, any properties that one discovers through simulation are guaranteed to be properties of the model itself rather than a simulation artifact (Toffoli, 1984). However, the lack of a rational and physics-based way to define evolution rules for CAs hindered their application to mechanics. Baez's and Gilliam's algebraic approach to discrete mechanical systems for the first time provides for this rational and physics-based way to define evolution rules, and shows how CAs can be seen as a subset of discrete mechanical systems.

A large body of literature has been devoted to estimating the errors introduced in Step 2 above. For example, Dow (1998), Oden *et al.* (2005) and a recent issue of the journal *Computer Methods in Applied Mechanics and Engineering* (2006) give a recent overview of results in the finite element discretization method. Peraire and coworkers have started developing algorithms for calculating guaranteed bounds on these errors (Sauer-Budge *et al.*, 2004; Xuan *et al.*, 2006); however (based on the published literature), their calculations are performed in floating-point arithmetic. Errors involved in step 3 have been vigorously attacked by the "reliable computing" community using interval analysis started by Warmus (1956) and Moore (1966); the reader I referred to the journal *Reliable Computing* (formerly *Interval Computations*) and to the web site (www.cs.utep.edu/interval-comp/main.html) for up-to-date information. Both types of errors are to be addressed during verification and validation of numerical models (Oberkampf *et al.*, 2003).

Discrete mechanical systems avoid these issues associated with Steps 2 and 3. On the other hand uncertainty may affect the available information on initial and boundary conditions, as well as information on a system's parameters. Moreover, when using finite computer words for a physical quantity (even if it is known exactly), real values must be truncated. Therefore, it seems worthwhile to *exactly* extend such uncertain information to a system's behavior: if this is not possible, guaranteed bounds on the system's evolution should be calculated. In this paper, it is

assumed that information on a physical quantity of interest is given as an interval on k ; we will refer to this assumption as *imprecision*. Generalized interval arithmetic (Dimitrova *et al.* 1992, Gardenes *et al.*, 1980a and 1980b, 1981, 1982, 1986, 2001; Kaucher, 1973; Kaucher *et al.*, 1977, 1980, 1977; Markov 1992, 1995, 1997, Ortolfo, 1969; Popova 1994, 2000, 2001, 2005; Popova and Ullrich, 1996, 1998; Ratschek, 1970, 1971; Spaniol, 1970) is used to extend information in a validated way because generalized interval arithmetic is an algebraically closed system.

For completeness, Talasila *et al.* (2004a, 2004b) have attempted to extend Baez's and Gilliam's work to floating point numbers, but have eventually developed a different theory based on discrete calculus. Finally, a note of caution: the term "discrete mechanics" is also adopted in the literature to denote mechanical systems whose configuration space is continuous and whose evolution proceeds in finite time steps (e.g., Marsden and West, 2001 and references therein); these systems are frequently used to develop structure-preserving and numerically stable time integrators.

In the following sections, some basic notions of discrete mechanics are recalled with more background definitions and explanation than in the available literature, so that these notions can be more easily grasped by an engineering audience. A simple example of a linear harmonic oscillator is used to highlight the properties of a discrete mechanical system. Likewise, the basic algebra of generalized intervals is reviewed. Subsequently, the discrete Euler-Lagrange equation is modified in order to work with generalized intervals and the harmonic oscillator example is extended to accommodate imprecise input values.

2. Basic Notions in Discrete Mechanics (Gilliam, 1996; Lang, 2002)

The configuration space of discrete mechanical systems is required to be no more than a ring or a group, without specific topological or analytical properties that allow for the use of the common concepts of tangent and cotangent vectors, spaces, bundles, etc. Since algebraic analogs for these concepts will be needed, let us review some basic definitions from abstract algebra.

Recall that a group G is a set with an associative law of composition $(x, y) \rightarrow xy$, having a unit element, and such that for every element $x \in G$, there exists an inverse element $y \in G$ such that $xy = yx = e$. If the law of composition is also commutative, a commutative group is obtained. A homomorphism $f: G \rightarrow G'$ is a mapping between two groups, G and G' , that preserves the product, i.e. $f(xy) = f(x)f(y)$, and that maps the unit element of G into that of G' . An isomorphism is a bijective homomorphism: if there is an isomorphism between G and G' , then one writes $G \cong G'$ to indicate that G and G' are isomorphic.

A ring R is a set, together with two laws of composition called multiplication and addition, respectively, and written as a product and sum respectively, satisfying the following conditions:

- With respect to addition, R is a commutative group (zero denotes the additive unit element).
- The multiplication is associative, and has a unit element (denoted as “1”).
- For all x, y, z in R one has (distributivity)

$$(x + y)z = xy + yz \quad (1)$$

Also recall that a module M over a ring R is a commutative group, usually written additively, together with an operation of R on M , such that, for all $a, b \in R$ and $x, y \in M$ one has

$$(a + b)x = ax + bx \text{ and } a(x + y) = ax + ay \quad (2)$$

Finally, an algebra is a module M with a bilinear map (product) $g : M \times M \rightarrow M$

Let $\{E^i\}$ be family of commutative groups. Their direct sum $G = \bigoplus_{i=0}^{\infty} E^i$ is the set of all sequences $(\sigma_1, \sigma_2, \dots, \sigma_p, \dots)$ where $\sigma_i \in E_i$, and all but a finite number of σ_i 's are zero. The direct sum becomes a group when the sum of two elements is defined componentwise. A graded algebra is an algebra that can be written as $G = \bigoplus_{i=0}^{\infty} E^i$, and such that for $s = (\sigma_0, \sigma_1, \dots)$ and $r = (\rho_0, \rho_1, \dots)$ in G , the product in G is defined as $sr = (\sigma_0\rho_0, \sigma_0\rho_1 + \sigma_1\rho_0, \dots, \sum_{i+j=p} \sigma_i\rho_j, \dots)$, in such a way that if $\sigma_p \in E^p$ and $\rho_q \in E^q$, then the product $\sigma_p\rho_q \in E^{p+q}$. Tensor algebras (e.g., **[Error! Reference source not found.]**, page 76) are examples of graded algebras in which the product $\sigma_p\rho_q$ is the outer product of tensors σ_p and ρ_q of order p and q , respectively. Another example of graded algebra is the algebra of polynomial functions described below, in which the product $\sigma_p\rho_q$ is the product of polynomials σ_p and ρ_q of order p and q , respectively.

In discrete mechanics, rather than working directly with configuration space, one works with the algebraic functions on the configuration space, which form a commutative algebra A over configuration space. For example, if the configuration space is an n -dimensional vector space over a field k , then one would use the algebra of polynomial functions in n variables over k ,

$k[x_1, \dots, x_n] = \bigoplus_{i=0}^{\infty} E^i$, where $E^i = \left\{ \sum_{j=1}^n \lambda_j x_j^{n_j} : \sum_{j=1}^n n_j = i \right\}$; $\lambda_j \in k$. The analog of a vector field

on configuration space is then a derivation on A , that is, a k -linear map $v : A \rightarrow A$ such that

$\nu(ab) = a\nu(b) + b\nu(a)$ for all $a, b \in A$. In order to define differential forms on A , let us introduce the concept of differential.

Let $\Omega = \bigoplus_{i=0}^{\infty} \Omega^i$ be a graded algebra. The differential of Ω is a map $d : \Omega \rightarrow \Omega$ such that if $\omega \in \Omega^p$, then

$$d(\omega\mu) = d(\omega)\mu + (-1)^p \omega d(\mu) \quad (3)$$

$$dd(\omega) = 0 \quad (4)$$

Let A be a commutative k -algebra (e.g., the algebraic functions on the configuration space). The algebraic differential forms $\Omega(A) = \bigoplus_{i=0}^{\infty} \Omega^i(A)$ are the graded algebra, in which $\Omega^0(A) = A$, in which the product is written as a wedge product, and which are generated by A and by the elements da , where $a \in A$, with the relations:

$$d(\lambda a) = \lambda da, \quad d(a+b) = da + db$$

$$d(ab) = da \wedge b + a \wedge db, \quad a \wedge db = db \wedge a$$

$$da \wedge db = -db \wedge da \quad da \wedge da = 0$$

for all $a, b \in A$, $\lambda \in k$, with the last necessary only if 2 has no multiplicative inverse in k . A p -form is an element of $\Omega^p(A)$.

Since A is the equivalent of the configuration space, the space of histories is the algebra $H = A^{\otimes(T+1)} = A_0 \otimes \dots \otimes A_T$, where the algebras A_i are simply copies of A with A_i thought of as the functions on configuration space at time i . The Lagrangian for the system, \mathcal{L} , is a fixed element of $A \otimes A$. In the algebra H , the discrete analog for the action functional in classical mechanics is

$$S = \sum_{i=0}^{T-1} \mathcal{L}_i \quad (5)$$

where $\mathcal{L}_i = 1 \otimes \dots \otimes \mathcal{L} \otimes \dots \otimes 1$, with \mathcal{L} occupying the i th and $(i+1)$ th slots.

In order to derive Lagrange equations from S , one needs to differentiate S , and thus one needs 1-forms on the space of histories H . Since for any algebra, A , one has that: $\Omega^1(A \otimes A) = A \otimes \Omega^1(A) \oplus \Omega^1(A) \otimes A$, by induction:

$$\Omega^1(H) = \bigoplus_{i=1}^T A_0 \otimes \dots \otimes \Omega^1(A_i) \otimes \dots \otimes A_T \quad (6)$$

Let $d_i = p_i d$ where $p_i: \Omega^1(H) \rightarrow \Omega^1(H)$ is the projection on the i th summand. The variation of S is effected by the operator $\delta = \sum_{i=1}^{T-1} d_i$, which keeps the first and the second summand of H fixed. Now, since $\mathcal{L} = a \otimes b$ with $a, b \in A$:

$$\begin{aligned} d_i \mathcal{L}_i &= p_i d (1 \otimes \dots \otimes \mathcal{L} \otimes \dots \otimes 1) = p_i (0 \otimes \dots \otimes d\mathcal{L} \otimes \dots \otimes 0) \\ &= p_i \left(\left(0 \otimes \dots \otimes da \otimes b \otimes \dots \otimes 0 \right) \oplus \left(0 \otimes \dots \otimes a \otimes db \otimes \dots \otimes 0 \right) \right) \\ &= \left(0 \otimes \dots \otimes da \otimes 0 \otimes \dots \otimes 0 \right) \oplus \left(0 \otimes \dots \otimes a \otimes 0 \otimes \dots \otimes 0 \right) \end{aligned} \quad (7)$$

and

$$\begin{aligned} d_i \mathcal{L}_{i-1} &= p_i \left(\left(0 \otimes \dots \otimes da \otimes b \otimes \dots \otimes 0 \right) \oplus \left(0 \otimes \dots \otimes a \otimes db \otimes \dots \otimes 0 \right) \right) \\ &= \left(0 \otimes \dots \otimes 0 \otimes b \otimes \dots \otimes 0 \right) \oplus \left(0 \otimes \dots \otimes 0 \otimes db \otimes \dots \otimes 0 \right) \end{aligned} \quad (8)$$

with $d_j \mathcal{L}_i = 0$ for $j \neq i, j \neq i-1$.

The variation of S is thus:

$$\delta S = \delta \sum_{i=0}^{T-1} \mathcal{L}_i = \sum_{i=0}^{T-1} \delta \mathcal{L}_i = \sum_{i=1}^{T-1} d_i \mathcal{L}_i + d_i \mathcal{L}_{i-1} \quad (9)$$

Finally, Eqs. (7) and (8) indicate that the last sum in Eq. (9) is actually a direct sum. Thus:

$$\delta S = 0 \Rightarrow d_i \mathcal{L}_i + d_i \mathcal{L}_{i-1} = 0 \quad (10)$$

Eq. (10) is the Euler-Lagrange equation for discrete systems. This 1-form does not vanish on the whole space of histories H , but only on the trajectories that satisfy the equations of motion. Since the Lagrangian is an element of $A \otimes A$, the equations of motion give the configuration at

the i -th time step as a function of the previous two time steps $i-1$ and $i-2$. This is formalized as a homomorphism $\varphi: A_2 \rightarrow A_0 \otimes A_1$, which defines a homomorphism $\Phi: A_1 \otimes A_2 \rightarrow A_0 \otimes A_1: a \otimes 1 \mapsto 1 \otimes a$ and $1 \otimes a \mapsto \varphi(a)$. One says that φ or Φ satisfies the equation of motion provided

$$\Phi_* d_i \mathcal{L} + d_i \mathcal{L}_{i-1} = 0; \quad (11)$$

where $\Phi_*: \Omega^1(A_1 \otimes A_2) \rightarrow \Omega^1(A_0 \otimes A_1)$ is the map induced by Φ , and d_i is the restriction of d_i on H to its sub-algebras $A_1 \otimes A_2$ and $A_0 \otimes A_1$.

EXAMPLE (modified from Baez and Gilliam, 1994). Let the base ring k be the ring of rational numbers, \mathbb{Q} , so that, in particular, 2 has an inverse. Consider the case of a particle in a polynomial potential constrained to move along a line with coordinate q . The algebra of functions on configuration space is $A \cong k[q] = \{\lambda_0, \lambda_0 + \lambda_1 q, \lambda_0 + \lambda_2 q^2, \lambda_0 + \lambda_1 q + \lambda_2 q^2, \lambda_0 + \lambda_3 q^3, \dots\}$, so that $A \otimes A \cong k[q_1, q_2]$, the polynomials in 2 variables over k , and $H \cong k[q_0, \dots, q_T]$, the polynomials in $T+1$ variables over k . Consider the Lagrangian \mathcal{L}_i (written here as a polynomial function) for a particle in a polynomial potential V as a function of consecutive positions q_i and q_{i+1} of the particle:

$$\mathcal{L}_i = \mathcal{L}(q_i, q_{i+1}) = \frac{1}{2} m \dot{q}_i^2 - V(q_i) \quad (12)$$

where one defines $\dot{q}_i = q_{i+1} - q_i$, and where m is in k , and represents the mass of the particle. Since $d_i \mathcal{L} = \partial_{q_i} \mathcal{L}(q_1, q_2) dq_i$, $i = 1, 2$, one obtains:

$$d_i \mathcal{L}_i = -m(q_{i+1} - q_i) - V'(q_i) dq_i = m \dot{q}_i dq_i \quad (13)$$

Likewise

$$d_i \mathcal{L}_{i-1} = m(q_i - q_{i-1}) = m \dot{q}_{i-1} dq_i \quad (14)$$

The Euler-Lagrange equation is thus:

$$m(\dot{q}_i - \dot{q}_{i-1}) = -V'(q_i), \quad (15)$$

which is the discrete analog for Newton's law, and yields the time evolution map

$$\varphi(q_2) = q_1 + \dot{q}_0 - m^{-1}V'(q_1) = 2q_1 - q_0 - m^{-1}V'(q_1) \quad (16)$$

and homomorphism Φ

$$\Phi(q_1) = q_1, \quad \Phi(q_2) = \varphi(q_2) = 2q_1 - q_0 - m^{-1}V'(q_1) \quad (17)$$

$$\Phi_*(dq_1) = dq_1, \quad \Phi_*(dq_2) = 2dq_1 - dq_0 - m^{-1}V''(q_1)dq_1 \quad (18)$$

Let us check that the time evolution map satisfies the equation of motion:

$$\begin{aligned} \Phi_*d_1\mathcal{L}_1 + d_1\mathcal{L}_0 &= \Phi_*\partial_{q_1}\mathcal{L}(q_1, q_2)dq_1 + \partial_{q_1}\mathcal{L}(q_0, q_1)dq_1 = \\ &= \Phi_*\left(\left(-m(q_2 - q_1) - V'(q_1)\right)dq_1\right) + m(q_1 - q_0)dq_1 \\ &= \left(-m\left(2q_1 - q_0 - m^{-1}V'(q_1) - q_1\right) - V'(q_1)\right)dq_1 + m(q_1 - q_0)dq_1 = 0 \end{aligned}$$

It can be seen that homomorphism Φ_* pulls back $d_1\mathcal{L}_1$ from $\Omega^1(A_1 \otimes A_2)$ to $\Omega^1(A_0 \otimes A_1)$. In this simple case, this entails substituting the expression for the time evolution map (16) into the expression Euler-Lagrange equation (15).

Figure 1 shows the evolution of a linear harmonic oscillator with: $m = 1$, $q_0 = 8$; $q_1 = 16$; $V = \frac{1}{2}sq^2$ (where s is the spring stiffness), $s = 1$. The mass takes positions: $\{8, 16, 8, -8, -16, -8, 8, 16, 8, -8, \dots\}$, and the mass revisits the same location in space after 6 steps. Notice that this time integration is *exact*, and can be *exactly* reversed.

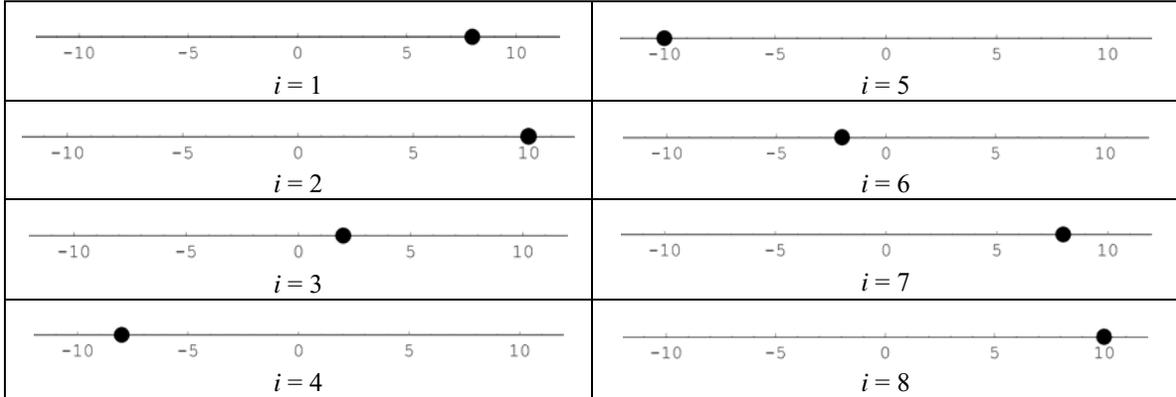


Figure 1. Evolution of a linear harmonic oscillator with $m = 1$, $q_0 = 8$; $q_1 = 16$; $V = \frac{1}{2}sq^2$, $s = 1$.

However, if the ratio s/m is not an integer, then the mass never revisits the same location twice; for example, for $s/m = 1/3$, the coordinates of the particle are (all calculations in this paper were carried out using Mathematica exact arithmetic): $\{8, 16, 56/3, 136/9, 176/27, -344/81, -3304/243, -13424/729, -37384\sqrt{2187}, -66104/6561, 5936/19683, 624616/59049, 3069656/177147\dots\}$. The numbers of digits in the numerator and denominator keep increasing at each time step as shown in Figure 2. In Figure 2, each digit in the $[0, 9]$ range is assigned a color. Each digit of the numerator occupies a cell, and numerator digits for the i -th step occupy the first cells from the left of the $(2i-1)$ -th row. Likewise, denominator digits for the i -th step occupy the first cells from the left of the $2i$ -th row.

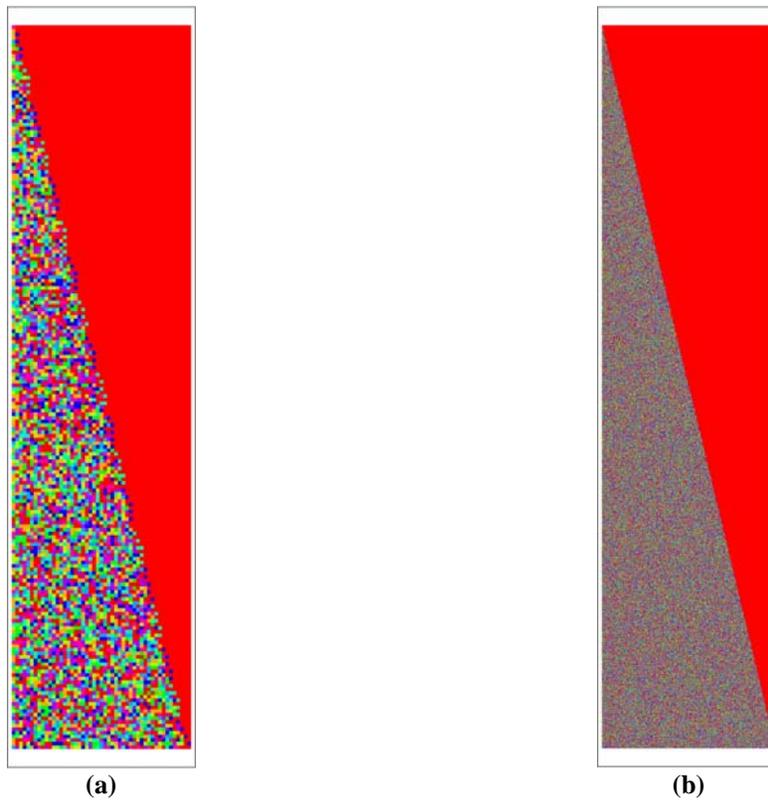


Figure 2. Graphical representation of the digits in the numerator and denominator of a particle coordinates. The particle is a linear harmonic oscillator with $m = 1$, $q_0 = 8$; $q_1 = 16$; $V = \frac{1}{2}sq^2$, $s = 1/3$.

(a) First 100 time steps; (b) First 1,000 time steps.

Since there is no periodicity in the pattern of digits, the CA depicted in Figure 2 belongs to the third CA class in the following Wolfram's classification (Wolfram, 1985a, 1985b):

- 1) Class 1: All components attain the same state; the final state is unique and unaffected by any change to the initial state;
- 2) Class 2: Simple stable states or periodic and separated structures emerge; small changes in the initial state only affect a fixed finite region around the area in which the values were changed;
- 3) Class 3: Chaotic non-periodic patterns are generated; a minimal perturbation to the initial state affects arbitrarily large regions; or
- 4) Class 4: Complex, localized, propagating structures are formed; some perturbations to some initial configurations appear to propagate arbitrarily far, whereas others die out.

Figures 3a through 3c show all the positions occupied by the particle after 100, 1,000, and 10,000 time steps. It can be seen that these positions are closer one to the other around the extremes of the current oscillation range (Figures 3a and 3b), where the particle velocity is smaller. The particle positions form clusters separated by empty segments (Figure 3b). After the first two steps and within 10,000 time steps, the particle never occupies a position having an integer coordinate: it is an open question whether it will eventually occupy integer coordinate positions. Another open question is whether the particle will visit all positions between the extremes reached, say, after 1,000 iterations, or there will always be "holes" in between.

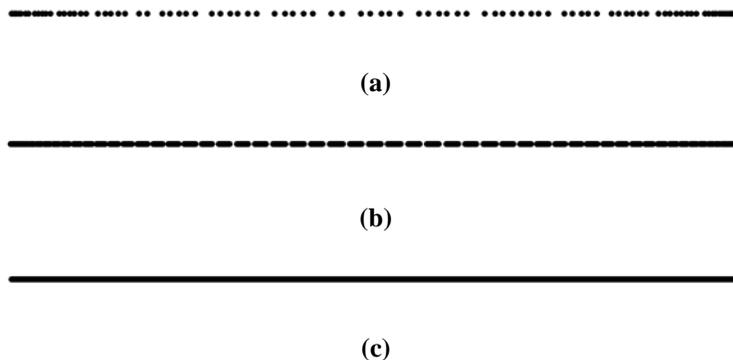


Figure 3. Cumulative positions occupied by a linear harmonic oscillator with $m = 1$, $q_0 = 8$; $q_1 = 16$; $V = \frac{1}{2}sq^2$, $s = 1/3$ (same as in Figure 2). (a) First 100 time steps; (b) First 1,000 time steps; (c) First 10,000 time steps (positions are indistinguishable at this scale).

Additionally, it is not possible to determine *a priori* the maximum and minimum coordinates reached by the particle for an infinite number of time steps. For any finite number of time steps,

the maximum and minimum coordinates are not symmetric about zero. For example, the maximum (minimum, resp.) coordinate after 100 time steps is approximately equal to 18.682435719981747 (-18.68289250959053), after 1,000 time steps it is approximately equal to 18.683060458305423 (-18.68289250959053), after 10,000 time steps it is approximately equal to 18.6839719940311 (-18.683972612054987). The maximum coordinate increases steadily, but the minimum coordinate remains constant between 100 and 1,000 time steps, and then it decreases further. Thus, even for a very simple linear harmonic oscillator without any forcing, it is impossible to find a shortcut to its range of oscillation: all we can do is to sit back and watch it evolve.

Let us now introduce some basic concepts of extended interval algebra and then see how the discrete mechanics formulation described in this section must be modified in the presence of imprecision.

3. Ordering of k and generalized interval arithmetic

In order to work with intervals, we need to introduce the concept of ordering. Let k be a ring. An ordering of k is a subset P of k having the following properties (Lang, 2002):

- 1) Given $x \in k$, either $x \in P$ or $x = 0$, or $-x \in P$, and these possibilities are mutually exclusive. In other words, k is a disjoint union of P , $\{0\}$, and $-P$.
- 2) If $x, y \in P$, then $x + y$ and $xy \in P$.

One also says that k is ordered by P and one calls P the set of positive elements. Let $x, y \in k$. Define $x < y$ (or $y > x$) to mean that $y - x \in P$; define $x \leq y$ to mean $x < y$ or $x = y$. Define $|x| = x$ if $x > 0$, and $|x| = -x$ if $x < 0$.

In generalized interval arithmetic, the set of proper intervals $\{[x^-, x^+] \mid x^- \leq x^+; x^-, x^+ \in k\}$ is extended by the set $\{[x^-, x^+] \mid x^- \geq x^+; x^-, x^+ \in k\}$ of improper intervals, thus obtaining the set $\mathcal{D} = \{\mathbf{x} = [x^-, x^+]; x^-, x^+ \in k\} \cong k^2$ of all ordered couples called generalized intervals (strictly speaking, generalized interval arithmetic is defined over the reals, but it is easy to see that the operations and properties used below are valid over any ordered ring, k). Denote the set of generalized intervals that involve zero by $\mathcal{F} = \{\mathbf{x} \in \mathcal{D} \mid x^- x^+ \leq 0\}$. In this paper, intervals are written in boldface type.

From a physical viewpoint, a proper interval, \mathbf{x} , can also be seen as a set $\mathbf{x} = [x^-, x^+] = \{x \in k \mid x^- \leq x \leq x^+; x^-, x^+ \in k\}$ of possible values of a physical quantity of interest, say \mathcal{X} . Improper intervals are introduced to make interval algebra closed: if, at the end of a calculation sequence, \mathcal{X} turns out to be an improper interval, then this means that the possible set of values of \mathcal{X} is the empty set (more refined semantics has been developed in modal interval analysis (Gardenes *et al.*, 2001), but this is beyond the scope of this paper).

The “dual” is an important operator that reverses the endpoints of the intervals. Let $\mathbf{x} = [x^-, x^+] \in \mathcal{D}$; its dual is defined as $Dual(\mathbf{x}) = \mathbf{x}_- = [x^+, x^-] \in \mathcal{D}$. In order to simplify the formulae below, we use the functional notation introduced by Popova (2001). Define $\Lambda = \{+, -\}$, and, for μ and $\nu \in \Lambda$, define the (commutative) product $\lambda = \mu\nu \in \Lambda$ by $\lambda = \{+, \text{ if } \mu = \nu, - \text{ otherwise}\}$.

For $\lambda \in \Lambda$, define:

$$x^\lambda = \begin{cases} x^+ & \text{if } \lambda = + \\ x^- & \text{if } \lambda = - \end{cases} \quad \text{and} \quad \mathbf{x}_\lambda = \begin{cases} \mathbf{x} & \text{if } \lambda = + \\ \mathbf{x}_- & \text{if } \lambda = - \end{cases}. \quad (19)$$

The direction of an interval, $\tau(\mathbf{x})$, its sign, $\sigma(\mathbf{x})$, and its relative magnitude, $\nu(\mathbf{x})$, are defined as, respectively:

$$\tau(\mathbf{x}) = \begin{cases} + & \text{if } x^- < x^+ \\ - & \text{if } x^- > x^+ \\ \pm & \text{if } x^- = x^+ \end{cases} \quad \sigma(\mathbf{x}) = \begin{cases} + & \text{if } x^{-\tau(\mathbf{x})} > 0 \\ - & \text{if } x^{\tau(\mathbf{x})} < 0 \end{cases} \quad \nu(\mathbf{x}) = \begin{cases} + & \text{if } |x^+| > |x^-| \\ - & \text{if } |x^+| < |x^-| \\ \pm & \text{if } |x^+| = |x^-| \end{cases} \quad (20)$$

Addition, multiplication, and subtraction of intervals are defined as follows:

$$\mathbf{x} + \mathbf{y} = [x^- + y^-, x^+ + y^+], \text{ for } \mathbf{x}, \mathbf{y} \in \mathcal{D} \quad (21)$$

$$\mathbf{xy} = \begin{cases} \left[x^{-\sigma(\mathbf{y})} y^{-\sigma(\mathbf{x})}, x^{\sigma(\mathbf{y})} y^{\sigma(\mathbf{x})} \right] & \mathbf{x}, \mathbf{y} \in \mathcal{D} \setminus \mathcal{I} \\ \left[x^{\sigma(\mathbf{x})\tau(\mathbf{y})} y^{-\sigma(\mathbf{x})}, x^{\sigma(\mathbf{x})\tau(\mathbf{y})} y^{\sigma(\mathbf{x})} \right] & \mathbf{x} \in \mathcal{D} \setminus \mathcal{I}, \mathbf{y} \in \mathcal{I} \\ \left[x^{-\sigma(\mathbf{y})} y^{\sigma(\mathbf{y})\tau(\mathbf{x})}, x^{\sigma(\mathbf{y})} y^{\sigma(\mathbf{y})\tau(\mathbf{x})} \right] & \mathbf{x} \in \mathcal{I}, \mathbf{y} \in \mathcal{D} \setminus \mathcal{I} \\ \left[\min\{x^- y^+, x^+ y^-\}, \max\{x^- y^-, x^+ y^+\} \right] & \mathbf{x}, \mathbf{y} \in \mathcal{I}, \tau(\mathbf{x}) = \tau(\mathbf{y}) \\ 0 & \mathbf{x}, \mathbf{y} \in \mathcal{I}, \tau(\mathbf{x}) = -\tau(\mathbf{y}) \end{cases} \quad (22)$$

$$\mathbf{x} - \mathbf{y} = \mathbf{x} + (-1)\mathbf{y} = \left[x^- - y^+, x^+ - y^- \right], \text{ for } \mathbf{x}, \mathbf{y} \in \mathcal{D};$$

$$-1 \text{ is the additive unit of } 1 \in k \quad (23)$$

Addition and multiplication are commutative and associative, and have unit elements, namely $[0, 0]$ for addition and $[1, 1]$ for multiplication. Any element $\mathbf{x} \in \mathcal{D}$ has a unique inverse element for addition, namely $-\mathbf{x}_- : \mathbf{x} - \mathbf{x}_- = 0$. Additionally, conditional distributivity laws hold and have been summarized by Popova (2001). To illustrate, let us introduce a law, which will be used in

the examples that follow. Denote $\hat{\mu}(\mathbf{x}) = \begin{cases} \sigma(\mathbf{x}) & \text{if } \mathbf{x} \in \mathcal{D} \setminus \mathcal{I} \\ \nu(\mathbf{x})\tau(\mathbf{x}) & \text{if } \mathbf{x} \in \mathcal{I} \setminus \{0\} \end{cases}$. For $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D} \setminus \{0\}$ and $\mathbf{s} = \mathbf{x}_1 + \mathbf{x}_2$, if $\mathbf{s} \in \mathcal{D} \setminus \mathcal{I}$, $\mathbf{y} \in \mathcal{D} \setminus (\mathcal{I} \cup k)$, then

$$(\mathbf{x}_1 + \mathbf{x}_2)\mathbf{y} = \mathbf{x}_1\mathbf{y}_{\hat{\mu}(\mathbf{x}_1)\hat{\mu}(\mathbf{s})} + \mathbf{x}_2\mathbf{y}_{\hat{\mu}(\mathbf{x}_2)\hat{\mu}(\mathbf{s})} \text{ iff} \quad (24)$$

either $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D} \setminus \mathcal{I}$, or

$\mathbf{x}_i \in \mathcal{I} \setminus \{0\}$ for some $i \in \{1, 2\}$ and either $x^- = 0$ or $x^+ = 0$ for all $\mathbf{x}_i \in \mathcal{I} \setminus \{0\}$.

Thus, \mathcal{D} is a conditional ring, and one could be tempted to blindly use all results derived by Baez and Gilliam (1994) using \mathcal{D} as the ring in which the system takes values. However, since the addition unit of \mathbf{x} is $-\mathbf{x}_-$, $\mathbf{x} - \mathbf{x} \neq 0$ unless \mathbf{x} is degenerate, i.e. $\tau(\mathbf{x}) = \pm$, and only conditional distributivity applies. Finally, we will use the following properties:

- An element $\mathbf{x} \in \mathcal{D} \setminus \mathcal{I}$ is a multiplication unit in \mathcal{D} iff all $x \in \mathbf{x}$ are units in k ; the multiplicative inverse of \mathbf{x} is then \mathbf{x}_-^{-1} with

$$1/\mathbf{x} = \left[1/x^+, 1/x^- \right]; 1/\mathbf{x}_- = \left[1/x^-, 1/x^+ \right]$$

- The dual operator is distributive with respect to finite addition ($Dual(\mathbf{x} + \mathbf{y}) = Dual(\mathbf{x}) + Dual(\mathbf{y})$) and multiplication ($Dual(\mathbf{xy}) = Dual(\mathbf{x})Dual(\mathbf{y})$) and is an automorphism.

Let $f(x): k \rightarrow k$ be a rational function. The generalized rational interval extension of f is the interval function $fR(\mathbf{x}): \mathcal{D} \rightarrow \mathcal{D}$ defined by the syntactic expression of f , where the variables in k are replaced by generalized intervals, and operations on k are replaced by the operations between generalized intervals described above. Likewise, the derivative $R_f'(\mathbf{x}): \mathcal{D} \rightarrow \mathcal{D}$, if it exists, is defined by the syntactic expression of f' , only replacing the argument x by its interval counterpart \mathbf{x} , and its operations on k by their corresponding interval operations. The united extension, R_f , is defined as the range of function values

$$R_f(\mathbf{x}) = \left[\min_{x \in \mathbf{x}} f(x), \max_{x \in \mathbf{x}} f(x) \right]$$

In general, $fR(\mathbf{x}) \supseteq R_f(\mathbf{x})$. Similar definitions apply for multi-dimensional cases.

Since calculating the united extension is an NP-hard problem involving global optimization, generalized interval arithmetic will be used to carry out symbolic manipulations, and an algorithm due to Popova (2005) will be used to calculate interval extensions for the rational functions of interest in such a way that $fR(\mathbf{x}) = R_f(\mathbf{x})$. The following thus rewrites Baez's and Gilliam's results using \mathcal{D} as the ring in which the system takes values. Time evolution still proceeds in integer steps: if one is interested in the evolution of a system in the interval of time $[i, i+n]$, such evolution is just the union of the results at each time step in $[i, i+n]$. Imprecision in time measurement is accounted for by allowing time-related quantities to be intervals, e.g., the initial velocity. Future research will deal with the case in which such physical quantities are measured in a time interval, e.g., the initial velocity measured between time steps i , and $i+n$.

4. Euler-Lagrange equation

As in Section 2, let A be a commutative algebra over \mathcal{D} and let $\Omega(A) = \bigoplus_{i=1}^{\infty} \Omega^i(A)$ be the graded-commutative differential graded algebra on A with differential d , and product written as a wedge product. Let time take value in the discrete set $\{0, \dots, T\}$ and A_i be a copy of A representing the system at time i . In order to satisfy Newton's first and second laws, the Euler-

Lagrange equation on the space of histories $H = A^{\otimes(T+1)} = A_0 \otimes \dots \otimes A_T$ must be modified as follows:

$$Dual(d_i \mathcal{L}_i) + d_i \mathcal{L}_{i-1} = 0 \quad (25)$$

where $\mathcal{L} \in A \otimes A$ is the Lagrangian, $\mathcal{L}_i = 1 \otimes \dots \otimes \mathcal{L} \otimes \dots \otimes 1$ (in which \mathcal{L} occupies the i th and $(i+1)$ th slots); $p_i: \Omega^1(H) \rightarrow \Omega^1(H)$ is the projection on the i th summand of $\Omega^1(H) = \bigoplus_{i=1}^T A_0 \otimes \dots \otimes \Omega^1(A_i) \otimes \dots \otimes A_T$, and $d_i = p_i d$. The example that follows illustrates why the *Dual* operator is necessary in Eq. (25).

Recall that the time evolution map is formalized as a homomorphism $\Phi: A_1 \otimes A_2 \rightarrow A_0 \otimes A_1: a \otimes 1 \mapsto 1 \otimes a$ and $1 \otimes a \mapsto \varphi(a)$, where $\varphi: A_2 \rightarrow A_0 \otimes A_1$ is a homomorphism that formalizes the equation of motion. Similarly to Section 2, φ or Φ satisfies the equation of motion provided

$$Dual(\Phi_* d_1 \mathcal{L}_1) + d_1 \mathcal{L}_0 = 0; \quad (26)$$

where $\Phi_*: \Omega^1(A_1 \otimes A_2) \rightarrow \Omega^1(A_0 \otimes A_1)$ is the map induced by Φ .

The *Dual* operator in Eq. (26) is necessary in order to ensure that distributivity be a necessary condition for the evolution map to satisfy the equation of motion, as shown in the following example.

EXAMPLE (modified from Baez and Gilliam, 1994). Suppose 2 is a unit in k and that the algebra $A \cong \mathcal{D}[q]$, so that $A \otimes A \cong \mathcal{D}[q_1, q_2]$ and $H \cong \mathcal{D}[q_0, \dots, q_T]$, the polynomials in $T+1$ variables over \mathcal{D} . Consider the Lagrangian \mathcal{L}_i (written here as a polynomial function) for a particle in a polynomial potential \mathbf{V} as a function of consecutive positions \mathbf{q}_i and \mathbf{q}_{i+1} of the particle:

$$\mathcal{L}_i = \mathcal{L}(\mathbf{q}_i, \mathbf{q}_{i+1}) = \frac{1}{2} \mathbf{m} \dot{\mathbf{q}}_i^2 - \mathbf{V}(\mathbf{q}_i) \quad (27)$$

where one defines $\dot{\mathbf{q}}_i = \mathbf{q}_{i+1} - \mathbf{q}_i$ (so that $\dot{\mathbf{q}}_i = 0$ iff $\mathbf{q}_{i+1} = \mathbf{q}_i$), and where \mathbf{m} is a unit in \mathcal{D} representing the mass of the particle. Notice that $\mathcal{L}_i = 0$ iff $\frac{1}{2} \mathbf{m} \dot{\mathbf{q}}_i^2 = \mathbf{V}(\mathbf{q}_i)$. Since

$d_i \mathcal{L} = \partial_{q_i} \mathcal{L}(\mathbf{q}_1, \mathbf{q}_2) dq_i$, $i = 1, 2$ and the *Dual* operator is distributive with respect to sum and product, one obtains:

$$\begin{aligned} Dual(d_i \mathcal{L}_i) &= Dual\left(-\mathbf{m}(\mathbf{q}_{i+1} - \mathbf{q}_{i-}) - \mathbf{V}'(\mathbf{q}_i)_-\right) dq_i = \\ &= -\mathbf{m}_-(\mathbf{q}_{i+1} - \mathbf{q}_{i-})_- dq_i - \mathbf{V}'(\mathbf{q}_i) dq_i = -\mathbf{m}_- \dot{\mathbf{q}}_i_- dq_i - \mathbf{V}'(\mathbf{q}_i) dq_i = \\ &= -(\mathbf{m} \dot{\mathbf{q}}_i)_- dq_i - \mathbf{V}'(\mathbf{q}_i) dq_i \end{aligned}$$

Likewise

$$d_i \mathcal{L}_{i-1} = \mathbf{m}(\mathbf{q}_i - \mathbf{q}_{i-1-}) = \mathbf{m} \dot{\mathbf{q}}_{i-1} dq_i$$

The Euler-Lagrange equation is thus:

$$(\mathbf{m} \dot{\mathbf{q}}_i)_- - \mathbf{m} \dot{\mathbf{q}}_{i-1} + \mathbf{V}'(\mathbf{q}_i) = 0, \quad (28)$$

which correctly yields $\mathbf{V}'(\mathbf{q}_i) = 0$ iff $\dot{\mathbf{q}}_i = \dot{\mathbf{q}}_{i-1}$ (compare with Eq. (10)).

The discrete analog for Newton's second law is immediately derived:

$$\begin{aligned} (\mathbf{m} \dot{\mathbf{q}}_i)_- - \mathbf{m} \dot{\mathbf{q}}_{i-1} + \mathbf{V}'(\mathbf{q}_i) = 0 &\Leftrightarrow (\mathbf{m} \dot{\mathbf{q}}_i)_- = (\mathbf{m} \dot{\mathbf{q}}_{i-1})_- - \mathbf{V}'(\mathbf{q}_i)_- \\ &\Leftrightarrow \dot{\mathbf{q}}_i_- = \dot{\mathbf{q}}_{i-1-} - \mathbf{m}^{-1} \mathbf{V}'(\mathbf{q}_i)_- \\ &\Leftrightarrow \dot{\mathbf{q}}_i - \dot{\mathbf{q}}_{i-1} = -\mathbf{m}^{-1} \mathbf{V}'(\mathbf{q}_i)_- \\ &\Leftrightarrow \dot{\mathbf{q}}_i - \dot{\mathbf{q}}_{i-1-} = -\mathbf{m}_-^{-1} \mathbf{V}'(\mathbf{q}_i) \end{aligned}$$

which yields the time evolution map (compare with Eq. (16))

$$\varphi(\mathbf{q}_2) = \mathbf{q}_1 + \dot{\mathbf{q}}_0 - \mathbf{m}^{-1} \mathbf{V}'(\mathbf{q}_1)_- = 2\mathbf{q}_1 - \mathbf{q}_{0-} - \mathbf{m}_-^{-1} \mathbf{V}'(\mathbf{q}_1) \quad (29)$$

and homomorphism Φ (compare with Eqs. (17) and (18))

$$\Phi(\mathbf{q}_1) = \mathbf{q}_1, \quad \Phi(\mathbf{q}_2) = \varphi(\mathbf{q}_2) = 2\mathbf{q}_1 - \mathbf{q}_{0-} - \mathbf{m}_-^{-1} \mathbf{V}'(\mathbf{q}_1) \quad (30)$$

$$\Phi_*(dq_1) = dq_1, \quad \Phi_*(dq_2) = 2dq_1 - dq_0 - \mathbf{m}_-^{-1} \mathbf{V}''(\mathbf{q}_1) dq_1 \quad (31)$$

Let us check that the time evolution map satisfies the equation of motion:

$$\begin{aligned}
Dual(\Phi_* d_1 \mathcal{L}_1) + d_1 \mathcal{L}_0 &= Dual(\Phi_* \partial_{q_1} \mathcal{L}(\mathbf{q}_1, \mathbf{q}_2) dq_1) + \partial_{q_1} \mathcal{L}(\mathbf{q}_0, \mathbf{q}_1) dq_1 = \\
&= Dual\left(\Phi_* \left((-\mathbf{m}(\mathbf{q}_2 - \mathbf{q}_{1-}) - \mathbf{V}'(\mathbf{q}_1)_-) dq_1 \right)\right) + \mathbf{m}(\mathbf{q}_1 - \mathbf{q}_{0-}) dq_1 \\
&= \left((-\mathbf{m}(2\mathbf{q}_1 - \mathbf{q}_{0-} - \mathbf{m}_-^{-1} \mathbf{V}'(\mathbf{q}_1) - \mathbf{q}_{1-}) - \mathbf{V}'(\mathbf{q}_1)_-) dq_1 \right)_- + \mathbf{m}(\mathbf{q}_1 - \mathbf{q}_{0-}) dq_1 \\
&= \left((-\mathbf{m}(\mathbf{q}_1 - \mathbf{q}_{0-} - \mathbf{m}_-^{-1} \mathbf{V}'(\mathbf{q}_1)) - \mathbf{V}'(\mathbf{q}_1)_-) dq_1 \right)_- + \mathbf{m}(\mathbf{q}_1 - \mathbf{q}_{0-}) dq_1
\end{aligned}$$

whether or not this quantity is equal to zero depends on the actual input data, as the two following numerical examples of a harmonic oscillator show (assume $k = \mathbb{Q}$):

1. Example 2a: assume $\mathbf{q}_0 = [40, 45]$, $\mathbf{q}_1 = [50, 65]$; $\mathbf{V}(\mathbf{q}_1) = \frac{1}{2} \mathbf{s} \mathbf{q}_1^2$; $\mathbf{s} = [1/25, 6/25]$; $\mathbf{V}'(\mathbf{q}_1) = \mathbf{s} \mathbf{q}_1$; $\mathbf{m} = [1, 2]$. Then $-\mathbf{m}(\mathbf{q}_1 - \mathbf{q}_{0-} - \mathbf{m}_-^{-1} \mathbf{V}'(\mathbf{q}_1)) - \mathbf{V}'(\mathbf{q}_1)_- = [-38, -13] \subset -\mathbf{m}(\mathbf{q}_1 - \mathbf{q}_{0-}) = [-40, -10]$ because only sub-distributivity holds. As a result, $Dual(\Phi_* d_1 \mathcal{L}_1) + d_1 \mathcal{L}_0 = [-3, 2]$, and thus generalized interval arithmetic leads to the conclusion that the time evolution map does not satisfy the equation of motion.
2. Example 2b: assume $\mathbf{q}_0 = [11, 000, 11, 100]$, $\mathbf{q}_1 = [10, 000, 11, 000]$; $\mathbf{V}(\mathbf{q}_1) = \frac{1}{2} \mathbf{s} \mathbf{q}_1^2$; $\mathbf{s} = [1/5000, 3/5500]$; $\mathbf{V}'(\mathbf{q}_1) = \mathbf{s} \mathbf{q}_1$; $\mathbf{m} = [1, 2]$. Then $-\mathbf{m}(\mathbf{q}_1 - \mathbf{q}_{0-} - \mathbf{m}_-^{-1} \mathbf{V}'(\mathbf{q}_1)) - \mathbf{V}'(\mathbf{q}_1)_- = [100, 2000] = -\mathbf{m}(\mathbf{q}_1 - \mathbf{q}_{0-}) = [100, 2000]$ and the time evolution map satisfies the equation of motion.

More in general, if Eq. (24) applies, then the time evolution map is satisfied using generalized interval arithmetic iff $\hat{\mu}(\mathbf{x}_1) \hat{\mu}(\mathbf{s}) = +$ AND $\hat{\mu}(\mathbf{x}_2) \hat{\mu}(\mathbf{s}) = +$, with $\mathbf{x}_1 = \mathbf{q}_1 - \mathbf{q}_{0-}$, $\mathbf{x}_2 = -\mathbf{m}_-^{-1} \mathbf{V}'(\mathbf{q}_1)$, and $\mathbf{y} = -\mathbf{m}$. If one defines the momentum as $\mathbf{p}_i = \mathbf{m} \dot{\mathbf{q}}_i = \mathbf{m}(\mathbf{q}_{i+1} - \mathbf{q}_{i-})$, then these conditions are equivalent to the momentum having the same sign, σ (Eq. (20)), as the spring force $-\mathbf{V}'(\mathbf{q}_{i+1})$. If the time evolution map does not satisfy the equation of motion because only subdistributivity holds, then Φ_* pulls $d_i \mathcal{L}_i$ back to a subset of $-d_i \mathcal{L}_{i-1}$.

This shows that the imprecision in the input data together with subdistributivity may lead to a time evolution map that does not satisfy the equation of motion, which is nevertheless satisfied when no imprecision exists. In other terms, the time evolution map may not satisfy the equation

of motion in \mathcal{D} even if it is always satisfied for every ring k . It may also happen that the time evolution map ceases to satisfy the equation of motion after a finite number of time steps: this occurs, for example, in Example 1b for $i = 13$. Let us analyze this crucial point in more detail.

What is happening is overestimation caused by the multi-incidence of some variables in the expressions to be evaluated: a well known problem in interval analysis. Since the evolution map is always defined in terms of the previous two time steps (e.g., Eq. (29)), multi-incidence occurs in the computation of the flow as well. The dependency problem in range computation over a domain of proper intervals is eliminated using the algorithm developed by Popova (2005), which applies to rational functions such as those arising here by working on a polynomial configuration space. Within the Mathematica environment, this is efficiently accomplished by transforming the function to be evaluated using the `IntervalComputations`Range` package (2005), which takes into account the function's monotonicity properties in each incidence.

In the Example above, the described algorithm leads $Dual(\Phi_*d_1L_1) + d_1L_0$ to be identically equal to zero. As for the flow, steps 3 to 15 of the flow for Example 2b are given below as a way to exemplify:

<u>97840</u>	<u>109989</u>
<u>11</u>	<u>10</u>
<u>23547324</u>	<u>1099670011</u>
<u>3025</u>	<u>100000</u>
<u>27741730507</u>	<u>10993400549989</u>
<u>4159375</u>	<u>1000000000</u>
<u>126999172635479</u>	<u>109890016499230011</u>
<u>22876562500</u>	<u>10000000000000</u>
<u>557422553635612563</u>	<u>1098350384969200989989</u>
<u>125821093750000</u>	<u>100000000000000000</u>
<u>2288250850107591605311</u>	<u>10976907699076049498790011</u>
<u>692016015625000000</u>	<u>10000000000000000000</u>
<u>8301862351155904852855067</u>	<u>109692138576901814927401429989</u>
<u>3806088085937500000000</u>	<u>100000000000000000000000</u>
<u>22075992059906839606189421799</u>	<u>1096042309491854446854100098350011</u>
<u>20933484472656250000000000</u>	<u>10000000000000000000000000</u>
<u>8361651439670606649600705226397</u>	<u>10950536289837415589895004868001869989</u>
<u>115134164599609375000000000000</u>	<u>100000000000000000000000000000</u>
<u>759751840694239559412888964794437809</u>	<u>109395544311273029696900192520168297910011</u>
<u>63323790529785156250000000000000</u>	<u>1000000000000000000000000000000000</u>
<u>81020510336064516583713118612745923335573</u>	<u>1092747861697407761918806463410658791002309989</u>
<u>348280847913818359375000000000000000</u>	<u>10000000000000000000000000000000000000</u>
<u>66115762062600742198853274199335175199046281</u>	<u>10914310054959154860924191209732935370254097470011</u>
<u>191554466352600097656250000000000000000</u>	<u>1000000000000000000000000000000000000000</u>
<u>481987991561468735303467618334524740762828702157</u>	<u>109000500619387361871742253662383095369480696402749989</u>
<u>1053549564939300537109375000000000000000000</u>	<u>1000</u>

These ranges exactly correspond to those computed using the Mathematica global optimization functions `Maximize` and `Minimize`, thus confirming that $fR(\mathbf{x}) = R_f(\mathbf{x})$. However, when using Mathematica global optimization functions, computational times are over

$13 \cdot 10^3$ times higher, and they could be impractically higher for more complex problems. As a way to interpret these results, recall that the time evolution map (Eq. (16)) is continuous in q_1 , q_0 , m and s . As a consequence, the meaning of interval \mathbf{q}_i calculated at the i -th step is as follows: the actual position of the particle at the i -th time step is a rational number in the interval \mathbf{q}_i . In a non-degenerate interval \mathbf{q}_i there are infinite (albeit countable) possible positions.

Similarly to the example in Section 2, the numbers of digits in the numerator and denominator increase at each time step. As time steps proceed, the shift to the left of the upper bound (slowest possible particle) is much smaller (1%) than the shift to the left undergone by the left bound (146%), which becomes negative at the 11th time step (fastest possible particle). At the 15th time step, the width of the position interval is equal to about 15,475, whereas at the 1st time step it was equal to 1,000. Thus, the width has increased by about 150%. Figure 4 illustrates this behavior using some snapshots of the evolution of the configuration space. Notice that when the fastest particle bounces back to the right after the 80th step, the lower bound remains constant. The upper bound keeps decreasing because the slowest particle keeps marching to the left, until the fastest particle (which is now marching to the right) overcomes the slowest particle between $i = 140$ and $i = 160$, and makes the upper bound increase again.

After the 160th time step, the interval never decreases because the fastest possible particle is always “much faster” than the slower possible particle. Similarly to the precise case of Section 2, it is impossible to determine the asymptotic values for the smallest and largest coordinates reached by the particle. Despite the fact that it is unknown whether the particle will actually visit all rational coordinate positions between the reached extremes (see Section 2), the continuity of the evolution map ensures that the particle may occupy *any* of the rational coordinate positions in the intervals depicted in Figure 4.



Figure 4. Snapshots of the evolution of the configuration interval for the harmonic oscillator in Example 2a.

When the function's monotonicity properties cannot be exploited because the hypotheses in (Popova, 2005) are not fulfilled, validated bounds on the system's evolutions can be calculated by a discrete version of Taylor models. Taylor models have proven very effective in reducing overestimation in validated calculations of the flow for continuous systems (e.g., Makino, K. and Berz, 2004 and Berz and Makino, 1998 and references therein). The extension of Taylor models to discrete mechanics is the subject of current study.

5. Conclusions

Discrete dynamical systems that take values in a ring k allow for an exact integration of their time evolution. When the ring k is the ring of rational numbers, it may be impossible to determine *a priori* the evolution of even the simpler linear systems.

When discrete dynamical systems take values in \mathcal{D} (the set of extended intervals defined on a ring k), one finds that:

- The definition of the Euler-Lagrange equation and of satisfaction of the equation of motion for the time evolution map must be modified by introducing the *Dual* operator for extended intervals.
- When monotonicity can be exploited, exact bounds on the system evolution can be calculated very efficiently at each time step without the use of global optimization. When monotonicity cannot be exploited, validated bounds can be calculated, but more research is needed in this field, where Taylor models look very promising.

Two interpretations of directed intervals have been used, namely directed interval as an ordered couple of elements of k and as a set of elements of k . The interpretation of directed intervals in terms of modal logic (Gardenes, 1986) opens the way to logical interpretations of mechanical systems (including cellular automata) and vice versa; this aspect will be investigated in the future.

Acknowledgements

The author is indebted to Evgenija D. Popova (Institute of Mathematics and Informatics, Bulgarian Academy of Sciences) for sharing her packages while providing fruitful discussion, and for providing reference (Gardenes *et al.*, 2001).

6. References

Baez, J.C. and Gilliam, J.W., An algebraic approach to discrete mechanics, *Letters in Math. Phys.* 31, 205-212 (1994).

- Berz, M. Makino, K. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models, *Reliable Computing*, 4, 361-369 (1998). (<http://www.bt.pa.msu.edu/cgi-bin/display.pl?name=rdaint>).
- Computer Methods in Applied Mechanics and Engineering*, 195, Issues 4-6, 205-480 (2006). Special issue on "Adaptive Modeling and Simulation".
- Dimitrova, N., Markov, S. M. and Popova, E., Extended interval arithmetics, new results and applications, in L. Atanassova and J. Herzberger (eds.), *Computer Arithmetic and Enclosure Methods*, Elsevier Sci. Publishers B. V., 1992, pp. 225-232.
- Dow, J.O., *A Unified Approach to the Finite Element Method and Error Analysis Procedures*. Academic Press, New York (1998).
- Gardenes, E. and Trepát, A., Fundamentals of SIGLA, an interval computing system over the completed set of intervals, *Computing*, 24, 161-179 (1980).
- Gardenes, E., Mielgo, H. and Trepát, A., Modal intervals, reason and ground semantics, in K. Nickel (ed.), *Interval Mathematics 1985*, Lecture Notes in Computer Science, Vol. 212, Springer-Verlag, Berlin, Heidelberg, 1986, pp. 27-35.
- Gardenes, E., Sainz, M.A., Jorba, L., Calm, R., Estela, R., Mielgo, H., and Trepát, A., Modal Intervals. *Reliable Computing* 7, 77-111 (2001).
- Gardenes, E., Trepát, A. and Janer, J. M., Approaches to simulation and to the linear problem in the SIGLA system. *Freiburger Interval-Berichte* 81/8, 1-28 (1981).
- Gardenes, E., Trepát, A. and Janer, J. M., SIGLA-PL/1 development and applications, in Nickel, K. (ed.), *Interval Mathematics 1980*, Academic Press, pp. 301-315 (1980).
- Gardenes, E., Trepát, A., and Mielgo H., Present perspective of the SIGLA interval system, *Freiburger Interval-Berichte* 82/9, 1-65 (1982).
- Gilliam, J.W., *Lagrangian and Symplectic Techniques in Discrete Mechanics*. Ph.D. Dissertation, Mathematics, University of California at Riverside (1996).
- Kaucher, E., Algebraische Erweiterungen der Intervallrechnung unter Erhaltung der Ordnungs und Verbandstrukturen, *Computing Suppl.* 1, 65-79 (1977).
- Kaucher, E., Interval analysis in the extended interval space IR, *Computing Suppl.* 2, 33-49 (1980).
- Kaucher, E., Ueber Eigenschaften und en der Anwendungsmoeglichkeiten der erweiterten Intervallrechnung und des hyperbolischen Fastkoerpers ueber \mathbb{R} , *Computing, Suppl.* 1, 81-94 (1977).
- Kaucher, E., *Ueber metrische und algebraische Eigenschaften einiger beim numerischen Rechnen auftretender Raume*. Ph.D. dissertation, University of Karlsruhe (1973).
- Lang, S., *Algebra*, Springer-Verlag, New York, 2002.
- Makino, K. and Berz, M. *Suppression of the Wrapping Effect by Taylor Model-Based Validated Integrators*. MSU Report MSUHEP 40910, 2004. (<http://www.bt.pa.msu.edu/cgi-bin/display.pl?name=VIRC03>)
- Markov, S. M., Extended interval arithmetic involving infinite intervals, *Mathematica Balkanica*, New Series, 6, 269-304 (1992).

- Markov, S. M., Isomorphic embedding of abstract interval systems, *Reliable Computing* 3(3), 199-207 (1997).
- Markov, S. M., On directed interval arithmetic and its applications, *J. UCS*, 1, 510-521 (1995).
- Marsden, J. E. and M. West, Discrete mechanics and variational integrators, *Acta Numerica*, 357-514 (2001).
- Moore, R. E., *Interval Analysis*, by, Prentice-Hall, Englewood Cliffs, NJ, 1966.
- Oberkampf, W.L., Trucano, T.G. and Hirsch, C. (2003), *Verification, Validation, and Predictive Capability in Computational Engineering and Physics*. Sandia National Laboratories Report SAND2003-3769. Albuquerque, NM.
- Oden, J. T., I. Babuska, F. Nobile, Y. Feng, R. Tempone, (2005), Theory and methodology for estimation and control of errors due to modeling, approximation, and uncertainty. *Comput. Methods Appl. Engrg.*, 194, 195-204.
- Ortolf, H. -J., Eine Verallgemeinerung der Intervallarimetik, *Berichte der Gesellschaft für Mathematik und Datenverarbeitung*, Bonn 11, 1-71 (1969).
- Popova, E. D. and Ullrich, C., *Directed Interval Arithmetic in Mathematica, Implementation and Applications*. Technical Report 96-3, University of Basel (1996).
- Popova, E. D. and Ullrich, C., Simplification of symbolic-numerical interval expressions, in O. Gloor (ed.), *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*, ACM Press, 1998, 207-214.
- Popova, E. D., Extended interval arithmetic in IEEE floating-point environment, *Interval Computations*, 4, 100-129 (1994).
- Popova, E.D., *All About Generalized Interval Distributive Relations*, Manuscript, 2000. Available at <http://www.math.bas.bg/~epopova/papers>.
- Popova, E.D., Multiplication distributivity of proper and improper intervals, *Reliable Computing*, 7, 129-140 (2001).
- Popova, E.D., *Solving Linear Systems whose Input Data are Rational Functions of Interval Parameters*. Preprint No 3, Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, December 2005 (2005).
- Ratschek, H., Die binären systeme det intervallmatematik, *Computing* 6, 295-308 (1970).
- Ratschek, H., Die subdistributivität der intervallarimetik, *Z. Angew.Math. Mech.* 51, 189-192 (1971).
- Sauer-Budge, A.M., Bonet, J., Huerta, A., and Peraire, J., Computing bounds for linear functionals of exact weak solutions to Poisson's equation, *SIAM Journal on Numerical Analysis*, 42, 4 1610-1630, (2004).
- Spaniol, O., Die subdistributivität in der intervallarimetik, *Computing* 5, 6-16 (1970).
- Talasila, V., Clemente Gallardo, J., and van der Schaft, A.J., Geometry and Hamiltonian mechanics on discrete spaces, *J. Phys. A: Math. Gen.*, 37, 9705-9734 (2004b).
- Talasila, V., Clemente Gallardo, J., and van der Schaft, A.J., *Hamiltonian Mechanics on Discrete Manifolds*. Technical Report, Department of Applied Mathematics, University of Twente, The Netherlands (2004a).

- Toffoli, T. and Margolus, N., *Cellular Automata Machines: a New Environment for Modeling*. MIT Press, Cambridge, Mass. (1987).
- Toffoli, T., Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics. *Physica 10D*, 117-127 (1984).
- von Neumann, J., The general and logical theory of automata. In (L.A. Jeffress ed.) *Cerebral Mechanisms in Behavior: The Hixon Symposium*. 1-32. Wiley (1951).
- Warmus, M., Calculus of Approximations. *Bull. Acad. Pol. Sci.*, Cl. III, 4. 253-259 (1956).
- Wasserman, R.H. *Tensors and Manifolds – With Applications to Mechanics and Relativity*. Oxford University Press, New York (1992).
- Wolfram, S., Cryptography with cellular automata. In Proc. *CRYPTO '85: Advances in Cryptology*. Santa Barbara, California (1985b).
- Wolfram, S., Origin of randomness in physical systems. *Phy. Rev. Lett.*, 55, 449-452 (1985a).
- www.cs.utep.edu/interval-comp/main.html
- Xuan, Z.C., Parés, N. and Peraire, J., Computing upper and lower bounds for the J-integral in two-dimensional linear elasticity. *Computer Meth. in Appl. Mech. and Engng.*, 195, 430-443 (2006).

Checking Computation of Numerical Functions by the Use of Functional Equations

F. Vainstein and C. Jones

Georgia Institute of Technology
feodor.vainstein@gtsav.gatech.edu

Abstract: Systematic use of functional equations for fault-tolerant computation of numerical functions have been introduced by M. Blum (1989) and then independently by F. Vainstein (1991). The later has introduced definition of polynomially checkable (PC) functions – the functions for which functional equations are polynomials, and proved that the class of PC functions is large and includes many commonly used functions. The functional equations that are used to check computations of numerical functions are called checking polynomials. In this paper we discuss an algorithm for computing coefficients of these polynomials. By using this algorithm we obtain checking polynomials for the commonly used functions.

Keywords: fault tolerance, algebraic methods, numerical functions, error checking, checking polynomials.

1. Introduction

Computers continue to take on more mission and safety critical operations in industrial, scientific, and consumer markets. Modern processors compute a wide range of numerical functions. Detecting and correcting errors due to numerical computations are critical aspects of processor design.

There have been numerous approaches to fault tolerant computation of numerical functions. These include hardware, information, time, and software redundancy methods (Lala 2001). However, each of these methods comes at a significant price to the system in space or time. And while the dimensions of chip technology are continually reduced the complexity of the systems placed on chips continues to rise.

The technique described here employs the algebraic concept of the transcendental degree of field extensions to exploit the structure of a specific numerical computation. This method requires significantly less hardware redundancy, offers good fault coverage, and has significant fault location capability (Vainstein 1993).

Algorithms used in numerical computations can be sophisticated and numerous implementations exist (Koren 2001; Muller 1997; Ercegovac, Lang et al 2000). Many considerations go into choosing a certain numerical algorithm based on specific application and design criteria. Contrary to the perception of many, the computation of numerical functions can be quite complex and susceptible to faults.

In order to give the flavor of an algorithm for computing an elementary function, consider Wong and Goto's algorithm for computing logarithms (Wong and Goto 1994). This description is based on the presentation of this algorithm given by Muller in (Muller 1997). See Muller's text for a complete description of the assumptions, details, and technical issues of the algorithm.

The notation

$$[z]_{a-b}$$

is the number obtained by zeroing all the bits of z but the bits a to b . For example, if $m = m_0.m_1m_2m_3m_4\dots$, then

$$[m]_{1-3} = 0.m_1m_2m_3000\dots$$

To compute the logarithm of a normalized IEEE-754 double precision floating-point number

$$x = m \times 2^{\text{exponent}}$$

We have to follow the steps below:

1. Obtain factor K_1 and $\ln(K_1)$ from tables.
2. Use a rectangular multiplier to multiply m by K_1 . Then K_2 is chosen such that K_1K_2m is close to 1. And $[\ln(K_2)]_{1-56}$ is obtained from tables.
3. Use a rectangular multiplier to multiply (mK_1) by K_2 . As in the previous step $[\ln(K_3)]_{1-56}$ is obtained from tables.
4. Use a rectangular multiplier to multiply (mK_1K_2) by K_3 . The result is $1 - \gamma$, where $0 \leq \gamma < 2^{-24}$. This result is close enough to 1 that a degree-3 Taylor polynomial approximation will give good accuracy.
5. Then, full multiplication and tables are used to compute

$$\left[\frac{\gamma^2}{2} \right]_{1-56}$$

and

$$\left[\frac{[\gamma]_{25-33}^3}{3} \right]_{1-56}$$

6. And, finally,

$$\ln(x) \approx exponent \times \ln(2) - \ln K_1 - \ln K_2 - \ln K_3 - \gamma - \left[\frac{\gamma^2}{2} \right]_{1-56} - \left[\frac{[\gamma]_{25-33}^3}{3} \right]_{1-56}$$

As we see from this example, even “simple” numerical functions such as the logarithmic function can be quite sophisticated and thus susceptible to faults.

The method presented in this paper seeks to address the fault-tolerant needs of numerical algorithms with low processor overhead. To illustrate this method, let us consider an example.

Suppose we have to compute the function $f(x) = e^{-x} \sin 5x, x \in [0,10]$

Let $a_1, a_2 \in R$;

Denote by $f_0 = f(x+0) = e^{-x} \sin 5x$,

$$f_1 = f(x+a_1) = e^{-(x+a_1)} \sin 5(x+a_1),$$

$$f_2 = f(x+a_2) = e^{-(x+a_2)} \sin 5(x+a_2).$$

Denote by $p_1 = e^{-a_1} \cos 5a_1$; $q_1 = e^{-a_1} \sin 5a_1$; $p_2 = e^{-a_2} \cos 5a_2$; $q_2 = e^{-a_2} \sin 5a_2$;

$$A = p_1q_2 - p_2q_1; B = -q_2; C = q_1$$

Then $Af_0 + Bf_1 + Cf_2 = 0$

For every $x \in R$.

It is very important that A, B and C do not depend on x and depend only on a_1 and a_2 . Taking (1) into consideration we can consider the following method for error detection.

Denote the computed values of function f at the points $x, x+a_1, x+a_2$ by $\tilde{f}_0, \tilde{f}_1, \tilde{f}_2$ respectively. Then if the computation is correct

$$A\tilde{f}_0 + B\tilde{f}_1 + C\tilde{f}_2 = 0 \quad (\text{Independently of } x!) \tag{2}$$

For error correction (single error in this case) we can proceed as follows.

$$\text{Consider } a_1 = a; a_2 = 2a \text{ and let } A\tilde{f}_0 + B\tilde{f}_1 + C\tilde{f}_2 \neq 0 \tag{3}$$

Because one of $\tilde{f}_i \neq f_i; i = 0, 1, 2$.

Suppose for example that $\tilde{f}_0 \neq f_0; \tilde{f}_1 = f_1; \tilde{f}_2 = f_2$

Then the correct value is given by the formula

$$f_0 = -\frac{B}{A}\tilde{f}_1 - \frac{C}{A}\tilde{f}_2 \tag{4}$$

Location of the error can be obtained by using (2) for the following triples:

$$x-2a \quad x-a \quad x \quad x+a \quad x+2a$$



It should be taken into consideration that computations are done in practice with a certain level of accuracy. Hence the formula 2 should be substituted by the formula

$$| Af_0 + Bf_1 + Cf_2 | \leq \delta, \tag{2'}$$

where δ is a small positive number specified by the precision of the computation.

2. Polynomial checking

For the readers convenience let us present the following definitions and results from the field extension theory (Lang 1992).

Definition 1. Let $K \subset L$ be a field extension and $K[T_1, \dots, T_n]$ be the set of all polynomials in T_1, \dots, T_n over K . The elements $a_1, \dots, a_n \in L$ are called algebraically dependent over K , if there exists a polynomial $P \in K[T_1, \dots, T_n], P \neq 0$, such that $P(a_1, \dots, a_n) = 0$. The elements $a_1, \dots, a_n \in L$ are called algebraically independent over K , if they are not algebraically dependent. By $K(T_1, \dots, T_n)$ we denote the quotient field of the ring $K[T_1, \dots, T_n]$.

Example 2. Consider the field extension $Q \subset R$. Then the numbers $\sqrt{2}$ and $\sqrt{3} \in R$ are algebraically dependent over Q . $P(T_1, T_2) = T_1^2 + T_2^2 - 5$. The numbers $1, \pi \in R$ are algebraically independent over Q .

Definition 2. Let $K \subset L$ be a field extension. Transcendental degree (Tr.deg.) of this extension is by definition the maximum possible number of elements from L algebraically independent over K .

If Tr.deg. of $K \subset L$ is equal to n and $m > n$, the any subset $\{ a_1, \dots, a_m \} \subset L$ is algebraically dependent.

Example 3. Tr.deg. of $R \subset R(T)$ equals to 1.

Tr.deg. of $R \subset R(x, e^x)$ equals to 2.

Tr.deg. of $R(x, \sin x, e^x) \subset R(x, \sin x, \cos x, e^x)$ equals to 0.

Definition 3. A field L is called algebraically closed if any polynomial $P \in L[T]$ has a root in L .

Example 4. R is not algebraically closed. $P(T) = T^2 + 1$ does not have roots in R . C is algebraically closed.

Definition 4. A field \overline{K} is called an algebraic closure of field K is

1. \overline{K} is algebraically closed.
2. Tr.deg. of $K \subset \overline{K}$ is equal to 0.

Theorem 1. For any field K its algebraic closure \overline{K} exists and is unique up to isomorphism.

Definition 5. A function $f:R \rightarrow R$ is called polynomially checkable (PC) if there exists an integer k , such that for any $a_1, \dots, a_k \in R$ the functions $f_0(x) = f(x), f_1(x) = f(x + a_1), \dots, f_k(x) = f(x + a_k)$ are algebraically dependent, i.e. there exists a polynomial $P \in R[T_0, \dots, T_k]$, such that $P(f_0, \dots, f_k) = 0$ (for any $x \in R$). The polynomial P is called a *checking polynomial* of the function f .

The computation of a PC function can be readily verified. For a given value of x , denote by $\tilde{f}_0, \tilde{f}_1, \dots, \tilde{f}_k$ the values of f at the points $x, x + a_1, \dots, x + a_k$, respectively. Then if all the values are computed correctly, the following equality holds:

$$P(\tilde{f}_0, \tilde{f}_1, \dots, \tilde{f}_k) = 0 \tag{5}$$

This property provides a unified approach to the problem of error detection/correction in computation of numerical functions. Indeed we can consider inequality similar to (2')

$$\left| P(\tilde{f}_0, \tilde{f}_1, \dots, \tilde{f}_k) \right| \leq \delta, \tag{5'}$$

where δ is a small positive number specified by the precision of the computation. In case of correct computation (5') is satisfied. We have to note, however, that even if (5') is satisfied it doesn't give us 100% warrantee that computation is correct. There are some faults that cannot be detected by (5').

The first class of faults (we can call them software faults) are result of the fact that some other PC function $g(x) \neq f(x)$ can have the same checking polynomial. For instance if $g(x) = f(x + b)$, where b is a constant, then $g(x)$ and $f(x)$ have the same checking polynomials. Preliminary results show that a PC function with bounded spectrum is uniquely defined by its checking polynomial (the set of shifts is fixed) and its values at a finite set of points. This property can be used to fight the software faults.

The second class of faults which can not be detected by using (5') are hardware faults. They are result of physical defects of a device which performs the calculation of function. Random faults are hardware faults. The fault coverage of random faults is calculated below for an important case. It is shown in (Abamowitz and Stegun 1965) that the class of PC functions is very broad even for a small k .

Denote by S the set of three functions: $x, e^x, \sin x$. Let $A \subseteq S$; denote by $R(A)$ the field of all rational functions in $g_j \in A$ and by $\overline{R(A)}$ its algebraic closure.

Example 5. a) $A = (x)$, $R(A) = \left\{ \frac{P_i(x)}{Q_i(x)} \right\}$, where P_i, Q_j are polynomials of one variable with real coefficients. Its algebraic closure $\overline{R(A)}$ includes, as a special case any function $g(x)$ which is a solution of an equation $P_n(x)g^n(x) + P_{n-1}(x)g^{n-1}(x) + \dots + P_0(x) = 0$ where $P_i(x), i = 0, 1, \dots, n$ are polynomials of one variable with real coefficients.

In particular, $\overline{R(A)}$ includes the set of all functions that can be obtained by application of finite number of additions, subtractions, multiplications, divisions, and raising to a rational power to the function $g(x) = x$.

b) $A = \{e^x, \sin x\}$; $R(A) = \frac{P_i(e^x, \sin x)}{Q_i(e^x, \sin x)}$, where P_i, Q_j are polynomials of two variables with real coefficients.

Theorem 2 Let $f: R \rightarrow R$ belong to the field $\overline{R(A)}, A \subseteq (x, e^x, \sin x)$. Then f is polynomially checkable with $k = |A|$.

Proof. We prove the theorem for the case $A = \{x, e^x, \sin x\}$. For the other cases the proof is analogous.

Let $f(x) \in \overline{R(x, e^x, \sin x)}$ and $a_1, a_2, a_3 \in R$; denote: $f_0(x) = f(x)$, $f_1(x) = f(x + a_1)$, $f_2(x) = f(x + a_2)$, $f_3(x) = f(x + a_3)$. We have to show that f_0, \dots, f_3 are algebraically dependent. This follows from the statements:

- 1) Tr.deg of $R \subset \overline{R(x, e^x, \sin x)}$ equals to 3.
- 2) For every $a \in R, f(x + a) \in \overline{R(x, e^x, \sin x)}$.

Indeed $f(x) \in \overline{R(x, e^x, \sin x)} \Leftrightarrow$ there exists a polynomial $A \in R(x, e^x, \sin x)[T]$ such that $A(f) = A_n(x)f(x) + \dots + A_1(x)f(x) + A_0(x) = 0$, where $A_i(x) \in R(x, e^x, \sin x)$. Let us denote $\rho(x) = A_n(x)f(x) + \dots + A_0(x)$. Then

$$\rho(x + a) = A_n(x + a)f(x + a) + \dots + A_0(x + a) = 0, A_i(x + a) \in \overline{R(x, e^x, \sin x, \cos x)}.$$

Hence $f(x + a) \in \overline{R(x, e^x, \sin x, \cos x)}$. But $\overline{R(x, e^x, \sin x, \cos x)} = \overline{R(x, e^x, \sin x)}$,

hence $f(x + a) \in \overline{R(x, e^x, \sin x)}$ and, therefore, $f_0, f_1, f_2, f_3 \in \overline{R(x, e^x, \sin x)}$.

But the Tr.deg. of $R \subset \overline{R(x, e^x, \sin x)}$ equals to 3, hence f_0, f_1, f_2, f_3 are algebraically dependent.

Let f be the result of application of a finite number of additions, subtractions, multiplications, divisions and raising to a rational power to the following functions:

$Const, x, e^x, \sin(r_i x + b_i), \cos(r_j x + b_j)$, where r_i, r_j are rational numbers.

Then f is a PC function with $k \leq 3$.

Example 6. The function $f(x) = \frac{(\sin(\frac{x}{11} + \frac{\pi}{7}) + e^x)^{\frac{3}{5}} + x^2 \cos^4 x}{x^5 + (x^4 + x^2(\sin 2x + xe^x)^3)^{\frac{1}{3}}}$ is a PC function with $k=3$.

Example 7. Consider the function

$$f(x) = \frac{((\sin x)^{\frac{1}{3}} + \cos x)^{\frac{1}{2}}}{\sin(x + 7) - (\cos 3x \sin(3x + 5) - 4)^{\frac{1}{17}}}; f(x) \in \overline{R(\sin x)}$$

Tr.deg. of extension $R \subset \overline{R(\sin x)}$ equals to 1, therefore $f(x)$ is a PC function with $k=1$.

Note. The theorem 2 states that the class of PC functions is very big. We have to note, however, that a number of commonly used functions like $\ln(x), \sin^{-1}(x), \cos^{-1}(x)$ are non PC functions.

3. Finding a Checking Polynomial by Least Square Estimation

To find a checking polynomial we consider the following optimization problem. Let

$$f : [A, B] \rightarrow R$$

Denote

$$\delta(\beta_0, \alpha_1, \dots, \alpha_k) = \int_A^B 2(f(x) - \alpha_1 f(x + a_1) - \dots - \alpha_k f(x + a_k) - \beta_0)^2 dx$$

Find such $\alpha_1, \dots, \alpha_k, \beta_0$, that $\delta(\beta_0, \alpha_1, \dots, \alpha_k)$ takes minimal value. To solve this problem consider the following equations:

$$\begin{aligned} \frac{\partial}{\partial \beta_0} \delta &= \int_A^B 2(f(x) - \alpha_1 f_1 - \dots - \alpha_k f_k - \beta_0)(-1) dx = 0 \\ \frac{\partial}{\partial \alpha_1} \delta &= \int_A^B 2(f(x) - \alpha_1 f_1 - \dots - \alpha_k f_k - \beta_0) f_1 dx = 0 \\ &\vdots \\ \frac{\partial}{\partial \alpha_k} \delta &= \int_A^B 2(f(x) - \alpha_1 f_1 - \dots - \alpha_k f_k - \beta_0) f_k dx = 0 \end{aligned}$$

Let us denote by $\langle f, g \rangle = \int_A^B (f \cdot g) dx$. Using this notation, we can express the system of equations in the form

$$\begin{aligned} \langle f_0, 1 \rangle &= \alpha_1 \langle 1, f_1 \rangle + \dots + \alpha_k \langle 1, f_k \rangle + \beta_0 (B - A) \\ \langle f_0, f_1 \rangle &= \alpha_1 \langle f_1, f_1 \rangle + \dots + \alpha_k \langle f_1, f_k \rangle + \beta_0 \langle f_1, 1 \rangle \\ \langle f_0, f_2 \rangle &= \alpha_1 \langle f_2, f_1 \rangle + \dots + \alpha_k \langle f_2, f_k \rangle + \beta_0 \langle f_2, 1 \rangle \\ &\vdots \\ \langle f_0, f_k \rangle &= \alpha_1 \langle f_k, f_1 \rangle + \dots + \alpha_k \langle f_k, f_k \rangle + \beta_0 \langle f_k, 1 \rangle \end{aligned}$$

Solving this system we obtain $\beta_0, \alpha_1, \dots, \alpha_k$. If $\delta(\beta_0, \alpha_1, \dots, \alpha_k) = 0$ then f is an LC function with the checking polynomial

$$f_0 - \alpha_1 f_1 - \dots - \alpha_k f_k - \beta_0 = 0$$

If $\delta(\beta_0, \alpha_1, \dots, \alpha_k) = \delta \neq 0$ then f does not have a checking polynomial of degree 1. However, if δ is a small number the formula

$$\left| \tilde{f}_0 - \alpha_1 \tilde{f}_1 - \dots - \alpha_k \tilde{f}_k - \beta \right| \leq \delta$$

can be used to verify the correctness of computations. A similar method can be used for obtaining a checking polynomial of degree > 1 .

Other methods for finding a checking polynomial are described in (Vainstein 1998).

4. Numerical Results

A Matlab program was developed to implement the algorithm described using techniques from linear algebra. This program can determine the coefficients of the checking polynomial, $\beta_0, \alpha_1, \dots, \alpha_k$, for various numerical functions.

From the system of equations defined above, denote

$$A = \begin{pmatrix} \langle 1, f_1 \rangle & \dots & \langle 1, f_k \rangle & (B - A) \\ \langle f_1, f_1 \rangle & \dots & \langle f_1, f_k \rangle & \langle f_1, 1 \rangle \\ \langle f_2, f_1 \rangle & \dots & \langle f_2, f_k \rangle & \langle f_2, 1 \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle f_k, f_1 \rangle & \dots & \langle f_k, f_k \rangle & \langle f_k, 1 \rangle \end{pmatrix}.$$

Define vector X as

$$X = \begin{pmatrix} \beta_0 \\ \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix}.$$

And, define vector B as

$$B = \begin{pmatrix} \langle f_0, 1 \rangle \\ \langle f_0, f_1 \rangle \\ \langle f_0, f_2 \rangle \\ \vdots \\ \langle f_0, f_k \rangle \end{pmatrix}.$$

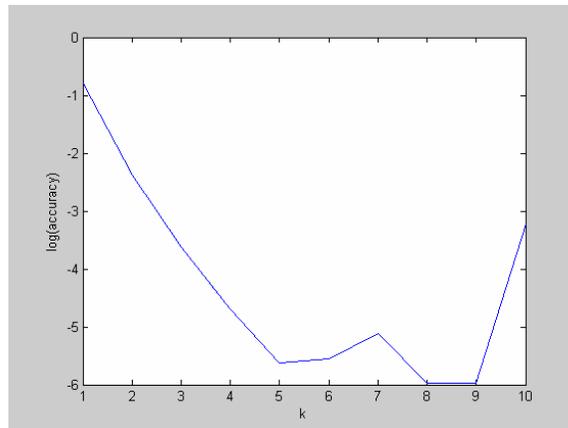
First, the program computes the coefficients of matrix A and vector B using the trapezoidal integration method. Then, the equation $AX = B$ is solved by a reduced row echelon form method. The resulting values of vector X then give us the coefficients of the checking polynomial, $\beta_0, \alpha_1, \dots, \alpha_k$.

The values of $\beta_0, \alpha_1, \dots, \alpha_k$ are then used to evaluate $\delta(\beta_0, \alpha_1, \dots, \alpha_k)$ as described above. This gives the value of δ .

In order to investigate the effect of increasing k , the program finds the values of δ for a given range of k values and determines which of these produces the minimum deviation. Shown below are the results from these computations for various numerical functions.

In Table 1 and Figure 1 we see the algorithm's results when applied to $f(x) = \ln(x)$.

Function	$\ln(x)$
Best k	9
Accuracy, δ	$1.063 * 10^{-6}$
α_1	13.636083507129
α_2	-48.3372588311351
α_3	33.8356474758064
α_4	104.402870905685
α_5	-173.871357832086
α_6	19.6783752309016
α_7	95.3406000928846
α_8	-40.7070503540183
α_9	-2.97970571433605
β_o	0.0104815654867488
Stepsize, h	0.0001
Lower limit, A	1
Upper limit, B	100

Table 1. Best results for $f(x) = \ln(x)$.Figure 1. Graph of the accuracy δ versus k for $f(x) = \ln(x)$.

These results show that, in the interval $[1,100]$, and with a step size of $h = 0.0001$, the checking polynomial of $f(x) = \ln(x)$ that returns the best accuracy, or minimum deviation δ , is the polynomial with values for $\beta_0, \alpha_1, \dots, \alpha_k$ as given in Table 1.

As a result of the computational experiments we observed that, as a rule, the deviation, δ , is decreasing with increasing k , for small values of k . However, as k continues to increase δ eventually begins to increase. The reason for this increase is the limited accuracy of computer arithmetic. In general, we are interested in the smallest value of k (since the overhead increases with k) that provides us with a satisfactory deviation.

As another example, let's consider a more complicated looking numerical function

$$f(x) = \cos(\sin(x) - \sqrt{x}) - \cos(x) + \sqrt{x} \tag{6}$$

The graph δ versus k for this function is shown below in Figure 3.

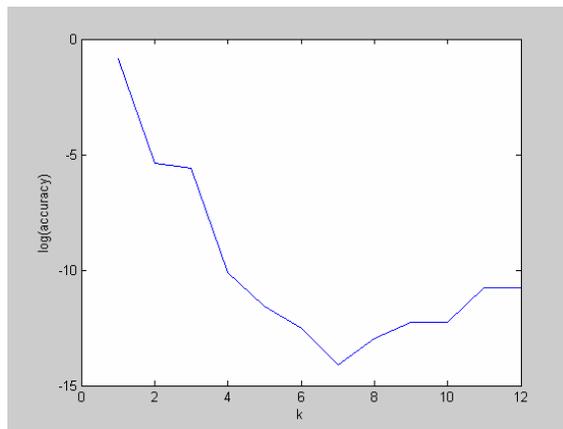


Figure 3. Graph of the accuracy δ versus k for $f(x) = \cos(\sin(x) - \sqrt{x}) - \cos(x) + \sqrt{x}$.

We note in the results of Figure 3 the two important features mentioned above: the deviation initially decreases but then eventually increases with increasing k .

Given in the table below are some other sample results.

Function	k	δ	$\alpha_1, \alpha_2, \dots, \alpha_k$	β
$\cos(x)$	2	$1.5655594 * 10^{-26}$	1.08060461 -0.999999999	- $4.6509012 * 10^{-15}$
x^2	2	$5.3305333 * 10^{-20}$	2.00000000 -1.00000000	2.00000000
\sqrt{x}	6	$3.5165389 * 10^{-9}$	17.6933009 -82.75929	-0.73834646

			138.617962 -51.0106681 -69.7915475 48.319250	
e^x	2	$5.4703917 * 10^{-22}$	0.00810684622 0.132352941	$-5.80812 * 10^{-12}$
$\log \left[x + (x^2 + 1)^{\frac{1}{2}} \right]$	3	$1.8593950 * 10^{-10}$	1.67126425 3.81365972 -5.13358327	2.2870721
$\cos(\sin(x) - \sqrt{x}) - \cos(x) + \sqrt{x}$	7	$7.6541494 * 10^{-15}$	-1.16902598 -0.58592973 -0.74339752 -1.48341738 -0.454274819 -0.230924645 0.684256990	7.6417875

Table 3. Sample results of the least square estimation method. All results for step size, $h = 0.0001$

These results in Table 3 give the values of $\beta_0, \alpha_1, \dots, \alpha_k$ that define the checking polynomial of the form

$$f_0 - \alpha_1 f_1 - \dots - \alpha_k f_k - \beta_0 = 0$$

for each numerical function.

A number of other common and specialized numerical functions were also tested. The results are shown in Table 4 below. Each of these functions were tested over a domain interval for which they are well behaved.

Function	k	δ
Airy Function: $A_i(z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\left(zt + \frac{t^3}{3}\right)} dt$; for $i = 1$	6	$8.01855604514853 * 10^{-13}$
Bessel Function of 1 st Kind: $J_1(z) = \sum \frac{(-1)^k}{\Gamma(k + \nu + 1)k!} \left(\frac{z}{2}\right)^{2k + \nu}$	7	$6.77944398369068 * 10^{-12}$
Beta Function: $B(a, a) = \int_0^1 t^{a-1} (1-t)^{a-1} dt = \frac{\Gamma(a)\Gamma(a)}{\Gamma(a+a)}$	5	$1.94247036365353 * 10^{-9}$
Scaled Complementary Error Function: $f(x) = e^{x^2} \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt$	6	$5.53736923276668 * 10^{-11}$

Exponential Integral: $f(x) = \int_x^\infty \frac{e^{-t}}{t} dt$	5	$1.09725876920181 * 10^{-10}$
Logarithm of the Gamma Function: $\log \Gamma(z) = \sum_{k=1}^\infty \left(\frac{z}{k} - \log \left(1 - \frac{z}{k} \right) \right) - \gamma z - \log(z)$	5	$2.24588944104907 * 10^{-7}$
Inverse Cosine: $\cos^{-1}(z) = \frac{\pi}{2} + i \log \left(iz + \sqrt{1-z^2} \right)$	6	$9.57918288982022 * 10^{-6}$
Inverse Hyperbolic Cosine: $\cosh^{-1}(z) = \log \left(z + \sqrt{z-1} \sqrt{z+1} \right)$	4	$5.68478815529586 * 10^{-9}$
Hyperbolic Cosine: $\cosh(z) = \frac{e^z + e^{-z}}{2}$	2	$1.63905802795885 * 10^{-21}$
Inverse Tangent: $\tan^{-1}(z) = \frac{i}{2} \log \left(\frac{i+z}{i-z} \right)$	5	$2.2975351543044 * 10^{-6}$
Complete Elliptic Integral of the First Kind: $K(z) = \int_0^1 \frac{1}{\sqrt{1-t^2} \sqrt{1-zt^2}} dt$	4	$1.29605146674611 * 10^{-11}$
Riemann Zeta Function: $\zeta(s) = \sum_{k=1}^\infty \frac{1}{k^s}$	3	$5.31429308150972 * 10^{-9}$
Dawson's Integral: $F(x) = e^{-x^2} \int_0^x e^{-t^2} dt$	7	$1.48383522053584 * 10^{-6}$
Fresnel Sine Integral: $S(x) = \int_0^x \sin \left(\frac{\pi}{2} \cdot t^2 \right) dt$	6	$4.04417425710405 * 10^{-5}$

Table 4. Results for various functions (Abamowitz and Stegun 1965; Wolfram 1999).

5. Conclusions and Future Work

We have demonstrated that checking polynomials can be effectively used for fault tolerant computations. In particular, checking polynomials for some common numerical functions and some specialized functions have been found.

A program was developed in Matlab that allow us to obtain an “approximate” checking polynomial for a wide range of numerical functions.

The examples considered showed that even for functions that do not appear simple an approximate checking polynomial provides a small value of deviation, δ .

A future paper will describe a hardware implementation of this fault tolerance technique. Issues related to computational overhead and comparisons to overhead incurred by other methods will be discussed.

We will also consider the problem of obtaining checking polynomials of degree greater than one. Once this theoretical foundation is established an approach such as that outlined in this paper will be developed for finding coefficients of higher degree checking polynomials.

References

- Abramowitz, M. and I.A. Stegun, *Handbook of Mathematical Functions*, National Bureau of Standards, Applied Math. Series #55, Dover Publications, 1965.
- Ercegovac, Milos D., Tomás Lang, Jean-Michel Muller, and Arnaud Tisserand. "Reciprocation, Square Root, Inverse Square Root, and Some Elementary Functions Using Small Multipliers." *IEEE Transactions on Computers*, 49(7):628-637, July 2000.
- Koren, Israel. *Computer Arithmetic Algorithms*. A K Peters, 2001.
- Lala, Parag K. *Self-Checking and Fault-Tolerant Digital Design*. Morgan Kaufmann, 2001.
- Lang, S., *Algebra*, Addison-Wesley Publishing Co., 1992.
- Muller, Jean-Michel. *Elementary Functions: Algorithms and Implementation*. Birkhäuser, 1997.
- Vainstein, Feodor. "Algebraic Methods in Hardware/Software Testing." Ph. D. Thesis, Boston University, 1993.
- Vainstein, F. S. "Self Checking Design Technique for Numerical Computations." *VLSI Design*, 1998, Vol. 5, No. 4, pp. 385-392.
- Wolfram, Stephen, *The Mathematica Book*, 4th Ed., Cambridge University Press, 1999.
- Wong, W.F. and Goto, E. "Fast hardware-based algorithms for elementary function computations using rectangular multipliers." *IEEE Transactions on Computers*, 43(3):278-294, March 1994.

A library of Taylor models for PVS automatic proof checker

Francisco Cháves[‡] and Marc Daumas[§]

*Laboratoire de l'Informatique du Parallélisme
UMR 5668 CNRS-ENS de Lyon-INRIA
email: Francisco.Jose.Chaves.Alonso@ENS-Lyon.Fr*

*Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier
UMR 5506 CNRS-UM2
email: Marc.Daumas@LIRMM.Fr*

*Visiting Laboratoire de Physique Appliquée et d'Automatique
EA 3679 UPVD
email: Marc.Daumas@Univ-Perp.Fr*

Abstract. We present in this paper a library to compute with Taylor models, a technique extending interval arithmetic to reduce decorrelation and to solve differential equations. Numerical software usually produces only numerical results. Our library can be used to produce both results and proofs. As seen during the development of Fermat's last theorem reported by Aczel (1996), providing a proof is not sufficient. Our library provides a proof that has been thoroughly scrutinized by a trustworthy and tireless assistant. PVS is an automatic proof assistant that has been fairly developed and used and that has no internal connection with interval arithmetic or Taylor models. We built our library so that PVS validates each result as it is produced. As producing and validating a proof, is and will certainly remain a bigger task than just producing a numerical result our library will never be a replacement to imperative implementations of Taylor models such as Cosy Infinity. Our library should mainly be used to validate small to medium size results that are involved in safety or life critical applications.

Keywords: PVS, program verification, interval arithmetic, Taylor models.

1. Introduction

Taylor models, see for example (Makino and Berz, 2003) and references herein, have recently emerged as a nice and convenient way to reduce decorrelation in interval arithmetic (Moore,

[‡] This material is based on work supported by the *Mathlogaps* (Mathematical Logic and Applications) project, an Early Stage Research Training grant of the European Union.

[§] This work has been partially supported by PICS 2533 from the French National Center for Scientific Research (CNRS).

1966; Neumaier, 1990; Jaulin *et al.*, 2001). Taylor models are even more attractive when one solves initial value problems ODEs as they provide a validated built-in integration operator.

Yet, it is now beyond doubt that programs and libraries contain bugs, no matter how precisely they have been specified and how thoroughly they have been tested (Rushby and von Henke, 1991; Ross, 2005). As a consequence, the highest Common Criteria Evaluation Assurance Level, EAL 7¹, has only been awarded so far to products that provide validation using a formal tool, specifically an automatic proof checker in first or higher order logic.

We present here our library of Taylor models in PVS (Owre *et al.*, 1992). Working with an automatic proof checker, we had to manage two tasks. The first task was to create a data type and operations on this new type to allow users to define and evaluate expressions using Taylor models. The second task was to provide proofs that each operator is correct and a strategy to recursively analyze compound expressions. Both tasks rely on the recently published library on interval arithmetic for PVS (Daumas *et al.*, 2005). As many mathematical developments are not yet available in PVS, we also had to develop an extended library on polynomials and prove a few theorems of analysis and algebra.

Our library on Taylor models can be used to derive quickly more or less accurate bounds. For example, users of formal tools have to provide proofs that radicals are non negative for all expressions using square roots. Some proofs use intricate analysis but most of them are very simple and interval arithmetic or low degree evaluations with Taylor models can produce appropriate proofs. Our library can also be used to expertly derive computer validated proofs of difficult results through an expert use of Taylor models.

The library will be available freely on the Internet as soon as it is stable. Side developments are integrated as they are produced to NASA Langley PVS libraries². Meanwhile, all files can be retrieved from the author's website.

<http://perso.ens-lyon.fr/francisco.jose.chaves.alonso/pvs-files/>

1.1. WORKING WITH AN AUTOMATIC PROOF CHECKER

Software is used extensively for a wide array of tasks. Some pieces of software should never fail. The ones used by transportation means (planes, buses, cars...), for medical care (controlling pumps, monitors, prescriptions...) or in the army (parts of weapons, alarms...) belong to the fast lengthening list of life or safety critical applications. A mindless modification of one parameter reportedly caused human losses in the *Instituto Oncologico Nacional* on Panama where eight people died and twenty others were hurt (Gage and McCormick, 2004). Many lethal and costly failures (Information Management and Technology Division, 1992; Lions and others, 1996) show beyond reasonable doubts that traditional software verification is not sufficient to guarantee correct behavior.

PVS³ (Prototype Verification System) by Owre *et al.* (1992; 2001a; 2001b) is one environment for the development and analysis of formal specifications that allows the elaboration of theories and proofs. The system deals with theories where users develop definitions, axioms and theorems. To

¹ <http://niap.nist.gov/cc-scheme/>.

² <http://shemesh.larc.nasa.gov/fm/ftp/larc/PVS-library/pvslib.html>.

³ <http://pvs.csl.sri.com/>.

verify that theorems are correct, PVS uses a typed higher order logic language where new types are defined from a list of basic types including booleans, natural numbers, integers. . . The type system allows the definition of functions, registers, tuples and abstract data types.

PVS uses *predicate subtypes*, subtypes where all objects satisfy a given predicate. For example $\{x : \text{real} \mid x \neq 0\}$ is the set of non-zero reals. Subtype predicates are used for operations that aren't defined for all possible inputs. This restriction is therefore visible in the signature of the operation. For example the division is an operation of real numbers such that the type of the denominator is a real number different from zero. As a result, all functions of PVS are total in the sense that the domain and the signature must exclude explicitly any input where a function could not be defined.

As predicates used by the system to define types are arbitrary, type verification is undecidable and it usually generates proofs obligations named *type correctness conditions* (TCCs). Users have to provide proofs of generated TCCs with the help of PVS.

In PVS the λ operator defines anonymous functions. Expression $\lambda x.e$ is a function that has parameter x and returns expression e . For example, the function that returns 0 for any value of its single parameter could be defined as $\lambda x.0$ and identity function that returns the same element that is given as parameter is $\lambda x.x$. Function $\lambda k : \text{nat}. \text{if } k = 0 \text{ then } 1 \text{ else } 0$ is the sequence that for input 0, returns 1, and returns 0 for any other input.

Nowadays, systems such as PVS are fully able to certify that programs are corrects (Ross, 2005) but programmers scarcely use them. Providing a formal proof of correct behavior is a difficult task, it requires a specific training and user interfaces of proof assistants are of little help for all the work that is not done automatically. Hope is that as more and more work is done automatically, users will need only limited interactions with automatic proof checkers down to the point where no interaction is required at all. This trend was recently coined as *invisible formal methods* (Tiwari *et al.*, 2003).

1.2. A FEW WORDS ABOUT INTERVAL ARITHMETIC

In interval arithmetic scalar variables x are replaced by pairs (a, b) with the semantic that x lies in the interval $[a, b]$. Later on, we compute bounds rather than values. We use operators commonly found in programming languages such as addition, subtraction, multiplication and so on (Jaulin *et al.*, 2001).

$$\begin{aligned} [a, b] + [a', b'] &= [a + a', b + b'] \\ [a, b] - [a', b'] &= [a - b', b - a'] \\ c \cdot [a, b] &= [c \cdot a, c \cdot b] \quad c \geq 0 \\ [a, b] \cdot [a', b'] &= [\min\{aa', ab', ba', bb'\}, \max\{aa', ab', ba', bb'\}] \end{aligned}$$

Working with automatic proof checkers, we convert operations into properties (Daumas *et al.*, 2005).

$$\text{For all } x \in [a, b], y \in [a', b'] \text{ and } c \in \mathbb{R} \left\{ \begin{array}{l} x + y \in [a, b] + [a', b'] \\ x - y \in [a, b] - [a', b'] \\ c \cdot x \in c \cdot [a, b] \\ x \cdot y \in [a, b] \cdot [a', b'] \end{array} \right.$$

Decorrelation is a problem intrinsic to interval arithmetic. There is decorrelation on interval evaluation of any expression where one or more variables appear more than once. For example, the most simple scalar expression

$$x - x$$

where $x \in [0, 1]$, is replaced in interval arithmetic by

$$[0, 1] - [0, 1] = [-1, 1].$$

Everyone agrees that $x - x$ lies in the interval $[0, 0]$ but interval arithmetic produces the correct but very poor $[-1, 1]$ interval. Decorrelation and other problems lead interval arithmetic to overestimate the domain of results. Techniques are used intensively to produce constrained results.

One of such techniques is based on Taylor's theorem with Lagrange remainder where f is n times continuously derivable between x_0 and x , f is $n + 1$ times derivable strictly between x_0 and x and $0 < \theta < 1$.

$$\begin{aligned} f(x) &= f(x_0) + (x - x_0)f'(x_0) + \frac{(x-x_0)^2}{2!}f''(x_0) \\ &+ \dots + \frac{(x-x_0)^n}{n!}f^{(n)}(x_0) \\ &+ \frac{(x-x_0)^{n+1}}{(n+1)!}f^{(n+1)}(x_0 + (x - x_0)\theta) \end{aligned}$$

Adapting Taylor's theorem to interval arithmetic, we obtain the formula below (Daumas *et al.*, 2005) for x and x_0 in I .

$$\begin{aligned} f(x) \in & f(x_0) + (I - x_0)f'(x_0) + \frac{(I-x_0)^2}{2!}f''(x_0) \\ &+ \dots + \frac{(I-x_0)^n}{n!}f^{(n)}(x_0) \\ &+ \frac{(I-x_0)^{n+1}}{(n+1)!}f^{(n+1)}(I) \end{aligned}$$

Using Taylor's theorem was appropriate in (Daumas *et al.*, 2005) but it has many drawbacks:

- It is difficult to hide the use of Taylor's theorem in order to provide *invisible formal methods*. This is due to the large number of quantities involved in instantiating the theorem in its generic form. Progress has been achieved by Muñoz after the publication of Daumas *et al.*
- To use Taylor's theorem, one has to express the derivatives of function f .
- For large expressions, f alone might be too large to be expressed in PVS.

Taylor models presented in the rest of this text overcome all the previous drawbacks to the price of a less accurate approximation. We have developed a set operations for PVS that includes addition, negation, scalar multiplication, multiplication, reciprocal and exponential. We present our developments in PVS, first quickly on polynomial functions and then on Taylor models. We finish with concluding remarks and a few toy examples.

2. Implementing polynomials in PVS

For the implementation of polynomials we considered a finite list of monomial functions, a finite sequence of coefficients and an infinite power series with finite support. Finite lists or sequences usually imply the construction of a new inductive type *à la* Coq⁴ (Bertot and Casteran, 2004). We implemented polynomials as power series with finite support. This scheme is appropriate for a proof system like PVS and is compatible with NASA series libraries⁵.

Working with sequences of coefficients rather than monomial functions means that we need the `powerseries` function to evaluate polynomial P on input x . It also means that some theorems can be established on finite support series rather than polynomial functions.

2.1. FINITE SUPPORT SERIES

Our implementation of polynomials is outlined in Figure 1. It mostly describes mathematical objects (definition, function, theorems...) with common words except for the notions introduced in Section 1.1

We define predicate `finite_support (a,N)` just after the preamble. Addition of sequences was already defined and is imported from previous work in the preamble. We had to define a product operator and a composition operator. The first operator applies to generic series. The second operator requires that the first sequences a returns zero for indices above input d .

In the second half of Figure 1 we proved that negation, addition, multiplication by a scalar, multiplication and composition return finite support series provided (both) inputs are finite support series. We also proved that Cauchy's product is meaningful for finite support series. The meaning of composition can only be assessed in regard to polynomial functions.

2.2. POLYNOMIAL

As we have mentioned earlier, we use `polynomial (a, n)` function to create a power series from finite support sequence a based on `powerseries(a)(x)(N)` function implemented in previous work. Extended results on polynomial functions are presented in Figure 2 based on NASA libraries.

$$polynomial(a, n)(x) = \sum_{k=0}^n a_k \cdot x^k$$

We proved in this file that Cauchy's multiplication applies to finite support series as well as polynomial functions. We also proved that the series obtained from composing two finite support series as defined in Section 2.1 defines the same polynomial function as the one that would be obtained by composing the polynomial functions associated to the two initial series.

Technical results are also presented in this file to provide more insights to our development.

⁴ See for example http://www.lfcia.org/staff/freire/phd-gilberto/gilberto_phd_html/.

⁵ <http://shemesh.larc.nasa.gov/fm/ftp/larc/PVS-library/pvslib.html>.

```

finite_support: THEORY
BEGIN

  IMPORTING series@series, reals@sqrt, series@power_series

  a, b, c: VAR sequence[real]
  N, M, L, n, m, l, i, j: VAR nat
  x: VAR real

  finite_support(a: sequence[real], N: nat): boolean =
     $\forall (n: \text{nat}): n > N \Rightarrow a(n) = 0$ 

  cauchy(a, b: sequence[real])(n: nat): real =
     $\Sigma(0, n,$ 
       $\lambda (k: \text{nat}):$ 
        IF  $n \geq k$ 
          THEN  $a(k) \times b(n - k)$ 
          ELSE 0
        ENDIF)

  comp(a, b: sequence[real], d: nat): RECURSIVE sequence[real] =
    IF  $d = 0$ 
      THEN  $(\lambda n: \text{IF } n = 0 \text{ THEN } a(0) \text{ ELSE } 0 \text{ ENDIF})$ 
    ELSE LET  $c = (\lambda n: \text{IF } n = d \text{ THEN } 0 \text{ ELSE } a(n) \text{ ENDIF})$  IN
       $a(d) \times \text{pow}(b, d) + \text{comp}(c, b, d - 1)$ 
    ENDIF
  MEASURE d

  neg_fs: LEMMA
    finite_support(a, N)  $\Rightarrow$  finite_support(-a, N)
  add_fs: LEMMA
    finite_support(a, N)  $\wedge$  finite_support(b, M)  $\wedge$   $L \geq \max(N, M) \Rightarrow$ 
    finite_support(a + b, L)
  scal_fs: LEMMA
    finite_support(a, N)  $\Rightarrow$  finite_support(x  $\times$  a, N)
  finite_support_mult: LEMMA
    finite_support(a, N)  $\wedge$  finite_support(b, M)  $\Rightarrow$ 
    finite_support(cauchy(a, b), N + M)
  finite_support_cauchy: LEMMA
    finite_support(a, N)  $\wedge$  finite_support(b, M)  $\Rightarrow$ 
    series(a)(N)  $\times$  series(b)(M) =
    series(cauchy(a, b))(N + M)
  finite_support_comp: LEMMA
    finite_support(a, N)  $\wedge$  finite_support(b, M)  $\Rightarrow$ 
    finite_support(comp(a, b, N), N  $\times$  M)

END finite_support

```

Figure 1. Abridged and reordered theory on finite support series (see file `finite_support.pvs`)

```

polynomials_ext: THEORY
BEGIN

  IMPORTING finite_support, trig_fnd@polynomial_deriv

  a, b, d: VAR sequence[real]
  n, N, M, L: VAR nat
  c: VAR real
  x, y: VAR real

  fs_powerseq: LEMMA
    finite_support(a, N) ⇒ finite_support(powerseq(a, x), N)

  fs_condition: LEMMA
    finite_support(a, N) ⇒
      (∀ (i: posnat): a(N + i) = 0)

  scal_polynomial1: LEMMA
    x × polynomial(a, N) = polynomial(x × a, N)

  powerseries_polynomial: LEMMA
    polynomial(a, n)(x) = powerseries(a)(x)(n)

  polynomial_zero: LEMMA
    polynomial((λ (n: nat): 0), N)(x) = 0

  mul_polynomial: LEMMA
    finite_support(a, N) ∧ finite_support(b, M) ⇒
      polynomial(a, N)(x) × polynomial(b, M)(x) =
      polynomial(cauchy(a, b), N + M)(x)

  pow_polynomial: LEMMA
    finite_support(a, N) ⇒
      polynomial(a, N)(x) ^ n =
      polynomial(pow(a, n), n × N)(x)

  comp_polynomial: LEMMA
    finite_support(a, N) ∧ finite_support(b, M) ⇒
      polynomial(a, N)(polynomial(b, M)(x)) =
      polynomial(comp(a, b, N), N × M)(x);

  geom_polynomial: LEMMA
    (1 - x) × Σ(0, N, λ (i: nat): x ^ i) =
    1 - x ^ (N + 1)

END polynomials_ext

```

Figure 2. Abridged extensions to the theory on polynomial (see file `polynomials_ext.pvs`)

3. Taylor models

Taylor models (Makino and Berz, 2003) are pairs $t = (P, I)$ where P are polynomial functions of fixed degree N and I are intervals. N is a constant that cannot be changed during the evaluation of expressions. In PVS, pairs are defined using components between (`#` and `#`). Components can be addressed independently using quotes `'`, that are `t'P` and `t'I`.

Taylor model t is a correct representation of function f if it satisfies the **containment** predicate stated Figure 3,

$$\forall x \in J \quad f(x) - t'P(x) \in t'I$$

where J is usually $[-1, 1]$.

Our first task was to define operations on Taylor models. Addition, negation and multiplication by a scalar are straight forward and can be read directly from Figure 3. Naive multiplication of Taylor models creates polynomials of degree $2N$. The high order terms of the polynomials must be truncated and are accounted for in the interval part.

The `inv` reciprocal operator uses the following equality where $r \in I$, $p(0) \neq 0$ and $p(x)$ has the same sign as $p(0)$.

$$\frac{1}{p(x) + r} = \frac{1}{p(0)} \cdot \frac{p(x)}{p(x) + r} \cdot \frac{1}{1 - \left(1 - \frac{p(x)}{p(0)}\right)} \quad (1)$$

We define $q(x) = 1 - \frac{p(x)}{p(0)}$ and we expand the last fraction of (??) using the geometrical series $\sum_{i=0}^N q^i$ truncated to keep only a polynomial of degree N .

Decorrelation forbids to evaluate the penultimate fraction of (??) directly and we defined a new operator based on the lower bound and the upper bound of $I/p(J)$ that returns directly

$$\left[\frac{1}{1 + \frac{1}{lb'(I/p(J))}}, \frac{1}{1 + \frac{1}{ub'(I/p(J))}} \right].$$

This operator cannot be replaced by a direct implementation of

$$\frac{1}{1 + p(J)/I} \quad or \quad \frac{1}{1 + \frac{1}{I/p(J)}}$$

because I usually contains 0 preventing anyone to use it as a divisor.

We also implemented the exponential of Taylor models using the following equality where $r \in I$ and \hat{e}^x is a rational approximation of e^x .

$$e^{p(x)+r} = \hat{e}^{p(0)} \cdot e^{p(x)-p(0)} \cdot \frac{e^{p(0)}}{\hat{e}^{p(0)}} \cdot e^r$$

The polynomial part of the result is obtained by developing and truncating the exponential series composed with $p(x)-p(0)$. The interval part is set accordingly to account for all discarded quantities.

The five `_sharp` lemmas of the second part of Figure 3, show that the **containment** predicate is preserved by our operators. It means that we can deduce properties from evaluations of expressions using Taylor models.

```

taylor_model[N: nat, (IMPORTING interval@interval) domInterval: Interval]: THEORY
BEGIN

tm: TYPE = [#P: fs_type, I: Interval#]

tm_equal: AXIOM
  t = u ≡
    polynomial(t'P, N) = polynomial(u'P, N) ∧ t'I = u'I;

t + u: tm: tm = (#P := t'P + u'P, I := t'I + u'I#);
-t: tm = (#P := -t'P, I := -t'I#);
c × t: tm = (#P := c × t'P, I := [c] × t'I#)
t × u: tm = (#P := trunc(cauchy(t'P, u'P), N), I := ... #)
inv(t: {t: tm | same condition as below tm_inv_sharp }):
  tm = (#P := ... , I := ... #)

containment(f: [domIntervalType → real], t: tm): bool =
  ∀ xu: (f(xu) - polynomial(t'P, N)(xu)) ## t'I

tm_add_sharp: LEMMA
  containment(f, t) ∧ containment(g, u) ⇒ containment(f + g, t + u)
tm_scal_sharp: LEMMA
  containment(f, t) ⇒ containment(x × f, x × t)
tm_neg_sharp: LEMMA
  containment(f, t) ⇒ containment(-f, -t)
tm_mult_sharp: LEMMA
  containment(f, t) ∧ containment(g, u) ⇒ containment(f × g, t × u)
tm_inv_sharp: LEMMA
  ∀ (f: [domIntervalType → nzreal],
    t: {t: tm |
      t'P(0) ≠ 0 ∧
      (t'I/intervalFromRealSeq(t'P, N))'lb ≠ 0 ∧
      (t'I/intervalFromRealSeq(t'P, N))'ub ≠ 0 ∧
      (t'I/intervalFromRealSeq(t'P, N)) > -1}):
    (∀ xu:
      polynomial(t'P, N)(xu) ≠ 0 ∧
      (f(xu) - polynomial(t'P, N)(xu))/polynomial(t'P, N)(xu)
      ≠ 1
      ∧
      polynomial(λ (i: nat):
        IF i = 0 THEN 0 ELSE -t'P(i)/t'P(0) ENDIF,
        N)
      (xu)
      ≠ 1)
    ∧ Zeroless?([t'P(0)]) ∧ Zeroless?( ... )
    ∧ Zeroless?(intervalFromRealSeq(t'P, N)) ∧ containment(f, t)
    ⇒ containment(1/f, inv(t))

END taylor_model

```

```

example: THEORY
BEGIN

  IMPORTING tm_exp[5, 5, (#lb := -1, ub := 1#)]

  ch(x: tm): tm =
    (1/2) × (exp(x) + exp(-x))

  sh(x: tm): tm =
    (1/2) × (exp(x) + -exp(-x))

  seq_px: fs_type =
    λ (n: nat): IF n = 1 THEN 1/1000 ELSE 0 ENDIF

  tm_x: tm = (#P := seq_px, I := [0]#)

  example1: tm = ch(2 × tm_x) × sh(3 × tm_x)

END example

```

Figure 4. A toy example of Taylor models (see file `example.pvs`)

In addition to prove mathematical theories, PVS provides a *ground evaluator*. It is an experimental feature of PVS that enables the animation of functional specifications. To evaluate them, the ground evaluator extracts Common Lisp code and then evaluates the code generated on PVS underlying Common Lisp machine.

Uninterpreted PVS functions can be written in Common Lisp. PVS only trusts Lisp codes generated automatically from PVS functional specifications, then one can not introduce inconsistencies in PVS. However, codes are not type-checked by PVS and can break inadvertently.

PVSio⁶ is a PVS package developed by Muñoz that extends the ground evaluator with a predefined library including imperative programming language features. PVSio loads in emacs interface using `M-x load-prelude-library PVSio` and then executes with `M-x pvsio`.

4. Toy example, concluding remarks and future work

Figure 4 show how easily we can define expressions. PVSio is used to evaluate Taylor model expressions and Figure 5 shows the polynomial and interval parts of the Taylor model of degree 5 of

$$ch\left(2 \cdot \frac{x}{1000}\right) \cdot sh\left(3 \cdot \frac{x}{1000}\right) = 3 \cdot \frac{x}{1000} + \frac{21}{2} \cdot \left(\frac{x}{1000}\right)^3 + \frac{521}{40} \cdot \left(\frac{x}{1000}\right)^5 + r$$

⁶ <http://research.nianet.org/munoz/PVSio>

- **Offer a first easy step to the usage of automatic proof checkers.** It is always frustrating to spend time on questions than can easily be solved by more or less elaborate techniques. As we now provide a PVS library for interval arithmetic and for Taylor models, one should be able to answer quickly to most of the easy questions about round-off, truncation and modeling errors. Concentrating only on intricate questions is rewarding from the academia and ensures financial support from the industry.

In the future, we will implement more operations on Taylor models like square root, sine, cosine, and arctangent. We will also create PVS strategies to hide more and more details of Taylor models to users. Our main goal remains to help provide *invisible formal methods*.

Acknowledgements

The authors wish to express all their gratitude to Cesar Muñoz from the National Institute of Aerospace in Hampton, Virginia, for his tutoring and help in the many manipulations around PVS. The authors would also like to thank NASA Langley Research Center for its free PVS Class held on May 24-27, 2005.

References

- Amir D. Aczel. *Fermat's last theorem: unlocking the secret of an ancient mathematical problem*. Four Walls Eight Windows, 1996.
- Yves Bertot and Pierre Casteran. *Interactive Theorem Proving and Program Development*. Springer-Verlag, 2004.
- Marc Daumas, Guillaume Melquiond, and César Muñoz. Guaranteed proofs using interval arithmetic. In Paolo Montuschi and Eric Schwarz, editors, *Proceedings of the 17th Symposium on Computer Arithmetic*, Cape Cod, Massachusetts, 2005.
- Debbie Gage and John McCormick. We did nothing wrong. *Baseline*, 1(28):32–58, 2004.
- Information Management and Technology Division. Patriot missile defense: software problem led to system failure at Dhahran, Saudi Arabia. Report B-247094, United States General Accounting Office, 1992.
- Luc Jaulin, Michel Kieffer, Olivier Didrit, and Eric Walter. *Applied interval analysis*. Springer, 2001.
- Jacques-Louis Lions et al. Ariane 5 flight 501 failure report by the inquiry board. Technical report, European Space Agency, Paris, France, 1996.
- Kyoko Makino and Martin Berz. Taylor models and other validated functional inclusion methods. *International Journal of Pure and Applied Mathematics*, 4(4):379–456, 2003.
- Ramon E. Moore. *Interval analysis*. Prentice Hall, 1966.
- Arnold Neumaier. *Interval methods for systems of equations*. Cambridge University Press, 1990.
- Sam Owre, John M. Rushby, and Natarajan Shankar. PVS: a prototype verification system. In Deepak Kapur, editor, *11th International Conference on Automated Deduction*, pages 748–752, Saratoga, New-York, 1992. Springer-Verlag.
- Sam Owre, Natarajan Shankar, John M. Rushby, and David W. J. Stringer-Calvert. *PVS Language Reference*. SRI International, 2001. Version 2.4.
- Sam Owre, Natarajan Shankar, John M. Rushby, and David W. J. Stringer-Calvert. *PVS System Guide*. SRI International, 2001. Version 2.4.
- Philip E. Ross. The exterminators. *IEEE Spectrum*, 42(9):36–41, 2005.
- John Rushby and Friedrich von Henke. Formal verification of algorithms for critical systems. In *Proceedings of the Conference on Software for Critical Systems*, pages 1–15, New Orleans, Louisiana, 1991.

Ashish Tiwari, Natarajan Shankar, and John Rushby. Invisible formal methods for embedded control systems.
Proceedings of the IEEE, 91(1):29–39, 2003.

Reliability of Structural Reliability Estimation

Isaac Elishakoff¹ and Roberta Santoro²

¹*Florida Atlantic University, Florida, USA, email: elishako@fau.edu*

²*University, of Palermo, Italy, email: santoro@diseg.uniap.it*

Abstract: In this paper we study the reliability of calculations of the structural reliability. It compares the exact reliability expression within the Bernoulli-Euler column theory with its counterpart obtained via the finite difference expression in the buckling context.

Keywords: Reliability, Probability of failure, Buckling

1. Introduction

The recent decade is characterized in intensive increase of application of probabilistic methods in engineering (Elishakoff *et al*, 2001; Arboz *at al*, 1995, Chryssanthopoulos, 1998). The matter of the accuracy of the probabilistic design of structures, therefore, becomes of paramount importance. In probabilistic design the main quantity of interest is the structural reliability. Since its calculation involves the numerical calculation the natural question arises on the reliability of the reliability calculation. The paper by Elishakoff (1999) was apparently the first one to address this issue in the structural analysis context. Here we extend Ref. Elishakoff (2001) for the buckling of structures. In particular, we deal with the reliability of finite difference method's application to structural reliability evaluation.

There are several studies that deal with the finite difference evaluation of the buckling phenomenon in deterministic setting. Namely, the papers by Falk (1956), Salvadori (1949), Wifi *et al* (1989) ought be mentioned. Seide (1975) was able, in his seminal paper, to evaluate the analytical expression for the buckling load, when the column is subdivided by N segments. The Seide's formula is a central one in this investigation to study the reliability of the reliability evaluation.

©2006 by authors. Printed in USA.

2. Recapitulation of Seide's Solution

The differential equation that governs the buckling of a column of uniform stiffness subjected at the end by a compressive load P , reads:

$$EI \frac{d^4 w}{dx^4} + P \frac{d^2 w}{dx^2} = 0 \quad (1)$$

where EI is the bending stiffness of the column, x is the axial coordinate, w is the transversal displacement and P is the axial load.

To solve complicated problems, the ordinary differential equations are usually replaced by a set of equivalent algebraic equations that are easier to solve than the differential one. One of such methods, known as finite-difference technique, is based on the fact that a derivative of a function at a point can be approximated by an algebraic expression consisting of the value of the function at that point and at several nearby points. Here, to study the reliability of reliability calculations, we investigate the case of an uniform column that possesses the exact solution, so a direct comparison is possible with the exact solution. We first recapitulate the solution derived by Seide (1995).

In particular, using first order central difference method, under the condition of uniform nodal points spacing and for any nodal point i the Eq(1) takes the following expression

$$w_{i-2} - \left(4 - \frac{Ph^2}{EI}\right)w_{i-1} + \left(6 - 2\frac{Ph^2}{EI}\right)w_i - \left(4 - \frac{Ph^2}{EI}\right)w_{i+1} + w_{i+2} = 0 \quad (2)$$

where h is the length of each segment given by the ratio between the length of the bar L and the total number N of segments.

To solve the difference equation (2) with constant coefficients we can express the solution in the following form:

$$w_i = A\lambda^i \quad (3)$$

The introduction of the expression (3) into Eq. (2) leads to the resulting equation in λ :

$$\left(\frac{1}{\lambda} + \lambda\right)^2 - \left(4 - \frac{Ph^2}{EI}\right)\left(\frac{1}{\lambda} + \lambda\right) + 4 - 2\frac{Ph^2}{EI} = 0 \quad (4)$$

Eq.4 has the following solutions:

$$\lambda_{1,2} = 1$$

$$\lambda_{3,4} = 1 - \frac{Ph^2}{2EI} \pm \sqrt{\left(\frac{Ph^2}{2EI}\right)^2 - \frac{Ph^2}{EI}} \quad (5)$$

The consideration that

$$\left(\frac{Ph^2}{2EI}\right)^2 - \frac{Ph^2}{EI} = -\left[1 - \left(1 - \frac{Ph^2}{2EI}\right)^2\right] \quad (6)$$

allows to rewrite the solutions $\lambda_{3,4}$ in Eq.(5) in a different way:

$$\lambda_{3,4} = 1 - \frac{Ph^2}{2EI} \pm i \sqrt{1 - \left(1 - \frac{Ph^2}{2EI}\right)^2} \quad (7)$$

Thus the general solution for w_i takes the form:

$$w_i = A_1 + A_2 i + A_3 \cos i\mathcal{G} + A_4 \sin i\mathcal{G} \quad (8)$$

in which A_1, A_2, A_3 and A_4 are arbitrary constants of integration and

$$\mathcal{G} = \cos^{-1}\left(1 - \frac{Ph^2}{2EI}\right) \quad (9)$$

Determination of the four constants of integration is obtained using the four boundary conditions, two at each end of the column. For a simply supported column at both edges we have:

$$w_0 = w_N = 0; \quad w_{-1} = -w_1; \quad w_{N+1} = -w_{N-1} \quad (10)$$

For a clamped columns at both ends the boundary conditions become:

$$w_0 = w_N = 0; \quad w_{-1} = w_1; \quad w_{N+1} = w_{N-1} \quad (11)$$

Let us concentrate on the case of a simply supported column. Introduction of the expression of displacement given by Eq.(8) into the boundary conditions (10) yields:

$$\begin{aligned}
 A_1 + A_3 &= 0 \\
 A_1 + A_3 \cos \vartheta &= 0 \\
 A_1 + A_2 N + A_3 \cos N\vartheta + A_4 \sin N\vartheta &= 0 \\
 A_1 + A_2 N + A_3 \cos \vartheta \cos N\vartheta + A_4 \cos \vartheta \sin N\vartheta &= 0
 \end{aligned} \tag{12}$$

Since the system of equations (12) is homogeneous, the determinant of the coefficients of A_1 , A_2 , A_3 and A_4 must vanish. The condition to satisfy is the following:

$$4N \left(\sin \frac{\vartheta}{2} \right)^4 \sin N\vartheta = 0 \tag{13}$$

Eq.(13) implies:

$$\sin N\vartheta = 0 \tag{14}$$

which has the solutions

$$N\vartheta = k\pi \quad k=1,2,3,\dots \tag{15}$$

From Eq.(9) we can evaluate the expression for $\cos \vartheta$ as follows:

$$\cos \vartheta = \cos \frac{k\pi}{N} = 1 - \frac{Ph^2}{2EI} \tag{16}$$

Using of trigonometric relations we obtain

$$\frac{Ph^2}{EI} = 4 \sin^2 \frac{k\pi}{2N} \tag{17}$$

in which k should be set equal to unity for the smallest critical load. Keeping in mind that h is the length of each of the N segments and it is the ratio between the length L of the column and the total numbers of segments N , the critical load for a simply supported column at both ends is expressed as follows:

$$P_{cr} = \frac{EI\pi^2}{L^2} \left(\frac{\sin \pi/2N}{\pi/2N} \right)^2 \quad (18)$$

When N tends to infinity we obtain the well-known solution for critical load of a simply supported column. The expression (18) belongs to Seide (1975).

3. Probabilistic Analysis of Seide's Result

Next step is to consider the case in which the elastic modulus of the column can be treated as a continuous random variable with probability distribution function $F_E(e)$ with $e > 0$, assuming that the other parameters are deterministic quantities.

The conventional requirement to avoid buckling phenomenon is that the critical load must be greater or equal than a fixed allowable value P_0

$$P_{cr} \geq P_0 \quad (19)$$

From the expression of P_{cr} given in Eq.(18) we see that if the modulus of elasticity E is a random variable, the left hand side P_{cr} of Eq (19) also becomes a random variable. We are interested in the interval of possible values of E for which the Eq (19) is satisfied. From its definition the reliability R is the probability of the event specified in Eq (19):

$$R = Prob(P_{cr} \geq P_0) \quad (20)$$

Introducing the expression of P_{cr} given in (18), the Eq.(20) can be rewritten as follows:

$$R = Prob \left[\frac{\pi^2 EI}{L^2} \left(\frac{\sin \pi/2N}{\pi/2N} \right)^2 \geq P_0 \right] \quad (21)$$

or

$$R = 1 - Prob \left[\frac{\pi^2 EI}{L^2} \left(\frac{\sin \pi/2N}{\pi/2N} \right)^2 \leq P_0 \right] \quad (22)$$

Thus

$$R = 1 - Prob \left[E \leq \frac{L^2 P_0}{\pi^2 I} \left(\frac{\pi/2N}{\sin \pi/2N} \right)^2 \right] \quad (23)$$

Given the probability density function of the random variable E , Eq.(23) becomes:

$$R = 1 - F_E \left[\frac{L^2 P_0}{\pi^2 I} \left(\frac{\pi/2N}{\sin \pi/2N} \right)^2 \right] \quad (24)$$

where the reliability of the column equals one minus the probability distribution function F_E of the modulus of elasticity at the level $\left(\frac{L^2 P_0}{\pi^2 I} \right) \left[\frac{(\pi/2N)}{(\sin \pi/2N)} \right]^2$.

4. Probabilistic Design of the Column

Once we know the expression of R we can pose the design problem of the column, under the consideration that the structure performs acceptably if the reliability exceeds or equals a codified reliability value r_0 :

$$R \geq r_0 \quad , \quad 0 < r_0 \leq 1 \quad (25)$$

The same problem can be dealt with introduction of unreliability of the structure, defined as the probability of failure as follows

$$P_f = 1 - R \leq p_0 \quad (26)$$

where p_0 is the level of unreliability which can be tolerated .When designing a structure the purpose is to keep the reliability as much as possible close at unity. If the random variable E is characterized through its probability density function, we can express some specific design parameter, in particular the length of the bar L , as depending on number of elements N and on the value of r_0 .

Since in buckling circumstances we know the exact expression for the critical load it is possible to also evaluate the exact reliability.

We can, therefore, evaluate general expression for the “actual” reliability, according to parameters N and r_0 , that can be obtained substituting the parameter L deduced from approximate analysis in Eq.(24) into the expression of the exact value of critical load for a simply supported column.

Accuracy of FDM, in the stochastic setting, can be evaluate from the actual reliability values compared with the required r_0 .

5. Example of the Exponentially Distributed Elasticity Modulus

To give a numerical example let us consider the case of a fixed distribution for the random modulus of elasticity, in particular an exponential distribution expressed by:

$$f_E(e) = \begin{cases} 0, & e < 0 \\ a \exp[-ae], & e \geq 0, a > 0 \end{cases}$$

Mathematical expectation and variance are respectively, $M[E] = 1/a$ and $\text{Var}[E] = 1/a^2$. Keeping in mind Eq.(24) the approximate reliability takes the following form

$$R_{approx} = \exp \left[-\frac{1}{M[E]} \frac{L^2 P_0}{\pi^2 I} \left(\frac{\pi/2N}{\sin \pi/2N} \right)^2 \right] \quad (27)$$

By demanding that R_{approx} equals its codified value r_0 , we obtain for the designed quantity, namely the length L of the column the following expression :

$$L_{approx} = L(N, r_0) = \pi \sqrt{\frac{IM[E]}{P_0} \ln \frac{1}{r_0} \left(\frac{\sin \pi/2N}{\pi/2N} \right)} \quad (28)$$

The exact expression for critical load of a simply supported end is given by $P_{cr} = \pi^2 EI/L^2$. We define the exact reliability as the following expression:

$$R_{exact} = Prob \left(\frac{\pi^2 EI}{L^2} \geq P_0 \right) \quad (29)$$

Substitution of approximate value for the length (Eq.27) in the expression of exact reliability allows to evaluate the actual reliability as follows:

$$R_{actual} = R_{actual}(N, r_0) = R_{exact} \Big|_{L=L_{approx}} = 1 - \left[1 - \exp \left(-\frac{1}{M[E]} \frac{P_0 L_{approx}^2}{\pi^2 I} \right) \right] \quad (30)$$

or

$$R_{actual} = \exp\left[\left(\frac{\sin \pi/4N}{\pi/4N}\right)^2 \ln(r_0)\right] = r_0^{[(\sin \pi/4N)/(\pi/4N)]^2} \quad (31)$$

Evaluating R_{actual} for increasing number of N we obtain values that are always greater than r_0 or equal r_0 .

In the Figures 1 the percentage errors between R_{actual} and r_0 for increasing value of N and for r_0 equal, respectively, to 0.90, 0.99, 0.999 and 0.9999 are depicted.

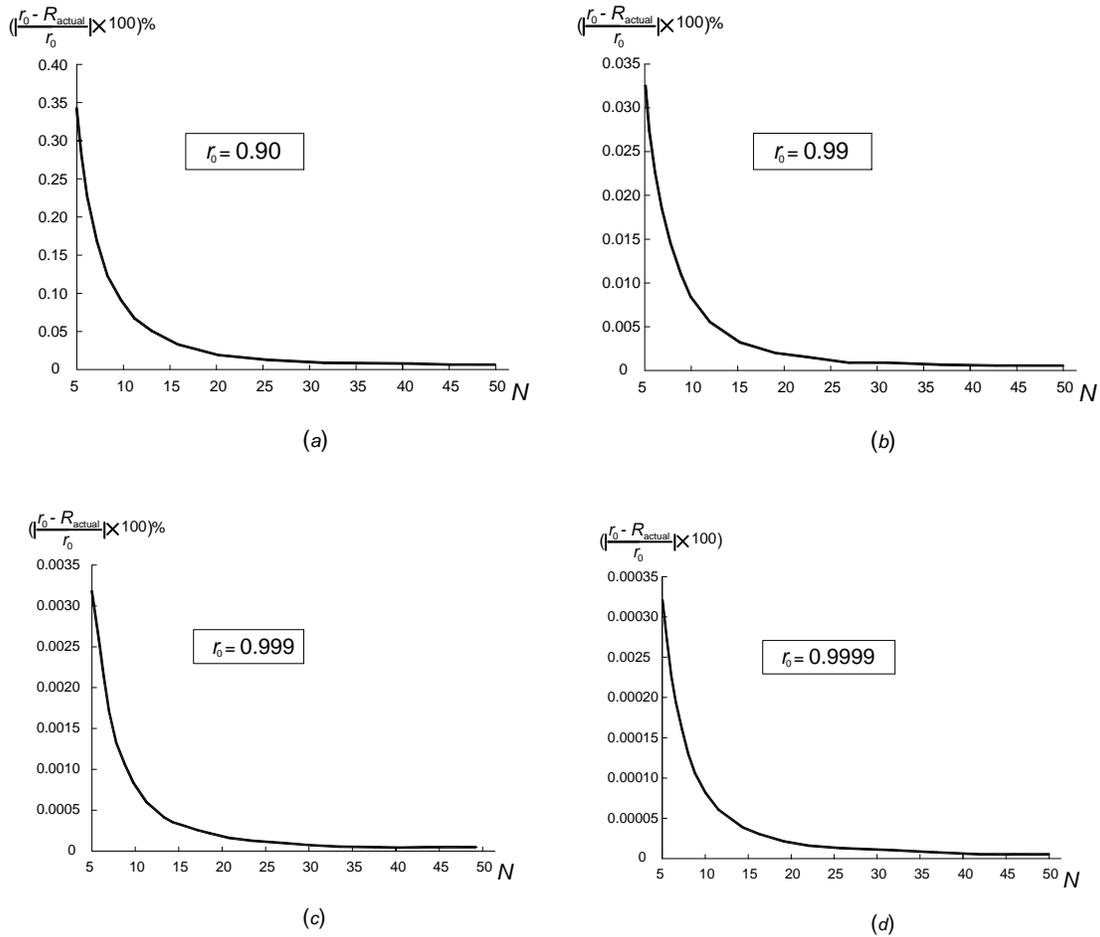


Fig. 1: Percentagewise difference between the codified and actual reliabilities

When the value of r_0 is fixed in 0.90 (Fig.1a) the percentage error goes from 0.343.% for $N=5$ ($R_{\text{actual}}=0.903084$) to 0.0864% for $N=10$ ($R_{\text{actual}}=0.900778$) to 0.0385% for $N=15$ ($R_{\text{actual}}=0.900346$).

Keeping in mind the relation between reliability and probability of failure we can evaluate analogously the *actual* probability of failure for fixed values of the tolerated one.

Fixing p_0 equal respectively to 0.1, 0.01, 0.001 and 0.0001, the Fig.2 shows the percentage error between $P_{f, \text{actual}}$ and p_0 for increasing value of N .

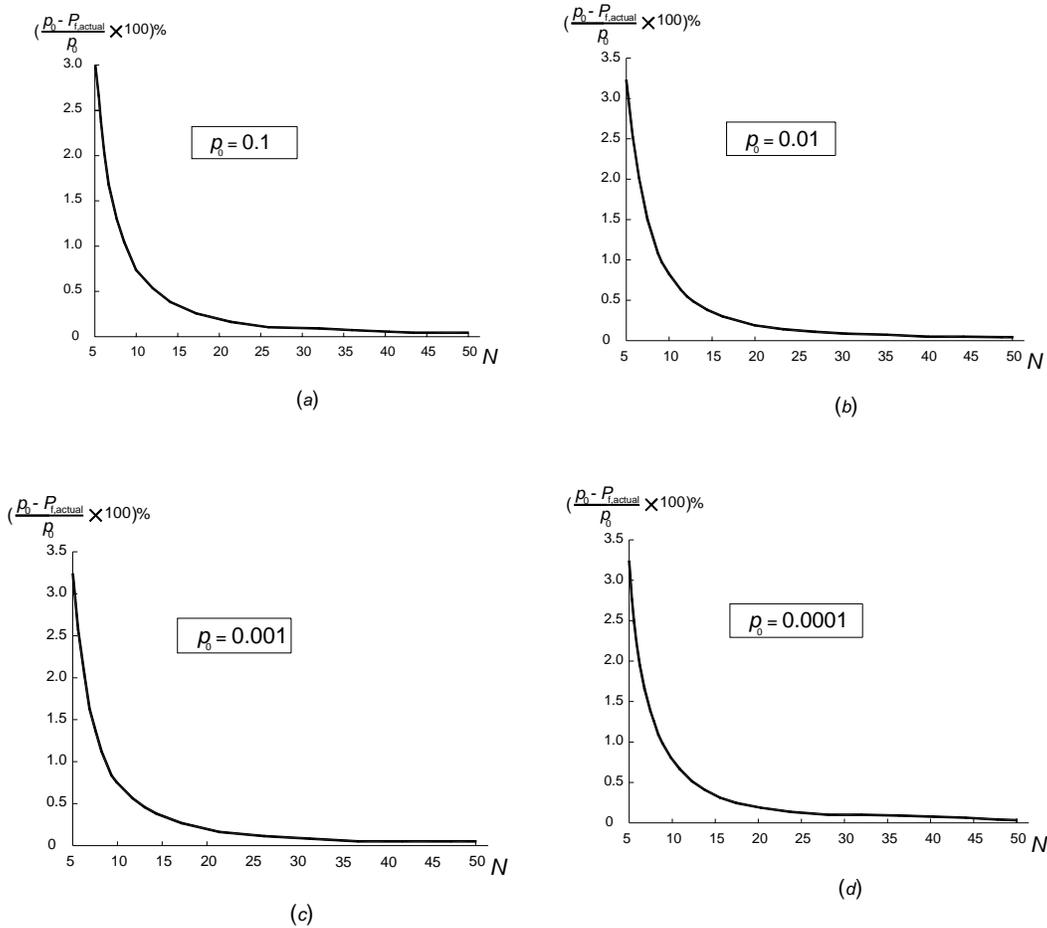


Fig.2: Error in probability of failure: Modulus of elasticity random variable with Exponential Distribution

Fixing the number of elements at $N=5$, we get $P_{f,actual}=0.0969159$ for $p_0=0.1$ ($\varepsilon=3.08411\%$), $P_{f,actual}=0.00967689$ for $p_0=0.01$ ($\varepsilon=3.23112\%$), $P_{f,actual}=0.000967547$ for $p_0=0.001$ ($\varepsilon=3.245\%$) and $P_{f,actual}=0.0000967533$ for $p_0=0.0001$ ($\varepsilon=3.24672\%$).

6. Conclusion

In this paper the reliability of the reliability calculations was studied in the buckling context. The question that was posed here is as follows: Is actual probability of failure greater than, equal to, or less than the tolerable probability of failure that is pertinent to the ideal, error-free situation?

In the example that is presented in this investigation the actual probability of failure turns out to be smaller than the tolerable level. This appears to be a good news for the reliability of the reliability calculation of the finite difference method. Whereas this conclusion cannot be extended to the other cases of the use of finite difference method, still it represents an interesting finding that was not anticipated *a priori* by the present investigators.

Acknowledgements

Isaac Elishakoff appreciates the partial financial support of the J. M. Rubin foundation of the Florida Atlantic University. Roberta Santoro appreciates the Ministry of Education of Italy for financial support of her stay at the Florida Atlantic University as a Visiting Scholar.

References

- Arbocz J., Hol J.M.A.M., Collapse of Axially Compressed Cylindrical Shells with Random Imperfections, *Thin-Walled Structures*, Vol.23, Issue 1-4, 131-158, 1995.
- Carnegie W. and Thomas J., Natural Frequencies of Long Tapered Cantilevers, *The Aeronautical Quarterly*, Vol.18, 309-320, 1967.
- Chryssanthopoulos M.K., Probabilistic Buckling Analysis of Plates and Shells, *Thin-Walled Structures*, Vol.30, Issue 1-4, 135-157, 1998.
- Elishakoff I., What may go wrong with probabilistic methods? In *Whys and Hows on Uncertainty Modelling* (Elishakoff, I. ed), pp.265-283, Springer Verlag, Vienna, 1999.
- Elishakoff I., Li Y.W., Starnes J.H., *Non-classical Problems in the Theory of Elastic Stability*, Cambridge University Press, 2001.
- Falk S., Die Knickformeln für den Stab mit n Teilstücken Konstanter Biegesteifigkeit, *Ingenieur-Archiv*, Vol.24(2), 85-91, 1956.
- Gawain T.H. and Ball R.E., Improved Finite Difference formulas for Boundary Value Problems, *International Journal of Numerical Methods in Engineering*, Vol.12, 1151-1160, 1978

- Salvadori M.G., Numerical computation of buckling loads by finite differences, *Proceedings ASCE*, Paper No.2441, Vol.75, 590-636,1949.
- Seide P., Accuracy of Some Numerical Methods For Column Buckling, *Journal of the Engineering Mechanics Division*, Proceedings of the American Society of Civil Engineers, Vol.101, No.EM5, October, 1975.
- Wifi A.S. et al, A simple discrete element mechanical model for the stability analysis of elastic structures, In *Current Advances in Mechanical Design and Production* (Kabil Y.H. and Said M.E., eds.),pp.145-156, Pergamon Press, 1989.

A Computational Environment for Interval Matrices in C++

Michael Nooner and Chenyi Hu*

Computer Science Department, University of Central Arkansas

Abstract. It is often required to manipulate interval matrices in reliable scientific computing. A portable computational environment for basic interval linear algebra subroutines (interval BLAS) is needed. We report such a environment recently developed based on the interval BLAS standard included in (Blackford and et al, 2001) and (Dongarra, J. et al, 2002).

The computational environment is object-oriented in ISO/ANSI standard C++. It consists of arithmetic, fundamental and utility functions, and set operations among intervals, interval vectors, and interval matrices. These operations are implemented as member functions of three classes: *Interval*, *IntervalVector*, and *IntervalMatrix*. This package is portable and robust with built-in error handling features. Instructions on package installation, testing, and usages are included. A sample application program is attached as an appendix.

Keywords: Interval arithmetic, Interval BLAS, C++

1. Introduction

Ever since Ray Moore introduced interval arithmetic (Moore, 1979) in 1970's, it has achieved numerous successful applications in scientific computing (Hansen, 1992; Coliss and Kearfott, 1999; de Korvin and Hu, 2004) and many of others. Similar to traditional floating point arithmetic, interval linear algebra is fundamental to most calculations and often the computationally intense part in applications of interval computing. Designers of computer programs involving linear algebraic operations have frequently chosen to implement certain low level operations, such as the dot product or the matrix vector product, as separate subprograms. This may be observed both in many published codes and in codes written for specific applications at many computer installations (Kågström, 1998; Duff and Vömel, 2002; Duff and Heroux, 2002; Li and et al., 2002; Sun Studio, 2005). With the same motivation we develop the computing environment for interval matrices.

2. The Interval BLAS Standard

In the Basic Linear Algebra Subprograms Technical (BLAST) Forum (Blackford and et al, 2001), we proposed an interval BLAS standard. Based on the standard, Sun Microsystems made its Fortran 95 implementation in its Sun Studio (Sun Studio, 2005; Walster, 2002). This software package is mostly based on the interval BLAS standard but implemented in C++. We need to briefly review some basic definitions of the standard here.

* Partially supported by NSF/CISE/CCR grant 0202042

A nonempty *mathematical interval* $[a, b]$ is the set $\{x \in \mathbb{R} \mid a \leq x \leq b\}$ where $a \leq b$. A *machine interval* $[a^*, b^*]$ is a mathematical interval whose endpoints are machine representable numbers. We say that $[a^*, b^*]$ is a machine representation of $[a, b]$ if $[a^*, b^*]$ *contains* $[a, b]$ i.e. $a^* \leq a$ and $b \leq b^*$. We say that the machine interval $[a^*, b^*]$ is a *tight representation* of a mathematical interval $[a, b]$ if and only if a^* is the greatest machine representable number which is less than or equal to a , and b^* is the least machine representable number which is greater than or equal to b . The *empty interval* \emptyset , which does not contain any real number, is required in interval BLAS. In our implementation, we use $[1, -1]$ to represent the empty interval.

Interval vectors and *interval matrices* are vectors and matrices whose entries are intervals. Both scalar (floating point number) and interval arguments are used for the specifications of routines in this paper. We use **boldface letters** to specify interval arguments. We also use overline and underline to specify the greatest lower bound and the least upper bound of an interval variable, respectively. For example, if \mathbf{x} is an interval vector, then $\mathbf{x} = [\underline{x}, \overline{x}]$.

Interval arithmetic on mathematical intervals is defined as follows.

Let \mathbf{a} and \mathbf{b} be two mathematical intervals. Let op be one of the arithmetic operations $+$, $-$, \times , \div . Then $\mathbf{a} \text{ op } \mathbf{b} \equiv \{a \text{ op } b : a \in \mathbf{a}, b \in \mathbf{b}\}$, provided that $0 \notin \mathbf{b}$ if op represents \div .

Table I gives explicit definitions of these four basic interval arithmetic operations and other operations on mathematical intervals used in this package. All operations inside a computer are performed on machine intervals. Arithmetic on machine intervals must satisfy the following property:

Containment Condition: Let $\mathbf{a} = [\underline{a}, \overline{a}]$ and $\mathbf{b} = [\underline{b}, \overline{b}]$ be intervals. Let $\mathbf{c} = [\underline{c}, \overline{c}]$ be the interval result of computing $\mathbf{a} \text{ op } \mathbf{b}$ where op is defined in Table I. If \mathbf{c} is nonempty, then \mathbf{c} must contain the exact mathematical interval $\mathbf{a} \text{ op } \mathbf{b}$.

In other words, interval arithmetic on nonempty machine intervals requires that we round down the lower bound and round up the upper bound to guarantee that the machine interval result contains the true mathematical interval result. This is needed to propagate guaranteed error bounds. To ensure our implementation satisfies the containment condition, we make use of the constants `DBL_EPSILON` and `DBL_MIN` provided in `< cfloat >` within the ISO/ANSI standard C++. As the default, our package uses double precision. Our testing shows that the rounding process we used effects the sixteenth to eighteenth significant digits on machines with 64 bits representation for the type double.

3. Functionality

This section reports the functionality (mathematical operations) and operators involving interval vectors and interval matrices. We group the functionalities into two tables. Table II lists the functionalities involving interval vectors. It includes basic algebraic operations, set operations, interval matrix-vector operations, and utility operations (including data movement) for interval vectors. Table III lists functionalities for interval matrix operations that include $O(n^2)$ and $O(n^3)$ algebraic operations, set operations, and utility operations (including data movement) for interval matrices. The \LaTeX puts the tables at the end of this paper.

Table I. Elementary interval operations

Operation	$\mathbf{a} \neq \emptyset$ and $\mathbf{b} \neq \emptyset$	Operator
Addition $\mathbf{a} + \mathbf{b}$	$[\underline{a} + \underline{b}, \bar{a} + \bar{b}]$	+
Subtraction $\mathbf{a} - \mathbf{b}$	$[\underline{a} - \bar{b}, \bar{a} - \underline{b}]$	-
Multiplication $\mathbf{a} * \mathbf{b}$	$[\min\{\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}\}, \max\{\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}\}]$	*
Division $\frac{\mathbf{a}}{\mathbf{b}}$, ($0 \notin \mathbf{b}$)	$[\min\{\underline{a}/\underline{b}, \underline{a}/\bar{b}, \bar{a}/\underline{b}, \bar{a}/\bar{b}\}, \max\{\underline{a}/\underline{b}, \underline{a}/\bar{b}, \bar{a}/\underline{b}, \bar{a}/\bar{b}\}]$	/
Intersection $\mathbf{a} \cap \mathbf{b}$	$[\max\{\underline{a}, \underline{b}\}, \min\{\bar{a}, \bar{b}\}]$ if $\max\{\underline{a}, \underline{b}\} \leq \min\{\bar{a}, \bar{b}\}$; Otherwise, \emptyset	&
Union $\mathbf{a} \cup \mathbf{b}$	$[\min\{\underline{a}, \underline{b}\}, \max\{\bar{a}, \bar{b}\}]$ if \mathbf{a} and \mathbf{b} are not disjoint;	
Cancellation $\mathbf{a} \ominus \mathbf{b}$	$[\underline{a} - \underline{b}, \bar{a} - \bar{b}]$ if $(\underline{a} - \underline{b}) \leq (\bar{a} - \bar{b})$; Otherwise, \emptyset	cancel()
Convex hull \mathbf{a}, \mathbf{b}	$[\min\{\underline{a}, \underline{b}\}, \max\{\bar{a}, \bar{b}\}]$	hull()
Square root	$\sqrt{\mathbf{a}}$	isqrt()
Exponential	$e^{\mathbf{a}}$	iexp()
Logarithm	$\log_{10}\mathbf{a}$	ilog()
Power	$\mathbf{a}^{\mathbf{b}}$	ipow()
Absolute value $ \mathbf{a} $	$\max\{ \underline{a} , \bar{a} \}$	iabs()
Trig functions		isin() icos() itan() iasin() iacos() iatan()
Disjoint test	<i>True</i> if $\mathbf{a} \cap \mathbf{b} = \emptyset$; <i>False</i> , otherwise	disjoint()
Enclosure test	<i>True</i> if $\underline{a} \leq \underline{b}$ and $\bar{b} \leq \bar{a}$; <i>False</i> , otherwise	encloses()
Interior test	<i>True</i> if $\underline{a} < \underline{b}$ and $\bar{b} < \bar{a}$; <i>False</i> , otherwise	interior()
Midpoint \mathbf{a}	$(\underline{a} + \bar{a})/2$	midpoint()
Width \mathbf{a}	$\bar{a} - \underline{a}$	width()
Assignment	$\mathbf{a} \leftarrow \mathbf{b}$	=
Insertion operator	for output	<<
Extraction operator	for input	>>
Equality test	<i>True</i> if $\underline{a} \equiv \underline{b}$ and $\bar{b} \equiv \bar{a}$; <i>False</i> , otherwise	==

4. Package Contents

This package contains three main classes (*Interval*, *IntervalVector*, and *IntervalMatrix*), and auxiliary classes such as (*IntervalVectorT*, *IntervalMatrixT*, and *INTERVAL_EXCEPTION*) for error handling and performance enhancement.

4.1. THE *Interval* CLASS

This class implements intervals and the fundamental operations among them. It is required by operations among the interval vectors and matrices. The private data members of the *Interval* class consists of the lower and upper bounds of a real interval with two double precision variables and a string for error handling. There are three built in constructors for the *Interval* class. They initiate

an interval object with zero to two double precision floating-point parameters (the interval $[0, 0]$ for zero parameter as default; directed rounding for single and double parameters).

The lower and upper bounds are accessed through the `[]` operator with index 1 or 2 respectively. For example, `x[1]` and `x[2]` return the lower and upper bounds of an *Interval* object *x*. Arithmetic operations of *Interval* objects are overloaded as arithmetic operators (`+`, `-`, `*`, `/`). Interval fundamental functions (Hu and Kearfott, 1993; Kearfott et al, 1994) are defined as friend functions (*isqrt*, *iexp*, *isin*, *icos*, *iatan*, etc). The insertion and extraction operators in C++ are also overloaded for the class. Table I lists the operators of this class.

4.2. THE *IntervalVector* AND *IntervalMatrix* CLASSES

An instance of *IntervalVector*, as per the name, is a vector whose elements are *Interval* objects. The *IntervalVector* class contains two private data members: an pointer of *Interval* type and an integer that indicates the dimension of an instance. The vector's dimension is set when the vector is created and it is not resizable.

An *IntervalMatrix* object, as per the name, is a matrix whose elements are *Interval* objects. As with the *IntervalVector* and *Interval* classes, the standard arithmetic operators are overloaded. The dimensions of an interval matrix are set when it is created with the *IntervalMatrix* constructors and is not resizable.

4.3. ERROR HANDLING AND AUXILIARY CLASSES

Error handling is done through exceptions. When an error occurs an *INTERVAL_EXCEPTION* is thrown. This *INTERVAL_EXCEPTION* type is a structure with two fields. The first is a numeric code that corresponds to the generated error type that allows for ease of programmatic error handling. The other field is a string message that corresponds to that error. The last error can be retrieved by calling the static *getLastError()* method of the *Interval* class. Similarly, the global error can be set using *Interval*'s static method *setError()*.

The *setError()* method takes two arguments. The first is the numeric code of the error that is to be set. There are twenty-six predefined error codes. Each of them has a default error message. If you send *setError()* an unknown code with no message, the global error's message is set to "Unknown Error!" A complete listing of all defined error codes and their associated messages can be found in the library's documentation for *INTERVAL_EXCEPTION*. The second argument is an optional string to use as a message. This will override the default message.

There are also two additional auxiliary template classes *IntervalVectorT* and *IntervalMatrixT* in the package for type checking and declaration. The template classes check dimensions strongly before assignment. For software robustness, the assignment operators for *IntervalMatrix* and *IntervalVector* classes do not check whether the source dimension matches the destination dimension or not. Instead, the vector or matrix is simply resized. The template versions enable checking. They require that the source and destination objects be of the same dimensions before allowing assignment.

We have also made efforts to effectively manage the memory especially for intermediary results in our implementation. Although this is opaque to general users, readers who are interested may refer the source code for the details.

5. Installation and Usage

5.1. INSTALLATION

The computational environment is available at www.geocities.com/mike_nooner as a single zip file containing the documentation, source code, test, a sample application program, a README and a LICENSE file. After unzipping the downloaded package, one may install it by following the installation instructions in the package. The package is ready to install directly on machines with GCC or Microsoft Visual Studio .NET 2003. It can be installed on other platforms with C++ compilers in compliance with ISO/ANSI standard. However, some minor modifications may be needed.

Included in the package, there are four sets of tests. The first three test whether the *Interval*, *IntervalVector*, and *IntervalMatrix* classes function properly. With the `-r` option, the containment condition is tested for matrix-matrix multiplication, matrix-vector multiplication, matrix scaled accumulation, and vector scaled accumulation with randomly generated numbers. It is recommended to run the standard test cases for the library using the packaged tester application. It is recommended to pass the test output results in a file. Otherwise, the test outputs will be written to the screen.

5.2. USING THE PACKAGE

To use the library in your applications, there are four required steps.

1. Include the file **IntBLAS.h**
2. All the classes, functions, and global variables are in the *intblas* namespace. So, make the appropriate **using** declaration(s).
3. You need to call the function *INIT_INTERVAL()* before making use of any of the classes, functions, or global variables. This function should ideally be called even before any declarations.
4. Finally, you will need to link to the **intblas.lib** static library.

A simple sample program, that performs level 0-3 interval basic linear algebra operations, is attached as the Appendix with I/O data.

6. Conclusions and future work

The computational environment for interval matrices reported in this paper has various commonly used functionalities in interval software development. It can be conveniently embedded into a standard C++ environment. We plan to further test the package and enhance it with more features. Using this package as a kernel, we are working on building applications for decision making systems based on fuzzy logic, interval valued databases, and interval matrices (Collins and Hu, 2005; de

Korvin et al, 2000; de Korvin and Hu, 2004; de Korvin et al, 2002). Suggestions, comments, and error reports are very appreciated for further improvements of the package.

References

- Blackford, G., Demmel, J., Dongarra, J., Duff, and et al. Basic Linear Algebra Subprograms Technical (BLAST) Forum Standard. *High Performance Computing Applications*, No.16, pp. 1-199, 2001. Also available at <http://www.netlib.org/blas/blast-forum/blast-forum.html>
- Collins, D. and Hu, C. Fuzzily Determined Interval Matrix Games. *Proc. 2005 Berkeley Initiative in Soft Computing, Forging the Frontiers*, in press, 2005.
- Dongarra, J. et al. An updated set of basic linear algebra subprograms (BLAS). *ACM Transactions on Mathematical Software*, 28(2), 2002.
- Corliss, G., and Kearfott, B. Rigorous Global Search: Industrial Applications. *Reliable Computing*, 1-16, 1999.
- de Korvin, A., Hu, C., and Sirisaengtaksin, O. On Firing Rules of Fuzzy Sets of Type II. *Int. J. of Applied Mathematics*, 3(2), 151-159, 2000.
- de Korvin, A., Hu, C., and Chen, P. Association Analysis with Interval Valued Fuzzy Sets and Body of Evidence. *Proc. 2002 IEEE Int. Conf. on Fuzzy Systems*, 518-523, 2002.
- de Korvin, A., Hu, C., and Chen, P. Generating and Applying Rule for Interval Valued Fuzzy Observations. *Lecture Notes in Computer Science* 3177, 279-284, 2004.
- Duff, I. S., Vömel, C. Algorithm 818: A reference model implementation of the sparse BLAS in fortran 95. *ACM Transactions on Mathematical Software*, 28(2), 2002.
- Duff, I. S., Heroux, M. A., Pozo, R. An overview of the sparse basic linear algebra subprograms: The new standard from the BLAS technical forum. *ACM Transactions on Mathematical Software*, 28(2), 2002.
- Hansen, E. Global optimization using interval analysis. Marcel Dekker, New York, 1992.
- Hu, C., Kearfott B., and Awad, A. On Bounding the Range of Some Elementary Functions in FORTRAN-77. *Interval Computations*, No.(3), 29-40, 1993.
- Hu, C., Xu, S., and Yang, X. An Introduction to Interval Computation. *Theory and Practice in System Science*, 23(4), 59-62, 2003.
- Kågström, B., van Loan, C. .
Algorithm 784: GEMM-based level 3 BLAS: portability and optimization. *ACM Transactions on Mathematical Software*, 24(3), 1998.
- Kågström, B., Ling, P., van Loan, C. GEMM-based level 3 BLAS: high-performance model implementations and performance evaluation benchmark. *ACM Transactions on Mathematical Software*, 24(3), 1998.
- Kearfott, B., Dawande, M., Du, K., and Hu, C. Algorithm 737: INTLIB: a Portable Fortran-77 Interval Standard Function Library. *ACM, Trans. on Math. Software*, 20, 447-459, 1994.
- Li, X., Demmel, J., and et al. Design, implementation and testing of extended and mixed precision BLAS. *ACM Transactions on Mathematical Software*, 28(2), 2002.
- Moore, R. E. Methods and Applications of Interval Analysis. Society for Industrial and Applied Mathematics, 1979.
- Sun Studio, Sun Microsystems, Sun Performance Library User's Guide. <http://docs.sun.com/source/819-0498/>, 2005.
- Walster, W. Interval Version of the Basic Linear Algebra Subprogram Standard (BLAS). <http://www.sun.com/software/sundev/whitepapers/blas.pdf>, 2002.

Appendix: A simple application with I/O

```

#include <iostream>
#include <iomanip>
#include <IntBLAS.h> //Include IntBLAS

using namespace std;
using namespace intblas; //The IntBLAS namespace

int main()
{
    INIT_INTERVAL(); //Must initialize the library
    Interval a( 10, 11), b ( 5.5 ), int_r;

    //The output operators use the streams format
    cout << setprecision( 25 ) << setiosflags( ios::scientific );

    int_r = a * b;
    cout << "\nLevel 0 example: [10,11]*[5.5, 5.5] =\n" << int_r << endl;

    IntervalVector m( 3 ), n( 3 ), vec_r;
    m[0] = 1.1; m[1] = 2.3; m[2] = 4;
    n[0] = a;   n[1] = b;   n[2] = a+b;

    //vec_r = 2*m + 3*n
    vec_r = scaledAccumulation( m, n, 2, 3 );

    cout << "\nLevel 1 example: 2*{1,2,4} + 3*{5,6,7} =\n" << vec_r << endl;

    IntervalMatrix x( 3, 3 );
    x[0][0] = 1, x[0][1] = 2, x[0][2] = b;
    x[1][0] = a, x[1][1] = 4, x[1][2] = 6;
    x[2][0] = 3, x[2][1] = 6, x[2][2] = Interval( 9.5 );

    vec_r = x * m;
    cout << "\nLevel 2 example: x * {1, 2, 4} =\n" << mat_r << endl;

    IntervalMatrix y( 3, 3 ), mat_r;
    y[0][0] = b, y[0][1] = 2, y[0][2] = 3;
    y[1][0] = 2, y[1][1] = a, y[1][2] = 6;
    y[2][0] = b, y[2][1] = 6, y[2][2] = int_r;

    mat_r = x * y;

```

```

cout << "\nLevel 3 example: x * y =\n" << mat_r << endl;

return 0;
}

```

THE OUTPUT OF THE ABOVE SAMPLE PROGRAM:

```

Level 0 example: [10,11]*[5.5, 5.5] =
[5.4999999999999978683717927e+01, 6.05000000000000021316282073e+01]

Level 1 example: 2*{1,2,4} + 3*{5,6,7} =
{ [3.2199999999999981525888870e+01, 3.5200000000000017053025658e+01]
  [2.1099999999999987210230756e+01, 2.1100000000000012079226508e+01]
  [5.4499999999999957367435854e+01, 5.7500000000000042632564146e+01] }

Level 2 example: x * {1, 2, 4} =
{ [2.7699999999999977973175191e+01, 2.7700000000000020605739337e+01]
  [4.4199999999999967315034155e+01, 4.5300000000000046895820560e+01]
  [5.5099999999999951683093968e+01, 5.5100000000000051159076975e+01] }

Level 3 example: x * y =
| [3.9749999999999964472863212e+01, 3.9750000000000035527136788e+01]
[5.4999999999999957367435854e+01, 5.7000000000000049737991503e+01]
[3.1749999999999971578290570e+02, 3.4775000000000028421709430e+02] |
| [9.5999999999999928945726424e+01, 1.0150000000000008526512829e+02]
[9.5999999999999928945726424e+01, 1.0200000000000009947598301e+02]
[3.8399999999999960209606797e+02, 4.2000000000000045474735089e+02] |
| [8.0749999999999928945726424e+01, 8.0750000000000071054273576e+01]
[1.2299999999999988631316228e+02, 1.2900000000000011368683772e+02]
[5.6749999999999943156581139e+02, 6.1975000000000056843418861e+02] |

```

Table II. Functionality Involving Interval Vectors:

Algebraic Operation	Mathematical Definition	Operator
Dot product	$\mathbf{r} \leftarrow \beta \mathbf{r} + \alpha \mathbf{x}^T \mathbf{y}$	scaledDot()
Vector norms	$r \leftarrow \ \mathbf{x}\ _1, r \leftarrow \ \mathbf{x}\ _2$ $r \leftarrow \ \mathbf{x}\ _\infty$	norm()
Sum	$\mathbf{r} \leftarrow \sum_i \mathbf{x}_i$	vectorSum()
Max magnitude & location	$k, \mathbf{x}_k; k = \arg \max_i \{ \underline{x}_i , \bar{x}_i \}$	max()
Min absolute value & location	$k, \mathbf{x}_k; k = \arg \min_i \{ \underline{x}_i , \bar{x}_i \}$	min()
Sum of squares	$(\mathbf{a}, \mathbf{b}) \leftarrow \sum_i \mathbf{x}_i^2, \mathbf{a} \cdot \mathbf{b}^2 = \sum_i \mathbf{x}_i^2$	sumOfSquares()
Reciprocal scale	$\mathbf{x} \leftarrow \mathbf{x}/\alpha$	reciprocalScale()
Scaled interval vector accumulation	$\mathbf{y} \leftarrow \alpha \mathbf{x} + \beta \mathbf{y}$	scaledAccumulation()
Scaled interval vector accumulation	$\mathbf{w} \leftarrow \alpha \mathbf{x} + \beta \mathbf{y}$	scaledAccumulation()
Scaled interval vector cancellation	$\mathbf{y} \leftarrow \alpha \mathbf{x} \ominus \beta \mathbf{y}$	scaledCancellation()
Scaled interval vector cancellation	$\mathbf{w} \leftarrow \alpha \mathbf{x} \ominus \beta \mathbf{y}$	scaledCancellation()
Set Operation		
Enclosed	\mathbf{x} is enclosed in \mathbf{y} if $\mathbf{x} \subseteq \mathbf{y}$	encloses()
Interior	\mathbf{x} is enclosed in the interior of \mathbf{y}	interior()
Disjoint	\mathbf{x} and \mathbf{y} are disjoint if $\mathbf{x} \cap \mathbf{y} = \emptyset$	disjoint()
Intersection	$\mathbf{y} \leftarrow \mathbf{x} \cap \mathbf{y}, \mathbf{z} \leftarrow \mathbf{x} \cap \mathbf{y}$	operator &
Hull	the convex hull of \mathbf{x} and \mathbf{y}	intervalHull()
Matrix-vector Operation		
Matrix vector product	$\mathbf{y} \leftarrow \alpha \mathbf{A} \mathbf{x} + \beta \mathbf{y}$ $\mathbf{y} \leftarrow \alpha \mathbf{A}^T \mathbf{x} + \beta \mathbf{y}$	scaledVectorMult()
Triangular solve	$\mathbf{x} \leftarrow \mathbf{T} \mathbf{x}, \mathbf{x} \leftarrow \mathbf{T}^T \mathbf{x}$ $\mathbf{x} \leftarrow \alpha \mathbf{T}^{-1} \mathbf{x}, \mathbf{x} \leftarrow \alpha \mathbf{T}^{-T} \mathbf{x}$	triangularMult()
Rank one updates	$\mathbf{A} \leftarrow \alpha \mathbf{x} \mathbf{y}^T + \beta \mathbf{A}$	rankeOneUpdate()
Utility Operation		
Vector copy	$\mathbf{x} \leftarrow \mathbf{y}$	=
Insertion operator	for output	<<
Extraction operator	for input	>>
Swap	$\mathbf{y} \leftrightarrow \mathbf{x}$	swap()
Permute vector	$\mathbf{x} \leftarrow P \mathbf{x}$	permute()
Empty element	k if $\mathbf{x}_k = \emptyset$; or -1	containsEmpty()
Left endpoint	$v \leftarrow \underline{x}$	lowerBounds()
Right endpoint	$v \leftarrow \bar{x}$	upperBounds()
Midpoint	$v \leftarrow (\underline{x} + \bar{x})/2$	midpoint()
Width	$v \leftarrow \bar{x} - \underline{x}$	width()
Construct	$\mathbf{x} \leftarrow u, v$	constructor

Table III. Functionality for Interval Matrices

Algebraic Operation	Mathematical Definition	Operator
Matrix norms	$r \leftarrow \ \mathbf{A}\ _1, r \leftarrow \ \mathbf{A}\ _F,$ $r \leftarrow \ \mathbf{A}\ _\infty, r \leftarrow \ \mathbf{A}\ _{\max}$	norm()
Diagonal scaling	$\mathbf{A} \leftarrow \mathbf{D}\mathbf{A}, \mathbf{A} \leftarrow \mathbf{A}\mathbf{D}$	diagonalScale()
Two sided diagonal scaling	$\mathbf{A} \leftarrow \mathbf{D}_1\mathbf{A}\mathbf{D}_2$	diagonalScale2()
Two sided diagonal scaling	$\mathbf{A} \leftarrow \mathbf{D}\mathbf{A}\mathbf{D}$	diagonalScale2()
Matrix acc and scale	$\mathbf{A} \leftarrow \mathbf{A} + \mathbf{B}\mathbf{D}$ $\mathbf{B} \leftarrow \alpha\mathbf{A} + \beta\mathbf{B},$ $\mathbf{B} \leftarrow \alpha\mathbf{A}^T + \beta\mathbf{B}$	scaledAccumulation() accTranspose()
Matrix add and scale	$\mathbf{C} \leftarrow \alpha\mathbf{A} + \beta\mathbf{B}$	scaledAccumulation()
Matrix matrix product	$\mathbf{C} \leftarrow \alpha\mathbf{A}\mathbf{B} + \beta\mathbf{C}, \mathbf{C} \leftarrow \alpha\mathbf{A}^T\mathbf{B} + \beta\mathbf{C},$ $\mathbf{C} \leftarrow \alpha\mathbf{A}\mathbf{B}^T + \beta\mathbf{C}, \mathbf{C} \leftarrow \alpha\mathbf{A}^T\mathbf{B}^T + \beta\mathbf{C}$ $\mathbf{C} \leftarrow \alpha\mathbf{B}\mathbf{A} + \beta\mathbf{C}, \mathbf{C} \leftarrow \alpha\mathbf{B}^T\mathbf{A} + \beta\mathbf{C},$ $\mathbf{C} \leftarrow \alpha\mathbf{B}\mathbf{A}^T + \beta\mathbf{C}, \mathbf{C} \leftarrow \alpha\mathbf{B}^T\mathbf{A}^T + \beta\mathbf{C}$	operator * operator *
Set Operation		
Enclosed	\mathbf{A} is enclosed in \mathbf{B} if $\mathbf{A} \subseteq \mathbf{B}$	encloses()
Interior	\mathbf{A} is enclosed in the interior of \mathbf{B}	interior()
Disjoint	\mathbf{A} and \mathbf{B} are disjoint if $\mathbf{A} \cap \mathbf{B} = \emptyset$	disjoint()
Intersection	$\mathbf{B} \leftarrow \mathbf{A} \cap \mathbf{B}, \mathbf{C} \leftarrow \mathbf{A} \cap \mathbf{B}$	operator &
Hull	the hull of \mathbf{A} and \mathbf{B}	intervalHull()
Utility Operations		
Matrix copy	$\mathbf{B} \leftarrow \mathbf{A}$ $\mathbf{B} \leftarrow \mathbf{A}^T$	operator = operator =
Matrix transpose	$\mathbf{A} \leftarrow \mathbf{A}^T$	transpose()
Permute matrix	$\mathbf{A} \leftarrow \mathbf{P}\mathbf{A}, \mathbf{A} \leftarrow \mathbf{A}\mathbf{P}$	permute()
Empty element	if \mathbf{A} has an empty interval element	containsEmpty()
Insertion operator	for output	<<
Extraction operator	for input	>>
Left endpoint	$C \leftarrow \underline{A}$	lowerBounds()
Right endpoint	$C \leftarrow \overline{A}$	upperBounds()
Midpoint	$C \leftarrow (\underline{A} + \overline{A})/2$	midpoint()
Width	$C \leftarrow \overline{A} - \underline{A}$	width()
Construct	$\mathbf{A} \leftarrow B, C$	constructor

Outlier Detection in Geodetic Applications with respect to Observation Imprecision

INGO NEUMANN and HANSJÖRG KUTTERER

*Geodetic Institute, University Hannover, Nienburger Strasse 1, D-30167 Hannover, Germany,
e-mail: neumann@gih.uni-hannover.de, kutterer@gih.uni-hannover.de*

STEFFEN SCHÖN

*Engineering Geodesy and Measurement Systems, Graz University of Technology, Steyrergasse 30,
A-8010 Graz, Austria, e-mail: steffen.schoen@TUGraz.at*

Abstract. The monitoring of buildings, slide slopes and crustal movements is a central task of geodetic engineering. The aim is the generation of meaningful motion and deformation models in order to quickly and specifically initiate constructional or geotechnical safety measures. The adequateness of the actions depends essentially on the quality of the observation and analysis techniques. Therefore it is important to correctly derive the model parameters and their uncertainty budget considering that the model parameters are typically estimated from a large number of heterogeneous and redundant observations by means of a least-squares adjustment. Here, the uncertainty budget is assumed to comprise both random variability and remaining systematics (imprecision). In practice, there are outliers in the data which have to be detected and eliminated. In conventional techniques only random effects are taken into account. When imprecision is considered additionally, the test strategies have to be extended accordingly. In this study imprecise extensions are obtained for the estimated outliers which are tested statistically using one- and multidimensional hypotheses. The applied procedure is outlined in detail showing both theory and numerical examples.

Keywords: outlier detection, imprecision, geodetic applications, adjustment, hypothesis testing

1. Introduction

In many engineering applications parameters are estimated from a large number of heterogeneous and redundant observations by means of a least-squares adjustment. The quality of the estimated parameters depends essentially on the adequate consideration of all uncertainties in the measurement and analysis process and on the reliability of the observations. In this paper, the uncertainty budget is assumed to comprise both random variability (stochastics) and remaining systematic effects (imprecision).

The outliers in the data occurring in practice have to be detected and then removed. Therefore the accordance of the collected data with the assumptions met in the model must be checked. This requires one- and multidimensional hypotheses tests with imprecise extensions for outlier detection and global tests based on estimated parameters and residuals (see Sections 4 and 5). In this study, the classical test approaches are extended in order to take observation imprecision into account.

© 2006 by authors. Printed in USA.

The calculation of observation imprecision is based on correction and reduction models applied to the raw observation data. It leads to intervals or fuzzy numbers for their description (see Section 2). The influence of the observation imprecision on the estimated parameters is propagated in a least-squares adjustment (see Section 3). The procedure and the criteria for the test decisions are shown in the context of fuzzy theory. They can be directly applied to pure interval mathematics. The presented approach is transferable to many other engineering applications.

Interval mathematic is an appropriate solution to describe observation imprecision by a real interval $[a]$ consisting of an upper bound a_u and a lower bound a_l or by a centre point a_m and radius a_r . The possibility of variation inside the interval demonstrates the absent knowledge about the correct value, cf. (Schön and Kutterer, 2005b). Intervals can also be defined by a suitable indicator function:

$$i_{[a]}(x) = \begin{cases} 1, & a_l \leq x \leq a_u \\ 0, & \text{else.} \end{cases} \quad (1)$$

Fuzzy-theory was founded by (Zadeh, 1995). It is an extension of the classical set theory. In the classical set theory the membership degree is either 1 (is element) or 0 (is not element). A fuzzy set \tilde{A} is uniquely defined by its membership functions $m_{\tilde{A}}(x)$ over a classical set X (e. g. $X = \mathbb{R}$) with a membership degree between 0 and 1:

$$\tilde{A} := \{(x, m_{\tilde{A}}(x)) \mid x \in X\} \quad \text{with} \quad m_{\tilde{A}} : X \rightarrow [0, 1]. \quad (2)$$

Three basic notions are relevant in the following (see Fig. 1):

$$\text{the } \alpha\text{-cut} \quad \tilde{A}_\alpha := \{x \in X \mid m_{\tilde{A}}(x) \geq \alpha\} \quad \text{with } \alpha \in [0, 1], \quad (3a)$$

$$\text{the support} \quad \text{supp}(\tilde{A}) := \{x \in X \mid m_{\tilde{A}}(x) > 0\}, \quad (3b)$$

$$\text{the core} \quad \text{core}(\tilde{A}) := \tilde{A}_1. \quad (3c)$$

It is obvious that α -cuts are classical sets. In case of convex fuzzy sets (monotonously decreasing reference functions), α -cuts are intervals. The integral over all α -cuts equals the membership function of a fuzzy set:

$$m_{\tilde{A}}(x) = \int_0^1 m_{\tilde{A}_\alpha}(x) d\alpha. \quad (4)$$

In geodetic data analysis, fuzzy numbers and fuzzy intervals are meaningful as they are convex fuzzy sets based on real numbers. Their core is either a single element (fuzzy number) which may refer to a particular observed or derived value or a classical interval which refers to a set of values (fuzzy intervals). In engineering applications LR- and LL-fuzzy numbers and intervals are of particular interest. LR-fuzzy numbers and intervals are defined by their left and right reference functions (see Eq. 5 for a LR-fuzzy interval). LR-fuzzy numbers or intervals with the same left and right reference functions are called LL-fuzzy numbers or intervals.

$$m_{\tilde{A}}(x) = \begin{cases} L\left(\frac{x_m - x - r}{c_l}\right), & x < x_m - r \\ 1, & x_m - r \leq x \leq x_m + r \\ R\left(\frac{x - x_m - r}{c_r}\right), & x > x_m + r \end{cases} \quad (5)$$

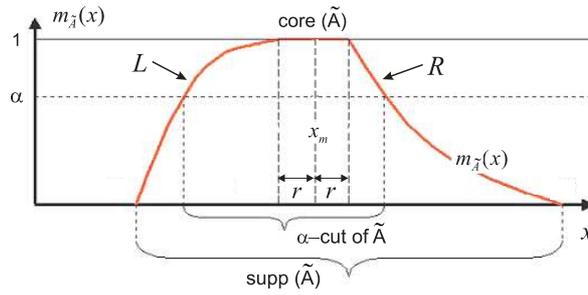


Figure 1. Fuzzy set (LR-fuzzy interval)

with x_m denoting the midpoint of the fuzzy interval, r its radius (see Fig. 1) and c_l , c_r the spread parameters of the reference functions. L-fuzzy numbers are obtained for $L = R$ and $c_l = c_r$. For further information on interval mathematics see (Alefeld and Herzberger, 1983; Jaulin et. al., 2001; Moore, 1979) and on fuzzy-theory, cf. (Bandemer and Näther, 1992; Dubois and Prade, 1980; Viertl, 1996) and (Zadeh, 1995). Studies of fuzzy data analysis in the geodetic context are presented by, e. g., (Kutterer, 2002) and (Schön and Kutterer, 2005b).

2. Observation intervals by means of a sensitivity analysis

Recently, many procedures have been introduced to calculate observation intervals in engineering applications, cf. e. g., (Braems et. al., 2000; Kieffer et. al., 2000; Morales and Son, 1998) and (Muhanna and Mullen, 2001). In geodetic data analysis, observations have to be preprocessed before they can be used for further calculation, e. g., in a least-squares adjustment. For this reason the definition of the observation intervals in geodesy is based on the correction and reduction steps for the raw observations which are based on observation error models. The applied procedure is described in detail in (Schön, 2003). The basic aspects are briefly summarized in the following.

Due to the imperfect knowledge of the influence factors of the preprocessing steps, the reduced observations are afflicted with two types of uncertainties: their stochastic behavior in terms of random variability and several non reducible remaining systematics (observation imprecision). The possible impact of remaining systematic effects is quantified by means of a sensitivity analysis of observation error models. The factual range is assessed based on expert knowledge and empirical studies. This procedure is in full accordance with international recommendations, cf. (ISO, 1995) (GUM). Note that the treatment of systematic errors is different as the GUM proposes variance propagation. An example for distance measurements was shown in (Schön, 2003). The computation in case of GPS measurements was presented in (Schön and Kutterer, 2005b). In this study, four types of observations are of particular interest: distance measurements \mathbf{l}_{dist} , direction measurements \mathbf{l}_{dir} , zenith angle measurements \mathbf{l}_z and GPS measurements \mathbf{l}_{GPS} .

3. Interval calculations for the parameters with a least-squares adjustment

The aim of geodetic applications is the estimation of parameters of interest from the observations, e. g., point coordinates (see Sect. 6 and (Koch, 1999)), deformation fields or strain tensors. Like in many other engineering application, the standard algorithm is a least-squares adjustment using a large number of heterogeneous and redundant observations. The typically non-linear observation equations are linearized in order to use linear model theory. The estimated parameters $\hat{\mathbf{x}}$ are obtained as

$$\mathbf{d}\hat{\mathbf{x}} = (\mathbf{A}^T \mathbf{P} \mathbf{A})^+ \mathbf{A}^T \mathbf{P} (\mathbf{l} - \mathbf{a}_0), \quad (6a)$$

$$\hat{\mathbf{x}} = \mathbf{x}_0 + \mathbf{d}\hat{\mathbf{x}}, \quad (6b)$$

with the $n \times u$ configuration matrix \mathbf{A} , the number of unknown parameters u , the number of observations n , the $n \times n$ weight matrix \mathbf{P} (i. e. the inverse of the variance covariance matrix (VCM) of the observations $\mathbf{C}_{\mathbf{ll}}$), the $n \times 1$ vector of observations \mathbf{l} and the vector of approximate values \mathbf{a}_0 . In geodetic networks the normal equation matrix $\mathbf{A}^T \mathbf{P} \mathbf{A}$ can be rank-deficient due to an incomplete definition of the coordinate frame through the configuration. If for example such a network is composed of distance observations only it is not possible to estimate coordinates which are required in practice. The value of rank deficiency is denoted with d . This problem can be overcome when the pseudoinverse matrix $(\mathbf{A}^T \mathbf{P} \mathbf{A})^+$ is used. A standard reference on parameter estimation (and hypotheses tests) is (Koch, 1999).

In case of observation imprecision, we assume the vector of observations as a symmetric interval $[\mathbf{l}]$, with its midpoints \mathbf{l}_m and interval radii \mathbf{l}_r , calculated by means of a sensitivity analysis (see Sect. 2). The midpoint of the interval vector $[\mathbf{l}]$ is carrier of the randomness, the remaining systematics of the analysis process are described by the vector of interval radii.

The observation imprecision is propagated to the coordinates by interval extension of the least-squares estimator, cf. (Schön and Kutterer, 2005a),

$$[\mathbf{d}\hat{\mathbf{x}}] = (\mathbf{A}^T \mathbf{P} \mathbf{A})^+ \mathbf{A}^T \mathbf{P} ([\mathbf{l}] - \mathbf{a}_0), \quad (7a)$$

$$[\hat{\mathbf{x}}] = \mathbf{x}_0 + [\mathbf{d}\hat{\mathbf{x}}], \quad (7b)$$

with the assumed precise vector of approximate values and point matrices for \mathbf{A} and \mathbf{P} . The vector \mathbf{a}_0 can also be chosen as an interval vector in order to take model uncertainties into account. Thus, both observation imprecision and model uncertainties can be treated with interval mathematical methods. Let $\mathbf{y} = \mathbf{l} - \mathbf{a}_0$ the vector of reduced observations ("observed minus computed"), then

$$[\mathbf{y}] = [\mathbf{l}] - \mathbf{a}_0 \quad (8)$$

and

$$\mathbf{y}_m = \mathbf{l}_m - \mathbf{a}_0, \quad (9a)$$

$$\mathbf{y}_r = \mathbf{l}_r. \quad (9b)$$

The parameter vector can be split up in a centre \mathbf{x}_m and radius \mathbf{x}_r part:

$$\hat{\mathbf{x}}_m = \mathbf{x}_0 + (\mathbf{A}^T \mathbf{P} \mathbf{A})^+ \mathbf{A}^T \mathbf{P} \mathbf{y}_m, \quad (10a)$$

$$\hat{\mathbf{x}}_r = | (\mathbf{A}^T \mathbf{P} \mathbf{A})^+ \mathbf{A}^T \mathbf{P} | \mathbf{y}_r, \quad (10b)$$

where $|\cdot|$ denotes the element by element absolute value of the matrix. Note that the parameter vector is exact component by component, but it overestimates the correct range, which is in general a convex polytope (zonotope), see (Schön and Kutterer, 2005a).

The residuals $\hat{\mathbf{v}}$ are estimated and treated in a similar way. They are obtained as

$$\begin{aligned}\hat{\mathbf{v}} &= \mathbf{A}\mathbf{d}\hat{\mathbf{x}} - \mathbf{y} \\ &= -\mathbf{C}_{\hat{\mathbf{v}}\hat{\mathbf{v}}}\mathbf{P}\mathbf{y},\end{aligned}\quad (11)$$

with

$$\mathbf{C}_{\hat{\mathbf{v}}\hat{\mathbf{v}}} = \mathbf{C}_{\mathbb{1}} - \mathbf{A}(\mathbf{A}^T\mathbf{P}\mathbf{A})^+\mathbf{A}^T, \quad (12)$$

the VCM of $\hat{\mathbf{v}}$. The interval extension of $\hat{\mathbf{v}}$ in terms of the midpoint $\hat{\mathbf{v}}_{\mathbf{m}}$ and the radius $\hat{\mathbf{v}}_{\mathbf{r}}$ of the residuals reads as (cf. (Kutterer, 2002)):

$$\hat{\mathbf{v}}_{\mathbf{m}} = -\mathbf{C}_{\hat{\mathbf{v}}\hat{\mathbf{v}}}\mathbf{P}\mathbf{y}_{\mathbf{m}}, \quad (13a)$$

$$\hat{\mathbf{v}}_{\mathbf{r}} = |-\mathbf{C}_{\hat{\mathbf{v}}\hat{\mathbf{v}}}\mathbf{P}|\mathbf{y}_{\mathbf{r}}. \quad (13b)$$

Then the minimum sum of the squared residuals is derived as

$$\Omega = \hat{\mathbf{v}}^T\mathbf{P}\hat{\mathbf{v}} = \mathbf{y}^T\mathbf{P}\mathbf{C}_{\hat{\mathbf{v}}\hat{\mathbf{v}}}\mathbf{P}\mathbf{y}. \quad (14)$$

4. One-dimensional hypothesis testing for outlier detection

This section presents hypotheses tests for imprecise data in the one-dimensional case. For a more general context and for a more comprehensive field of engineering applications, the test is described in fuzzy-theory. Intervals are special cases of fuzzy sets. Thus, the tests can be directly applied to the examples given in Section 6. The presented test strategy is based on (Römer and Kandel, 1995) and (Viertl, 1996). It is given in detail in (Kutterer, 2004).

4.1. TEST STRATEGY AND GENERAL TEST DECISION CRITERION

First the regions of acceptance (\tilde{A}) and rejection ($\tilde{R} = \tilde{A}^C$) have to be described with fuzzy sets. Here, the presentation is restricted to L-fuzzy intervals which are mostly relevant in the application. Hence, the region of acceptance is given as:

$$m_{\tilde{A}}(x) = \begin{cases} L_A\left(\frac{-k-x}{A_s}\right), & x < -k \\ 1, & -k \leq x \leq k \\ L_A\left(\frac{x-k}{A_s}\right), & x > k \end{cases} \quad (15)$$

with the constants k and $A_s \neq 0$ to control the shape of the region of acceptance.

Consequently, the L-fuzzy test statistic \tilde{T} with midpoint T_m and radius r is introduced:

$$m_{\tilde{T}}(x) = \begin{cases} L_T \left(\frac{T_m - r - x}{T_s} \right), & x < T_m - r \\ 1, & T_m - r \leq x \leq T_m + r \\ L_T \left(\frac{x - T_m - r}{T_s} \right), & x > T_m + r \end{cases} \quad (16)$$

with $T_s \neq 0$ the spread parameter of the test statistics.

Then the degree of agreement $\gamma_{\tilde{R}}(\tilde{T})$ of the test statistic with the region of rejection and the degree of disagreement $\delta_{\tilde{A}}(\tilde{T}) = 1 - \gamma_{\tilde{A}}(\tilde{T})$ of the test statistics with the region of acceptance are computed. With $F(\mathbb{R})$ the space of fuzzy sets over \mathbb{R} and $F(\mathbb{R} \times \mathbb{R})$ the space of fuzzy sets over $\mathbb{R} \times \mathbb{R}$, the degree of agreement $\gamma : F(\mathbb{R} \times \mathbb{R}) \rightarrow [0, 1]$ of a non empty fuzzy set $\tilde{M} \in F(\mathbb{R})$ with a fuzzy set $\tilde{N} \in F(\mathbb{R})$ ist defined by:

$$\gamma_{\tilde{N}}(\tilde{M}) := \gamma(\tilde{M}, \tilde{N}) = \frac{h(\tilde{M} \cap \tilde{N})}{h(\tilde{M})}. \quad (17)$$

The class of functions $h : F(\mathbb{R}) \rightarrow [0, \infty)$ is defined by the conditions

$$\tilde{U} = \emptyset \Leftrightarrow h(\tilde{U}) = 0, \quad (18a)$$

$$\tilde{U} \subseteq \tilde{V} \Leftrightarrow h(\tilde{U}) \leq h(\tilde{V}), \quad (18b)$$

with \emptyset the empty set. Examples for the class of functions are given in Section 4.2.

Now, the hypotheses for the imprecise test statistics (\tilde{T}) have to be introduced. The hypotheses considered here are:

$$\begin{aligned} H_0 &: E(T_m) = \mu = \mu_0 \\ H_A &: E(T_m) = \mu = \mu_0 + \delta, \quad \delta \neq 0 \\ &\text{with } T_m \sim N(\mu, 1) \end{aligned}$$

The expected value of the midpoint of the test statistics T_m , which describes the stochastic behavior, follows a standardized normal distribution N (under H_0). The presented test strategy also allows to handle empirical test values (e. g. t -distribution) and imprecise variances. The degree of rejectability $\rho_{\tilde{R}}(\tilde{T})$ of the null hypothesis H_0 is then given by

$$\rho_{\tilde{R}}(\tilde{T}) := \min(\gamma_{\tilde{R}}(\tilde{T}), \delta_{\tilde{A}}(\tilde{T})). \quad (19)$$

It is compared with a precise critical value ρ_{crit} , what leads to the test decision:

$$\rho_{\tilde{R}}(\tilde{T}) \begin{cases} \leq \\ > \end{cases} \rho_{crit} \in [0, 1] \implies \begin{cases} \text{do not reject } H_0 \\ \text{reject } H_0 \end{cases} \quad (20)$$

The imprecise test statistics \tilde{T} is only rejected if it both agrees with \tilde{R} and does not agree with \tilde{A} .

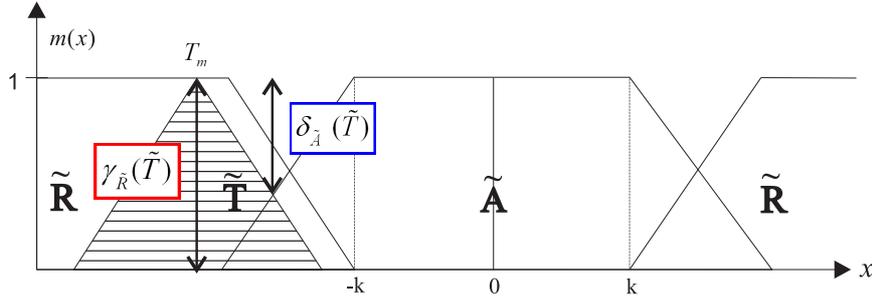


Figure 2. Geometric interpretation of the *height* criterion with a L-fuzzy test value ($r = 0$)

4.2. TWO TEST DECISION CRITERIONS

Now a suitable choice for the class of functions h in Eq. (17) has to be introduced. Section 4.2.1 describes the *height* criterion and Section 4.2.2 the *card* criterion. The *height* criterion allows an easy-to-handle test decision in case of complex fuzzy sets and the *card* criterion allows a better description of the degree of agreement between two fuzzy sets from a practical point of view, cf. (Kutterer, 2004).

4.2.1. The height criterion

For the *height* criterion, the function h in Equation (17) is defined as:

$$h(\tilde{U}) = \text{height}(\tilde{U}). \quad (21)$$

If the region of acceptance (\tilde{A}) and the test statistics (\tilde{T}) are L-fuzzy intervals this leads to the degree of rejectability:

$$\rho_{\tilde{R}}(\tilde{T}) = \min(\gamma_{\tilde{R}}(\tilde{T}), \delta_{\tilde{A}}(\tilde{T})) = \begin{cases} 0, & \text{core}(\tilde{T}) \cap \text{core}(\tilde{A}) \neq \emptyset \\ \delta_{\tilde{A}}(\tilde{T}), & \text{core}(\tilde{T}) \cap \text{core}(\tilde{A}) = \emptyset \end{cases} \quad (22)$$

with $\delta_{\tilde{A}}(\tilde{T}) = 1 - \text{height}(\tilde{T} \cap \tilde{A})$ and $\gamma_{\tilde{R}}(\tilde{T}) = \text{height}(\tilde{T} \cap \tilde{R})$. The geometric interpretation of the test is given in Figure 2. In the case of $\text{core}(\tilde{T}) \cap \text{core}(\tilde{A}) \neq \emptyset$, the null hypothesis H_0 cannot be rejected ($\rho_{\tilde{R}}(\tilde{T}) = 0$). Here, an exemplary test scenario with $T_m > 0$ is considered.

In case of different types of reference functions for the test statistics and the region of acceptance, the degree of rejectability under H_0 can not be given explicitly, it has to be computed numerically. Therefore the point of intersection x_{num} between the reference function of the test statistics and the reference function for the region of acceptance is computed by root-finding, e. g. using a Newton- or bisection algorithm (see (Jaulin et. al., 2001)).

$$L_T\left(\frac{T_m - r - x}{T_s}\right) - L_A\left(\frac{x - k}{A_s}\right) = 0 \quad \text{for } k \leq x \leq T_m - r. \quad (23)$$

This numerical solution $x_{num} > 0$ can be used for the computation of the degree of rejectability of the null hypothesis H_0 :

$$\rho_{\tilde{R}}(\tilde{T}) = \begin{cases} 1, & (\tilde{T} \cap \tilde{A}) = \emptyset \\ 1 - L_T \left(\frac{T_m - r - x_{num}}{T_s} \right), & (\tilde{T} \cap \tilde{A}) \neq \emptyset \end{cases} \quad (24)$$

Note that the numerical solution x_{num} is a function of the region of acceptance (A_s and k). The test decision in case of $\tilde{T} \cap \tilde{A} \neq \emptyset$ is now based on the comparison of the degree of rejectability with the critical value ρ_{crit} , see Eq. (25). In any case the null hypothesis is rejected for $\tilde{T} \cap \tilde{A} = \emptyset$. If $(\tilde{T} \cap \tilde{A}) \neq \emptyset$,

$$\rho_{\tilde{R}}(\tilde{T}) = 1 - L_T \left(\frac{T_m - r - x_{num}}{T_s} \right) > \rho_{crit} \implies \text{reject } H_0. \quad (25)$$

If the reference functions for the region of acceptance and the test statistics are of same type, the degree of rejectability of the null hypothesis H_0 can be computed explicitly:

$$\rho_{\tilde{R}}(\tilde{T}) = \begin{cases} 1, & (\tilde{T} \cap \tilde{A}) = \emptyset \\ 1 - L_T \left(\frac{T_m - k - r}{T_s + A_s} \right), & (\tilde{T} \cap \tilde{A}) \neq \emptyset \end{cases} \quad (26)$$

In case of $(\tilde{T} \cap \tilde{A}) \neq \emptyset$, the test decision is now described by:

$$\rho_{\tilde{R}}(\tilde{T}) = 1 - L_T \left(\frac{T_m - k - r}{T_s + A_s} \right) > \rho_{crit} \implies \text{reject } H_0. \quad (27)$$

4.2.2. The card criterion

The *card* criterion is a second possibility for the function h in Equation (17):

$$h(\tilde{U}) = \text{card}(\tilde{U}) := \int_{\mathbb{R}} m_{\tilde{U}}(x) dx. \quad (28)$$

The *card* criterion gives a suitable description of the agreement between two fuzzy sets, but the computational complexity is much higher than using the *height* criterion, in particular for complex fuzzy sets. The degree of rejectability of H_0 has to be computed based on the cardinality of the fuzzy sets resulting from the intersection of the test statistics and the region of acceptance ($\text{card}(\tilde{T} \cap \tilde{A})$) and region of rejection ($\text{card}(\tilde{T} \cap \tilde{R})$), respectively; see Eq. (29). Figure 3 shows a geometric interpretation of the *card* criterion.

$$\rho_{\tilde{R}}(\tilde{T}) := \min(\gamma_{\tilde{R}}(\tilde{T}), \delta_{\tilde{A}}(\tilde{T})), \quad (29)$$

$$\text{with } \gamma_{\tilde{R}}(\tilde{T}) = \frac{\text{card}(\tilde{T} \cap \tilde{R})}{\text{card}(\tilde{T})} \text{ and } \delta_{\tilde{A}}(\tilde{T}) = 1 - \frac{\text{card}(\tilde{T} \cap \tilde{A})}{\text{card}(\tilde{T})}.$$

In case of classical intervals for the regions of acceptance, the degree of rejectability of the null hypothesis H_0 is now easy to handle and reads:

$$\rho_{\tilde{R}}(\tilde{T}) = \gamma_{\tilde{R}}(\tilde{T}) = \delta_{\tilde{A}}(\tilde{T}) = \begin{cases} 1, & (\tilde{T} \cap \tilde{A}) = \emptyset \\ 1 - \frac{\text{card}(\tilde{T} \cap \tilde{A})}{\text{card}(\tilde{T})}, & (\tilde{T} \cap \tilde{A}) \neq \emptyset. \end{cases} \quad (30)$$

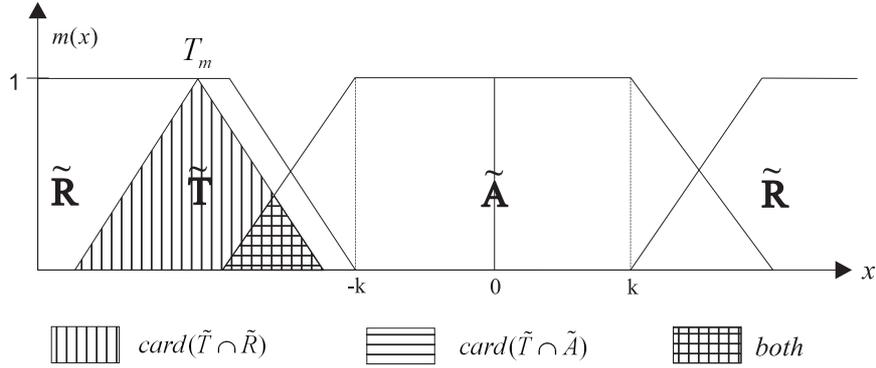


Figure 3. Geometric interpretation of the *card* criterion with a L-fuzzy test value ($r = 0$)

5. Global and multiple tests using α -cut optimization

5.1. THE PURE STOCHASTIC CASE

The pure stochastic case in multidimensional hypothesis is well known in many engineering applications, cf. (Koch, 1999). The test is based on a quadratic form $\mathbf{z}^T \mathbf{M} \mathbf{z}$ with \mathbf{z} a $n \times 1$ vector and \mathbf{M} a semi-positive definite symmetric matrix. With the expected value $E(\mathbf{z}) = \mu$ and its VCM $\mathbf{C}_{\mathbf{z}\mathbf{z}}$, the expected value of the quadratic form is given by:

$$E(\mathbf{z}^T \mathbf{M} \mathbf{z}) = \text{trace}(\mathbf{M} \mathbf{C}_{\mathbf{z}\mathbf{z}}) + \mu^T \mathbf{M} \mu. \quad (31)$$

If $\mathbf{M} \mathbf{C}_{\mathbf{z}\mathbf{z}}$ is idempotent and \mathbf{z} is normal distributed according to $\mathbf{z} \sim N(\mu, \mathbf{C}_{\mathbf{z}\mathbf{z}})$, the quadratic form $\mathbf{z}^T \mathbf{M} \mathbf{z}$ follows a non-central χ^2 -distribution, cf. (Koch, 1999):

$$\mathbf{z}^T \mathbf{M} \mathbf{z} \sim \chi^2(\text{rank}(\mathbf{M}), \mu^T \mathbf{M} \mu) = \chi^2(f, \lambda), \quad (32)$$

with $f = \text{rank}(\mathbf{M})$ the degrees of freedom and $\lambda = \mu^T \mathbf{M} \mu$ the non-centrality parameter.

From the results of a least-squares adjustment, the quadratic form may be given by the Equation (14) that follows a central $\chi^2(n - u + d, 0)$ -distribution ($\lambda = 0$) with $n - u + d$ degrees of freedom:

$$\mathbf{y}^T (\mathbf{P} \mathbf{C}_{\hat{\mathbf{v}}\hat{\mathbf{v}}} \mathbf{P}) \mathbf{y} \sim \chi^2(f, 0) \quad \text{with } f = n - u + d \quad \text{under the null hypothesis } H_0 : E(\hat{\mathbf{v}}) = \mathbf{0}. \quad (33)$$

5.2. GLOBAL AND MULTIPLE TESTS WITH OBSERVATION IMPRECISION

Now Eq. (14) has to be treated with fuzzy techniques with a given imprecise vector of reduced observations $\tilde{\mathbf{y}}$, e. g. from Section 2. We consider intentionally point matrices for $\mathbf{C}_{\hat{\mathbf{v}}\hat{\mathbf{v}}}$ and \mathbf{P} . Each kind of model uncertainty is transformed into the imprecise vector of observations, cf. (Schön and Kutterer, 2005a). The fuzzy evaluation of the quadratic form

$$\Omega = \mathbf{y}^T (\mathbf{P} \mathbf{C}_{\hat{\mathbf{v}}\hat{\mathbf{v}}} \mathbf{P}) \mathbf{y} \quad (34)$$

is based on Zadeh's extension principle. If the quadratic form fulfills the criteria

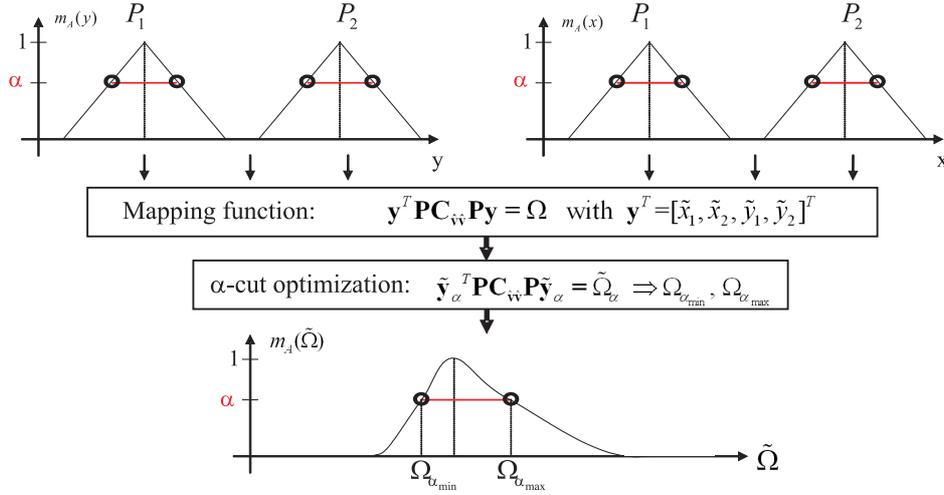


Figure 4. α -cut optimization for a point test

- convexity of the quadratic function (semi-positive definite matrix $\mathbf{PC}_{\hat{v}\hat{v}}\mathbf{P}$)
- continuity onto mapping (e. g. no change of the algebraic sign)
- convex input fuzzy sets

the extension principle can be replaced by a min-max operator of an optimization problem, cf. (Möller and Beer, 2004). The properties above are given in a least-squares adjustment with convex fuzzy numbers or fuzzy vectors and play a key role for a strict realization of the extension principle with an optimization problem. Furthermore, in case of a convex function local optimization problems can be applied. We propose a recursive Newton algorithm for minimizing/maximizing a quadratic function subject to bounds of the variables, cf. (Coleman and Li, 1996). In case of observation intervals, the optimization algorithm has to be applied only once. For fuzzy input variables the optimization algorithm is applied for a sufficient number of α -cuts of the input variables to compute the min-max values for the associated α -cut of the fuzzy output variable. The minimum and maximum values of each α -cut are given by $\Omega_{\alpha_{min}}$ and $\Omega_{\alpha_{max}}$ and the test statistics is constructed as $\tilde{T} = \tilde{\Omega}$. Figure 4 shows an example of α -cut optimization for a point test in the two-dimensional space.

Now the test strategy from Section 4 is applied, what leads for the *card*-criterion to the test scenario given in Figure 5. The test hypotheses and the test decision, respectively, are given by:

$$\begin{aligned}
 H_0 &: E(\hat{\mathbf{v}}_{\mathbf{m}}) = \mathbf{0} \\
 H_A &: E(\hat{\mathbf{v}}_{\mathbf{m}}) \neq \mathbf{0} \\
 \text{with } T_m &\sim \chi^2(f, 0) \text{ and } f = n - u + d
 \end{aligned}$$

$$\rho_{\tilde{R}}(\tilde{T}) = \left\{ \begin{array}{l} \leq \\ > \end{array} \right\} \rho_{crit} \in [0, 1] \implies \left\{ \begin{array}{l} \text{do not reject } H_0 \\ \text{reject } H_0 \end{array} \right. \quad (35)$$

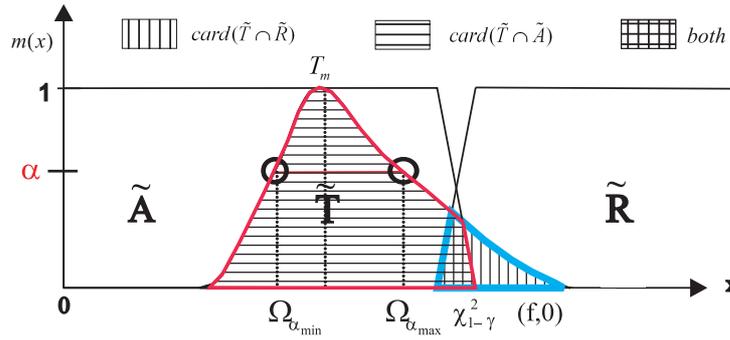


Figure 5. χ^2 -test with the *card* criterion

In the case of $\rho_{\tilde{R}}(\tilde{T}) > \rho_{crit}$, the null hypothesis H_0 is rejected and one can assume, that there are outliers in the observations. Further reasons for the rejection of the null hypothesis are non suitable choices of the functional or stochastic model-components. Therefore each observation or multiple observations has to be tested using one- and multidimensional tests to detect the outliers in the data.

Note, it is also possible to refer this problem directly to the influence factors of a sensitivity analysis, even though it has not been shown formally.

6. Examples with geodetic applications

Now selected examples for outlier detection in a three dimensional geodetic network for the monitoring of the lock Uelzen I are shown. We focus our presentation on the multidimensional case because the one-dimensional is straightforward from the given test specifications. Due to the imprecise vector of observations (see Sect. 2), the *card* criterion is used for the test decisions. The regions of acceptance are given by classical intervals with a significance level of $\gamma = 5\%$. The critical value ρ_{crit} is chosen as 0.5. Note that all numerical examples for the test statistics presented in this section are based on the *support* of the test statistics ($supp(\tilde{T})$) in order to have a clearer representation.

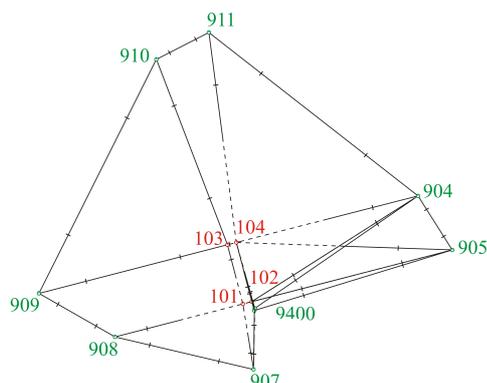
Figure 6.a shows the lock and Figure 6.b the geometric configuration of the geodetic monitoring network. The network is composed of eight control points around the lock and four object points (101-104) on top of the lock. The aim is the formulation of a meaningful deformation model for the object points in order to quickly and specifically initiate constructional or geotechnical safety measures. Therefore different measurements between the network points are carried out with special geodetic equipment such as GPS receivers and automatic tacheometers. Typical geodetic examples for the collected measurements are horizontal directions, zenith angles, distances and GPS baselines.

Table I. Interval radii and standard deviations of the observations

	Distances	Directions	Zenith angles	GPS baselines
l_r	2.0 mm	0.3 mgon	0.5 mgon	0.2 mm
σ	5.0 mm	1.0 mgon	3.0 mgon	3.0 mm



a. The lock Uelzen I



b. The geodetic monitoring network

Figure 6. Lock Uelzen

After the computation of the observation intervals based on the uncertainty of the measurements and the preprocessing steps (cf. Sect. 2), the uncertainty budget is transferred to the parameters of interest (cf. Sect. 3). Here, the parameters of interest are the 3-d point coordinates of the geodetic network points which are estimated in a least-squares adjustment. The orders of magnitude of the interval radii and the standard deviations of the observations are given in table I (for the presented examples).

6.1. PROBLEM DEFINITION

Outliers in the collected measurements may falsify point coordinates. Consequently they don't show the actual movements of the lock points. This may prevent a proper initiation of constructional or geotechnical safety measures. For this reason, the outliers in the data have to be detected and then removed. A general strategy which is typically used in Geodesy were presented by (Baarda, 1968). This strategy uses standardized residuals for the test decision in the one-dimensional case (data snooping):

$$T = \frac{\hat{v}_i}{\sigma_{\hat{v}_i}} \sim N(0, 1) \quad \text{with } H_0 : E(\hat{v}_i) = 0, \quad H_A : E(\hat{v}_i) \neq 0 \quad (36)$$

with the estimated residual \hat{v}_i , its standard deviation $\sigma_{\hat{v}_i}$ and the standardized normal distribution $N(0, 1)$. In the multidimensional case the given vector of observations is tested within a quadratic

form (cf. Section 5.1 and (Koch, 1999)). The test statistics is then given by:

$$T = \hat{\mathbf{v}}^T \mathbf{P} \hat{\mathbf{v}} \sim \chi^2(f, 0) \quad \text{with } H_0 : E(\hat{\mathbf{v}}) = \mathbf{0}, \quad H_A : E(\hat{\mathbf{v}}) \neq \mathbf{0}, \quad (37)$$

with $f = n - u + d$ the degree of freedom.

If the value of the test statistics T exceeds the chosen fractile value, the null hypothesis H_0 is rejected and the outlier is considered as revealed. This strategy is standard in geodesy.

6.2. ONE-DIMENSIONAL CASE (DISTANCE)

The first example is a one-dimensional test for the observed distance between the control point 910 and the object point 103. The midpoint $\hat{\mathbf{v}}_{\mathbf{m}}$ and the radius $\hat{\mathbf{v}}_{\mathbf{r}}$ of the residuals are computed according to the Eq. (13a) and (13b). Each observation i is tested individually and the midpoint T_{m_i} and the radius r_i of the test statistics in the imprecise case read as:

$$T_{m_i} = \frac{\hat{v}_{m_i}}{\sqrt{C_{\hat{v}\hat{v}_{ii}}}} \quad \text{under } H_0 : E(\hat{v}_{m_i}) = 0, \quad H_A : E(\hat{v}_{m_i}) \neq 0 \quad (38)$$

$$r_i = \frac{\hat{v}_{r_i}}{\sqrt{C_{\hat{v}\hat{v}_{ii}}}} \Rightarrow \text{supp}(\tilde{T}_i) = [T_{m_i} - r_i, T_{m_i} + r_i] \quad (39)$$

In this case, the numerical values for the observed distance between the points 910 and 103 are obtained by

$$T_{m_{910-103}} = \frac{\hat{v}_{m_{910-103}}}{\sqrt{C_{\hat{v}\hat{v}_{910-103}}}} = \frac{0.0101\text{m}}{0.0047\text{m}} = 2.131 \quad (40)$$

$$r_{910-103} = \frac{\hat{v}_{r_{910-103}}}{\sqrt{C_{\hat{v}\hat{v}_{910-103}}}} = \frac{0.0041\text{m}}{0.0047\text{m}} = 0.865 \Rightarrow \text{supp}(\tilde{T}_{910-103}) = [1.266, 2.996]. \quad (41)$$

Now, the test decision based on the $z_{1-\frac{\gamma}{2}}$ fractile value for the two-sided hypothesis test with $\gamma = 5\%$ reads:

$$\rho_{\tilde{R}}(\tilde{T}) = 0.60 > \rho_{crit} = 0.5 \implies \text{reject } H_0. \quad (42)$$

Obviously in case of $\rho_{crit} = 0.5$ the test is rejected, if the midpoint of the symmetric test statistics is outside the region of acceptance $T_m > z_{1-\frac{\gamma}{2}}$. In case of $\rho_{crit} > 0.5$ the midpoint of the test statistics may be outside without rejecting the test, this is caused by taking observation imprecision into account. In case of classical regions of acceptance, the value ρ_{crit} must not be chosen too small because observation imprecision is an additive term of uncertainty.

6.3. MULTIPLE TESTS (GPS BASELINE)

Second, a multiple test for a GPS baseline between the points 907 and 908 is presented. According to the pure stochastic case (see (Koch, 1999)), the imprecise quadratic form for the test reads as:

$$\Omega = \mathbf{y}^T (\mathbf{P} \mathbf{C}_{\hat{\mathbf{v}}\hat{\mathbf{v}}} \mathbf{P}) (\mathbf{B} (\mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T) (\mathbf{P} \mathbf{C}_{\hat{\mathbf{v}}\hat{\mathbf{v}}} \mathbf{P}) \mathbf{y} \quad \text{with } \mathbf{y} \in \tilde{\mathbf{y}} \quad (43)$$

and

$$\mathbf{B}^T = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \end{bmatrix}. \quad (44)$$

$\Delta x_{GPS} \quad \Delta y_{GPS} \quad \Delta z_{GPS}$

The relevant VCM of the observation to be tested in a multiple hypothesis has to be selected from the data. For this reason, the matrix \mathbf{B} is introduced, which is in case of GPS baselines defined by Eq. (44). Hence, the asymmetric imprecise test statistics \tilde{T} can be computed by means of the optimization algorithm (cf. Section 5.2 and (Coleman and Li, 1996)):

$$\begin{aligned} \text{supp}(\tilde{T}) = \text{supp}(\tilde{\Omega}) &= [9.907, 10.291] \quad \text{with } T_m = 10.097 \sim \chi^2(p, 0) \\ H_0 : E(\hat{\mathbf{v}}_{\mathbf{m}_{GPS}}) &= \mathbf{0} \quad \text{and} \quad H_A : E(\hat{\mathbf{v}}_{\mathbf{m}_{GPS}}) \neq \mathbf{0} \end{aligned} \quad (45)$$

The test decision with the fractile value $\chi_{p,1-\gamma}^2 = 7.814$ (with $p = 3$ the number of simultaneously tested observations) reads as:

$$\rho_{\tilde{R}}(\tilde{T}) = 1.0 > \rho_{crit} = 0.5 \implies \text{reject } H_0 \quad (46)$$

The GPS baseline between the points 907 and 908 is revealed as an outlier and removed from the data. In case of GPS observations in small geodetic networks ($< 5\text{km}$) with less changes in altitude, the observation imprecision is small. For this reason the spreads of the test statistics are tight and close to symmetric.

6.4. GLOBAL TEST IN LEAST SQUARES ADJUSTMENTS

In the last example we compute the imprecise global test in least-squares adjustment. The starting procedure is the fuzzy evaluation of Eq. (34) with the described optimization method. The imprecise test statistics \tilde{T} is then given by

$$\begin{aligned} \text{supp}(\tilde{T}) = \text{supp}(\tilde{\Omega}) &= [306.756, 315.851] \quad \text{with } \tilde{T}_m = 310.211 \sim \chi^2(f, 0) \\ H_0 : E(\hat{\mathbf{v}}_{\mathbf{m}}) &= \mathbf{0} \quad \text{and} \quad H_A : E(\hat{\mathbf{v}}_{\mathbf{m}}) \neq \mathbf{0} \end{aligned} \quad (47)$$

and the fractile value for the test decision reads as ($\gamma = 5\%$):

$$\chi_{f,1-\gamma}^2 = 310.396 \quad (f = 271) \quad (48)$$

Hence, the test decision

$$\rho_{\tilde{R}}(\tilde{T}) = 0.579 > \rho_{crit} = 0.5 \implies \text{reject } H_0, \quad (49)$$

shows, that with the given significance level of $\gamma = 5\%$ the global test is rejected, although the midpoint of the specified test statistics is inside of the region of acceptance ($\rho_{crit} = 0.5$). The test rejection is caused by the asymmetric imprecise test statistics which considers the quadratic impact of the imprecise influence parameters on the specified test statistic.

7. Conclusions

In this study, one and multidimensional hypotheses tests in case of observation imprecision are developed. The consideration of observation imprecision is an independent extension of the classical test approach. New approaches for outlier detection are shown, based on the intervals or fuzzy numbers of the observations. The presented test strategy allows to handle with all types of uncertainty, given as imprecise vectors of observations and can be applied to least-squares adjustments in many engineering applications. Thus, it is an essential observation-based contribution to the quality management in engineering.

Furthermore, this paper shows that an automated joint treatment of stochasticity and imprecision from the original observation up to the target parameters is possible. It turns out that remaining systematics have to be taken into account in geodetic data analysis. This allows an improved interpretation of the parameters of interest.

Finally, the presented test strategy allows a numerical calculation of the fractile value $z_{1-\frac{\alpha_{impr}}{2}}$ of the standard normal distribution. The evaluation of type I and type II errors in the imprecise case is possible.

The main focus of the following studies lies on the analysis and reanalysis of simulated and real data sets in order to make more improved decisions, in e.g. about the critical value ρ_{crit} . In addition, more extensive works in numerical computations with the *card* criterion and in the comparison between the *height* and *card* criterion has to be done.

Acknowledgements

The presented paper shows results and new ideas developed during the research project KU 1250/4-1 "Geodätische Deformationsanalysen unter Verwendung von Beobachtungsimpräzision und Objektunschärfe", which is funded by the German Research Foundation (DFG). This is gratefully acknowledged. The third author stays as a Feodor-Lynen-Fellow with F. K. Brunner. He thanks his host for giving the possibility to contribute to this paper and the Alexander von Humboldt Foundation for the financial support.

References

- Alefeld, G. and Herzberger, J.: *Introduction to Interval Computations*. Academic Press, New York, 1983.
- Baarda, W.: A Testing procedure for use in Geodetic Networks, in: *Netherlands Geodetic Commission, Publications on Geodesy* **Vol. 2 no. 5**, 1968.
- Bandemer, H. and Näther, W.: *Fuzzy Data Analysis*. Kluwer Academic Publishers, Dordrecht, 1992.
- Braems, I., Berthier, F., Jaulin, L., Kieffer, M. and Walter, E.: Guaranteed Estimation of Electrochemical Parameters by Set Inversion Using Interval Analysis, in: *Journal of Electroanalytical Chemistry* **Vol. 495**, 2000, pp. 1-9.
- Coleman, T. F. and Li, Y.: A Reflective Newton Method for Minimizing a Quadratic Function Subject to Bounds on some of the Variables, in: *SIAM Journal on Optimization* **Vol. 6 no. 4**, 1996, pp. 1040-1058.
- Dubois, D. and Prade, H.: *Fuzzy Sets and Systems*. Academic Press, New York, 1980.

- ISO (International Organisation for Standardization): *Guide to the Expression of Uncertainty in Measurements*. Printed in Switzerland, 1995.
- Jaulin, J., Kieffer, M., Didrit, O. and Walter, E.: *Applied Interval Analysis*. Springer, London, 2001.
- Kieffer, M., Jaulin, L., Walter, E. and Meizel, D.: Robust Autonomous Robot Localization Using Interval Analysis in: *Reliable Computing* **Vol. 6 no. 3**, 2000, pp. 337-362.
- Koch, K. R.: *Parameter Estimation and Hypothesis Testing in Linear Models*. Springer, Berlin and New York, 1999.
- Kutterer, H.: *Zum Umgang mit Ungewissheit in der Geodäsie - Bausteine für eine neue Fehlertheorie*. Deutsche Geodätische Kommission, Reihe C, Nr. 553, München, 2002.
- Kutterer, H.: Statistical Hypothesis tests in case of imprecise data, in: F. Sanso (Ed.): *Proceedings of the 5. Hotine-Marussi-Symposium*. International Association of Geodesy Symposia, Springer, Berlin and New York, 2004, pp. 49-56.
- Möller, B. and Beer, M.: *Fuzzy Randomness - Uncertainty in Civil Engineering and Computational Mechanics*. Springer, Berlin and New York, 2004.
- Moore R. E.: *Methods and Applications of Interval Analysis*. Siam Verlag, Philadelphia, 1979.
- Morales, D. and Son, T. C.: Interval Methods in Robot Navigation in: *Reliable Computing* **Vol. 4 no. 1**, 1998, pp. 55-61.
- Muhanna, R. L. and Mullen, R. L.: Uncertainty in Mechanism Problems - Interval-Based Approach in: *Journal of Engineering Mechanics* **Vol. 127**, 2001, pp. 557-566.
- Neumann, I. and Kutterer, H.: Geodetic Deformation Analysis with respect to Observation Imprecision and Object Fuzziness, in: *Proceedings of the International FIG Symposium in Modern Technologies, Education and Professional Practice in Geodesy and Related Fields*. Sofia, Bulgaria, 2005, pp. 147-157.
- Römer, Chr. and Kandel, A.: Statistical tests for Fuzzy Data, in: *Fuzzy Sets and Systems* **Vol. 72**, 1995, pp. 1-26.
- Schön, S.: *Analyse und Optimierung geodätischer Messanordnungen unter besonderer Berücksichtigung des Intervallansatzes*. Deutsche Geodätische Kommission, Reihe C, Nr. 567, München, 2003.
- Schön, S. and Kutterer, H.: Using Zonotopes for Overestimation-Free Interval Least-Squares -Some Geodetic Applications-, in: *Reliable Computing* **Vol. 11**, 2005, pp. 137-155.
- Schön, S. and Kutterer, H.: Realistic uncertainty measures for GPS observations, in: F. Sanso (Ed.): *A Window to the Future of Geodesy*, Sapporo, Japan, International Association of Geodesy Symposia, **No. 128**, 2005, pp. 54-59.
- Viertl, R.: *Statistical Methods for Non-Precise DATA*. CRC Press, Boca Raton, New York, London and Tokyo, 1996.
- Zadeh L. A.: Fuzzy sets, in: *Information Control* **Vol. 8**, 1965, pp. 338-353.

Modeling Hysteresis in CLIP – The Tank Flow Problem

David K. Wittenberg and Timothy J. Hickey

*Brandeis University Department of Computer Science, Waltham, MA, USA ,
dkw@cs.brandeis.edu, tim@cs.brandeis.edu*

Abstract.

Hickey and Wittenberg (Hickey and Wittenberg, 2004) study the “Two Tanks Problem”, a hybrid system described by Stursberg *et al.* (Stursberg et al., 1997). In this paper, we expand on the use of CLIP (Hickey, 2000) (a Constraint Logic Programming over Intervals and Functions language) to formally describe more complex systems. We add complexity in several forms. The simplest is to have a larger system. We move from a system with two tanks to one with four tanks, and we add non-linear valves to the pipes connecting the tanks. This example easily generalizes to an N-tanks problem where the tanks, connected by pipes, form an arbitrarily complex graph. The more important addition is the refinement of the model in several places. We rigorously model a valve in which the flow varies exponentially with the valve position over much of the valve’s range, and then discontinuously as the valve is almost closed. We introduce hysteresis in our analysis to avoid an infinite loop of zero-time transitions, and we discuss why our techniques should not have trouble with “Zeno” transitions.

The possibility of Zeno behaviour (Zhang et al., 2001) can arise either from physical reasons (a value near zero, so the sign of the changes is hard to know) or for modeling reasons (the system is near the boundary between two behaviour regimes, and while both regimes describe similar behaviour near the boundary, the model might switch between the two regimes infinitely often in a finite time). An elegant feature of our model is that we use the same technique of hysteresis to prevent the Zeno behaviour from either cause. This is easily done in CLIP by changing the conditions for a state change from one to the other to include hysteresis.

Keywords: Hybrid Systems, CLP, Intervals, Interval Arithmetic

1. Introduction

We use CLIP (a CLP language over analytic functions) to rigorously model hybrid systems. This paper extends our earlier work by using hysteresis to preserve the rigor of the model in the face of both non-analytic points and so called “Zeno” behaviour of a model.

There are two reasons for a point in the family of ODEs which describe a system to be non-analytic. One is simply that the ODE is non-analytic, and the other is that the system changes from a regime in which one ODE applies to a regime in which a different ODE applies. In a hybrid system, an ODE change can occur either because of a state change (the digital controller changes state) or because of what we call a “regime change” which is a point in which the system evolves from one regime to another - perhaps because a level passes a critical point.

© 2006 by authors. Printed in USA.

A separate problem for hybrid systems is what are called “Zeno” systems. A Zeno system is one in which the model makes an infinite number of state changes in a finite amount of simulated time. This obviously causes the model to fail.

1.1. HYBRID SYSTEMS

A *Hybrid System* is a system composed of a digital part (typically a small computer) and an analog part (typically a physical system with sensors and actuators). The field of hybrid systems is the study of systems in which discrete events and continuous dynamic events interact. All computer controlled or monitored processes in the real world are hybrid systems. As a field of study, “Hybrid Systems” has come to include the study of the analog part of a system in an area where reliability is at a premium, typically because of the cost (in lives or money) of a failure of such a system. Some hybrid systems papers study only the analysis of the analog part of a system. Hybrid systems research grew out of real time computation, control theory, and program verification. Hybrid systems research strives to prove properties such as stability about complex safety critical systems. In the chemical engineering literature, hybrid systems are sometimes called “combined discrete/continuous processes”. An important use of hybrid systems is to prove “safety properties”, which are statements of the form “measurement x is within range $[a, b]$ ” such as “the water level in this tank never overflows”. Because safety properties are constraints, they fit naturally in a CLP approach. One widely studied hybrid system is the tank flow problem introduced by Kowalewski *et al.* (Kowalewski et al., 1999). It is this system that we discuss here.

The history of hybrid systems starts with Fahrland’s 1970 paper (Fahrland, 1970) which asked “Why limit the modeling to either discrete event or continuous when situations are evolving that require more interdisciplinary solutions”. Very little was done for the next twenty years, and Fahrland’s work is rarely cited. Fahrland may have been influenced by Roger Brockett who was also at Case Institute of Technology, and who later did some seminal work on hybrid systems. The first conference on the subject was the 1991 REX workshop titled Real Time: Theory in Practice (de Bakker et al., 1991) where the term “hybrid automata” was introduced. Since that time, real time systems and hybrid systems work has diverged, with real time work focusing more on the computer with its latency issues, and hybrid systems focusing more on accurate modeling of the analog part of the system. While it’s not clear how to put a real time model (explicit limits on the time for processing) in the standard formalism for hybrid automata, the techniques introduced in this thesis can easily model digital components as long as their latency can be bounded.

There has been considerable research on developing formal models of hybrid systems. Among others, Davoren and Nerode developed logics (Davoren and Nerode, 2000), Maler *et al.* (Maler et al., 1991), Lynch *et al.* (Lynch et al., 1999; Lynch et al., 2001), Henzinger *et al.* (Henzinger, 1996), and Alur *et al.* (Alur et al., 1995) developed formal models. From our point of view, a limitation of these models is the difficulty in applying them to real systems, and the amount of overhead that must be relied on to trust the results.

1.2. EARLIER INTERVAL AND CLP APPROACHES TO HYBRID SYSTEMS

We are not the first to apply interval arithmetic techniques to the problem of rigorously modeling hybrid systems. HyperTech (Henzinger et al., 2000) took a major step towards reliability of their

results by using interval arithmetic ODE solving as a tool to add rigor to the very successful HyTech system. Our system merges, for the first time, the rigor of the formal model approaches and the practicality of the more engineering-based approaches by employing validated ODE solving. Our approach has several advantages over earlier interval models:

- CLIP is declarative, so that it describes the system being modeled directly.
- CLIP is logic based, so it can be viewed directly as a theorem prover using CLP logic.
- CLIP is constraint based. It doesn't require one to fully specify a system. CLIP allows one to understand some properties of a system based on initial assumptions.

Others have used constraint logic programming to model and analyze hybrid systems. Gupta *et al.* (Gupta et al., 1995)(Gupta et al., 1996) introduced a ground breaking approach called “hybrid cc” which allowed one to formally describe hybrid systems using a logic programming language with constraints. Urbina (Urbina, 1996) has pioneered another approach using CLP(\mathcal{R})(Jaffar et al., 1992) to model and analyze hybrid systems. Delzanno and Podelski (Delzanno and Podelski, 1999; Delzanno and Podelski, 2001) have explored analyzing hybrid systems using CLP(Q,R) (Holzbaur, 1995), a system which handles linear constraints with real and/or rational coefficients, as well as Boolean constraints. Their approach is to define a translator from Shankar's guarded command language (Shankar, 1993) to CLP(Q,R).

2. CLIP

CLP(I) is an interval-based constraint logic programming (CLP) language whose domain is the set of real numbers. The class of CLP languages (and their syntax and semantics) was introduced by Jaffar and Lassez in 1987 (Jaffar and Lassez, 1987). Jaffar and Maher provide an excellent survey (Jaffar and Maher, 1994) of the fundamental concepts of CLP. The idea of calculating over intervals of reals comes from Moore's 1966 book on Interval Arithmetic (Moore, 1966). The idea of combining CLP and Interval Arithmetic was first conceived by Cleary (Cleary, 1987) but the first production quality CLP(I) interpreter was the BNR Prolog system developed by Older, Vellino, and Benhamou (Research, 1988), (Benhamou and Older, 1997),(Older and Vellino, 1993). BNR Prolog was designed to be verifiably correct in the sense that the intervals it returned were mathematically guaranteed to contain all solutions to the underlying arithmetic constraints. The system however was proprietary and the underlying algorithms were never published in the scientific literature.

CLIP was originally developed as an open source implementation of CLP(I) by Qun Ju and Tim Hickey (Hickey and Ju,) (Hickey and Ju., 1997) CLIP has subsequently been extended by Tim Hickey, who added the CLP(F) language, which provides constraints over functions. CLIP is built on top of Prolog (Prolog 95, 1995), (Deransart et al., 1996), and currently runs on GNU Prolog (Diaz, 2002) and ALS Prolog. The fundamental philosophy is to have a relatively small base of sound primitive constraint contractors which are simple enough so that one can argue convincingly, if not formally prove, that they are correct, and then build more complex solvers on top of the proven system. Since the complex solvers built on CLIP primitives are made up of sound simple solvers,

they are also sound. An important feature of CLP languages is that they are theorem provers, so that each answer generated by a CLP program has a direct interpretation as a theorem about the underlying domain.

CLIP can be considered to be a constraint engine over intervals and functions which interfaces to the Prolog engine (a constraint solver over general finite domains). The CLP(F) language solves analytic constraints by soundly approximating sufficiently differentiable functions by power series with remainder terms and introducing arithmetic constraints among the Taylor coefficients of the functions at the endpoints, at points in the interval, and over the entire range.

3. Generalized Tank Flow Problem

In a hybrid system, the interface between the analog and the digital part involves imperfect hardware whose description must include error bars. The models of system behaviour are often particularly imprecise near boundaries. We use intervals to handle the issue of imprecision in measurements, and use intervals in a novel way to rigorously model the behaviour of systems near boundary points. We start by adding valves to the model. We note that the model in Kowalewski *et al.* has the behaviour of the valve discontinuous at 0 (by 5% of full flow), and show how a broad constraint describes that.

In (Wittenberg, 2004) we showed how CLIP could model the simple two tanks problem. In this paper, we show how the CLIP model can easily be extended to the “tank flow problem”, an extension of the two tanks system to an arbitrary number of tanks, and to model it more rigorously than other methods can. Here, we consider a four tank version with valves between each pair of tanks and at the output.

3.1. MATHEMATICS OF THE TANK FLOW PROBLEM

The problem we study is diagrammed in Fig. 1 and the parameters and variables are shown in Table I. The problem can be described as follows: There are n tanks, numbered from 1 to n , with the bottom of each tank lower than the bottom of the previous tank. The depth of the water in tank j at time t is given by $D_j(t)$. The depths D_j are measured from the bottom of their respective tanks. The altitude of the bottom of tank j is H_j above an arbitrary horizontal datum, perhaps sea level. Each tank j has a horizontal pipe leaving from the bottom of the tank. The flow through that pipe is I_j , and there is a valve V_j on the pipe. There is a constant inflow of water into tank 1 (the uppermost tank) where the flow rate is given by a constant f_{00} .

The general equation for flow through a pipe is that the rate of flow is proportional to pipe coefficient times the square root of the height difference of the water levels at each end. Specifically, the flow $I_j(t)$ through pipe j connecting tank j to tank $j + 1$ is governed by a pair of ODEs in the resistance $R_j(t)$ to flow.

$R_j(t)$ is a function of the pipe coefficient C_j , valve coefficient E_j , and the valve position $P_j(t)$ and to the square root of the pressure difference. The pipe coefficient C_j describes how easily water flows through the pipe when the valve is in the fully open position. The valve coefficient E_j is the exponent describing how much the valve cuts off the flow as a function of the valve position. The pressure difference is proportional to the difference in water heights on each end of the pipe.

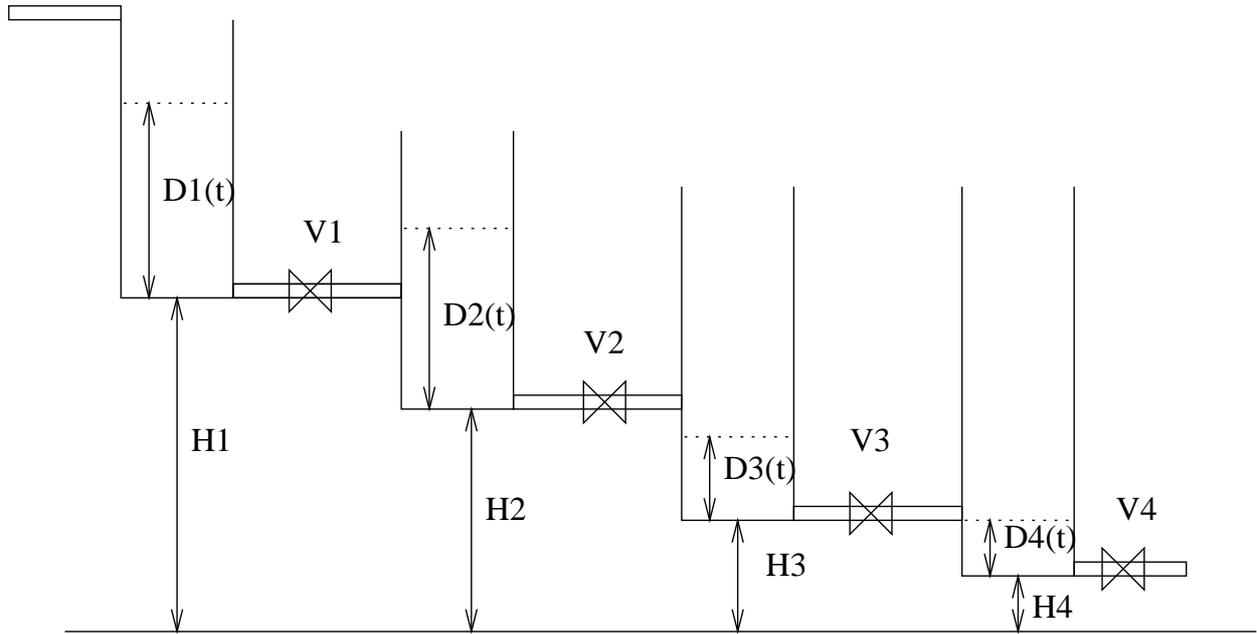


Figure 1. Diagram of Tank Flow system for $n = 4$

Table I. Parameters and Variables

H_j	Height of tank j above sea level
V_j	inverse of time for valve j to open or close (valve speed)
C_j	pipe coefficient of pipe j when the valve is fully open
E_j	exponent for describing the valve's behaviour
$P_j(t)$	position of valve j . 0 is fully closed, 1 is fully open
$M_j(t)$	valve motion – closing, opening, halted
$R_j(t)$	program variable for valve regime - shut, transition, normal
$D_j(t)$	Depth of water in tank j at time t (measured from bottom of tank.)
$I_j(t)$	rate of flow through pipe j at time t

If the water level in the lower tank is below the pipe bringing water in, there is no back pressure in the pipe, so we can ignore the water level in the lower tank. If the water level in the lower tank is higher than the input pipe, we have to include the effect of back pressure on the flow through the pipe. Therefore, we have a pair of ODEs for each pipe. One ODE of the pair holds when the water in the lower tank ($j + 1$) is below the level of the connecting pipe ($D_{j+1}(t) < H_j - H_{j+1}$), the other member of the pair holds when the water level is above the connecting pipe ($D_{j+1}(t) > H_j - H_{j+1}$). When the water level is equal to the height of the connecting pipe, the ODEs are the same, so we choose one arbitrarily. Later (Section 5.2) we will show how to rigorously handle this point where

the ODEs change, and which is therefore not analytic. Note that even if the water level is above both ends of a pipe, if the water levels (measured from sea level) are equal, the ODE is non-analytic because the square root¹ function's derivative is infinite at 0.

In our simulation, later tanks do not have higher water level than earlier tanks, so we do not consider "backwards" flow, though it is a simple extension.

We handle these two different reasons for an ODE to be non-analytic in exactly the same manner, described in Section 5.

The valve decreases the flow by a fraction which decreases exponentially with the valve position. Recall that since D_j is the depth in a tank, I_{j-1} the flow into that tank, and I_j the flow out, $D'_j = I_{j-1} - I_j$. Define HDiff_j to be the difference in altitude between the bottom of tank j and the bottom of tank $j + 1$. That is: $\text{HDiff}_j = H_j - H_{j+1}$. The ODEs for flows in pipe j are:

$$I_j(t) = \begin{cases} 0 & P_j(t) = 0 \\ e^{E_j \cdot (1-P_j(t))^3} \cdot C_j \sqrt{D_j(t) - D_{j+1}(t) + \text{HDiff}_j} & D_{j+1}(t) > \text{HDiff}_j \\ e^{E_j \cdot (1-P_j(t))^3} \cdot C_j \sqrt{D_j(t)} & D_{j+1}(t) \leq \text{HDiff}_j \end{cases}$$

Where P_j is the position of the valve; C_j is the pipe coefficient; the value under the radical is the effective difference in height between the water levels of the two tanks, and the exponential term is the fraction by which the valve decreases the flow.

4. Handling State Changes

A hybrid system of any size will have different ODEs to describe it at different times. Writing each ODE explicitly (as we did for a simpler example in (Hickey and Wittenberg, 2004)) is impractical because of a combinatorial explosion in the number of ODEs. To avoid this problem, we parameterize the ODEs describing the system, so a state change is modeled by a change in some of the parameters to an ODE rather than by making a different ODE active.

The ODEs governing a hybrid system can change for either of two reasons. The first is if the digital part of the system has a state change which affects the ODEs. We call this a *program control change*. The other is if the continuous system evolves in such a way as to change the ODEs, such as evolving to a point where a tank overflows, or the water level in a tank rises above the input pipe to that tank, causing back pressure. We call these events *regime changes*. One case of regime change is when a valve that had been opening (or closing) becomes fully open (or closed). That affects the ODEs, by changing the rate at which the valve position changes, not by changing the water flow directly. A helpful feature of CLP(F) is that we can model changes in ODEs caused by program control and those caused by regime changes in exactly the same way.

¹ We really want a function which is the positive square root of a positive number, and the negative square root of the absolute value of a negative number to properly describe the fluid flow. This function is also not analytic at 0.

Figure 2 is a state diagram for each valve except the last in the tank flow problem. (The last valve has no lower tank, so the level in the lower tank can't rise above the height of the pipe.) The states are described by two ternary variables, M (valve motion regime) describes the motion of the valve as one of (**opening**, **closing**, **halted**), while R (valve position regime) is one of (**shut**, **trans**, **normal**). When R takes the value **shut** it means that the valve is closed, **normal** means that the valve is open, and not too near the closed position. When R takes value **trans** the valve is in a transitional region and is nearly, but not quite closed. The transitional region is used to model the regime where the ODEs are not well understood, so we use a simple over-approximation constraint in that regime.

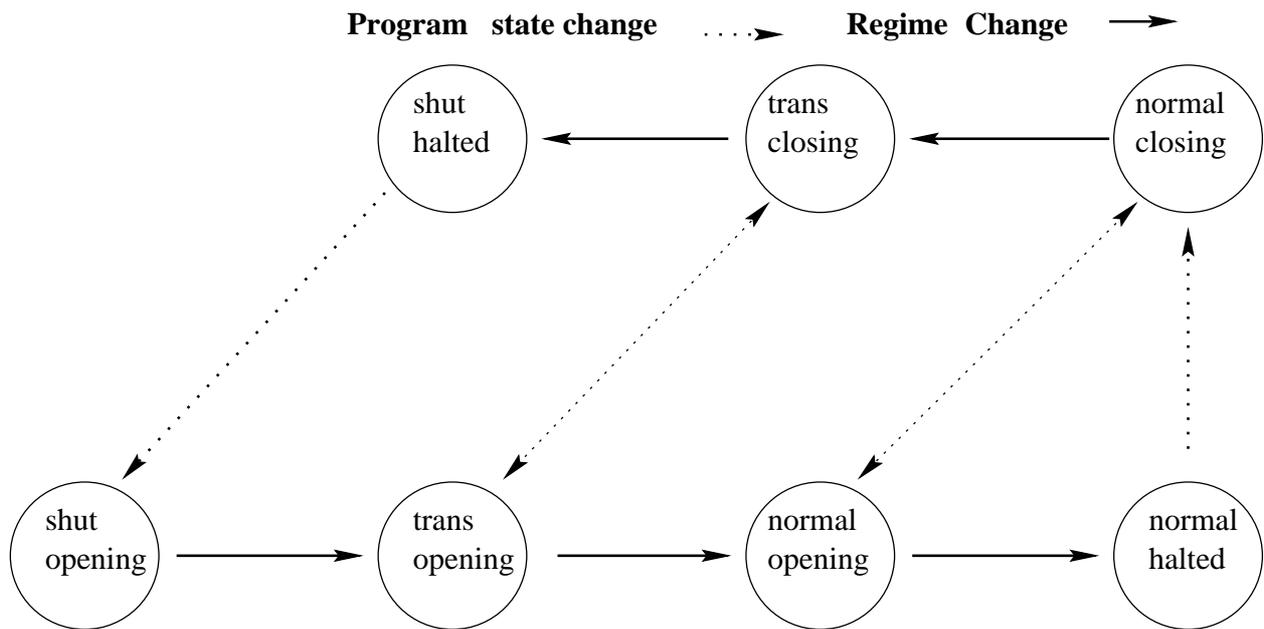


Figure 2. State diagram for ODEs

5. Unavoidable Sources of Error

An important issue in modeling hybrid systems is to realize that almost none of the parameters are known exactly. This is true both of the parameters to the differential equations which describe the system – These parameters are often determined by curve fitting to a set of measured points or are calculated from physical models which include simplifying abstractions, and of measurements taken by sensors in the system – These are measured with some accuracy, which is often specified as an

error-bar.² Because CLP(F) treats everything as an interval, it models these error bars naturally. If a CLP(F) program shows that a system has a safety property (proves that it avoids a region), that proof is valid even when each parameter takes the worst possible value within the given error bars. To deal with this issue by sensitivity analysis (sensitivity analysis conference, 2004; Arsham, ; Taylor, 1997) on the inputs would be extremely difficult.

One of the problems in rigorous modeling is that often there are areas where one's original model breaks down for some reason. This can occur at an area where the physics are unclear, a point where the defining functions are not analytic, or perhaps a function which is poorly defined at a limit point. The point of this paper is that CLIP makes it simple to use hysteresis to deal with all of these problems.

5.1. DEALING WITH POORLY DEFINED REGIONS

Ideally a modeling system allows stepwise refinement of the model. We demonstrate this in CLP(F) by adding valves to our model of the tank flow system. Adding the valves to the model was easy despite using a rather complex model of the valve's behaviour.

One problem which is rarely addressed in modeling hybrid systems is modeling the area around where a component or valve changes state. Using constraints, we can provide a rigorous answer by describing the output of the component while it changes state as being between the output it has in one state and the output it has in the other, and keeping that constraint for however long the component takes to change state. If more precision is required, one can add a description of the component's behaviour during the state transition. Since the description consists of upper and lower bounds for the component's output, one can progressively refine the bounds as one learns more about the component's behaviour.

In many systems, the physics in some regions is not well-understood. Most hybrid system techniques ignore this and simply assume that the ODEs which work in most areas work near boundaries as well. For example, in the tank flow problem when a tank is almost empty, the flow from it may be irregular and come in discrete drops rather than as a continuous flow. At these points, we don't claim to understand the details of the flow, but we can model them rigorously by writing constraints which clearly include any possible behaviour of the flow. We don't consider an empty tank in this case, except to constrain our description of the system to cases in which the water level in each tank is at least E , where E is a negligibly small positive value.

A further problem is that even away from boundary conditions, the physics of the system may not be understood perfectly. In most cases, one measures a value (here, the flow through a valve as a function of how open the valve is) at several points, uses physical theory to decide what form the curve should be (in this case, an exponential of the valve position), and then uses a least-squares fit to find a curve which best describes the measurements. There is, of course, error in the measurement of each point, so the coefficients for the exponential curve have some (hard to calculate) error bars. In addition, the behaviour when the valve is almost closed does not follow the exponential decay curve, and is extremely difficult to measure precisely.

² Note that the problem of imperfect measurement is inherent in the physical world. Heisenberg's uncertainty principle prohibits perfect measurement, and Burridan's principle (Lampert, 1986) further limits the speed at which one can usefully take measurements.

For example Kowalewski *et al.* (Kowalewski et al., 1999) describe a valve by a function $K_i(P)$ giving the pipe coefficient and valve coefficient of the valve as an observed function of how open the valve is. For the first valve, the function they give (converted to our notation) is:

$$K_1(P_1) = \begin{cases} 1.85 \cdot 10^{-4} \cdot e^{-3.1 \cdot 1 - P_1^3} \frac{m^{5/2}}{s} & \text{if } 0 < P_1 \leq 1 \\ 0 \frac{m^{5/2}}{s} & \text{if } P_1 = 0 \end{cases}$$

and for the second valve, they give:

$$K_2(P_2) = \begin{cases} 2.26 \cdot 10^{-4} \cdot e^{-5.7 \cdot 1 - P_2^3} \frac{m^{5/2}}{s} & \text{if } 0 < P_2 \leq 80 \\ 0 \frac{m^{5/2}}{s} & \text{if } P_2 = 0 \end{cases}$$

In neither case do they give error bars. The valve position is described by a real number in $[0, 1]$ with 0 corresponding to fully closed and 1 to fully open. Figure 3 shows a graph of R vs. P for valve 1. The curve is an exponential decay, whose value when the valve is almost closed is about 5% of the flow when the valve is wide open, but they define the flow for a fully closed valve as 0. By straightforward calculation, we find that $R_1(1) \approx 1.85 \cdot 10^{-4}$, while $R_1(\varepsilon) \approx 8.570 \cdot 10^{-6}$ (this is about 5% of full flow), and $R_1(0)$ is defined to be zero. It is likely that this is not fully correct, as a discontinuity of that magnitude is not common. We model this discontinuous point by having three constraints for three different regimes. When the valve is fully closed, R is 0. When the valve position is above the transition region, R is given by the ODE above. The interesting case is when the valve position is in the transition region. We model this case with a constraint which says that if the valve position P is near 0 (here we specify < 0.02), R is small. To choose the upper end of R 's range, we choose a value slightly above the calculated value of R at any point in P 's range for that region. The choice of where the transition region ends is somewhat arbitrary.

Figure 4 shows how we rigorously model this system for P near 0. For the part of the curve where the equations are reliable, we enclose the specified curve on each side by the ODE describing the valve. Because the parameters of the ODE are intervals, the value of the function at any point is an interval. In the area where the curve is discontinuous, we use a constraint which includes all possible values the function could take. This introduces some uncertainty into the formal model, but that uncertainty was already present in the description of the physical system. Using constraints makes that uncertainty explicit, and models it rigorously.

5.2. DEALING WITH REGIME CHANGE POINTS

One of the advantages of using CLP(F) is that one can often use one technique to handle multiple issues. In section 5.1 we use separate ODEs, often with rather simple-minded constraints, to deal with regions where the physics is unclear. Here we use a similar system to deal with non-analytic (or even discontinuous) points in an ODE.

When the water level in the lower tank is above the input pipe (in regime **above**), one set of ODEs holds, when the level is below the input pipe (in regime **below**), another set of ODEs holds. We model this by having a regime change at that point. An obvious problem arises: Our model would allow an infinite number of transitions (each taking zero time) between the two states, and therefore never get to calculating the change in water level which would move clearly into one state

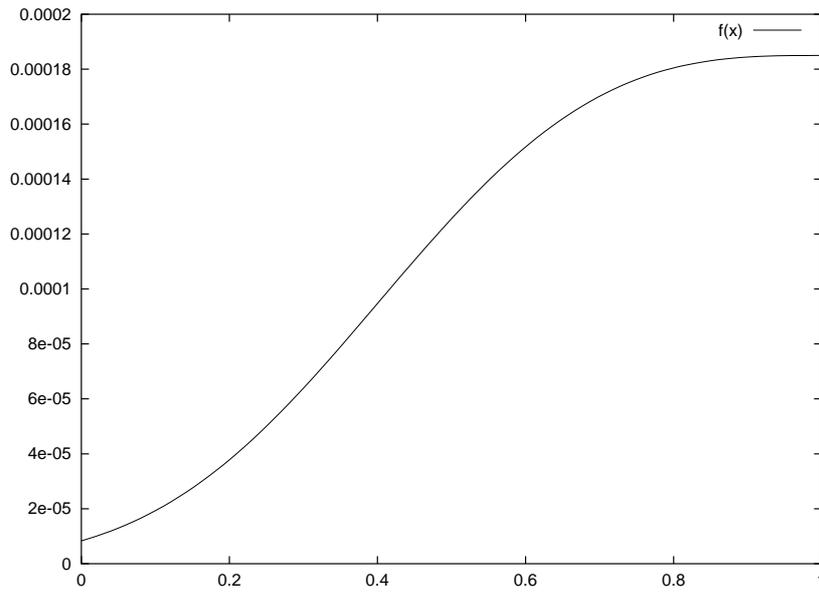


Figure 3. Relative flow as a function of valve position for valve 1 – function from Kowalewski *et al.*

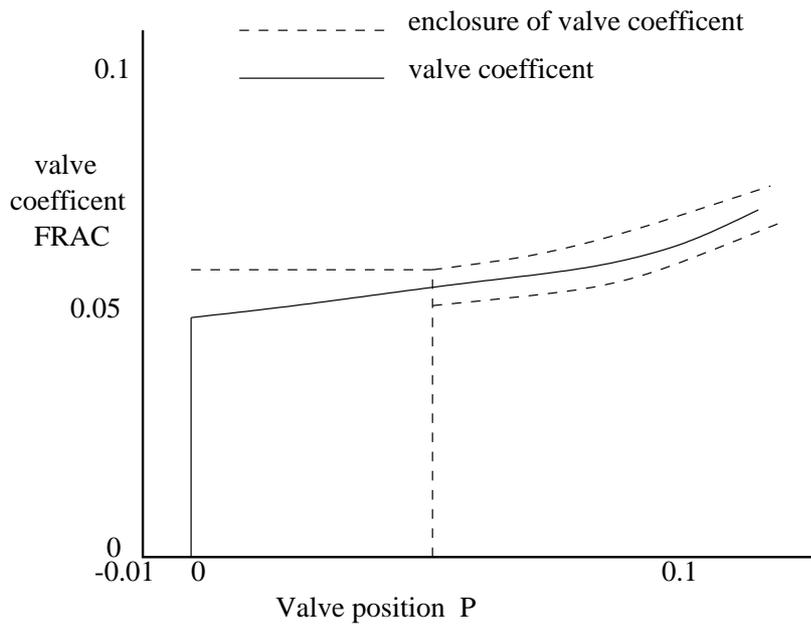


Figure 4. Graph of flow against valve position for valve 1 showing enclosure by simple constraint (scale greatly enlarged)

(Actually, since CLPs are non-deterministic, there would be an infinite path and also a one-step path out of that condition). We handle this by creating a special artificial state **near** for water levels near the boundary. We then artificially put in hysteresis, so that on leaving that middle state, one cannot immediately re-enter it. This problem is related to the problem of Zeno automata, discussed in Section 7.

This example clarifies two issues, as there are two separate reasons for using **near** state between **above** and **below**. The first reason is that as the water in the lower tank reaches the level of the pipe the physics get a little unclear - what happens when the water covers half the pipe? This issue is clearer in the case where the ODEs are discontinuous, as in section 5.1. The second issue is that in order to model a change of ODEs, we need two regimes, with appropriate transitions between them. This issue arises even when the physics are clear, such as when one has a pipe between two tanks and the relative water levels in the two tanks is changing. At the point where the water levels are equal, the ODE is non-analytic (because the square root function is non-analytic at 0), so we would have to have a change of regimes. If the rule for a regime change was simply that the water levels were equal, when the levels became equal there would be a legal infinite path of zero-time changes from one regime to the other. One could look at the derivative to know which direction the regime change goes in, but if the water level is almost constant, the derivative will be near zero, and the same issue is still there. To avoid this case, we artificially add hysteresis to an already artificial regime change.

6. Overview of Code for Tank Flow

The complete program for the $n=4$ case of the tank-flow problem is in the Appendix of (Wittenberg, 2004). Here we discuss some of the more interesting snippets from the code.

6.1. EVOLVE AND ITERATE

We model a hybrid system in CLP(F) by modeling a series of steps. A step begins either at a specified initial state, or when the previous step ends, and ends when either the length of the step (amount of time simulated) reaches a maximum step size `delta`, or a change of ODEs occurs (whether caused by program control or a regime change). The following part of the program is the main code, which runs the system through one step, increments the state counter, and continues.

```
evolve(S0,C,N,S2) :-
    evolve0(S0,C,N,S1),
    enforce_ODEs(S1,C,S2),
    copy_discrete_state(S1,S2).
```

`evolve(S0,C,N,S2)` is true if and only if the system described by `C` can evolve to a boundary state `S1` in `N` steps and then evolve from state `S1` to state `S2`.

```
evolve0(S0,_C,N,S1) :- {N=0},eqstate(S0,S1).
```

```

evolve0(S0,C,N,S2) :-
    opt_next_step(S0,C,S1),
    print(ons(S0,C,S1)),nl,nl,
    {N=M+1},
    evolve0(S1,C,M,S2).

```

A direct reading of the program is as follows: In zero steps, the system does not change state. The `evolve0` predicate says that a system can evolve from `S0` to `S2` if `S1` is the next step from `S0`, $N = M + 1$, and the system can evolve from `S1` to `S2` in `M` steps. The variable `C` in all cases is the set of constants which describe the system parameters.

```

% next_step(InitialState, ProblemConstants, FinalState)
next_step(S0,C,S1) :-
    enforce_ODEs(S0,C,S1),
    find_state_change(S0,C,S1).

```

The call to `next_step` states that the ODEs are followed (`enforce_ODEs`), and finally that system has run to an appropriate point (`find_state_change`). All the variables in the states (`S0`, `S1`) and the constants term (`C`) are variables over the reals. Variables over functions are used in `enforce_ODEs` to specify constraints over the real variables in `S0`, `C`, `S1`.

6.2. FINDING STATE OR REGIME CHANGES

In each case, the step ends when any of the requirements becomes true. Figure 5, shows how `find_state_change` is defined to be true when any one of the following happen:

- One of the `find_flow_state_change` predicates becomes true because the water level in one of the tanks goes from above the input pipe in state `S0` to below in state `S1`, or vice versa (one of the tanks changes regime)
- one of the `find_valve_state_change` predicates becomes true, because the valve position is such that a change in regime occurs at state `S1`
- one of the `find_program_state_change` predicates becomes true because the program (ie. the digital part of the hybrid system) changes state at state `S1`
- `find_step_change` is true because state `S1` is `Delta` time after state `S0` and no other state changes have occurred.

The CLP(F) code to check for this (excerpted in Figure 5 looks rather repetitive. This is true only because in this example we use the same behaviour for each valve and for each tank. In a less symmetric case, this code would not grow, but there might have to be multiple versions of `find_??_state_change` to describe the different behaviours.

```

find_state_change(S0,C,S1) :-
% TANK FLOW REGIME CHANGE
    find_flow_state_change(r1,h1,h2,d2,C,S0,S1);
    find_flow_state_change(r2,h2,h3,d3,C,S0,S1);
    find_flow_state_change(r3,h3,h4,d4,C,S0,S1);

% VALVE REGIME CHANGE
    find_valve_state_change(p1,vr1,vm1,C,S0,S1);
    find_valve_state_change(p2,vr2,vm2,C,S0,S1);
    find_valve_state_change(p3,vr3,vm3,C,S0,S1);
    find_valve_state_change(p4,vr4,vm4,C,S0,S1);

% PROGRAMMED STATE CHANGES
    find_program_state_change(v1,d2,C,S0,S1);
    find_program_state_change(v2,d3,C,S0,S1);
    find_program_state_change(v3,d4,C,S0,S1);

% no regime or state changes before the time limit is reached
    find_step_size_change(S0,C,S1).

```

Figure 5. Code to Find State Changes

6.3. ENFORCING ODES

One section describes all of the analog parts of the system. It consists of three large assertions. The first (and largest) section is purely bookkeeping. All of the ODEs are in the last two parts of `enforce_ODEs`. In order to make the lists of parameters smaller, we use lists to keep all variables of each type together. `lookup`, `evalall`, and `decls` are helper functions to deal with the lists.

The bookkeeping section states that the individual variables correspond to what the lists say they are, and constrains the domain and range of the functions. It uses `lookup` to bind the values of constants (from `C`), and conditions at the start of the step (from `S0`), and the end of the step (from `S1`) to variables. Then it uses `decls` to declare several function variables (and their domains) at once, and finally specifies which ODEs each tank should obey while in the state specified by `S0`. Figure 6 shows sections of the first part of `enforce_ODEs`. Much of that section is repetitive, so only representative fragments are reproduced here. We interpret the code as follows: `enforce_ODEs` is true if and only if all of the following elements are true (including, of course, those that are elided here.)

- `P` is a vector containing `P1,P2,P3,P4`
- each element of `P` is a function defined on $[T0, T1max]$
- each element of `P` is a function whose range is $[0, 1]$

- `Ps0` is a vector containing `P10,P20,P30,P40`
- `Ps1` is a vector containing `P11,P21,P31,P41`
- the values of `Ps0` are as specified in `C` in state `S0`
- the values of `Ps1` are as specified in `C` in state `S1`
- `evala11` applied to each element in the list `P` evaluated at `T0` gives the corresponding value from `Ps0`, and when evaluated at `T1` gives the corresponding value from `Ps1`
- the value of `v` in the list of constants `C` is `V`
- each of the valves obeys `valve_ODE` given the valve position, velocity, and motion regime
- each of the tanks obeys the appropriate tank ODE

Another section handles the flow restriction caused by the valves.

```

valve_coef(normal,FRAC,P,E) :- {[FRAC=exp(E*((1-P)**3),
                                FRAC in [0,1], P in [0.01,1] ]}.
valve_coef(trans,FRAC,P,_) :- {[ FRAC in [0,0.06], P in [0,0.01] ]}.
valve_coef(shut,FRAC,P,_) :- {[FRAC=0.0*FRAC, P=0*P ]}.

valve_ODE(P,_,halted) :- {[ ddt(P,1) = 0.0*P, P in [0,1] ]}.
valve_ODE(P,V,opening) :- {[ ddt(P,1) = V+0*P, P in [0,1] ]}.
valve_ODE(P,V,closing) :- {[ ddt(P,1) = NV + 0*P, P in [0,1],
                               NV= - V ]}.

```

The ODE code is completely straightforward, as ODEs can be described directly in CLP(F). `FRAC`, `E`, and `P` are all functions of `T`. The first line says that in the valve regime `normal`,

$$\text{FRAC} = e^{E \cdot (1-P)^3}, \quad \text{FRAC} \in [0, 1], \quad P \in [0.01, 1]$$

The second line says that in valve regime `trans` `FRAC` $\in [0, 0.06]$ and `P` $\in [0, 0.01]$. The third line says that flow through a shut valve is 0. The idiom `FRAC=0.0*FRAC` is a workaround used instead of `FRAC=0` because CLIP does not allow functions to be set equal to a constant. The second line of the code is needed to implement the technique of rigorously modeling discontinuous functions discussed in section 5.1. Observe that this procedure constrains `P` to take values inside the appropriate region (for `normal`, `trans`, `shut`).

Similarly, the last three lines specify the derivative of `P` (the valve position) to be 0 when halted, `V` for opening, and `-V` for closing.

The last assertions in the ODE section specify the flow into and out of tanks. There are seven cases, as the first and last tanks have different configurations than tanks in the middle, and for all but the last tank, the ODEs differ according to which regime the tanks is (among `below`, `near`, and `above`) corresponding to whether the water level in the lower tank is above or below the pipe entering the lower tank. We consider the case of a middle tank in regime `above`, as that is the most complex.

```

enforce_ODEs(S0,C,S1) :-
...

% VALVE Position ODEs
% create valve position functions on [T0,T1max]
P=[P1,P2,P3,P4],
decls(P,function(T0,T1max)),
% put bounds on the range of the function
bound_functions(P,[0,1]),
% set their values at times T0 and T1
Ps0=[P10,P20,P30,P40],
Ps1=[P11,P21,P31,P41],
lookup([p1=P10,p2=P20,p3=P30,p4=P40],S0),
lookup([p1=P11,p2=P21,p3=P31,p4=P41],S1),
evalall(P,T0,Ps0), evalall(P,T1,Ps1),
% lookup the valve speed
lookup([v=V],C),
% add the ODE constraints
valve_ODE(P1,V,M1),
valve_ODE(P2,V,M2),
valve_ODE(P3,V,M3),
valve_ODE(P4,V,M4),
...

% apply the ODEs corresponding to each tank
% Ri = ode governing tank i, Di = depth in tank i,
% Fi = flow out of tank i, Hi = height of tank i,
% Pi = valve opening out of tank i, Ki = valve coefficient,
% F00 = flow into tank 1

first_tank( R1,    D1,F1,D2,  F00,H1,C1,FRAC1,H2,E),
middle_tank(R2,  F1,D2,F2,D3,    H2,C2,FRAC2,H3,E),
middle_tank(R3, F2,D3,F3,D4,    H3,C3,FRAC3,H4,E),
last_tank(      F3,D4,F4,          C4,FRAC4).

```

Figure 6. Parts of Enforce ODEs code

```
middle_tank(above,F1,D2,F2,D3,H2,C2,FRAC2,H3,_E) :-
  {[F2=C2*FRAC2*psqrt(D2-D3+H), ddt(D2,1)=F1-F2, H=H2-H3,
    D3 in [H,1000] ]}.
```

This says that given a middle tank 2 (middle tank here means that tank 2 is not the first tank, and tank 3 is not the last tank) in regime `above` with the following parameters:

F1 flow into the upper tank
 D2 water height of the upper tank
 F2 flow out of the upper tank (into the lower tank)
 D3 water height of the lower tank
 H2 height of the upper tank above sea level
 C2 parameter of flow through the pipe between upper and lower tanks
 FRAC2 fraction of the maximum flow the valve allows
 H3 height of the lower tank above sea level
 _E an error term (the underscore before the E means ignore this term .)
 then:

$$F2 = C2 \cdot \text{FRAC2} \sqrt{D2 - D3 + H}, \quad \frac{dD2}{dT} = F2 - F1, \quad H = H2 - H3, \quad D3 \in [H, 1000]$$

Here H2, C2, H3 and E are constants, and all the other variables are function variables, though that must be implied from earlier declarations. Again note that the the constraint requires the depth D3(T) to be in the region for the `above` case or on the boundary with another case. Note how the ODEs translate directly into CLIP.

6.4. FINDING STATE CHANGES

The last section of code we describe in detail determines that a regime change has occurred. Parts of this code are in Figure 7 and Figure 8. This code is called from `find_state_change` which says that `find_state_change` is true if at least one of `find_flow_state_change`, `find_valve_state_change`, `find_program_state_change` or `find_step_change`, is true.

`find_flow_state_change` (Figure 7) is true if and only if the two lookup assertions are true, `update_discrete_state` is satisfied, and `flow_state_change` is satisfied. The lookup assertions state that the values of constants passed to the assertion match the constants stored in C. `update_discrete_state` here states that the only difference in discrete variables between state S0 and state S1 is that in state S0, Ri has value R.before and in state S1, Ri has value R.after.

`flow_state_change` lists the four possible transitions, and the water levels which allow them. Note the hysteresis – to enter state `near` the water level has to be within E of the critical level (H1 – H2), while to leave state `near` the water level has to be 2*E away from the critical level. This is to prevent an infinite sequence of zero-time transitions when the water level is at a critical point.

Figure 8 shows the code for changes in the valve's regime. `find_valve_state_change` is very similar to the code for `find_flow_state_change`, except that it twice calls `update_discrete_state` to update the two ternary variables for the two sets of regimes a valve has. One (M) is the valve

```

% Detection of regime change due to tank depth exceeding input
% pipe height e.g. find_flow_state_change(r2,h2,h3,d3,C,S0,S1).
% note j=i+1 here and Ri in {above,near,below}

find_flow_state_change(Ri,Hi,Hj,Dj,C,S0,S1) :-
    lookup([Hi=H1,Hj=H2],C), lookup([Dj=D],S1),
    update_discrete_state(Ri,R_before,R_after,S0,S1),
    flow_state_change(R_before,D,R_after,H1,H2).

% We use hysteresis in our analysis to avoid an infinite loop of zero
% time state changes as it goes from near to below and back again.
flow_state_change(below,D,near,H1,H2) :-
    E=0.00001, {D = H1-H2-E}.
flow_state_change(near,D,below,H1,H2) :-
    E=0.00001, {D = H1-H2-2*E}.
flow_state_change(above,D,near,H1,H2) :-
    E=0.00001, {D = H1-H2+E}.
flow_state_change(near,D,below,H1,H2) :-
    E=0.00001, {D = H1-H2+2*E}.

```

Figure 7. Code for Regime Change as Water Level Changes

motion regime, which can be one of **opening**, **halted**, **closing**, the other (**R**) is the valve position regime, which can be one of **shut**, **trans**, **norm**. The valve position regime is necessary because of the discontinuity in the valve ODEs at zero. **norm** means that the valve is in the regime where the standard ODE applies, **shut** means that the valve is fully closed, and there is no flow through it, and **trans** is the transition regime, where we simply apply a coarse constraint because we don't understand the physics in that regime.

7. Zeno hybrid systems

Johansson *et al.* (Johansson et al., 1999) introduce what they call a “Zeno phenomenon”. This is a problem with some hybrid models in which an infinite number of steps occur in a finite amount of time. At best, this leads to calculations which never finish, while at worst, it leads to false proofs of safety properties in systems which don't have those properties. The canonical examples of Zeno phenomena are a bouncing ball which with each bounce achieves some fraction of the height of the previous bounce in a fixed fraction of the time, and a water tank example discussed below. In the bouncing ball case, a simulation would have to calculate an infinite number of bounces before terminating unless the model included some handling of the idea that when the height of each bounce is less than one atom's diameter, the model must change.

The water tanks example of Johansson *et al.* is shown in Figure 9. There is a flow of water i into a valve which can direct the water into either of two tanks. Each tank has a water level (h_1 , h_2),

```

% check to see if valve n has hit a state change
% and if so, update the discrete part of S1 accordingly
% e.g. find_valve_state_change(p2,v2,C,S0,S1).
% v2 in {opening,closing,halted}, p2 in [0,1],
% note that this is a regime change, not a state change.
% Also, we have to handle the regime change from shut to transition
% to normal. The transition to shut implies a transition to halted,
% but not vice versa.

find_valve_state_change(Pn,Rn,Mn,_C,S0,S1) :-
% use S2, as temp states to have 2 discreet vars change
  lookup([Pn=P_before],S0),lookup([Pn=P_after],S1),
  update_discrete_state(Mn,M_before,M_after,S0,S2),
  update_discrete_state(Rn,R_before,R_after,S2,S1),
  valve_state_change(M_before,R_before,P_before,M_after,
                    R_after,P_after).

% regime change rules for valve motion (and in closing case, position)
valve_state_change(opening,normal,_P_before,halted,normal,P_after) :-
  {P_after=1}.
valve_state_change(closing,trans,_P_before,halted,shut,P_after) :-
  {P_after=0}.

% regime change rules for valves position
valve_state_change(opening,trans,_P_before, opening,normal,P_after) :-
  {P_after=0.01}.
valve_state_change(opening,normal,_P_before,constant,normal,P_after):-
  {P_after=1.0}.
valve_state_change(opening, shut,_P_before, opening,trans,P_after) :-
  {P_after=0.0}.
valve_state_change(closing, normal,_P_before,closing,trans,P_after) :-
  {P_after=0.01}.
valve_state_change(closing,trans,_P_before, constant,shut,P_after) :-
  {P_after=0.0}.

```

Figure 8. Code for Valve Regime Changes

and a required level (r_1, r_2). The safety property is that h_1 is always above r_1 , and h_2 is always above r_2 . The water flow out of each tank is proportional to the ratio of the area of the tank to the area of the output pipe, and to the square root of the water height. If the input flow i is chosen to be larger than either output flow o_1 or o_2 , but less than their sum (when h_1 and h_2 are near r_1 and r_2), it is clear that the level in at least one of the tanks must fall below its required level. Consider the program which whenever one of tanks gets to its required level switches the flow to that tank. As the water level gets lower, the switching will happen more and more often, and the valve will switch an infinite number of times in a finite period, during which time the water level in each tank will still be at or above the required level.

Johansson *et al.* note that the Zeno phenomenon usually occurs as a result of over abstraction in the model, as happens in these cases. Real systems can have valves that chatter, but the chattering cannot involve an infinite number of state changes in a finite time. If the real system has chatter, one should model it by a constraint giving a minimum time for a valve to change state. The infinite chattering is an artifact of some models, and should be removed by the modeler. Zhang *et al.* (Zhang *et al.*, 2001) give examples of cases where overly abstract models (with the Zeno property) of real systems (without the Zeno property) lead to incorrect proofs of safety properties. In most cases, the Zeno problem can be eliminated by a more accurate model, often by simply modeling the time a valve or switch takes to change state. In our example, we avoid Zeno phenomena because there is a lower bound on the time required for twelve consecutive state changes. This bound is implied in different ways for different sets of state changes. For example, the water flow through any pipe is proportional to the square root of the water height, and we bound the water height in each tank. That limit on the water flow limits how quickly the water level in any tank can change. The only code added to avoid Zeno phenomena is the hysteresis. One case in which we do not avoid Zeno phenomena is if the discrete part of a hybrid automata describes a Zeno phenomena. If, for example, the program specified that at some water level a valve would switch from open to closed and from closed to open, that behaviour would be modeled, and the simulation might never finish. There is nothing to be done here. If a user specifies a poorly-formed program, analysis may fail.

7.1. CLP(F) AND ZENO SYSTEMS

How does a CLP(F) model handle a Zeno system? Consider the bouncing ball first. If the modeler does not note that the physics change for very small bounces, the simulation has to include an infinite number of vanishingly small bounces, but because everything in CLP(F) is an interval, the height of the bounce will at some point reduce to $[0, S]$, where S is the smallest number representable in the floating point system. CLP(F)'s non-determinism means that it should eventually explore the path where the bounce height is 0, and the motion ends. Because CLP(F) currently uses depth-first search, it is non-deterministic whether it will try the finite or the infinite path at each branch. If CLP(F) were to use breadth-first search, it would clearly show the possibility that the motion ended, while still modeling the Zeno execution as another possibility. This is probably the best one can hope for. If one gives a computer a model which includes a Zeno execution, the model must show that. If the model can also show that the behaviour is within measurement (or calculation) error, one hopes the user will realize that the initial model is insufficiently defined.

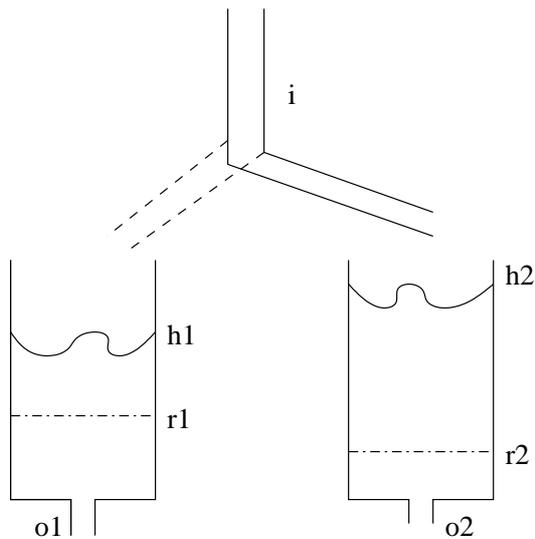


Figure 9. Flow system with Zeno behaviour (after Zhang *et al.*)

References

- Alur, R., C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine: 1995, 'The Algorithmic Analysis of Hybrid Systems'. *Theoretical Computer Science* **138**, 3–34.
- Arsham, H., 'Sensitivity Analysis'. A collection of recent developments on sensitivity analysis in several fields, <http://ubmail.ubalt.edu/~harsham/senanaly/SenAnaly.htm>.
- Benhamou, F. and W. J. Older: 1997, 'Applying Interval Arithmetic to Real, Integer, and Boolean Constraints'. *Journal of Logic Programming* **32**(1), 1–24.
- Cleary, J.: 1987, 'Logical arithmetic'. *Future Computing Systems* **2**, 125–149.
- Davoren, J. and A. Nerode: 2000, 'Logics for Hybrid Systems'. *Proceedings of the IEEE* **88**(7), 985–1010.
- de Bakker, J., C. Huizing, W. de Roever, and G. Rozenberg (eds.): 1991, 'Real-Time: Theory in Practice. REX Workshop', Vol. 600 of *Lecture Notes in Computer Science*. Mook, The Netherlands: REX (Research and Education in Concurrent Systems, Springer Verlag).
- Delzanno, G. and A. Podelski: 1999, 'Model Checking in CLP'. In: R. Cleaveland (ed.): *Proceedings of the Fifth International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS '99)*, Vol. 1579 of *Lecture Notes in Computer Science*. pp. 223–239.
- Delzanno, G. and A. Podelski: 2001, 'Constraint-based Deductive Model Checking'. *International Journal on Software Tools for Technology Transfer (STTT)* **3**(3).
- Deransart, P., A. Ed-Dbali, and L. Cervoni: 1996, *Prolog: The Standard; Reference Manual*. Springer Verlag.
- Diaz, D.: 2002, 'GNU Prolog Manual'. 1.7 edition.
- Fahrland, D. A.: 1970, 'Combined discrete event continuous systems simulation'. *Simulation* **14**(2), 61–72.
- Gupta, V., R. Jagadeesan, and V. Saraswat: 1996, 'Hybrid cc, Hybrid Automata and Program Verification'. In: R. Alur, T. A. Henzinger, and E. D. Sontag (eds.): *Hybrid Systems III: Verification and Control*, Vol. 1066 of *Lecture Notes in Computer Science*. pp. 52–63.
- Gupta, V., R. Jagadeesan, V. Saraswat, and D. G. Bobrow: 1995, 'Programming in hybrid constraint languages'. In: P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry (eds.): *Hybrid Systems II*, Vol. 999 of *Lecture Notes in Computer Science*. pp. 226–251.
- Henzinger, T. A.: 1996, 'The Theory of Hybrid Automata'. In: *Proceedings, 11th Symposium on Logic in Computer Science (LICS '96)*. pp. 278–292.

- Henzinger, T. A., B. Horowitz, R. Majumdar, and H. Wong-Toi: 2000, ‘Beyond HYTECH: Hybrid Systems Analysis Using Interval Numerical Methods’. In: N. Lynch and B. H. Krogh (eds.): *Hybrid Systems: Computation and Control (HSCC 2000)*, Vol. 1790 of *Lecture Notes in Computer Science*. pp. 130–144.
- Hickey, T. J.: 2000, ‘CLIP: A CLP(Intervals) Dialect for Metalevel Constraint Solving’. In: E. Pontelli and V. S. Costa (eds.): *Practical Aspects of Declarative Languages: PADL 2000*, Vol. 1753 of *Lecture Notes in Computer Science*. pp. 200–214. A later version is (Hickey, 2001).
- Hickey, T. J.: 2001, ‘Metalevel Interval Arithmetic and Verifiable Constraint Solving’. *Journal of Functional and Logic Programming* **2001**(7). <http://danae.uni-muenster.de/lehre/kuchen/JFLP/articles/2001/S01-02/JFLP-A01-07.pdf>.
- Hickey, T. J. and Q. Ju, ‘clip 1.0 A CLP(Intervals) interpreter, based on Sicstus Prolog’. interval.sourceforge.net/interval/prolog/clip.
- Hickey, T. J. and Q. Ju.: 1997, ‘Efficient Implementation of Interval Arithmetic Narrowing Using IEEE Arithmetic’. Technical report, Brandeis University CS Dept. www.cs.brandeis.edu/~tim/narrow_multiply/paper.ps.
- Hickey, T. J. and D. K. Wittenberg: 2004, ‘Rigorous Modeling of Hybrid Systems using Interval Arithmetic Constraints’. In: R. Alur and G. J. Pappas (eds.): *Hybrid Systems: Computation and Control HSCC 2004*, Vol. 2993 of *Lecture Notes in Computer Science*. pp. 402–416.
- Holzbaur, C.: 1995, ‘OFAI CLP(Q,R) Manual’. Austrian Research Institute for Artificial Intelligence, Vienna, 1.3.3 edition. TR-95-05.
- Jaffar, J. and J. Lassez: 1987, ‘Constraint Logic Programming’. In: *Proceedings 14th ACM Symposium on the Principles of Programming Languages*. pp. 111–119.
- Jaffar, J. and M. J. Maher: 1994, ‘Constraint Logic Programming: A Survey’. *Journal of Logic Programming* **19/20**, 503–581.
- Jaffar, J., S. Michaylov, P. J. Stuckey, and R. H. C. Yap: 1992, ‘The CLP(\mathcal{R}) Language and System’. *ACM Transactions on Programming Languages and Systems* **14**(3), 339–395.
- Johansson, K. H., M. Egerstedt, J. Lygeros, and S. Sastry: 1999, ‘On the regularization of Zeno hybrid automata’. *System and Control Letters* **38**, 141–150.
- Kowalewski, S., O. Stursberg, M. Fritz, H. Graf, I. Hoffman, J. Preußig, M. Remelhe, S. Simon, and H. Treseler: 1999, ‘A Case Study in Tool-Aided Analysis of Discretely Controlled Continuous Systems: The Two Tanks Problem’. In: P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, and S. Sastry (eds.): *Hybrid Systems V*, Vol. 1567 of *Lecture Notes in Computer Science*. pp. 163–185.
- Lamport, L.: 1986, ‘Buridan’s Principle’. Revised from a version of Oct. 1984; <http://research.microsoft.com/users/lamport/pubs/buridan.ps>.
- Lynch, N., R. Segala, and F. Vaandrager: 2001, ‘Hybrid I/O Automata Revisited’. In: M. D. D. Benedetto and A. Sangiovanni-Vincentelli (eds.): *Hybrid Systems: Communication and Control*, Vol. 2034 of *Lecture Notes in Computer Science*. pp. 403–417.
- Lynch, N., R. Segala, F. W. Vaandrager, and H. Weinberg: 1999, ‘Hybrid I/O automata’. Technical Report CSI-R9907, Computing Science Institute Nijmegen; Faculty of Mathematics and Informatics; Catholic University of Nijmegen, Toernooiveld 1; 6525 ED Nijmegen; The Netherlands.
- Maler, O., Z. Manna, and A. Pnueli: 1991, ‘From Timed to Hybrid Systems’. in (de Bakker et al., 1991), pp. 447–484.
- Moore, R. E.: 1966, *Interval Analysis*. Prentice-Hall.
- Older, W. and A. Vellino: 1993, ‘Constraint Arithmetic on Real Intervals’. In: A. Colmerauer and F. Benhamou (eds.): *Constraint Logic Programming: Selected Research*. MIT Press.
- Prolog 95: 1995, ‘ISO Prolog Standard ISO/IEC 13211-1, Information Technology — Programming Languages — Prolog — Part 1: General Core’. available from www.iso.org/iso/en/.
- Research, B. N.: 1988, *BNR Prolog user guide and reference manual*. Bell Northern Research.
- sensitivity analysis conference: 2004, ‘Fourth International Conference on Sensitivity Analysis of Model Output’. Santa Fe, New Mexico: <http://www.samo2004.org>.
- Shankar, A. U.: 1993, ‘An Introduction to Assertional Reasoning for Concurrent Systems’. *ACM Computing Surveys* **25**(3), 225–262.

- Stursberg, O., S. Kowalewski, I. Hoffman, and J. Preußig: 1997, 'Comparing Timed and Hybrid Automata as Approximations of Continuous Systems'. In: P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry (eds.): *Hybrid Systems IV*, Vol. 1273 of *Lecture Notes in Computer Science*. pp. 361–377.
- Taylor, J. R.: 1997, *An introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. University Science Books, second edition.
- Urbina, L.: 1996, 'Analysis of Hybrid Systems in CLP(\mathcal{R})'. In: E. C. Freuder (ed.): *Principles and Practice of Constraint Programming – CP96*, Vol. 1118 of *Lecture Notes in Computer Science*. pp. 451–467.
- Wittenberg, D. K.: 2004, 'CLP(F) Modeling of Hybrid Systems'. Ph.D. thesis, Brandeis University. <http://www.cs.brandeis.edu/~dkw/papers/thesis.ps>.
- Zhang, J., K. H. Johansson, J. Lygeros, and S. Sastry: 2001, 'Zeno Hybrid Systems'. *International Journal of Robust and Nonlinear Control* **11**(5), 435–451.

Worst case bounds in the presence of correlated uncertainty

113

Arnold Neumaier

*Fakultät für Mathematik, Universität Wien
Nordbergstr. 15, A-1090 Wien, Austria
email: Arnold.Neumaier@univie.ac.at
WWW: <http://www.mat.univie.ac.at/~neum/>*

February 14, 2006

Extended Abstract

This paper presents a method for computing rigorous bounds on the solution of linear systems whose coefficients have large, correlated uncertainties, with a computable overestimation factor that is frequently quite small.

Linear systems of equations are among the most frequently used tools in applied mathematics. In realistic applications, the data entering the coefficients of these equations are generally uncertain. Since linear equations become nonlinear when coefficients are uncertain and become variable, traditional sensitivity analysis remains valid only for sufficiently small errors. Unfortunately, it is usually unclear when the errors are sufficiently small for its validity: For errors larger than some unknown, problem-dependent margin, sensitivity analysis may be severely biased, since it does not account for the nonlinearities in the problem.

In problems where safety is an issue, worst case results are needed. For example, current safety regulation laws in civil engineering require a worst case analysis, and hence interval techniques, although current practice is still Monte Carlo with its deficiencies.

Recently, NEUMAIER & POWNUK (2) showed that using interval analysis, it is possible to do quantitative worst case sensitivity analysis even in high dimensions. The techniques presented there provide good and valid enclosures of all quantities of interest, and thus enables engineers to obtain guarantees whether the worst case satisfies all safety requirements.

However, previous worst case methods (including monotonicity methods which work only under additional assumptions) only compute the worst case when all uncertainties vary independently. This is frequently an unrealistic assumption. In the past, correlated uncertainties could be handled only with Monte Carlo methods which always underestimate the worst case, and sometimes drastically.

In this paper we develop a method for the worst case analysis of solutions of linear systems of the form

$$(K + A^T D A)u = a + Fb,$$

where D is diagonal, with correlated uncertainties in D and b , and no uncertainty in K , A , a , and F . This includes the case of linear systems arising in truss modeling.

The new results are obtained by generalizing those of NEUMAIER & POWNUK (2) to the case where the uncertainties are bounded by ellipsoids rather than boxes, thus reflecting the known correlations. A basic tool used is the following optimal bound for linear combinations of numbers ranging in an ellipsoid.

Proposition. Let $C \in \mathbb{R}^{n \times n}$ be symmetric and positive definite. If

$$x^T C^{-1} x \leq q$$

then

$$|a^T x| \leq \sqrt{q a^T C a}, \quad (1)$$

with equality iff $x = \lambda C a$ with $|\lambda| = \sqrt{q/a^T C a}$.

Proof. Any symmetric and positive definite matrix C has a Cholesky factorization $C = LL^T$ with nonsingular L . Using this, we have

$$\|L^{-1}x\|_2 = \sqrt{(L^{-1}x)^T L^{-1}x} = \sqrt{x^T C^{-1}x} \leq \sqrt{q},$$

$$\|L^T a\|_2 = \sqrt{(L^T a)^T L^T a} = \sqrt{a^T C a}.$$

Now the Cauchy-Schwarz inequality implies

$$|a^T x| = |(L^{-1}x)^T L^T a| \leq \|L^{-1}x\|_2 \|L^T a\|_2 \leq \sqrt{q a^T C a},$$

giving (1). Equality can only hold if equality holds in the Cauchy-Schwarz inequality, hence if $L^{-1}x$ and $L^T a$ are parallel. This requires $x = \lambda C a$, and by substituting this into the equality case of (1), we find that $|\lambda| = \sqrt{q/a^T C a}$. \square

In addition, we employ ellipsoid arithmetic (cf. NEUMAIER (1)) to enclose the intermediate expressions in the calculations.

As an application, we show that it is feasible to compute worst case error bounds for the displacements of truss structures with uncertain stiffness coefficient, in the important case when the uncertainties are correlated. This includes the discussion of an appropriate deterministic model for uncertainty correlation.

Examples of numerical computations will be given for large truss structures with correlated uncertainty in the stiffness.

References

- A. Neumaier, The wrapping effect, ellipsoid arithmetic, stability and confidence regions, Computing Supplementum 9 (1993), 175–190.
- A. Neumaier and A. Pownuk, Linear systems with large uncertainties, with applications to truss structures, Reliable Computing, to appear.

Modeling Correlation and Dependence Among Intervals

Scott Ferson and Vladik Kreinovich

Applied Biomathematics, Setauket, New York 11733 USA, scott@ramas.com

University of Texas, El Paso, Texas 79968 USA, vladik@utep.edu

Abstract: This note introduces the notion of dependence among intervals to account for observed or theoretical constraints on the relationships among uncertain inputs in mathematical calculations. We define dependence as any restriction on the possible pairings of values within respective intervals and define nondependence as the degenerate case of no restrictions (which we carefully distinguish from independence in probability theory). Traditional interval calculations assume nondependence, but alternative assumptions are possible, including several which might be practical in engineering settings that would lead to tighter enclosures on arithmetic functions of intervals. We give best possible formulas for addition of intervals under several of these dependencies. We also suggest some potentially useful models of correlation, which are single-parameter families of dependencies, often ranging from the identity dependence ($u=v$) representing maximal correlation, through nondependence, to opposite dependence ($1-u=v$) representing maximally negative correlation.

Keywords: dependence, correlation, copula, multivariate interval, nondependence

1. Introduction

Interval analysis has an inadequate model of dependence between variables. Because of this deficiency, many analysts discount the utility of interval arithmetic in propagating uncertainty through mathematical expressions because it does not account for natural dependencies that can occur between input values. Many reject interval methods and appeal instead to probability theory because it provides a well developed model of dependence in terms of correlations and the general theory of copulas (Nelsen 1999). This perceived advantage of probabilistic over interval methods is undeserved, however, because interval analysis *could* also offer a model of dependence, and it would be considerably simpler and perhaps more workable than that required for event probabilities or random numbers.

There are two uses of a model of dependence among intervals. The first is to account for dependencies that exist between distinct inputs. Such dependencies can be implied by the physical or biological mechanisms governing the underlying system. For instance, if both the

© 2006 by authors. Printed in USA

size and mass of a component are interval inputs in a calculation, it is likely there is a connection between these two inputs such that large values of one are associated with large values of the other and that precludes certain contrary combinations of values within their intervals. For other variables there might be reasons why large values of one cannot co-occur with large values of another. Dependencies such as these might be deduced from the mathematical relationships between the variables. They might alternatively be evidenced by empirical information, or simply asserted a priori by the analyst. In any case, it is legitimate and essential to take account of these dependencies if doing so tightens the interval outputs of analysis.

Although not a primary focus of this note, the second use of a model of dependence among intervals is as underpinning for a strategy to address the repeated parameter issue (also known as the “dependence” issue) in which a single interval input appears multiple times within a mathematical expression. For example, the terms in the expression $A - A^2$ are dependent in that knowing A 's value tells us the value of A^2 exactly. Such dependencies arise because of mathematical identities or repeated variables in expressions, rather than empirical dependencies discussed above. One could argue that one kind of dependence is a special case of the other kind of dependence, and they are clearly closely intertwined.

2. Dependence between intervals

So what is dependence between uncertain numbers characterized by intervals? We define dependence as any restriction on the possible pairings of the uncertain numbers. An interval *dependence relation* D is a subset of the unit square $U = [0,1] \times [0,1] = \{(u,v) : u \in [0,1], v \in [0,1]\}$ such that there exists in the relation at least one pair (u,v) for every value of u and v . That is, $D \subseteq U$ is a dependence relation if and only if, for any $u \in [0,1]$, there exists some pair $(u,v) \in D$ for some $v \in [0,1]$, and, likewise, for any $v \in [0,1]$, there is a pair $(u,v) \in D$ for some $u \in [0,1]$. Consider two intervals $A = [a_1, a_2]$ and $B = [b_1, b_2]$. We say that A and B are dependent according to a dependence relation D if

$$f(A, B) = \{c : c = f(a, b), \text{ where } a = u(a_2 - a_1) + a_1, b = v(b_2 - b_1) + b_1, \text{ and } (u,v) \in D\}$$

for all binary functions f . In this case, A and B are said to have the dependence D . Any pair of values (a,b) is called a *possible pair* from the intervals A and B if $a \in A$, $b \in B$, and $((a - a_1)/(a_2 - a_1), (b - b_1)/(b_2 - b_1)) \in D$.

We use \mathcal{D} to denote the set of all such dependence relations, of which $U \in \mathcal{D}$ is a privileged special case. If a dependence relation is all of U , it is called the *noninteractive dependence*

relation or, more simply, the *all-pairs relation*. It is the largest possible dependence in that it encloses all other possible dependence relations. We can say that intervals having this degenerate relation are *nondependent*. (We conscientiously refrain from calling such intervals ‘independent’ because this term already has a firmly entrenched meaning in probability theory that is not equivalent to—and indeed is quite different from—nondependence.)

If there is only one pair in the set for each value of u and only one pair for each v , it is called a *one-pair dependence relation*. There are two special cases of one-pair relations that are especially important. The first is the identity relation $P = \{(u,v) : u = v, u \in [0,1], v \in [0,1]\}$. This is the case of perfect dependence between the two intervals. Low values of one interval are perfectly paired with low values of the other, and high values of one are paired with high values of the other. The second special case of a one-pair relation is the opposite relation $O = \{(u,v) : 1-u = v, u \in [0,1], v \in [0,1]\}$ which reverses the association so high values of one variable are paired with low values of the other. Both of these special cases are monotone relations, but not all one-pair relations are so well behaved. Even if the value within A perfectly determines the associated value within B and vice versa, their dependence may still be very complicated. The notion of “shuffles” (Nelsen 1999) from probability theory generalizes to interval dependence.

Between the degenerate all-pairs dependence relation and various possible one-pair dependence relations there is a huge variety of dependence relations. Indeed, this variety is infinite-dimensional, although it is vastly less complex than the analogous diversity in copulas modeling dependence between random numbers in probability theory. The key to developing practical strategies for handling dependence among intervals is to define classes or families of dependence that are appropriate models of the kinds of associations commonly encountered. The next section introduces some candidates.

3. Correlation models

A *complete* model of correlation is any map ρ from $[-1,+1]$ to \mathcal{D} (the set of all bivariate dependence relations) such that $\rho(-1) = O$, $\rho(0) = U$, and $\rho(1) = P$. There are infinitely many such maps (just as there are in the analogous probability theory). Nevertheless, it is useful to identify some models of correlation that might be workable in practical engineering settings. For instance, it might be convenient to define the family of dependence relations depicted in *Figure 1*. The figure shows eleven dependence relations, ranging from O at the far left to P at the far right. Each dependence relation is depicted as an area in black within the unit square. The abscissas are the u values and the ordinates are the v values.

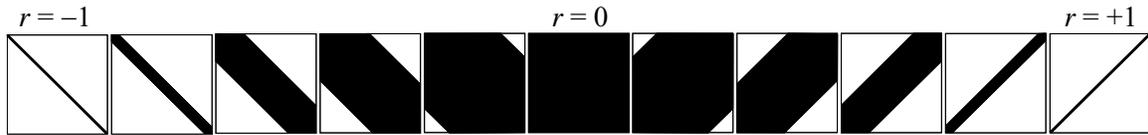


Figure 1. A complete model of correlation between intervals.

In this family, there is a dependence relation for each value of the correlation r from negative one through zero and on to positive one. In this case, the dependence relation for a given r is defined as

$$D(r) = \{ (u,v) : \max(0, -u-r, u-1+r) \leq v \leq \min(1, u+1-r, -u+2+r), u \in [0,1], v \in [0,1] \}.$$

The signal characteristic of the model of correlation represented by this parameterized family of dependence relations is the way in which pairs are excluded that would contradict the assertion of correlation at magnitude r : the counterindicated corners of the dependence relation are shaved away.

There are actually many complete models of correlation that are possible. For example, *Figure 2* shows four different families, each of which smoothly morph from the opposite dependence O for a correlation r of -1 though the all-pairs dependence at correlation zero to the perfect dependence P at correlation $+1$. These families are composed of relations having rhomboidal shapes with straight-line edges. Other families could be devised out of other curved shapes as well.

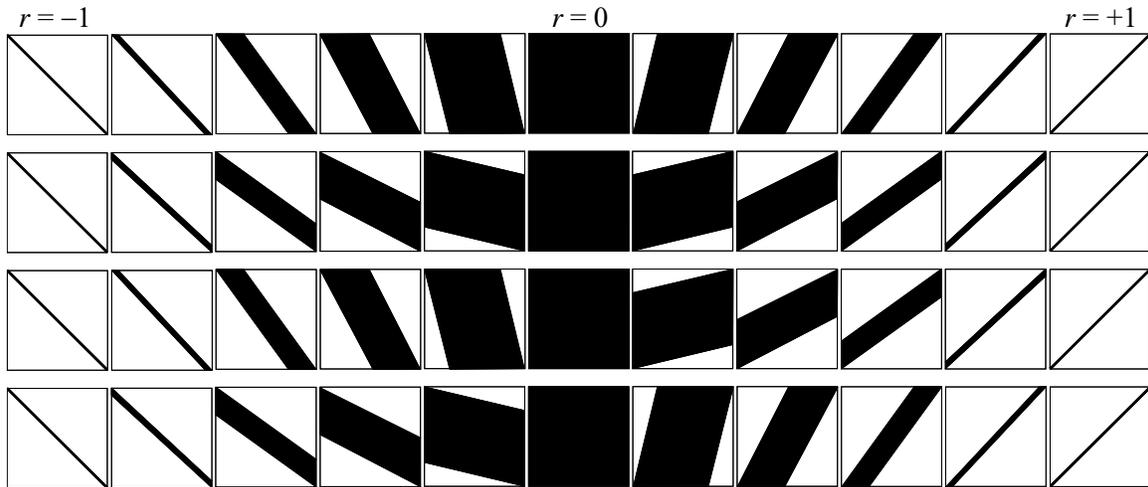


Figure 2. Four alternative complete models of correlation.

A *non-complete* model of correlation would be a map from a proper subset of $[-1,+1]$ to \mathcal{D} , or a map from $[-1,+1]$ to \mathcal{D} that didn't send $-1, 0$ and $+1$ to O, U and P respectively. One non-complete model of correlation that will likely be very useful in practical problems is the ellipse model (Chernousko 1988, 1994; Kreinovich et al. 2005, 2006). This model maps all of the range $[-1,+1]$ and it goes from O and P , but its dependence for zero correlation is not U . Instead, it is the inscribed circle $E_0 = \{(u,v) : (u-1/2)^2 + (v-1/2)^2 \leq 1/4\}$. As the correlation coefficient varies from zero to $+1$, the dependence relation is a rotated ellipse inscribed within U . In the limit, as the correlation reaches $+1$, it becomes a degenerate rotated ellipse equivalent to the perfect dependence relation P . Likewise the negative correlations go from the circle to the opposite dependence O . This family of ellipses is depicted in *Figure 3*. It is parameterized by the point u^* of the ellipse's tangency with the u -axis (where $v = 0$). Because this point ranges over $[0,1]$, we can define another correlation index $r = 1 - 2u^*$, ranging over $[-1,+1]$.

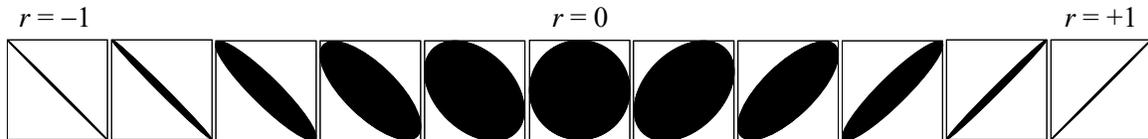


Figure 3. Elliptic family of dependence relations.

Given an elliptic correlation r , the dependence relation is the interior of the ellipse $E(r) \subseteq U$, which is tangent to the u -axis at $u^* = (1-r)/2$. This dependence relation is

$$E(r) = \{(u,v) : 4((u+v-1)^2 - 2(1+r)(u-1/2)(v-1/2))/(1-r^2) \leq 1, u \in [0,1], v \in [0,1]\}.$$

Chernousko (1988, 1994) and Kreinovich et al. (2005, 2006) considered such ellipses for modeling dependence among intervals. Kreinovich et al. (2006) reviewed the use of an elliptic model of interval dependence in quadratic response surface models.

There are many, many other dependence families that might be useful. When, for example, an interval expression involves repeated subexpressions inducing a mathematical dependence, the relevant family of dependence relations represents the mathematical relationship. Consider, for instance, intervals A and A^2 . Depending on the numerical values within A , their dependence must be an arc of a parabola and might be one of the dependence relations depicted in *Figure 4*.

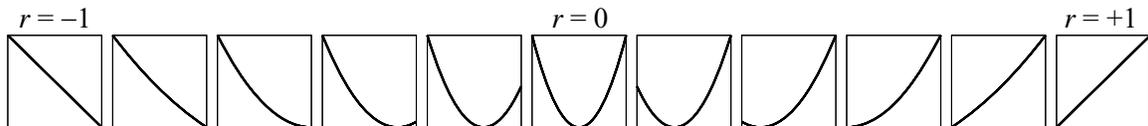


Figure 4. Parabolic family of dependence relations.

These dependencies can be called the parabolic family of dependence relations. A parameterization of the family is

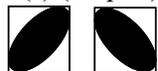
$$Q(r) = \{(u,v) : ((u - \lambda)^2 - q) / (\max(\lambda^2, (1 - \lambda)^2) - q) = v, u \in [0,1], v \in [0,1]\}.$$

where $\lambda = \tan(-\pi r/2) + 1/2$ is the location on the u -axis of the parabola's minimum, and q is zero if $0 \leq \lambda \leq 1$, or $\min(\lambda^2, (1-\lambda)^2)$ otherwise. Some of these dependencies are one-pair relations (when they represent only one branch of the parabola), in which case calculations may be relatively easy, but this is not always so. Because the dependence relation is scaled on the unit square, this family of dependences can be parameterized by a single-dimensional scalar value that depends on whether the interval A straddles zero or not.

In principle, other intervals could have parabolic dependence as well. For instance, the interval $B = [4,11]$ could not be a square of the interval $A = [3,5]$ because their ranges would be inconsistent, but these two intervals could have a parabolic dependence if the pairings of $a \in A$ and $b \in B$ were constrained so that $((a - 3)/2, (b - 4)/7) \in Q(r)$ for some r as depicted in Figure 4.

4. Arithmetic operations under specified dependence

Accounting for the dependence between intervals can improve the enclosures that can be computed for arithmetic expressions that involve them, and the numerical results can be considerably tighter than would be obtained by applying the default methods of interval arithmetic that do not consider dependence. The table below gives formulas for the sum of $A = [a_1, a_2]$ and $B = [b_1, b_2]$ under a variety of dependence relations between them. On the left side of the table are given the name of the dependence relation, a graphical depiction of its shape and the constraints that define it (in terms of u and v , which are each implicitly assumed to lie within $[0,1]$). On the right side of the table are formulas to compute best-possible bounds on the sum $A+B$. Some of the formulas involve the envelope function $\text{env}(x,y) = [\min(x, y), \max(x, y)]$, and the proportional component function $w([x_1,x_2], p)$ which is $p(x_2 - x_1) + x_1$, or just x_1 if p is less than zero, or x_2 if p is greater than one.

Dependence	Addition formula
P (perfect)  $u = v$	$[a_1+b_1, a_2+b_2]$
O (opposite)  $1 - u = v$	$\text{env}(a_1+b_2, a_2+b_1)$
D(r) (correlated)  $\max(-u-r, u-1+r) \leq v \leq \min(u+1-r, -u+2+r)$	$[\text{env}(w(A, -r)+b_1, a_1+w(B, -r)), \text{env}(a_2+w(B, 1+r), w(A, 1+r)+b_2)]$
E(r) (elliptic)  $4((u+v-1)^2-2(1+r)(u-\frac{1}{2})(v-\frac{1}{2}))/((1-r^2)^2) \leq 1$	$\text{env}(p-q^-, p-q^+, -p+q^+, -p+q^-)+(x_1+x_2+y_1+y_2)/2$, where $p = \sqrt{4z}/(((y^2-4xz)/(y-2z))^2-y^2+4xz)$, $q^\pm = yp \pm \sqrt{y^2p^2-4z(xp^2-1)}/2z$, $x = 4/(a_2-a_1)^2(1-r^2)$, $y = -8/(a_2-a_1)(b_2-b_1)(1-r^2)$, $z = 4/(b_2-b_1)^2(1-r^2)$
Upper, left  $u \leq v$	$[a_1+b_1, a_2+b_2]$
Lower, left  $1 - u \geq v$	$\text{env}(a_2+b_1, \text{env}(a_1+b_2, a_1+b_1))$
Upper, right  $1 - u \leq v$	$\text{env}(a_2+b_1, \text{env}(a_1+b_2, a_2+b_2))$
Lower, right  $u \geq v$	$[a_1+b_1, a_2+b_2]$
Diamond 	$[\text{env}(a_1+w(B, \frac{1}{2}), w(A, \frac{1}{2})+b_1), \text{env}(a_2+w(B, \frac{1}{2}), w(A, \frac{1}{2})+b_2)]$

$ u - \frac{1}{2} + v - \frac{1}{2} \leq \frac{1}{2}$	
U (nondependent)  (u,v)	$[a_1+b_1, a_2+b_2]$

This and comparable tables for other arithmetic operations such as subtraction, multiplication, division, minimum, maximum, powers, etc., together would constitute an extension to naïve interval arithmetic that can begin to account for dependence between inputs.

The table above gives formulas for single arithmetic sums. For example, suppose the dependence relation between $A = [0,1]$ and $B = [1,11]$ is of the form $D(r = -0.5)$ as depicted in *Figure 1*, then the sum $A + B$ is surely within $[\text{env}(w(A, -r)+b_1, a_1+w(B,-r)), \text{env}(a_2+w(B,1+r),w(A,1+r)+b_2)] = [\text{env}(w([0,1], 0.5)+1, 0+w([1,11], 0.5)), \text{env}(1+w([1,11],1-0.5),w([0,1],1-0.5)+11)] = [\text{env}(0.5+1, 0+6), \text{env}(1+6, 0.5+11)] = [[1.5,6], [7,11.5]] = [1.5, 11.5]$. This interval is an improvement to both bounds over $[1,12]$ obtained by standard interval analysis that does not consider their dependence. The bounds accounting for this kind of dependence will be tighter than $[a_1+b_1, a_2+b_2]$ whenever r is less than zero. Another example involves a special case of the $D(r)$ dependence family which is the opposite dependence relation $O = D(-1)$. If A and B have this dependence, then their sum $A+B$ is sure to be within $[2,11]$. The tighter result arises in this case because the possible pairs of values from the two intervals are restricted to single combinations:

$a \in A$	$b \in B$	$a+b$
0	11	11,
⋮	⋮	⋮
0.1	10	10.1,
⋮	⋮	⋮
0.2	9	9.2,
⋮	⋮	⋮
0.3	8	8.3,
⋮	⋮	⋮
0.8	3	3.8,
⋮	⋮	⋮
0.9	2	2.9,
⋮	⋮	⋮
1	1	2.

For this reason, the formula for addition under opposite dependence simplifies to $\text{env}(a_1+b_2, a_2+b_1)$ as shown in the table.

There is an important caveat about the difficulty of the deriving formulas for arithmetic functions for different dependence relations. Monotonicity of the dependence relation does not ensure that the bounds on an arithmetic function can be found by testing two endpoints. Consider bounding the addition of intervals $A = [3,5]$ and $B = [4,11]$ that have a parabolic dependence defined by the constraint $(u - 1)^2 = v$. This corresponds to $\lambda = 1$, $q = 0$, and $r = -2 \operatorname{atan}(1/2)/\pi \approx -0.295$ and is the left branch of a parabola, so the dependence is monotone. (It is depicted as the decreasing curve in the third graph from the left in Figure 4.) The endpoints of the dependence relation might seem to suggest that the bounds on the sum would be $a+b = 3+11 = 14$ and $a+b = 5+4 = 9$. But the minimal value of the sum is actually obtained from the combination of $a = 4^{5/7}$ with $b = 4^{1/7}$, which is $8^{6/7} \approx 8.857$. The values correspond to $u = 6/7$ and $v = 1/49$. This example shows that even when the dependence is a one-pair relation that is a monotone function, even the simplest arithmetic function, addition, cannot be evaluated by enveloping the results at the endpoints. Inspection of the endpoints or corners of the dependence relation only generally suffices to find the bounds on the arithmetic function if the edges of the dependence relation are straight lines and the arithmetic function is addition.

Accounting for dependence can sometimes lead to substantial numerical improvements over interval calculations that make no account of dependence. Although they are generally modest for addition, they can be large for other mathematical operations. For instance, if $A = [0,1]$ and $B = [1,11]$ have the opposite dependence relation O , the range of their product $A \times B$ is $[0,3.025]$, which is only a third of the width of the interval $[0,11]$ obtained by the standard calculation.

5. Uncertainty about the dependence

Specifying the dependence relations between input intervals is the prerogative and responsibility of the analyst. They should represent available information about constraints between the inputs. Because the specification of such dependencies is a matter of engineering judgment or empirical evidence, there may be uncertainty about how it should be done. In particular, for instance, one may not be able to ascribe a precise value to a correlation coefficient r . In such cases, it might be reasonable to use an interval to characterize r . The bounds on an arithmetic function of intervals in this case can be found by taking the union (or convex hull) of bounds obtained under each possible correlation coefficient within the interval.

When one does not know anything about the dependence at all, the all-pairs dependence relation we call nondependence should be used. This reduces all arithmetic calculations to the traditional interval formulas. This choice allows an analyst to compute *conservative* answers that enclose all possible results. Such a simple strategy is not available in probability theory. Assuming independence (or, indeed, any dependence) between random variables would not allow one to

find the bounds on an arithmetic function when their dependence is unknown. To do this, one must resort to computing the Fréchet convolutions (Ferson et al. 2004). This difference shows that nondependence is not really analogous to independence as it is recognized by probabilists. Although interval researchers often refer to nondependence as independence, and nondependence has sometimes been considered a kind of independence (Couso et al. 2000; cf. Ferson et al. 2004), we think that they are such distinct ideas that special care should be made to distinguish between them.

6. Multivariate dependence relations

So far, we have discussed only bivariate dependence relations, but there are multivariate generalizations as well. For example, $D_3 \subseteq [0,1] \times [0,1] \times [0,1]$ is a trivariate dependence relation if it contains at least one triple for every marginal value. Likewise, $D_k \subseteq [0,1]^k$ is a k -dimensional dependence relation if it contains at least one element for every marginal value. We can denote the set of all possible k -dimensional dependence relations as \mathcal{D}_k . We have been calling \mathcal{D}_2 simply \mathcal{D} . There is a k -dimensional generalization of P, but not of O.

The problem of accounting for dependencies among intervals in complex mathematical expressions may be much more difficult than it is for the binary operations considered in this note. Strategies for conveniently calculating best possible bounds await development. It may be difficult to properly handle such calculations as a sequence of binary operations on intervals. For example, suppose $A = [2,4]$, $B = [4,7]$, and $C = [3,9]$, where A and C have the opposite dependence relation O, and that the mathematical expression to be evaluated is $AB+C$. Approached as a composition of binary operations, the calculation would need to evaluate AB first and only then the sum. However, the information about the dependence between A and C is inaccessible once the multiplication occurs. What does dependence information about two variables imply about the dependence between functions of these variables? Simple simulations show that the best possible bounds on the function $AB+C$ given the opposite dependence between A and C are $[17,31]$. This interval can be obtained by assuming opposite dependence between C and the product AB , but it is not clear that assumptions like this are always permissible, or, in general, what theory governs dependence in interval calculations.

7. Conclusions

This paper has introduced the notion of dependence within interval calculations. Dependence is defined to be any restriction on the possible pairings of values from the respective intervals. Such

restrictions can be modeled as subsets of the unit square, which are relations (rather than functions) between the margins of a multivariate interval. As copulas abstract the notion of dependence out of joint distributions in probability theory, these structures extract the dependence out of multivariate intervals.

We have derived some exemplary formulas for bounding the results of interval addition under a handful of possible dependence relations, but the general computational problem of accounting for dependencies among intervals in arbitrary interval computations remains largely unstudied. Dependence information about two interval variables does not necessarily imply the dependence between functions of these variables. Further work is necessary to develop and implement convenient algorithms to enable routine calculations that take account of dependence among intervals. Further work is also needed to explore the role that conditionalization might play in the context of interval dependence.

Acknowledgments

Troy Tucker kindly read this note and offered several suggestions. Lev Ginzburg originated our concept of global correlation for interval uncertainty and derived the bounds on the sum under elliptic dependence. Work on this note began in reaction to a question posed by Bill Walster at the Second Scandinavian Workshop on Intervals and Their Applications in Copenhagen. This work was supported by Sandia National Laboratories through contract 19094, a part of Sandia's Epistemic Uncertainty Project directed by William Oberkampf, and the National Institutes of Health through SBIR grant 5R44ES010511-03. The opinions herein are those of the authors only.

References

- Chernousko, F.L. 1988. *Estimation of the Phase Space of Dynamical Systems*. Nauka Publishers, Moscow, (in Russian).
- Chernousko, F.L. 1994. *State Estimation for Dynamical Systems*. CRC Press, Boca Raton, Florida.
- Couso, I., D. Moral and P. Walley. 2000. A survey of concepts of independence for imprecise probabilities. *Risk Decision and Policy* 5: 165-181.
- Ferson, S., W. Troy Tucker and W.L. Oberkampf. 2004. The notion of independence when probabilities are imprecise. 9th ASCE EMD/SEI/GI/AD Joint Specialty Conference on Probabilistic Mechanics and Structural Reliability (PMC2004), Albuquerque, New Mexico.

- Ferson, S., R.B. Nelsen, J. Hajgos, D.J. Berleant, J. Zhang, W.T. Tucker, L.R Ginzburg, and W.L. Oberkampf. 2004. *Dependence in Probabilistic Modeling, Dempster-Shafer Theory, and Probability Bounds Analysis*. SAND2004-3072, Sandia National Laboratories, Albuquerque, NM. <http://www.ramas.com/depend.pdf>
- Kreinovich, V., J. Beck and H.T. Nguyen. 2005. Ellipsoids and ellipsoid-shaped fuzzy sets as natural multi-variate generalizations of intervals and fuzzy numbers: how to elicit them from users, and how to use them in data processing. *Information Sciences* [to appear]. <http://www.cs.utep.edu/vladik/2005/tr05-13.pdf>
- Kreinovich, V., J. Hajagos, L.R. Ginzburg and S. Ferson. 2006 [tentative]. Propagating uncertainty through quadratic approximations to complex models. SAND2006-xxxx, Sandia National Laboratories, Albuquerque, NM. <http://www.ramas.com/quadratic.pdf>
- Nelsen, R.B. 1999. *An Introduction to Copulas*. Lecture Notes in Statistics 139, Springer-Verlag, New York.

How To Take Into Account Dependence Between the Inputs: From Interval Computations to Constraint-Related Set Computations, with Potential Applications to Nuclear Safety, Bio- and Geosciences

Martine Ceberio¹, Scott Ferson², Vladik Kreinovich¹, Sanjeev Chopra^{1,3}, Gang Xiang¹,
Adrian Murguia^{1,4}, and Jorge Santillan¹

¹*Department of Computer Science, University of Texas, El Paso, TX 79968, USA,
mceberio@cs.utep.edu, vladik@utep.edu, gxiang@utep.edu*

²*Applied Biomathematics, 100 North Country Road, Setauket, New York 11733, USA,
scott@ramas.com*

³*Lexmark International, Inc., 740 New Circle Road NW, Lexington, KY 40550, USA,
sachopra@gmail.com*

⁴*XIMIS, Inc., 6006 N. Mesa, Suite 709, El Paso, TX 79912, USA*

Abstract. In many real-life situations, in addition to knowing the intervals \mathbf{x}_i of possible values of each variable x_i , we also know additional restrictions on the possible combinations of x_i ; in this case, the set \mathbf{x} of possible values of $x = (x_1, \dots, x_n)$ is a proper subset of the original box $\mathbf{x}_1 \times \dots \times \mathbf{x}_n$. In this paper, we show how to take into account this dependence between the inputs when computing the range of a function $f(x_1, \dots, x_n)$.

Keywords: constraints, interval computations, dependence between the inputs

1. Introduction

1.1. GENERAL PROBLEM OF DATA PROCESSING UNDER UNCERTAINTY

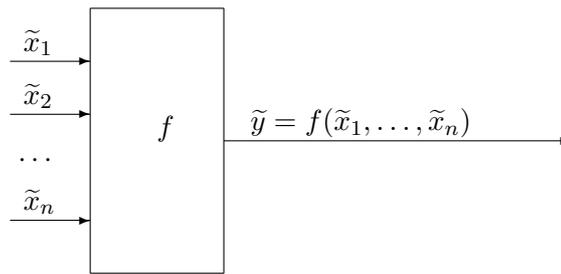
In many real-life situations, there exist quantities which are difficult (or even impossible) to measure directly: e.g., the amount of oil in an oil field, or the temperature inside a reactor. Since we cannot measure the corresponding quantity *directly*, we can measure it *indirectly*: by measuring the values of easier-to-measure quantities x_1, \dots, x_n which are related to the desired quantity y by a known dependence $y = f(x_1, \dots, x_n)$.

The resulting indirect measurement consists of the following:

- first, we measure the quantities x_1, \dots, x_n , and
- then, we apply the function f to the results $\tilde{x}_1, \dots, \tilde{x}_n$ of these measurements.

The resulting value $\tilde{y} = f(\tilde{x}_1, \dots, \tilde{x}_n)$ is our estimate for the desired quantity y .

© 2006 by authors. Printed in USA.



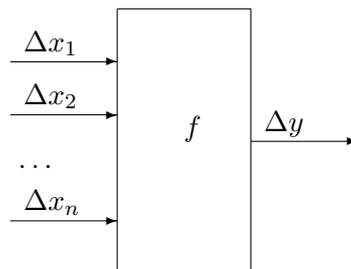
If measurements were absolutely accurate, then we would be able to get the exact values of x_i , and thus, compute the exact value of the desired quantity y . In reality, however, measurements are never 100% accurate; hence, the result \tilde{x}_i of i -th measurement is, in general, different from the actual value x_i of the corresponding quantity. In other words, we have a non-zero *measurement error* $\Delta x_i \neq 0$. Hence, the result $\tilde{y} = f(\tilde{x}_1, \dots, \tilde{x}_n)$ of applying the function f to the measured values is, in general, different from the actual (unknown) value y of the desired quantity – i.e., from the result $y = f(x_1, \dots, x_n)$ of applying the function f to the actual (unknown) values of the quantities x_i .

A natural question is: what can we say about the error $\Delta y \stackrel{\text{def}}{=} \tilde{y} - y$ of indirect measurement?

Comment. In some real-life situations, we also do not know the exact function f , and this uncertainty in f needs to be added to the uncertainty caused by errors of direct measurements $\Delta x_i \neq 0$. In this paper, for simplicity, we consider only the cases when we know the exact expressions for the function f .

1.2. PROBABILISTIC AND INTERVAL UNCERTAINTY

The error Δy of indirect measurement is caused by the measurement errors Δx_i of direct measurements. Thus, to deduce the desired information about Δy , we must use the known information about Δx_i .



Traditionally, in engineering and science, we assume that we know the joint probability distribution for Δx_i . Usually, it is assumed that these measurement errors are independent and normally distributed, with 0 mean and known standard deviations; however, there are also known ways of handling possible dependence and non-Gaussian (non-normal) distributions.

In many real-life situations, we do know these distributions: they come from the process of comparing the currently used measuring instruments (MI) with much more accurate “standard” MIs used in the national or international standards centers. Specifically, we repeatedly measure the same quantity by our MI and by the standard MI. The standard MI is, by definition, much more accurate than our MI, i.e., $|x_i^{\text{stand}} - x_i| \ll |\tilde{x}_i - x_i|$. Hence, the difference $\tilde{x}_i - x_i^{\text{stand}}$ between the results of these two measurements is very close to the actual (unknown) measurement error $\Delta x_i = \tilde{x}_i - x_i$. Thus, by analyzing the sample of such differences, we can infer the probability distribution for the measurement error Δx_i .

This “calibration” of measuring instruments is indeed often performed. However, there are two important classes of situations where this calibration is not done.

The first such class is situations from *fundamental science*. If we are interested in the accuracy of a typical over-the-counter voltmeter, then it is possible to design a more accurate voltmeter and used this more accurate MI to calibrate our MI. However, when we are trying to analyze the accuracy of, say, measurements performed by using the newest particle super-collider, it would nice to have a much more accurate instrument available for calibration, but the existing instrument is the best we have. Similarly, to analyze the accuracy of measurements made by using the Hubble telescope, it would be nice to have a much more accurate instrument floating nearby, but the Hubble is the best we have so far.

Another class of situations is related to *manufacturing*. In manufacturing, in principle, it is possible to calibrate all the sensors. However, a detailed individual calibration of each sensor often costs orders of magnitude more than the sensors themselves. As a result, manufacturers are trying to avoid detailed calibration of all the sensors, and use whatever information is available without spending a lot of money.

In such cases, we *do not* know the probability distribution of the measurement errors Δx_i . What *do* we know in such situations? For sure, the manufacturer of the measuring instrument must supply us with an upper bound Δ_i on the (absolute value of) the measurement error $|\Delta x_i|$. Indeed, if such guaranteed bound is provided, this means that the actual value x_i of the measured quantity can be as far away as possible from the measured value \tilde{x}_i . For example, we measure the current as 1 A, but the actual current current can be 1000 or 0. This is a wild guess, not a measurement. For an instrument to be called a measuring instrument, some bound has to be provided. The manufacturer *may* provide some additional information about Δx_i , but the upper bound *has* to be provided.

Once the upper bound Δ_i on $|\Delta x_i|$ is provided, then, based on the measured value \tilde{x}_i , we can conclude that the actual (unknown) value x_i of the i -th quantity belongs to the interval

$$x_i \in [\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i].$$

In other words, we know the values x_i with *interval uncertainty*.

For example, if the measured current is 1.0 V and the upper bound on the measurement error is 0.1 V, then we are guaranteed that the actual (unknown) value of the current is in the interval $[1.0 - 0.1, 1.0 + 0.1] = [0.9, 1.1]$.

1.3. INTERVAL COMPUTATIONS: A PROBLEM

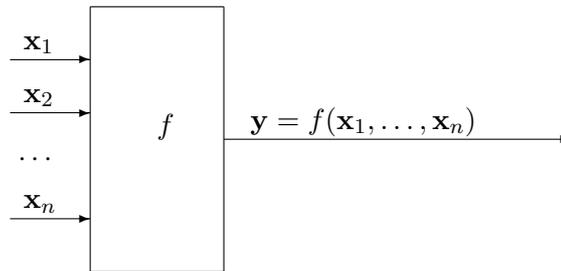
We have just mentioned that in many important real-life situations, we know x_i with interval uncertainty, i.e.:

- we know the ranges \mathbf{x}_i of possible values of x_i , and
- we do not have any information about the probability of different values within these ranges.

In such situations, the only information that we can have about the desired quantity $y = f(x_1, \dots, x_n)$ is the range of possible values of y when $x_i \in \mathbf{x}_i$. In other words, we face the following problem:

- *Given:*
 - an algorithm $y = f(x_1, \dots, x_n)$ that transforms n real numbers x_i into a number y ; and
 - n intervals $\mathbf{x}_i = [\underline{x}_i, \bar{x}_i]$.
- *Compute:* the corresponding range of y :

$$\mathbf{y} = [\underline{y}, \bar{y}] = \{f(x_1, \dots, x_n) \mid x_1 \in [\underline{x}_1, \bar{x}_1], \dots, x_n \in [\underline{x}_n, \bar{x}_n]\}.$$



The problem of computing this range is often called the *main problem* of interval computations; see, e.g., (Jaulin et al., 2001).

It is known that even for quadratic f , the problem of computing the exact range \mathbf{y} is difficult to compute (in precise terms, NP-hard); see, e.g., (Kreinovich et al., 1997; Vavasis, 1991). Crudely speaking, NP-hard means that¹ it is not possible to find an efficient algorithm that would compute the *exact* range for *all* possible problems. Since no such general algorithm is possible, to solve practical problems, we thus need to do the following:

- find classes of problems for which efficient algorithms are possible; and
- for problems outside these classes, find efficient techniques for *approximating* uncertainty of y .

This is what interval computations community has been doing for several decades.

¹ unless P is equal to NP, which most computer scientists do not believe

1.4. WHY NOT MAXIMUM ENTROPY?

From the engineering practical viewpoint, a natural question is: why not use the Maximum Entropy approach? Let us explain what this question means and how to answer it.

Our problems come from the fact that we do not know the exact probability distribution for $\Delta x = (\Delta x_1, \dots, \Delta x_n)$. In real life, this is a frequent situation: in many practical applications, it is very difficult to come up with the probabilities.

The traditional engineering approach recommends that we use probabilistic techniques. If we do not know the exact probability distribution, this means that there are many different probability distributions which are consistent with the same observations and measurements. The traditional engineering solution to this problem is to select one of these distributions – e.g., the one with the largest entropy; see, e.g., (Jaynes, 2003) for the detailed description of this Maximum Entropy (MaxEnt) approach.

For example, suppose that we have only one variable x , and all we know about the actual value of this variable is that it belongs to the interval $[\underline{x}, \bar{x}]$. Since we have no information about the relative probability of different values from this interval, there is no reason to assume that some values are more probable than the others. It is therefore reasonable to assume that all the values within this interval are equally probable, i.e., in precise terms, that we have a uniform distribution on this interval $[\underline{x}, \bar{x}]$. Not surprisingly, this is exactly what MaxEnt leads to.

In case we have several variables Δx_i and we have no information about their correlation, then we have no reason to assume that they are positively or negatively correlated; it is thus reasonable to assume that they are independent. For example, if all we know is that Δx_i belongs to the interval $[-\Delta_i, \Delta_i]$, then the only information that we have about the vector $\Delta x = (\Delta x_1, \dots, \Delta x_n)$ is that it is located in the box $[-\Delta_1, \Delta_1] \times \dots \times [-\Delta_n, \Delta_n]$. Since we have no reason to assume that some values from this box are more probable than the others, it seems reasonable to assume that all the values from the box are equally probable – i.e., in precise terms, that we have a uniform distribution on this box. One can easily see that the uniform distribution on the box means that:

- the variables Δx_i are independent, and
- each variable Δx_i is uniformly distributed in the corresponding interval.

Why should we not use this approach? Because, as we will show, this approach can sometimes seriously underestimate the error of indirect measurement. Indeed, let us consider the simplest possible case, when:

- the desired quantity y is simply the sum of n values x_1, \dots, x_n , i.e., $f(x_1, \dots, x_n) = x_1 + \dots + x_n$, and
- all direct measurements have the same error bound $\Delta_1 = \dots = \Delta_n = \Delta$.

In this case, $\Delta y = \Delta x_1 + \dots + \Delta x_n$, with $\Delta x_i \in [-\Delta_i, \Delta_i]$.

In practice, it is quite possible that all n measurement errors are caused by the same factor; in this case, it is possible that $\Delta x_1 = \dots = \Delta x_n$ and thus, $\Delta y = n \cdot \Delta x_1$. Since the measurement error Δx_1 can take any values from the interval $[-\Delta, \Delta]$, it is possible that $\Delta x_1 = \Delta$ and therefore, it is possible that $\Delta y = n \cdot \Delta$.

On the other hand, when we apply the MaxEnt approach to this situation, we thus assume that the values $\Delta x_i \in [-\Delta, \Delta]$ are independent identically distributed random variables uniformly distributed on the interval $[-\Delta, \Delta]$. For the uniform distribution, the mean is 0, and the variance is $\frac{1}{3} \cdot \Delta^2$. When we add independent random variables, their means and variances add up, so the sum Δy has a mean 0 and variance $V = \frac{1}{3} \cdot n \cdot \Delta^2$.

It is known that, due to the Central Limit Theorem (see, e.g., (Wadsworth, 1990)), for large n , the sum Δy of n independent identically distributed random variables is almost normally distributed. Thus, within the MaxEnt approach, for large n , the measurement error Δy is (almost) normally distributed with 0 means and variance $V = \frac{1}{3} \cdot n \cdot \Delta^2$. It is also well known that for a normally distributed random variable, the probability of a value which is more than, say, 6σ away from the mean is negligibly small ($\approx 10^{-8}$). Thus, from the MaxEnt approach, we conclude that with probability $\geq 1 - 10^{-8}$ (i.e., practically, with certainty), the measurement error Δy is bounded by $6\sigma = 6 \cdot \sqrt{V} \sim \sqrt{n}$.

So, by using the MaxEnt approach, we get an error bound $\sim \sqrt{n}$, but in reality, due to possible correlations, we may have $\Delta y \sim n \gg \sqrt{n}$. Our conclusion is that using a single distribution – even the most reasonable one – can be very misleading, especially if we want guaranteed results, e.g., in high-risk application areas such as space exploration or nuclear engineering.

We therefore need to solve the original problem of interval computations.

1.5. GENERAL APPROACH: INTERVAL-TYPE STEP-BY-STEP TECHNIQUES

In this paper, we will modify the standard interval computation techniques. To explain the needed modification, let us recall these techniques in detail.

As we have mentioned, the main difficulty of solving the main problem of interval computations is that it is (provably) computationally difficult to compute the exact range \mathbf{y} for an arbitrary function $f(x_1, \dots, x_n)$. The solution provided by interval computations is to compute an *enclosure* \mathbf{Y} for this range, i.e., a set \mathbf{Y} for which $\mathbf{y} \subseteq \mathbf{Y}$.

Algorithms for computing an enclosure start with an observation that for arithmetic operations $f(x_1, x_2)$, we have explicit formulas for the range. When $x_1 \in \mathbf{x}_1 = [\underline{x}_1, \bar{x}_1]$ and $x_2 \in \mathbf{x}_2 = [\underline{x}_2, \bar{x}_2]$, then:

- The range $\mathbf{x}_1 + \mathbf{x}_2$ for $x_1 + x_2$ is $[\underline{x}_1 + \underline{x}_2, \bar{x}_1 + \bar{x}_2]$.
- The range $\mathbf{x}_1 - \mathbf{x}_2$ for $x_1 - x_2$ is $[\underline{x}_1 - \bar{x}_2, \bar{x}_1 - \underline{x}_2]$.
- The range $\mathbf{x}_1 \cdot \mathbf{x}_2$ for $x_1 \cdot x_2$ is $[\underline{y}, \bar{y}]$, where

$$\underline{y} = \min(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \bar{x}_2, \bar{x}_1 \cdot \underline{x}_2, \bar{x}_1 \cdot \bar{x}_2); \quad \bar{y} = \max(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \bar{x}_2, \bar{x}_1 \cdot \underline{x}_2, \bar{x}_1 \cdot \bar{x}_2).$$

- The range $1/\mathbf{x}_1$ for $1/x_1$ is $[1/\bar{x}_1, 1/\underline{x}_1]$ (if $0 \notin \mathbf{x}_1$).

These formulas are called formulas of *interval arithmetic*.

The main idea behind straightforward interval computations is that within a computer, only elementary arithmetic operations are hardware supported². No matter how complex the function $f(x_1, \dots, x_n)$ is, the compiler *parses* it, i.e., represents its computation as a sequence of elementary arithmetic operations. The main idea is that if we only know the inputs with interval uncertainty, then we perform the same arithmetic operations in the same order, but with intervals instead of numbers. It is known that the resulting interval is an enclosure for the desired range.

Let us consider a toy example of estimating the range of a function $f(x) = (x - 2) \cdot (x + 2)$ on the interval $x \in [1, 2]$. How will the computer compute this function? It will first compute $x - 2$, then $x + 2$, and then multiply the results. If we denote i -th intermediate computational result by r_i , then we get the following sequence of elementary arithmetic operations:

- $r_1 := x - 2$;
- $r_2 := x + 2$;
- $r_3 := r_1 \cdot r_2$.

If we perform the same operations, but with *intervals* instead of *numbers*, then we get the following intervals:

- $\mathbf{r}_1 := [1, 2] - [2, 2] = [-1, 0]$;
- $\mathbf{r}_2 := [1, 2] + [2, 2] = [3, 4]$;
- $\mathbf{r}_3 := [-1, 0] \cdot [3, 4] = [-4, 0]$.

As a result, we get an interval $[-4, 0]$.

In this toy example, $f(x) = x^2 - 4$, so the actual range of this function on the interval $[1, 2]$ is easy to compute: it is equal to $f(\mathbf{x}) = [-3, 0]$. We can thus see that our computed range $\mathbf{Y} = [-4, 0]$ is indeed the enclosure for the actual range $\mathbf{y} = [-3, 0]$.

Comment. To avoid misunderstanding, we should emphasize that this is just a toy example. There exist more efficient ways of computing an enclosure $\mathbf{Y} \supseteq \mathbf{y}$ than straightforward interval computations (see, e.g., (Jaulin et al., 2001)); however, most of these more efficient and more sophisticated techniques are based on the main ideas of straightforward interval computations.

1.6. FROM “THEORETICAL” INTERVAL COMPUTATIONS TO COMPUTER-REPRESENTABLE INTERVAL COMPUTATIONS: THE NEED FOR ROUNDING

The above formulas for interval arithmetic assumed that all rational numbers can be exactly represented in a computer. In reality, only some binary-rational numbers can be represented. To represent numbers like $1/3$ in a computer, we must therefore *round* these numbers, i.e., replace these theoretically correct numbers with nearby machine-representable ones.

To get a guaranteed enclosure, we must always:

² Actually, only addition, subtraction, and multiplication are directly hardware supported; division a/b is usually implemented as $a \cdot (1/b)$.

- round the lower endpoint of the interval downwards (i.e., replace it with a smaller number), and
- round the upper endpoint of the interval upwards (i.e., replace it with a larger number).

1.7. INTERVAL COMPUTATIONS: ANALYSIS

As we have mentioned, the main problem with computing the *exact* range of the function under interval uncertainty is that this computation is NP-hard, which means that in the worst case, this computation probably require the time which is exponentially growing the size T of the expression f – i.e., grows as 2^T of faster. As a result, for reasonable size algorithms f , with T in hundreds, the required computation time will be unrealistic – e.g., it may exceed the lifetime of the universe.

From this viewpoint, a natural question to ask is: how long will computations take for the above straightforward computations techniques of computing the *enclosure* for the exact range. In straightforward interval computations, each original elementary arithmetic operation is replaced with one operation of interval arithmetic. Each interval arithmetic operation consists of several arithmetic operations with numbers: addition of two intervals means two additions of numbers, etc. The largest number of operation with numbers per single interval arithmetic operation is for interval multiplication, which requires 4 multiplications of numbers. Thus, when we move from the original computations to interval computations, we replace each arithmetic operation with ≤ 4 operations. As a result, the computation time for the straightforward computations is $\leq 4 \cdot T$, i.e., it is $O(T)$, where T is the number of operations in (i.e., in effect, the running time of) the original algorithm.

As a result of straightforward interval computations, we compute the enclosure $\mathbf{Y} \supseteq \mathbf{y}$, often with excess width. As we have seen on the toy example, the main reason why there is an excessive width is that:

- there is a relation between intermediate results, and
- in straightforward interval computations, we ignore this relation.

For example, in the above toy example, the intervals ranges for r_1 and r_2 were exact. However, when we multiplied the corresponding intervals \mathbf{r}_1 and \mathbf{r}_2 , we used the general formulas for interval multiplication, formulas that implicitly assume that all pairs (r_1, r_2) from the corresponding box $\mathbf{r}_1 \times \mathbf{r}_2$ are possible. Thus, we ignored the fact that the values r_1 and r_2 are actually related – since they are both functions of the same variable x – and so, not all pairs (r_1, r_2) are possible.

In addition to algorithms for computing an enclosure, there also exist algorithms for computing the *exact* range; e.g., algorithms based on Tarski’s ideas can be applicable for arbitrary algebraic functions f ; see, e.g., (Kreinovich et al., 1997) and references therein. These algorithms, however, require exponential time $\sim 2^T$ (or even higher) and are, thus, not applicable for large T .

1.8. INTERVAL COMPUTATIONS: THE FIRST PROBLEM

Summarizing the above discussion, we conclude that we have, in effect, two classes of algorithms for solving the main problem of interval computations:

- fast and efficient $O(T)$ algorithms – which often have large excess width;
- slow and inefficient (often non-feasible) algorithms – with no excess width.

In practice, we are often not satisfied with the excess width of a faster algorithm, but we do not have enough time to apply the algorithm for computing the exact range. To take care of such situations, it is desirable to develop a *sequence* of feasible algorithms with:

- longer and longer computation time and
- smaller and smaller excess width.

The development of such a sequence is one of the objectives of this paper.

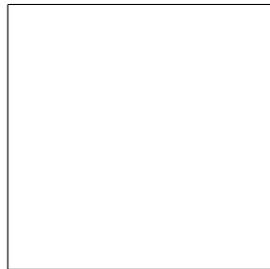
2. Formulation of the Main Problem

2.1. INTERVAL COMPUTATIONS: LIMITATIONS

In traditional interval computations:

- we know the intervals \mathbf{x}_i of possible values of different parameters x_i , and
- we assume that an arbitrary combination of these values is possible.

In geometric terms, this assumption means that the set of possible combinations $x = (x_1, \dots, x_n)$ is a *box* $\mathbf{x} = \mathbf{x}_1 \times \dots \times \mathbf{x}_n$.



In many real-life situations, in addition to knowing the intervals \mathbf{x}_i of possible values of each variable x_i , we also know additional restrictions on the possible combinations of x_i . In this case, the set \mathbf{x} of possible values of x is a (proper) *subset* of the original box. For example, in addition to knowing the bounds on x_1 and x_2 , we may also know that the difference between x_1 and x_2 cannot exceed a certain amount. Informally speaking, the parameters x_i are no longer independent – in the sense that the set of possible values of x_i may depend on the values of other parameters.

In such situations, it is desirable to be able to compute the range of possible values of $f(x_1, \dots, x_n)$ for all combinations (x_1, \dots, x_n) which satisfy the given restrictions. Computing this range is the main objective of this paper.

Comment. In interval computations, we start with independent inputs; as we follow computations, we get dependent intermediate results: e.g., for $x_1 - x_1^2$, the values of x_1 and $x_2 = x_1^2$ are strongly dependent in the sense that only values (x_1, x_1^2) are possible within the box $\mathbf{x}_1 \times \mathbf{x}_2$.

- In interval computations, there are many techniques for handling similar dependence between the *intermediate* computational *results*.
- In this paper, we extend these techniques to handle a different type of dependence – dependence between the *inputs*.

Before we start describing the corresponding ideas and algorithms, let us first give two examples of such restrictions.

2.2. EXAMPLE FROM GEOSCIENCES

Our civilization greatly depends on the things we extract from the Earth, such as fossil fuels (oil, coal, natural gas), minerals, and water. Our need for these commodities is constantly growing, and because of this growth, they are being exhausted. Even under the best conservation policies, there is (and there will be) a constant need to find new sources of minerals, fuels, and water.

The only sure-proof way to guarantee that there are resources such as minerals at a certain location is to actually drill a borehole and analyze the materials extracted. However, exploration for natural resources using indirect means began in earnest during the first half of the 20th century. The result was the discovery of many large relatively easy to locate resources such as the oil in the Middle East.

However, nowadays, most easy-to-access mineral resources have already been discovered. For example, new oil fields are mainly discovered either at large depths, or under water, or in very remote areas – in short, in the areas where drilling is very expensive. It is therefore desirable to predict the presence of resources as accurately as possible before we invest in drilling.

From previous exploration experiences, we usually have a good idea of what type of structures are symptomatic for a particular region. For example, oil and gas tend to concentrate near the top of natural underground domal structures. So, to be able to distinguish between more promising and less promising locations, it is desirable to determine the structure of the Earth at these locations. To be more precise, we want to know the structure at different depths z at different locations (x, y) .

Another vitally important application where the knowledge of the Earth structure is crucial is the assessment of earth hazards. Earthquakes can be very destructive, so it is important to be able to estimate the probability of an earthquake, where one is most likely to occur, and what will be the magnitude of the expected earthquake. Geophysicists have shown that earthquakes result from accumulation of mechanical stress; so if we know the detailed structure of the corresponding Earth locations, we can get a good idea of the corresponding stresses and faults present and the potential for occurrence of an earthquake. From this viewpoint, it is also very important to determine the structure of the Earth.

In general, to determine the Earth structure, we can use different measurement results that can be obtained without actually drilling the boreholes: e.g., gravity and magnetic measurements, analyzing the travel-times and paths of seismic ways as they propagate through the earth, etc.

The relation between the Earth structure and the related measurable quantities is usually known. So, when we know the exact structure at a given Earth location, we can predict, with reasonable accuracy, the corresponding values of the measured quantities – we can predict the local value of the gravity field, the time that a seismic signal needs to travel from its origin to the sensor, etc. Such problems are usually called *forward* problems.

Forward problems enable us, given a model of the Earth, to predict the values of different signals. What we need in the above geophysical applications is the opposite: given the measured values of different signals, we need to reconstruct the structure of the Earth at the location where the measurements have been made. Such problems are therefore called *inverse problems*.

Some measurements – like gravity and magnetic measurements – describe the overall effect of a large area. These measurements can help us determine the average mass density in the area, or the average concentration of magnetic materials in the area, but they often do not determine the detailed structure of this area. This detailed structure can be determined only from measurements which are narrowly focused on small sub-areas of interest.

The most important of these measurements are usually *seismic measurements*. Seismic measurements involve the recording of vibrations caused by distant earthquakes, explosions, or mechanical devices. For example, these records are what seismographic stations all over the world still use to detect earthquakes. However, the signal coming from an earthquake carries not only information about the earthquake itself, it also carries the information about the materials along the path from an earthquake to the station: e.g., by measuring the travel-time of a seismic wave, checking how fast the signal came, we can determine the velocity of sound v in these materials. Usually, the velocity of sound increases with increasing density, so, by knowing the velocity of sound at different 3-D points, we will be able to determine the density of materials at different locations and different depths.

The main problem with the analysis of earthquake data (i.e., *passive* seismic data) is that earthquakes are rare events, and they mainly occur in a few seismically active belts. Thus, we have a very uneven distribution of sources and receivers that results in a “fuzzy” image of earth structure in many areas.

To get a better understanding of the Earth structure, we must therefore rely on *active* seismic data – in other words, we must make artificial explosions, place sensors around them, and measure how the resulting seismic waves propagate. The most important information about the seismic wave is the *travel-time* t_i , i.e., the time that it takes for the wave to travel from its source to the sensor. To determine the geophysical structure of a region, we measure seismic travel times and reconstruct velocities at different depths from these data. The problem of reconstructing this structure is called the *seismic inverse problem*. There are several algorithms for solving this inverse problem; see, e.g., (Hole, 1992; Parker, 1994; Zelt et al., 1998).

In principle, we can determine the paths from the source to each sensor. The travel-time t_i along i -th path can then be determined as the sum of travel-times in different cells j through which this path passes: $t_i = \sum_j \ell_{ij} v_j$, where ℓ_{ij} denotes the length of the part of i -th path within cell j .

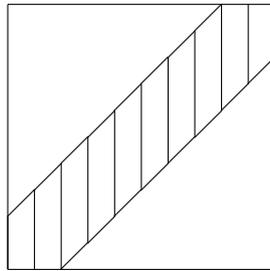
This formula can be somewhat simplified if we replace the velocities v_j by their inverses $s_j \stackrel{\text{def}}{=} \frac{1}{v_j}$,

called *slownesses*. In terms of slownesses, the formula for the travel-time takes the simpler form $t_i = \sum_j \ell_{ij} \cdot s_j$.

For each cell j , a geophysicist usually provides us with the smallest and largest possible value of slowness for this cell. In other words, for each cell j , the expert provides us with an interval $[\underline{s}_j, \bar{s}_j]$ that is guaranteed to contain the actual (unknown) value of slowness s_j . Based on these estimates, we can find the range $[\underline{t}_i, \bar{t}_i]$ of possible values of t_i , where $\underline{t}_i = \sum_j \ell_{ij} \cdot \underline{s}_j$ and $\bar{t}_i = \sum_j \ell_{ij} \cdot \bar{s}_j$. If

the measured travel time \tilde{t}_i is outside this interval, this means that the observed travel-times are *inconsistent* with the intervals $[\underline{s}_j, \bar{s}_j]$. This information should be reported back to the experts, so that the experts will be able to adjust their bounds for s_j in such a way that the new bounds will be consistent with the observations; see, e.g., (Averill et al., 2005).

The above bounds \underline{t}_i and \bar{t}_i were obtained under the assumption that the only information that we have about the slownesses s_j is that each slowness lies in the corresponding interval. In reality, in addition to bounds on slownesses s_j at different points, we also know that slowness cannot change too fast between the neighboring points. To be more precise, the experts usually provide us with a value Δ such that $|s_j - s_k| \leq \Delta$ for all neighboring pairs (j, k) :



It is therefore necessary to find the range of a linear function $t_i = \sum_j \ell_{ij} \cdot s_j$ under such constraints.

2.3. EXAMPLE FROM SAFETY-CRITICAL ENGINEERING

In engineering of safety-critical systems, e.g., in nuclear engineering, it is vitally important to provide safety, i.e., to guarantee that certain quantities y like temperature, pressure, radiation level, do not exceed the required thresholds y_0 . The value of each such quantity y depends on several parameters x_1, \dots, x_n , all of which may somewhat deviate from their nominal values. These parameters may include parameters of the design (such as the exact thickness of the protective layer) or external parameters such as the outdoors temperature.

We usually know the dependence $y = f(x_1, \dots, x_n)$ of the desired quantity y on these parameters. So, the problem of guaranteeing safety means guaranteeing that the upper endpoint \bar{y} of the range $\mathbf{y} = [\underline{y}, \bar{y}]$ of the function $f(x_1, \dots, x_n)$ over all possible combinations (x_1, \dots, x_n) does not exceed y_0 .

We usually know the ranges \mathbf{x}_i of possible values of each of the parameters. Thus, we know that all possible combinations (x_1, \dots, x_n) are within the box $\mathbf{x}_1 \times \dots \times \mathbf{x}_n$. So, in principle, we can guarantee safety if we guarantee that $f(x_1, \dots, x_n) \leq y_0$ for all possible values from this box. In

other words, we can find the range $\tilde{y} = [\underline{y}, \bar{y}]$ of the function $f(x_1, \dots, x_n)$ on the box, and make sure that $\bar{y} \leq y_0$.

This approach does lead to guarantee safety, but it may be too conservative. Indeed, the maximum of the function $f(x_1, \dots, x_n)$ on the box $\mathbf{x}_1 \times \dots \times \mathbf{x}_n$ is often attained at one its endpoints, i.e., at one of the possible combinations of extreme values of x_i . This fact is true, e.g., if the function $f(x_1, \dots, x_n)$ is monotonic in each of its variables. However, experts often claim that combinations of extreme values are impossible. In other words, experts claim that the actual set S of possible values of (x_1, \dots, x_n) is a proper *subset* of the original box – i.e., that there are additional constraints which describe the relation between the parameters x_i .

How can we describe such a subset? In real life, whenever we have a cluster formed by real-life data points, this cluster has a reasonably smooth boundary. This cluster can be a disk (solid circle), a ball (solid sphere in multi-D space), an ellipsoid, or a more complex structure, but it is practically always smooth. The fact that it is smooth means that we can describe its border by an equation $b(x_1, \dots, x_n) = C$ for some smooth function $b(x_1, \dots, x_n)$ and for some constant C . As a result, the set S itself can be describe either by the inequality

$$b(x_1, \dots, x_n) \leq C_0 \quad (1)$$

or by the inequality $b(x_1, \dots, x_n) \geq C_0$. In the second case, the inequality can be transformed into an equivalent form $b'(x_1, \dots, x_n) \leq C'$, where the function $b'(x_1, \dots, x_n) = -b(x_1, \dots, x_n)$ is also smooth, and $C' = -C_0$. So, without loss of generality, we can assume that the set S is described by the inequality (1), for some smooth function $b(x_1, \dots, x_n)$.

An arbitrary smooth function can be approximated by a polynomial, so, instead of the the general set (1), we can consider the approximating set

$$a(x_1, \dots, x_n) \leq C_0, \quad (2)$$

where $a(x_1, \dots, x_n)$ is a polynomial that approximates the smooth function $b(x_1, \dots, x_n)$.

The simplest possible polynomials are linear polynomials $a(x_1, \dots, x_n) = a_0 + a_1 \cdot x_1 + \dots + a_n \cdot x_n$. However, for a linear function $a(x_1, \dots, x_n)$, the set of all the vectors x for which $a(x) \leq C_0$ is a half-space, i.e., a set that is not bounded in many directions, while we want a set S that is inside the box – and hence, bounded in all directions. Thus, if we restrict ourselves to only linear terms, we do not get a good approximation to the set (1).

To get a reasonable approximation, we must consider quadratic and higher order polynomial approximating functions $a(x_1, \dots, x_n)$. In particular, for the simplest non-linear polynomials – quadratic polynomials – the approximating set (2) takes the following form:

$$a(x_1, \dots, x_n) = a_0 + \sum_{i=1}^n a_i \cdot x_i + \sum_{i=1}^n \sum_{j=1}^n a_{i,j} \cdot x_i \cdot x_j \leq C. \quad (3)$$

Ellipsoids indeed provide a reasonable description of the set of possible values of (x_1, \dots, x_n) . To get an even better description of the actual set (1), we can, in principle, use 3rd, 4th, and higher order polynomials.

2.4. HOW THIS INFORMATION IS PROCESSED NOW

At present, to estimate the range of a given function over given constraints, we use problem-specific structure of the objective function $f(x_1, \dots, x_n)$ and of the corresponding constraints.

In geophysical problems, to estimate the range of a linear function $t_i = \sum_j \ell_{ij} \cdot s_j$ under linear constraints $\underline{s}_j \leq s_j \leq \bar{s}_j$ and $|s_j - s_k| \leq \Delta$, we can use linear programming techniques – techniques that were specifically designed for such linear constraint optimization.

Another idea is used to estimate the range of a given function over an ellipsoid in safety-critical engineering; see, e.g., (Kreinovich et al., to appear). Usually, the range of each variable x_i is reasonably narrow, so we can expand the dependence $f(x_1, \dots, x_n)$ in Taylor series around nominal values, and restrict ourselves to quadratic terms in this expansion. As a result, the problem of estimating the range of a given function $f(x_1, \dots, x_n)$ over the range S turns into the problem of estimating the range of the given quadratic function $f(x_1, \dots, x_n)$ over an ellipsoid, i.e., over the range described by quadratic constraints $b(x_1, \dots, x_n) \leq C_0$.

For this constraint optimization problem, the Lagrange multiplier technique reduces it to the problem of unconstrained optimization of a quadratic function

$$F(x_1, \dots, x_n) = f(x_1, \dots, x_n) + \lambda \cdot (b(x_1, \dots, x_n) - C_0).$$

For this quadratic function, we can find the maximum by simply solving an easy-to-solve system of n linear equations with n unknowns: $\frac{\partial F}{\partial x_i} = 0$.

Both ideas can only be used for special objective functions and special constraints. It is therefore desirable to develop *general* techniques for estimating the range of a given function under given constraints.

3. Main Idea

3.1. SIMILAR SITUATION: STATISTICS

In statistics, to get a *complete* description of a multi-dimensional probability distribution of n variables $x = (x_1, \dots, x_n)$, ideally, we should take into account dependence between all the variables. It is, however, often too computationally taxing to find all these dependencies. Therefore, in statistics, it is often necessary to only use partial information about the n -dimensional distribution.

First, we need to find the probability distribution for each of n variables. As we have mentioned earlier, if we have no information about the dependence between these variables, then it is reasonable to assume that these variables are independent. This resulting probability distribution often forms a reasonable *first approximation* to the actual n -dimensional distribution.

To get a more accurate description, the *next* reasonable *step* is to take into account pairwise dependencies, i.e., dependencies between pairs of variables (x_i, x_j) . In the traditional statistical practice in engineering and science, this is done by estimating correlation, covariance, and/or other characteristics of pairwise dependence.

To get an even better picture of the distribution, we can consider dependencies between triples, etc.

As a result, we get a sequence of methods – independent variables, pairwise dependence, dependence between the triples, etc., all the way to a complete description of dependence between all n variables. As we go from independence to taking more and more information about the dependence into account, we get a sequence of methods which:

- require more and more time
- but at the same time lead to more and more accurate results.

3.2. LET US USE A SIMILAR IDEA FOR INTERVAL UNCERTAINTY

How can we use a similar idea to take into account dependence between the inputs in interval computation?

In straightforward interval computations, we consider only intervals of possible values of x_i .

A natural next approximation is when we consider:

- sets \mathbf{x}_i of possible values of x_i , and also
- sets \mathbf{x}_{ij} of possible pairs (x_i, x_j) .

Comment. This idea is similar to *constrained fuzzy arithmetic* developed by G. J. Klir; see, e.g., (Klir, 2000).

The third approximation is when we also consider possible sets of triples \mathbf{x}_{ijk} , etc., all the way to the situation when we completely describe the dependence between x_i by describing the set $\mathbf{x}_{12\dots n}$ of possible values of $x = (x_1, x_2, \dots, x_n)$.

Of course, the more dependence we take into account, the more information we need to store and process and thus, the more computation time the methods will take.

- For straightforward interval computations, all we need to store is intervals of possible values.
- For pairs, we need to store sets of possible values of pairs, i.e., subsets of 2-D boxes. To describe an arbitrary such set with accuracy ε , we must know, for each of $1/\varepsilon^2$ sub-boxes of size $\varepsilon \times \varepsilon$, whether this box belongs to the desired set or not. Thus, we need to store $1/\varepsilon^2$ bits of information.
- For triples, we similarly need $1/\varepsilon^3$ bits of information about whether each of $1/\varepsilon^3$ 3-D boxes of size $\varepsilon \times \varepsilon \times \varepsilon$ belongs to the desired set or not.
- For quadruples, we need $1/\varepsilon^4$ bits, etc.

As a result, we (hope to) get a sequence of methods which:

- require more and more time

- but at the same time lead to more and more accurate results.

3.3. HOW TO IMPLEMENT THIS IDEA

In straightforward interval computations:

- First, we *describe* the initial uncertainty by intervals.
- Then, we show how, by using interval arithmetic, we can *propagate* this uncertainty through the algorithm f , so that at the end, we get an enclosure for the desired range.
- Finally, we show how to adjust operations of interval arithmetic so that all intermediate intervals are *computer-representable* – and at the same time the result is still a guaranteed enclosure.

Similarly, to implement the new idea, we must be able to achieve the following:

- First, we must *describe* the initial uncertainty by sets of pairs etc.
- Second, we must learn how to *propagate* the corresponding uncertainty through algorithms, so that at the end, we will get a better enclosure for the desired range, an enclosure that takes into account the dependence between the inputs.
- Finally, we must learn how to *represent* and process sets of pairs etc, in the *computer*, so that the result will still be a guaranteed enclosure.

We have already decided on how to represent uncertainty by sets of pairs etc. In the following subsections, we will show how we can achieve the two remaining tasks.

3.4. HOW TO PROPAGATE THIS UNCERTAINTY

In the beginning, we know the intervals $\mathbf{r}_1, \dots, \mathbf{r}_n$ corresponding to the input variables $r_i = x_i$, and we know the sets \mathbf{r}_{ij} for i, j from 1 to n .

The question: is how to propagate this information through an intermediate computation step, a step of computing $r_k = r_a * r_b$ for some arithmetic operation $*$ and for previous results r_a and r_b ($a, b < k$). By the time we come to this step, we know the intervals \mathbf{r}_i and the sets \mathbf{r}_{ij} for $i, j < k$. We want to find the interval \mathbf{r}_k for x_k , and the sets \mathbf{r}_{ik} for $i < k$. The following is a natural way to find these sets:

- The range \mathbf{r}_k can be naturally found as $\{r_a * r_b \mid (r_a, r_b) \in \mathbf{r}_{ab}\}$.
- The set \mathbf{r}_{ak} is described as $\{(r_a, r_a * r_b) \mid (r_a, r_b) \in \mathbf{r}_{ab}\}$.
- The set \mathbf{r}_{bk} is described as $\{(r_b, r_a * r_b) \mid (r_a, r_b) \in \mathbf{r}_{ab}\}$.
- For $i \neq a, b$, the set \mathbf{r}_{ik} is described as $\{(r_i, r_a * r_b) \mid (r_i, r_a) \in \mathbf{r}_{ia}, (r_i, r_b) \in \mathbf{r}_{ib}\}$.

Comment. From the mathematical viewpoint, a subset \mathbf{r}_{ij} of the set of all possible pairs $\mathbf{r}_i \times \mathbf{r}_j$ is a *relation*. It is therefore not surprising that processing this uncertainty is similar to processing relations in other application areas such as relational database systems; see, e.g., (Ullman et al., 2002). For example, a natural intermediate step in computing \mathbf{r}_{ik} is when, given the relations \mathbf{r}_{ia} and \mathbf{r}_{ib} , we form a new relation $\{(r_a, r_i, r_b) \mid (r_a, r_i) \in \mathbf{r}_{ia}, (r_i, r_b) \in \mathbf{r}_{ib}\}$. In relational algebra, this intermediate relation is called a *join* and denoted by $\mathbf{r}_{ia} \bowtie_i \mathbf{r}_{ib}$.

3.5. HOW TO REPRESENT SETS IN A COMPUTER

How can we represent a set of pairs or a set of triples in a computer? A natural idea is to do it in a way cumulative probability distributions (cdf) are represented in RiskCalc package (Ferson, 2002): by *discretization*.

In RiskCalc, we divide the interval $[0, 1]$ of possible values of probability into, say, 10 subintervals of equal width and represent cdf $F(x)$ by 10 values x_1, \dots, x_{10} at which $F(x_i) = i/10$.

Similarly, to describe a set $\mathbf{x}_{ij} \subseteq \mathbf{x}_i \times \mathbf{x}_j$, we:

- divide the box $\mathbf{x}_i \times \mathbf{x}_j$ into, say, 10×10 subboxes, and
- describe the set \mathbf{x}_{ij} by listing all subboxes which contain possible pairs.

Comment. This representation of a set by the union of grid cells which intersect with this set is well known in data mining as an upper approximation in the sense of *rough set* theory; see, e.g., (Pawlak, 1991; Polkowski, 2002).

Of course, in reality, there is no need to actually list these subboxes: to describe an arbitrary set, it is sufficient to store $10 \times 10 = 100$ bits of information describing whether each of the 10×10 subboxes belongs to the list. In other words, a set can be represented as 10×10 array of Boolean values. Similarly, for triples, we can represent the corresponding set as a 3-D array of size $10 \times 10 \times 10$, etc.

Comment. The above approach is a good way to describe generic sets, but in practice, the resulting description may be redundant.

- For example, even if we know that all the values (x_1, x_2) are possible, we still need 100 Boolean values to describe this set.
- Similarly, if the set consists of all the values for which $x_1 = x_2$, then out of 100 subboxes, only 10 diagonal boxes are affected, but we still need all 100 Boolean values.

A more efficient idea is to represent sets is by using a *paving* – in the style of (Jaulin et al., 2001). In this approach, we start with a 2×2 subdivision. For each of the $2 \times 2 = 4$ subboxes, we:

- mark this subbox as “in” if it is completely inside the desired set;
- mark this subbox as “out” if it is completely outside the desired set;

- otherwise, if this subbox contains both points from the desired set and point outside the desired set, we subdivide this box into $2 \times 2 = 4$ subboxes, and repeat the procedure.

As a result, we get a list consisting of boxes of different sizes – starting with larger ones and only decreasing the size when necessary.

3.6. HOW TO PROPAGATE THIS UNCERTAINTY: AN ALGORITHM

Let us show how this representation can be propagated through an intermediate computational step, a step of computing $r_k = r_a * r_b$ for some arithmetic operation $*$ and for previous results r_a and r_b ($a, b < k$). We start by dividing each original interval range into the same number C of equal sub-intervals. By the time we come to this step, we know the intervals \mathbf{r}_i and the sets \mathbf{r}_{ij} for $i, j < k$. Each of these sets is described as a union of the subboxes.

We want to find the interval \mathbf{r}_k for x_k , and the sets \mathbf{r}_{ik} for $i < k$. First, we compute the range \mathbf{r}_k :

- In our representation, the set \mathbf{x}_{ab} consists of small 2-D boxes $\mathbf{X}_a \times \mathbf{X}_b$.
- For each small box $\mathbf{X}_a \times \mathbf{X}_b$, we use interval arithmetic to compute the range $\mathbf{X}_a * \mathbf{X}_b$ of the value $r_a * r_b$ over this box.
- Then, we take the union (interval hull) of all these ranges.

Then, we divide this range interval into C equal sub-intervals, and compute the sets \mathbf{r}_{ik} as follows:

- We consider the sets \mathbf{r}_{ab} , \mathbf{r}_{ai} , and \mathbf{r}_{bi} .
- For each small box $\mathbf{R}_a \times \mathbf{R}_b$ from \mathbf{r}_{ab} , we:
 - consider all subintervals \mathbf{R}_i for which $\mathbf{R}_a \times \mathbf{R}_i$ is in \mathbf{r}_{ai} and $\mathbf{R}_b \times \mathbf{R}_i$ is in \mathbf{r}_{bi} , and then
 - we add $(\mathbf{R}_a * \mathbf{R}_b) \times \mathbf{R}_i$ to the set \mathbf{r}_{ki} .

To be more precise, since the interval $\mathbf{R}_a * \mathbf{R}_b$ may not have bounds exactly matching the subdivision of the range interval \mathbf{r}_k into C parts, we may need to expand the interval $\mathbf{R}_a * \mathbf{R}_b$ to get within bounds of this subdivision (numerical examples are given in the following text).

Comment. How long does each computation take? For each i , we need to consider $\leq C^2$ small boxes $\mathbf{R}_a \times \mathbf{R}_b$, and for each such subbox, we must consider C subintervals \mathbf{R}_i , so the computation of each new range \mathbf{r}_{ik} requires $O(C^2) \cdot C = O(C^3)$ computational steps. Since C is a fixed constant, this number does not affect the asymptotic complexity of the proposed algorithm.

We repeat these computations step by step until we get the desired estimate for the range of the final result of the computations.

Comment. Our main objective is to be able to take into account the prior dependence between the inputs x_1, \dots, x_n . However, as a side effect of this technique, in addition to taking into account

dependence between the inputs, we also take care of the (more traditional) dependence between individual results. For example, when we compute the range of $x_1 - x_1^2$, we first compute $x_2 = x_1^2$ and then compute $x_3 = x_1 - x_2$; in our methodology, when we compute x_2 , we automatically generate the set \mathbf{x}_{12} of possible values of pairs (x_1, x_2) . We will see that this set is close to the graph of the function x^2 . On the next step, when we compute $x_3 = x_1 - x_2$, we take into account not only the intervals \mathbf{x}_1 and \mathbf{x}_2 , but also the set \mathbf{x}_{12} , and thus, the resulting estimate for the range for x_3 is close to the ideal.

4. Examples

4.1. FIRST EXAMPLE: COMPUTING THE RANGE OF $x - x$

Let us start with the simplest example where straightforward interval computations lead to over-estimation: the problem of estimating the range of the function $f(x) = x - x$ on the interval $[0, 1]$.

Of course, this function is identically 0, so its actual range is the degenerate interval $[0, 0]$. Let us trace what happens if we apply straightforward interval computations to this function. Parsing leads to the following sequence of elementary arithmetic operations: $r_1 = x$, $r_2 = r_1$, and $r_3 = r_1 - r_2$. So, if we replace each elementary arithmetic operation with the corresponding operation of interval arithmetic, we get $\mathbf{r}_1 = [0, 1]$, $\mathbf{r}_2 = [0, 1]$, and thus, the final range is $\mathbf{r}_3 = \mathbf{r}_1 - \mathbf{r}_2 = [0, 1] - [0, 1] = [-1, 1]$ – an enclosure with excess width.

In straightforward interval computations, we have $r_1 = x$ with the exact interval range $\mathbf{r}_1 = [0, 1]$, we have $r_2 = x$ with the exact interval range $\mathbf{x}_2 = [0, 1]$. We get excess width because the variables r_1 and r_2 are dependent, but we ignore this dependence. In effect, when computing the range \mathbf{r}_3 , we use formulas based on the assumption that the set of possible combinations of (r_1, r_2) is the entire box $\mathbf{r}_1 \times \mathbf{r}_2$.

In the new approach, we still have $\mathbf{r}_1 = \mathbf{r}_2 = [0, 1]$. However, since $r_2 = r_1$, we know that not all pairs (r_1, r_2) from the box $\mathbf{r}_1 \times \mathbf{r}_2$ are possible – the set \mathbf{r}_{12} of possible values of (r_1, r_2) is the diagonal $\mathbf{r}_{12} = \{(r_1, r_2) \mid r_1, r_2 \in [0, 1], r_1 = r_2\}$.

When we compute the range \mathbf{r}_3 of $r_3 = r_1 - r_2$, we only use pairs (r_1, r_2) from the diagonal set \mathbf{r}_{12} . For each point from this diagonal set, $r_3 = r_1 - r_2 = 0$. Thus, with the new techniques, we get the exact range $[0, 0]$ for the function $f(x) = x - x$.

Comment. Similarly, the new method computes the exact range for $x \cdot x$: we have $r_1 = x$, $r_2 = r_1$, and $r_3 = r_1 \cdot r_2$. In contrast, if we use straightforward interval computations, then for $\mathbf{x} = [-1, 1]$, instead of the correct range $[0, 1]$, we get an enclosure $[-1, 1] \cdot [-1, 1] = [-1, 1]$, with excess width.

4.2. SECOND EXAMPLE: COMPUTING THE RANGE OF $x - x^2$

In the example of the degenerate function $f(x) = x - x$, it is easy to avoid excess width without using any new techniques. Indeed, in this example, it is sufficient to simplify the expression for the function $f(x)$ to 0. Many existing compilers can detect the possibility of such a simplification and perform it.

There are less trivial examples of excess width, where a simplification is either impossible or at least is not so easy to find. A simple example of such a situation is the function $f(x) = x - x^2$ on the interval $[0, 1]$.

For this quadratic function, the range can be easily obtained by using the standard calculus technique: namely, according to calculus, to find the range of a function of one variable on a given interval, it is sufficient to find the values of this function on the endpoints and on all the stationary points (i.e., points where the derivative $f'(x)$ is equal to 0). The smallest of these values is the lower endpoint of the range, and the largest of these values is the upper endpoint of the range. For the given function, the only stationary point $f'(x) = 1 - 2x = 0$ is the point $x = 0.5$. So, to find the range of this function, it is sufficient to find its value for $x = 0$ (where $f(0) = 0$), for $x = 0.5$ (where $f(0.5) = 0.25$), and for $x = 1$ (where $f(1) = 0$). Thus, the actual range of this function is $[\min(0, 0.25, 0), \max(0, 0.25, 0)] = [0, 0.25]$.

In straightforward interval computations:

- we have $r_1 = x$ with interval $\mathbf{r}_1 = [0, 1]$;
- we have $r_2 = x^2$ with interval $\mathbf{x}_2 = [0, 1]$;
- the variables r_1 and r_2 are dependent, but we ignore this dependence and estimate \mathbf{r}_3 as $[0, 1] - [0, 1] = [-1, 1]$.

In the new approach, we still have $\mathbf{r}_1 = \mathbf{r}_2 = [0, 1]$, but, since $x_2 = x_1^2$, we now also have the set $\mathbf{r}_{12} = \{(x_1, x_2) \mid x_1, x_2 \in [0, 1], x_2 = x_1^2\}$. When we compute the range \mathbf{r}_3 of $r_3 = r_1 - r_2$, we only use pairs (r_1, r_2) from this set. For each point from this diagonal set, $r_3 = r_1 - r_2 = r_1 - r_1^2$. Thus, with the new techniques, the computed range \mathbf{r}_3 is exactly the range $[0, 0.25]$ of the original function $f(x) = x - x^2$ – with no excess width.

4.3. DISTRIBUTIVITY: $a \cdot (b + c)$ VS. $a \cdot b + a \cdot c$

It is known that interval arithmetic is not distributive in the following sense: when we want to compute the range of the function $f(x_1, x_2, x_3) = x_1 \cdot (x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3$, straightforward interval computations sometimes lead to different enclosures depending on which of the two equal expressions we use.

This is true, e.g., when $x_1 \in \mathbf{x}_1 = [0, 1]$, $\mathbf{x}_2 = [1, 1]$, and $\mathbf{x}_3 = [-1, -1]$. In this case, $x_2 + x_3 = 0$, so $f(x_1, x_2, x_3) = x_1 \cdot (x_2 + x_3) = 0$ for all possible x_i . Hence, the actual range is $[0, 0]$.

For the expression $f(x_1, x_2, x_3) = x_1 \cdot (x_2 + x_3)$, straightforward interval computations lead to $\mathbf{x}_1 \cdot (\mathbf{x}_2 + \mathbf{x}_3) = [0, 1] \cdot [0, 0] = [0, 0]$, i.e., to the exact range. However, for $f(x_1, x_2, x_3) = x_1 \cdot x_2 + x_1 \cdot x_3$, we get $\mathbf{x}_1 \cdot \mathbf{x}_2 + \mathbf{x}_1 \cdot \mathbf{x}_3 = [0, 1] \cdot 1 + [0, 1] \cdot (-1) = [0, 1] + [-1, 0] = [-1, 1]$, i.e., excess width.

The reason for this excess width is that we have the exact ranges for $r_1 = x_1$, $r_2 = x_2$, $r_3 = x_3$, $r_4 = x_1 \cdot x_2$, and $r_5 = x_1 \cdot x_3$, but we ignore the dependence between r_4 and r_5 when computing the range of the final result $r_6 = r_4 + r_5$.

In the new approach, we start with the intervals $\mathbf{r}_1 = \mathbf{x}_1$, $\mathbf{r}_2 = \mathbf{x}_2$, and $\mathbf{r}_3 = \mathbf{x}_3$. Since we are not assuming any dependence between the variables r_1 , r_2 , and r_3 , we thus assume that for these variables, all pairs are possible, i.e., $\mathbf{r}_{12} = \mathbf{r}_1 \times \mathbf{r}_2$, $\mathbf{r}_{23} = \mathbf{r}_2 \times \mathbf{r}_3$, and $\mathbf{r}_{13} = \mathbf{r}_1 \times \mathbf{r}_3$.

When we compute $r_4 = r_1 \cdot r_2$, we also compute the ranges \mathbf{r}_{14} , \mathbf{r}_{24} , and \mathbf{r}_{34} , as

$$\mathbf{r}_{14} = \{(r_1, r_1 \cdot r_2) \mid r_1 \in \mathbf{r}_1, r_2 \in \mathbf{r}_2\}, \quad \mathbf{r}_{24} = \{(r_2, r_1 \cdot r_2) \mid r_1 \in \mathbf{r}_1, r_2 \in \mathbf{r}_2\}, \quad \mathbf{r}_{34} = \mathbf{r}_3 \times \mathbf{r}_4.$$

When we compute $r_5 = r_1 \cdot r_3$, we also compute the range \mathbf{r}_{45} for pairs (r_4, r_5) as

$$\{(r_4, r_1 \cdot r_3) \mid (r_1, r_4) \in \mathbf{r}_{14}, (r_3, r_4) \in \mathbf{r}_{34}\}.$$

From our description of \mathbf{r}_{14} and \mathbf{r}_{34} , we conclude that

$$\mathbf{r}_{45} = \{(r_4, r_1 \cdot r_3) \mid \exists r_2 \in \mathbf{r}_2 \text{ s.t. } r_4 = r_1 \cdot r_2, r_3 \in \mathbf{r}_3\}.$$

Thus,

$$\mathbf{r}_{45} = \{(r_1 \cdot r_2, r_1 \cdot r_3) \mid r_1 \in \mathbf{r}_1, r_2 \in \mathbf{r}_2, r_3 \in \mathbf{r}_3\}.$$

Based on this set, the range of possible values of $r_6 = r_4 + r_5$ coincides with the set

$$\{r_1 \cdot r_2 + r_1 \cdot r_3 \mid r_1 \in \mathbf{r}_1, r_2 \in \mathbf{r}_2, r_3 \in \mathbf{r}_3\},$$

i.e., with the exact range of the function $f(x_1, x_2, x_3) = x_1 \cdot (x_2 + x_3)$.

4.4. TOY EXAMPLE WITH PRIOR DEPENDENCE

Let us consider the problem of finding the range of $r_1 - r_2$ when $\mathbf{r}_1 = [0, 1]$, $\mathbf{r}_2 = [0, 1]$, and $|r_1 - r_2| \leq 0.1$. In this case, the actual range of the difference $r_1 - r_2$ is, of course, $[-0.1, 0.1]$.

Straightforward interval computations cannot take the prior dependence into account. Thus, the only result we can get by using straightforward interval computations is the interval $\mathbf{r}_1 - \mathbf{r}_2 = [0, 1] - [0, 1] = [-1, 1]$.

In the new approach, $\mathbf{r}_{12} = \{(r_1, r_2) \mid r_1 \in [0, 1], r_2 \in [0, 1], |r_1 - r_2| \leq 0.1\}$. The range of the function $r_1 - r_2$ over this set is exactly the desired interval $[-0.1, 0.1]$.

5. Numerical Examples

Let us show that the advantages of the new approach are preserved even when we take into consideration the need to approximate the sets.

5.1. FIRST EXAMPLE: COMPUTING THE RANGE OF $x - x$

As we have mentioned, for $f(x) = x - x$ on $[0, 1]$, the actual range is $[0, 0]$, but straightforward interval computations lead to an enclosure $[0, 1] - [0, 1] = [-1, 1]$. In straightforward interval computations, we have $r_1 = x$ with the exact interval range $\mathbf{r}_1 = [0, 1]$, and we have $r_2 = x$ with the exact interval range $\mathbf{x}_2 = [0, 1]$. The variables r_1 and r_2 are dependent, but we ignore this dependence.

In the new approach: we have $\mathbf{r}_1 = \mathbf{r}_2 = [0, 1]$, and we also have \mathbf{r}_{12} :

r_2					×
			×		
		×			
	×				
	×				
	r_1				

For each small box, we have $[-0.2, 0.2]$, so the union is $[-0.2, 0.2]$.

If we divide into more pieces, we get an interval closer to 0.

5.2. SECOND EXAMPLE: COMPUTING THE RANGE OF $x - x^2$

In straightforward interval computations, we have $r_1 = x$ with the exact interval range interval $\mathbf{r}_1 = [0, 1]$, and we have $r_2 = x^2$ with the exact interval range $\mathbf{x}_2 = [0, 1]$. The variables r_1 and r_2 are dependent, but we ignore this dependence and estimate \mathbf{r}_3 as $[0, 1] - [0, 1] = [-1, 1]$.

In the new approach: we have $\mathbf{r}_1 = \mathbf{r}_2 = [0, 1]$, and we also have \mathbf{r}_{12} . First, we divide the range $[0, 1]$ into 5 equal subintervals \mathbf{R}_1 . The union of the ranges \mathbf{R}_1^2 corresponding to these 5 subintervals \mathbf{R}_1 is $[0, 1]$, so $\mathbf{r}_2 = [0, 1]$. We divide this interval \mathbf{r}_2 into 5 equal sub-intervals $[0, 0.2]$, $[0.2, 0.4]$, etc. We now compute the set \mathbf{r}_{12} as follows:

- for $\mathbf{R}_1 = [0, 0.2]$, we have $\mathbf{R}_1^2 = [0, 0.04]$, so only sub-interval $[0, 0.2]$ of the interval \mathbf{r}_2 is affected;
- for $\mathbf{R}_1 = [0.2, 0.4]$, we have $\mathbf{R}_1^2 = [0.04, 0.16]$, so also only sub-interval $[0, 0.2]$ is affected;
- for $\mathbf{R}_1 = [0.4, 0.6]$, we have $\mathbf{R}_1^2 = [0.16, 0.25]$, so two sub-intervals $[0, 0.2]$ and $[0.2, 0.4]$ are affected, etc.

r_2					×
			×	×	
			×		
		×	×		
	×	×	×		
	r_1				

For each possible pair of small boxes $\mathbf{R}_1 \times \mathbf{R}_2$, we have $\mathbf{R}_1 - \mathbf{R}_2 = [-0.2, 0.2]$, $[0, 0.4]$, or $[0.2, 0.6]$, so the union of $\mathbf{R}_1 - \mathbf{R}_2$ is $\mathbf{r}_3 = [-0.2, 0.6]$.

If we divide into more and more pieces, we get the enclosure which is closer and closer to the exact range $[0, 0.25]$.

5.3. HOW TO COMPUTE \mathbf{r}_{ik}

The above example is a good case to illustrate how we compute the range \mathbf{r}_{13} for $r_3 = r_1 - r_2$. Indeed, since $\mathbf{r}_3 = [-0.2, 0.6]$, we divide this range into 5 subintervals $[-0.2, -0.04]$, $[-0.04, 0.12]$, $[0.12, 0.28]$, $[0.28, 0.44]$, $[0.44, 0.6]$.

- For $\mathbf{R}_1 = [0, 0.2]$, the only possible \mathbf{R}_2 is $[0, 0.2]$, so $\mathbf{R}_1 - \mathbf{R}_2 = [-0.2, 0.2]$. This covers $[-0.2, -0.04]$ and $[-0.04, 0.12]$.
- For $\mathbf{R}_1 = [0.2, 0.4]$, the only possible \mathbf{R}_2 is $[0, 0.2]$, so $\mathbf{R}_1 - \mathbf{R}_2 = [0, 0.4]$. This interval covers $[-0.04, 0.12]$, $[0.12, 0.28]$, and $[0.28, 0.44]$.
- For $\mathbf{R}_1 = [0.4, 0.6]$, we have two possible \mathbf{R}_2 :
 - for $\mathbf{R}_2 = [0, 0.2]$, we have $\mathbf{R}_1 - \mathbf{R}_2 = [0.2, 0.6]$; this covers $[0.12, 0.28]$, $[0.28, 0.44]$, and $[0.44, 0.6]$;
 - for $\mathbf{R}_2 = [0.2, 0.4]$, we have $\mathbf{R}_1 - \mathbf{R}_2 = [0, 0.4]$; this covers $[-0.04, 0.12]$, $[0.12, 0.28]$, and $[0.28, 0.44]$.
- For $\mathbf{R}_1 = [0.6, 0.8]$, we have $\mathbf{R}_1^2 = [0.36, 0.64]$, so three possible \mathbf{R}_2 : $[0.2, 0.4]$, $[0.4, 0.6]$, and $[0.6, 0.8]$, to the total of $[0.2, 0.8]$. Here, $[0.6, 0.8] - [0.2, 0.8] = [-0.2, 0.6]$, so all 5 subintervals are affected.
- Finally, for $\mathbf{R}_1 = [0.8, 1.0]$, we have $\mathbf{R}_1^2 = [0.64, 1.0]$, so two possible \mathbf{R}_2 : $[0.6, 0.8]$ and $[0.8, 1.0]$, to the total of $[0.6, 1.0]$. Here, $[0.8, 1.0] - [0.6, 1.0] = [-0.2, 0.4]$, so the first 4 subintervals are affected.

r_3			×	×	
		×	×	×	×
		×	×	×	×
	×	×	×	×	×
	×		×	×	×
	r_1				

5.4. DISTRIBUTIVITY: $a \cdot (b + c)$ vs. $a \cdot b + a \cdot c$

We want to estimate the range of the function $f(x_1, x_2, x_3) = x_1 \cdot x_2 + x_1 \cdot x_3$ when $x_1 \in \mathbf{x}_1 = [0, 1]$, $\mathbf{x}_2 = [1, 1]$, and $\mathbf{x}_3 = [-1, -1]$. The actual range is $[0, 0]$, but straightforward interval computations lead to $[0, 1] \cdot 1 + [0, 1] \cdot (-1) = [0, 1] + [-1, 0] = [-1, 1]$, i.e., to excess width. The reason is that we have exact ranges for $r_4 = x_1 \cdot x_2$ and $r_5 = x_1 \cdot x_3$, but we ignore the dependence between r_4 and r_5 .

Here, parsing leads to $r_4 = r_1 \cdot r_2$, $r_5 = r_1 \cdot r_3$, and $r_6 = r_4 + r_5$. We start with $\mathbf{r}_1 = [0, 1]$, $r_2 = 1$, and $r_3 = -1$. In the new idea, when we get $r_4 = r_1 \cdot r_2$, we compute the ranges \mathbf{r}_{14} , \mathbf{r}_{24} , and \mathbf{r}_{34} ; the only non-trivial range is \mathbf{r}_{14} :

r_4					×
				×	
			×		
		×			
	×				
×					
					r_1

For $r_5 = r_1 \cdot r_3$, we get $\mathbf{r}_5 = [-1, 0]$. To compute the range \mathbf{r}_{45} , for each possible box $\mathbf{R}_1 \times \mathbf{R}_3$, we:

- consider all boxes \mathbf{R}_4 for which $\mathbf{R}_4 \times \mathbf{R}_1$ is possible and $\mathbf{R}_4 \times \mathbf{R}_3$ is possible; and
- add $\mathbf{R}_4 \times (\mathbf{R}_1 \cdot \mathbf{R}_3)$ to the set \mathbf{r}_{45} .

The result is as follows:

r_5	×				
		×			
			×		
				×	
					×
					r_4

Hence, for $r_6 = r_4 + r_5$, we get $[-0.2, 0.2]$.

If we divide into more pieces, we get the enclosure closer to 0.

5.5. TOY EXAMPLE WITH PRIOR DEPENDENCE

The problem is to find the range of $r_1 - r_2$ when $\mathbf{r}_1 = [0, 1]$, $\mathbf{r}_2 = [0, 1]$, and $|r_1 - r_2| \leq 0.1$. Here, the actual range is $[-0.1, 0.1]$, but straightforward interval computations return $[0, 1] - [0, 1] = [-1, 1]$.

In the new approach, first, we describe the constraint in terms of subboxes:

r_2				×	×
			×	×	×
		×	×	×	
	×	×	×		
×	×	×			
×	×				
					r_1

Next, we compute $\mathbf{R}_1 - \mathbf{R}_2$ for all possible pairs and take the union. The result is $[-0.6, 0.6]$.

If we divide into more pieces, we get the enclosure closer to $[-0.1, 0.1]$.

6. Discussion

When we apply straightforward interval computations to a T -step algorithm,

- we need to compute T intervals \mathbf{r}_i , $i = 1, \dots, T$;
- so, it requires $O(T)$ steps.

In the new approach:

- we need to compute T^2 sets \mathbf{r}_{ij} , $i, j = 1, \dots, T$;
- so, it requires $O(T^2)$ steps.

Thus, the new method takes longer than straightforward interval computations, but it is still feasible.

We have already mentioned that the range estimation problem is, in general, NP-hard (even without any dependency between the inputs). This means that no feasible method can completely avoid excess width. In particular, this means that our quadratic time method cannot completely avoid excess width. So sometimes, we will need better estimates.

To get better estimates, in addition to sets of pairs, we can also consider sets of *triples* \mathbf{r}_{ijk} . This will be a T^3 time version of our approach. If the use of a full subdivision of each box $\mathbf{r}_i \times \mathbf{r}_j \times \mathbf{r}_k$ into $C \times C \times C$ subboxes requires too much computation time, then, instead of using the full 3-D approach, we can use an intermediate “ $2\frac{1}{2}$ -D” approach in which we divide each box into $C \times C \times c$ subboxes, with $c \ll C$.

We can also go to *quadruples* with time $O(T^4)$, etc. When we have tuples with as many elements as the number of variables, we get the exact range. Thus, as we planned, we have a sequence of more and more accurate feasible algorithms for estimating the range, the sequence whose algorithm require longer and longer computation time as the accuracy improves.

Comment. Similar ideas can be applied to the case of expert systems, when we have partial information about probabilities (Ceberio et al., 2005; Ceberio et al., to appear; Chopra, 2005).

Traditionally, expert systems use technique similar to straightforward interval computations: we parse F and replace each computation step with corresponding probability operation. The problem with this approach is that at each step, we ignore the dependence between the intermediate results F_j . As a result, the resulting intervals of possible values of probability are too wide (or, if we use numerical estimates instead of intervals, these numerical estimates can be way off).

This phenomenon can be illustrated on the simple example of estimating the probability $P(A \vee \neg A)$ when $P(A) = 0.5$. In reality, $A \vee \neg A$ is always true, so this probability should be equal to 1. In the interval-type approach, we parse the expression $A \vee \neg A$ into the following sequence: $F_1 = A$, $F_2 = \neg F_1$, and $F_3 = F_1 \vee F_2$. So, first we conclude that $P(F_1) = 0.5$, then that $P(F_2) = 1 - P(F_1) = 1 - 0.5 = 0.5$. However, when we compute the probability $P(F_1 \vee F_2)$, we ignore the dependence between F_1 and F_2 and only use the fact that $P(F_1) = P(F_2) = 0.5$. In this case, the probability $P(F_1 \vee F_2)$ can take any value from the interval $[0.5, 1]$. This interval is what the system returns – with excess width.

A solution to this problem is that, similarly to the above algorithm, on each intermediate step, besides $P(F_j)$, we also compute $P(F_j \& F_i)$ (or $P(F_{j_1} \& \dots \& F_{j_k})$). On each step, we use all combinations of l such probabilities to get new estimates. As a result, we get a new technique in which, e.g., $P(A \vee \neg A)$ is always estimated as 1.

The fact that similar ideas work in interval and in probabilistic cases should not be surprising, because the set of possible values \mathbf{x}_{ij} which described the dependence between two interval-valued quantities is a natural analog between copulas – which describe dependence between two random variables; see, e.g., (Nelsen, 1999).

Acknowledgements

This work was largely inspired by suggestions from Luc Jaulin, Arnold Neumaier, and Bill Walster during the 2005 Scandinavian Workshop on Interval Computations.

This work was supported in part by NASA under cooperative agreement NCC5-209, NSF grants EAR-0225670 and DMS-0532645, Army Research Lab grant DATM-05-02-C-0046, Star Award from the University of Texas System, and Texas Department of Transportation grant No. 0-5453

References

- Averill, M. G., K. C. Miller, G. R. Keller, V. Kreinovich, R. Araiza, and S. A. Starks, Using Expert Knowledge in Solving the Seismic Inverse Problem. In: *Proceedings of the 24th International Conference of the North American Fuzzy Information Processing Society NAFIPS'2005*, Ann Arbor, Michigan, June 22–25, 2005, pp. 310–314.
- Ceberio, M., V., Kreinovich, S. Chopra, and B. Ludäscher, Taylor Model-Type Techniques for Handling Uncertainty in Expert Systems, with Potential Applications to Geoinformatics. In *Proceedings of the 17th World Congress of the International Association for Mathematics and Computers in Simulation IMACS'2005*, Paris, France, July 11–15, 2005.
- Ceberio, M., V. Kreinovich, S. Chopra, L. Longpré, B. Ludäscher, and C. Baral, Interval-Type and Affine Arithmetic-Type Techniques for Handling Uncertainty in Expert Systems”, *Journal of Computational and Applied Mathematics* (to appear).
- Chopra, S. *Affine Arithmetic-Type Techniques for Handling Uncertainty in Expert Systems*. Master’s Thesis, Department of Computer Science, University of Texas at El Paso, 2005.
- Ferson, S. *RAMAS RiskCalc: Risk Assessment with Uncertain Numbers*. CRC Press, Boca Raton, Florida, 2002.
- Ferson, S., L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, Exact Bounds on Finite Populations of Interval Data, *Reliable Computing*, 11(3):207–233, 2005.
- Hansen, E. Sharpness in interval computations, *Reliable Computing*, 3:7–29, 1997.
- Hole, J. A. Nonlinear High-Resolution Three-Dimensional Seismic Travel Time Tomography. *J. Geophysical Research*, 97(B5):6553–6562, 1992.
- Jaulin, L., M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer-Verlag, London, 2001.
- Jaynes, E. T. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, Massachusetts, 2003.
- Klir, G. J. *Fuzzy Sets: An Overview of Fundamentals, Applications, and Personal Views*. Beijing Normal University Press, Beijing, 2000.
- Kreinovich, V., J. Beck, and H. T. Nguyen, Ellipsoids and Ellipsoid-Shaped Fuzzy Sets as Natural Multi-Variate Generalization of Intervals and Fuzzy Numbers: How to Elicit Them from Users, and How to Use Them in Data Processing, *Information Sciences* (to appear).

- Kreinovich, V., A. Lakeyev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*. Kluwer, Dordrecht, 1997.
- Nelsen, R. B. *Introduction to Copulas*. Springer Verlag, New York, 1999.
- Parker, R. L. *Geophysical Inverse Theory*. Princeton University Press, Princeton, New Jersey, 1994.
- Pawlak, Z. *Rough Sets*. Kluwer Academic Publishers, Dordrecht, 1991.
- Polkowski, L. *Rough sets. Mathematical foundations*. Physica Verlag, A Springer-Verlag Co., Heidelberg, New York, 2002.
- Ullman, J. D., and J. Widom, *A First Course in Database Systems*, Prentice Hall, Upper Saddle River, New Jersey, 2002.
- Vavasis, S. A. *Nonlinear Optimization: Complexity Issues*. Oxford University Press, New York, 1991.
- Wadsworth Jr., H. M. *Handbook of Statistical Methods for Engineers and Scientists*. McGraw-Hill, N.Y., 1990.
- Zelt, C. A., and P. J. Barton, Three-dimensional seismic refraction tomography: A comparison of two methods applied to data from the Faeroe Basin, *J. Geophysical Research*, 103(B4):7187–7210, 1998.

Appendix

A. Open Questions

When is the New Method Exact? It is known that straightforward interval computations produce the exact range for single-use expressions (SUE), in which each variable occurs exactly once; see, e.g., (Hansen, 1997; Jaulin et al., 2001). A natural question is: is there a similar syntactic class of expressions for which our pair-wise method leads to the exact range?

One seemingly natural hypothesis does not work here. Namely, we have shown that our new method leads to the exact range for expressions $x - x$, $x - x^2$, and $x_1 \cdot x_2 + x_1 \cdot x_3$. In all these expressions, each variable occurs no more than twice. It may therefore seem natural to conjecture that the new method is exact for all such “double-use” expressions. Alas, this is not true: it is known (see, e.g., (Ferson et al., 2005)) that computing the range of the variance $V = \frac{1}{n} \cdot \sum_{i=1}^n x_i^2 -$

$\left(\frac{1}{n} \cdot \sum_{i=1}^n x_i\right)^2$ on interval data \mathbf{x}_i is NP-hard. Since variance is an example of a double-use expression, and our algorithm is feasible, we can thus conclude that for some double-use problems, it must lead to excess width.

If we allow prior constraints, then the problem of estimating the range become NP-hard even for SUE expressions with linear SUE constraints. Indeed, we can take an arbitrary non-SUE algebraic expression, replace each occurrence of each variable x_i with different new variables x_{i1}, x_{i2}, \dots —this will make this expression SUE, and then add SUE linear constraint $x_{i1} = x_{i2}, x_{i2} = x_{i3}, \dots$. Under these constraints, the range of the new expression is exactly the same as the range of the original expression, and we already know that computing the range of even quadratic expressions is NP-hard.

What Are the Possible Shapes of \mathbf{r}_{ij} ? It is easy to show that for 1-D ranges, for algebraic functions $f(x_1, \dots, x_n)$ (i.e., solutions of polynomial equations with polynomial coefficients), the endpoints

of the range intervals are algebraic numbers, and that, vice versa, every interval with algebraic endpoints is a range of an appropriate algebraic function; see, e.g. (Kreinovich et al., 1997).

It is easy to show that when we have two algebraic functions $f(x_1, \dots, x_n)$ and $g(x_1, \dots, x_n)$, then the set of possible values of pairs (f, g) is semi-algebraic (i.e., is described by a finite set of polynomial equalities and inequalities). A natural question is: can every semi-algebraic set in \mathbb{R}^2 be thus represented? What about sets in \mathbb{R}^3 ? in \mathbb{R}^n for an arbitrary n ?

Validated Solution of Initial Value Problems for ODEs with Interval Parameters

Youdong Lin and Mark A. Stadtherr*

*Department of Chemical and Biomolecular Engineering, University of Notre Dame,
Notre Dame, IN 46556, USA*

Abstract. In initial value problems for ODEs with interval-valued parameters, it is desirable in many applications to be able to determine a validated enclosure of all possible solutions to the ODE system. Much work has been done for the case in which initial values are given by intervals, and there are several available software packages that deal with this case. However, relatively little work has been done on the case in which parameters are given by intervals. We demonstrate here a new method for obtaining validated solutions of initial value problems for ODEs with interval-valued parameters. The method also accounts for interval-valued initial values. The effectiveness of the method is demonstrated using numerical examples involving kinetics in a bioreactor and motion of a double pendulum.

Keywords: ODE, IVP, Parametric uncertainty, Bioreactor kinetics, Double pendulum

1. Introduction

Initial value problems for ODEs arise naturally in many applications in engineering and science. It is often the case that the problem involves parameters and/or initial values that are not known with certainty but that can be expressed as intervals. For this situation it is desirable to be able to determine an enclosure of all possible solutions to the ODEs. Interval methods (validated methods) not only can determine such guaranteed error bounds on the true solution, but can also verify that a unique solution to the problem exists. An excellent review of interval methods for initial value problems has been given by Nedialkov et al. (1999). Much work has been done for the case in which the initial values are given by intervals, and there are several available software packages, including AWA (Lohner, 1992), VNODE (Nedialkov et al., 2001) and COSY VI (Berz and Makino, 1998), that deal with this case. However, relatively little work has been done on the case in which parameters are given by intervals. We concentrate here on the case of such parametric ODEs. However, the method demonstrated will also account for interval-valued initial values.

Since available general-purpose validated ODE solvers are focused on dealing with uncertainties in the initial values, the presence of interval parameters can cause inefficiencies because they lead to a wrapping effect. An alternative approach is to treat time-invariant interval parameters as additional state variables, with zero first-order derivatives, as suggested by Lohner (1988). Since the parameters are now treated as independent variables, tighter enclosures can be obtained.

* Author to whom all correspondence should be addressed. E-mail: markst@nd.edu

However, the increase in the number of state variables, m , can result in a significant increase in the computational expense. For example, a matrix of order m must be factored at each time step in the usual methods (e.g., QR factorization) for controlling the wrapping effect. In this work, we will demonstrate a new method for efficiently determining validated solutions of ODEs with interval parameters; instead of increasing the number of state variables, this method will treat the parametric uncertainty directly. The method makes use, in a novel way, of the Taylor model approach that Makino and Berz (1996) used to deal with the dependence problem in interval arithmetic, and which they applied in COSY VI (Berz and Makino, 1998).

2. Background

2.1. INTERVAL ANALYSIS

A real interval X is defined as the set of real numbers lying between (and including) given upper and lower bounds; that is,

$$X = [\underline{X}, \overline{X}] = \{x \in \mathfrak{R} \mid \underline{X} \leq x \leq \overline{X}\}. \quad (1)$$

Here an underline is used to indicate the lower bound of an interval and an overline is used to indicate the upper bound. A real interval vector $\mathbf{X} = (X_1, X_2, \dots, X_n)^T$ has n real interval components and can be interpreted geometrically as an n -dimensional rectangle or box. Note that in this context uppercase quantities are intervals, and lowercase quantities or uppercase quantities with underline or overline are real numbers.

Basic arithmetic operations with intervals are defined by

$$X \text{ op } Y = \{x \text{ op } y \mid x \in X, y \in Y\}, \quad (2)$$

where $\text{op} = \{+, -, \times, \div\}$. Interval versions of the elementary functions can be similarly defined. It should be emphasized that, when machine computations with interval arithmetic operations are done, as in the procedures outlined below, the endpoints of an interval are computed with a directed (outward) rounding. That is, the lower endpoint is rounded down to the next machine-representable number and the upper endpoint is rounded up to the next machine-representable number. In this way, through the use of interval, as opposed to floating-point arithmetic, any potential rounding error problems are avoided. Several good introductions to interval analysis, as well as interval arithmetic and other aspects of computing with intervals, are available (Jaulin et al., 2001; Hansen and Walster, 2004; Kearfott, 1996; Neumaier, 1990). Implementations of interval arithmetic and elementary functions are also readily available, and recent compilers from Sun Microsystems directly support interval arithmetic and an interval data type.

For an arbitrary function $f(\mathbf{x})$, the interval extension $F(\mathbf{X})$ encloses all possible values of $f(\mathbf{x})$ for $\mathbf{x} \in \mathbf{X}$; that is, it encloses the range of $f(\mathbf{x})$ over \mathbf{X} . It is often computed by substituting the given interval \mathbf{X} into the function $f(\mathbf{x})$ and then evaluating the function using interval arithmetic. This so-called “natural” interval extension is often wider than the actual range of function values, though it always includes the actual range. This overestimation of the function range is due to

the “dependency” problem, which may arise when a variable occurs more than once in a function expression. There are a variety of approaches that can be used to try to tighten interval extensions (Jaulin et al., 2001; Hansen and Walster, 2004; Kearfott, 1996; Neumaier, 1990), including the use of Taylor models, as described in the next subsection.

2.2. TAYLOR MODELS

Makino and Berz (1996; 1999) have described a remainder differential algebra (RDA) approach for bounding function ranges and controlling the dependency problem of interval arithmetic. This method employs high-order computational differentiation to express a function by a model consisting of a Taylor polynomial, usually a truncated Taylor series, and an interval remainder bound.

Consider a function $f : \mathbf{x} \in \mathbf{X} \subset \mathbb{R}^m \rightarrow \mathbb{R}$ that is $(q+1)$ times partially differentiable on \mathbf{X} and let $\mathbf{x}_0 \in \mathbf{X}$. The Taylor theorem states that for each $\mathbf{x} \in \mathbf{X}$, there exists a $\zeta \in \mathbb{R}$ with $0 < \zeta < 1$ such that

$$f(\mathbf{x}) = \sum_{i=0}^q \frac{1}{i!} [(\mathbf{x} - \mathbf{x}_0) \cdot \nabla]^i f(\mathbf{x}_0) + \frac{1}{(q+1)!} [(\mathbf{x} - \mathbf{x}_0) \cdot \nabla]^{q+1} f[\mathbf{x}_0 + (\mathbf{x} - \mathbf{x}_0)\zeta], \quad (3)$$

where the partial differential operator $[\mathbf{g} \cdot \nabla]^k$ is

$$[\mathbf{g} \cdot \nabla]^k = \sum_{\substack{j_1 + \dots + j_m = k \\ 0 \leq j_1, \dots, j_m \leq k}} \frac{k!}{j_1! \dots j_m!} g_1^{j_1} \dots g_m^{j_m} \frac{\partial^k}{\partial x_1^{j_1} \dots \partial x_m^{j_m}}. \quad (4)$$

The last (remainder) term in (3) can be quantitatively bounded over $0 < \zeta < 1$ using interval arithmetic or other methods to obtain an interval remainder bound. The Taylor model for $f(\mathbf{x})$ then consists of a q -th order polynomial in $(\mathbf{x} - \mathbf{x}_0)$, $p_f(\mathbf{x} - \mathbf{x}_0)$ (the summation in (3)), and an interval remainder bound R_f . This Taylor model is denoted by $T_f = (p_f, R_f)$.

Arithmetic operations with Taylor models can be done using the RDA approach described by Makino and Berz (1996; 1999; 2003). Let T_f and T_g be the Taylor models of the functions $f(\mathbf{x})$ and $g(\mathbf{x})$ respectively over the interval $\mathbf{x} \in \mathbf{X}$. The Taylor model of $f \pm g$ can be represented as

$$T_{f \pm g} = (p_f, R_f) \pm (p_g, R_g) = (p_f \pm p_g, R_f \pm R_g) = (p_{f \pm g}, R_{f \pm g}). \quad (5)$$

For the the product $f \times g$,

$$f \times g \in (p_f, R_f) \times (p_g, R_g) \subseteq p_f \times p_g + p_f \times R_g + p_g \times R_f + R_f \times R_g. \quad (6)$$

Note that $p_f \times p_g$ is a polynomial of order $2q$. In order to be consistent with the q -th order polynomial in a Taylor model, this term is split into the sum of a polynomial $p_{f \times g}$ of up to q -th order, and an extra polynomial p_e containing the higher order terms. A Taylor model for the product $f \times g$ can then be given by $T_{f \times g} = (p_{f \times g}, R_{f \times g})$, with

$$R_{f \times g} = B(p_e) + B(p_f) \times R_g + B(p_g) \times R_f + R_f \times R_g. \quad (7)$$

Here $B(p) = P(\mathbf{X} - \mathbf{x}_0)$ denotes an interval bound of the polynomial $p(\mathbf{x} - \mathbf{x}_0)$ over $\mathbf{x} \in \mathbf{X}$. Similarly, an interval bound on an overall Taylor model $T = (p, R)$ will be denoted by $B(T) = B(p) + R$.

In storing and operating on a Taylor model, only the coefficients of the polynomial part $p(\mathbf{x} - \mathbf{x}_0)$ are used, and these are point valued. However, when these coefficients are computed in floating point arithmetic, numerical errors may occur and they must be bounded. To do this in our current implementation of Taylor model arithmetic, we have used the “tallying variable” approach, as described by Makino and Berz (2003). This approach has been analyzed in detail by Revol et al. (2005). This results in an error bound on the floating point calculation of the coefficients in $p(\mathbf{x} - \mathbf{x}_0)$ being added to the interval remainder bound R .

Taylor models for the reciprocal operation, as well as the intrinsic functions (exponential, logarithm, square root, sine, cosine, etc.) can also be obtained (Makino, 1998; Makino and Berz, 1996; Makino and Berz, 2003). Using these, together with the basic arithmetic operations defined above, it is possible to start with simple functions such as the constant function $k(\mathbf{x}) = k$, for which $T_k = (k, [0, 0])$, and the identity function $i(x_i) = x_i, i = 1, \dots, m$, for which $T_i = (x_{i0} + (x_i - x_{i0}), [0, 0])$, and to then compute Taylor models for very complicated functions. Altogether, it is possible to compute a Taylor model for any function that can be represented in a computer environment by simple operator overloading through RDA operations. It has been shown that, compared to other rigorous bounding methods, the Taylor model often yields sharper bounds for modest to complicated functional dependencies (Makino and Berz, 1996; Makino and Berz, 1999; Neumaier, 2002).

3. Validated Solution of Parametric ODEs

Traditional interval methods usually consist of two processes applied at each integration step (Moore, 1966; Nedialkov et al., 1999). In the first process, existence and uniqueness of the solution are proven using the Picard-Lindelöf operator and the Banach fixed point theorem (Eijgenraam, 1991), and a rough enclosure of the solution is computed. In the second process, a tighter enclosure of the solution is computed. In general, both processes are realized by applying interval Taylor series (ITS) expansions with respect to time, and using automatic differentiation to obtain the Taylor coefficients. We will demonstrate here the use of a new method (Lin and Stadtherr, 2005) for the validated solution of parametric ODEs, which is used to produce guaranteed bounds on the solutions of dynamic systems with interval-valued initial states and parameters. The method uses the traditional two-phase approach, but in the second phase makes use of Taylor models to deal with the uncertain quantities (parameters and initial values). We will summarize here the basic ideas of this approach. Additional details are given by Lin and Stadtherr (2005).

Consider the following parametric ODE system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \in \mathbf{X}_0, \quad \boldsymbol{\theta} \in \boldsymbol{\Theta}, \quad (8)$$

where \mathbf{x} is the m -dimensional state vector, $\boldsymbol{\theta}$ is a p -dimensional parameter vector, and $t \in [t_0, t_N]$ for some $t_N > t_0$. The interval vectors \mathbf{X}_0 and $\boldsymbol{\Theta}$ represent enclosures of initial values and parameters, respectively. It is desired to determine a validated enclosure of all possible solutions to this initial value problem. Also note that nonautonomous (time dependent) problems can be converted to the autonomous form given in (8). We denote by $\mathbf{x}(t; t_j, \mathbf{X}_j, \boldsymbol{\Theta})$ the set of solutions $\mathbf{x}(t; t_j, \mathbf{X}_j, \boldsymbol{\Theta}) = \{\mathbf{x}(t; t_j, \mathbf{x}_j, \boldsymbol{\theta}) \mid \mathbf{x}_j \in \mathbf{X}_j, \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, where $\mathbf{x}(t; t_j, \mathbf{x}_j, \boldsymbol{\theta})$ denotes a solution of $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$ for the

initial condition $\mathbf{x} = \mathbf{x}_j$ at t_j . We will describe a method for determining enclosures \mathbf{X}_j of the state variables at each time step $j = 1, \dots, N$, such that $\mathbf{x}(t_j; t_0, \mathbf{X}_0, \Theta) \subseteq \mathbf{X}_j$.

Assume that at t_j we have an enclosure \mathbf{X}_j of $\mathbf{x}(t_j; t_0, \mathbf{X}_0, \Theta)$, and that we want to carry out an integration step to compute the next enclosure \mathbf{X}_{j+1} . Then, in the first phase of the method, the goal is to find a step size $h_j = t_{j+1} - t_j > 0$ and a prior enclosure $\tilde{\mathbf{X}}_j$ of the solution such that a unique solution $\mathbf{x}(t; t_j, \mathbf{x}_j, \theta) \in \tilde{\mathbf{X}}_j$ is guaranteed to exist for all $t \in [t_j, t_{j+1}]$, all $\mathbf{x}_j \in \mathbf{X}_j$, and all $\theta \in \Theta$. We apply the traditional interval method, with high order enclosure, to the parametric ODEs by using an interval Taylor series (ITS) with respect to time. That is, we determine h_j and $\tilde{\mathbf{X}}_j$ such that for $\mathbf{X}_j \subseteq \tilde{\mathbf{X}}_j^0$,

$$\tilde{\mathbf{X}}_j = \sum_{i=0}^{k-1} [0, h_j]^i \mathbf{F}^{[i]}(\mathbf{X}_j, \Theta) + [0, h_j]^k \mathbf{F}^{[k]}(\tilde{\mathbf{X}}_j^0, \Theta) \subseteq \tilde{\mathbf{X}}_j^0. \quad (9)$$

Here k denotes the order of the Taylor expansion, and the coefficients $\mathbf{F}^{[i]}$ are interval extensions of the Taylor coefficients $\mathbf{f}^{[i]}$ of $\mathbf{x}(t)$ with respect to time, which can be obtained recursively in terms of $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \theta)$ by

$$\begin{aligned} \mathbf{f}^{[0]} &= \mathbf{x} \\ \mathbf{f}^{[1]} &= \mathbf{f}(\mathbf{x}, \theta) \\ \mathbf{f}^{[i]} &= \frac{1}{i} \left(\frac{\partial \mathbf{f}^{[i-1]}}{\partial \mathbf{x}} \mathbf{f} \right) (\mathbf{x}, \theta), \quad i \geq 2. \end{aligned} \quad (10)$$

Satisfaction of (9) demonstrates that there exists a unique solution $\mathbf{x}(t; t_j, \mathbf{x}_j, \theta) \in \tilde{\mathbf{X}}_j$ for all $t \in [t_j, t_{j+1}]$, all $\mathbf{x}_j \in \mathbf{X}_j$, and all $\theta \in \Theta$.

In phase 2, we compute a tighter enclosure $\mathbf{X}_{j+1} \subseteq \tilde{\mathbf{X}}_j$, such that $\mathbf{x}(t_{j+1}; t_0, \mathbf{X}_0, \Theta) \subseteq \mathbf{X}_{j+1}$. This will be done by using an ITS approach to compute a Taylor model $\mathbf{T}_{\mathbf{x}_{j+1}}$ of \mathbf{x}_{j+1} in terms of the initial values and parameters, and then obtaining the enclosure $\mathbf{X}_{j+1} = B(\mathbf{T}_{\mathbf{x}_{j+1}})$. For the Taylor model computations, we begin by representing the interval initial states and parameters by the Taylor models $\mathbf{T}_{\mathbf{x}_0}$ and \mathbf{T}_{θ} , respectively, with components

$$T_{x_{i0}} = (m(X_{i0}) + (x_{i0} - m(X_{i0})), [0, 0]), \quad i = 1, \dots, m, \quad (11)$$

and

$$T_{\theta_i} = (m(\Theta_i) + (\theta_i - m(\Theta_i)), [0, 0]), \quad i = 1, \dots, p. \quad (12)$$

Then, we can determine Taylor models $\mathbf{T}_{\mathbf{f}^{[i]}}$ of the interval Taylor series coefficients $\mathbf{f}^{[i]}(\mathbf{x}_i, \theta)$ by using RDA operations to compute $\mathbf{T}_{\mathbf{f}^{[i]}} = \mathbf{f}^{[i]}(\mathbf{T}_{\mathbf{x}_j}, \mathbf{T}_{\theta})$. Using an interval Taylor series for \mathbf{x}_{j+1} with coefficients given by $\mathbf{T}_{\mathbf{f}^{[i]}}$, and incorporating a novel approach for using the mean value theorem on Taylor models, one can obtain a result for $\mathbf{T}_{\mathbf{x}_{j+1}}$ in terms of the parameters and initial states. This result can be improved (tightened) by applying additional steps, based on a QR factorization approach, to further control the wrapping effect. Complete details of the computation of $\mathbf{T}_{\mathbf{x}_{j+1}}$ are given by Lin and Stadtherr (2005).

4. Results

We now report results of numerical experiments using a C++ implementation of the method outlined above. This implementation is called VSPODE (Validating Solver for Parametric ODEs). The results for VSPODE were obtained using a $k = 17$ order interval Taylor series method, and with a $q = 5$ order Taylor model. All tests were performed on a workstation running Linux with an Intel Pentium 4 3.2GHz CPU.

4.1. BIOREACTOR KINETICS

In a bioreactor, a simple microbial growth process (Bastin and Dochain, 1990), which involves a single biomass and single substrate, can be described using the following ODE model,

$$\dot{X} = (\mu - \alpha D)X \quad (13)$$

$$\dot{S} = D(S^i - S) - k\mu X, \quad (14)$$

where X and S are concentrations of biomass and substrate, respectively; α is the process heterogeneity parameter; D and S^i are the dilution rate and the influent concentration of substrate, respectively; k is the yield coefficient; and μ is the growth rate, which is dependent on S . We consider two models for μ , the Monod law,

$$\mu = \frac{\mu_m S}{K_S + S}, \quad (15)$$

and the Haldane law,

$$\mu = \frac{\mu_m S}{K_S + S + K_I S^2}, \quad (16)$$

where μ_m is the maximum growth rate, K_S is the saturation parameter, and K_I is the inhibition parameter. In this study, the initial value of biomass concentration X_0 , and the process kinetic parameters (μ_m , K_S , and K_I) are assumed to be uncertain and given by intervals. Thus, for the Monod law, there are three uncertain quantities, and four for the Haldane law. The values of the initial conditions (X_0 , S_0), the inputs (D and S^i), and parameters (α , k , μ_m , K_S , and K_I) are given in Table I.

For purposes of comparison, as a representative of traditional interval methods, we used the popular VNODE package (Nedialkov et al., 2001), with a $k = 17$ order interval Hermite-Obreschkoff QR method. Though, like other available solvers, VNODE is designed to deal with uncertain initial values, it can take interval parameter values as input. However, better performance can be obtained by treating the uncertain parameters as additional state variables with zero time derivatives; thus the parametric uncertainties become uncertainties in the initial values of the extra state variables.

Enclosures of the state variables S and X for $t \in [0, 20]$ were computed using VSPODE and VNODE with constant step size $h = 0.1$. The results were shown in Fig. 1 and Fig. 2 for the Monod law and the Haldane law, respectively. VSPODE clearly provides a better enclosure, with VNODE failing at $t = 9.3$ for the Monod law, and at $t = 6.6$ for the Haldane law. In order to allow VNODE to solve the problem all the way to $t_N = 20$, we divided the intervals into a

Table I. Bioreactor microbial growth parameters

Parameter	Value	Units	Parameter	Value	Units
α	0.5	-	μ_m	[1.19, 1.21]	day ⁻¹
k	10.53	g S/ g X	K_S	[7.09, 7.11]	g S/l
D	0.36	day ⁻¹	K_I	[0.49, 0.51]	(g S/l) ⁻¹
S^i	5.7	g S/l	X_0	[0.82, 0.84]	g X/l
S_0	0.80	g S/l			

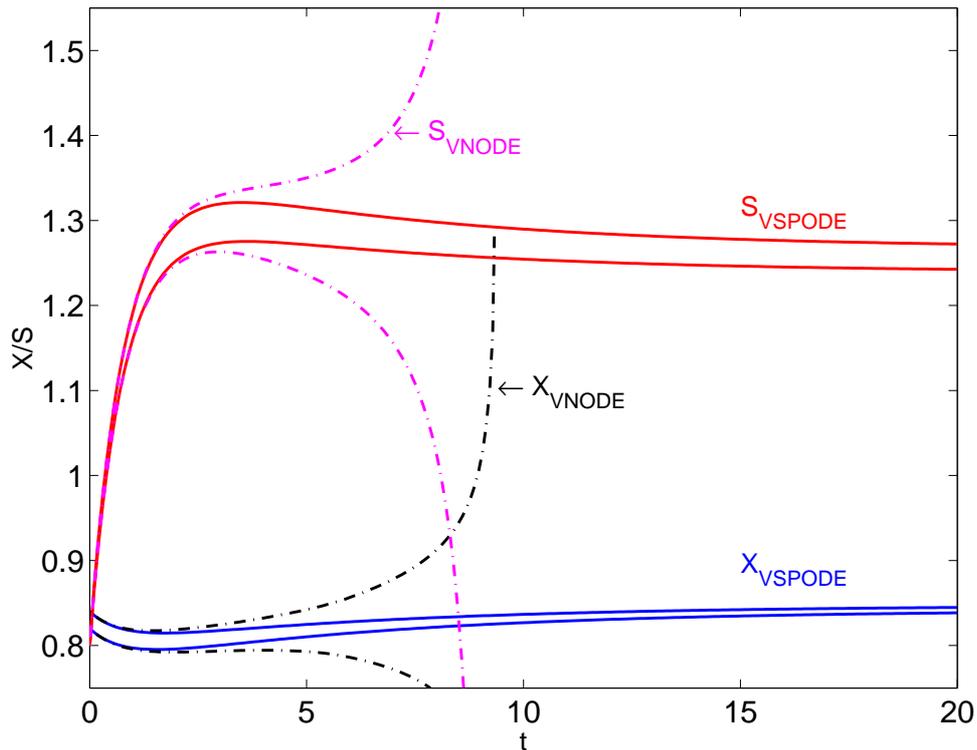


Figure 1. Enclosures for bioreactor state using the Monod law

certain number of equal-sized sub-boxes and then used VNODE to determine the solution for each sub-box. The final solution enclosure is then the union of all the enclosures resulting from each sub-box. Results showing the final solution enclosures ($t_N = 20$) and their widths, as determined using VSPODE (with no box subdivision) and VNODE with an increasing number of sub-boxes, are given in Table II for the Monod law. For example, VNODE-1000 in Table II indicates the use of 1000 sub-boxes in VNODE. Even with 1000 sub-boxes, the solution enclosure determined by VNODE is still significantly wider than that obtained from a single calculation with VSPODE, and requires about 200 times more computational time.

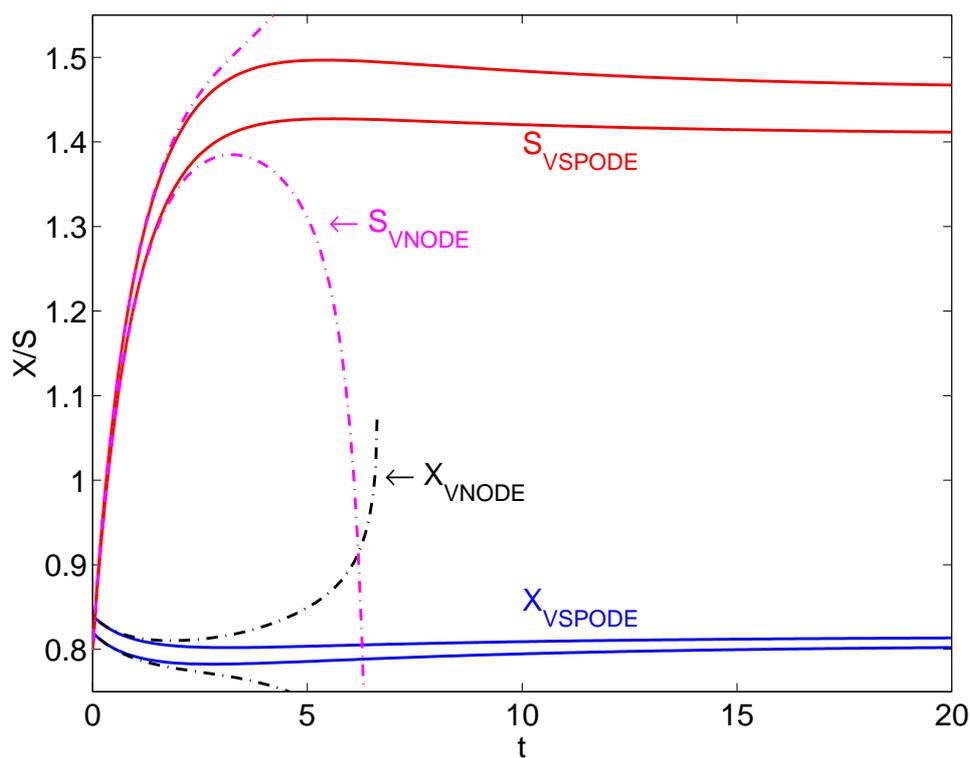


Figure 2. Enclosures for bioreactor state using the Haldane law

Table II. Results for the Monod law, showing final enclosures ($t_N = 20$).

Method	Enclosure	Width	CPU time (s)
VSPODE	[0.8386, 0.8450]	0.0064	1.34
	[1.2423, 1.2721]	0.0298	
VNODE-343	[0.8359, 0.8561]	0.0202	68.6
	[1.2309, 1.2814]	0.0505	
VNODE-512	[0.8375, 0.8528]	0.0153	102.8
	[1.2331, 1.2767]	0.0436	
VNODE-1000	[0.8380, 0.8502]	0.0122	263.1
	[1.2359, 1.2732]	0.0373	

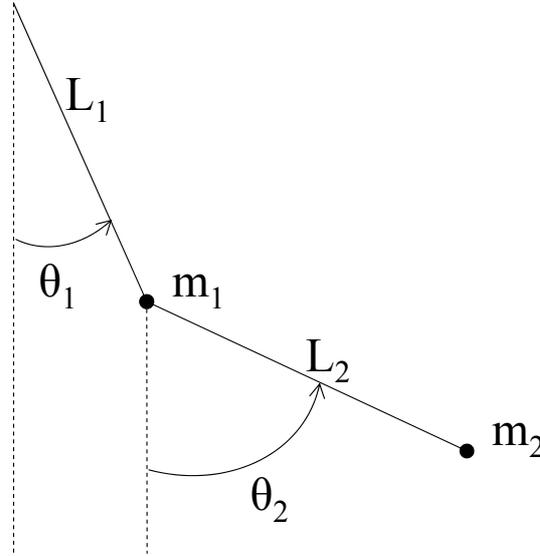


Figure 3. Schematic of double pendulum

4.2. DOUBLE PENDULUM

In this problem, we consider the motion of a double pendulum, as depicted in Fig. 3. This system is described by the nonlinear state equations:

$$\begin{aligned}
 \dot{\theta}_1 &= \omega_1 \\
 \dot{\theta}_2 &= \omega_2 \\
 \dot{\omega}_1 &= \frac{-g(2m_1 + m_2) \sin \theta_1 - m_2 g \sin(\theta_1 - 2\theta_2) - 2m_2 \sin(\theta_1 - \theta_2) [\omega_2^2 L_2 - \omega_1^2 L_1 \cos(\theta_1 - \theta_2)]}{L_1 [2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2)]} \\
 \dot{\omega}_2 &= \frac{2 \sin(\theta_1 - \theta_2) [\omega_1^2 L_1 (m_1 + m_2) + g(m_1 + m_2) \cos \theta_1 + \omega_2^2 L_2 m_2 \cos(\theta_1 - \theta_2)]}{L_2 [2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2)]},
 \end{aligned} \tag{17}$$

where θ_1 and θ_2 are the angles of the pendulum rods ($0 =$ vertical downwards, counter-clockwise is positive), and ω_1 and ω_2 are the angular velocities of the top and bottom rod, respectively. The mass parameters are set to $m_1 = m_2 = 1$ kg and the length parameters are set to $L_1 = L_2 = 1$ m. The parameter g is the local acceleration of gravity, which varies with latitude (greatest at the poles, lowest at the equator) and altitude. In this problem, we will treat g as an uncertain parameter in the interval $[9.79, 9.81]$ m/s². This corresponds roughly to the variation in the sea level value between 25° and 49° latitude (i.e., spanning the contiguous United States). The initial conditions determine the amount of potential and kinetic energy given to the system. We consider two set of initial values: 1) a relatively high-energy case with initial state of $(\theta_1, \theta_2, \omega_1, \omega_2)_0 = (0.75\pi, 0.5\pi, 0, 0)$ and 2) a relatively low-energy case with initial state of $(\theta_1, \theta_2, \omega_1, \omega_2)_0 = (0, -0.25\pi, 0, 0)$.

Enclosures of the state variables for both cases were computed using VSPODE with variable step size (automatically determined by program). The results for θ_1 and θ_2 are shown in Fig. 4 for

the high-energy case and Fig. 5 for the low-energy case. The computational times were 8.1 and 12.7 seconds, respectively. For the high-energy case, good enclosures were maintained through two full rotations of the lower pendulum and one of the upper. For the low-energy case, good enclosures were maintained through several cycles of motion. The enclosures of all state variables at some time instances, as well as the break-down time, are shown in Table III and Table IV.

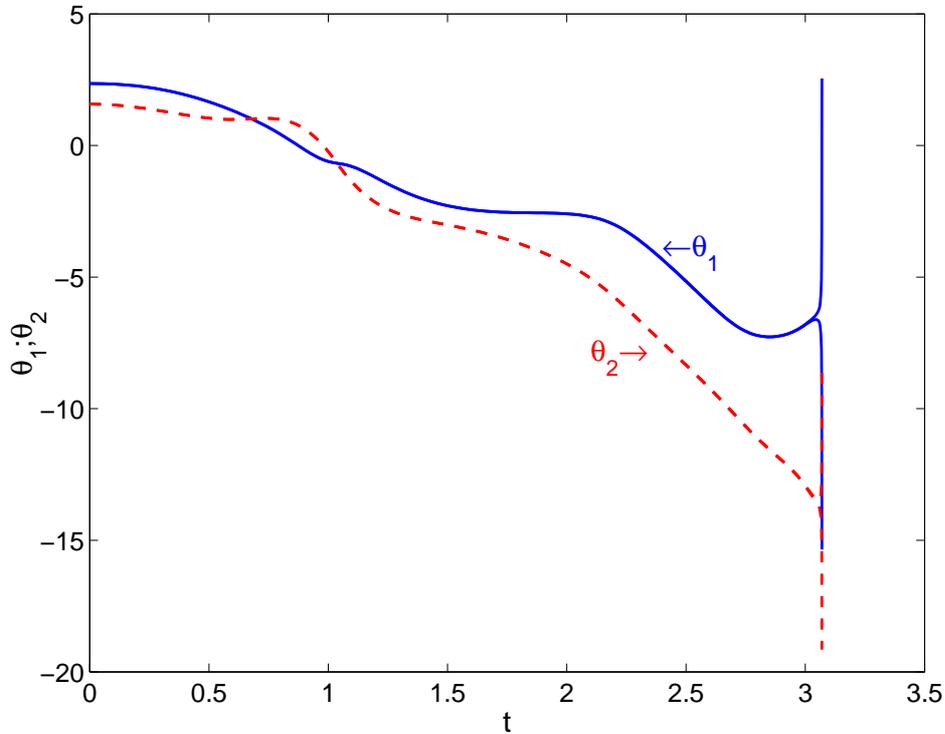


Figure 4. Enclosures of θ_1 and θ_2 for the double pendulum model, high-energy case.

Table III. Enclosures for the double pendulum model, high-energy case.

t	Enclosure			
	θ_1	θ_2	ω_1	ω_2
0.5	[1.6563, 1.6579]	[1.0368, 1.0372]	[-3.0472, -3.0408]	[-0.9327, -0.9298]
1.0	[-0.6093, -0.6067]	[-0.2392, -0.2277]	[-2.5851, -2.5397]	[-11.3007, -11.2407]
1.5	[-2.2913, -2.2883]	[-3.0230, -3.0203]	[-1.9772, -1.9647]	[-1.7512, -1.7474]
2.0	[-2.5990, -2.5978]	[-4.5055, -4.4958]	[-0.5943, -0.5778]	[-4.7957, -4.7668]
2.5	[-5.1731, -5.1512]	[-8.3532, -8.3308]	[-8.5972, -8.5801]	[-8.7733, -8.7566]
3.0	[-6.8254, -6.8000]	[-12.9548, -12.9121]	[6.3803, 6.4705]	[-12.7200, -12.5692]
3.07	FAIL			

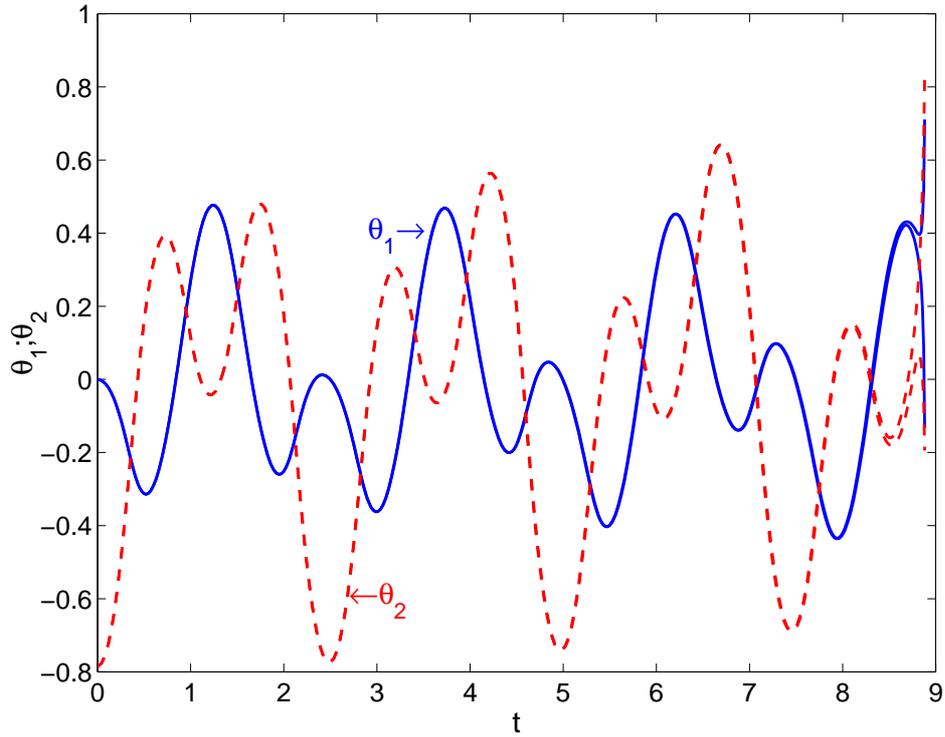


Figure 5. Enclosures of θ_1 and θ_2 for the double pendulum model, low-energy case.

Table IV. Enclosures for the double pendulum model, low-energy case.

t	Enclosure			
	θ_1	θ_2	ω_1	ω_2
1.0	[0.2688, 0.2704]	[0.1153, 0.1167]	[1.5710, 1.5734]	[-1.3996, -1.3971]
2.0	[-0.2517, -0.2509]	[0.1744, 0.1792]	[0.3534, 0.3688]	[-2.3539, -2.3340]
3.0	[-0.3623, -0.3622]	[0.1393, 0.1444]	[0.0310, 0.05463]	[1.6578, 1.6810]
4.0	[0.2161, 0.2224]	[0.3409, 0.3483]	[-1.5371, -1.5321]	[1.7902, 1.8019]
5.0	[-0.0034, -0.0004]	[-0.7395, -0.7382]	[-0.6020, -0.5848]	[0.2325, 0.2808]
6.0	[0.2927, 0.3014]	[-0.0856, -0.0818]	[1.3907, 1.4213]	[-0.6345, -0.5891]
7.0	[-0.0976, -0.0922]	[0.1977, 0.2179]	[0.7267, 0.7672]	[-2.8614, -2.8031]
8.0	[-0.4260, -0.4213]	[0.1059, 0.1150]	[0.3924, 0.4578]	[0.7548, 0.8205]
8.89	FAIL			

5. Concluding Remarks

We have demonstrated a new method for obtaining validated solutions of initial value problems for ODEs with interval-valued parameters and initial values. The dependence of the solution on t is handled using ITS methods, as in VNODE (Nedialkov et al., 2001). However, the dependence on the parameter vector θ and the initial state \mathbf{x}_0 is handled through a novel use of Taylor models of the form described by Makino and Berz (Makino and Berz, 1996; Makino and Berz, 2003). Numerical results on a bioreactor kinetics problem and a double pendulum motion problem demonstrate that this approach provides a very efficient way to obtain a tight enclosure of all possible solutions to a parametric ODE system under uncertain conditions.

Acknowledgements

This work was supported in part by the State of Indiana 21st Century Research and Technology Fund under Grant #909010455, and by the Department of Energy under Grant DE-FG02-05CH11294.

References

- Bastin, G. and D. Dochain: 1990, *On-line Estimation and Adaptive Control of Bioreactors*. Amsterdam: Elsevier.
- Berz, M. and K. Makino: 1998, 'Verified Integration of ODEs and Flows Using Differential Algebraic Methods on High-Order Taylor Models'. *Reliable Computing* **4**, 361–369.
- Eijgenraam, P.: 1991, 'The Solution of Initial Value Problems Using Interval Arithmetic'. Technical Report Mathematical Centre Tracts No. 144, Stichting Mathematisch Centrum, Amsterdam.
- Hansen, E. and G. W. Walster: 2004, *Global Optimization Using Interval Analysis*. New York: Marcel Dekker.
- Jaulin, L., M. Kieffer, O. Didrit, and É. Walter: 2001, *Applied Interval Analysis*. London: Springer-Verlag.
- Kearfott, R. B.: 1996, *Rigorous Global Search: Continuous Problems*. Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Lin, Y. and M. A. Stadtherr: 2005, 'Validated Solutions of Initial Value Problems for Parametric ODEs'. Submitted.
- Lohner, R. J.: 1988, 'Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen'. Ph.D. thesis, Universität Karlsruhe.
- Lohner, R. J.: 1992, 'Computations of Guaranteed Enclosures for the Solutions of Ordinary Initial and Boundary Value Problems'. In: I. G. J. Cash (ed.): *Computational Ordinary Differential Equations*. Oxford: Clarendon Press, pp. 425–435.
- Makino, K.: 1998, 'Rigorous Analysis of Nonlinear Motion in Particle Accelerators'. Ph.D. thesis, Michigan State University, East Lansing, Michigan, USA.
- Makino, K. and M. Berz: 1996, 'Remainder Differential Algebras and Their Applications'. In: M. Berz, C. Bischof, G. Corliss, and A. Griewank (eds.): *Computational Differentiation: Techniques, Application, and Tools*. Philadelphia: SIAM, pp. 63–74.
- Makino, K. and M. Berz: 1999, 'Efficient Control of the Dependency Problem Based on Taylor Model Methods'. *Reliable Computing* **5**, 3–12.
- Makino, K. and M. Berz: 2003, 'Taylor Models and Other Validated Functional Inclusion Methods'. *Int. J. Pure Appl. Math.* **4**, 379–456.
- Moore, R. E.: 1966, *Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall.

- Nedialkov, N. S., K. R. Jackson, and G. F. Corliss: 1999, 'Validated Solutions of Initial Value Problems for Ordinary Differential Equations'. *Appl. Math. Comput.* **105**, 21–68.
- Nedialkov, N. S., K. R. Jackson, and J. D. Pryce: 2001, 'An Effective High-Order Interval Method for Validating Existence and Uniqueness of The Solution of an IVP for an ODE'. *Reliable Computing* **7**, 449–465.
- Neumaier, A.: 1990, *Interval Methods for Systems of Equations*. Cambridge, England: Cambridge University Press.
- Neumaier, A.: 2002, 'Taylor Forms—Use and Limits'. *Reliable Computing* **9**, 43–79.
- Revol, N., K. Makino, and M. Berz: 2005, 'Taylor Models and Floating Point Arithmetic: Proof That Arithmetic Operations Are Bounded in COSY'. *J. Logic Algebr. Progr.* **64**, 135–154.

Online Implementation of a Robust Controller using Hybrid Global Optimization Techniques

Paluri S. V. Nataraj and Nandkishor Kubal

Systems and Control Engineering Group, ACRE Building, IIT Bombay, Mumbai 400076, India

Abstract. In this work, we report the experimental implementation of a Quantitative Feedback Theory (QFT) based robust controller, designed *online* using hybrid global optimization and constraint propagation techniques. The hybrid global optimization combines interval global optimization and nonlinear local optimization methods. The constraint propagation techniques are very effective in discarding infeasible controller parameter regions in the optimization search. The obtained experimental results show the effectiveness of hybrid global optimization for the *online* design of robust control systems.

Keywords: global optimization, interval analysis, robust control

1. Introduction

Most of the practical system consists of uncertainties in the form of disturbances, measurement noise and unmodelled or imprecisely modeled dynamics. Therefore the design has to seek a control system that functions adequately over a wide range of uncertain parameters. Such a system is said to be *robust* when, it has low sensitivities, is stable over a wide range of parameter variations, and the performance stays within prescribed limit bounds in the presence of parameter variations. Sometimes the parameter variations are beyond the uncertainty bounds, then there is need of retuning (*adaptation*) of controller parameters online.

Quantitative Feedback Theory (QFT), developed by Horowitz (1993) is a frequency domain based technique for robust controller design. It converts the design specifications of a closed loop system and plant uncertainties into robust stability bounds and performance bounds on the open loop transmission of the nominal system and then synthesizes a controller by using the gain-phase loop shaping technique. Traditionally, this synthesis was done *manually* by the designer, relying on design experience and skill. Recently, several researchers have attempted to *automate* this step, see, for instance, (Ballance and Gawthrop, 1991; Bryant and Halikias, 1995; Chait et al., 1999; Gera and Horowitz, 1980; Thompson and Nwokah, 1994)

The main drawback of the approaches cited above lies in attempting to solve a complicated nonlinear optimization problem using convex or linear programming techniques, which generally leads to conservative designs. To overcome these difficulties, Chen *et al.* (1998) reformulated the problem as one of parameter optimization of a fixed order controller and used genetic algorithms for obtaining the solutions. However, it is well known that with genetic algorithms one may obtain a *local* minimum instead of the *global* minimum (Dallwig et al., 1997). Moreover, genetic algorithms tend to become slower as one tries to increase the probability of success.

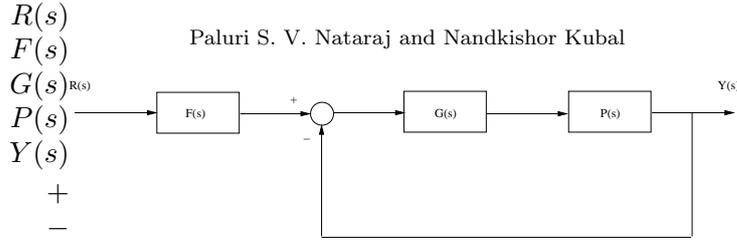


Figure 1. Two Degree of Freedom Structure for QFT.

In this paper, we used an efficient method for automatic loop shaping in QFT, proposed in (Nataraj and Kubal, 2005). The QFT controller synthesis problem is posed as a constrained optimization problem, where the objective function is the high frequency gain of the controller, and the constraint set for the optimization is the set of possibly nonconvex, nonlinear magnitude-phase QFT bounds at the various design frequencies. The method uses hybrid optimization techniques and constraint propagation ideas to solve the optimization problem. The hybrid optimization part efficiently combines interval global optimization (Moore, 1979; Ratschek and Rokne, 1988; Hansen, 1992; Kearfott, 1996) and nonlinear local optimization methods. The method supplement the optimization tools with a new so-called *quick solution* approach, developed based on ideas of constraint propagation techniques. The quick solution approach can quickly discard sizable portions of the infeasible controller parameter regions using simple arithmetic calculations.

In the present work, automatic loop shaping using *hybrid global optimization and constraint propagation* is used for the experimental implementation of QFT based robust adaptive controller on a coupled tank system.

The paper is organized as follows: Section 2 deals with the background of QFT. Problem formulation is given in Section 3. Section 4 give details of hybrid global optimization and constraint propagation. Case study of coupled-tank system is described in Section 5.

2. Overview of QFT

Consider a two degree freedom feedback system configuration (see Fig 1), where $G(s)$ and $F(s)$ are the controller and prefilter respectively. The uncertain plant $P(s)$ is given by $P(s) \in \{P(s, \lambda) : \lambda \in \boldsymbol{\lambda}\}$, where $\lambda \in \mathbb{R}^l$ is a vector of plant parameters whose values vary over a parameter box $\boldsymbol{\lambda}$

$$\boldsymbol{\lambda} = \{\lambda \in \mathbb{R}^l : \lambda_i \in [\underline{\lambda}_i, \overline{\lambda}_i], \underline{\lambda}_i \leq \overline{\lambda}_i, i = 1, \dots, l\}$$

This gives rise to a parametric plant family or set

$$\mathcal{P} = \{P(s, \lambda) : \lambda \in \boldsymbol{\lambda}\}$$

The open loop transmission function is defined as

$$L(s, \lambda) = G(s)P(s, \lambda) \tag{1}$$

and the nominal open loop transmission function is

$$L_0(s) = G(s)P(s, \lambda_0) \tag{2}$$

The objective in QFT is to synthesize $G(s)$ and $F(s)$ such that the various stability and performance specifications are met for all $P(s) \in \mathcal{P}$. In general following specifications are considered in QFT:

1. Robust stability margin

$$\left| \frac{L(j\omega)}{1 + L(j\omega)} \right| \leq \omega_s$$

2. Robust tracking performance

$$|T_L(j\omega)| \leq \left| \frac{F(j\omega)L(j\omega)}{1 + L(j\omega)} \right| \leq |T_U(j\omega)|$$

3. Robust input disturbance rejection performance

$$\left| \frac{G(j\omega)}{1 + L(j\omega)} \right| \leq \omega_{d_i}(w)$$

4. Robust output disturbance rejection performance

$$\left| \frac{1}{1 + L(j\omega)} \right| \leq \omega_{d_o}(w)$$

In practice, the objective is to satisfy the given specifications over a finite design frequency set Ω . The main steps of QFT design specifications are

1. **Generating templates:** For a given uncertain plant $P(s) \in \mathcal{P}$, at each design frequency $\omega_i \in \Omega$, calculate the value set of the plant $P(j\omega_i)$ in the complex plane.
2. **Computation of QFT bounds:** At each design frequency ω_i , combines the stability and performance specifications with the plant templates which results in the stability margin and performance bounds. The bound at ω_i is denoted as $B_i(\angle L_0(j\omega), \omega_i)$ or simply B_i
3. **Design of Controller :** Design a controller $G(s)$ such that
 - The bound constraints at each design frequency ω_i are satisfied.
 - The nominal closed loop system is stable.
4. **Design of Prefilter:** Design a prefilter $P(s)$ such that the robust tracking specifications are satisfied.

3. Problem Formulation

We consider the controller structure in the gain-pole-zero form as

$$G(s, x) = \frac{k_G \prod_{i_1=1}^{n_z} (s + z_{i_1})}{\prod_{k_1=1}^{n_p} (s + p_{k_1})} \quad (3)$$

where

$$x = (k_G, z_1, \dots, z_{n_z}, p_1, \dots, p_{n_p}) \quad (4)$$

is the controller parameter vector. The magnitude and phase functions of $G(s, x)$ are defined as

$$G_{mag}(\omega, x) = |G(s, x)|; G_{ang}(\omega, x) = \angle G(s, x) \quad (5)$$

Now, the QFT controller synthesis problem can be formulated as: Given the QFT bounds and the nominal plant, develop a controller automatically which provides nominal closed loop stability, satisfies all the bound constraints, with minimum high frequency controller gain k_G . Minimization of the high frequency gain of the controller tends to reduce the amplification of the sensor noise in the high frequency range, as shown in (Horowitz, 1993).

The QFT synthesis problem can be posed as a constrained optimization problem

$$\begin{aligned} \min_{x \in \mathbf{x}} f &= k_G \\ \text{subject to } H(x) &\leq 0 \end{aligned} \quad (6)$$

- x is the vector of controller parameters, \mathbf{x} is some suitably specified initial search box of controller parameter values.
- $H(x) = \{h_i(x)\}$ is set of bound constraints at each design frequency ω_i

$$\text{single valued upper bound constraint : } h_i^u(x) = |L_0(j\omega_i, x)| - B_i(\angle L_0(j\omega_i, x), \omega_i) \leq 0 \quad (7)$$

$$\text{single valued lower bound constraint : } h_i^l(x) = B_i(\angle L_0(j\omega_i, x), \omega_i) - |L_0(j\omega_i, x)| \leq 0 \quad (8)$$

A multiple valued bound constraint denoted as h_i^{ul} can be split into a single-valued upper bound constraint h_i^u and a single-valued lower bound constraint h_i^l , and then the condition of both the bounds consider together.

- The bound constraint on the controller parameter vector, i.e. the controller parameter values should lie in the initial search region.
- The nominal closed loop stability test is based on finding out the zeros of $1 + L(s, z_0, \lambda_0)$ for some $z_0 \in \mathbf{z} \subseteq \mathbf{x}$

4. Hybrid Global Optimization

Let $\mathbf{z} = (\mathbf{k}_G, \mathbf{z}_1, \dots, \mathbf{z}_{n_z}, \mathbf{p}_1, \dots, \mathbf{p}_{n_p})$ be the controller parameter box. Let $G_{mag}(\omega_i, \mathbf{z})$ and $G_{phase}(\omega_i, \mathbf{z})$ denote the natural interval extensions of controller magnitude and phase functions respectively. The natural interval extensions of nominal open loop transfer function magnitude and phase are defined as

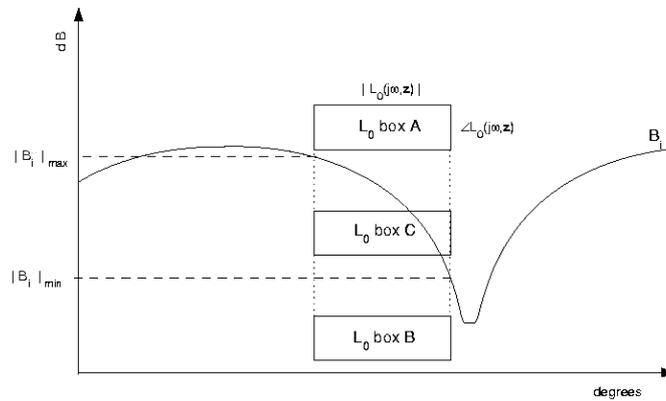
$$L_{0mag}(\omega_i, \mathbf{z}) = |L_0(j\omega_i, \mathbf{z})| = G_{mag}(\omega_i, \mathbf{z}) |P(\omega_i, \lambda_0)| \quad (9)$$

$$L_{0phase}(\omega_i, \mathbf{z}) = \angle L_0(j\omega_i, \mathbf{z}) = G_{phase}(\omega_i, \mathbf{z}) + \angle P(\omega_i, \lambda_0) \quad (10)$$

The evaluations of the natural interval extensions at a given frequency ω_i give magnitude and phase intervals that define a box-like region in the Nichols chart. This is called as the L_0 box at ω_i .

The algorithm proposed in (Nataraj and Kubal, 2005) mainly consists of seven major components: a quick solution approach, feasibility test, local optimization call, initialization, list sorting and handling, a bisection strategy and a termination criteria.

1. **Feasibility test:** Based on the location of the L_0 box w.r.t. bounds B_i , the parameter box \mathbf{z} is determined as feasible, infeasible or indeterminate at ω_i , see Fig. 2. The $flag_z$ represents the feasibility of parameter box \mathbf{z} . The details for the feasibility test are given in sec. 4.1.
2. **Quick Solution approach:** The quick solution approach discards the portion of the controller parameter box \mathbf{z} based on the location of the L_0 box w.r.t. the bounds (for details see sec. 4.3).
3. **Initialization:** The current processing box \mathbf{z} is assigned to the initial search box. The quick solution and 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 $\forall \tilde{\mathbf{z}} \in \mathbf{z}$, hence, the algorithm exits and print the message 'No solution exist in the given initial search box'. Else, a list L is initialized with triple $(\mathbf{z}, z, flag_z)$, where $z = \inf \mathbf{z}(1)$ is the minimum value of the high frequency gain based on the current parameter box \mathbf{z} .
4. **Local optimization call:** A constrained local optimization routine is called to solve the constrained optimization problem (6). For details see sec. 4.2.
5. **Bisection:** At each iteration, the box \mathbf{z} of leading triple is bisected into two subboxes \mathbf{v}^1 and \mathbf{v}^2 .
6. **List sorting and handling:** At each iteration, the leading triple is deleted from the list L and the indeterminate bisected triples are added into the list. The list is sorted and arranged in the non decreasing order of the value of objective function.
7. **Termination:**
 - a) As the list is sorted and arranged in the non decreasing order of the value of objective function z at each iteration, the leading triple always contains the minimum value of the objective function. Hence, at any iteration, if the box \mathbf{z} of the leading triple is feasible, then the algorithm can be terminated by printing the optimal controller parameter box \mathbf{z}



PSfrag replacements

$$|B_i|_{max}$$

Figure 2. Feasibility conditions for different locations of L_0 box w.r.t. single valued lower bound at ω_i . Box A shows feasible case, box B shows infeasible case and box C shows the indeterminate case.

- b) If the relative gain width of the box \mathbf{z} of leading triple is less than a specified relative gain tolerance, and the box \mathbf{z} contains the feasible parameter vector (i.e. feasible local solution) then the algorithm can be terminated by printing the optimal parameter vector z_{local}

4.1. FEASIBILITY CHECK

The feasibility check for a controller parameter box \mathbf{z} consists of checks for the bound constraints satisfaction in (7) and (8) at a given ω_i , $i = 1, \dots, n$.

4.1.1. Feasibility check for bound satisfaction

Let $|B_i|_{max}$ and $|B_i|_{min}$ be the top most and bottom most value of the single valued lower bound for the entire phase interval $\angle L_0(j\omega_i, \mathbf{z})$. Based on the location of the L_0 box w.r.t. the single valued lower bound one of the following cases arises (see Fig.2)

1. If the entire L_0 box lies on or above $|B_i|_{max}$ (box A in Fig. 2) then h_i^l is satisfied for any controller parameter vector $z \in \mathbf{z}$, so that the entire box \mathbf{z} is feasible at ω_i .
2. If the entire L_0 box lies below $|B_i|_{min}$ (box B in Fig. 2) then h_i^l is not satisfied for any controller parameter vector $z \in \mathbf{z}$, so that the entire box \mathbf{z} is infeasible at ω_i .
3. Else box \mathbf{z} is indeterminate (box C in Fig. 2).

4.2. LOCAL OPTIMIZATION

Local optimization gives an early knowledge of the approximate global minimum. However, the main difficulty is to decide of when to call a local optimization algorithm in a hybrid algorithm. If local optimization is called at each algorithmic iteration, then the computational costs will grow dramatically. Hence, the following decision rule is made regarding when to call the local optimization routine.

- Let z , be any parameter vector belong to the parameter box \mathbf{z} .
- z is compared with all previous starting points of local optimization, say z_v ,
- If z is sufficiently different (say, for instance more than 10%) from all previous starting points z_v , then call the local optimization routine for z .

4.3. QUICK SOLUTION

We can easily show from (2), (3) that the magnitude and phase of L_0 vary *monotonically* over the gain, zero and pole intervals. Further, from Fig. 3, we also observe that the coordinate $(\inf \angle L_0, \sup |L_0|)$ is contributed by supremum values of gain and zero intervals and infimum values of pole intervals, while the coordinate $(\sup \angle L_0, \inf |L_0|)$ is contributed by infimum values of gain and zero intervals and supremum values of pole intervals

The proposed *quick solution* approach uses these simple observations and a few arithmetic calculations for discarding infeasible parts of gain, pole and zero intervals. In general, optimization techniques alone would take perhaps many iterations to achieve the same.

5. Case study

5.1. PLANT DESCRIPTION

The coupled tank system whose schematic is given in Fig. 4 consists of two hold-up tanks which are coupled by an orifice. Water is pumped in to the first tank by variable speed pump. The orifice allows this water to flow into the second tank and hence out to a reservoir. The aim is to control the water level in the second tank by changing the flow rate to the first tank by varying the speed of the pump. The speed of the pump is varied by varying the control voltage (0-10V) to the pump. The liquid level in the tank is measured using a depth sensor whose output is voltage (0-10V), which is proportional to the level.

The input to the plant is the voltage to the variable speed pump and the output is the water level in the second tank in terms of voltage signal.

The control voltage to the pump motor drive is from a digital computer along with the Advantech 5000 series data acquisition system. The mentioned data acquisition comprises 8-channel analog input module and 4-channel Analog output module. The analog input channel accepts the signal of 0 – 10 volts. The analog output channels can generate an output of 0 – 10 volts. Communication

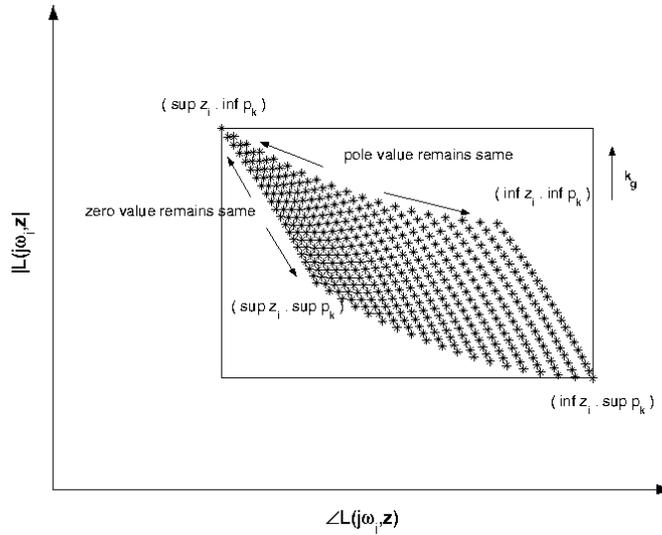


Figure 3. Variation of $|L_0(j\omega_i, \mathbf{z})|$, $\angle L_0(j\omega_i, \mathbf{z})$ w.r.t. gain, zero and pole intervals. The outer rectangle shows the L_0 box.

PSfrag replacements

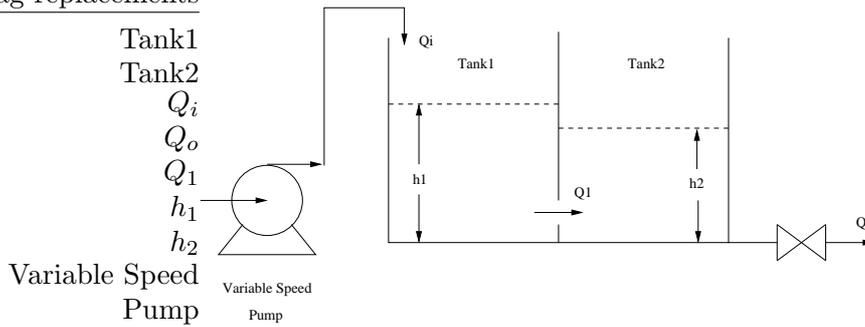


Figure 4. Schematic of Coupled Tank System.

between data acquisition system and the digital computer is via serial port. The control design algorithm is implemented on a PC in Microsoft FORTRAN 95 with interval arithmetic support INTLIB (Kearfott et al., 1994).

5.2. REAL-TIME PARAMETER ESTIMATION

On-line determination of process parameters is a key element in adaptive control system. In the present work recursive least square method is used for parameter estimation. In recursive identification method, the parameter estimates are computed recursively in time. This method has a small requirement on memory since only a modest amount of information is stored. This amount will not increase with time.

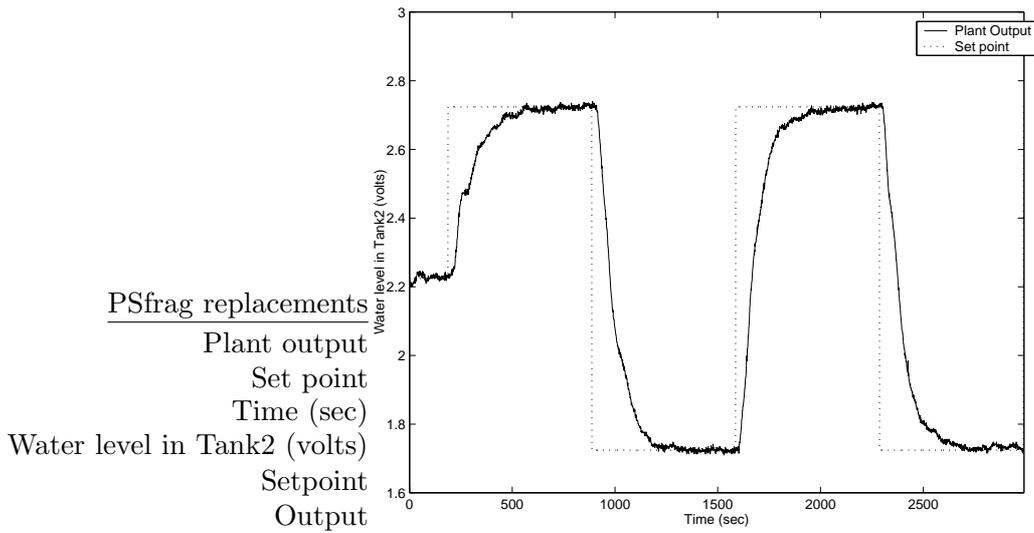


Figure 5. Experimental closed-loop responses to setpoint changes for the coupled-tank system.

5.3. CONTROLLER DESIGN AND IMPLEMENTATION

For the design of robust QFT controller, the closed-loop specifications include the robust QFT stability margins and tracking performance which are specified by

$$\left| \frac{L(j\omega)}{1 + L(j\omega)} \right| \leq 3dB$$

and

$$|T_L(\omega)| \leq \left| \frac{F(j\omega)L(j\omega)}{1 + L(j\omega)} \right| \leq |T_U(\omega)|$$

respectively, where

$$T_U(s) = \frac{16.67s + 1}{2140s^2 + 56.44s + 1}$$

and

$$T_L(s) = \frac{1}{4.495 \times 10^4 s^3 + 4740s^2 + 139.2s + 1}$$

A second order model structure is selected for the coupled-tank plant, whose parameters are estimated *online* using recursive least square method. The method mentioned in sec. 4 is used to design the controller *online*. The implementation results are shown in the Figs. 5 and 6.

It can be noticed in Fig. 6 that the obtained closed-loop responses satisfy the given time-domain specifications.

References

- Ballance, D. J. and P. J. Gawthrop. Control Systems Design Via a QFT approach. *Proceedings of the IEE conference Control*, 1:476–481, 1991.

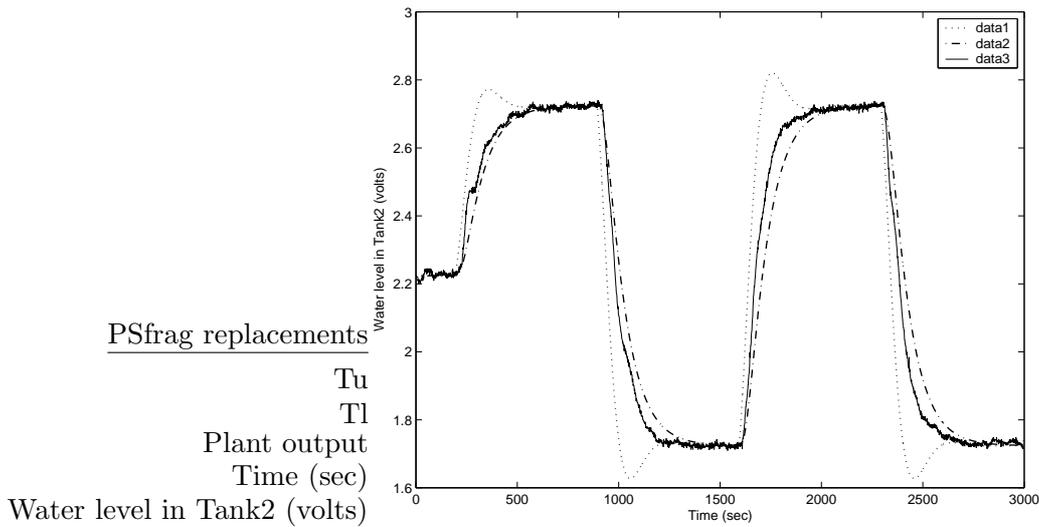


Figure 6. Experimental responses along with the given time-domain specifications.

- Bryant, G.F. and G.D Halikias. Optimal loop-shaping for systems with large parameter uncertainty via linear programming. *International journal of Control*, 62(3):557–568, 1995.
- Chait Yossi, Qain Chen, and C. V. Hallot. Automatic Loop-Shaping of QFT Controllers Via Linear Programming. *Trans. of the ASME Journal of Dynamic Systems, Measurement and Control*, 121:351–357, 1999.
- Chen Wenhua, Donald J. Ballance, and Yun Li. Automatic Loop-Shaping in QFT Using Genetic Algorithms. *Proceedings of 3rd Asia-Pacific Conference on Control and Measurement*, 63–67, 1998.
- Dallwig, S., A. Neumaier, and H. Schichl. *GLOPT - A Program for Constrained Global Optimization*. In I. Bomze et al., eds., *Developments in Global Optimization*, pages 19–36, Kluwer, Dordrecht, 1997.
- Gera, A. and I. M. Horowitz. Optimization of the loop transfer function. *International journal of Control*, 31:389–398, 1980.
- Hansen, E. *Global optimization using Interval Analysis*. Marcel Dekker, 1992.
- Horowitz, I. M. *Quantitative feedback design theory (QFT)*. QFT Publications, 1993.
- Kearfott, R. B., M. Dawande, K. Du, and Ch. Hu. INTLIB: A portable FORTRAN 77 Interval Standard Function Library *ACM Trans. Math. Software*, 20(4):447–459, 1994.
- Kearfott, R. B. *Rigorous global search: continuous problems* Kluwer Academic Publishers, Dordrecht, 1996
- Moore, R.E. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.
- Nataraj S. V. Paluri and Nandkishor Kubal. Automatic loop shaping in QFT using hybrid optimization and constraint propagation. submitted to *Int. journal of Robust and Nonlinear Control*.
- Ratschek, H. and J. Rokne *New computer methods for global optimization* Wiley, New York, 1988
- Thompson, D. F. and O. D. I. Nwokah. Analytical Loop Shaping Methods in Quantitative Feedback Theory. *Trans. of the ASME Journal of Dynamic Systems, Measurement and Control*, 116(2):169–177, 1994.

Reduction in Space Complexity And Error Detection/Correction of a Fuzzy controller

F. Vainstein¹, E. Marte², V. Osoria³, R. Romero⁴

¹Georgia Institute of Technology, 210 Technology Circle Savannah, GA 31407
feodor.vainstein@gtsav.gatech.edu

²Pontificia universidad Católica Madre y Maestra
Autopista Duarte Km. 1 ½, Santiago de los Caballeros, Dominican Republic
edwin.marte@engineeringgroup.com

³Universidad Tecnológica de Santiago, Av. Estrella Sadhalá, Santiago de los Caballeros
valentin.osoria@verizon.net.do

⁴Instituto Tecnológico de Santo Domingo, Av. Los Proceres, Galá, Santo Domingo
rafael.romero@engineeringgroup.com

Keywords: Space Complexity, Reduction, Fuzzy, Error Detection, Controller

Abstract: Fuzzy controllers prove to be very useful for practical applications, especially in the cases when there is no appropriate mathematical model of behavior of the controlled object. Control signal is computed by fuzzy controller with the use of rule base table. In this paper we propose a mathematical method for reduction in space complexity of the system by decreasing the number of address lines in the memory used to store If-Then rules. The idea of the method is to use variable radix in representing integers. We also propose incorporation of error-correcting codes in the memory used to store If-then rules without substantial increasing of space complexity and application of signature analysis for error detection/location in a fuzzy controller.

1. Introduction

Fuzzy sets and some basic ideas pertaining to their theory were first introduced in 1965 by Lofti A. Zadeh, a Professor of Electrical Engineering at the University of California a Berkeley. The development of fuzzy set theory and fuzzy logic experimented changes since their introduction. Therefore the “Fuzzy Boom” (since 1989) has been characterized by a rapid increase in successful industrial applications that have netted impressive revenues.

Major research centers have been established devoted to this field. This all been accompanied by a tremendous increase in the number of contributors as well as in the number of relevant publications, including several dedicated journals.

Braunstingl [1] developed a wall-following robot that used a fuzzy logic controller and local navigation strategy to determine its movement. The fuzzy logic controller uses the input variables to control the firing of 33 rules.

A fuzzy system developed by Surmann [2] controls the navigation of an autonomous mobile robot. The entire system has about 180 fuzzy rules that associate 30 fuzzy inputs with 11 outputs. Potentially, the number of fuzzy rules can be very large.

The contribution of this paper is to tackle the space complexity of the system by decreasing the number of addresses lines used to store If-Then rules. Also the incorporation of error correcting codes in the memory used to store If-Then rules without substantial increasing space complexity as well as application of signatures analysis for error detection/location in a fuzzy controller.

2. Fuzzification, Rule Base and Defuzzification

Fuzzy controllers follow standard procedures for their design, which consists of fuzzification, control rule base establishment, and defuzzification as shown by fig. 1. We are using a fuzzy controller with a sensor 0 and sensor k where r_0 and r_k denote the number of membership possible values.

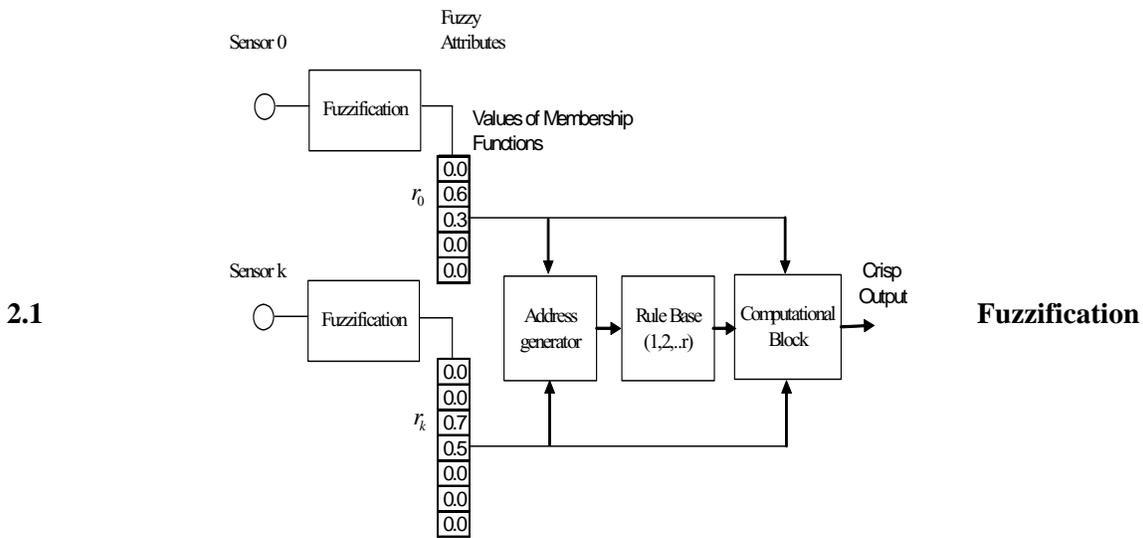


Fig. 1 Basic Fuzzy Controller Block diagram

Fuzzification is mapping from the crisp domain into the fuzzy domain. Fuzzification also means the assigning of linguistic value, defined by relative small number of membership functions to variable.

2.2 Rule Base

The rule base is in fact a big database of rules that keep the knowledge of how is best to control the system.

2.3 Computational block

The Computational Block runs the inference engine which goes through all the rules, evaluating the firing strength of each rule which in turn is proportional to the truth-value of the preconditions.

After all the rules are computed, we have the firing strength of each rule. A problem then arises - we might have several rules with similar consequents, but different firing strength. Such a situation will result with different membership values for the same output. Here the defuzzifier comes in.

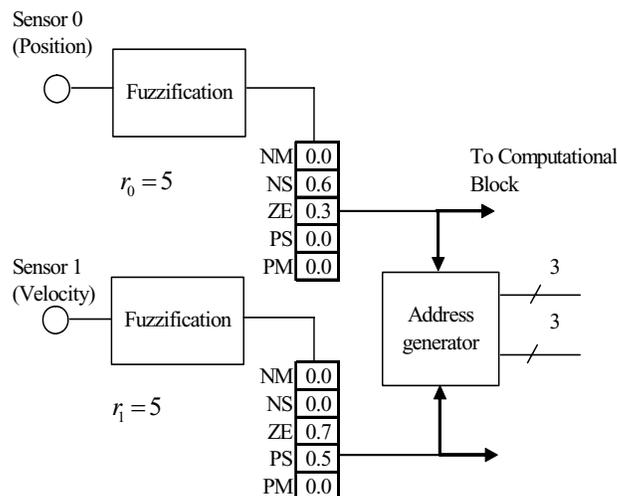
2.4 Defuzzification

The defuzzifier block task is to receive as input the membership values of the outputs, or in other words, the fuzzy outputs. Then it returns the actual numerical output, which may be a drug dose, a desired temperature, or any other variable. There are many methods of solving that problem and all of them aggregate the membership values of the outputs, in some form of an average, to find out the actual output.

3. Address Generation

Fig. 2 below shows a classical fuzzy controller implementation with two sensors, velocity and position.

Normally different values of position and velocity will trigger different and probably simultaneous rules which will lead the computational block to take a decision with these values. Base Rules in a fuzzy controller are preset values for decision taking, as an example for the controller in Fig.2 we created a 2-dimensional table where we are implementing five different decisions (1 to 5). Therefore the computational block for delivering 2 outputs in this system will need 3 bits for each output to represent these values in classical binary format (Fig. 3).



REC 2006 – Vainstein F.,Marte E.,Osoria V., and Romero R.181
 Fig. 2 Example of Address Generation

		Position				
		NM	NS	ZE	PS	PM
velocity	NM	1	4	3	2	1
	NS	5	2	1	3	2
	ZE	1	3	2	1	4
	PS	5	5	3	2	1
	PM	1	2	5	1	2
		Rule Base				

to computational block

Fig. 3 Bidimensional Table for Decision Take of the fuzzy Controller

3.1 Number of Lines and Space Complexity

Denote by N_0 the number of input address lines of the ROM storing Rule Base Table.

With the straightforward approach, the value of N_0 is obtained from the following formula:

$$N_0 = \sum_{i=0}^k [\log_2 r_i] \tag{1}$$

Example: Let $k=9, r_0 = r_1 = \dots = r_9 = 5$ Using (1) we obtain $N_0 = 30$.

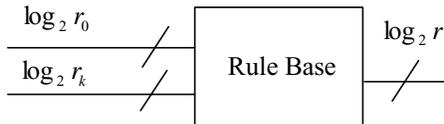


Fig. 4 Straight forward Address Generation

Let us assume that Rules Base is stored in a ROM. The space complexity of a Rom is proportional to 2^N where N is the number of address lines in the Rom (See Fig. 5).

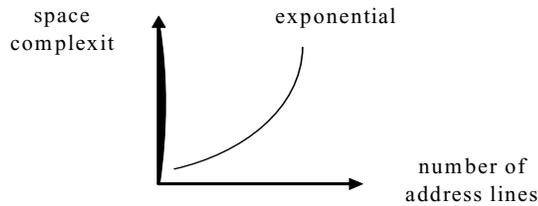


Fig. 5 Space Complexity Vs Address Lines

4. Variable Radix Numbers (Multi Radix)

In this paper we introduce a new representation of numbers. Unlike the usual decimal, and binary numbers, where the radix is fixed, in our representation the radix differ from position to position. We call this numbers Variable Radix (or Multy Radix) Numbers.

By Definition Multy Radix number can be written as follow:

$$a_r = (a_k, \dots, a_0), \tag{2}$$

Where,

$$\begin{aligned} r &= \{r_0, \dots, r_k\} \\ a_0 &\in \{0, \dots, r_0 - 1\} \\ a_1 &\in \{0, \dots, r_1 - 1\} \\ &\dots\dots\dots \\ a_k &\in \{0, \dots, r_k - 1\} \end{aligned}$$

The Multy Radix number a_r has the value

$$a_r = a_0 + a_1 r_0 + a_2 r_0 r_1 + \dots + a_k r_0 r_1 \dots r_{k-1} \tag{3}$$

Example:

Let's assume that $r_0 = 5, r_1 = 2, r_2 = 7$

Then the multi radix number 604 it then has the decimal value $a_r = 64_{10}$

Usage of multi radix numbers in a fuzzy controller will reduce the space complexity of the system.

4.1 Important Note

There exists natural one-to-one correspondence between the set of multi radix numbers and the set of fuzzy rules. It is demonstrated on Fig. 6 and Fig. 7.

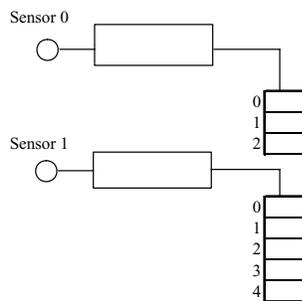


Fig. 6 Set of fuzzy Rules

		0	1	2	3	4
0	00	01	02	03	04	
	1	4	3	5	1	
1	10	11	12	13	14	
	3	4	4	1	2	
2	20	21	22	23	24	
	2	1	1	1	2	

Fig. 7 One-to-One Mapping

4.2 Multi Radix to Binary Converter

Denote by $w_1 = r_0, w_2 = r_0r_1, \dots, w_k = r_0r_1 \dots r_k$.

Then $a_r = a_0 + a_1w_1 + a_2w_2 + \dots + a_kw_k$ (4)

The block diagram of Multy Radix to Binary Converter is shown in Fig. 8 for the case of $r_0 = 5, r_1 = 5, r_2 = 3, r_3 = 6$

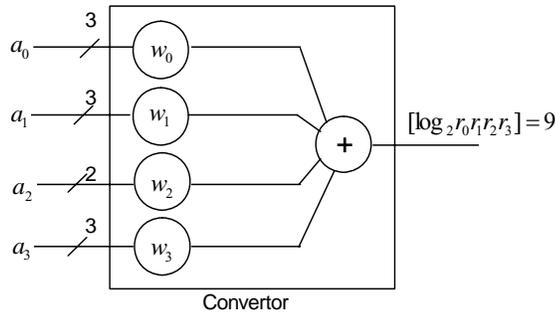


Fig. 8 Multi Radix to Binary Converter

3-input line ROMs can be used as multipliers by the constant w_i .

5. Reduction in space complexity

We can reduce the space complexity of the Rule Base by using Multi Radix numbers as shown in Fig. 9.

If we denote by N_0 the initial number of addresses lines and by N the number of addresses lines after the Multi Radix to Binary Converter then the following statement is true:

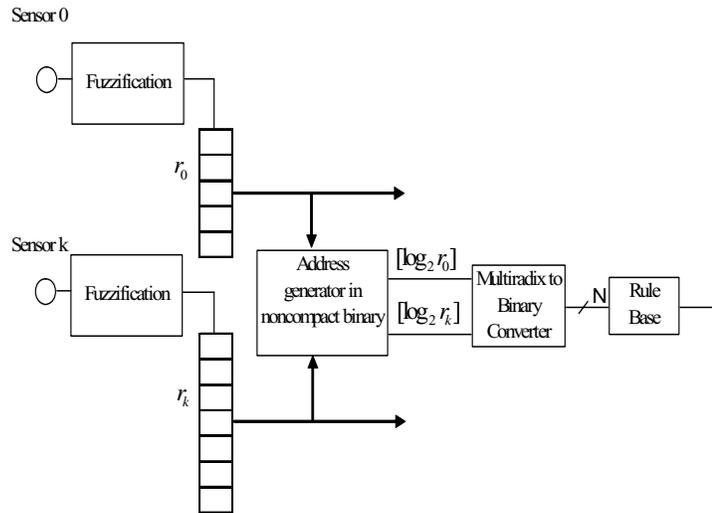


Fig. 9 Rule Base with Multi radix to Binary Converter

$$N = \lceil \log_2 r_0 r_1 \dots r_k \rceil \leq \lceil \log_2 r_0 \rceil + \dots + \lceil \log_2 r_k \rceil = N_0 \tag{5}$$

Example:

Let $k = 9, r_0 = \dots = r_9 = 5$. Then the number of initial addresses lines $N_0 = 30$.

The number of addresses lines after the Multi Radix to Binary Converter is equal to $N = \lceil \log_2 r_k \rceil = 24$

6. Data compression and error correcting codes in Rule Base

Data compression can be demonstrated by the following examples:

Example:

Suppose we have 2 Sensors $r_0 = r = 5, r = 5$. Normally, for this case it will take 15 bits for representing a single row as shown in Fig.10.

The string of numbers in the center row, as shown in Fig. 11, can be considered as a radix 5 number. Since this number has 5 digits (positions), the biggest possible number represented by this string is equal to $5^5 - 1$.

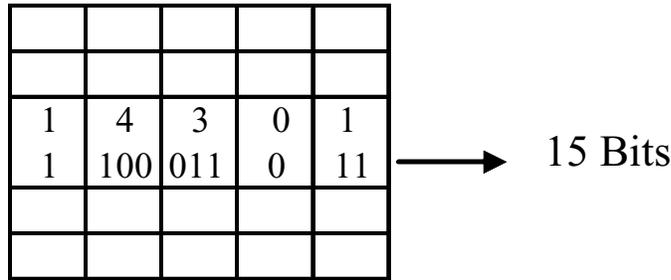


Fig. 10 Single Row 2 Sensors Representation



Fig. 11 Center Row

To convert it to binary we need $\lceil \log_2 5^5 - 1 \rceil = 12$ bits. We saved 3 bits. These bits can be used for error correction

Example:

Suppose that we have 3 sensors, $r_0 = r_1 = r_2 = 5$; $r = 5$. In this case we have a number with 25 digits. The biggest possible number $5^{25} - 1$. To convert it to binary we need 59 bits. Therefore we saved $75 - 59 = 16$ bits, see Fig. 12.

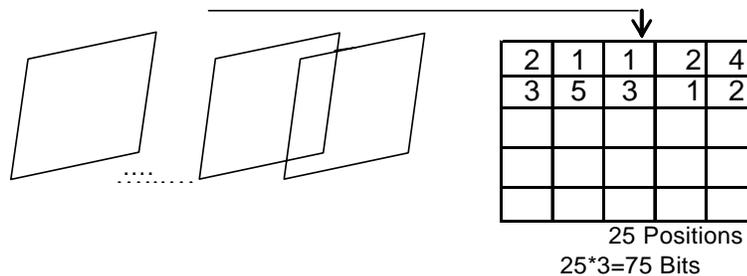


Fig. 12 3 Sensors representation

6.1 Rule Base with data compression and Error Detection/correction

The block diagram of the Rule Base with data compression and Error Detection/Correction is shown in Fig. 13

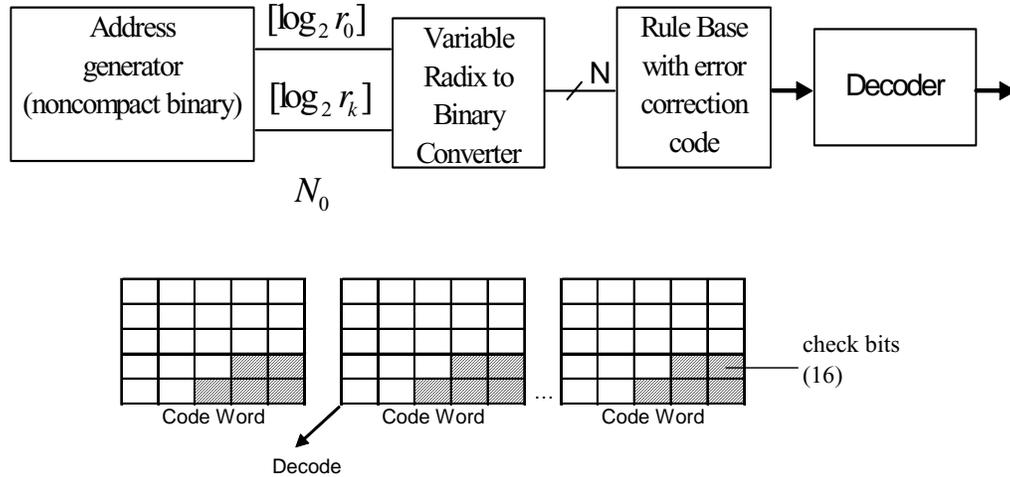


Fig. 13 Rule Base block Diagram with data compression and Error Detection/Correction

6.2 Testing of a Fuzzy Controller by signature Analysis of test Response

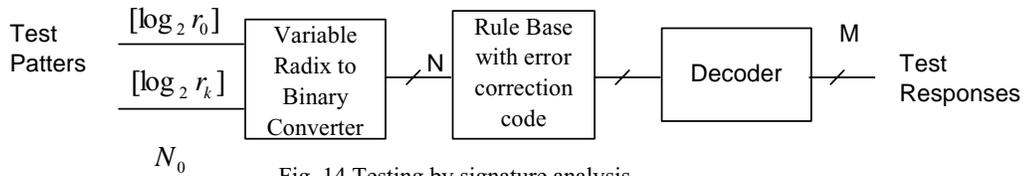


Fig. 14 Testing by signature analysis

Since initially we have N_0 address lines and after the decoder we have M address lines, we can consider a mapping

$$f : Z_2^{N_0} \longrightarrow Z_2^M \tag{6}$$

Note the Z_2^M can be considered as a field of 2^M elements $GF(2^M)$.

Error detection is performed the following way: first we precompute the signature

$$S_f = \sum \alpha_x f(x) \tag{7}$$

where $x \in Z_2^{N_0}$, $\alpha_x \in GF(2^M)$. S_f is to be considered fault-free.

The signature observed is computed with the same set of weights α_x

$$S_{\bar{f}} = \sum \alpha_x \bar{f}(x) \quad (8)$$

$$\text{Here } \bar{f} \text{ is (possibly faulty) function } \bar{f}: Z_2^{N_0} \longrightarrow Z_2^M \quad (9)$$

For testing if f and \bar{f} are the same we compute the signatures. If the signatures S_f and \bar{S}_f are the same, the test is passed.

7. Conclusion

In this paper we introduce a new representation of numbers – Variable Radix Number system. Using a Variable Radix Number system we decreased the number of addresses lines in a ROM that is used to store If-Then rules, thus reducing the space complexity of a fuzzy controller. Also we incorporated error correcting codes in the memory used to store If-Then rules without substantial increasing space complexity.

References

- Braunsting R., J. Mujika, and J. P. Uribe, “A wall following robot with a fuzzy logic controller optimized by a genetic algorithm,” in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 1995, pp. 77–82.
- Gharieb W., G. Nagib “Fuzzy Intervention in PID Controller Design” in ISIE 2001, PUNSAN KOREA, pp 1639-1643.
- Ibrahim A. S., “Nonlinear PID Controller Design Using Fuzzy Logic” in IEEE MELECOM 2002, pp 595-599.
- Klir G. J., “Fuzzy Logic” in IEEE Potentials Oct/Nov 1995, pp 10-15.
- Surmann H., J. Huser, and L. Peters, “A fuzzy system for indoor mobile robot navigation,” in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 1995, pp. 83–88.
- Wardana A. N. I., “PID-Fuzzy Controller for Grate Cooler in Cement Plant” in the *5th Asian Control Conf.* 2004, pp 1563-1567.

Interval Arithmetic Logic Unit for Signal Processing and Control Applications

William Edmonson, Ruchir Gupte, Senanu Ocloo, Jaya Gianchandani, Winsor Alexander
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695

Abstract. There are many applications within digital signal processing (DSP) and controls that require the user to know how various numerical errors affect the result, i.e. uncertainty. This uncertainty is represented by replacing non-interval values with intervals. Since most DSPs operate in real time environments, fast processors are needed. The goal is to develop a platform in which interval arithmetic operations are performed at the same computational speed as present day signal processors. We have proposed a design for an interval based arithmetic logic unit (I-ALU) whose computational time for implementing interval arithmetic operations is equivalent to many digital signal processors.

Many DSP and control applications require a small subset of arithmetic operations that must be computed efficiently. This design has two independent modules operating in parallel to calculate the lower bound and upper bound of the output interval. The functional unit of the ALU performs the basic fixed-point interval arithmetic operations of addition, subtraction, multiplication and the interval set operations of union and intersection. In addition, the ALU is optimized to perform dot products through the multiply-accumulate instruction. Division traditionally is not implemented on digital signal processors unless computed with a shift operation. In this design, division by shifting is implemented. The ALU is designed to have maximum throughput while minimizing area.

Keywords: arithmetic logic unit, interval arithmetic, signal processing

1. Introduction

Interval based algorithms continue to find applications as the solution for signal processing and controls problems. For instance, in signal processing, there is usually the need to determine the optimal solution to a problem, i.e., to minimize a cost function. The ability of interval global optimization approaches to guarantee convergence to global minimum point(s) (?) is one that makes such approaches attractive in digital signal processing (DSP) and control applications. DSP and control algorithms need to be designed in such a way that roundoff and truncation errors that occur naturally due to the discrete nature of computing do not cause the algorithm to become unstable. Interval analysis provides a means of managing such errors. It is therefore possible to obtain numerically accurate and reliable results.

Interval based algorithms are however slower than non-interval counterparts when run on current processor architectures. Such algorithms are usually implemented in software and so extra work

needs to be done in software to change rounding modes, perform memory management and perform error checks. These steps are time consuming and therefore make the algorithms run slowly.

If interval based algorithms are to become more practical, the throughput problem will have to be solved. This can be achieved by using arithmetic logic units (ALU) that are specially designed to manipulate interval numbers. Such an Interval ALU (I-ALU) can be used as the core of any digital signal processor. The throughput of such an ALU will have to be comparable to that of non-interval units. In contrast to general purpose microprocessors that are designed to handle general computing tasks, digital signal processors are designed and optimized to operate on algorithms that are characterized by repetitive multiply-and-add operations. They use a modified Harvard architecture with separate data and program memory (?). In general, they feature fast multiply-accumulate instructions, multiple-access memory, specialized program control for interrupt handling and I/O, and fast and efficient access to peripherals.

Interval floating-point ALUs have been proposed by ?. In this paper, we propose a fixed-point I-ALU. Fixed-point processors have the advantage of requiring less silicon, featuring faster clocks and being cheaper (?). The ALU is designed to perform the basic arithmetic operations of addition, subtraction and multiplication. Division by shifting is also implemented. Other operations that can be performed include multiply-accumulate (MAC), and the set operations of union and intersection.

The paper is organized as follows: section ?? discusses various aspects of the hardware design based on a modified Harvard architecture, section ?? shows the results, and finally, section ?? provides the conclusion.

2. Interval ALU

2.1. OVERALL ALU DESIGN

Consider the intervals $X = [x_L, x_U]$ and $Y = [y_L, y_U]$. The ALU is designed to perform operations $Z = X \text{ op } Y = \{x \text{ op } y \mid x \in X, y \in Y\}$ where $op \in \{+, -, \times, /\}$. It is also designed to perform the set operations of *union*, \cup and *intersection*, \cap . As is typical with digital signal processors, only division by powers of 2 is implemented. That is, given X / Y , Y is degenerate and a power of two. This allows for the division operation to be achieved by simply shifting the bits of the numerator, X . The ALU is also capable of calculating the dot product of two vectors by a *multiply-accumulate* operation.

In general, the result of each operation is a single interval. However, there is one situation where two intervals may result. This is the case when the union of two disjoint intervals is desired. Consider the operation $X \cup Y$ where X and Y are disjoint intervals. The result will then be two intervals, X and Y , and they will be placed on the output lines in two successive clock cycles.

The ALU is a fixed-point unit which represents numbers in two's complement format. One bit, the leftmost and most significant bit (MSB), is used as the *sign* bit. The remaining bits are used to represent the number. Figure ?? shows the structure of an N -bit signed number in two's complement format as used in our implementation.

Table ?? lists the inputs to the ALU. Operands are specified as 16-bit numbers. The ALU has input lines that allow selection of the *multiply-accumulate* mode and the number of bits for the

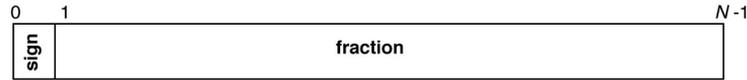


Figure 1. Fixed-point two's complement number format

output, $rctl$. The ALU has two 24-bit output lines for specifying the lower and upper bounds of the resulting interval. Table ?? shows the details of the output lines.

Table I. Description of ALU Inputs

Input	Description	Bit Width
x_L	Lower bound on left-hand operand	16 bits
x_U	Upper bound on left-hand operand	16 bits
y_L	Lower bound on right-hand operand	16 bits
y_U	Upper bound on right-hand operand	16 bits
cmd	Mathematical operation to be performed	3 bits
acc_select	Perform multiply-accumulate when asserted	1 bit
$rctl$	Width of output results (Choice of 16- or 24-bits)	2 bits

Table II. Details of ALU outputs

Output	Description	Bit Width
z_L	Lower bound on result	24 bits
z_U	Upper bound on result	24 bits

The hardware model is divided into three parts, namely, the *flag generator*, *lower bound* and *upper bound* modules. Figure ?? shows a schematic of the ALU. The flag generator module is responsible for generating flags that are used during multiplication. The nature of the input operands (x_L , x_U , y_L , y_U) and their values relative to zero are used to determine the value of the flag. There are nine possible cases that need to be identified. The 4-bit mul flag is used to distinguish between these possible cases. Table ?? shows the nine cases of multiplication and the associated mul flag values. The flag produced by the flag generator module is used by the lower and upper bound modules to determine the appropriate output values when multiplication is the operation desired. Note that there is one case where the result is the union of two intervals. This is the case where both X and Y contain 0 ($mul = 0000$). We shall refer to this case as *special case multiplication*.

The lower and upper bound modules are independent but equivalent in operation. These units are used for calculating the lower and upper bounds on the resulting interval(s). Both modules take the same set of inputs, namely the operands, mul flag and the command, cmd . Figure ?? shows the

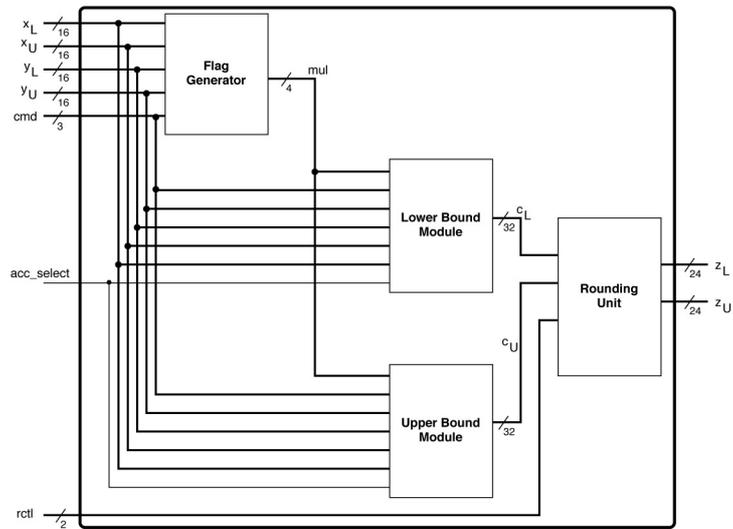


Figure 2. Schematic of Interval ALU

schematic for the lower and upper bound modules. Each module has an adder, a subtraction unit and a multiplier for performing computations, together with units of set operations.

Table III. Various cases of multiplication and associated *mul* flag values

MUL	Case	X*Y
0001	$x_L \geq 0; y_L \geq 0$	$[x_L y_L, x_U y_U]$
0010	$y_L \geq 0; y_L < 0 < y_U$	$[x_U y_L, x_U y_U]$
0011	$y_L \geq 0; y_U \leq 0$	$[x_U y_L, y_L y_U]$
0100	$y_L < 0 < x_U; y_L \geq 0$	$[y_L y_U, x_U y_U]$
0101	$y_L < 0 < x_U; y_U \leq 0$	$[x_U y_L, y_L y_L]$
0110	$x_U \leq 0; y_L \geq 0$	$[y_L y_U, x_U y_L]$
0111	$x_U \leq 0; y_L < 0 < y_U$	$[y_L y_U, y_L y_L]$
1000	$x_U \leq 0; y_U \leq 0$	$[x_U y_U, y_L y_U]$
0000	$y_L < 0 < x_U; y_L < 0 < y_U$	$[\min(x_U y_L, y_L y_U), \max(y_L y_L, x_U y_U)]$

Note that c_{out} is equal to c_L or c_U for the lower bound and upper bound modules respectively. A register is used to latch the output of each module.

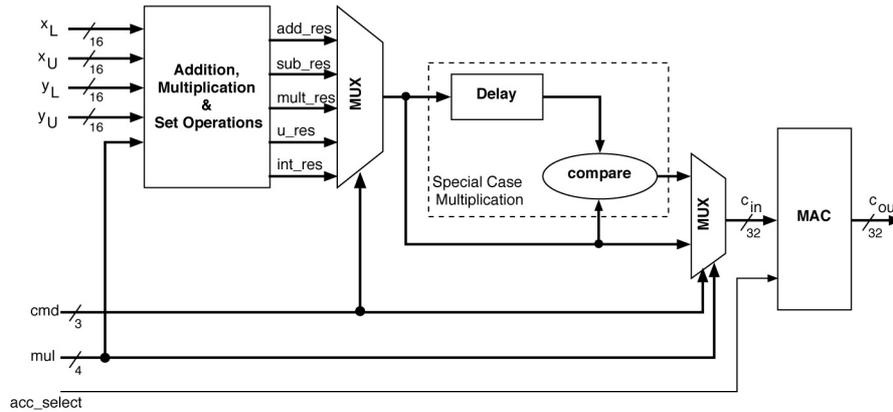


Figure 3. Block diagram of lower and upper bound modules

It is important to note that a delay unit is needed for special case multiplications. A comparator is also needed to perform the *min* and *max* operations. The presence of the delay unit implies that special case multiplications require an extra clock cycle to produce the final result. This is in contrast to the other arithmetic and set operations, namely, addition, subtraction, division, union and intersection, where the final result is obtained after one clock cycle.

2.2. MULTIPLY ACCUMULATE (MAC) UNIT

A dedicated multiply-accumulate functional unit is present in each of the lower and upper bound modules to execute the MAC instruction efficiently. An external input line *acc_select* is provided to determine when the accumulation needs to be performed. When this input line is held high, the accumulator is in *accumulate* mode; otherwise it serves the purpose of a simple latch. Figure ?? shows the block diagram of the accumulator.

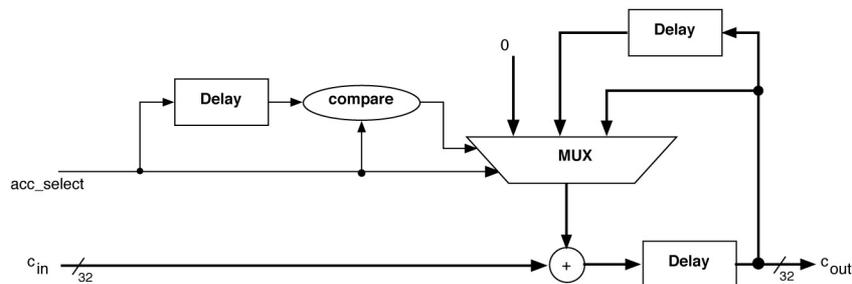


Figure 4. Multiply Accumulate (MAC) Unit

2.3. ROUNDING UNIT

One of the requirements of the I-ALU is that it should provide accurate, reliable results. Roundoff errors occur naturally in binary computations because floating-point and fixed-point number systems are unable to represent every possible number. In situations where the true result is not exactly representable, it is typical to round the number to the nearest machine-representable number (?). However, this introduces errors.

To avoid such errors, care must be taken to ensure that the resulting interval is guaranteed to contain the true result. This is achieved by using *outward rounding*. Outward rounding on an interval $X = [x_L, x_U]$ is achieved by rounding the lower bound, x_L , to the largest machine representable number smaller than x_L , and the upper bound x_U , to the smallest machine-representable number larger than x_U . This ensures that the true is result is contained in the resulting interval.

The I-ALU accepts 16-bit inputs and thus, the largest possible output (obtained through multiplication) requires 32-bits. However, if desired, the ALU can provide 16-bit or 24-bit results. Obviously, simple truncation of the original 32-bit result will lead to errors. Rounding is therefore required in such situations.

In twos complement number representation which is what is used for the ALU, the most significant bit of the result determines the sign of the answer. Different rounding modes need to be applied based on this sign of the output and also, based on whether the output value is to be rounded up or down. The bits of lower significance in the output have to be rounded up or down depending on the module under consideration.

2.3.1. Rounding Algorithm for Lower Bound Module

The output of the lower bound module will always be rounded toward negative infinity. The bits of lower significance simply need to be discarded based on the total number of bits to be retained. Figure ?? illustrates the rounding algorithm for the lower bound module.

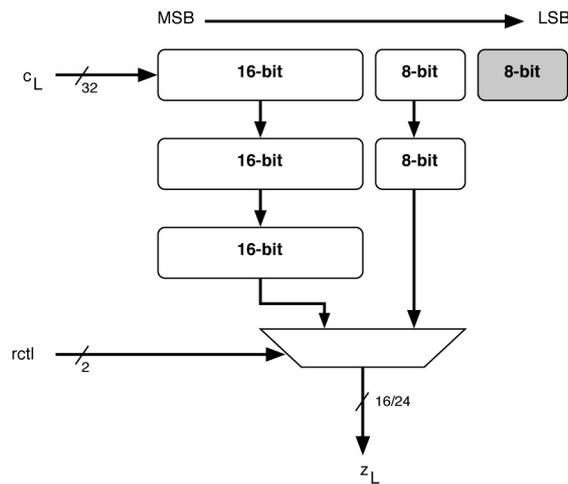


Figure 5. Rounding Unit for Lower Bound Module

2.3.2. Rounding Algorithm for Upper Bound Module

Rounding for the output is slightly more complicated. If any of the bits that are to be discarded is a 1, a 1 is added to the part that is going to be retained after rounding. Otherwise, simple truncation is performed. Figure ?? illustrates this rounding algorithm in brief.

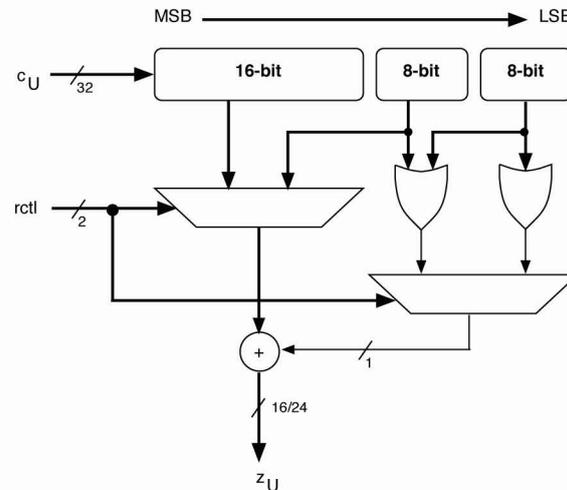


Figure 6. Rounding Unit for Upper Bound Module

3. Performance & Results

The I-ALU presented in this paper was designed in such a way that its performance would compare with non-interval ALUs. The two performance metrics of interest are *throughput* and *area*. Throughput was the more important performance metric so the design was first optimized for throughput and then for area. In other words, throughput was maximized while area was minimized. The design was implemented using Verilog HDL and synthesized using Synopsys. The $0.18\mu\text{m}$ technology library was used.

In order to compare favorably with non-interval ALUs, an interval ALU should be able to produce the results of computations in a single clock cycle. The design presented produces results in one clock cycle, except for cases where the result is a union of two disjoint intervals. The fastest clock obtained for the design was 45.8MHz.

Once the design was optimized for throughput, it was then optimized for area. The goal was to minimize the area. The minimum area obtained for our design was $218,569 \mu\text{m}^2$.

4. Conclusion

We have presented in this paper a design for an Interval based Arithmetic Logic Unit having computational efficiency comparable to many present day digital signal processors. This ALU operates on intervals represented by fixed point numbers in twos complement form. To make the ALU specific to DSP and control applications, dedicated hardware with a reduced instruction set of addition, subtraction, multiplication and for filtering operations, the multiply-accumulate operator is implemented. To bind the errors that accrue due to rounding, the outward rounding has been implemented. Throughput and area of the design has been optimized to obtain the best results.

References

- Hansen, E., Global Optimization Using Interval Analysis. Marcel Dekker, Inc., 1992.
- Kulisch, U., Advanced Arithmetic for the Digital Computer. New York: Springer-Verlag, 2002.
- Schulte, M. and J. E. Swartzlander, A Family of Variable-Precision Interval Arithmetic Processors, *IEEE Trans. Comput.*, vol. 49, no. 5, pp. 387–397, May 2000.
- Kuo, S. M. and W.-S. Gan, Digital Signal Processors: Architectures, Implementations and Applications. Prentice Hall, 2004.
- ANSI/IEEE, IEEE Standard for Binary Floating-Point Arithmetic. New York: ANSI/IEEE Std 754-1985, 1985.
- Moore, R. E. Interval Analysis. Prentice-Hall, Inc., 1966.

Interval-Based Robust Statistical Techniques for Non-Negative Convex Functions with Application to Timing Analysis of Computer Chips

Michael Orshansky¹, Wei-Shen Wang¹, Gang Xiang², and Vladik Kreinovich²

¹*Department of Electrical and Computer Engineering, University of Texas at Austin*
{orshansky,wswang}@mail.utexas.edu

²*NASA Pan-American Center for Earth and Environmental Studies (PACES)*
University of Texas at El Paso
{gxiang,vladik}@utep.edu

Abstract: In chip design, one of the main objectives is to decrease its clock cycle; however, the existing approaches to timing analysis under uncertainty are based on fundamentally restrictive assumptions. Statistical timing analysis techniques assume that the full probabilistic distribution of timing uncertainty is available; in reality, the complete probabilistic distribution information is often unavailable. Additionally, the existing alternative of treating uncertainty as interval-based, or affine, is limited since it cannot handle probabilistic information in principle. In this paper, a fundamentally new paradigm for timing uncertainty description is proposed as a way to consistently and rigorously handle partially available descriptions of timing uncertainty. The paradigm is based on a formal theory of interval probabilistic models that permit handling parameter uncertainty that is described in a distribution-free mode - just via the range, the mean, and the variance. This strategy permits effectively handling multiple real-life challenges, including imprecise and limited information about the distributions of process parameters, parameters coming from different populations, and the sources of uncertainty that are too difficult to handle via full probabilistic measures (e.g. on-chip supply voltage variation). Specifically, analytical techniques for bounding the distributions of probabilistic interval variables are proposed. Also, a provably correct strategy for fast Monte Carlo simulation based on probabilistic interval variables is introduced. A path-based timing algorithm implementing the novel modeling paradigm, as well as handling the traditional variability descriptions, has been developed. The results indicate the proposed technique can improve the upper bound of the 95th-percentile circuit delay, on average, by 4.8% across the ISCAS'85 benchmark circuits, compared to the worst-case timing analysis that uses only the interval information of the partially specified parameters.

Keywords: interval uncertainty, Monte-Carlo techniques

© 2006 by authors. Printed in USA.

1. Need for New Models of Uncertainty: Probabilistic Interval Analysis

The area of statistical static timing analysis (SSTA) has recently made substantial progress in terms of algorithmic and modeling advances. Efficient block-based and incremental computation techniques based on the first-order delay model are now well developed (Visweswariah et al, 2004; Chang and Sapatnekar, 2003). Extensions of the basic framework of SSTA to higher-order models have been recently investigated to capture non-linear effects and non-Gaussian process parameter distributions (Zhan et al, 2005; Chang et al, 2005; Zhang et al, 2005). Statistical delay computation for interconnect based on affine interval arithmetic has been studied (Ma and Rutenbar, 2004). These developments in the theory of SSTA came in response to the increased variability in the process parameters, the inadequacy of the corner models, and the need to use explicit probabilistic descriptions of key process parameters.

The fundamental assumption behind all of the above techniques is that the probabilistic descriptions are readily available. In all the algorithms for SSTA (Visweswariah et al, 2004; Chang and Sapatnekar, 2003; Zhan et al, 2005; Chang et al, 2005; Zhang et al, 2005), the complete knowledge about the distributions of process and environmental parameters is given, e.g. it is assumed that the process parameters are normally distributed, with the known mean and variance. Then, first-order models link delay variability with process parameters, allowing delay to be normally distributed as well (Visweswariah et al, 2004; Chang and Sapatnekar, 2003). If linear delay models are not sufficiently accurate, higher-order models can be used, at the cost of the resulting non-Gaussian distribution of delay. The non-Gaussianity of process parameters or timing can be handled by numerical processing leading to a substantial (3-10X) increase in the run-time of the algorithm (Chang et al, 2005)0.

In this paper we argue that in a practical setting of cutting-edge IC design the full probabilistic information about parameter uncertainty is not available. The process characterization data is often incomplete and of limited nature, especially at the ramp-up phase of the industrial manufacturing. With limited number of measurements and characterization lots, there may be a large uncertainty in the statistic metrics (the mean and the variance) of the process parameters. Some sources of on-chip uncertainty cannot be described probabilistically: supply voltage (V_{dd}), temperature, and systematic variation sources with the unit of repeatability larger than a single chip (e.g. aberration-caused L_{gate} variation).

Interval and affine methods, which tremendously improve on the conservatism of the traditional interval techniques, can be used in circuit timing analysis (Ma and Rutenbar, 2004). However, in many instances, *some but not full* probabilistic information is available. For example, variation of supply voltage in *time* depends on the input vectors applied to the chip. Because of the difficulty of performing temporal input-dependent analysis, the uncertainty about supply voltage is most typically represented by the range information (Ernst et al, 2004), however, the mean and, possibly, the variance of the distribution can be estimated more easily. For example, the supply voltage may vary between 90-100% of the nominal value, with the mean equal to 97% of the nominal value. The distribution is unknown because its characterization is

computationally expensive (Kouroussis et al, 2005). Statistical STA cannot meaningfully handle such a realistic scenario. The affine methods are fundamentally non-probabilistic and their extensions to handling statistics are heuristic in nature (Ma and Rutenbar, 2004).

Thus, in addition to the existing techniques, a new way of treating uncertain variables with partial probabilistic information is needed to enable practical design under uncertainty. This paper develops a solution of timing analysis under uncertainty based on the principles of probabilistic interval models. These models have been developed over the last decade in the field of robust statistics, reliable computing, and computer science (Kouznetsov, 1991). They are based on the generalization of classical random variables to variables described by families of distributions.

Conceptually, the most general description of an uncertain variable is an interval, e.g. $x \in [\underline{x}, \bar{x}]$. Such descriptions form the basis of interval arithmetic and its enhancement in terms of affine arithmetic (Moore, 1966; Stolfi and de Figueiredo, 2003). An interval description does not permit making statements about which values of the variable are more likely. Thus, if in addition to the range, the statistic metrics, such as mean and variance, are known the interval methods are incapable of utilizing this additional information in computing the arithmetic operations (+, -, *, /, max, min). *Probabilistic interval analysis* is a natural synergy of pure interval arithmetic and probabilistic analysis. It permits the use of partial statistic information (e.g. range, mean, and variance) to quantify the likelihood of the variable in the range. The estimates are guaranteed to be conservative regardless of the precise form of the distribution. For the fully specified random variable (e.g. Gaussian) the most general representation is its cumulative distribution function (*cdf*) (Feller, 1968). For a partially-specified random variable, the most general representation is a set of cumulative distribution functions, which can be represented as bounds on the *cdf*, forming a so-called probability box.

Following the above philosophy, this paper develops timing analysis techniques that produce reliable timing estimates even if the characterization data is incomplete. The essential contribution of this paper is in handling *incomplete* and *imprecise* uncertainty description. Compared to affine methods, the developed techniques can handle both the interval and probabilistic descriptions consistently and formally. The paper describes in detail how the probability boxes can be computed effectively. Importantly, the proposed techniques are compatible with the existing SSTA tools and can handle both full and partial probabilistic descriptions simultaneously.

This paper is organized as follows. Section 2 describes the paradigm of modeling non-probabilistic uncertainty based on probabilistic interval analysis, which enables us to use partial statistic metrics for timing analysis. The computation of path delay due to Gaussian variables and probabilistic interval variables is derived. Besides, a statistical technique of robustly estimating circuit delay distribution is proposed. The experimental results are presented in Section 3.

2. Timing Analysis with Partial Probabilistic Information

In this section, an application of the new probabilistic interval techniques to timing analysis is introduced. First, the construction of the path-delay probability box is described. Second, the bound of the circuit delay distribution is constructed. Finally, a method to combine the results of the traditional SSTA with the above derivations is described.

2.1. PATH DELAY COMPUTATION

The timing model used in this work is based on the additive delay model containing both the uncertainty due to classical random variables and the newly introduced probabilistic interval variables. The probabilistic interval variables (as opposed to random variables) are variables for which only partial statistic metrics, mean and variance, are available in addition to the known range, or interval. The delay model can be expressed as:

$$d_i = \mu_i + \sum_{j=1}^n a_{i,j} \Delta x_{i,j} + \sum_{k=1}^m b_{i,k} \Delta y_{i,k} \quad (1)$$

where μ_i is the mean value of the gate delay, $\Delta x_{i,j}$ is a zero-mean Gaussian random variable, and $\Delta y_{i,k}$ is a zero-mean probabilistic interval variable. The coefficients $a_{i,j}$ and $b_{i,k}$ are the sensitivities of gate delays, which are the first-order derivatives of gate delays with respect to the variables. Note that this delay model can be easily transformed into an affine arithmetic representation if variables are scaled such that the variables are limited within $[-1, 1]$.

A concise representation of the gate delay model can be obtained by resorting to the matrix form:

$$d_i = \mu_i + A_i^T X_i + B_i^T Y_i \quad (2)$$

where the matrices $A_i = [a_{i,1} \cdots a_{i,n}]^T$, $B_i = [b_{i,1} \cdots b_{i,m}]^T$, $X_i = [\Delta x_{i,1} \cdots \Delta x_{i,n}]^T$, and $Y_i = [\Delta y_{i,1} \cdots \Delta y_{i,m}]^T$.

The variation of parameters can be further decomposed into the linear sum of perfectly correlated die-to-die components (X_{dd}, Y_{dd}), and independent within-die components ($X_{i,wd}, Y_{i,wd}$):

$$d_i = \mu_i + A_i^T X_{i,wd} + A_i^T X_{dd} + B_i^T Y_{i,wd} + B_i^T Y_{dd} \quad (3)$$

The path delay of a path P_j can be represented by:

$$\begin{aligned} D^j &= \sum_{i \in P_j} (\mu_i + A_i^T X_{i,wd} + A_i^T X_{dd} + B_i^T Y_{i,wd} + B_i^T Y_{dd}) \\ &= \sum_{i \in P_j} \mu_i + \sum_{i \in P_j} g_i + \sum_{i \in P_j} u_i \end{aligned} \quad (4)$$

where $g_i = A_i^T X_{i,wd} + A_i^T X_{dd}$, and $u_i = B_i^T Y_{i,wd} + B_i^T Y_{dd}$

It is convenient to separate the contributions of random delay uncertainty (D_R) and probabilistic interval uncertainty (D_{PI}): $D_R^j = \sum_{i \in P_j} g_i$ and $D_{PI}^j = \sum_{i \in P_j} \mu_i + \sum_{i \in P_j} u_i$. Computing the path delay distribution when the gate delays are normal random variables is straightforward. Therefore, we focus on the *delay variation resulting from probabilistic interval variables* i.e. D_{PI}^j . The range of the gate delay variation, u_i , is:

$$u_i \in \left[\sum_{k=1}^m b_{i,k} \underline{\Delta y_{i,k}}, \sum_{k=1}^m b_{i,k} \overline{\Delta y_{i,k}} \right] \quad (5)$$

where $\underline{\Delta y_{i,k}}$ and $\overline{\Delta y_{i,k}}$ are the lower and upper bound of $\Delta y_{i,k}$. Then we can compute the range of D_{PI}^j .

Because the mean values of probabilistic interval variables are zero, the mean of the path delay is:

$$E[D_{PI}^j] = \sum_{i \in P_j} \mu_i \quad (6)$$

The variance of the path delay can be computed by:

$$Var\{D_{PI}^j\} = \sum_{i \in P_j} B_i^T \Sigma_{i,wd} B_i + \left(\sum_{i \in P_j} B_i^T \right) \Sigma_{dd} \left(\sum_{i \in P_j} B_i \right) \quad (7)$$

where $\Sigma_{i,wd}$ and Σ_{dd} are the covariance matrices of $Y_{i,wd}$ and Y_{dd} , respectively. Since different kinds of parameters are uncorrelated, the covariance matrices are actually diagonal matrices, with the diagonal elements equal to the variance of variables.

While the ultimate objective of the paper is to derive the circuit delay distribution, being able to describe individual path delay distributions is also essential. Now that the range, the mean and the variance of D_{PI}^j are known, the challenge is to compute the probability box that contains the family of distributions satisfying the partial statistical information that is available. Actually, the computation of the probability bound can be formulated as an optimization problem:

Let $F_i : \mathfrak{R} \rightarrow [0, 1]$ ($1 \leq i \leq n$) be a possible cumulative distribution function of a random variable X , and F_i satisfies the partial statistical information: $E[X] = \mu$, $Var[X] = \sigma^2$, and $X \in [\underline{X}, \overline{X}]$. The lower bound for the cumulative probability of X at a specific value x , can be computed by solving the optimization problem considering all possible F_i :

$$\max p \text{ s.t. } F_i(x) \geq p, \quad 1 \leq i \leq n.$$

Similarly, the upper bound can be computed by:

$$\min p \text{ s.t. } F_i(x) \leq p, 1 \leq i \leq n.$$

However, because we seek a fast analytical solution, we prefer to use an inequality, which is a combination of the Chebyshev inequality and Cantelli inequality (Godwin, 1964). This inequality applies when, in addition to the first two moments of the variable, its support (range) is also known, resulting in a tighter bound on the *cdf*. The upper bound of the cumulative probability of a random variable X is given by (Ferson, Kreinovich, Ginzburg, Myers, and Sentz, 2002):

$$\begin{aligned} P(X \leq x) &= 0 & x < \underline{X} \\ P(X \leq x) &\leq 1/(1+(\mu-x)^2/\sigma^2) & \underline{X} \leq x < \mu + \sigma^2/(\mu - \bar{X}) \\ P(X \leq x) &\leq 1-(m^2-my+s^2)/(1-y) & \mu + \sigma^2/(\mu - \bar{X}) \leq x \\ & & \text{and } x < \mu + \sigma^2/(\mu - \underline{X}) \\ P(X \leq x) &= 1 & \mu + \sigma^2/(\mu - \underline{X}) \leq x \end{aligned} \quad (8)$$

where μ denotes the mean, σ^2 denotes the variance, \underline{X} is the lower bound, \bar{X} is the upper bound, $y = (x - \underline{X})/(\bar{X} - \underline{X})$, $m = (\mu - \underline{X})/(\bar{X} - \underline{X})$, and $s^2 = \sigma^2/(\bar{X} - \underline{X})^2$.

Similarly, the lower bound of the cumulative probability is:

$$\begin{aligned} P(X \leq x) &= 0 & x < \mu + \sigma^2/(\mu - \bar{X}) \\ P(X \leq x) &\geq 1-(m(1+y)-s^2-m^2)/y & \mu + \sigma^2/(\mu - \bar{X}) \leq x \\ & & \text{and } x < \mu + \sigma^2/(\mu - \underline{X}) \\ P(X \leq x) &\geq 1/(1+\sigma^2/(x-\mu)^2) & \mu + \sigma^2/(\mu - \underline{X}) \leq x < \bar{X} \\ P(X \leq x) &= 1 & \bar{X} \leq x \end{aligned} \quad (9)$$

Thus, expressions (8) and (9) can be used to compute the bound for the path delay cumulative probability. An example of applying this set of inequalities is shown in Figure 2(a).

The same analytical structure can be used when the mean and variance are known only with certain accuracy (Ferson, 2002). First, the maximum of the variance should be used in the generalized Chebyshev inequality because it primarily determines the span of the *cdf*. Second, the upper bound of the mean should be used when computing the lower (right-side) bound of the probability using (9), because it leads to the *worst* lower bound of the probability. Similarly, the lower bound of the mean should be used in (8).

Having computed the distribution of path delay variation due to probabilistic interval variables, now we combine it with the delay variation resulting from Gaussian variables. Since parameters of different categories are independent, it means that the delay variations D_R^j and D_{PI}^j are independent, and the bound for the *cdf* of the sum can be computed by convolution:

$$CDF(D^j) = CDF(D_{PI}^j) \otimes f(D_R^j) \quad (10)$$

where $f(D_R^j)$ is the probability density function of D_R^j . We use the lower and upper bounds of $CDF(D_{PI}^j)$ in convolution respectively, and then obtain the bounds of $CDF(D^j)$. Finally, we have the bound for the path delay distribution, which enables computing the bound of delay at any quantile.

2.2. CIRCUIT TIMING COMPUTATION

In this section, we develop techniques for efficient construction of probability boxes on the distribution of circuit delay, i.e. the maximum of all path delays. New techniques are proposed to perform this task efficiently and robustly. From (4), the bound of the circuit delay can be computed by:

$$\begin{aligned} D_{\max} &= \max(D^1, \dots, D^N) \\ &= \max\left(\sum_{i \in P_1} (\mu_i + g_i + u_i), \dots, \sum_{i \in P_N} (\mu_i + g_i + u_i)\right) \\ &\leq \max(D_R^1, \dots, D_R^N) + \max(D_{PI}^1, \dots, D_{PI}^N) \end{aligned} \quad (11)$$

Let $D_{R_{\max}} = \max(D_R^1, \dots, D_R^N)$ be the term due to random probabilistic variability, and the second term $D_{PI_{\max}} = \max(D_{PI}^1, \dots, D_{PI}^N)$ be the term due to interval-probabilistic variability. In deriving the probability box for D_{\max} , we adopt a strategy in which the sources of uncertainty described probabilistically are separated from interval probabilistic uncertainty. The distribution of $D_{R_{\max}}$ can be computed by the statistical timing analysis algorithm based on the first-order delay models (Visweswariah et al, 2004; Chang and Sapatnekar, 2003; Agarwal et al, 2003; Orshansky and Bandyopadhyay, 2004). Therefore, in the remainder, we concentrate on the computation of $D_{PI_{\max}}$. The two terms are then combined to generate the bounds on the full distribution of circuit delay.

In constructing the probability box for the circuit delay distribution, ideally, we would like to use analytical means as was done in Section 2.1. Expressions (8) and (9) can be used to find the bounds on the distribution of $D_{PI_{\max}}$, once the mean, the variance, and the range are known.

However, in general functions of probabilistic interval variables, $f(u_1, \dots, u_N)$, finding the bounds on the variance is NP-hard (Ferson, Ginzburg, Kreinovich, Longpré, and Aviles, 2002). We show below that for *convex* functions the exact bound on the variance can be computed. Let us first establish the convexity of the term $D_{PI \max}$. The path delay $D_{PI}^j = \sum_{i \in P_j} u_i$ is a linear and thus convex function of u_i . The circuit delay is given by $D_{PI \max} = \max(D_{PI}^1, \dots, D_{PI}^N)$ which is also a convex function of probabilistic interval variables (Boyd and Vandenberghe, 2004). Convexity is essential to our efficient analysis strategy, since as the theorem below shows determining the probability bound and moments of distributions of convex functions is much easier.

Our strategy is essentially based on the development of the robust (guaranteed) approach to Monte Carlo sampling from an unknown distribution (Orshansky et al, 2006). The Monte-Carlo simulation is a widely-used technique to solve complex numerical problems (Fishman, 1995). It can be used as a powerful tool for estimating the timing performance of integrated circuits when the distributions are known (Jyu et al, 1993; Lemke et al, 2002). Without the full distributional knowledge of the parameters, a possible way to perform the simulation is to heuristically generate a variety of distributions that correspond to the given partial information. However, this method is not mathematically robust because it is impossible to enumerate all possible distributions. Besides, the high run time accounting for numerous distributions prevents this method from practical use. We show that for convex functions the *robust Monte Carlo* simulation can be rigorously and efficiently performed. Compared to the traditional approach to Monte-Carlo simulation, the selection of distribution is *justified* in our simulation strategy; only distributions that cause the extreme value of the target function need to be considered. Therefore, this selective strategy is guaranteed to produce a bounding distribution, and achieves high efficiency in terms of the run time. Theorem 1 effectively defines the algorithm for such robust Monte Carlo (Orshansky et al, 2006).

Theorem 1. Let $\{v_1, \dots, v_M\}$ be a set of *independent* random variables, where $v_i \in [v_i, \bar{v}_i]$, and $E[v_i] = E_i$ for $i=1$ to M . Let $f(v_1, \dots, v_M)$ be a non-negative convex function of the random variable v_i , for $i=1$ to M . The probabilistic bound of $f(v_1, \dots, v_M)$, at a confidence level α , is defined as:

$$D^\alpha = \min\{D \in \mathbb{R} : P(f(v_1, \dots, v_M) \leq D) \geq \alpha\}$$

Assume D^α decreases if any interval $[v_i, \bar{v}_i]$ is narrowed down.

Then, among all possible *cdfs* of $\{v_i : i=1..M\}$ that correspond to the partial statistical information of the range and the mean, the bound D^α achieves the maximum value when each random variable v_i follows the 2-point distribution,

$$\begin{aligned} P(v_i = \underline{v}_i) &= \underline{p}_i \\ P(v_i = \overline{v}_i) &= \overline{p}_i \end{aligned} \quad (12)$$

where $\underline{p}_i = \frac{\overline{v}_i - E_i}{\overline{v}_i - \underline{v}_i}$, and $\overline{p}_i = \frac{E_i - v_i}{\overline{v}_i - v_i}$.

Effectively, Theorem 1 reduces the number of possible distributions that must be considered in order to find the bounding distribution, which will result in a sought probability box for the function of probabilistic interval variables. However, this robust Monte Carlo simulation still suffers from the common problems of Monte Carlo - the slow decrease of the estimation error, especially, at high percentiles. To address this concern, we have developed a fast *hybrid* approach, *the fast robust Monte Carlo simulation*, in which robust Monte Carlo is used to get a quick estimate of the moments (a much faster computation) and then analytical techniques are used for establishing bounds. The justification of the technique is based on the corollary to Theorem 1.

Corollary. The k^{th} moment of the function, $E[y^k]$, where $y = f(v_1, \dots, v_M)$, achieves the maximum value when each random variable v_i follows the 2-point distribution in (12). Furthermore, $E[y^k]$ achieves the minimum when $P(v_i = E_i) = 1$.

Therefore, using the above sampling procedure also guarantees that the bounds of $E[f(v_1, \dots, v_M)]$ are accurately estimated.

In the fast robust Monte Carlo simulation, a limited number of random samples are drawn using the algorithm following Theorem 1. The corollary guarantees that we will get an accurate estimate of the range of the mean circuit delay. As for the variance of the circuit delay, it can also be bounded by the sample variance because the 2-point distribution in (12) results in the maximum variance of gate delays thus maximizes the variance of path delays and the circuit delay. Therefore, expressions (8) and (9) can be then used to compute the bound of the distribution analytically.

Figure 1 illustrates the algorithm of the fast robust Monte Carlo simulation. This proposed strategy estimates the upper bound of sample mean and sample variance with only a limited number of runs. In practice, a few hundred runs are sufficient to generate an estimate with reasonable accuracy. This can be verified by considering the standard error of the sample mean and the confidence level of the true mean i.e. the mean of the population. From (Rice, 1988), the 99% confidence interval of the true mean (μ) for a variable X is $\overline{X} \pm 2.575 \sigma_X / \sqrt{N}$, where \overline{X} is the sample mean, σ_X is the true standard deviation, and N is the number of samples. For example, consider a circuit with extremely large span in the delay domain: the 3σ value of circuit delay is 45% of the mean. Then we estimate the confidence level:

$$P(|\bar{X} - \mu| \leq 2.575 \cdot 0.15\mu / \sqrt{N}) = 0.99.$$

The error of the sample mean for $N = 500$ is less than 1.7% with probability equal to 0.99, which has a very limited impact on the result of using expressions (8) and (9). Thus, the accuracy of Monte Carlo for such a sample size is acceptable for our analysis.

Once the lower bound on the distribution of $D_{PI_{\max}}$ is generated, the overall circuit delay distribution D_{\max} can be obtained by combining $D_{PI_{\max}}$ and $D_{R_{\max}}$. Since these two components of delay variation are independent, the distribution of the sum can be computed by convolution, similar to (10). The lower bounds of the *cdf* (i.e. the upper bound of the delay) are used in the convolution because it is a more important metric for circuit timing.

```

for  $i = 1..N$ 
  Generate a sample for each die-to-die parameter.
  for each gate
    Generate a sample for each within-die parameter.
    Compute gate delay.
  end
  Use static timing analysis to compute the circuit delay,  $D_i$ .
end

```

Compute the mean and the variance of samples:

$$\bar{D} = \sum_{i=1}^N D_i / N$$

$$s_D^2 = \sum_{i=1}^N (D_i - \bar{D})^2 / (N - 1)$$

With \bar{D} , s_D^2 , and the range of the circuit delay, use (9) to compute the lower bound of the *cdf*.

Figure 1. Algorithm of the fast robust Monte Carlo simulation.

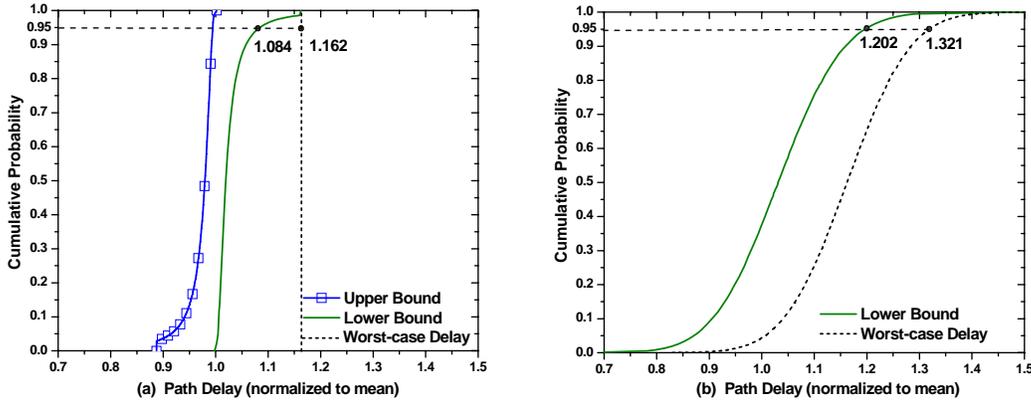


Figure 2. The path delay analysis algorithm improves the worst-case path delay by 9.0% for the critical path of circuit c6288 at the 95th percentile. a) Delay due to probabilistic interval variables; b) Total path delay.

3. Experiments

The algorithms for timing analysis using partial description of uncertainty described in Section 2 have been implemented in C++, and have been tested on a set of combinational ISCAS'85 benchmark circuits. Variability of process parameters (L , V_{th} , and T_{ox}) and the environmental fluctuation (V_{dd}) are taken into account. The 3σ values for process parameters are set at 20% of the mean, including 50% die-to-die variations. The standard deviation of V_{dd} is 4% of the maximum, and the range of V_{dd} is 84-100% of the maximum value. In the experiments, V_{th} , T_{ox} and V_{dd} are modeled as probabilistic interval variables. Sensitivities of parameters are from SPICE simulations for a cell library of BPTM 0.13um technology (Cao et al, 2000).

The proposed timing analysis algorithms separately handle the contributions of the random probabilistic uncertainty and the interval probabilistic uncertainty. Thus, the comparison of our algorithms and the worst-case timing analysis i.e. only using the range (interval) of the interval uncertainty, should be done in two phases. We first compare the bounds of D_{PI}^j computed by the proposed algorithm and the worst-case timing analysis, then compare the bound of the total delay, which is the sum of D_{PI}^j and D_R^j . Note that the sum of the bound from the worst-case timing analysis for interval uncertainty and D_R^j can be computed by simply shifting the *cdf* of D_R^j by the worst-case delay value. A similar comparison is also made for the bounds on circuit delay distribution.

Figure 2(a) illustrates the importance of probabilistic interval analysis in path delay analysis. The upper bound of the 95th- percentile path delay (D_{PI}^j) from the proposed algorithm for the critical path of circuit c6288 is only 8.4% over the mean path delay, while the worst-case timing estimate is 16.2% over the mean. Therefore, the proposed path timing analysis algorithm reduces the worst-case timing estimate by 6.7%. Similarly, the 95th-percentile total path delay

$(D_R^j + D_{PT}^j)$ is 20.2% over the mean for the proposed algorithm, which is a better bound than the worst-case delay (32.1% over the mean) in Figure 2(b). Thus, the proposed strategy improves the worst-case estimate by 9.0% for the overall path delay at the 95th percentile.

For circuit delay distribution, the proposed statistical technique has been run on a Sun workstation with 1280 MHz CPU and 8GB memory. We ran the fast robust Monte Carlo simulation (FRMC) to estimate the sample mean and the variance using 1,000 samples, and then analytically computed the lower bound of the cumulative probability. The run time of the fast robust Monte Carlo ranges from 12 to 114 seconds. Figure 3 shows the circuit delay variation due to probabilistic interval variables of circuit c7552, from the proposed statistical technique and the worst-case timing analysis. It shows that FRMC is able to provide a superior bound to the worst-case delay at lower than the 87th percentile.

For the total circuit delay (D_{\max}), FRMC improves the estimates from the worst-case timing analysis by 4.8% across the benchmark circuits, for the 95th percentile delay. Table I shows the upper bound of the total circuit delay at high percentiles (90th and 95th percentiles) for FRMC, and

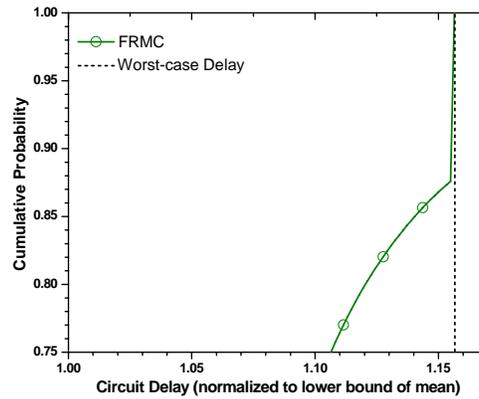


Figure 3. Upper bounds for circuit delay due to probabilistic interval variables for circuit c7552. The fast Monte Carlo simulation provides a bound superior to the worst-case timing estimate at lower than the 87th percentile.

the worst-case timing analysis. Figure 4 shows an example of the total circuit delay for the circuit c7552, in which FRMC reduces the worst-case delay estimate by 4.5% at the 95th percentile. Indeed, the joint use of SSTA and our statistical technique for probabilistic interval variables is a promising synergy, and it can be easily extended to incorporate more circuit parameters, to fully assess the impact on timing performance.

Another important feature of the proposed techniques is the capability of handling skewed distributions. Some environmental parameters are not symmetrically distributed (e.g. V_{dd}); however, the normal assumption implies the distribution is symmetrical to the mean, which may cause inaccurate estimation of the circuit delay. Figure 5(a) compares path delay distributions of

two cases with the same interval and variance of V_{dd} uncertainty: the right-skewed V_{dd} uncertainty and the symmetrical case. Because the voltage drop increases delay, the right-skewed V_{dd} uncertainty decreases the upper bound of delays, compared to the center-meant V_{dd} distribution. From Figure 5(b), the similar trend can be also observed in the distribution of the total circuit delay. Thus, our timing analysis algorithm can be used to handle asymmetrical distributions (e.g. non-Gaussian), and provide a more accurate timing estimate.

4. Conclusions

In this paper, we propose a set of statistical techniques for estimating the path and circuit delay distributions. Given partial statistic metrics of the uncertainty, the proposed algorithm is able to analytically compute the bounds of the path delay. A fast robust Monte Carlo simulation technique is proposed to assess the impact of the uncertainty, and estimate the upper bound of the circuit delay. With justified selection of the distribution used in the simulation, the proposed technique can efficiently provide a guaranteed bound of the circuit delay distribution.

Table I. Upper bounds for circuit delay at high percentiles and the run time of the proposed technique.

Circuit	Number of Gates	Fast Robust Monte Carlo Simulation				Run Time (s)	Worst-case Delay	
		90 th Percentile		95 th Percentile			90 th Percentile	95 th Percentile
		Delay (ps)	Reduction (%)	Delay (ps)	Reduction (%)		Delay (ps)	Delay (ps)
c880	456	2383	5.62	2467	4.97	12	2525	2596
c1355	605	2264	4.59	2335	4.26	18	2373	2439
c1908	975	2820	5.56	2919	4.89	26	2986	3069
c2670	1544	3124	5.65	3232	5.08	38	3311	3405
c3540	1787	4097	5.49	4237	4.94	52	4335	4457
c6288	2448	17547	5.28	18081	4.82	87	18526	18996
c5315	2600	3579	5.49	3703	4.88	79	3787	3893
c7552	3874	3136	4.88	3236	4.46	114	3297	3387

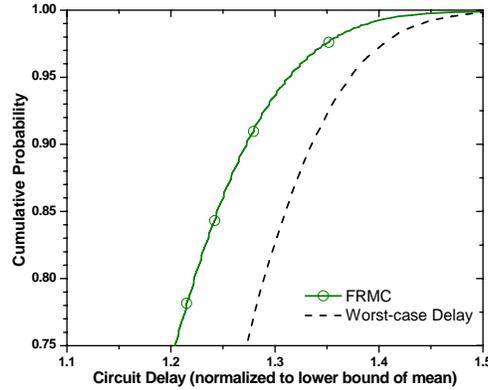


Figure 4. Upper bounds for the overall circuit delay of c7552. FRMC improves the worst-case delay estimate by 4% at the 95-percentile.

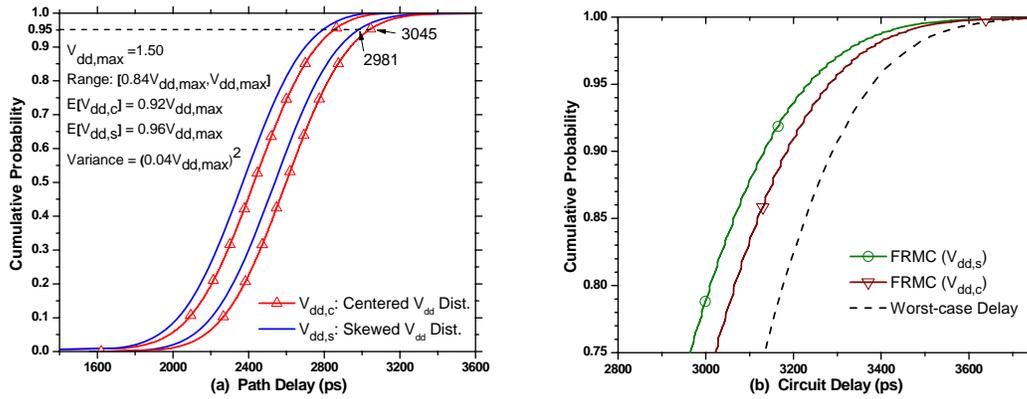


Figure 5. The right-skewed V_{dd} distribution improves bounds of (a) path delay; and (b) circuit delay of center-meant V_{dd} distribution.

Acknowledgements

This work was supported in part by NASA under cooperative agreement NCC5-209, NSF grants EAR-0225670 and DMS-0532645, Army Research Lab grant DATM-05-02-C-0046, Star Award from the University of Texas System, Texas Department of Transportation grant No. 0-5453, GSRC, NSF, SRC, Sun, and Intel.

References

- Agarwal, A., D. Blaauw, V. Zolotov, S. Sundareswaran, M. Zhao, K. Gala, and R. Panda. Path-based statistical timing analysis considering inter- and intra-die correlations. *TAU*, 2002.
- Barmish, R. and H. Kettani. Monte Carlo analysis of resistive networks without apriori probability distributions. *Proc. of ISCAS*, 2000.
- Boyd, S. and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- Cao, Y., T. Sato, D. Sylvester, M. Orshansky, and C. Hu. New paradigm of predictive MOSFET and interconnect modeling for early circuit design. *Proc. of Custom Integrated Circuits Conference*, pp. 201-204, 2000.
- Chang, H. and S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single PERT-like traversal. *Proc. of International Conference on Computer Aided Design*, 2003.
- Chang, H., V. Zolotov, S. Narayan, and C. Visweswariah. Parameterized block-based statistical timing analysis with non-Gaussian parameters and nonlinear delay functions. *Proc. of Design Automation Conference*, 2005.
- Ernst, D., S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, N. S. Kim, and K. Flautner. Razor: circuit-level correction of timing errors for low-power operation. *IEEE Micro*, 24(6):10-20, November 2004.
- Feller, W. *An Introduction to Probability Theory and Its Applications*. Wiley and Sons, 3rd Edition, 1968.
- Ferson, S., V. Kreinovich, L. Ginzburg, D. S. Myers, and K. Sentz. Constructing probability boxes and Dempster-Shafer structures. Sandia Report, 2002.
- Ferson, S. *RAMAS Risk Calc 4.0 Software: Risk Assessment with Uncertain Numbers*. CRC Press, 2002.
- Ferson, S., L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles. Computing variance for interval data is NP-hard. *ACM SIGACT News*, Vol. 23(2), pp. 108-118, June 2002.
- Fishman, G. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, 1995.
- Godwin, H. *Inequalities on Distribution Functions*. Hafner, 1964.
- Hitchcock, R. Timing verification and the timing analysis program. *Proc. of Design Automation Conference*, 1982.
- Jyu, H.-F., S. Malik, S. Devadas, and K. Keutzer. Statistical timing analysis of combinational logic circuits. *IEEE Trans. on VLSI Systems*, vol.1, (no.2), pp. 126-37, 1993.
- Kouroussis, D., I. A. Ferzli, and F. N. Najm. Incremental partitioning-based vectorless power grid verification. *Proc. of International Conference on Computer Aided Design*, 2005.
- Kouznetsov, V. P. *Interval Statistical Models*. Radio i Svyaz, Moscow, 1991 (In Russian).

- Lemke, A., L. Hedrich, and E. Barke. Analog circuit sizing based on formal methods using affine arithmetic. *Proc. of International Conference on Computer Aided Design*, 2002.
- Ma, J. D. and R. A. Rutenbar. Interval-valued reduced order statistical interconnect modeling. *Proc. of International Conference on Computer Aided Design*, 2004.
- Moore, R. E. *Interval Analysis*. Prentice-Hall, 1966.
- Orshansky, M. and A. Bandyopadhyay. Fast statistical timing analysis handling arbitrary delay correlations. *Proc. of Design Automation Conference*, 2004.
- Orshansky, M., W. -S. Wang, M. Ceberio, and G. Xiang. Interval-based robust statistical techniques for non-negative convex functions, with application to timing analysis of computer chips. *to appear in ACM Symposium on Applied Computing*, 2006.
- Rice, J. *Mathematical Statistics and Data Analysis*, Wadsworth & Brooks, 1988.
- Stolfi, J. and L.H. de Figueiredo. An introduction to affine arithmetic. *TEMA Tend. Mat. Apl. Computing*, 4, No. 3 (2003), 297-312.
- Visweswariah, C., K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan. First-order incremental block-based statistical timing analysis. *Proc. of Design Automation Conference*, 2004.
- Zhan, Y., A. Strojwas, X. Li, L. Pileggi, D. Newmark, and M. Sharma. Correlation-aware statistical timing analysis with non-Gaussian delay distributions. *Proc. of Design Automation Conference*, 2005.
- Zhang, L., W. Chen, Y. Hu, J. Gubner, and C. C. -P. Chen. Correlation-Preserved Non-Gaussian Statistical Timing Analysis with Quadratic Timing Model. *Proc. of Design Automation Conference*, 2005.

On reliability of finite element method in fluid-structure interaction problems *

Petr Sváček

*Faculty of Mechanical Engineering, Czech Technical University in Prague, Department of
Technical Mathematics, Karlovo nám. 13, 121 35 Praha 2*

Abstract. In this paper we are concerned with numerical methods for fluid-structure interaction (FSI) problems and with their verification and validation. The fluid-structure interaction modelling is very complicated problem, where the most complicated and crucial part is modelling of the fluid flow. Therefore the main interest of this paper is the numerical approximation of two dimensional incompressible viscous fluid over a flexibly supported profile. In technical problems the relevant Reynolds numbers are usually very high ($10^4 - 10^6$) and the fluid flow is turbulent. The correct numerical approximation requires very fine mesh refining as well as very small time steps involved in the computation. On the other hand in many technical applications the Reynolds Averaged Navier-Stokes equations are being used together with a suitable turbulence model. Here, both (laminar) Navier-Stokes equations as well as Reynolds Averaged Navier-Stokes equations are considered, numerically approximated by the Finite Element Method (FEM), stabilized by Galerkin-Least-Squares technique, and the obtained solution compared to the experimental data.

Keywords: aeroelasticity, Reynolds Averaged Navier-Stokes equations, Navier-Stokes equations

Nomenclature

$L(t), D(t), M(t)$	=	aerodynamic lift and drag force and torsional moment
m	=	mass of the airfoil
S_α, I_α	=	static and inertia moments around the elastic axis EO
$k_{hh}, k_{\alpha\alpha}$	=	bending and torsional stiffness
l, c	=	airfoil depth and chord
α, h	=	rotational and vertical displacements around the elastic axis EO
\mathcal{G}_t	=	computational domain occupied by fluid at time t
$\partial\mathcal{G}_t$	=	boundary of the domain \mathcal{G}_t
P	=	time averaged kinematic pressure,
ρ, ν	=	constant fluid density and (laminar) kinematic viscosity of the fluid
ν_T	=	turbulent kinematic viscosity
τ_{ij}, σ_{ij}	=	fluid stress tensor and Reynolds stress tensor
Ω_{ij}	=	tensor rotation of the fluid velocity

* This research was supported under grant No. 201/05/P142 of the Czech Grant Agency and under Research Plan MSM 6840770003 of the Ministry of Education of the Czech Republic.

1. Introduction

The fluid-structure interaction problems can be met in many technical applications (for details see, e.g., (Dowell, 1995; Naudasher and Rockwell, 1994)). The treatment of fully coupled interaction problem of a structure and fluid flow is very difficult. Therefore, it is usually modelled with several simplifications. The main objective of commercial codes (as, e.g., NASTRAN) is to determine the critical fluid flow velocity. The post-flutter behaviour can not be captured. The special problems of aero-elasticity mainly in linear domain are solved.

The paper focus on numerical simulations of two dimensional viscous incompressible air flow around an airfoil. The main objective is the correct numerical resolution of the flow and the fluid forces acting on the airfoil. The relevant flow velocities for the selected class of problems are in the range $0-120 \text{ m s}^{-1}$. The flow is described by the incompressible Navier-Stokes equations. The other possibility is to use the model of compressible flow. Nevertheless, the numerical approximation of low Mach number flows at incompressible limit is quite complicated and a modification of governing equations has to be used.

The numerical approximation of incompressible flow can be carried out with the use of various methods. In CFD, the finite volume method is rather popular. In our paper the finite element method is used for the spatial discretization of the problem. In this case several sources of instabilities have to be treated. First, in order to guarantee the stability of the scheme the finite elements for velocity and pressure need to be selected in a proper way to satisfy the Babuška-Brezzi condition. Moreover, very high Reynolds numbers result in the appearance of spurious oscillations in the approximate solution. In last decades a number of stabilization procedures have been developed. In this paper the stabilization based on GLS (Galerkin Least-Squares) method together with grad-div stabilization is employed. The combination of this method with the mesh refinement (e.g., performed by the anisotropic mesh generator, see (Dolejší, 2001)) results in a very robust and efficient method. The choice of stabilization parameters is based on the numerical analysis of the problem as well as numerical experience, see (Lube, 1994), (Sváček and Feistauer, 2004). The presented method is applied to the solution of incompressible (laminar) Navier-Stokes equations and also to the solution of Reynolds Averaged Navier-Stokes (RANS) equations. In this paper the application of the finite element method to RANS system of equations is discussed. For the description of application onto (laminar) Navier-Stokes equations, see (Sváček, Feistauer, and Horáček, 2004). The Reynolds stresses involved in the RANS equations are modelled with the aid of the Spallart-Almaras turbulence model (for an overview of turbulence models used in computational fluid dynamics, see, e.g. (Wilcox, 1993)).

The structure motion is simulated by the solution of a system of nonlinear ordinary differential equations for the vertical and angular displacements. The airfoil motion results in deformations of the computational domain, which are treated with the aid of Arbitrary Lagrangian-Eulerian(ALE) method, see (Nomura and Hughes, 1992), (LeTallec and Mouro, 1998).

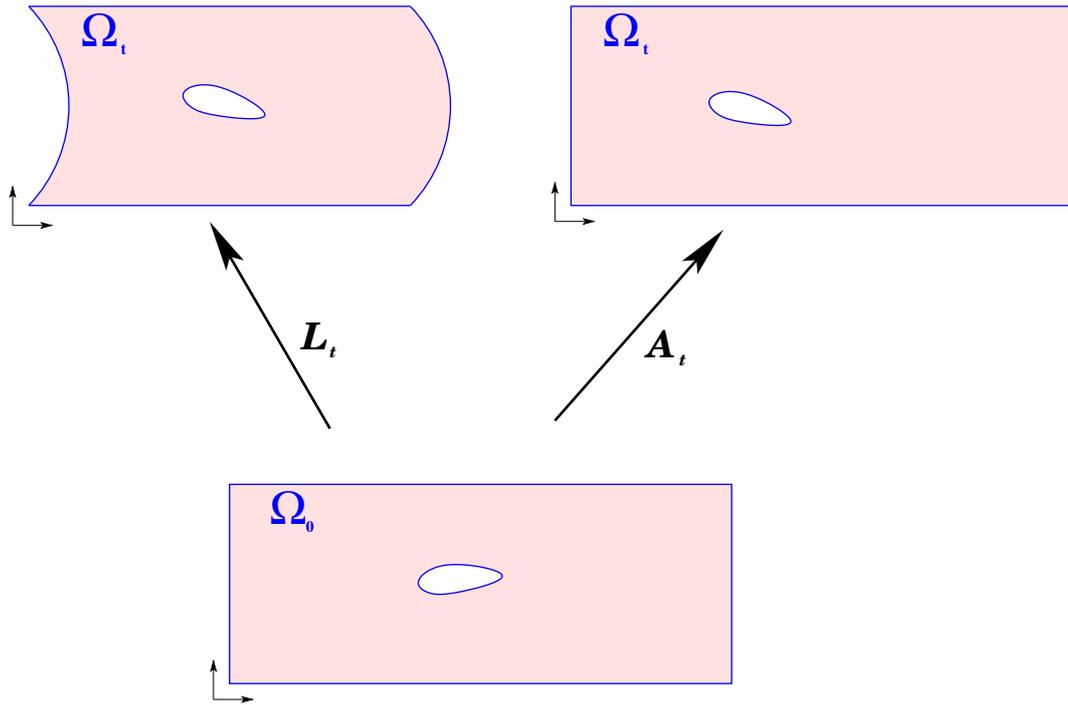


Figure 1. Comparison of Lagrangian and Arbitrary Lagrangian-Eulerian mappings. In this figure the demonstration of Lagrangian mapping (on the left) and ALE mapping (on the right) is shown. Although the Lagrangian mapping allows the structure to be deflected, the other (artificial) boundaries are also deformed, which is unusable in practical computations. ALE mapping is then the “compromise” between having fixed artificial boundaries and deflected the structure boundary.

2. Problem description

In this section the addressed aeroelastic model is presented. The fluid flow is described with the aid of the Reynolds Averaged Navier-Stokes (RANS) incompressible equations. The Reynolds stresses are modelled with the aid of the one equation Spallart-Almaras model. The aerodynamical forces are then evaluated and used in the structural model, which is presented here as the system of two ordinary differential equations. In order to describe the mathematical model for the case of moving meshes, the concept of Arbitrary Lagrangian-Eulerian formulation is briefly explained. The discretization of incompressible Navier-Stokes equations (INSE) can be considered as a special case of the RANS equations with turbulent viscosity ν_T set to $\nu_T \equiv 0$.

2.1. ARBITRARY LAGRANGIAN-EULERIAN FORMULATION

The numerical approximation of the time derivative by a time difference leads to complications in the case of time dependent domains and moving meshes. These complications are mainly caused by the fact that the grid points change their location during every time step. With the use of Arbitrary Lagrangian-Eulerian (ALE) method the original mathematical model can be reformulated in a

suitable way and the finite element space discretization together with a suitable time discretization can be introduced. The ALE method is based on the definition of an ALE mapping of the original configuration computational domain \mathcal{G}_0 onto the computational domain \mathcal{G}_t and the definition of the ALE domain velocity as the time derivative of the ALE mapping \mathcal{A}_t , i.e.

$$\mathcal{A}_t : \mathcal{G}_0 \mapsto \mathcal{G}_t, \quad \tilde{\mathbf{w}}_g = \frac{\partial \mathcal{A}_t}{\partial t}, \quad \mathbf{w}_g = \tilde{\mathbf{w}}_g \circ \mathcal{A}_t^{-1}.$$

With the aid of the time differentiation with respect to the original configuration \mathcal{G}_0 , leading to the so-called ALE derivative denoted by $\frac{D^{A_t}}{Dt}$, the time derivative of any function can be rewritten as $\frac{\partial}{\partial t} = \frac{D^{A_t}}{Dt} - (\mathbf{w}_g \cdot \nabla)$. For more details about ALE method, see, e.g., (Nomura and Hughes, 1992).

2.2. REYNOLDS AVERAGED NAVIER-STOKES EQUATIONS AND TURBULENCE MODELLING

Let us assume that at each time instant t the boundary \mathcal{G}_t is split into three disjoint parts $\partial \mathcal{G}_t = \Gamma_D \cup \Gamma_O \cup \Gamma_{W_t}$. The turbulent fluid flow is modelled with the numerical solution of Reynolds Averaged Navier-Stokes equations

$$\begin{aligned} \frac{\partial U_i}{\partial t} - \nu \sum_j \frac{\partial}{\partial x_j} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) + (\mathbf{U} \cdot \nabla) U_i + \frac{\partial P}{\partial x_i} = - \sum_j \frac{\partial}{\partial x_j} \overline{u'_i u'_j} + \mathbf{f}, \\ \nabla \cdot \mathbf{U} = 0, \end{aligned} \quad (1)$$

where the right hand side terms are so called Reynolds stresses $\sigma_{ij} = -\overline{u'_i u'_j}$.

The system (1) is equipped with the following boundary conditions

$$\begin{aligned} \text{a)} \quad & \mathbf{U} = \mathbf{U}_D, \quad \text{on } \Gamma_D, \\ \text{b)} \quad & \mathbf{U} = \mathbf{w}_g, \quad \text{on } \Gamma_{W_t}, \\ \text{c)} \quad & -\nu \sum_j \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \mathbf{n}_j + (P - P_{ref}) \mathbf{n}_i = \sum_j \sigma_{ij} \mathbf{n}_j, \quad \text{on } \Gamma_O, \end{aligned} \quad (2)$$

and with the initial condition $\mathbf{U}(x, 0) = \mathbf{U}_0(x)$ for $x \in \mathcal{G}_0$. If we set $\sigma_{ij} \equiv 0$, then the boundary condition (2,c) is reduced to the well-known “do-nothing” boundary condition. The Reynolds stress tensor $\sigma = (\sigma_{ij})$ requires further modelling. One possibility is to use the Boussinesq assumption consisting of taking σ in the form

$$\sigma_{ij} = -\frac{2}{3} k \delta_{ij} + \nu_T \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right)$$

In the present paper the turbulent kinematic viscosity is modelled with the aid of one-equation Spallart-Almaras model and the volumetric part $-\frac{2}{3} k \delta_{ij}$ is included in the pressure term. In this approach, the system of equations (1) is coupled with the following nonlinear partial differential equation

$$\frac{D^{A_t} \tilde{\nu}}{Dt} + ((\mathbf{U} - \mathbf{w}_g) \cdot \nabla) \tilde{\nu} = \frac{1}{\beta} \left[\sum_{i=1}^2 \frac{\partial}{\partial x_i} \left((\nu + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_i} \right) + c_{b2} (\nabla \tilde{\nu})^2 \right] + G - Y, \quad (3)$$

equipped with the boundary conditions $\tilde{\nu} = 0$ on Γ_{W_t} and $\frac{\partial \tilde{\nu}}{\partial \mathbf{n}} = 0$ on $\Gamma_O \cup \Gamma_D$. The functions G and Y are functions of the tensor of rotation of mean velocity Ω and of the wall distance y , i.e.

$$\begin{aligned} G &= c_{b_1} \tilde{S} \cdot \tilde{\nu}, & Y &= c_{w_1} \frac{\tilde{\nu}^2}{y^2} \left(\frac{1+c_{w_3}}{1+c_{w_3}/g^6} \right)^{\frac{1}{6}}, & \tilde{S} &= \left(S + \frac{\tilde{\nu}}{\kappa^2 y^2} f_{v_2} \right), & f_{v_2} &= 1 - \frac{\chi}{1+\chi f_{v_1}}, \\ g &= r + c_{w_2} (r^6 - r), & r &= \frac{\tilde{\nu}}{\tilde{S} \kappa^2 y^2}, & S &= \sqrt{2\Omega(\mathbf{U}) : \Omega(\mathbf{U})}, & \Omega(\mathbf{U}) &= \frac{1}{2}(\nabla \mathbf{U} - \nabla \mathbf{U}^T). \end{aligned}$$

The following choice of constants is used

$$\begin{aligned} c_{b_1} &= 0.1355, & c_{b_2} &= 0.622, & \beta &= \frac{2}{3}, & c_v &= 7.1, \\ c_{w_3} &= 0.3, & c_{w_3} &= 2.0, & \kappa &= 0.41, & c_{w_1} &= c_{b_1}/\kappa^2 + (1 + c_{b_2})/\beta. \end{aligned}$$

The Reynolds stresses then are computed as

$$\sigma_{ij} = -\nu_T \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right), \quad \nu_T = \tilde{\nu} \frac{\chi^3}{\chi^3 + c_v^3}, \quad \chi = \frac{\tilde{\nu}}{\nu},$$

where the volumetric part of σ has been included in the pressure term, i.e. $P^* = P + \frac{2}{3}k$. In what follows we shall not distinguish between P and P^* , we shall simply use the symbol P .

The space discretization of the problem is carried out by the finite element method, which starts from the so called weak formulation. To this end we introduce the velocity spaces W, X , the pressure space Q and the turbulence model space Λ :

$$W = (H^1(\mathcal{G}_t))^2, \quad X = \{\mathbf{v} \in W; \mathbf{v}|_{\Gamma_D \cup \Gamma_{W_t}} = 0\}, \quad Q = L^2(\mathcal{G}_t), \quad \Lambda = \{\phi \in W; \phi|_{\Gamma_{W_t}} = 0\}$$

where $L^2(\mathcal{G}_t)$ is the Lebesgue space of square integrable functions over the domain \mathcal{G}_t and $H^1(\mathcal{G}_t)$ is the Sobolev space of functions square integrable together with their first order derivatives.

Now, multiplying the system of equation (1) by test functions $\mathbf{v} \in X$ and $q \in Q$, integrating over the domain \mathcal{G}_t and using Green's theorem, we obtain the weak formulation: find $\mathbf{U} : \langle 0, T \rangle \mapsto W$ such that for all t the Dirichlet boundary conditions (2 a-b) are satisfied and $P : \langle 0, T \rangle \mapsto Q$ such that for all $t \in \langle 0, T \rangle$ the following equality holds

$$\mathbf{a}(\mathbf{U} - \mathbf{w}_g; \mathbf{U}, P; \mathbf{v}, q) = \mathbf{L}(\mathbf{v}, q), \quad \forall \mathbf{v} \in X, q \in Q \quad (4)$$

where

$$\begin{aligned} \mathbf{a}(\mathbf{b}; \mathbf{U}, P; \mathbf{v}, q) &= \left(\frac{D^{A_t} \mathbf{U}}{Dt}, \mathbf{v} \right)_{\mathcal{G}_t} + \nu \left(\nabla \mathbf{U}, \nabla \mathbf{v} \right)_{\mathcal{G}_t} + \sum_{i,j} \left(\sigma_{ij}(\mathbf{U}), \frac{\partial \mathbf{v}_i}{\partial x_j} \right)_{\mathcal{G}_t} \\ &\quad + \left((\mathbf{b} \cdot \nabla) \mathbf{U}, \mathbf{v} \right)_{\mathcal{G}_t} - \left(P, \nabla \cdot \mathbf{v} \right)_{\mathcal{G}_t} + \left(\nabla \cdot \mathbf{U}, q \right)_{\mathcal{G}_t}, \\ \mathbf{L}(\mathbf{v}, q) &= (\mathbf{f}, \mathbf{v})_{\mathcal{G}_t}. \end{aligned}$$

Now, by multiplying the equation (3) by a test function $\phi \in \Lambda$, integrating over the domain \mathcal{G}_t and using the Green's theorem, we obtain the weak formulation of the Spallart-Almaras turbulence

one-equation model: Find $\tilde{\nu} : [0, T] \mapsto \Lambda$ such that for all $\phi \in \Lambda$ and for any time $t \in [0, T]$ the following equation holds

$$\left(\frac{D^{\mathcal{A}_t} \tilde{\nu}}{Dt}, \phi \right)_{\mathcal{G}_t} + \left((\mathbf{U} - \mathbf{w}_g) \cdot \nabla \tilde{\nu}, \phi \right)_{\mathcal{G}_t} + \left((\nu + \tilde{\nu}) \nabla \tilde{\nu}, \nabla \psi \right)_{\mathcal{G}_t} + (Y, \psi)_{\mathcal{G}_t} = (G, \psi)_{\mathcal{G}_t} + \left(\frac{c_{b2}}{\beta} (\nabla \tilde{\nu})^2, \psi \right). \quad (5)$$

2.3. STRUCTURAL MODEL AND FLUID-STRUCTURE COUPLING

The nonlinear equations of motion for a flexibly supported body, see (Sváček, Feistauer, and Horáček, 2004), read

$$\begin{aligned} m \ddot{h} + S_\alpha \ddot{\alpha} \cos \alpha - S_\alpha \dot{\alpha}^2 \sin \alpha + k_{hh} h &= -L(t), \\ S_\alpha \ddot{h} \cos \alpha + I_\alpha \ddot{\alpha} + k_{\alpha\alpha} \alpha &= M(t), \end{aligned} \quad (6)$$

where the possibility of large values of α and h have been considered. For small values of the angle α , when $\alpha \approx 0$, $\sin \alpha \approx 0$ and $\cos \alpha \approx 1$, the system (6) can be rewritten in a simplified form (see, e.g., (Dowell, 1995), (Naudasher and Rockwell, 1994)). The aerodynamical forces acting on the airfoil can be evaluated

$$L = - \int_{\Gamma_{W_t}} \sum_{j=1}^2 \tau_{2j} n_j dS, \quad M = - \int_{\Gamma_{W_t}} \sum_{i,j=1}^2 \tau_{ij} n_j r_i^{\text{ort}} dS, \quad (7)$$

$$(8)$$

where $r_1^{\text{ort}} = -(x_2 - x_{EO2})$, $r_2^{\text{ort}} = x_1 - x_{EO1}$ and τ is the stress tensor, i.e.

$$\tau_{ij} = \rho \left[p \delta_{ij} + \nu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \right].$$

One should notice that the fluid flow model (1) and the structural model (6) can not be solved independently: clearly the aerodynamical forces $L(t)$ and $M(t)$, determined by the solution of the fluid flow model, appear in right hand side of (6) and, on the other hand, the deformation of the computational domain \mathcal{G}_t depends on the angle of rotation $\alpha = \alpha(t)$ and the translation $h = h(t)$, which form the solution of the system (6).

3. Discretization of the problem

3.1. SPACE-TIME DISCRETIZATION

First, we start with time partition $0 = t_0 < t_1 < \dots < T$, $t_k = k\Delta t$, with a time step $\Delta t > 0$ and approximate the function $\mathbf{U}(t_n)$, $P(t_n)$ and $\tilde{\nu}(t_n)$ defined in \mathcal{G}_{t_n} at time t_n by \mathbf{U}^n , P^n and $\tilde{\nu}^n$. The ALE derivative can be approximated by the finite differences

$$\left. \frac{D^{\mathcal{A}} \mathbf{u}}{Dt} \right|_{t^{n+1}} = \frac{3\mathbf{u}^{n+1} - 4\hat{\mathbf{u}}^n + \hat{\mathbf{u}}^{n-1}}{2\Delta t}, \quad \left. \frac{D^{\mathcal{A}} \tilde{\nu}}{Dt} \right|_{t^{n+1}} = \frac{3\tilde{\nu}^{n+1} - 4\hat{\tilde{\nu}}^n + \hat{\tilde{\nu}}^{n-1}}{2\Delta t}, \quad (9)$$



Figure 2. The fluid velocity and pressure isolines for inlet velocity $U = 25 \text{ m s}^{-1}$

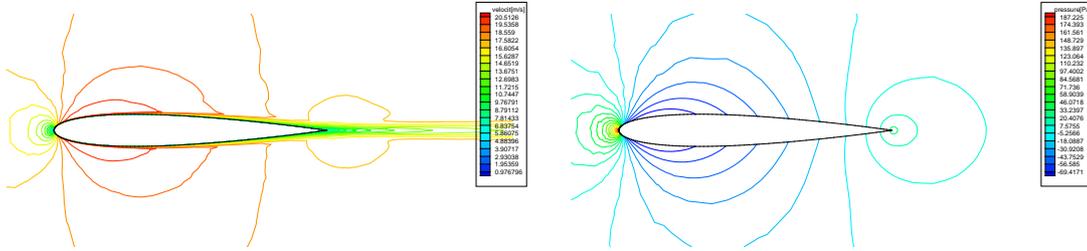


Figure 3. The time averaged fluid velocity and pressure isolines for inlet velocity $U = 25 \text{ m s}^{-1}$, stationary solution.

where for a function $f : \mathcal{G}_i \mapsto R$ the function $\hat{f}^i : \mathcal{G}_{n+1} \mapsto R$ is defined as $\hat{f}^i = f \circ \mathcal{A}_{t_i} \circ \mathcal{A}_{t_{n+1}}^{-1}$ at a fixed time step t_{n+1} . Then the form \mathbf{a} is modified in the following way:

$$\begin{aligned} \mathbf{a}(\mathbf{b}; \mathbf{U}, P; \mathbf{v}, q) &= \left(\frac{3\mathbf{U}^{n+1}}{2\Delta t}, \mathbf{v} \right)_{\mathcal{G}_{n+1}} + \nu \left(\nabla \mathbf{U}, \nabla \mathbf{v} \right)_{\mathcal{G}_{n+1}} + \sum_{i,j} \left(\sigma_{ij}(\mathbf{U}), \frac{\partial \mathbf{v}_i}{\partial x_j} \right)_{\mathcal{G}_{n+1}} \\ &\quad + \left((\mathbf{b} \cdot \nabla) \mathbf{U}, \mathbf{v} \right)_{\mathcal{G}_{n+1}} - \left(P, \nabla \cdot \mathbf{v} \right)_{\mathcal{G}_{n+1}} + \left(\nabla \cdot \mathbf{U}, q \right)_{\mathcal{G}_{n+1}}, \\ \mathbf{L}(\mathbf{v}, q) &= \left(\frac{4\hat{\mathbf{U}}^n - \hat{\mathbf{U}}^{n-1}}{2\Delta t}, \mathbf{v} \right)_{\mathcal{G}_{n+1}}, \end{aligned}$$

and the semi-implicit weak form of the Spallart-Almaras turbulence reads: Find $\tilde{\nu}^{n+1} \in \Lambda$ such that for all $\phi \in \Lambda$ holds the following equation

$$\mathbf{c}(\tilde{\nu}^{n+1}, \phi) = \mathbf{l}(\phi), \quad (10)$$

where

$$\begin{aligned} \mathbf{c}(\tilde{\nu}^{n+1}, \phi) &= \left(\frac{3\tilde{\nu}^{n+1}}{2\Delta t}, \phi \right)_{\mathcal{G}_{n+1}} + \left((\mathbf{U}^{n+1} - \mathbf{w}_g) \cdot \nabla \tilde{\nu}^{n+1}, \phi \right)_{\mathcal{G}_{n+1}} + \left(\frac{\nu + \tilde{\nu}^n}{\beta} \nabla \tilde{\nu}^{n+1}, \nabla \phi \right)_{\mathcal{G}_{n+1}} + \left(s^{(n)} \tilde{\nu}^{n+1}, \nabla \phi \right)_{\mathcal{G}_{n+1}} \\ \mathbf{l}(\phi) &= \left(\frac{4\hat{\nu}^n - \hat{\nu}^{n-1}}{2\Delta t}, \phi \right)_{\mathcal{G}_{n+1}} + \left(G^{(n)}, \phi \right)_{\mathcal{G}_{n+1}} + \left(\frac{c_{b2}}{\beta} (\nabla \hat{\nu}^n)^2, \phi \right)_{\mathcal{G}_{n+1}}, \end{aligned}$$

and

$$s^{(n)} = c_{w1} \frac{\tilde{\nu}^n}{y^2} \left(\frac{1 + c_{w3}^6}{1 + c_{w3}^6/g^6} \right)^{1/6}, \quad G^{(n)} = c_{b1} \bar{S} \hat{\nu}^n.$$

In order to apply the Galerkin FEM, we approximate the spaces W, X, Q from the weak formulation by finite dimensional subspaces $W_\Delta \subset W, Q_\Delta \subset Q, \Lambda_\Delta \subset \Lambda$ for $\Delta \in (0, \Delta_0)$ and we set

$$X_\Delta = \{ \mathbf{v}_\Delta \in W_\Delta; \mathbf{v}_\Delta|_{\Gamma_D \cap \Gamma_{Wt}} = 0 \}.$$

Hence, we define the *discrete problem* to find an approximate solution $\mathbf{U}_\Delta \in W_\Delta$ and $P_\Delta \in Q_\Delta$ such that \mathbf{U}_Δ satisfies approximately boundary conditions and the identity

$$a(\mathbf{U} - \mathbf{w}_g; \mathbf{U}, P; \mathbf{v}, q) = \mathbf{L}(\mathbf{v}, q), \quad \forall \mathbf{v}, q \quad (11)$$

The couple (X_Δ, Q_Δ) of the finite element spaces should satisfy the *Babuška-Brezzi (BB) inf-sup condition* (see, e.g. (Girault and Raviart, 1986)). In our computations, the well-known Taylor-Hood P_2/P_1 conforming elements on triangular meshes are used for the velocity/pressure approximation.

The standard Galerkin discretization (11) may produce approximate solutions suffering from spurious oscillations for high Reynolds numbers. In order to avoid this drawback, the stabilization via *Galerkin Least-Squares technique* is applied (see, e.g. (Lube, 1994), (Gelhard, Lube, and Olshanskii, 2003)). The stabilization terms are defined as

$$\begin{aligned} \mathcal{L}_\Delta(\mathbf{b}; \mathbf{U}, p; \mathbf{v}, q) &= \sum_{K \in \mathcal{T}_\Delta} \sum_{i=1}^2 \delta_K \left(\frac{3}{2\Delta t} U_i - \nu \Delta U_i + (\mathbf{b} \cdot \nabla) U_i + \frac{\partial P}{\partial x_i} - \sum_{j=1}^2 \frac{\partial \sigma_{ij}(\mathbf{U})}{\partial x_j}, (\mathbf{b} \cdot \nabla) v_i + \frac{\partial q}{\partial x_i} \right)_K, \\ \mathcal{F}_\Delta(\mathbf{v}) &= \sum_{K \in \mathcal{T}_\Delta} \sum_{i=1}^2 \delta_K \left(\frac{4\hat{U}_i^n - \hat{U}_i^{n-1}}{2\Delta t} + \mathbf{f}_i, (\mathbf{b} \cdot \nabla) v_i + \frac{\partial q}{\partial x_i} \right)_K, \end{aligned} \quad (12)$$

and the additional grad-div stabilization terms

$$\mathcal{P}_\Delta(\mathbf{U}, \mathbf{v}) = \sum_{K \in \mathcal{T}_\Delta} \tau_K (\nabla \cdot \mathbf{U}, \nabla \cdot \mathbf{v})_K, \quad (13)$$

are introduced with suitably chosen parameters $\delta_K \geq 0$ and $\tau_K \geq 0$.

The *stabilized discrete problem* reads: Find $\mathbf{U}_\Delta \in W_\Delta$ and $P_\Delta \in Q_\Delta$ such that \mathbf{U}_Δ satisfies approximately conditions (2), a), b) and

$$\begin{aligned} \mathbf{a}(\mathbf{U} - \mathbf{w}_g; \mathbf{U}, P; \mathbf{v}, q) + \mathcal{L}_\Delta(\mathbf{U} - \mathbf{w}_g; \mathbf{U}, P; \mathbf{v}, q) + \mathcal{P}_\Delta(\mathbf{U}, \mathbf{v}) &= \mathbf{L}(\mathbf{v}, q) + \mathcal{F}_\Delta(V_\Delta) \\ \text{for all } \mathbf{v}_\Delta \in X_\Delta, q_\Delta \in Q_\Delta. \end{aligned} \quad (14)$$

Furthermore, the approximate solution of the RANS system (1) is coupled with the Spallart-Almaras turbulence model given by the solution of (10). The nonlinear algebraic discrete system (14) and (10) is solved on each time level t_{n+1} with the aid of the linearized Oseen iterative process. More detailed description of Oseen iterative process can be found in (Sváček, Feistauer, and Horáček, 2004) for (laminar) Navier-Stokes equations.

4. Numerical results. Conclusions

In this paper we present the comparison of the presented method with NASTRAN computation and with the numerical simulation with the aid of Spallart-Almaras turbulence model. The parameters of the structural model was set as

$$m = 0.086622 \text{ kg}, \quad S_\alpha = -0.000779673 \text{ kg m}, \quad I_\alpha = 0.000487291 \text{ kg m}^2, \\ k_{hh} = 105.109 \text{ N m}^{-1}, \quad k_{\alpha\alpha} = 3.695582 \text{ N m rad}^{-1}, \quad l = 0.05 \text{ m}, \quad c = 0.3 \text{ m}.$$

The elastic axis is located at 40% of the airfoil, $\rho = 1.225 \text{ kg m}^{-3}$, $\nu = 1.5 \cdot 10^{-5} \text{ m s}^{-2}$. The numerical computations were performed for airfoils NACA 0012 (turbulent case) and NACA 63₂ – 415 (laminar case).

First, the numerical approximation of the coupled model with RANS equations was obtained for velocities in the range $5 - 40 \text{ m s}^{-1}$. The aeroelastic responses of the airfoil are shown in Figures 4, 5 and 6 for different values of the far field velocity U_∞ in the stable region. In Figure 7 the aeroelastic response for far field velocity $U_\infty = 38 \text{ m s}^{-1}$ is shown, where the coupled model is unstable (This is in agreement with NASTRAN computations by STRIP model, where the determined critical velocity was shown for $U_\infty = 37.7 \text{ m s}^{-1}$). In Figure 8 the comparison of the frequencies and damping coefficient determined from the aeroelastic response of the coupled model and frequencies and damping coefficient from NASTRAN computations (see (Čečrdle and Maleček, 2002)) is shown.

The similar computations were performed for (laminar) Navier-Stokes equations and the flow over an airfoil NACA 63₂ – 415. Figure 9 shows the behaviour of the coupled model in this case. The post-flutter behaviour in this case is shown in Figure 10. In order to validate the results for large large structural displacements the numerical simulation of vibrating airfoil was performed and compared to the experimental results. Figure 11 shows the streamlines patterns, which is in good agreement to the experimental results, see Naudasher and Rockwell, Figure 7.11.

The result shows that both laminar and turbulent approximation of fluid flow leads to comparable results, the determined critical velocity by the presented method is in agreement with the NASTRAN computation (Čečrdle and Maleček, 2002). The main difference is demonstrated in Figures 2 and 3, where for the turbulence model leads to the stationary solution in the case of fixed airfoil, which is not the case of ‘laminar’ simulations.

References

- Čečrdle J. and Maleček J. Verification FEM model of an aircraft construction with two and three degrees of freedom. *Tech. Rep. Research Report R-3418/02*, Aeronautical Research and Test Institute, Prague, Letňany, 2002.

- Dobeš, J., Ricchiuto, M., Deconinck, H., 2005. Implicit space-time residual distribution method for unsteady laminar viscous flows. *Computers and Fluids* 34 (4-5), 593–616, 2005.
- Dolejší, V. Anisotropic mesh adaptation technique for viscous flow simulation. *East-West Journal of Numerical Mathematics* 9 (1), 1–24, 2001
- Dowell, E. H., A Modern Course in Aeroelasticity. Kluwer Academic Publishers, Dodrecht, 1995
- Gelhard, T., Lube, G., Olshanskii, M. A., 2004. Stabilized finite element schemes with LBB-stable elements for incompressible flows. *Journal of Computational and Applied Mathematics* (accepted).
- Girault, V., Raviart, P.-A., Finite Element Methods for the Navier-Stokes Equations. Springer-Verlag, Berlin, 1986.
- LeTallec, P., Mouro, J., 1998. Fluid structure interaction with large structural displacements. In: 4th ECCOMASS Computational Fluid Dynamics Conference. pp. 1032–1039, 1998
- Lube, G., 1994. Stabilized Galerkin finite element methods for convection dominated and incompressible flow problems. *Numerical Analysis and Mathematical Modelling* 29, 85–104, 1994
- Naudasher, E., Rockwell, D. Flow-Induced Vibrations A.A. Balkema, Rotterdam, 1994.
- Nomura, T., Hughes, T. J. R., An arbitrary Lagrangian-Eulerian finite element method for interaction of fluid and a rigid body. *Computer Methods in Applied Mechanics and Engineering* 95, 115–138, 1992
- Sváček, P., Feistauer, M. Application of a stabilized FEM to problems of aeroelasticity. In: Feistauer, M., Dolejší, V., K., N. (Eds.), *Numerical Mathematics and Advanced Applications*, ENUMATH2003. Springer, Heidelberg, pp. 796–805, 2004
- Sváček, P., Feistauer, M., Horáček, J., Numerical simulation of flow induced airfoil vibrations with large amplitudes. *Journal of Fluids and Structures*(submitted).
- Wilcox, D. C., 1993. Turbulence Modeling for CFD. DCW Industries.

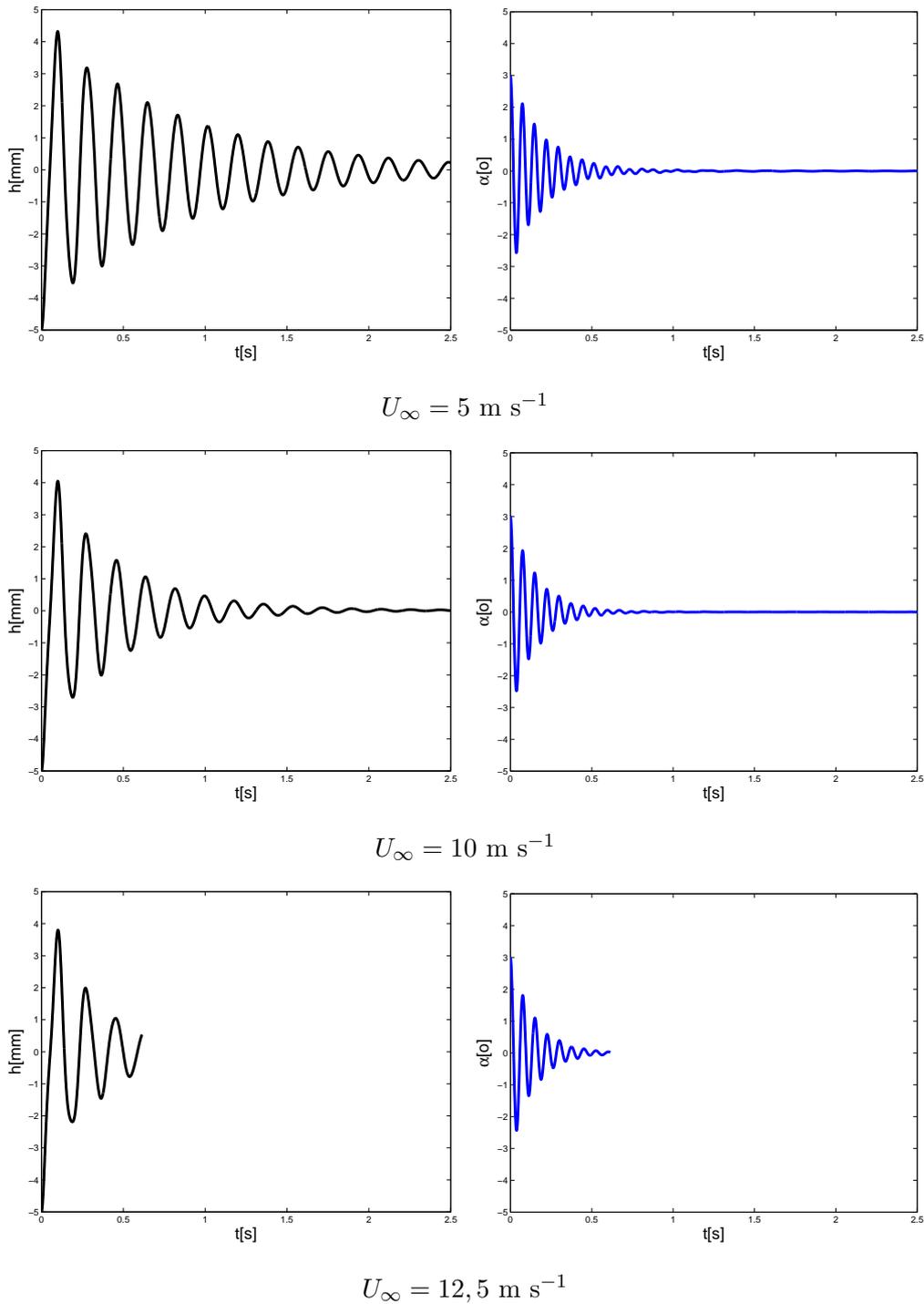


Figure 4. RANS simulations with Spallart-Almaras turbulence model for $U_{\infty} = 5, 10,$ and 12.5 m s^{-1} . The graph of the airfoil displacements in h (on the left) and α (on the right). In this case the coupled model is in stable region and two main frequencies can be identified in the aeroelastic response of the airfoil. Furthermore, with increasing far field velocity the aerodynamical damping is increasing.

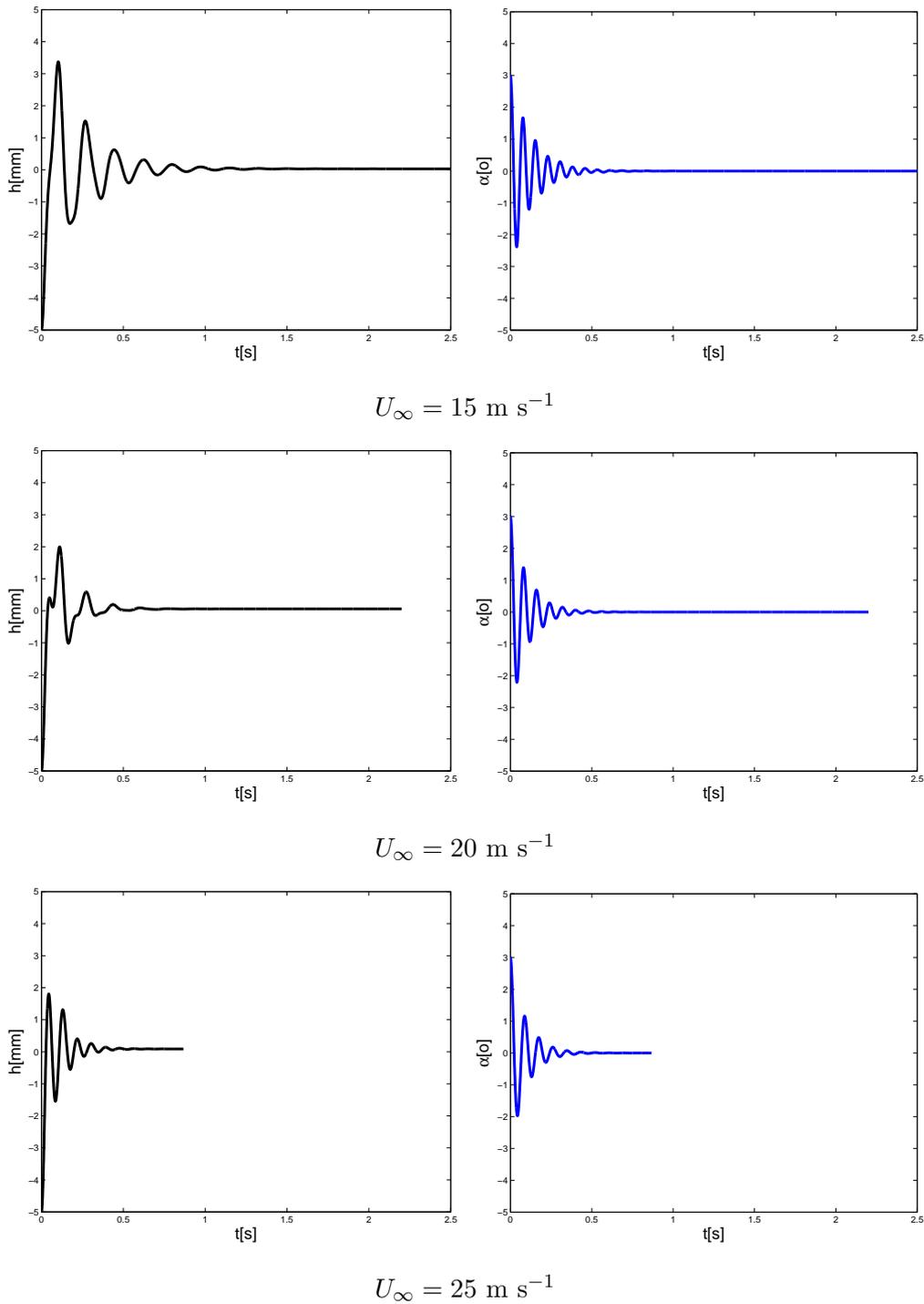
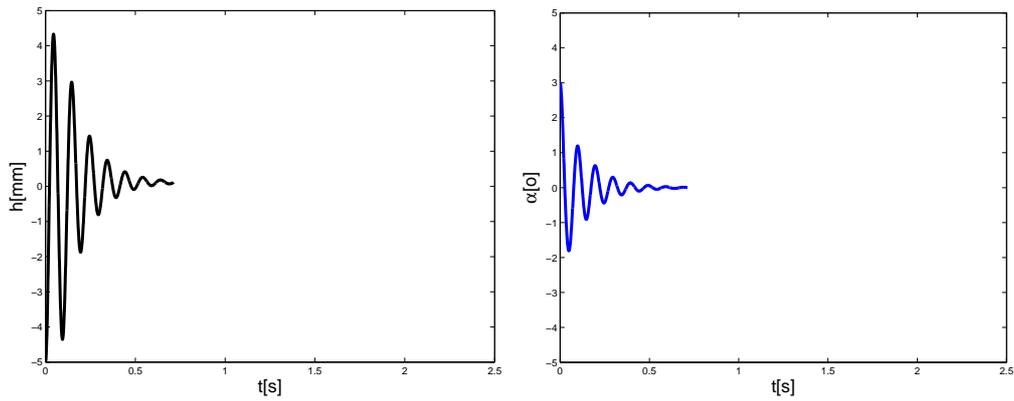
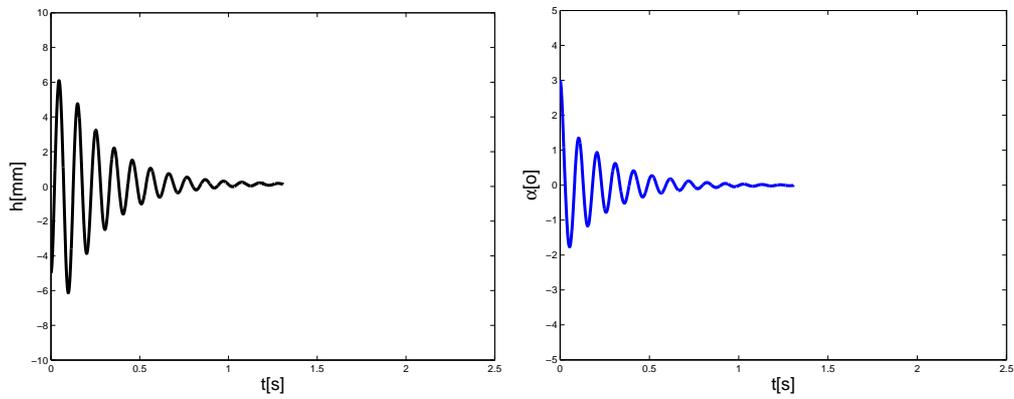


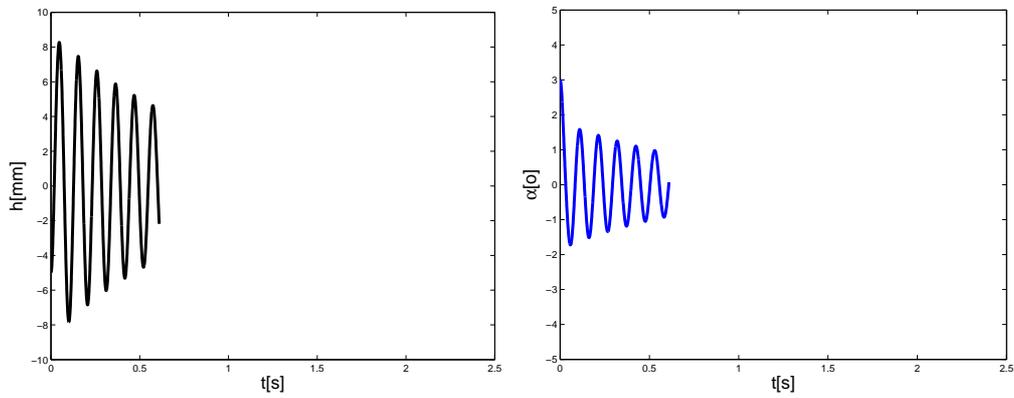
Figure 5. RANS simulations with Spallart-Almaras turbulence model for field velocity $U_{\infty} = 15, 20, 25 \text{ m s}^{-1}$. The graph of the airfoil displacements in h (on the left) and α (on the right). The aeroelastic behaviour is still in stable region, two main frequencies can be identified, for far field velocity $U_{\infty} = 25 \text{ m s}^{-1}$ the aerodynamical damping is maximal.



$$U_{\infty} = 30 \text{ m s}^{-1}$$



$$U_{\infty} = 32.5 \text{ m s}^{-1}$$



$$U_{\infty} = 35 \text{ m s}^{-1}$$

Figure 6. RANS simulations with Spallart-Almaras turbulence model for $U_{\infty} = 30, 32, 35 \text{ m s}^{-1}$. The graph of the airfoil displacements in h (on the left) and α (on the right). In this region of velocities only one frequency can be identified in the aeroelastic response of the airfoil and with increasing far field velocity the aerodynamical damping starts to be decreasing.

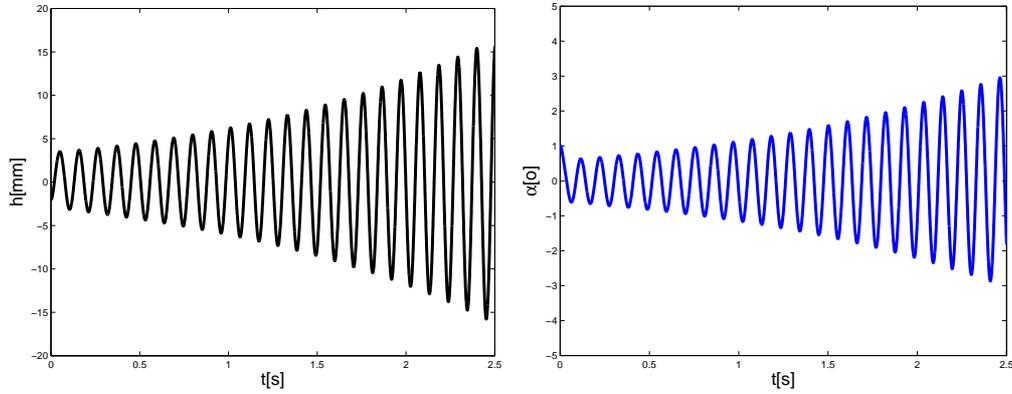


Figure 7. RANS simulations with Spallart-Almaras turbulence model for post-critical velocity $U_\infty = 38 \text{ m s}^{-1}$. For this value of far field velocity the aeroelastic problem is unstable, the vibrations slowly increases in time.

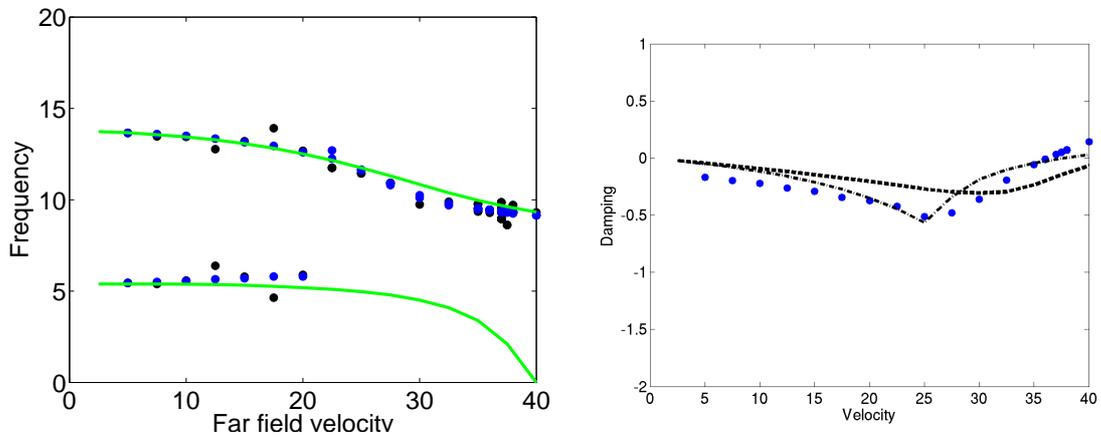


Figure 8. Comparison of frequency and damping for the aeroelastic response for the presented FE simulations of RANS equations and NASTRAN computations.

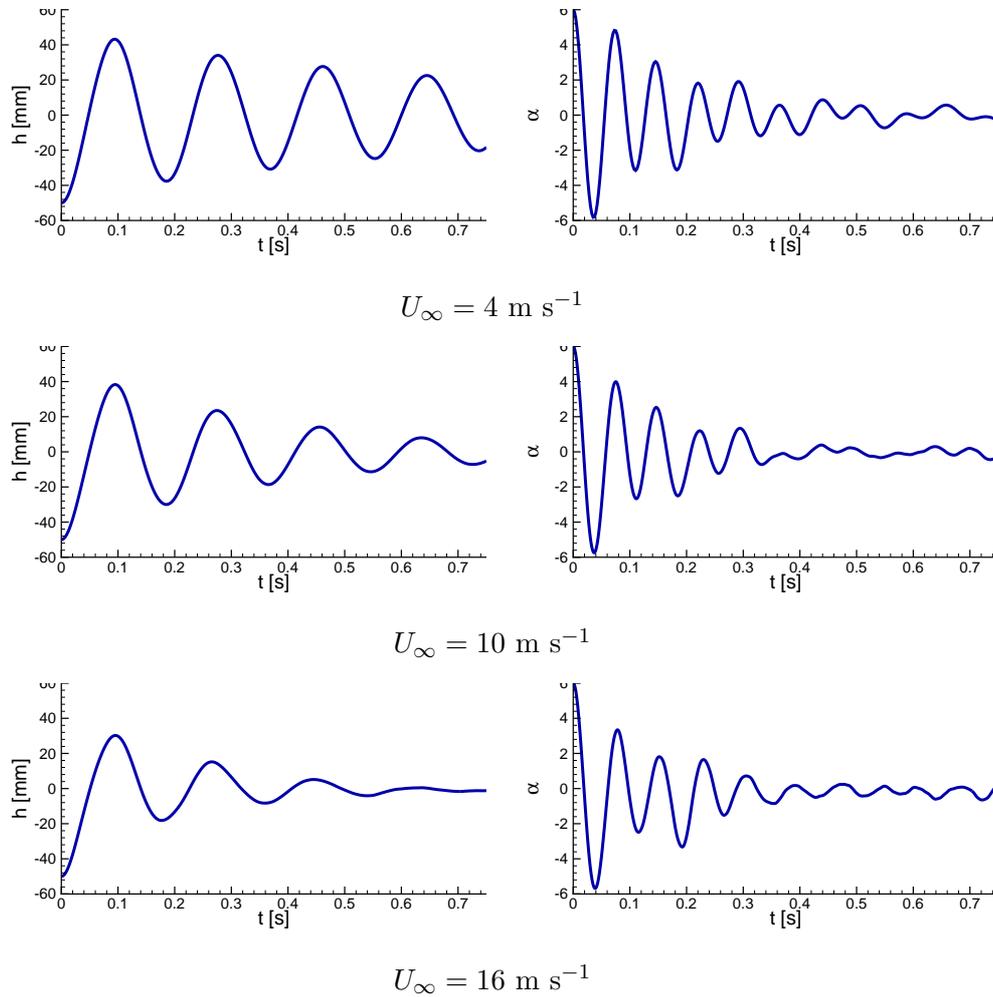


Figure 9. Navier-Stokes equations (laminar) simulations for the aeroelastic simulations and subcritical velocity $U_{\infty} = 5, 10, \text{ and } 16 \text{ m s}^{-1}$

The graph of the airfoil displacements in h (on the left) and α (on the right). In this case the coupled model is in stable region and two main frequencies can be identified in the aeroelastic response of the airfoil. Furthermore, with increasing far field velocity the aerodynamical damping is increasing. For the far field velocity $U_{\infty} = 16 \text{ m s}^{-1}$ the vibrations are not fully damped as it was the case for the RANS simulations, but the aeroelastic model still remains clearly stable.

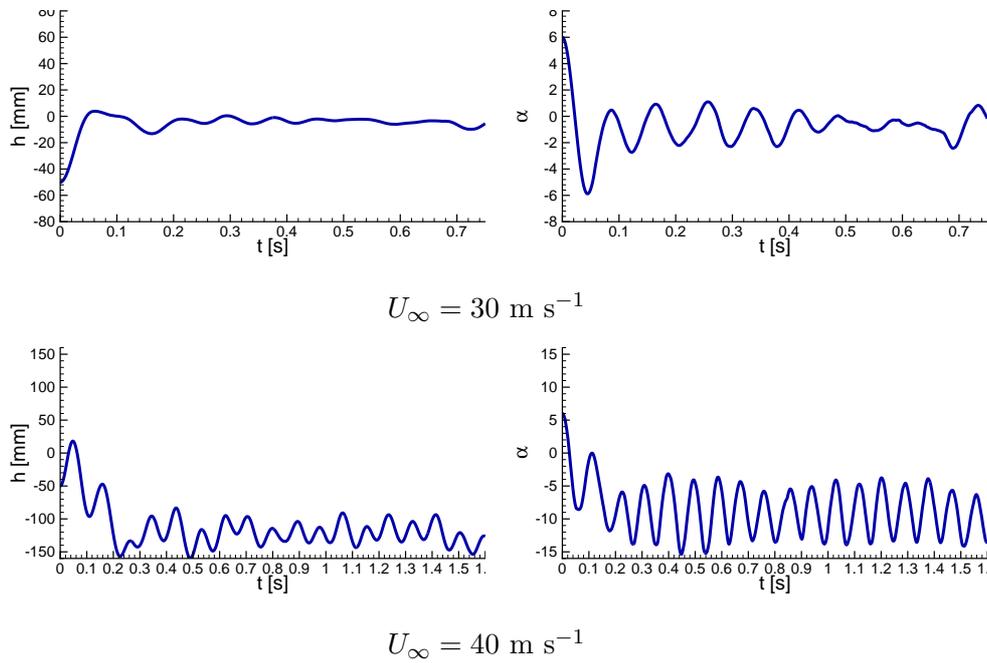


Figure 10. Navier-Stokes equations (laminar) simulations for the aeroelastic simulations and for sub-critical velocity $U_\infty = 30 \text{ m s}^{-1}$ and for post-critical velocity $U_\infty = 40 \text{ m s}^{-1}$

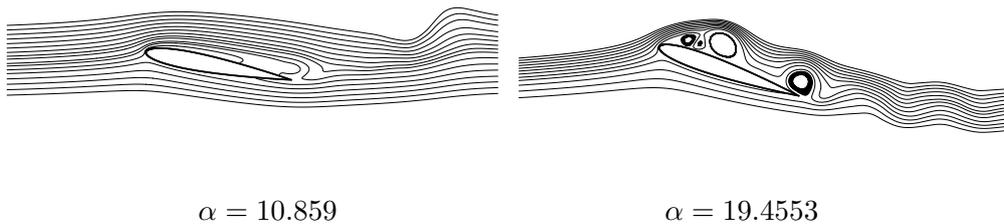


Figure 11. Incompressible (laminar) Navier-Stokes equations simulations. Instantaneous streamline patterns for vibrating airfoil $Re = 5000$, $\alpha = 10^\circ + 10^\circ \sin(2\pi f_s t)$ at $f_s c / U_\infty = 1/2\pi$ showing the ‘dynamic stall vortex’ (after (Naudasher and Rockwell, 1994))

Interval Finite Element Methods: New Directions

Rafi Muhanna¹, Vladik Kreinovich², Pavel Šolín²,
Jack Chessa², Roberto Araiza², and Gang Xiang²

¹*Center for Reliable Engineering Computing (REC),*

*Department of Civil & Environmental Engineering, Georgia Institute of Technology,
210 Technology Circle, Savannah, GA 31407-3038, USA,
rafi.muhanna@gtrep.gatech.edu*

²*University of Texas at El Paso, El Paso, TX 79968, USA,
vladik@utep.edu, solin@utep.edu, jfchessa@utep.edu,
raraiza@utep.edu, gxiang@utep.edu*

Keywords: interval FEM, *hp*-FEM, discrete maximum principles

1. Introduction

Many problems in computational engineering and science, such as solid and fluid mechanics, electromagnetics, heat transfer, or chemistry, are sufficiently well described on the macroscopic level in terms of partial differential equations (PDEs). In practice, these processes may be very complex, and the presence of multiple spatial and/or temporal scales, or even discontinuities in the solution, often makes their computer simulation challenging. There exist advanced numerical methods to tackle these problems, such as finite element methods (FEM). Lately, new advanced version of these methods have appeared, such as hierarchic higher-order finite element methods (*hp*-FEM) and extended finite element methods (X-FEM). Most of these methods work on a traditional basis where no uncertainty considerations are present in the modeling or computation. However, the need for numerical treatment of uncertainty becomes increasingly urgent. In many cases a given problem can be solved efficiently and accurately for a given set of input data (such as geometry, boundary conditions, material parameters, etc.), but little can be said about how the solution depends on uncertainties in these parameters.

However, the design of an engineered system requires the performance of the system to be guaranteed over its lifetime. One of the major difficulties a designer must face is that neither the external demands of the systems nor its manufacturing variations are known exactly. In order to overcome this uncertainty, the designer must provide excessive capabilities and over design the system. As analysis tools continue to be developed, the predictive skills of designers have become finer. In addition, the demands of the market place require that more efficient designs be developed. In order to satisfy these current requirements in designs subject to uncertainties, the uncertainties in the performance of the system must be included in the analysis.

At present, analytical and Monte-Carlo techniques are used to handle probabilistic uncertainty, and interval finite element methods are used to handle interval uncertainty. In many practical situations, we have both probabilistic and interval uncertainty. The problem of efficient combination of

probabilistic and interval uncertainties have to be explored for problems where neither Monte Carlo nor standard interval methods can be used. Therefore, advanced interval arithmetic techniques, ideally handling probabilistic uncertainty as well, need to be implemented into modern finite element methods both on the practical and theoretical levels. When developing these techniques, we need to take into account recent developments in interval computation techniques are their applications and developments in promising finite element techniques such as *hp*-FEM and X-FEM, together with results obtained with interval finite element methods for problems of structural mechanics (reviewed in Section 2).

2. Interval Finite Element Methods: A Brief Overview

There are various ways in which the types of uncertainty might be classified. One is to 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 (Melchers, 1999). 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, traditional probabilistic modeling techniques cannot handle situations with incomplete or little information on which to evaluate a probability, or when that information is nonspecific, ambiguous, or conflicting (Walley, 1991; Ferson and Ginzburg, 1996; Sentz and Ferson, 2002). Many generalized models of uncertainty have been developed to treat such situations, including fuzzy sets and possibility theory (Zadeh, 1978), Dempster-Shafer theory of evidence (Dempster, 1967; Shafer, 1976), random sets (Kendall, 1974), probability bounds (Berleant, 1993; Ferson and Ginzburg, 1996; Ferson et al., 2003), imprecise probabilities (Walley, 1991), convex models (Ben-Haim and Elishakoff, 1990), and others.

These generalized models of uncertainty have a variety of mathematical descriptions. However, they are all closely connected with interval analysis (Moore, 1966), in which imprecision is described by an interval (or, more generally, a set). For example, a fuzzy number can be viewed as a nested collection of intervals corresponding to different levels of confidence α (so-called α -cuts). Thus, the mathematical analysis associated with fuzzy set theory can be performed as interval analysis on different α -levels (Muhanna and Mullen, 1995; Lodwick and Jamison, 2002), and fuzzy arithmetic can be performed as interval arithmetic on α cuts. A Dempster-Shafer structure (Dempster, 1967; Shafer, 1976) 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 (Berleant, 1993; Ferson and Ginzburg, 1996; Ferson et al., 2003) is a combination 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 within the context of the finite element method 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 fur-

ther rigorous formulation. Finite element method with interval valued parameters results in Interval Finite Element Method (IFEM). The numerical solution of an IFEM is the focus of this section. Different formulations of IFEM have been developed. The use of IFEM solution techniques can be broadly classified into two groups, namely the optimization approach and the non-optimization approach.

In the optimization approaches (Koçluoglu et al., 1995; Rao and Chen, 1998; Akpan et al., 2001; Möller et al., 2000), optimizations are performed to compute the minimal and maximal structural responses when the uncertain parameters are constrained to belong to intervals. This approach often encounters practical difficulties. Firstly it requires efficient and robust optimization algorithm. In most structural engineering problems, the objective function is nonlinear and complicated, thus often only an approximate solution is achievable. Secondly, this approach is computationally expensive. For each response quantity, two optimization problems must be solved to find the lower and the upper bounds.

More recently, non-optimization approaches for the interval finite element analysis have been developed in a number of papers. For linear elastic problems, this approach leads to a system of linear interval equations, then the solution is sought using various methods developed for this purpose. The major difficulty associated with this approach is the “dependency problem” (Moore, 1979; Neumaier, 1990; Hansen, 1992; Muhanna and Mullen, 2001). In general, dependency problem arises when one or several interval variables occur more than once in an interval expression. The dependency in interval arithmetic leads to an overestimation of the solution. 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 IFEM). Usually such a use results in meaninglessly wide and even catastrophic results (Muhanna and Mullen, 2001).

In the non-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 (Muhanna and Mullen, 1995; Rao and Berke, 1997). This approach gives exact solution in special cases of linear elastic problems. However, it is computationally tedious and expensive, and is limited to the solutions of small-scale problems.

A convex modeling and superposition approach was proposed to analyze load uncertainty in (Pantelides and Ganzerli, 2001), and exact solution was obtained. However, the superposition is only applicable to load uncertainty.

A combinatorial approach was used in (Ganzerli and Pantelides, 1999) to treat interval modulus of elasticity.

A static displacement bounds analysis was developed in (Chen et al., 2002) have developed using matrix perturbation theory. The first-order perturbation was used and the second-order term was neglected. The result is approximate and not guaranteed to contain the exact bounds.

The paper (McWilliam, 2000) 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.

In (Dessombz et al., 2001), an interval FEM was introduced in which the interval parameters were factored out during the assembly process of the stiffness matrix. Then an enhanced iterative

algorithm from (Rump, 1983) 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 (Muhanna and Mullen, 1995; Mullen and Muhanna, 1996; Mullen and Muhanna, 1999), 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, in (Muhanna and Mullen, 2001), an interval FEM was developed based on an element-by-element technique and Lagrange multiplier method. Uncertain modulus of elasticity was considered. Most sources of overestimation were eliminated, and a sharp result for displacement was obtained. However, this formulation can only handle uncertain modulus of elasticity, and it can not obtain the sharp enclosures for element internal forces.

A new formulation for interval finite element analysis for linear static structural problems is developed in the work of (Muhanna et al., 2005). Material and load uncertainties are handled simultaneously, and sharp enclosures on the system's displacement as well as the internal forces are obtained efficiently.

Recently, new advancements has been made in the area of interval FEM, e.g., (Corliss et al., 2004; Popova et al., 2003), and the significant development in (Neumaier and Pownuk, to appear) where sharp results are achieved for linear truss problems even with large uncertainty.

3. First Challenge: Combination of Interval and Probabilistic Techniques

In many problems, e.g., of fundamental physics, one knows the exact equations, one knows the exact values of the parameters of these equations, and all one needs is to solve these equations as fast and as accurately as possible. These are the cases when the traditional FEM techniques directly lead to practically useful results. In engineering practice one approximates both the actual computational domain and function space using a collection of finite elements, the FEM solution only is an approximation to the actual continuous field, but as one increases the number and/or polynomial degree of the finite elements (using h , p , or hp -adaptivity), the FEM results become more and more accurate, and at some point one gets the desired solution with a very high accuracy.

There are many other application problems, however, where one only knows the approximate equations, or where one knows the equations, but one only knows the approximate values of the corresponding parameters. For example, in many civil engineering problems, one does not know the exact values of the Young modulus; one only knows the bounds for these values coming from the fact that one knows the material, and one knows the bounds for this type of material. In such problems, even if one uses an extremely fine mesh to make the discretization error negligible, the resulting FEM solution may still be very different from the actual behavior of an analyzed system – because of the uncertainty in the parameters and/or equations.

In such situations, to make the FEM results practically useful, one must be able to estimate how different the true and approximate solutions can be. In other words, one needs to be able to estimate how the uncertainty in the parameters of the system can affect the FEM results.

This question is of paramount importance in science and engineering, and, of course, there has already been a lot of research aiming to answer this question. Most of this research is based on

the assumption that one knows the exact probability distributions corresponding to all uncertain parameters. In this stochastic FEM case one can, in principle, apply the Monte Carlo method: Simulate all the parameters according to their known distributions, apply FEM for the system with the simulated values of the corresponding parameters, and then perform the statistical analysis of the FEM results – and thus, get the probability distribution for these results.

This stochastic FEM approach works well in many practical situations. In many other situations, on the other hand, the probabilities of different values of the uncertain parameters are not known. For example, in civil engineering one often only knows the lower and upper bounds on the Young modulus, but the probabilities of different values within the corresponding interval may depend on the manufacturing process, and thus they may differ from one building to another dramatically. In situations which require reliable estimates, e.g., when one analyzes the stability of a building, it is not enough to select one possible distribution and confirm that the building is stable under this distribution; to get a reliable result, one must make sure that the building remains stable for all possible distributions on the given interval.

Lately, there has been a lot of progress in applying interval computation techniques to FEM with interval uncertainty. This area of research was started in the early 1990s, and it was advanced in the series of papers reviewed in Section 3.

The software tools developed recently by R. Muhanna in the U.S., as well as similar tools developed by A. Neumaier in Austria, allowed us to prove reasonable interval FEM estimates – at least for the situations like civil engineering, when one can get a reasonable description of a structure by using several hundreds of finite elements only. These methods have led to very useful practical applications to the reliability of buildings and associated problems.

However, there still are practical problems for which the interval FEM is not fully adequate. As of now, there are two main methods to handle uncertainty in FEM problems:

- Stochastic FEM methods for situations when one knows the exact probability distribution of all uncertain parameters.
- Interval FEM methods for situations when no information about the probability distributions is available – one only knows the intervals of possible value of these parameters.

In other words, at present one only knows how to handle uncertainty in two extreme situations:

- One has full information about the probabilities.
- One has no information about the probabilities.

Many practical situations lie in between these two extremes: one has a partial information about the probabilities. For example, one may also have interval bounds for some of the parameters, but one may know the probability distribution for other parameters. For example, one may know only intervals of possible values of the manufacturing-related parameters, but, when one has good records, one may also know probabilities of different values of, say, weather-related parameters.

It is therefore highly desirable to extend the interval and stochastic FEM techniques to the case when one has a combination of interval and probabilistic uncertainty. Extension of interval and statistical methods to such a technique is, at present, an active area of research. While these

combined techniques have been developed and applied to different practical situations, there are still very few applications to FEM.

Our preliminary results have already led to an idea of such an extension for an important case when one has interval uncertainty for some parameters and probabilistic uncertainty for some other parameters. In such situations, one can apply Monte Carlo techniques to simulate parameters with known probability distributions. For each such simulation, one can then use interval FEM techniques to take into account the corresponding interval uncertainty. As a result of applying interval FEM techniques, one gets the interval bounds for the resulting FEM inaccuracy. By repeating this simulation several times, one gets several bounds – and hence, the resulting bounds distribution. By using this bounds distribution, one can now supplement the interval FEM information that the FEM inaccuracy Δy is bounded by a certain value Δ with the information that with probability 90%, one can get a narrower bound that bounds Δy in at least 90% of the case, yet narrower bound which holds in at least 80% of the cases, etc. Similar techniques need to be developed and applied to more complex situations with combined interval and probabilistic uncertainty.

Comment. The above idea is applicable in situations in which we already have well-developed interval FEM techniques. Another important research topic is the extension of interval FEM techniques to other advanced FEM techniques such as *hp*-FEM. These adaptive higher-order FEM techniques has proved to be superior to traditional lowest-order FEM in many practical problems, both in terms of higher accuracy and dramatically smaller size of the resulting stiffness matrices and substantially shorter CPU time; see, e.g., see (Demkowicz et al., 2001; Šolín, 2005) and the references therein.

4. Second Challenge: Nonlinear FEM with Stochastic Variations and Uncertainty for Microstructure

Significant amount of work was done in the use of both the probabilistic and non-probabilistic finite element methods for the assessment of uncertainty for linear PDEs. Several methods have proven to be successful: stochastic methods, interval methods, fuzzy number methods (Elishakoff and Ren, 1999; Haldar and Mahadevan, 2000; Schuëller, 2001). These approaches have been primarily applied to problems academic in nature. The issue of uncertainty and verification in practical engineering problems still seems to be a little addressed issue. By verification one is referring to the definition from (Babuška and Oden, 2004), where correct empirically derived model parameters are used.

An area of emerging importance is the application of stochastic and interval finite element methods to nonlinear continuum mechanics problems. Specifically, effects of uncertainty in the microstructural state of materials need to be studied. In this area, enriched finite element methods, particularly the extended finite element methods (X-FEM) (Moës et al., 1999; Belytschko et al., 2001; Stazi et al., 2003), need to be combined with interval and stochastic methods to investigate the effect of uncertainty on the position and state of the microstructure.

The X-FEM uses a local partition of unity technique to construct finite elements which are capable of reproducing discontinuities and singularities without mesh refinement. This approach has been used to model crack growth (Moës et al., 1999; Chen and Belytschko, 2003; Stazi et al.,

2003), material inhomogeneities (Sukumar et al., 2000; Chessa et al., 2003) as well as various other phenomena (Chessa et al., 2002; Chessa and Belytschko, 2003; Chessa and Belytschko, to appear). In all of these methods, the location of the material interfaces is implicitly defined by a level set field (Sethian, 1999). Thus, material models with a significantly increased number of defects and inclusions are computationally tractable.

This technique should be extended to non-linear problems of fracture mechanics, e.g., to non-linear Stefan-type equations that describe the dynamics of crack growth.

In principle, both for linear and nonlinear problems, we can use a straightforward perturbation approach as in (Liu et al., 1999). However, such approaches allow for only small variations in the variables. To allow for large stochastic variations, a combined approach of interval finite element methods and homogeneous chaos methods need to be developed.

5. Third Challenge: Enhancing *hp*-FEM with Advanced Interval Techniques

The *hp*-FEM is distinguished from the traditional FEM by combining elements of variable size and polynomial degree to achieve extremely fast convergence. The method originates in the early works of I. Babuška et al. (Babuška and Gui, 1986; Babuška et al., 1999). In the last few years, significant progress was made towards the solution of practical problems related to the computer implementation of the *hp*-FEM (design of optimal algorithms and data structures, automatic *hp*-adaptive strategies, optimal higher-order shape functions, etc.), see (Ainsworth and Senior, 1997; Karniadakis and Sherwin, 1999; Paszynski et al., 2004; Rachowicz et al., 2004; Šolín et al., 2003; Šolín and Demkowicz, 2004). Typically, the *hp*-FEM is capable of solving PDE problems using dramatically fewer degrees of freedom compared to standard FEM. Several such examples, obtained using a modular *hp*-FEM system HERMES which is being developed at the University of Texas at El Paso, are presented in the recent monograph (Šolín, 2005). It is therefore desirable to extend interval FEM techniques to *hp*-FEM.

We believe that for *hp*-FEM, the existing interval techniques will be even more efficient than for more traditional FEM techniques. Indeed, one of the main advantages of *hp*-FEM in comparison to lowest-order methods is that in many practical situations, for the same approximation accuracy, *hp*-FEM techniques require dramatically fewer degrees of freedom (unknown solution coefficients). In other words, they can decrease the size of the matrices in the corresponding linear systems substantially. When we solve systems of linear equations with interval uncertainty, in general, we get enclosures with excess width, and this excess width drastically increases with the size of a system. Thus, the decrease in the system's size will allow us to get more accurate estimates for the resulting interval uncertainty.

6. Fourth Challenge: Using Interval Computations to Prove Results about FEM Techniques

Finally, it is desirable to use interval computation techniques – techniques which provide guaranteed bounds for functions on continuous domains – in proving results about FEM methods, results which

should be valid for all possible values of the corresponding parameters. In this section, we describe our preliminary results in this direction and related challenges.

6.1. FORMULATION OF THE PROBLEM

Our preliminary results are related to elliptic partial differential equations $Lu = f$. The simplest case to begin with is the one-dimensional Poisson equation $-u'' = f$ with homogeneous Dirichlet boundary conditions.

For elliptic differential equations $Lu = f$, there is a known *Maximum Principle*: If $f(x) \leq 0$ for all points x from the domain Ω , then (under reasonable smoothness conditions) the solution u attains its maximum on the boundary of Ω . Because of the maximum principle:

- for the same f , we have a continuous dependence of the solution on the boundary conditions: namely, if u_1 and u_2 are two solutions with the same right-hand side f , then the sup-norm distance $\sup_{x \in \Omega} |u_1(x) - u_2(x)|$ between u_1 and u_2 (defined as the supremum over *all* x from Ω) is equal to the supremum $\sup_{x \in \partial\Omega} |u_1(x) - u_2(x)|$ of the difference over the boundary $\partial\Omega$ of the domain Ω ;
- similarly, there is a continuous dependence of u on f .

This allows us to provide *guaranteed bounds* on the solution based on the uncertainty with which we know the right-hand side f and the boundary values of u .

In the Finite Element Method we consider piecewise-polynomial functions $u_{h,p}(x)$ (which span a finite-dimensional space $V_{h,p}$). Of course, $u_{h,p}$ is not an exact solution of the original problem $Lu = f$. Instead, we look for an exact solution to the *discrete weak formulation* of the partial differential equation, see, e.g., (Šolín, 2005).

It is known that, sometimes, $f(x) \leq 0$ for all $x \in \Omega$, but the maximum of the resulting finite element solution $u_{h,p}$ is not necessarily attained on the boundary. As a result,

- even when we know the bounds on the uncertainty in f and in the boundary conditions, it is difficult to find guaranteed bounds on the uncertainty in $u_{h,p}$,
- the approximate solution $u_{h,p}$ may be unphysical, e.g., it may attain negative values when it represents absolute temperature, concentration, etc.

It is therefore desirable to find discrete analogues of the classical maximum principles, which are called *discrete maximum principles*. Such analogues are known for lowest-order (piecewise linear) FEM since the early 1970s (Ciarlet, 1970; Ciarlet et al., 1973). For the latest results, see, e.g., (Korotov et al., 2000; Křížek and Liu, 2003; Karátson and Korotov, 2005).

Until recently, no extensions to higher-order FEM were known. Moreover, a rather discouraging result (Höhn and Mittelman, 1981) stated that the discrete maximum principle did not hold for the Poisson equation $-u'' = f$ discretized with quadratic elements except with unrealistic conditions on the triangulation. After that, it was assumed for a long time that no discrete maximum principles for hp -FEM can be proved.

In (Šolín and Vejchodský, 2005), we solved the Poisson equation in one spatial dimension, equipped with homogeneous Dirichlet boundary conditions $u(-1) = u(1) = 0$ and with a right-hand side $f(x) = 200 \cdot e^{-10 \cdot (x+1)}$. According to the standard maximum principle, the actual solution $u(x)$ is nonnegative in the entire interval $(-1, 1)$. Let us consider this whole domain as a single element, and let us approximate the desired solution by a 3-rd degree polynomial $u_{h,p}(x)$ which satisfies the desired boundary conditions $u_{h,p}(-1) = u_{h,p}(1) = 0$. We want $-\int_{-1}^1 u'_{h,p}(x)v'(x) - f(x)v(x) dx = 0$ for all 3-rd other polynomials $v(x)$ such that $v(\pm 1) = 0$.

Due to linearity of the problem, the satisfaction of this integral condition for *all* these polynomials is equivalent to the fact that this condition must hold for any basis, for example, $v_1(x) = 1 - x^2$, $v_2(x) = x(1 - x^2)$. Thus, in terms of the coefficients of the unknown polynomial $u_{h,p}(x)$, we get an easy-to-solve system of linear equations, whose solution

$$u_{h,p}(x) = \frac{1}{40} \cdot [54 + 66 \cdot e^{-20} - (73 - 133 \cdot e^{-20}) \cdot x] \cdot (1 - x^2)$$

is negative, e.g., at $x = 0.9$.

6.2. FORMULATION OF THE RESULT

The reason for the above negativity is that, as one can easily check, the weak solution corresponding to the original function $f(x)$ is the same as the weak solution corresponding to the *projection* $f_{h,p}(x)$ of the function $f(x)$ on the set of polynomials of 3-rd degree – i.e., for the 3-rd degree polynomial $f_{h,p}(x)$ for which $\int (f(x) - f_{h,p}(x)) \cdot v(x) dx = 0$ for all 3-rd degree polynomials $v(x)$. For the above function $f(x)$, the projection

$$f_{h,p}(x) = -8.25 + 29.175 \cdot x + 54.75 \cdot x^2 - 93.625 \cdot x^3$$

is no longer nonnegative: e.g., it is negative for $x = 0$.

It is therefore reasonable to ask whether the Discrete Maximum Principle for higher-order FEM holds if we restrict ourselves to the case when not only the function $f(x)$ is nonnegative, but its projection $f_{h,p}(x)$ (i.e., the polynomial of the corresponding degree) is nonnegative as well.

So, we arrive at the following problem. For some integer p , we have a p -th degree polynomial $f_{h,p}(x)$ defined on the interval $(-1, 1)$. We are looking for a weak solution $u_{h,p}(x)$ to the equation $-u'' = f$ with the boundary conditions $u(-1) = u(1) = 0$, i.e., for a polynomial $u_{p,h}(x)$ of p -th degree for which $\int_{-1}^1 (-u''_{h,p}(x) - f(x)) \cdot v(x) dx = 0$ for all polynomials $v(x)$ of degree p . We want to prove that if the polynomial $f_{h,p}(x)$ is nonnegative on the entire interval $(-1, 1)$, then the weak solution $u_{h,p}(x)$ is also nonnegative for all $x \in (-1, 1)$. By using interval computations, we can prove this statement for $p = 2, 3, 4, \dots, 10$; see (Šolín and Vejchodský, 2005; Šolín, 2005) for details.

6.3. HOW WE USE INTERVAL COMPUTATIONS

To prove the above result, we use a special basis in the linear space of all polynomials of p -th degree which vanish for $x = -1$ and $x = 1$: the basis of *Lobatto shape functions* (see, e.g., (Šolín, 2005))

$$l_k(x) = \frac{1}{\|L_{k-1}\|_{L^2}} \cdot \int_{-1}^x L_{k-1}(\xi) d\xi, \quad 2 \leq k,$$

where L_0, L_1, \dots are Legendre polynomials with $\|L_{k-1}\|_{L^2} = \sqrt{2/(2k-1)}$. In terms of these functions, the general solution to the above problem can be represented in the following form

$$u_{h,p}(x) = \int_{-1}^1 f_{h,p}(z) \cdot \Phi_p(x, z) dz, \quad (1)$$

where the *Green's function* $\Phi_p(x, z)$ has the form

$$\Phi_p(x, z) = \sum_{i=1}^{p-1} l_{i+1}(x) \cdot l_{i+1}(z).$$

For every $p > 1$, the function $\Phi_p(x, z)$ is a given bivariate polynomial defined in the square $(-1, 1)^2$. We want to use the expression (1) to prove that $u_{h,p}(x)$ is nonnegative for all $x \in (-1, 1)$. This is done in two steps:

1. First, we identify a subdomain Ω_p^+ of the interval $(-1, 1)$ where the function Φ_p is positive.
2. After that, we find a quadrature rule of the order of accuracy $2p$ (exact for all polynomials of degree less or equal to $2p$) with positive weights and points lying in Ω_p^+ .

The construction of the subdomains Ω_p^+ and the corresponding quadrature rules finishes the proof. The concrete subdomains Ω_p^+ along with the quadrature rules can be found in (Šolín and Vejchodský, 2005).

The interval computation technique is used to verify that the functions Φ_p are positive in the subdomains Ω_p^+ . Let us demonstrate the procedure on the quartic case, where we deal with the function $\Phi_4(x, z) = \sum_{i=1}^3 l_{i+1}(x) \cdot l_{i+1}(z)$. Since each polynomial $l_i(x)$ vanishes at $x = -1$ and at $x = 1$, this polynomial is proportional to $(x+1) \cdot (x-1) = x^2 - 1$, so the Green's function $\Phi_4(x, z)$ can be represented as $\Phi_4(x, z) = (x^2 - 1) \cdot (z^2 - 1) \cdot \Psi_4(x, z)$, where

$$\Psi_4(x, z) = \frac{3}{8} + \frac{5}{8} \cdot x \cdot z + \frac{7}{128} \cdot (5x^2 - 1) \cdot (5z^2 - 1). \quad (2)$$

The graph of the function $\Phi_4(x, z)$ is shown in Fig. 1.

To prove that the Green's function $\Phi_4(x, z) = (x^2 - 1) \cdot (z^2 - 1) \cdot \Psi_4(x, z)$ is nonnegative in the entire square $[-1, 1]^2$, it is sufficient to prove that $\Psi(x, z) \geq 0$ for all $(x, z) \in [-1, 1]^2$. We prove this nonnegativity by using straightforward interval computations; see, e.g., (Jaulin et al., 2001).

In interval computations, one deals with intervals instead of numbers, and standard unary and binary operations are extended from numbers to intervals in a natural way. For example, $[\underline{a}, \bar{a}] + [\underline{b}, \bar{b}] = [\underline{a} + \underline{b}, \bar{a} + \bar{b}]$, $[\underline{a}, \bar{a}] - [\underline{b}, \bar{b}] = [\underline{a} - \bar{b}, \bar{a} - \underline{b}]$, and so on. If we replace every operation with numbers by the corresponding operation of interval arithmetic, we get an enclosure for the range of the analyzed function on given intervals (Jaulin et al., 2001).

Let us use this technique to prove the nonnegativity of the function $\Psi_4(x, z)$ in the square $[-1, 1]^2$: Substituting a pair of intervals $X = [\underline{x}, \bar{x}]$ and $Z = [\underline{z}, \bar{z}]$ into the formula for $\Psi_4(x, z)$, we obtain an enclosure

$$[\underline{\Psi}_4, \bar{\Psi}_4] \supseteq \Psi_4(X, Z) = \{\Psi_4(x, z); x \in X, z \in Z\}.$$

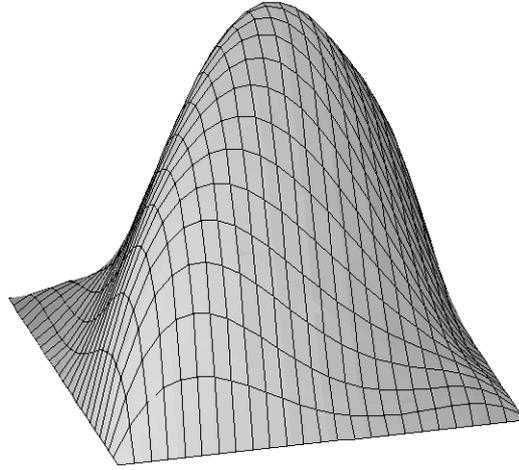


Figure 1. The function $\Phi_4(x, z)$.

Since the function $\Psi_4(x, z)$ is polynomial and it only contains rational coefficients, its evaluation for rational intervals can be done using exact integer arithmetic.

Step 1: Consider the intervals $X_1 = Z_1 = [-1, 1]$, and compute the enclosure $[\underline{\Psi}_4, \overline{\Psi}_4]$ for $\Psi_4(X_1, Z_1)$:

$$[\underline{\Psi}_4, \overline{\Psi}_4] = [-25/16, 95/32] \supseteq \Psi_4(X_1, Z_1).$$

If the left endpoint $\underline{\Psi}_4$ of the enclosure interval $[\underline{\Psi}_4, \overline{\Psi}_4]$ was nonnegative, then the proof would be finished. Since this is not the case, we refine the grid by halving both the intervals X_1 and Z_1 . We obtain four subdomains $[-1, 0] \times [-1, 0]$, $[-1, 0] \times [0, 1]$, $[0, 1] \times [-1, 0]$, and $[0, 1] \times [0, 1]$.

Step 2: Compute the enclosures for these subdomains:

- for $[-1, 0] \times [-1, 0]$, we get $[\underline{\Psi}_4, \overline{\Psi}_4] = [5/32, 15/8] \supseteq \Psi_4([-1, 0], [-1, 0])$;
- for $[-1, 0] \times [0, 1]$, we get $[\underline{\Psi}_4, \overline{\Psi}_4] = [-15/32, 5/4] \supseteq \Psi_4([-1, 0], [0, 1])$;
- for $[0, 1] \times [-1, 0]$, we get $[\underline{\Psi}_4, \overline{\Psi}_4] = [-15/32, 5/4] \supseteq \Psi_4([0, 1], [-1, 0])$;
- for $[0, 1] \times [0, 1]$, we get $[\underline{\Psi}_4, \overline{\Psi}_4] = [5/32, 15/8] \supseteq \Psi_4([0, 1], [0, 1])$.

This proves that the function Ψ_4 (and hence also Φ_4) is nonnegative in the subdomains $[-1, 0] \times [-1, 0]$ and $[0, 1] \times [0, 1]$. As for the remaining subdomains $[-1, 0] \times [0, 1]$ and $[0, 1] \times [-1, 0]$, we divide each of them into four equal subdomains, compute the enclosure for each new subdomain, etc.

After five iterations of this procedure, we get a partition of $[-1, 1]^2$ for which the left endpoints of the enclosures are nonnegative. So we have proved that Ψ_4 (and hence also Φ_4) is nonnegative in $[-1, 1]^2$.

The Java programs and output files with details on the computations for $p = 4, 5, \dots, 10$ can be viewed on the web page <http://www.math.utep.edu/Faculty/solin/intcomp>

6.4. NEW CHALLENGES

Can we extend the above one-dimensional result to a multi-dimensional case? The following example shows that the assumption of nonnegativity of the polynomial L^2 -projection of the right-hand side f will no longer be sufficient.

To illustrate this, let us consider a triangular domain Ω given by the vertices $[-1, -1]$, $[1, -1]$, $[-1, 1]$, and the stationary heat transfer equation $-\Delta\theta = f$ in Ω equipped with zero Dirichlet boundary conditions $\theta(\mathbf{x}) = 0$ for all $\mathbf{x} \in \partial\Omega$. The heat sources f are chosen to be a nonnegative cubic polynomial $f(x_1, x_2) = 1000 \cdot (x_1 + 1)^3$. In this case the exact solution θ is nonnegative in the domain Ω due to the classical (continuous) maximum principle for the Poisson equation.

The problem is discretized using a one-element mesh $K = \Omega$ with the polynomial degree $p(K) = 10$. It is shown in Fig. 2 that the approximate temperature $\theta_{h,p}$ is negative, i.e., nonphysical, near the right corner of Ω .

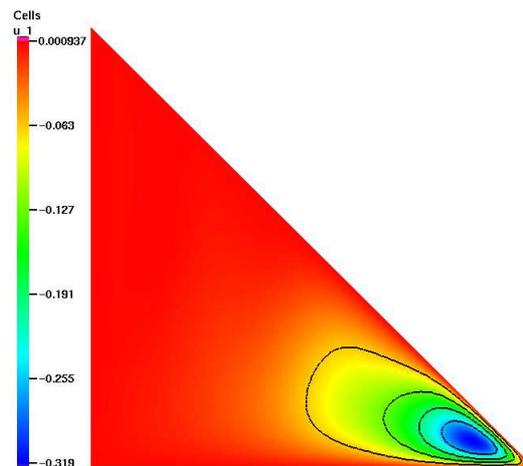


Figure 2. Nonphysical finite element solution of stationary heat transfer equation with zero boundary conditions and positive heat sources.

The formulation of conditions on the data and/or triangulation, which would guarantee the nonnegativity of the approximate solution, are an open problem. So far, we have found only partial conditions. Once these conditions are found, we will need to use interval computation techniques to prove the desired nonnegativity.

Acknowledgements

This work has been supported in part by NASA under cooperative agreement NCC5-209, NSF grants EAR-0225670 and DMS-0532645, Army Research Lab grant DATM-05-02-C-0046, Star Award of the University of Texas System, Texas Department of Transportation grant No. 0-5453, and the Grant Agency of the Czech Republic grant No. 102/05/0629.

References

- Ainsworth, M., and B. Senior, *Aspects of an hp-Adaptive Finite Element Method: Adaptive Strategy, Conforming Approximation and Efficient Solvers*, University of Leicester, England, U.K., Department of Mathematics and Computer Science, Technical Report 1997/2, 1997.
- Akpan, U. O., T. S. Koko, I. R. Orisamolu, and B. K. Gallant, Practical fuzzy finite element analysis of structures. *Finite Elem. Anal. Des.*, 38:93–111, 2001.
- Babuška, I., M. Griebel and J. Pitkäranta, The Problem of Selecting the Shape Functions for p -Type Elements, *Int. J. Numer. Meth. Engrg.* 28:1891–1908, 1999.
- Babuška, I., and W. Gui, The h , p , and hp -Versions of the Finite Element Method in One Dimension. Parts I–III, *Numer. Math.*, 49:577–683, 1986.
- Babuška, I., and T. J. Oden, Verification and Validation in Computational Engineering and Science: Basic Concepts, *Computer Methods in Applied Mechanics and Engineering*, 193:4057–4066, 2004.
- Belytschko, T., N. Moës, S. Usui, and C. Parimi, Arbitrary Discontinuities in Finite Elements, *International Journal of Numerical Methods in Engineering*, 50(4):993–1013, 2001.
- Ben-Haim, Y. and I. Elishakoff, *Convex Models of Uncertainty in Applied Mechanics*. Elsevier Science, Amsterdam, 1990.
- Berleant, D., Automatically verified reasoning with both intervals and probability density functions. *Interval Computations* (2):48–70, 1993.
- Chen, H., and T. Belytschko, An Enriched Finite Element Method for Elastodynamic Crack Propagation, *International Journal of Numerical Methods in Engineering*, 58(12):1873–1905, 2003.
- Chen, S. H., H. D. Lian, and X. W. Yang, Interval static displacement analysis for structures with interval parameters. *Int. J. Numer. Methods Engrg.* 53:393–407, 2002.
- Chessa, J., and T. Belytschko, An Extended Finite Element Method for Two-Phase Fluids, *J. Appl. Mech.*, 70(1):10–17, 2003.
- Chessa, J., and T. Belytschko, A Local Space-Time Discontinuous Finite Element Method, *Computer Methods in Applied Mechanics and Engineering*, to appear.
- Chessa, J., P. Smolinski, and T. Belytschko, The Extended Finite Element Method (X-FEM) for Solidification Problems, *Int. J. Numer. Methods Engrg.*, 53:1959–1977, 2002.
- Chessa, J., H. Wang, and T. Belytschko, On the Construction of Blending Elements for Local Partition of Unity Enriched Finite Elements, *Int. J. Numer. Methods Engrg.*, 57(7):1015–1038, 2003.
- Ciarlet, P. G. Discrete Maximum Principle for Finite Difference Operators, *Aequationes Math.*, 4:338–352, 1970.
- Ciarlet, P. G., and P. A. Raviart, Maximum principle and uniform convergence for the finite element method, *Computer Methods in Applied Mechanics and Engineering*, 2:17–31, 1973.
- Corliss, G., C. Foley, and R. B. Kearfott, Formulation for reliable analysis of structural frames, In: Muhanna, R. L., and R. L. Mullen (eds.), *Proc. NSF Workshop on Reliable Engineering Computing*, Savannah, Georgia, 2004, <http://www.gtsav.gatech.edu/rec/recworkshop/index.html>
- Demkowicz, L., W. Rachowicz, P. Devloo, *A fully automatic hp-adaptivity*, The University of Texas at Austin, TICAM Report 01-28, 2001.
- Dempster, A. P., Upper and lower probabilities induced by a multi-valued mapping. *Ann. Mat. Stat.* 38:325–339, 1967.

- Dessombz, O., F. Thouverez, J.-P. Lainé, and L. Jézéquel, Analysis of mechanical systems using interval computations applied to finite elements methods. *J. Sound. Vib.*, 238(5):949–968, 2001.
- Elishakoff, I., and Y. Ren, The Bird's Eye View on Finite Element Method for Stochastic Structures, *Computer Methods in Applied Mechanics and Engineering*, 168:51–61, 1999.
- Ferson, S. and L. R. Ginzburg, Different methods are needed to propagate ignorance and variability. *Reliab. Engng. Syst. Saf.* 54:133–144, 1996.
- Ferson, S., V. Kreinovich, L. Ginzburg, D. S. Myers, and K. Sentz. *Constructing Probability Boxes and Dempster-Shafer structures*, Sandia National Laboratories, Technical Report SAND2002-4015, 2003.
- Ganzerli, S. and C. P. Pantelides, Load and resistance convex models for optimum design. *Struct. Optim.* 17:259–268, 1999.
- Haldar, A., and S. Mahadevan, *Reliability Assessment Using Stochastic Finite Element Analysis*. John Wiley & Sons, New York, 2000.
- Hansen, E., *Global Optimization Using Interval Analysis*. Marcel Dekker, Inc., New York, 1992.
- Höhn, W., and H. D. Mittelmann, Some Remarks on the Discrete Maximum Principle for Finite Elements of Higher-Order, *Computing*, 27:145–154, 1981.
- Jaulin, L., M. Kieffer, O. Didrit, E. Walter, *Applied Interval Analysis*, Springer Verlag, London, 2001.
- Karátson, J., and S. Korotov, Discrete maximum principles for finite element solutions of nonlinear elliptic problems with mixed boundary conditions, *Numer. Math.* 99:669–698, 2005.
- Karniadakis, G. E., and S. J. Sherwin, *Spectral/hp Element Methods for CFD*, Oxford University Press, Oxford, 1999.
- Kendall, D. G., Foundations of a theory of random sets. In: Harding, E., and D. Kendall (eds.): *Stochastic Geometry*. New York, pp. 322–376.
- Korotov, S., M. Krížek, and P. Neittaanmäki, Weakened acute type condition for tetrahedral triangulations and the discrete maximum principle, *Math. Comp.*, 70:107–119, 2000.
- Koyluoglu, U., S. Cakmak, N. Ahmet, and R. K. Soren, Interval algebra to deal with pattern loading and structural uncertainty. *J. Engrg. Mech.*, 121(11):1149–1157, 1995.
- Krížek, M., and L. Liu, On the maximum and comparison principles for a steady-state nonlinear heat conduction problem, *ZAMM Z. Angew. Math. Mech.*, 83:559–563, 2003.
- Liu, W. K., T. Belytschko, and A. Mani, Probabilistic Finite Elements for Nonlinear Structural Dynamics. *Computer Methods in Applied Mechanics and Engineering*, 56:61–81, 1986.
- Lodwick, W. A. and K. D. Jamison, Special issue: interface between fuzzy set theory and interval analysis. *Fuzzy Sets and Systems*, 135:1–3, 2002.
- McWilliam, S., Anti-optimisation of uncertain structures using interval analysis. *Comput. Struct.*, 79:421–430, 2000.
- Melchers, R. E., *Structural Reliability Analysis and Prediction*, 2nd Edition, John Wiley & Sons, West Sussex, England, 1999.
- Moës, N., J. Dolbow, and T. Belytschko. A Finite Element Method for Crack Growth Without Remeshing, *Int. J. Numer. Methods Engrg.*, 46:131–150, 1999.
- Möller, B., W. Graf, and M. Beer, Fuzzy structural analysis using level-optimization. *Comput. Mech.*, 26(6):547–565, 2000.
- Moore, R. E., *Interval Analysis*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1966.
- Moore, R. E.: 1979, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.
- Muhanna, R. L. and R. L. Mullen, Development of interval based methods for fuzziness in continuum mechanics. In: *Proc. ISUMA-NAFIPS'95*, 1995, pp. 23–45.
- Muhanna, R. L. and R. L. Mullen, Formulation of fuzzy finite element methods for mechanics problems. *Compu.-Aided Civ. Infrastruct. Engrg.*, 14:107–117, 1999.
- Muhanna, R. L. and R. L. Mullen, Uncertainty in mechanics problems-interval-based approach. *J. Engrg. Mech.*, 127(6):557–566, 2001.
- Muhanna, R. L., R. L. Mullen, and H. Zhang, Penalty-Based Solution for the Interval Finite-Element Methods, *ASCE, Engineering Mechanics*, 131(10):1102–1111, 2005.
- Mullen, R. L. and R. L. Muhanna, Structural analysis with fuzzy-based load uncertainty. In: *Proc. 7th ASCE EMD/STD Joint Spec. Conf. on Probabilistic Mech. and Struct. Reliability*. Mass., 1996, pp. 310–313.

- Mullen, R. L. and R. L. Muhanna, Bounds of structural response for all possible loadings. *J. Struct. Engrg., ASCE*, 125(1):98–106, 1999.
- Neumaier, A., *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge, 1990.
- Neumaier, A., and A. Pownuk, Linear systems with large uncertainties, with applications to truss structures, *Reliable Computing*, to appear.
- Pantelides, C. P. and S. Ganzerli, Comparison of fuzzy set and convex model theories in structural design. *Mech. Systems Signal Process.*, 15(3):499–511, 2001.
- Paszynski, M., J. Kurtz, L. Demkowicz, *Parallel, Fully Automatic hp-Adaptive 2D Finite Element Package*, The University of Texas at Austin, TICAM Report 04-07, 2004.
- Popova, E. D., M. Datcheva, R. Iankov, and T. Schanz, Mechanical models with interval parameters. In: Gürlebeck, G., L. Hempel, C. Könke, editors, *IKM2003: Digital Proceedings of 16th International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering*, Weimar, Germany, 2003.
- Rachowicz, W., D. Pardo, L. Demkowicz, *Fully Automatic hp-Adaptivity in Three Dimensions*, The University of Texas at Austin, ICES Report 04-22, 2004.
- Rao, S. S. and L. Berke, Analysis of uncertain structural systems using interval analysis. *AIAA J.* 35(4):727–735, 1997.
- Rao, S. S. and L. Chen, Numerical solution of fuzzy linear equations in engineering analysis. *Int. J. Numer. Meth. Engrg.*, 43:391–408, 1998.
- Rump, S. M., Solving algebraic problems with high accuracy. In: Kulisch, U., and W. Miranker (eds.): *A New Approach to Scientific Computation*. Academic Press, New York, 1983.
- Schüller, G. Computational Stochastic Mechanics – Recent Advances, *Computers and Structures*, 79:2225–2234, 2001.
- Sentz, K. and S. Ferson, *Combination of Evidence in Dempster-Shafer Theory*, Sandia National Laboratories, Technical Report SAND2002-0835, 2002.
- Sethian, J. A. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- Shafer, G., *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, New Jersey, 1976.
- Šolín, P. *Partial Differential Equations and the Finite Element Methods*, J. Wiley & Sons, Hoboken, New Jersey, 2005.
- Šolín, P., and L. Demkowicz, Goal-Oriented hp-Adaptivity for Elliptic Problems, *Computer Methods in Applied Mechanics and Engineering*, 193:449–468, 2004.
- Šolín, P., K. Segeth, and I. Doležel, *Higher-Order Finite Element Methods*, Chapman & Hall/CRC Press, Boca Raton, 2003.
- Šolín, P., and T. Vejchodský, *On the Discrete Maximum Principle for the hp-FEM*, University of Texas at El Paso, Department of Mathematical Science, Technical Report, February 2005, http://www.math.utep.edu/Faculty/solin/new_papers/dmp.pdf ; see also http://www.math.utep.edu/Faculty/solin/new_papers/dmp-coll.pdf.
- Šolín, P., T. Vejchodský, and R. Araiza, *Discrete Conservation of Nonnegativity or Elliptic Problems Solved by the hp-FEM*, University of Texas at El Paso, Department of Computer Science, Technical Report UTEP-CS-05-29, August 2005, <http://www.cs.utep.edu/vladik/2005/tr05-29.pdf>
- Stazi, F., E. Budyn, J. Chessa, and T. Belytschko, An Extended Finite Element Method With Higher-Order Elements for Crack Problems With Curvature, *Computational Mechanics*, 31(1–2):38–48, 2003.
- Sukumar, N., D. L. Chopp, N. Moës, and T. Belytschko, Modeling Holes and Inclusions by Level Sets in the Extended Finite Element Method, *Computer Methods in Applied Mechanics and Engineering*, 190(46–47):6183–6200, 2000.
- Walley, P., *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991.
- Zadeh, L. A., Fuzzy Sets as a Basis for a Theory of Possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

Bounding the Response of Mechanical Structures with Uncertainties in All the Parameters *

Evgenija Popova

Institute of Mathematics & Informatics, Bulgarian Academy of Sciences

Roumen Iankov

Institute of Mechanics, Bulgarian Academy of Sciences

Zdravko Bonev

UACEG, Sofia, Bulgaria

Abstract. The application of a general-purpose self-verified parametric iteration for bounding the response of mechanical systems involving rational dependencies between interval parameters is investigated. Based on the availability of self-validated parametric linear solver, a general framework of computer-assisted proof of global and local monotonicity properties is presented. By the discussed methodology and software tools some frame structures with uncertainties in cross-sectional properties, applied loadings, material properties, geometry and connections are analyzed. The results are compared to literature data produced by other methods and a comparison of different measures of overestimation is done.

Keywords: self-verified methods, structural frames, parameter-dependent interval linear systems

1. Introduction

Uncertainty quantification is an emerging discipline which is nowadays well recognized by SIAM and structural engineering community. One of the research directions in this field utilizes intervals for representing the uncertain quantities and interval-based methods for reliable bounding the model response under variations in the uncertain parameters.

Many mechanical problems, e.g. linear static problems, modelled by finite element method, can be described by systems of linear equations involving uncertain model parameters. When the uncertain parameters are introduced by bounded intervals, the problem can be transformed into an interval linear system which should be solved appropriately to bound the mechanical system response. This approach is usually called Interval Finite Element Method. Overview of recent developments in the area of uncertainty treatment using interval finite element methods and their applications in structural engineering mechanics can be found in (Muhanna et al., 2004), (Muhanna et al., 2005). Although known for a decade, a self-validated parametric iteration method (Rump, 1994) is not adopted (even for a comparison purpose) and has single mechanical applications (Dessombz et al., 2001), (Popova et al., 2003). Instead, a construction method, called Element-By-Element approach (Mullen and Muhanna, 1999), is developed which introduces extra variables

* This work was partially supported by the Bulgarian National Science Fund under grant No. MM1301/03.

and equations in order to eliminate the dependencies between interval parameters. The penalty and Lagrange multiplier methods are used to impose the necessary constraints for compatibility and equilibrium (Muhanna and Mullen, 2001), (Muhanna et al., 2005). Non-parametric interval fixed-point iteration is modified and used to solve the model parametric interval linear system. During this transformation of the original parametric system, self-verifying properties of the interval iteration are lost or delayed to the final phase of solving non-parametric interval linear system. Recently, accounting for the structure of input data in systems related to truss structures, by splitting the iteration into two parts, Neumaier and Pownuk (2005) achieved an advance in self-verified methods applied to truss structures. Assuming a particular structure of the dependencies their method removes the restriction of most self-validating methods for linear systems to have a strongly regular matrix.

Depending on what model is adopted and which model parameters are considered to be uncertain or how they are involved into the interval linear system to be solved, the latter can be classified into two types: parametric linear systems involving affine-linear dependencies between the parameters and parametric linear systems involving arbitrary nonlinear dependencies between the interval parameters. So far mainly problems involving affine-linear dependencies have been solved. In this work we come back to the *parametric* fixed-point iteration, initially introduced by S. Rump (1994), and first time apply it for bounding the response of structural engineering systems involving nonlinear dependencies between the model parameters. In (Popova, 2005) the inclusion method is combined with a simple interval arithmetic technique providing inner and outer bounds for the range of monotone rational functions. The arithmetic on proper and improper intervals (Gardeñes et al., 2001) is considered as an intermediate computational tool for eliminating the dependency problem in range computation and for obtaining inner estimations by outwardly rounded interval arithmetic. This methodology is implemented into a number of supporting software tools with result verification, developed in the environment of *Mathematica*, (Popova, 2005).

Combinatorial approach and the monotonicity approach have been favored by many authors in solving linear elastic problems involving particular uncertain parameters (Rao and Berke, 1997), (Ganzerli and Pantelides, 1999), (McWilliam, 2000), (Pownuk, 2000). A rigorous application of these approaches requires validation of their assumptions which are not generally valid. In Section 2.2 of this paper we present a general framework of computer-aided proof of global and local monotonicity properties of parametric solutions provided that a self-verified solver of parametric linear systems is available.

A recent work (Corliss et al., 2004), see also (Corliss and Foley, 2005), identified typical parameter uncertainties in finite element models of structural steel frames with partially constrained connections and by applying a sequence of interval-based methods the response of a simple one-bay steel frame to variations in cross-sectional properties, loading, material properties, and connections is bounded. Taking occasion of the appeal at the end of the presentation (Corliss and Foley, 2005) for other reliable methods solving parameter-dependent linear systems, in Section 3.1 this work we expand the structural analysis performed in (Corliss et al., 2004) by application of the self-verified parametric iteration, a rigorous hybrid monotonicity approach, and interval subdivision technique to the same problem and to a larger structural steel frame. The goal is to increase the awareness of the engineering community about the variety of interval-based methods with result verification that can be used in the analysis of mechanical structures involving uncertain parameters.

The paper is organized in two parts. Section 2 briefly describes the methodological and software tools that are used in the second part. Section 3 contains the analysis of structural frames.

2. Methodology and Software Tools

In this section we give a brief summary of the numerical methods and software tools that will be used in solving linear elastic mechanical problems with uncertainties in all the model parameters. The methods have general purpose and do not assume any particular structure of the input data.

Consider linear algebraic system

$$A(p) \cdot x = b(p), \quad (1a)$$

where $A(p)$ is an $n \times n$ matrix, $b(p)$ is an n -dimensional vector and $p = (p_1, \dots, p_k)^\top$ is a k -dimensional parameter vector. The elements of $A(p)$ and $b(p)$ are, in general, nonlinear functions of the parameters

$$a_{ij}(p) = a_{ij}(p_1, \dots, p_k), \quad (1b)$$

$$b_i(p) = b_i(p_1, \dots, p_k), \quad i, j = 1, \dots, n. \quad (1c)$$

The parameters are considered to be unknown or uncertain and varying within prescribed intervals

$$p \in [p] = ([p_1], \dots, [p_k])^\top. \quad (1d)$$

When the parameters vary within a box $[p] \in \mathbb{IR}^k$ the set of solutions, called parametric solution set is

$$\Sigma^p = \Sigma(A(p), b(p), [p]) := \{x \in \mathbb{R}^n \mid A(p) \cdot x = b(p) \text{ for some } p \in [p]\}. \quad (2)$$

In general, a solution set has very complicated structure, and does not need even to be convex. The parametric solution set Σ^p is bounded if $A(p)$ is nonsingular for every $p \in [p]$. For a nonempty bounded set $\Sigma \subseteq \mathbb{R}^n$, define interval hull $\square : P\mathbb{R}^n \rightarrow \mathbb{IR}^n$ by

$$\square \Sigma := [\inf \Sigma, \sup \Sigma] = \cap \{[x] \in \mathbb{IR}^n \mid \Sigma \subseteq [x]\}.$$

Since it is quite expensive to obtain Σ^p or $\square \Sigma^p$, the solution of interest is seeking an interval vector $[y] \in \mathbb{IR}^n$ such that $[y] \supseteq \square \Sigma^p \supseteq \Sigma^p$, and the goal is $[y]$ to be as narrow as possible.

Below we use the following notations. $\mathbb{R}^n, \mathbb{R}^{n \times m}$ denote the set of real vectors with n components and the set of real $n \times m$ matrices, respectively. By normal (*proper*) interval we mean a real compact interval $[a] = [a^-, a^+] := \{a \in \mathbb{R} \mid a^- \leq a \leq a^+\}$. By $\mathbb{IR}^n, \mathbb{IR}^{n \times m}$ we denote interval n -vectors and interval $n \times m$ matrices. The end-point functionals $(\cdot)^-, (\cdot)^+$, the mid-point function $\text{mid}(\cdot)$, where $\text{mid}([a^-, a^+]) := (a^- + a^+)/2$, and the diameter (width) function $\omega(\cdot)$, where $\omega([a^-, a^+]) := a^+ - a^-$, are applied to interval vectors and matrices componentwise. The absolute value of a matrix $A = (a_{ij})$ is denoted by $|A| = (|a_{ij}|)$; for $[a] \in \mathbb{IR}$, $|[a]| := \max\{|a| \mid a \in [a]\}$. For two matrices of the same size matrix (vector) inequalities $A \leq B$ and the interval subset relations $[A] \subseteq [B]$ are understood componentwise. $A < B$ if $A \leq B$ and $A \neq B$, analogously $[A] \subset [B]$ if $[A] \subseteq [B]$ and $[A] \neq [B]$. The above matrix notations apply to vectors, considered as one-column matrices, as well. $\varrho(A)$ is

the spectral radius of a matrix A , I denotes the identity matrix. For interval quantities $[A], [B]$, operations between them are always interval operations. The result is the smallest interval quantity containing the corresponding result when using power set operations. For example,

$$[A] \in \mathbb{IR}^{n \times n}, [b] \in \mathbb{IR}^n : [A] \cdot [b] := \cap \{[c] \in \mathbb{IR}^n \mid \forall a \in [A], \forall b \in [b] : a \cdot b \in [c]\}.$$

We assume the reader is familiar with conventional interval arithmetic, cf. (Moore, 1979), (Neumaier, 1990).

2.1. INCLUSION THEOREMS

The inclusion theorems for the solution set of a parametric linear system given here present a direct consequence from the inclusion theory for nonparametric problems developed by S. Rump and discussed in many works, cf. (Rump, 1986; Rump, 1990; Rump, 1994). The basic idea of combining the Krawczyk-operator and the existence test by Moore was further elaborated by S. Rump (1986) who proposed several improvements leading to inclusion theorems for the solution of nonparametric interval linear systems $[A] \cdot x = [b]$. Computing verified inclusions for the solution set of an interval linear system with data dependencies was first considered by C. Jansson (1991). He treated systems with symmetric and skew-symmetric matrices, as well as dependencies in the right hand side, by modifying the general nonparametric inclusion theorem to account for the dependencies in the system. In (Rump, 1994, Theorem 4.8) S. Rump gives a straightforward generalization to affine-linear dependencies in the matrix and the right hand side. The affine-linear dependencies between the parameters in $A(p), b(p)$ allow an explicit representation of the ranges of the residual vector $z(p) := R \cdot (b(p) - A(p) \cdot \tilde{x})$ and the iteration matrix $C(p) := I - R \cdot A(p)$ by interval expressions, as it is stated by the following theorem.

Theorem 2.1. Consider parametric linear system (1a) where $A(p), b(p)$ are defined by

$$a_{ij}(p) := a_{ij}^{(0)} + \sum_{\nu=1}^k p_{\nu} a_{ij}^{(\nu)}, \quad b_i(p) := b_i^{(0)} + \sum_{\nu=1}^k p_{\nu} b_i^{(\nu)}, \quad i, j = 1, \dots, n.$$

Let $R \in \mathbb{R}^{n \times n}$, $[y] \in \mathbb{IR}^n$, $\tilde{x} \in \mathbb{R}^n$ be given and define $[z] \in \mathbb{IR}^n$, $[C] \in \mathbb{IR}^{n \times n}$ by

$$[z] := R \cdot (b^{(0)} - A^{(0)} \tilde{x}) + \sum_{\nu=1}^k [p_{\nu}] (R \cdot b^{(\nu)} - R \cdot A^{(\nu)} \cdot \tilde{x}),$$

$$[C] := I - R \cdot A^{(0)} - \sum_{\nu=1}^k [p_{\nu}] (R \cdot A^{(\nu)}),$$

where $A^{(0)} := (a_{ij}^{(0)})$, $\dots, A^{(k)} := (a_{ij}^{(k)}) \in \mathbb{R}^{n \times n}$, $b^{(0)} := (b_i^{(0)})$, $\dots, b^{(k)} := (b_i^{(k)}) \in \mathbb{R}^n$. Define $[v] \in \mathbb{IR}^n$ by means of the following Einzelschrittverfahren

$$1 \leq i \leq n : [v_i] := \{[z] + [C] \cdot [u]\}_i, \quad u := (v_1, \dots, v_{i-1}, y_i, \dots, y_n)^{\top}.$$

If $[v] \subsetneq [y]$, then R and every matrix $A(p), p \in [p]$ are regular, and for every $p \in [p]$ the unique solution $\hat{x} = A^{-1}(p)b(p)$ of (1a) satisfies $\hat{x} \in \tilde{x} + [v]$.

The above theorem generalizes Theorem 4.8 from (Rump, 1994) by requiring computation of the range of $C(p)$ instead of using an interval extension $C([p])$, cf. (Popova, 2004c). Although a sharp enclosure of the iteration matrix is used also by other authors (Dessombz et al., 2001; Muhanna et al., 2005), the necessity of this improvement is not well justified therein. The generalization of Theorem 4.8 from (Rump, 1994) is proven theoretically and demonstrated by several numerical examples in (Popova, 2004b; Popova, 2004c). Indeed, for a class of so-called column-dependent parametric matrices (Popova, 2004b), the following relation holds

$$[Cp] := \square\{C(p) \mid p \in [p]\} \subset C([p]) =: [C],$$

which implies $\| [Cp] \| < \| [C] \|$. If in addition, $\| [Cp] \| + \| [C] \|$ is irreducible, from the theory of nonnegative matrices it follows that $\rho(\| [Cp] \|) < \rho(\| [C] \|)$. Thus the range enclosure of $C(p)$ will provide convergence of the iteration method for $\rho(\| [Cp] \|) < 1$, while a worse enclosure (e.g. $C([p])$) may not for some column-dependent parametric matrices and some interval domains for the parameters. Examples demonstrating the expanded scope of application of the generalized Theorem 2.1 can be found in (Popova, 2004b; Popova, 2004c; Popova and Krämer, 2004).

In case of arbitrary nonlinear dependencies between the parameters of a linear system we can give only a general formulation of the inclusion theorem, as bellow.

Theorem 2.2. Consider parametric linear system defined by (1a–1d). Let $R \in \mathbb{R}^{n \times n}$, $[y] \in \mathbb{IR}^n$, $\tilde{x} \in \mathbb{R}^n$ be given and define $[z] \in \mathbb{IR}^n$, $[C] \in \mathbb{IR}^{n \times n}$ by

$$\begin{aligned} [z] &:= \square\{R(b(p) - A(p)\tilde{x}) \mid p \in [p]\}, \\ [C] &:= \square\{I - R \cdot A(p) \mid p \in [p]\}. \end{aligned}$$

Define $[v] \in \mathbb{IR}^n$ by means of the following Einzelschrittverfahren

$$1 \leq i \leq n : [v_i] := \{[z] + [C] \cdot [u]\}_i, \quad u := (v_1, \dots, v_{i-1}, y_i, \dots, y_n)^\top.$$

If $[v] \subsetneq [y]$, then R and every matrix $A(p)$ with $p \in [p]$ are regular, and for every $p \in [p]$ the unique solution $\hat{x} = A^{-1}(p)b(p)$ of (1a–1d) satisfies $\hat{x} \in \tilde{x} + [v]$.

In case of arbitrary nonlinear dependencies between the uncertain parameters in a system, computing $[z]$ and $[C]$ in Theorem 2.2 requires sharp range enclosure for nonlinear functions. This is a key problem in interval analysis and there exists a variety of methods and techniques devoted to this problem. The quality of the range enclosure for $z(p) := R \cdot (b(p) - A(p) \cdot \tilde{x})$ will determine the sharpness of the parametric solution set enclosure. The verification iteration based on Theorem 2.2 will be convergent if the interval matrix $\square\{R \cdot A(p) \mid p \in [p]\}$ is regular which we call strong regularity of the parametric matrix $A(p)$ in the domain $[p]$, following the term initially introduced in (Neumaier, 1990). Since the left preconditioning introduces an affine transformation on the columns of $A(p)$, only systems with column-dependent parametric matrices may benefit from a sharper enclosure of $C(p) = I - R \cdot A(p)$.

In (Popova, 2005) the above inclusion theorem is combined with a simple interval arithmetic technique providing inner and outer bounds for the range of monotone rational functions. The arithmetic of generalised (proper and improper) intervals is considered as an intermediate computational tool

for eliminating the dependency problem in range computation and for obtaining inner estimations by outwardly rounded interval arithmetic (Gard e es et al., 2001). A detailed presentation of this technique and the corresponding algorithm with result verification, which solves linear systems whose input data are rational functions of interval parameters, can be found in (Popova, 2005). This methodology, rigorously implemented in software tools presented in Section 2.4, will be used in Section 3 for solving linear systems obtained by FE modelling of mechanical structures with uncertainties in all the parameters determining the structure behavior.

The above theorems define how to compute an outer enclosure of the solution set of an interval linear system, i.e. an interval vector which is verified to contain the exact solution set hull, respectively the true solution set of the system. However, it is important to know the quality of the computed enclosure, in other words: how much such an enclosure overestimates the exact hull of the solution set. The amount of overestimation can be approximated by an inner inclusion of the solution set hull which is a componentwise inner estimation of the solution set (Neumaier, 1987; Rump, 1990).

Definition 2.1. An interval vector $[x] \in \mathbb{IR}^n$ is called componentwise inner approximation for some set $\Sigma \in \mathbb{R}^n$ if

$$\inf_{\sigma \in \Sigma} \sigma_i \leq x_i^- \quad \text{and} \quad x_i^+ \leq \sup_{\sigma \in \Sigma} \sigma_i, \quad \text{for every } 1 \leq i \leq n.$$

The interval vector $[x]$ from the above definition is an inner inclusion of the solution set hull and should be distinguished from an inner inclusion of the solution set, that is $[x] \subseteq [\inf(\Sigma), \sup(\Sigma)]$ but $[x] \not\subseteq \Sigma$.

Basing on ideas developed in (Neumaier, 1987), a cheap method for computing rigorous inner inclusion of the solution set hull is proposed in (Rump, 1990). The next theorem establishes how to compute the componentwise inner estimation of the parametric solution set.

Theorem 2.3. Let $A(p) \cdot x = b(p)$, where $A(p) \in \mathbb{R}^{n \times n}$, $b(p) \in \mathbb{R}^n$, $p \in [p] \in \mathbb{IR}^k$, and $R \in \mathbb{R}^{n \times n}$, $\tilde{x} \in \mathbb{R}^n$, $[y] \in \mathbb{IR}^n$ be given. Define

$$\begin{aligned} [z] &:= \square \{R \cdot (b(p) - A(p) \cdot \tilde{x}) \mid p \in [p]\}, \\ [\Delta] &:= [C] \cdot [y], \quad \text{where } [C] := \square \{I - R \cdot A(p) \mid p \in [p]\}. \end{aligned}$$

Let the solution set $\Sigma^p = \Sigma(A(p), b(p), [p])$ be defined as in (2) and assume

$$[z] + [\Delta] \subsetneq [y].$$

Then

$$[\tilde{x} + [z]^- + [\Delta]^+, \tilde{x} + [z]^+ + [\Delta]^-] \subseteq \square \Sigma^p \subseteq \tilde{x} + [z] + [\Delta]$$

or, in coordinate notations, for all $i = 1, \dots, n$ there exists $x^-, x^+ \in \Sigma^p$ with

$$\begin{aligned} \tilde{x}_i + [z_i]^- + [\Delta_i]^- &\leq x_i^- \leq \tilde{x}_i + [z_i]^- + [\Delta_i]^+ & \text{and} \\ \tilde{x}_i + [z_i]^+ + [\Delta_i]^- &\leq x_i^+ \leq \tilde{x}_i + [z_i]^+ + [\Delta_i]^+. \end{aligned}$$

In order to have a guaranteed inner inclusion all the computations should be done in computer arithmetic with directed roundings, cf. (Popova, 2005).

The method from Theorem 2.3 has its limits. When widening the intervals for the parameters, respectively the interval components of the linear system, the inner inclusion becomes smaller and smaller, and finally vanishes. The latter means that no quantitative measure for the quality of the outer enclosure can be given. For wide parameter intervals empty inner inclusion usually means bad outer enclosure and, when further widening the input intervals, the outer solution enclosure will fail at a certain point. Numerical examples demonstrating this effect can be found in (Popova and Krämer, 2004). The same result of empty inner inclusion intervals can be obtained also for very tight parameter intervals due to the rounding errors in computing inner approximations. A necessary and sufficient condition for non-empty inner inclusions is provided by the relation $\omega([\Delta_i]) \leq \omega([z_i])$, where the notations are as in Theorem 2.3, $[\Delta_i]$ is computed with outward rounding and $[z_i]$ is computed with inward rounding.

When somehow we have sharpen the outer solution enclosure $\square \Sigma^p \subseteq [\hat{v}] \subseteq [v] = \tilde{x} + [z] + [\Delta]$, then the improved outer estimation $[\hat{v}]$ can replace $[v]$ in Theorem 2.3 to get an improved inner estimation of $\square \Sigma^p$. Numerical example demonstrating this property can be found in (Popova, 2001).

2.2. RIGOROUS MONOTONICITY APPROACH

For many mechanical systems the exact bounds of the system response can be obtained by the so-called combinatorial approach. The combinatorial solution is computed as a convex hull of the solutions to all point linear systems corresponding to an exhaustive combination of the bounds of the interval parameters. Combinatorial hull is a quality of particular parametric solution sets which is not valid in general. The combinatorial approach gives the exact solution set hull in exact arithmetic in the special case when the parametric solution is monotone with respect to all the parameters. If the combinatorial hull property is not proven theoretically (as by Neumaier and Pownuk (2005)) or numerically (as below), any other non-rigorous application of combinatorial or monotonicity approach would result in an interval box underestimating the true parametric solution set. This is the reason by which combinatorial and monotonicity approaches are usually referred as methods giving inner inclusion of the solution set hull (Muhanna et al., 2005; Neumaier and Pownuk, 2005).

In this section we briefly sketch a rigorous application of the combinatorial/monotonicity approach within a general framework for solving parametric linear systems. The rigorousness is provided by computer-assisted numerical proofs of global and local monotonicity properties of the parametric solution. Since an essential ingredient of this approach is a self-verified solver for parametric linear systems, we call this approach a rigorous hybrid monotonicity approach (Popova, 2004a).

The general framework of the rigorous hybrid monotonicity approach consists of three basic components:

1. self-verified solver for parametric linear systems;
2. computer-assisted proof of global and local monotonicity properties of a parametric solution;
3. guaranteed solution enclosure for point linear systems.

Provided that we have a self-verified solver for parametric linear systems, we can verify the global and local monotonicity properties of the parametric solution $x(p) = A(p)^{-1} \cdot b(p)$. Below we use the following notations. For $[a] = [a^-, a^+] \in \mathbb{IR}$, define $\text{sign}([a]) = \{1 \text{ if } a^- \geq 0, -1 \text{ if } a^+ \leq 0, 0 \text{ if } a^- a^+ < 0\}$. For a set of indices $\mathcal{I} = \{i_1, \dots, i_n\}$, the vector $(x_{i_1}, \dots, x_{i_n})^\top$ will be denoted by $x_{\mathcal{I}}$ and $[x_{\mathcal{I}}] = [x_{\mathcal{I}}^-, x_{\mathcal{I}}^+]$ where $x_{\mathcal{I}}^- = (x_{i_1}^-, \dots, x_{i_n}^-)^\top$, $x_{\mathcal{I}}^+ = (x_{i_1}^+, \dots, x_{i_n}^+)^\top$.

The global monotonicity properties are verifiable by solving k parametric linear systems in the global domain $[p] \in \mathbb{IR}^k$

$$A(p) \frac{\partial x}{\partial p_\nu} = \frac{\partial b(p)}{\partial p_\nu} - \frac{\partial A(p)}{\partial p_\nu} \cdot [x^*], \quad \nu = 1, \dots, k, \quad (3)$$

where $[x^*] \supseteq \Sigma^p$ is an initial enclosure of the parametric solution set. Let us suppose that for fixed i , $1 \leq i \leq n$ there exist index sets

$$L_+ = \{\nu \mid \text{sign} \left[\frac{\partial x_i}{\partial p_\nu} \right] = 1\}, \quad L_- = \{\nu \mid \text{sign} \left[\frac{\partial x_i}{\partial p_\nu} \right] = -1\}.$$

If $L_- \cup L_+ = \{1, \dots, k\}$, then

$$[\inf \Sigma_i^p, \sup \Sigma_i^p] = \{[A^{-1}(p_{L_+}^-, p_{L_-}^+) \cdot b(p_{L_+}^-, p_{L_-}^+)]_i, [A^{-1}(p_{L_+}^+, p_{L_-}^-) \cdot b(p_{L_+}^+, p_{L_-}^-)]_i\}.$$

Monotonicity can also be used even when some solution components are not globally monotonic with respect to some parameters. Suppose that for some i , $1 \leq i \leq n$, there exist index sets

$$L_+ = \{\nu \mid \text{sign} \left[\frac{\partial x_i}{\partial p_\nu} \right] = 1\}, \quad L_- = \{\nu \mid \text{sign} \left[\frac{\partial x_i}{\partial p_\nu} \right] = -1\}, \quad L_0 = \{\nu \mid \text{sign} \left[\frac{\partial x_i}{\partial p_\nu} \right] = 0\},$$

such that $L_0 \neq \{1, \dots, k\}$ and $L_0 \neq \emptyset$. Consider two new parametric linear systems

$$A^-(p_{L_0}) \cdot y = b^-(p_{L_0}) \quad (4)$$

$$A^+(p_{L_0}) \cdot z = b^+(p_{L_0}), \quad (5)$$

wherein

$$\begin{aligned} a_{ij}^-(p_{L_0}) &:= a_{ij}(p_{L_+}^-, p_{L_-}^+, p_{L_0}), & b_i^-(p_{L_0}) &:= b_i(p_{L_+}^-, p_{L_-}^+, p_{L_0}) \\ a_{ij}^+(p_{L_0}) &:= a_{ij}(p_{L_+}^+, p_{L_-}^-, p_{L_0}) & b_i^+(p_{L_0}) &:= b_i(p_{L_+}^+, p_{L_-}^-, p_{L_0}) \end{aligned}$$

for $i, j = 1, \dots, n$ and $p_{L_0} \in [p_{L_0}]$.

Let $[y^*] \supseteq \Sigma(A^-(p_{L_0}), b^-(p_{L_0}), [p_{L_0}])$ and $[z^*] \supseteq \Sigma(A^+(p_{L_0}), b^+(p_{L_0}), [p_{L_0}])$. In general,

$$[\inf \Sigma_i^p, \sup \Sigma_i^p] \subseteq [y_i^*] \cup [z_i^*].$$

However, we may prove some monotonicity properties of the parametric solutions to (4), (5) by solving the corresponding parametric derivative systems in a considerably reduced interval domain $[p_{L_0}]$.

$$\begin{aligned} A^-(p_{L_0}) \frac{\partial y}{\partial p_\nu} &= \frac{\partial b^-(p_{L_0})}{\partial p_\nu} - \frac{\partial A^-(p_{L_0})}{\partial p_\nu} \cdot [y^*] \\ A^+(p_{L_0}) \frac{\partial z}{\partial p_\nu} &= \frac{\partial b^+(p_{L_0})}{\partial p_\nu} - \frac{\partial A^+(p_{L_0})}{\partial p_\nu} \cdot [z^*], \end{aligned}$$

for all $\nu \in L_0$, where $[y^*], [z^*]$ are initial enclosures of the solution sets of (4), resp. (5), or an initial enclosure of $\Sigma(A(p), b(p), [p])$.

This way, a computer-aided proof of global and local monotonicity properties of the parametric solution can be performed by self-validated solving of parametric linear systems. The success of the numerical proof depends very much on the quality of the parametric solution enclosure and on the quality of the initial enclosure (Popova, 2004a). Some specific issues related to this approach will be discussed in a separate work.

2.3. MEASURES OF OVERESTIMATION

The quality of a solution enclosure is measured by estimating how much an outer solution enclosure overestimates the true parametric solution set or an inner inclusion of the solution set hull (since the true hull is usually not known). A discussion about different methods used for obtaining inner hull estimations can be found in (Neumaier and Pownuk, 2005). The inclusion method, presented in Section 2.1, is equipped with an easy computable guaranteed inner estimation of the solution set hull. In this work we shall measure the overestimation of the outer solution enclosure with respect to a combinatorial solution and to the guaranteed inner estimation of the solution hull provided by the method.

Provided that we have computed the exact solution set hull or some inner estimation(s) of the hull, the amount of overestimation should be quantified. The endeavor of providing sharper solution enclosures has resulted in utilization of different measures of overestimation. In the next section we shall use and compare the quality quantifications provided by the following measures of overestimation.

For two intervals $[a], [b] \in \mathbb{IR}$ such that $[a] \subseteq [b]$, the standard measure of overestimation that is usually applied is the percentage by which $[b]$ overestimates the interval $[a]$, defined as $\mathcal{O}_\omega : \mathbb{IR} \times \mathbb{IR} \rightarrow \mathbb{R}_+$

$$\mathcal{O}_\omega([a], [b]) := 100(1 - \omega([a])/\omega([b])).$$

Distance-based measures of overestimation are sometimes used in the engineering literature, e.g. (Muhanna et al., 2005). $\mathcal{O}_d : \mathbb{IR} \times \mathbb{IR} \rightarrow \mathbb{R} \times \mathbb{R}$ is defined by

$$\mathcal{O}_d([a], [b]) := 100(1 - a^-/b^-, 1 - a^+/b^+).$$

Since we will compare part of our results to those obtained in (Corliss et al., 2004), we will need the measure of overestimation used therein. For $[a], [b] \in \mathbb{IR}$, $[a] \subseteq [b]$ and $c \in \mathbb{R}$, $c \in [a]$, define $\mathcal{O}_c : \mathbb{IR} \times \mathbb{IR} \times \mathbb{R} \rightarrow \mathbb{R}_+$ by

$$\mathcal{O}_c([a], [b], c) := 100(b^- - a^- + a^+ - b^+)/c.$$

Overestimation measures are applied to interval vectors componentwise.

The presented parametric fixed-point method provides a guaranteed inner estimation $[v]$ of the solution hull $[h]$ at no additional cost. Since the computation of $[v]$ uses the computed outer enclosure $[u]$ in a “symmetric” way, it can be expected that $[v]$ is almost symmetric to $[u]$ with respect to the exact solution set hull. That is why, $\frac{1}{2}\mathcal{O}_\omega([v], [u]) \approx \mathcal{O}_\omega([h], [u])$ will be used for measuring the quality of a solution enclosure.

2.4. SOFTWARE TOOLS

Interval methods discussed in this paper and elsewhere are implemented in the environment of *Mathematica* (Wolfram, 1999). The *Mathematica* package `IntervalComputations` ‘`LinearSystems`’ contains a collection of functions which compute guaranteed inclusions for the solution set of an interval linear system (Popova, 2004a). The particular solvers differ upon the type of the linear system to be solved and the implemented solution method. Except for a C-XSC module solving parametric linear systems with affine-linear dependencies (Popova and Krämer, 2004), the above *Mathematica* package is the only by now public software for solving parameter-dependent interval linear systems.

`ParametricNSolve[Ap, bp, tr]` is the function which solves linear systems involving affine-linear dependencies between interval parameters. The function is based on entirely numerical computations and therefore it is fast. The function is updated to handle sparse arrays as input data.

`ParametricSSolve[Ap, bp, tr]` computes a guaranteed enclosure of the solution set to a parametric linear system $\mathbf{A}p \cdot \mathbf{x} = \mathbf{b}p$ involving rational dependencies by the algorithm presented in (Popova, 2005). The parameters and their interval values are specified by a list `tr` of transformation rules¹. All iterative solvers can take two optional arguments affecting the computational process, respectively the output of the function. `InnerEstimation` is an option which when set to `True` specifies the computing of component-wise inner approximation of the solution set in addition to the outer enclosure. The option is set to `False` by default. Even set to `True`, the option is active only if the *Mathematica* package `IntervalComputations` ‘`GeneralisedIntervals`’ is available. `Refinement` is an option which set to `True` implies an iterative refinement procedure applied to the computed outer solution enclosure. The default setting is `False`.

Due to a previous improvement of the inclusion theory, new functions generating guaranteed inclusions of the solutions to nonsquare over-/underdetermined (parametric) linear systems are developed. Several functions supporting the hybrid monotonicity approach and a subdivision strategy are also part of the package.

Approaching to parametric linear systems with rational dependencies, the integration of symbolic-algebraic and self-validating numerical computations based on interval arithmetic is found to be a fruitful synergism. The power of *Mathematica* to support rigorous exact and/or variable precision interval computations, the functionality of a generalized interval arithmetic package and the tools provided by the other interval packages, make a suitable environment for exploration and solving parametric problems with interval uncertainties.

In order to provide a broad access to solvers for parametric interval linear systems a web interface for the available *Mathematica* software is developed which can be found at

<http://cose.math.bas.bg/webComputing/>

Accessing the webComputing pages users enter or upload data, choose between different options, and submit data to build up a sequence of results in a numeric, symbolic, graphics or combined form. The end-users do not need to buy, install, and maintain software; they do not need to develop user software or to learn different software applications training time being considerably reduced. They can be certain that use the most recent version. The technical professionals and

¹ *Mathematica* transformation rules have the form `name -> value`.

interval researchers can easily explore newly developed methods; compare the efficiency of different methods and software tools; teach interval methods involving students in an active exploration by doing. Since algebraic computations are time consuming and web *Mathematica* applications have a fixed time limit for using the *Mathematica* kernel, the nonlinear parametric web solver is suitable only for small size problems, while large problems involving affine-linear dependencies can be solved remotely. The parametric web solvers allow uploading data files from the client machine onto the server. For a parametric system, 3 data files (containing the matrix, the right-hand side vector and the rules for the parameters) are required. Present restriction to the maximum size of a data file is 4MB. Matrix/vector data in a file presently should be specified by *Mathematica* lists, or as sparse arrays (Wolfram, 1999). Future enhancement of the solvers include different data formats, downloading the generated results on the client machine and combining/reusing the results from different pages.

3. Numerical Examples

3.1. ONE-BAY STEEL FRAME

In this section we consider a simple one-bay structural steel frame, shown in Figure 1, that was initially considered and analyzed by Corliss et al. (2004). In their work the authors survey typical uncertainties for the parameters characterizing the structural behavior and apply the Muhanna-Mullen Element-by-Element approach (Muhanna and Mullen, 2001), interval subdistributivity properties, scaling, and constraint propagation in order to demonstrate the feasibility of interval techniques for bounding structural responses in the presence of interval parameters. Here the analysis of Corliss et al. is expanded by the methods presented in Section 2.

Figure 1. One-bay structural steel frame (after Corliss et al. (2004)).

In order to compare the results generated by the different methods, we strictly follow the structure system and the uncertainties for the parameters considered in (Corliss et al., 2004). Following the usual practice, the authors have assembled the following linear system corresponding to the portal structure in Figure 1.

$$\begin{pmatrix}
 \frac{A_b E_b}{L_b} + \frac{12 E_c I_c}{L_c^3} & 0 & \frac{6 E_c I_c}{L_c^2} & 0 & 0 \\
 0 & \frac{A_c E_c}{L_c} + \frac{12 E_b I_b}{L_b^3} & 0 & \frac{6 E_b I_b}{L_b^2} & \frac{6 E_b I_b}{L_b^2} \\
 \frac{6 E_c I_c}{L_c^2} & 0 & \alpha + \frac{4 E_c I_c}{L_c} & -\alpha & 0 \\
 0 & \frac{6 E_b I_b}{L_b^2} & -\alpha & \alpha + \frac{4 E_b I_b}{L_b} & \frac{2 E_b I_b}{L_b} \\
 0 & \frac{6 E_b I_b}{L_b^2} & 0 & \frac{2 E_b I_b}{L_b} & \alpha + \frac{4 E_c I_c}{L_c} \\
 -\frac{A_b E_b}{L_b} & 0 & 0 & 0 & 0 \\
 0 & -\frac{12 E_b I_b}{L_b^3} & 0 & -\frac{6 E_b I_b}{L_b^2} & -\frac{6 E_b I_b}{L_b^2} \\
 0 & 0 & 0 & 0 & -\alpha
 \end{pmatrix} \quad (6)$$

$$\begin{pmatrix}
-\frac{A_b E_b}{L_b} & 0 & 0 \\
0 & -\frac{12 E_b I_b}{L_b^3} & 0 \\
0 & 0 & 0 \\
0 & -\frac{6 E_b I_b}{L_b^2} & 0 \\
0 & -\frac{6 E_b I_b}{L_b^2} & -\alpha \\
\frac{A_b E_b}{L_b} + \frac{12 E_c I_c}{L_c^3} & 0 & \frac{6 E_c I_c}{L_c^2} \\
0 & \frac{A_c E_c}{L_c} + \frac{12 E_b I_b}{L_b^3} & -\frac{6 E_b I_b}{L_b^2} \\
\frac{6 E_c I_c}{L_c^2} & -\frac{6 E_b I_b}{L_b^2} & \alpha + \frac{4 E_c I_c}{L_c}
\end{pmatrix}
\begin{pmatrix}
d2_x \\
d2_y \\
r2_z \\
r5_z \\
r6_z \\
d3_x \\
d3_y \\
r3_z
\end{pmatrix}
=
\begin{pmatrix}
H \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{pmatrix}$$

It is readily seen that this is a linear system involving rational dependencies between the frame parameters. Typical nominal parameter values and the corresponding worst case uncertainties, as proposed in (Corliss et al., 2004), are shown in Table I.

Table I. Parameters involved in the steel frame example, their nominal values, and worst case uncertainties.

parameter		nominal value	uncertainty
Young modulus	E_b	$29 * 10^6$ lbs/in ²	$\pm 348 * 10^4$
	E_c	$29 * 10^6$ lbs/in ²	$\pm 348 * 10^4$
Second moment	I_b	510 in ⁴	± 51
	I_c	272 in ⁴	± 27.2
Area	A_b	10.3 in ²	± 1.3
	A_c	14.4 in ²	± 1.44
External force	H	5305.5 lbs	± 2203.5
Joint stiffness	α	$2.77461 * 10^9$ lb-in/rad	$\pm 1.26504 * 10^9$
Length	L_c	144 in, L_b 288 in	

Initially, the system (6), where L_b, L_c are replaced by their nominal values, is solved with parameter uncertainties which are 1% of the values presented in the last column of Table I,

$$\begin{aligned}
E_b, E_c &\in [28965200, 29034800], \quad I_b \in [509.49, 510.51], \quad I_c \in [271.728, 272.272], \\
A_b &\in [10.287, 10.313], \quad A_c \in [14.3856, 14.4144], \quad \alpha \in [276195960, 278726040], \\
H &\in [5283.465, 5327.535].
\end{aligned} \tag{7}$$

Applying the rigorous monotonicity approach we have found the monotonicity profile of the system response presented in Table II which proves that the combinatorial approach gives the exact hull in exact arithmetic. Note, that all solution components are only locally monotone with

respect to Ab . The exact hull $[h]$ of the solution set for this problem, computed in rational arithmetic and then rounded outwardly to 10 digits accuracy, is presented in (Popova, 2005).

Table II. One-bay steel frame example with uncertain parameters (7): monotonicity properties of the system response.

solution component	parameter							
	E_b	E_c	I_b	I_c	A_b	A_c	α	H
1. $d2_x$	-1	-1	-1	-1	-1, -1	-1	-1	1
2. $d2_y$	1	-1	1	-1	-1, -1	-1	1	1
3. $r2_z$	1	1	1	1	1 1	1	1	-1
4. $r5_z$	1	1	1	1 1	1 1	1	-1	-1
5. $r6_z$	1	1	1	1	-1 -1	1	-1	-1
6. $d3_x$	-1	-1	-1	-1	1 1	-1	-1	1
7. $d3_y$	-1	1	-1	1	-1 -1	1	1	-1
8. $r3_z$	1	1	1	1	-1 -1	1	1	-1

The parametric linear system (6) is solved by the presented general parametric fixed-point iteration. The system involves eight uncertain parameters which are considered to vary independently within tolerance intervals (7). The guaranteed outer enclosure $[u]$ of the system response and an inner estimation $[v]$ of the outer enclosure, obtained in just one single execution of the parametric solver function, are presented with 10 digits accuracy in (Popova, 2005). The quality of the obtained enclosure is measured by the three measures of overestimation, defined in Section 2.3, and also compared to the quality of the solution enclosures for the same problem obtained by alternative methods used in (Corliss et al., 2004), see Table III.

The second and third columns in Table III demonstrate the relation $\frac{1}{2}\mathcal{O}_w([v], [u]) \approx \mathcal{O}_w([h], [u])$. The distance-based measure \mathcal{O}_d gives two numbers with different signs corresponding to the endpoints of the intervals. As demonstrated by the results in Table III, this measure yields values which are two orders of magnitude less than the overestimation measure $\mathcal{O}_w([h], [u])$. The other overestimation measure $\mathcal{O}_c([h], [u], \mu)$ is also not comparable to $\mathcal{O}_w([h], [u])$ giving values with one order of magnitude less than the latter.

The last three columns in Table III present the quality of the solution enclosures obtained in (Corliss et al., 2004) by the application of EBE approach (Muhanna and Mullen, 2001) to the system (6)–(7). The application of the EBE approach was successively improved in (Corliss et al., 2004) by applying subdistributivity property and scaling which has resulted in improved solution enclosures measured by $\mathcal{O}_c([\tilde{h}], [u_i], \tilde{\mu})$, where $[\tilde{h}]$ is the solution set hull reported in (Corliss et al., 2004), and $[u_i]$ is the corresponding solution enclosure. Comparing the best solution enclosure, obtained by the EBE approach — $\mathcal{O}_c([\tilde{h}], [u_3], \tilde{\mu})$, to the quality $\mathcal{O}_c([h], [u], \mu)$ of the solution enclosure obtained by the present parametric method, we see the superiority of the present method by one order of magnitude. The results in Table III show also that the different components of the system response have different sensitivity to variations in the system parameters.

Table III. One-bay steel frame example with uncertain parameters (7): comparison of overestimation measures in %. $\mathcal{O}_c([h], [u_i])$ are after (Corliss et al., 2004), $i = 3$ – Table V, $i = 2$ – Table IV, $i = 1$ – Table III, respectively, dash means no available data.

solution comp.	$\frac{1}{2}\mathcal{O}_\omega$ ($[v], [u]$)	\mathcal{O}_ω ($[h], [u]$)	$10^2\mathcal{O}_d$ ($[h], [u]$)	\mathcal{O}_c ($[h], [u], \mu$)	\mathcal{O}_c ($[\tilde{h}], [u_3], \mu$)	\mathcal{O}_c ($[\tilde{h}], [u_2], \mu$)	\mathcal{O}_c ($[\tilde{h}], [u_1], \mu$)
1. $d2_x$	0.83	0.83	-0.75, 0.38	0.011	0.29	0.40	78.02
2. $d2_y$	0.57	0.57	-0.86, 0.20	0.011	0.004	0.13	85.38
3. $r2_z$	4.58	4.31	3.01, -3.53	0.065	0.75	0.84	81.18
4. $r5_z$	8.65	7.73	5.89, -6.31	0.122	1.62	1.63	85.32
5. $r6_z$	13.54	11.99	9.81, -10.32	0.201	–	–	–
6. $d3_x$	0.84	0.84	-0.76, 0.39	0.011	–	–	–
7. $d3_y$	0.79	0.79	0.33, -1.21	0.015	–	–	–
8. $r3_z$	3.40	3.23	2.19, -2.70	0.049	–	–	–

It is well-known that the parametric fixed-point iteration gives sharper solution enclosures for smaller interval tolerances. To illustrate this effect we have subdivided the ranges (7) of some interval-valued parameters and obtain enclosure of the system response as a hull of the solution enclosures in all sub-domains. The results obtained after the application of the subdivision approach, reported in (Popova, 2005), show an improvement between 0.37% and 3.05% in the solution enclosure obtained by subdivision of the intervals. The overestimation for the different components of the system response is different ranging from 0.2% to 9.22%.

Table IV. One-bay steel frame example with worst-case parameter uncertainties (Table I) solved by subdivision of the parameter intervals $(E_b, E_c, I_b, I_c, A_b, A_c, \alpha, H)^\top$ correspondingly into $(2, 2, 2, 2, 1, 1, 1, 1)^\top$ equal subintervals. Inner $[v_s]$ and outer $[u_s]$ inclusions of the solution set hull are compared to the combinatorial solution $[\tilde{h}]$.

	$d2_x$	$d2_y$	$r2_z$	$r5_z$	$r6_z$	$d3_x$	$d3_y$	$r3_z$
$\frac{1}{2}\mathcal{O}_\omega([v_s], [u_s])$	19.97	15.87	–	–	–	20.12	23.50	–
$\mathcal{O}_\omega([\tilde{h}], [u_s])$	18.41	12.23	26.23	41.43	41.84	18.56	18.77	26.70

The presented parametric fixed-point iteration fails in solving the parametric linear system (6) for the worst case (over 40%) parameter uncertainties given in Table I. For very large uncertainties the parametric matrix is not strongly regular as required by the method. But we can solve the problem by subdividing the parameter intervals. As small are the sub-domains as better will

be the solution enclosure. Inclusions (inner and outer) of the solution set hull are obtained by subdivision of the worst-case parameter intervals $(E_b, E_c, I_b, I_c, A_b, A_c, \alpha, H)^\top$ correspondingly into $(2, 2, 2, 2, 1, 1, 1, 1)^\top$ equal subintervals. The quality of the obtained outer enclosure is presented in Table IV. Although the inner estimations for the most sensitive solution components are empty set intervals, a minimal number of subdivisions provided an outer enclosure overestimating the combinatorial solution with 12% to 42%. These results show that even for comparatively large parameter intervals, the presented parametric fixed-point iteration is able to enclose the solution. Although the parametric matrix is strongly regular (which provides convergence of the method) even for the large parameter uncertainties that are chosen, a poor accuracy of the residual vector enclosure may be the reason for overestimating the system response.

3.2. TWO-BAY TWO-STORY FRAME

As large frame examples we consider rectangular multi-story multi-bay frames. We model the two-bay two-story steel frame with IPE 400 beams and HE 280 B columns as shown in Figure 2.

Figure 2. Two-bay two-story steel frame.

The frame is subjected to lateral static forces and vertical uniform loads. Beam-to-column connections are considered to be semi-rigid and are modelled by single rotational spring elements. The use of spring models is better fitted to steel frames with bolted connections, for example beam-to column connections of extended-end-plate system. Semi-rigid steel and reinforced concrete frames have been widely used to reduce the seismic loading. However many structures of this type have been strongly damaged or collapsed during the Northridge earthquake, which struck Southern California in 1994. The main reason has been found to be the increased flexibility of entire frame being strongly influenced by P- Δ effect. Semi-rigid frames are in large extent sensitive to physical properties of the beam-to-column connections and this was the main reason to direct our research in this direction.

Structure elements are specified to be beam or column. Columns are chosen to be traditional 2D frame elements for the elastic analysis having three degrees of freedom per node – two translations and one rotation. Beam elements also have three degrees of freedom at each node – two translations and one rotation. The rotational springs are added to both ends and internal rotations are eliminated. Beam elements allow for application of traditional finite element procedure which requires matrices of order 6×6 . Basement nodes are fixed and are not able to displace. Applying conventional methods for analysis of frame structures, cf. (Zienkiewicz, 1971), a system of 18 linear equations is composed where the coefficients are rational functions of the model parameters. The distributed beam loading is transformed to the equivalent nodal forces. In this manner the parameters related to the geometric properties are included in the global loading vector.

In contrast to the system considered in Section 3.1 the linear system describing present two-bay two-story frame in Figure 2 has the following right-hand side vector whose components depend also on parameters of the beams, not only on the applied loadings

$$\left(f_2, -\frac{1}{2}w_1Lb_1, -\frac{w_1Lb_1^2}{12\left(1 + \frac{2Eb_1Ib_1}{cLb_1}\right)}, 0, -\frac{w_1Lb_1}{2} - \frac{w_2Lb_2}{2}, \frac{w_1Lb_1^2}{12\left(1 + \frac{2Eb_1Ib_1}{cLb_1}\right)} - \frac{w_2Lb_2^2}{12\left(1 + \frac{2Eb_2Ib_2}{cLb_2}\right)}, \right.$$

$$0, -\frac{w_2 L b_2}{2}, \frac{w_2 L b_2^2}{12(1 + \frac{2E b_2 I b_2}{c L b_2})}, f_1, -\frac{1}{2w_3 L b_3}, -\frac{w_3 L b_3^2}{12(1 + \frac{2E b_3 I b_3}{c L b_3})}, \\ 0, -\frac{w_3 L b_3}{2} - \frac{w_4 L b_4}{2}, \frac{w_3 L b_3^2}{12(1 + \frac{2E b_3 I b_3}{c L b_3})} - \frac{w_4 L b_4^2}{12(1 + \frac{2E b_4 I b_4}{c L b_4})}, 0, -\frac{w_4 L b_4}{2}, \frac{w_4 L b_4^2}{12(1 + \frac{2E b_4 I b_4}{c L b_4})} \Big)^T.$$

The following data, taken according to the European Standard (Eurocode 3, 2003), are used in the model.

	Columns (HE 280 B)	Beams (IPE 400)
Cross-sectional area	$A_c = 0.01314 \text{ m}^2$,	$A_b = 0.008446 \text{ m}^2$
Moment of inertia	$I_c = 19270 * 10^{-8} \text{ m}^4$,	$I_b = 23130 * 10^{-8} \text{ m}^4$
Modulus of elasticity	$E_c = 2.1 * 10^8 \text{ kN/m}^2$,	$E_b = 2.1 * 10^8 \text{ kN/m}^2$
Length	$L_c = 3 \text{ m}$,	$L_b = 2L_c \text{ m}$
Rotational spring stiffness	$c = 10^8 \text{ kN}$ (8)	
Uniform vertical load	$w_1 = \dots = w_4 = 30 \text{ kN/m}$	
Concentrated lateral forces	$f_1 = f_2 = 100 \text{ kN}$	

As a first problem a system structure having 13 uncertain parameters: $A_c, I_c, E_c, A_b, I_b, E_b, c, w_1, \dots, w_4, f_1, f_2$ was considered. The system parameters were initially taken to vary within 1% tolerance intervals $[p - p/200, p + p/200]$ where p is the corresponding parameter nominal value from (8).

Table V. Solutions for displacements and rotations of two-bay two-story frame system with 13 parameters having 1% uncertainties.

	$dx_1(\text{m})$	$dy_1(\text{m})$	$\theta_1(\text{rad})$	$dx_3(\text{m})$	$dy_3(\text{m})$	$\theta_3(\text{rad})$
$10^3[v]$	[12.80, 13.20]	[-.2143, -.2062]	[-2.168, -2.099]	[12.21, 12.60]	[-.3439, -.3333]	[-.2079, -.1554]
$10^3[u]$	[12.78, 13.21]	[-.2145, -.2060]	[-2.175, -2.092]	[12.20, 12.62]	[-.3441, -.3331]	[-.2146, -.1487]
$\mathcal{O}_\omega([h], [u])$	4.90	3.20	9.24	4.98	2.93	11.04
$\mathcal{O}_c([h], [u])$	0.16	0.13	0.35	0.17	0.09	3.99

The parametric solver, presented in this paper, found a guaranteed outer enclosure $[u]$ of the system response and a corresponding inner estimation $[v]$ of the solution set hull. The results for displacements and rotations of selected nodes are given in Table V. The system response at the first three nodes is most sensitive to the variations in model parameters. The bounds for the solution are captured by sharp intervals. Applying rigorously the monotonicity approach based on verified parametric solver, it was numerically proven that the combinatorial approach gives the exact solution set hull. That is why, the last two rows of Table V list the percentage by which the outer enclosures produced by the parametric solver overestimate the true bounds of the system response. The results in Table V show that the rotations are about three times more sensitive to the

variations in model parameters than the displacements. The same behavior was observed during the analysis of the portal structure in Section 3.1.

Further, we solve the same parametric system where the element material properties are taken to vary within 1% tolerances while the spring stiffness and all applied loadings are taken to vary within large 10% tolerance intervals. Table VI presents the results obtained for the nodes one and three. The

Table VI. Interval solutions for displacements and rotations of two-bay two-story frame system with 13 parameters. The material properties have 1% uncertainties while the spring stiffness and the applied loadings have 10% uncertainties.

	dx ₁ (m)	dy ₁ (m)	θ ₁ (rad)	dx ₃ (m)	dy ₃ (m)	θ ₃ (rad)
10 ³ [h]	[12.16, 13.85]	[-.2308, -.1902]	[-2.316, -1.956]	[11.60, 13.24]	[-.3604, -.3174]	[-.3545, -.0100]
10 ³ [u]	[11.92, 13.89]	[-.2311, -.1850]	[-2.333, -1.896]	[11.36, 13.28]	[-.3607, -.3119]	[-.3724, .04663]
$\mathcal{O}_\omega([h], [u])$	14.51	12.10	17.60	14.61	12.04	17.82
$\mathcal{O}_c([h], [u])$	2.19	2.65	3.59	2.25	1.73	41.05
solutions after applying the monotonicity properties w.r.t. the applied loadings						
10 ³ [u]	[12.13, 13.88]	[-.2314, -.1896]	[-2.323, -1.949]	[11.56, 13.28]	[-.3613, -.3165]	[-.3599, -4.e ⁻⁶]
$\mathcal{O}_\omega([h], [u])$	3.99	2.75	3.89	3.99	3.83	3.05
$\mathcal{O}_c([h], [u])$	0.54	0.54	0.68	0.55	0.50	5.95

first row in Table VI gives the combinatorial solution which is used for measuring the overestimation produced by the parametric solver. Except for θ₃, interval bounds for the system response are reasonable although not quite sharp. The percentage of overestimation increases with increasing the width of the parameter intervals. The lower quality of the solution enclosures for large parameter intervals is probably due to a poor range estimation of the residual vector in the algorithm. Proving monotonicity properties of the system response with respect to the loadings parameters $w_1, \dots, w_4, f_1, f_2$ and solving corresponding parametric systems involving reduced number of parameters results in a quite sharp solution enclosure presented in the second part of Table VI.

It should be noted that for 10% tolerance intervals of the model parameters even the combinatorial solution is such that the interval for θ₃ contains zero.

As a second larger problem of this type we consider the same system structure as above but assuming that each structure element has properties varying independently within 1% tolerance intervals. This leads to an 18 × 18 parametric system involving 37 interval parameters. The results for displacements and rotations of the selected nodes, listed in Table VII, are similar to those obtained for the system involving 13 parameters having the same uncertainties.

While the combinatorial solution for the problem involving 37 uncertain parameters requires solving 2³⁷ ≈ 1.37 * 10¹¹ point linear systems (in rational arithmetic), or applying Monte Carlo simulation usually takes 10⁶ trials in order to assess the quality of a solution enclosure, just one single execution of our parametric solver yields both guaranteed outer solution enclosure [u] and its

Table VII. Solutions for displacements and rotations of two-bay two-story frame system with 37 parameters having 1% uncertainties.

	dx_1 (m)	dy_1 (m)	θ_1 (rad)	dx_3 (m)	dy_3 (m)	θ_3 (rad)
$[v] * 10^3$	[12.67, 13.32]	[-.224, -.1964]	[-2.222, -2.045]	[12.09, 12.73]	[-.3571, -.3199]	[-.2569, -.1062]
$[u] * 10^3$	[12.62, 13.37]	[-.2249, -.1954]	[-2.237, -2.030]	[12.04, 12.77]	[-.3584, -.3186]	[-.2716, -.0915]
$\frac{1}{2}\mathcal{O}_\omega([v], [u])$	6.05	3.44	7.28	6.16	3.19	8.18
$\frac{1}{2}\mathcal{O}_c([v], [u])$	0.34	0.48	0.70	0.36	0.37	8.09

inner estimation $[v]$, based on which $1/2\mathcal{O}_\omega([v], [u])$ measures the quality of the obtained solution bounds.

4. Conclusion

The application of a self-verified parametric iteration method for bounding the response of uncertain mechanical structures modelled by finite element method is presented. The method can solve linear systems involving arbitrary non-linear dependencies between the uncertain input data, provided that it is combined with good tools for range enclosure. It is demonstrated that very sharp solution enclosures are generated for small parameter tolerances. Powerful range enclosing techniques are necessary to provide good accuracy of the solution enclosure when the system parameters are subjected to large uncertainties which retain the strong regularity property of the parametric matrix.

We have demonstrated the feasibility of the general-purpose parametric iteration method for bounding structure responses in the presence of uncertainties in all model parameters. It was illustrated by the numerical examples that for small intervals the method is superior to other, although not self-verified, methods like the EBE approach. Even for quite large parameter uncertainties, the interval subdivision guarantee the feasibility of the method and the accuracy of the inclusions.

The most attractive feature of the discussed methodology and software tools consists in the fact that they yield validated inclusions computed by a finite precision arithmetic. To provide this feature a rigorous computer implementation by interval arithmetic with directed roundings is necessary. Any self-verified parametric solver can be incorporated in a general framework for computer-assisted proof of global and local monotonicity properties of a parametric solution. Basing on these properties, a guaranteed and highly accurate enclosure of the solution set hull can be computed.

Contrary to other approaches for modelling uncertain mechanical systems that apply special techniques at the level of constructing the linear system to be solved in order to reduce the dependencies, the present method requires no preliminary specialized construction methods. For example, there is no need to overcome the coupling as in the EBE approach. Present method is highly automated since engineers need to apply only conventional methods for obtaining the linear system in a parametric form by software tools widely available in modern computing environments

(Matlab, *Mathematica*, etc.). Uncertainties in all the system parameters (e.g., material, load and geometry properties) can be considered and handled simultaneously. A combination of interval methods can ensure very sharp bounds for the system response. Furthermore, the present method and all the methods combined to obtain sharp bounds for the system response, are implemented in software tools which are freely available and ready for application. When the construction methods, used for assembling the global stiffness matrix and the global loading vector, cannot eliminate all the dependencies between the input parameters, a parametric iteration, respectively the implemented parametric solver, should be used instead of a non-parametric one.

Being the only general-purpose parametric linear solver, the presented methodology and software tools are applicable in the context of any problem which requires solving of linear systems whose input data depend on uncertain (interval) parameters.

References

- Corliss, G., C. Foley, and R. B. Kearfott. Formulation for Reliable Analysis of Structural Frames. In R. L. Muhanna and R. L. Mullen, editors, *Proceedings of NSF workshop on Reliable Engineering Computing*, Savannah, Georgia, September 2004, USA.
- Corliss, G., and C. Foley. Intervals in Structural Engineering, Second Scandinavian Workshop on Interval Methods and Their Applications, TU Denmark, August 2005. <http://www2.imm.dtu.dk/~km/int-05/Slides/corliss.pdf>
- Dessombz O., F. Thouverez, J.-P. L  n  , and L. J  z  quel. Analysis of Mechanical Systems Using Interval Computations Applied to Finite Element Methods. *Journal of Sound and Vibration* 239(5):949–968, 2001.
- Eurocode 3. European Standard. Eurocode 3: Design of Steel Structures. European Committee for Standardization, Ref.No. prEN 1993-1-1:2003 E, Brussels, 2003.
- Ganzerli, S., and C. P. Pantelides. Load and Resistance Convex Models for Optimum Design. *Struct. Optim.* 17:259–268, 1999.
- Gardenes, E., M. A. Sainz, L. Jorba, R. Calm, R. Estela, H. Mielgo, A. Trep  t. Modal intervals. *Reliable Computing* 7(2):77–111, 2001.
- Jansson, C. Interval Linear Systems with Symmetric Matrices, Skew-Symmetric Matrices and Dependencies in the Right Hand Side. *Computing* 46:265–274, 1991.
- McWilliam, S. Ant-Optimisation of Uncertain Structures Using Interval Analysis. *Comput. Struct.* 79:421–430, 2000.
- Moore, R. E. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.
- Muhanna, R. L., R. L. Mullen. Uncertainty in Mechanics Problems — Interval-Based Approach. *J. Eng. Mech* 127(6):557–566, 2001.
- Muhanna, R. L., R. L. Mullen, and H. Zhang. Interval Finite-Element as a Basis for Generalized Models of Uncertainty in Engineering Mechanics. In R. L. Muhanna and R. L. Mullen, editors, *Proceedings of NSF workshop on Reliable Engineering Computing*, Savannah, Georgia, September 2004, USA.
- Muhanna, R. L., R. L. Mullen, and H. Zhang. Penalty-Based Solution for the Interval Finite-Element Methods. *J. Eng. Mech* 131(10):1102–1111, 2005.
- Mullen, R. L., and R. L. Muhanna. Bounds of Structural Response for All Possible Loadings. *J. Struct. Eng.* 125(1):98–106, 1999.
- Neumaier, A. Overestimation in Linear Interval Equations. *IAM J. Numer. Anal.* 24:207–214, 1987.
- Neumaier, A. *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge, England, 1990.
- Neumaier, A., and A. Pownuk. Linear Systems with Large Uncertainties, with Applications to Truss Structures. 2005. <http://www.mat.univie.ac.at/~neum/ms/linunc.pdf>
- Popova, E. D. On the Solution of Parametrised Linear Systems. In W. Kraemer and J. Wolff von Gudenberg, editors, *Scientific Computing, Validated Numerics, Interval Methods*, Kluwer Academic Publishers, 2001.
- Popova, E. D. Parametric Interval Linear Solver. *Numerical Algorithms* 37(1–4):345–356, 2004(a).
- Popova, E. D. Strong Regularity of Parametric Interval Matrices. In I. Dimovski et al., editors, *Mathematics and Education in Mathematics*, Proceedings of 33rd Spring Conference of the Union of Bulgarian Mathematicians, Sofia, Bulgaria, UBM, 446–451, 2004(b). (<http://www.math.bas.bg/~epopova/papers/04smbEP.pdf>)
- Popova, E. D. Generalizing the Parametric Fixed-Point Iteration. *PAMM: Proceedings in Applied Mathematics & Mechanics* 4(1):680–681, 2004(c).
- Popova, E. D. Solving Linear Systems whose Input Data are Rational Functions of Interval Parameters. Preprint 3/2005, Institute of Mathematics and Informatics, BAS, Sofia, 2005.
- Popova, E. D., M. Datcheva, R. Iankov, T. Schanz. Mechanical Models with Interval Parameters. In K. G  rlebeck, L. Hempel, C. K  nke, editors, IKM2003: Digital Proceedings of 16th International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering, Weimar, 2003, Germany.
- Popova, E., and W. Kr  mer. Inner and Outer Bounds for the Solution Set of Parametric Linear Systems. *J. of Computational and Applied Mathematics*, 2004, in press.
- Pownuk, A. Calculations of Displacement in Elastic and Elastic-Plastic Structures with Interval Parameters. 33rd Solid Mechanics Conference, Zakopane, Poland, September, 2000.
- Rao, S. S., and L. Berke. Analysis of Uncertain Structural Systems using Interval Analysis. *AIAA J.* 35(4):727–735.

- Rump, S. New Results on Verified Inclusions. In W. L. Miranker, and R. Toupin, editors, *Accurate Scientific Computations*, Springer LNCS 235, 31–69, 1986.
- Rump, S. Rigorous sensitivity analysis for systems of linear and nonlinear equations. *Mathematics of Computation* 54(190):721–736, 1990.
- Rump, S. Verification methods for dense and sparse systems of equations. In J. Herzberger, editor, *Topics in Validated Computations*, N. Holland, 63–135, 1994.
- Wolfram, S. *The Mathematica Book*. 4th ed., Wolfram Media/Cambridge U. Press, 1999.
- Zienkiewicz, O. C. *The Finite Element Method in Engineering Science*. McGraw-Hill, London, 1971.

Overview of Reliability Analysis and Design Capabilities in DAKOTA

M. S. Eldred*

Sandia National Laboratories[†], Albuquerque, NM 87185

B. J. Bichon[‡]

Vanderbilt University, Nashville, TN 37235

B. M. Adams[§]

Sandia National Laboratories, Albuquerque, NM 87185

Abstract. Reliability methods are probabilistic algorithms for quantifying the effect of uncertainties in simulation input on response metrics of interest. In particular, they compute approximate response function distribution statistics (probability, reliability, and response levels) based on specified probability distributions for input random variables. In this paper, recent algorithm research in first and second-order reliability methods is overviewed for both the forward reliability analysis of computing probabilities for specified response levels (the reliability index approach (RIA)) and the inverse reliability analysis of computing response levels for specified probabilities (the performance measure approach (PMA)). A number of algorithmic variations have been explored, and the effect of different limit state approximations, probability integrations, warm starting, most probable point search algorithms, and Hessian approximations is discussed. These reliability analysis capabilities provide the foundation for reliability-based design optimization (RBDO) methods, and bi-level and sequential formulations are presented. These RBDO formulations may employ analytic sensitivities of reliability metrics with respect to design variables that either augment or define distribution parameters for the uncertain variables. Relative performance of these reliability analysis and design algorithms is presented for a number of benchmark test problems using the DAKOTA software, and algorithm recommendations are given. These recommended algorithms are subsequently being applied to real-world applications in the probabilistic analysis and design of micro-electro-mechanical systems, and initial experiences with this deployment are provided.

Keywords: Uncertainty, Reliability, Reliability-based design optimization, Software, MEMS

* Principal Member of Technical Staff, Optimization and Uncertainty Estimation Department.

[†] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

[‡] NSF IGERT Fellow, Department of Civil and Environmental Engineering.

[§] Limited Term Employee, Optimization and Uncertainty Estimation Department.

1. Introduction

Reliability methods are probabilistic algorithms for quantifying the effect of uncertainties in simulation input on response metrics of interest. In particular, they perform uncertainty quantification (UQ) by computing approximate response function distribution statistics based on specified probability distributions for input random variables. These response statistics include response mean, response standard deviation, and cumulative or complementary cumulative distribution function (CDF/CCDF) response level and probability/reliability level pairings. These methods are often more efficient at computing statistics in the tails of the response distributions (events with low probability) than sampling-based approaches since the number of samples required to resolve a low probability can be prohibitive. Thus, these methods, as their name implies, are often used in a reliability context for assessing the probability of failure of a system when confronted with an uncertain environment.

A number of classical reliability analysis methods are discussed in (Haldar and Mahadevan, 2000), including Mean-Value First-Order Second-Moment (MVFOSM), First-Order Reliability Method (FORM), and Second-Order Reliability Method (SORM). More recent methods which seek to improve the efficiency of FORM analysis through limit state approximations include the use of local and multipoint approximations in Advanced Mean Value methods (AMV/AMV+ (Wu et al., 1990)) and Two-point Adaptive Nonlinearity Approximation-based methods (TANA (Wang and Grandhi, 1994; Xu and Grandhi, 1998)), respectively. Each of the FORM-based methods can be employed for “forward” or “inverse” reliability analysis through the reliability index approach (RIA) or performance measure approach (PMA), respectively, as described in (Tu et al., 1999).

The capability to assess reliability is broadly useful within a design optimization context, and reliability-based design optimization (RBDO) methods are popular approaches for designing systems while accounting for uncertainty. RBDO approaches may be broadly characterized as bi-level (in which the reliability analysis is nested within the optimization, e.g. (Allen and Maute, 2004)), sequential (in which iteration occurs between optimization and reliability analysis, e.g. (Wu et al., 2001; Du and Chen, 2004)), or unilevel (in which the design and reliability searches are combined into a single optimization, e.g. (Agarwal et al., 2004)). Bi-level RBDO methods are simple and general-purpose, but can be computationally demanding. Sequential and unilevel methods seek to reduce computational expense by breaking the nested relationship through the use of iterated or simultaneous approaches.

In order to provide access to a variety of uncertainty quantification capabilities for analysis of large-scale engineering applications on high-performance parallel computers, the DAKOTA project (Eldred et al., 2003) at Sandia National Laboratories has developed a suite of algorithmic capabilities known as DAKOTA/UQ (Wojtkiewicz et al., 2001). This package contains the reliability analysis capabilities described in this paper and enables the RBDO approaches, and is freely available for download worldwide through an open source license.

This paper overviews recent algorithm research activities that have explored a variety of approaches for performing reliability analysis. In particular, forward and inverse reliability analyses have been explored using multiple limit state approximation, probability integration, warm starting, Hessian approximation, and optimization algorithm selections. These uncertainty quantification capabilities have also provided a foundation for exploring bi-level and sequential RBDO formulations.

Sections 2 and 3 describe these algorithmic components, Section 4 summarizes computational results for four benchmark test problems, Section 5 presents initial deployment of these methodologies to the probabilistic analysis and design of MEMS, and Section 6 provides concluding remarks.

2. Reliability Method Formulations

2.1. MEAN VALUE

The Mean Value method (MV, also known as MVFOSM in (Haldar and Mahadevan, 2000)) is the simplest, least-expensive reliability method because it estimates the response means, response standard deviations, and all CDF/CCDF response-probability-reliability levels from a single evaluation of response functions and their gradients at the uncertain variable means. This approximation can have acceptable accuracy when the response functions are nearly linear and their distributions are approximately Gaussian, but can have poor accuracy in other situations. The expressions for approximate response mean μ_g , approximate response standard deviation σ_g , response target to approximate probability/reliability level mapping ($\bar{z} \rightarrow p, \beta$), and probability/reliability target to approximate response level mapping ($\bar{p}, \bar{\beta} \rightarrow z$) are

$$\mu_g = g(\mu_{\mathbf{x}}) \quad (1)$$

$$\sigma_g = \sum_i \sum_j Cov(i, j) \frac{dg}{dx_i}(\mu_{\mathbf{x}}) \frac{dg}{dx_j}(\mu_{\mathbf{x}}) \quad (2)$$

$$\beta_{cdf} = \frac{\mu_g - \bar{z}}{\sigma_g} \quad (3)$$

$$\beta_{ccdf} = \frac{\bar{z} - \mu_g}{\sigma_g} \quad (4)$$

$$z = \mu_g - \sigma_g \bar{\beta}_{cdf} \quad (5)$$

$$z = \mu_g + \sigma_g \bar{\beta}_{ccdf} \quad (6)$$

respectively, where \mathbf{x} are the uncertain values in the space of the original uncertain variables (“x-space”), $g(\mathbf{x})$ is the limit state function (the response function for which probability-response level pairs are needed), and β_{cdf} and β_{ccdf} are the CDF and CCDF reliability indices, respectively.

With the introduction of second-order limit state information, MVSOSM calculates a second-order mean as

$$\mu_g = g(\mu_{\mathbf{x}}) + \frac{1}{2} \sum_i \sum_j Cov(i, j) \frac{d^2g}{dx_i dx_j}(\mu_{\mathbf{x}}) \quad (7)$$

This is commonly combined with a first-order variance (Eq. 2), since second-order variance involves higher order distribution moments (skewness, kurtosis) (Haldar and Mahadevan, 2000) which are often unavailable.

The first-order CDF probability $p(g \leq z)$, first-order CCDF probability $p(g > z)$, β_{cdf} , and β_{ccdf} are related to one another through

$$p(g \leq z) = \Phi(-\beta_{cdf}) \quad (8)$$

$$p(g > z) = \Phi(-\beta_{ccdf}) \quad (9)$$

$$\beta_{cdf} = -\Phi^{-1}(p(g \leq z)) \quad (10)$$

$$\beta_{ccdf} = -\Phi^{-1}(p(g > z)) \quad (11)$$

$$\beta_{cdf} = -\beta_{ccdf} \quad (12)$$

$$p(g \leq z) = 1 - p(g > z) \quad (13)$$

where $\Phi()$ is the standard normal cumulative distribution function. A common convention in the literature is to define g in such a way that the CDF probability for a response level z of zero (i.e., $p(g \leq 0)$) is the response metric of interest. The formulations in this paper are not restricted to this convention and are designed to support CDF or CCDF mappings for general response, probability, and reliability level sequences.

2.2. MPP SEARCH METHODS

All other reliability methods solve a nonlinear optimization problem to compute a most probable point (MPP) and then integrate about this point to compute probabilities. The MPP search is performed in uncorrelated standard normal space (“u-space”) since it simplifies the probability integration: the distance of the MPP from the origin has the meaning of the number of input standard deviations separating the mean response from a particular response threshold. The transformation from correlated non-normal distributions (x-space) to uncorrelated standard normal distributions (u-space) is denoted as $\mathbf{u} = T(\mathbf{x})$ with the reverse transformation denoted as $\mathbf{x} = T^{-1}(\mathbf{u})$. These transformations are nonlinear in general, and possible approaches include the Rosenblatt (Rosenblatt, 1952), Nataf (Der Kiureghian and Liu, 1986), and Box-Cox (Box and Cox, 1964) transformations. The nonlinear transformations may also be linearized, and common approaches for this include the Rackwitz-Fiessler (Rackwitz and Fiessler, 1978) two-parameter equivalent normal and the Chen-Lind (Chen and Lind, 1983) and Wu-Wirsching (Wu and Wirsching, 1987) three-parameter equivalent normals. The results in this paper employ the Nataf nonlinear transformation which occurs in the following two steps. To transform between the original correlated x-space variables and correlated standard normals (“z-space”), the CDF matching condition is used:

$$\Phi(z_i) = F(x_i) \quad (14)$$

where $F()$ is the cumulative distribution function of the original probability distribution. Then, to transform between correlated z-space variables and uncorrelated u-space variables, the Cholesky factor \mathbf{L} of a modified correlation matrix is used:

$$\mathbf{z} = \mathbf{L}\mathbf{u} \quad (15)$$

where the original correlation matrix for non-normals in x-space has been modified to represent the corresponding correlation in z-space (Der Kiureghian and Liu, 1986).

The forward reliability analysis algorithm of computing CDF/CCDF probability/reliability levels for specified response levels is called the reliability index approach (RIA), and the inverse reliability analysis algorithm of computing response levels for specified CDF/CCDF probability/reliability levels is called the performance measure approach (PMA) (Tu et al., 1999). The differences between the

RIA and PMA formulations appear in the objective function and equality constraint formulations used in the MPP searches. For RIA, the MPP search for achieving the specified response level \bar{z} is formulated as

$$\begin{aligned} & \text{minimize} && \mathbf{u}^T \mathbf{u} \\ & \text{subject to} && G(\mathbf{u}) = \bar{z} \end{aligned} \quad (16)$$

and for PMA, the MPP search for achieving the specified reliability/probability level $\bar{\beta}, \bar{p}$ is formulated as

$$\begin{aligned} & \text{minimize} && \pm G(\mathbf{u}) \\ & \text{subject to} && \mathbf{u}^T \mathbf{u} = \bar{\beta}^2 \end{aligned} \quad (17)$$

where \mathbf{u} is a vector centered at the origin in u-space and $g(\mathbf{x}) \equiv G(\mathbf{u})$ by definition. In the RIA case, the optimal MPP solution \mathbf{u}^* defines the reliability index from $\beta = \pm \|\mathbf{u}^*\|_2$, which in turn defines the CDF/CCDF probabilities (using Eqs. 8-9 in the case of first-order integration). The sign of β is defined by

$$G(\mathbf{u}^*) > G(\mathbf{0}) : \beta_{cdf} < 0, \beta_{ccdf} > 0 \quad (18)$$

$$G(\mathbf{u}^*) < G(\mathbf{0}) : \beta_{cdf} > 0, \beta_{ccdf} < 0 \quad (19)$$

where $G(\mathbf{0})$ is the median limit state response computed at the origin in u-space (where $\beta_{cdf} = \beta_{ccdf} = 0$ and first-order $p(g \leq z) = p(g > z) = 0.5$). In the PMA case, the sign applied to $G(\mathbf{u})$ (equivalent to minimizing or maximizing $G(\mathbf{u})$) is similarly defined by $\bar{\beta}$

$$\bar{\beta}_{cdf} < 0, \bar{\beta}_{ccdf} > 0 : \text{maximize } G(\mathbf{u}) \quad (20)$$

$$\bar{\beta}_{cdf} > 0, \bar{\beta}_{ccdf} < 0 : \text{minimize } G(\mathbf{u}) \quad (21)$$

and the limit state at the MPP ($G(\mathbf{u}^*)$) defines the desired response level result.

When performing PMA with specified \bar{p} , one must compute $\bar{\beta}$ to include in Eq. 17. While this is a straightforward one-time calculation for first-order integrations (Eqs. 10-11), the use of second-order integrations complicates matters since the $\bar{\beta}$ corresponding to the prescribed \bar{p} is a function of the Hessian of G (see Eq. 38), which in turn is a function of location in u-space. A generalized reliability index (Eq. 50), which would allow a one-time calculation, may not be used since equality with $\mathbf{u}^T \mathbf{u}$ is not meaningful. The $\bar{\beta}$ target must therefore be updated in Eq. 17 as the minimization progresses (e.g., using Newton's method to solve Eq. 38 for $\bar{\beta}$ given \bar{p} and κ_i). This works best when $\bar{\beta}$ can be fixed during the course of an approximate optimization, such as for the AMV²⁺ and TANA methods described in Section 2.2.1. For second-order PMA without limit state approximation cycles (i.e., PMA SORM), the constraint must be continually updated and the constraint derivative should include $\nabla_{\mathbf{u}} \bar{\beta}$, which would require third-order information for the limit state to compute derivatives of the principal curvatures. This is impractical, so the PMA SORM constraint derivatives are only approximated analytically or estimated numerically. Potentially for this reason, PMA SORM has not been widely explored in the literature.

2.2.1. Limit state approximations

There are a variety of algorithmic variations that can be explored within RIA/PMA reliability analysis. First, one may select among several different limit state approximations that can be used to reduce computational expense during the MPP searches. Local, multipoint, and global approximations of the limit state are possible. (Eldred et al., 2005) investigated local first-order limit state approximations, and (Eldred et al., 2006) investigated local second-order and multipoint approximations. These techniques include:

1. a single Taylor series per response/reliability/probability level in \mathbf{x} -space centered at the uncertain variable means. The first-order approach is commonly known as the Advanced Mean Value (AMV) method:

$$g(\mathbf{x}) \cong g(\mu_{\mathbf{x}}) + \nabla_{\mathbf{x}}g(\mu_{\mathbf{x}})^T(\mathbf{x} - \mu_{\mathbf{x}}) \quad (22)$$

and the second-order approach has been named AMV²:

$$g(\mathbf{x}) \cong g(\mu_{\mathbf{x}}) + \nabla_{\mathbf{x}}g(\mu_{\mathbf{x}})^T(\mathbf{x} - \mu_{\mathbf{x}}) + \frac{1}{2}(\mathbf{x} - \mu_{\mathbf{x}})^T \nabla_{\mathbf{x}}^2g(\mu_{\mathbf{x}})(\mathbf{x} - \mu_{\mathbf{x}}) \quad (23)$$

2. same as AMV/AMV², except that the Taylor series is expanded in \mathbf{u} -space. The first-order option has been termed the \mathbf{u} -space AMV method:

$$G(\mathbf{u}) \cong G(\mu_{\mathbf{u}}) + \nabla_{\mathbf{u}}G(\mu_{\mathbf{u}})^T(\mathbf{u} - \mu_{\mathbf{u}}) \quad (24)$$

where $\mu_{\mathbf{u}} = T(\mu_{\mathbf{x}})$ and is nonzero in general, and the second-order option has been named the \mathbf{u} -space AMV² method:

$$G(\mathbf{u}) \cong G(\mu_{\mathbf{u}}) + \nabla_{\mathbf{u}}G(\mu_{\mathbf{u}})^T(\mathbf{u} - \mu_{\mathbf{u}}) + \frac{1}{2}(\mathbf{u} - \mu_{\mathbf{u}})^T \nabla_{\mathbf{u}}^2G(\mu_{\mathbf{u}})(\mathbf{u} - \mu_{\mathbf{u}}) \quad (25)$$

3. an initial Taylor series approximation in \mathbf{x} -space at the uncertain variable means, with iterative expansion updates at each MPP estimate (\mathbf{x}^*) until the MPP converges. The first-order option is commonly known as AMV+:

$$g(\mathbf{x}) \cong g(\mathbf{x}^*) + \nabla_{\mathbf{x}}g(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) \quad (26)$$

and the second-order option has been named AMV²+:

$$g(\mathbf{x}) \cong g(\mathbf{x}^*) + \nabla_{\mathbf{x}}g(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T \nabla_{\mathbf{x}}^2g(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \quad (27)$$

4. same as AMV+/AMV²+, except that the expansions are performed in \mathbf{u} -space. The first-order option has been termed the \mathbf{u} -space AMV+ method.

$$G(\mathbf{u}) \cong G(\mathbf{u}^*) + \nabla_{\mathbf{u}}G(\mathbf{u}^*)^T(\mathbf{u} - \mathbf{u}^*) \quad (28)$$

and the second-order option has been named the \mathbf{u} -space AMV²+ method:

$$G(\mathbf{u}) \cong G(\mathbf{u}^*) + \nabla_{\mathbf{u}}G(\mathbf{u}^*)^T(\mathbf{u} - \mathbf{u}^*) + \frac{1}{2}(\mathbf{u} - \mathbf{u}^*)^T \nabla_{\mathbf{u}}^2G(\mathbf{u}^*)(\mathbf{u} - \mathbf{u}^*) \quad (29)$$

5. a multipoint approximation in x-space. This approach involves a Taylor series approximation in intermediate variables where the powers used for the intermediate variables are selected to match information at the current and previous expansion points. Based on the two-point exponential approximation concept (TPEA, (Fadel et al., 1990)), the two-point adaptive nonlinearity approximation (TANA-3, (Xu and Grandhi, 1998)) approximates the limit state as:

$$g(\mathbf{x}) \cong g(\mathbf{x}_2) + \sum_{i=1}^n \frac{\partial g}{\partial x_i}(\mathbf{x}_2) \frac{x_{i,2}^{1-p_i}}{p_i} (x_i^{p_i} - x_{i,2}^{p_i}) + \frac{1}{2} \epsilon(\mathbf{x}) \sum_{i=1}^n (x_i^{p_i} - x_{i,2}^{p_i})^2 \quad (30)$$

where n is the number of uncertain variables and:

$$p_i = 1 + \ln \left[\frac{\frac{\partial g}{\partial x_i}(\mathbf{x}_1)}{\frac{\partial g}{\partial x_i}(\mathbf{x}_2)} \right] \bigg/ \ln \left[\frac{x_{i,1}}{x_{i,2}} \right] \quad (31)$$

$$\epsilon(\mathbf{x}) = \frac{H}{\sum_{i=1}^n (x_i^{p_i} - x_{i,1}^{p_i})^2 + \sum_{i=1}^n (x_i^{p_i} - x_{i,2}^{p_i})^2} \quad (32)$$

$$H = 2 \left[g(\mathbf{x}_1) - g(\mathbf{x}_2) - \sum_{i=1}^n \frac{\partial g}{\partial x_i}(\mathbf{x}_2) \frac{x_{i,2}^{1-p_i}}{p_i} (x_{i,1}^{p_i} - x_{i,2}^{p_i}) \right] \quad (33)$$

and \mathbf{x}_2 and \mathbf{x}_1 are the current and previous MPP estimates in x-space, respectively. Prior to the availability of two MPP estimates, x-space AMV+ is used.

6. a multipoint approximation in u-space. The u-space TANA-3 approximates the limit state as:

$$G(\mathbf{u}) \cong G(\mathbf{u}_2) + \sum_{i=1}^n \frac{\partial G}{\partial u_i}(\mathbf{u}_2) \frac{u_{i,2}^{1-p_i}}{p_i} (u_i^{p_i} - u_{i,2}^{p_i}) + \frac{1}{2} \epsilon(\mathbf{u}) \sum_{i=1}^n (u_i^{p_i} - u_{i,2}^{p_i})^2 \quad (34)$$

where:

$$p_i = 1 + \ln \left[\frac{\frac{\partial G}{\partial u_i}(\mathbf{u}_1)}{\frac{\partial G}{\partial u_i}(\mathbf{u}_2)} \right] \bigg/ \ln \left[\frac{u_{i,1}}{u_{i,2}} \right] \quad (35)$$

$$\epsilon(\mathbf{u}) = \frac{H}{\sum_{i=1}^n (u_i^{p_i} - u_{i,1}^{p_i})^2 + \sum_{i=1}^n (u_i^{p_i} - u_{i,2}^{p_i})^2} \quad (36)$$

$$H = 2 \left[G(\mathbf{u}_1) - G(\mathbf{u}_2) - \sum_{i=1}^n \frac{\partial G}{\partial u_i}(\mathbf{u}_2) \frac{u_{i,2}^{1-p_i}}{p_i} (u_{i,1}^{p_i} - u_{i,2}^{p_i}) \right] \quad (37)$$

and \mathbf{u}_2 and \mathbf{u}_1 are the current and previous MPP estimates in u-space, respectively. Prior to the availability of two MPP estimates, u-space AMV+ is used.

7. the MPP search on the original response functions without the use of any approximations.

The Hessian matrices in AMV² and AMV²+ may be available analytically, estimated numerically, or approximated through quasi-Newton updates. The quasi-Newton variant of AMV²+ is conceptually

similar to TANA in that both approximate curvature based on a sequence of gradient evaluations. TANA estimates curvature by matching values and gradients at two points and includes it through the use of exponential intermediate variables and a single-valued diagonal Hessian approximation. Quasi-Newton AMV²+ accumulates curvature over a sequence of points and then uses it directly in a second-order series expansion. Therefore, these methods may be expected to exhibit similar performance.

The selection between x-space or u-space for performing approximations depends on where the approximation will be more accurate, since this will result in more accurate MPP estimates (AMV, AMV²) or faster convergence (AMV+, AMV²+, TANA). Since this relative accuracy depends on the forms of the limit state $g(x)$ and the transformation $T(x)$ and is therefore application dependent in general, DAKOTA/UQ supports both options. A concern with approximation-based iterative search methods (i.e., AMV+, AMV²+ and TANA) is the robustness of their convergence to the MPP. It is possible for the MPP iterates to oscillate or even diverge. However, to date, this occurrence has been relatively rare, and DAKOTA/UQ contains checks that monitor for this behavior. Another concern with TANA is numerical safeguarding. First, there is the possibility of raising negative x_i or u_i values to nonintegral p_i exponents in Eqs. 30, 32-34, and 36-37. This is particularly likely for u-space. Safeguarding techniques include the use of linear bounds scaling for each x_i or u_i , offsetting negative x_i or u_i , or promotion of p_i to integral values for negative x_i or u_i . In numerical experimentation, the offset approach has been the most effective in retaining the desired data matches without overly inflating the p_i exponents. Second, there are a number of potential numerical difficulties with the logarithm ratios in Eqs. 31 and 35. In this case, a safeguarding strategy is to revert to either the linear ($p_i = 1$) or reciprocal ($p_i = -1$) approximation based on which approximation has lower error in $\frac{\partial g}{\partial x_i}(\mathbf{x}_1)$ or $\frac{\partial G}{\partial u_i}(\mathbf{u}_1)$.

2.2.2. Probability integrations

The second algorithmic variation involves the integration approach for computing probabilities at the MPP, which can be selected to be first-order (Eqs. 8-9) or second-order integration. Second-order integration involves applying a curvature correction (Breitung, 1984; Hohenbichler and Rackwitz, 1988; Hong, 1999). Breitung applies a correction based on asymptotic analysis (Breitung, 1984):

$$p = \Phi(-\beta_p) \prod_{i=1}^{n-1} \frac{1}{\sqrt{1 + \beta_p \kappa_i}} \quad (38)$$

where κ_i are the principal curvatures of the limit state function (the eigenvalues of an orthonormal transformation of $\nabla_{\mathbf{u}}^2 G$, taken positive for a convex limit state) and $\beta_p \geq 0$ (select CDF or CCDF probability correction to obtain correct sign for β_p). An alternate correction in (Hohenbichler and Rackwitz, 1988) is consistent in the asymptotic regime ($\beta_p \rightarrow \infty$) but does not collapse to first-order integration for $\beta_p = 0$:

$$p = \Phi(-\beta_p) \prod_{i=1}^{n-1} \frac{1}{\sqrt{1 + \psi(-\beta_p) \kappa_i}} \quad (39)$$

where $\psi() = \frac{\phi()}{\Phi()}$ and $\phi()$ is the standard normal density function. (Hong, 1999) applies further corrections to Eq. 39 based on point concentration methods.

To invert a second-order integration and compute β_p given p and κ_i (e.g., for second-order PMA as described in Section 2.2), Newton's method can be applied as described in (Eldred et al., 2006). Combining the no-approximation option of the MPP search with first-order and second-order integration approaches results in the traditional first-order and second-order reliability methods (FORM and SORM). Additional probability integration approaches can involve importance sampling in the vicinity of the MPP (Hohenbichler and Rackwitz, 1988; Wu, 1994), but are outside the scope of this paper. While second-order integrations could be performed anywhere a limit state Hessian has been computed, the additional computational effort is most warranted for fully converged MPPs from AMV+, AMV²+, TANA, FORM, and SORM, and is of reduced value for MVFOSM, MVSOSM, AMV, or AMV².

2.2.3. Hessian approximations

To use a second-order Taylor series or a second-order integration when second-order information ($\nabla_{\mathbf{x}}^2 g$, $\nabla_{\mathbf{u}}^2 G$, and/or κ) is not directly available, one can estimate the missing information using finite differences or approximate it through use of quasi-Newton approximations. These procedures will often be needed to make second-order approaches practical for engineering applications.

In the finite difference case, numerical Hessians are commonly computed using either first-order forward differences of gradients using

$$\nabla^2 g(\mathbf{x}) \cong \frac{\nabla g(\mathbf{x} + h\mathbf{e}_i) - \nabla g(\mathbf{x})}{h} \quad (40)$$

to estimate the i^{th} Hessian column when gradients are analytically available, or second-order differences of function values using

$$\nabla^2 g(\mathbf{x}) \cong \frac{g(\mathbf{x}+h\mathbf{e}_i+h\mathbf{e}_j)-g(\mathbf{x}+h\mathbf{e}_i-h\mathbf{e}_j)-g(\mathbf{x}-h\mathbf{e}_i+h\mathbf{e}_j)+g(\mathbf{x}-h\mathbf{e}_i-h\mathbf{e}_j)}{4h^2} \quad (41)$$

to estimate the ij^{th} Hessian term when gradients are not directly available. This approach has the advantage of locally-accurate Hessians for each point of interest (which can lead to quadratic convergence rates in discrete Newton methods), but has the disadvantage that numerically estimating each of the matrix terms can be expensive.

Quasi-Newton approximations, on the other hand, do not reevaluate all of the second-order information for every point of interest. Rather, they accumulate approximate curvature information over time using secant updates. Since they utilize the existing gradient evaluations, they do not require any additional function evaluations for evaluating the Hessian terms. The quasi-Newton approximations of interest include the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{B}_k}{\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k} + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} \quad (42)$$

which yields a sequence of symmetric positive definite Hessian approximations, and the Symmetric Rank 1 (SR1) update

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{(\mathbf{y}_k - \mathbf{B}_k \mathbf{s}_k)(\mathbf{y}_k - \mathbf{B}_k \mathbf{s}_k)^T}{(\mathbf{y}_k - \mathbf{B}_k \mathbf{s}_k)^T \mathbf{s}_k} \quad (43)$$

which yields a sequence of symmetric, potentially indefinite, Hessian approximations. \mathbf{B}_k is the k^{th} approximation to the Hessian $\nabla^2 g$, $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ is the step and $\mathbf{y}_k = \nabla g_{k+1} - \nabla g_k$ is the corresponding yield in the gradients. The selection of BFGS versus SR1 involves the importance of retaining positive definiteness in the Hessian approximations; if the procedure does not require it, then the SR1 update can be more accurate if the true Hessian is not positive definite. Initial scalings for \mathbf{B}_0 and numerical safeguarding techniques (damped BFGS, update skipping) are described in (Eldred et al., 2006).

2.2.4. Optimization algorithms

The next algorithmic variation involves the optimization algorithm selection for solving Eqs. 16 and 17. The Hasofer-Lind Rackwitz-Fissler (HL-RF) algorithm (Haldar and Mahadevan, 2000) is a classical approach that has been broadly applied. It is a Newton-based approach lacking line search/trust region globalization, and is generally regarded as computationally efficient but occasionally unreliable. DAKOTA/UQ takes the approach of employing robust, general-purpose optimization algorithms with provable convergence properties. This paper employs the sequential quadratic programming (SQP) and nonlinear interior-point (NIP) optimization algorithms from the NPSOL (Gill et al., 1998) and OPT++ (Meza, 1994) libraries, respectively.

2.2.5. Warm Starting of MPP Searches

The final algorithmic variation involves the use of warm starting approaches for improving computational efficiency. (Eldred et al., 2005) describes the acceleration of MPP searches through warm starting with approximate iteration increment, with $z/p/\beta$ level increment, and with design variable increment. Warm started data includes the expansion point and associated response values and the MPP optimizer initial guess. Projections are used when an increment in $z/p/\beta$ level or design variables occurs. Warm starts were consistently effective in (Eldred et al., 2005), with greater effectiveness for smaller parameter changes, and are used for all computational experiments presented in this paper.

3. Reliability-Based Design Optimization

Reliability-based design optimization (RBDO) methods are used to perform design optimization accounting for reliability metrics. The reliability analysis capabilities described in Section 2 provide a rich foundation for exploring a variety of RBDO formulations. (Eldred et al., 2005) investigated bi-level, fully-analytic bi-level, and first-order sequential RBDO approaches employing underlying first-order reliability assessments. (Eldred et al., 2006) investigated fully-analytic bi-level and second-order sequential RBDO approaches employing underlying second-order reliability assessments. These methods are overviewed in the following sections.

3.1. BI-LEVEL RBDO

The simplest and most direct RBDO approach is the bi-level approach in which a full reliability analysis is performed for every optimization function evaluation. This involves a nesting of two

distinct levels of optimization within each other, one at the design level and one at the MPP search level.

Since an RBDO problem will typically specify both the \bar{z} level and the $\bar{p}/\bar{\beta}$ level, one can use either the RIA or the PMA formulation for the UQ portion and then constrain the result in the design optimization portion. In particular, RIA reliability analysis maps \bar{z} to p/β , so RIA RBDO constrains p/β :

$$\begin{aligned} & \text{minimize} && f \\ & \text{subject to} && \beta \geq \bar{\beta} \\ & && \text{or } p \leq \bar{p} \end{aligned} \quad (44)$$

And PMA reliability analysis maps $\bar{p}/\bar{\beta}$ to z , so PMA RBDO constrains z :

$$\begin{aligned} & \text{minimize} && f \\ & \text{subject to} && z \geq \bar{z} \end{aligned} \quad (45)$$

where $z \geq \bar{z}$ is used as the RBDO constraint for a cumulative failure probability (failure defined as $z \leq \bar{z}$) but $z \leq \bar{z}$ would be used as the RBDO constraint for a complementary cumulative failure probability (failure defined as $z \geq \bar{z}$). It is worth noting that DAKOTA is not limited to these types of inequality-constrained RBDO formulations; rather, they are convenient examples. DAKOTA supports general optimization under uncertainty mappings (Eldred et al., 2002) which allow flexible use of statistics within multiple objectives, inequality constraints, and equality constraints.

An important performance enhancement for bi-level methods is the use of sensitivity analysis to analytically compute the design gradients of probability, reliability, and response levels. When design variables are separate from the uncertain variables (i.e., they are not distribution parameters), then the following first-order expressions may be used (Hohenbichler and Rackwitz, 1986; Karamchandani and Cornell, 1992; Allen and Maute, 2004):

$$\nabla_{\mathbf{d}} z = \nabla_{\mathbf{d}} g \quad (46)$$

$$\nabla_{\mathbf{d}} \beta_{cdf} = \frac{1}{\|\nabla_{\mathbf{u}} G\|} \nabla_{\mathbf{d}} g \quad (47)$$

$$\nabla_{\mathbf{d}} p_{cdf} = -\phi(-\beta_{cdf}) \nabla_{\mathbf{d}} \beta_{cdf} \quad (48)$$

where it is evident from Eqs. 12-13 that $\nabla_{\mathbf{d}} \beta_{ccdf} = -\nabla_{\mathbf{d}} \beta_{cdf}$ and $\nabla_{\mathbf{d}} p_{ccdf} = -\nabla_{\mathbf{d}} p_{cdf}$. In the case of second-order integrations, Eq. 48 must be expanded to include the curvature correction. For Breitung's correction (Eq. 38),

$$\nabla_{\mathbf{d}} p_{cdf} = \left[\Phi(-\beta_p) \sum_{i=1}^{n-1} \left(\frac{-\kappa_i}{2(1 + \beta_p \kappa_i)^{\frac{3}{2}}} \prod_{\substack{j=1 \\ j \neq i}}^{n-1} \frac{1}{\sqrt{1 + \beta_p \kappa_j}} \right) - \phi(-\beta_p) \prod_{i=1}^{n-1} \frac{1}{\sqrt{1 + \beta_p \kappa_i}} \right] \nabla_{\mathbf{d}} \beta_{cdf} \quad (49)$$

where $\nabla_{\mathbf{d}} \kappa_i$ has been neglected and $\beta_p \geq 0$ (see Section 2.2.2). Other approaches assume the curvature correction is nearly independent of the design variables (Rackwitz, 2002), which is equivalent to neglecting the first term in Eq. 49.

To capture second-order probability estimates within an RIA RBDO formulation using well-behaved β constraints, a generalized reliability index can be introduced where, similar to Eq. 10,

$$\beta_{cdf}^* = -\Phi^{-1}(p_{cdf}) \quad (50)$$

for second-order p_{cdf} . This reliability index is no longer equivalent to the magnitude of \mathbf{u} , but rather is a convenience metric for capturing the effect of more accurate probability estimates. The corresponding generalized reliability index sensitivity, similar to Eq. 48, is

$$\nabla_{\mathbf{d}}\beta_{cdf}^* = -\frac{1}{\phi(-\beta_{cdf}^*)}\nabla_{\mathbf{d}}p_{cdf} \quad (51)$$

where $\nabla_{\mathbf{d}}p_{cdf}$ is defined from Eq. 49. Even when $\nabla_{\mathbf{d}}g$ is estimated numerically, Eqs. 46-51 can be used to avoid numerical differencing across full reliability analyses.

When the design variables are distribution parameters of the uncertain variables, $\nabla_{\mathbf{d}}g$ is expanded with the chain rule and Eqs. 46 and 47 become

$$\nabla_{\mathbf{d}}z = \nabla_{\mathbf{d}\mathbf{x}}\nabla_{\mathbf{x}}g \quad (52)$$

$$\nabla_{\mathbf{d}}\beta_{cdf} = \frac{1}{\|\nabla_{\mathbf{u}}G\|}\nabla_{\mathbf{d}\mathbf{x}}\nabla_{\mathbf{x}}g \quad (53)$$

where the design Jacobian of the transformation ($\nabla_{\mathbf{d}\mathbf{x}}$) may be obtained analytically for uncorrelated \mathbf{x} or semi-analytically for correlated \mathbf{x} ($\nabla_{\mathbf{d}}\mathbf{L}$ is evaluated numerically) by differentiating Eqs. 14 and 15 with respect to the distribution parameters. Eqs. 48-51 remain the same as before. For this design variable case, all required information for the sensitivities is available from the MPP search.

Since Eqs. 46-53 are derived using the Karush-Kuhn-Tucker optimality conditions for a converged MPP, they are appropriate for RBDO using AMV+, AMV²+, TANA, FORM, and SORM, but not for RBDO using MVFOSM, MVSOSM, AMV, or AMV².

3.2. SEQUENTIAL/SURROGATE-BASED RBDO

An alternative RBDO approach is the sequential approach, in which additional efficiency is sought through breaking the nested relationship of the MPP and design searches. The general concept is to iterate between optimization and uncertainty quantification, updating the optimization goals based on the most recent probabilistic assessment results. This update may be based on safety factors (Wu et al., 2001) or other approximations (Du and Chen, 2004).

A particularly effective approach for updating the optimization goals is to use the $p/\beta/z$ sensitivity analysis of Eqs. 46-53 in combination with local surrogate models (Zou et al., 2004). In (Eldred et al., 2005) and (Eldred et al., 2006), first-order and second-order Taylor series approximations were employed within a trust-region model management framework (Giunta and Eldred, 2000) in order to adaptively manage the extent of the approximations and ensure convergence of the RBDO process. Surrogate models were used for both the objective function and the constraints, although the use of constraint surrogates alone is sufficient to remove the nesting.

In particular, RIA trust-region surrogate-based RBDO employs surrogate models of f and p/β within a trust region Δ^k centered at \mathbf{d}_c . For first-order surrogates:

$$\begin{aligned} & \text{minimize} && f(\mathbf{d}_c) + \nabla_{\mathbf{d}} f(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) \\ & \text{subject to} && \beta(\mathbf{d}_c) + \nabla_{\mathbf{d}} \beta(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) \geq \bar{\beta} \\ & && \text{or } p(\mathbf{d}_c) + \nabla_{\mathbf{d}} p(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) \leq \bar{p} \\ & && \|\mathbf{d} - \mathbf{d}_c\|_{\infty} \leq \Delta^k \end{aligned} \quad (54)$$

and for second-order surrogates:

$$\begin{aligned} & \text{minimize} && f(\mathbf{d}_c) + \nabla_{\mathbf{d}} f(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) + \frac{1}{2} (\mathbf{d} - \mathbf{d}_c)^T \nabla_{\mathbf{d}}^2 f(\mathbf{d}_c) (\mathbf{d} - \mathbf{d}_c) \\ & \text{subject to} && \beta(\mathbf{d}_c) + \nabla_{\mathbf{d}} \beta(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) + \frac{1}{2} (\mathbf{d} - \mathbf{d}_c)^T \nabla_{\mathbf{d}}^2 \beta(\mathbf{d}_c) (\mathbf{d} - \mathbf{d}_c) \geq \bar{\beta} \\ & && \text{or } p(\mathbf{d}_c) + \nabla_{\mathbf{d}} p(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) + \frac{1}{2} (\mathbf{d} - \mathbf{d}_c)^T \nabla_{\mathbf{d}}^2 p(\mathbf{d}_c) (\mathbf{d} - \mathbf{d}_c) \leq \bar{p} \\ & && \|\mathbf{d} - \mathbf{d}_c\|_{\infty} \leq \Delta^k \end{aligned} \quad (55)$$

For PMA trust-region surrogate-based RBDO, surrogate models of f and z are employed within a trust region Δ^k centered at \mathbf{d}_c . For first-order surrogates:

$$\begin{aligned} & \text{minimize} && f(\mathbf{d}_c) + \nabla_{\mathbf{d}} f(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) \\ & \text{subject to} && z + \nabla_{\mathbf{d}} z(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) \geq \bar{z} \\ & && \|\mathbf{d} - \mathbf{d}_c\|_{\infty} \leq \Delta^k \end{aligned} \quad (56)$$

and for second-order surrogates:

$$\begin{aligned} & \text{minimize} && f(\mathbf{d}_c) + \nabla_{\mathbf{d}} f(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) + \frac{1}{2} (\mathbf{d} - \mathbf{d}_c)^T \nabla_{\mathbf{d}}^2 f(\mathbf{d}_c) (\mathbf{d} - \mathbf{d}_c) \\ & \text{subject to} && z + \nabla_{\mathbf{d}} z(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) + \frac{1}{2} (\mathbf{d} - \mathbf{d}_c)^T \nabla_{\mathbf{d}}^2 z(\mathbf{d}_c) (\mathbf{d} - \mathbf{d}_c) \geq \bar{z} \\ & && \|\mathbf{d} - \mathbf{d}_c\|_{\infty} \leq \Delta^k \end{aligned} \quad (57)$$

where the sense of the z constraint may vary as described previously. The second-order information in Eqs. 55 and 57 will typically be approximated with quasi-Newton updates.

4. Benchmark Problems

(Eldred et al., 2005) and (Eldred et al., 2006) have examined the performance of first and second-order reliability analysis and design methods for four analytic benchmark test problems: lognormal ratio, short column, cantilever beam, and steel column.

4.1. RELIABILITY ANALYSIS RESULTS

Within the reliability analysis algorithms, various limit state approximation (MVFOSM, MVSOSM, x-/u-space AMV, x-/u-space AMV², x-/u-space AMV+, x-/u-space AMV²+, x-/u-space TANA, FORM, and SORM), probability integration (first-order or second-order), warm starting, Hessian

Table I. RIA results for short column problem.

RIA Approach	SQP Function Evaluations	NIP Function Evaluations	CDF p Error Norm	Target z Offset Norm
MVFOSM	1	1	0.1548	0.0
MVSOSM	1	1	0.1127	0.0
x-space AMV	45	45	0.009275	18.28
u-space AMV	45	45	0.006408	18.81
x-space AMV ²	45	45	0.002063	2.482
u-space AMV ²	45	45	0.001410	2.031
x-space AMV+	192	192	0.0	0.0
u-space AMV+	207	207	0.0	0.0
x-space AMV ² +	125	131	0.0	0.0
u-space AMV ² +	122	130	0.0	0.0
x-space TANA	245	246	0.0	0.0
u-space TANA	296*	278*	6.982e-5	0.08014
FORM	626	176	0.0	0.0
SORM	669	219	0.0	0.0

approximation (finite difference, BFGS, or SR1), and MPP optimization algorithm (SQP or NIP) selections have been investigated. A sample comparison of reliability analysis performance, taken from the short column example, is shown in Tables I and II for RIA and PMA analysis, respectively, where “*” indicates that one or more levels failed to converge. Consistent with the employed probability integrations, the error norms are measured with respect to fully-converged first-order results for MV, AMV, AMV², AMV+, and FORM methods, and with respect to fully-converged second-order results for AMV²+, TANA, and SORM methods. Also, it is important to note that the simple metric of “function evaluations” is imperfect, and (Eldred et al., 2006) provides more detailed reporting of individual response value, gradient, and Hessian evaluations.

Overall, reliability analysis results for the lognormal ratio, short column, and cantilever test problems indicate several trends. MVFOSM, MVSOSM, AMV, and AMV² are significantly less expensive than the fully-converged MPP methods, but come with corresponding reductions in accuracy. In combination, these methods provide a useful spectrum of accuracy and expense that allow the computational effort to be balanced with the statistical precision required for particular applications. In addition, support for forward and inverse mappings (RIA and PMA) provide the flexibility to support different UQ analysis needs.

Relative to FORM and SORM, AMV+ and AMV²+ has been shown to have equal accuracy and consistent computational savings. For second-order PMA analysis with prescribed probability levels, AMV²+ has additionally been shown to be more robust due to its ability to better manage $\bar{\beta}$ updates. Analytic Hessians were highly effective in AMV²+, but since they are often unavailable in practical applications, finite-difference numerical Hessians and quasi-Newton Hessian approximations were also demonstrated, with SR1 quasi-Newton updates being shown to be sufficiently

Table II. PMA results for short column problem.

PMA Approach	SQP Function Evaluations	NIP Function Evaluations	CDF z Error Norm	Target p Offset Norm
MVFOSM	1	1	7.454	0.0
MVSOSM	1	1	6.823	0.0
x-space AMV	45	45	0.9420	0.0
u-space AMV	45	45	0.5828	0.0
x-space AMV ²	45	45	2.730	0.0
u-space AMV ²	45	45	2.828	0.0
x-space AMV+	171	179	0.0	0.0
u-space AMV+	205	205	0.0	0.0
x-space AMV ² +	135	142	0.0	0.0
u-space AMV ² +	132	139	0.0	0.0
x-space TANA	293*	272	0.04259	1.598e-4
u-space TANA	325*	311*	2.208	5.600e-4
FORM	720	192	0.0	0.0
SORM	535	191*	2.410	6.522e-4

accurate and competitive with analytic Hessian performance. Relative to first-order AMV+ performance, AMV²+ with analytic Hessians had consistently superior efficiency, and AMV²+ with quasi-Newton Hessians had improved performance in most cases (it was more expensive than AMV+ only when a more challenging second-order \bar{p} problem was being solved). In general, second-order reliability analyses appear to serve multiple synergistic needs. The same Hessian information that allows for more accurate probability integrations can also be applied to making MPP solutions more efficient and more robust. Conversely, limit state curvature information accumulated during an MPP search can be reused to improve the accuracy of probability estimates.

For nonapproximated limit states (FORM and SORM), NIP optimizers have shown promise in being less susceptible to PMA u-space excursions and in being more efficient than SQP optimizers in most cases. Warm starting with projections has been shown to be consistently effective for reliability analyses, with typical savings on the order of 25%. The x-space and u-space linearizations for AMV, AMV², AMV+, AMV²+, and TANA were both effective, and the relative performance was strongly problem-dependent (u-space was more efficient for lognormal ratio, x-space was more efficient for short column, and x-space and u-space were equivalent for cantilever). Among all combinations tested, AMV²+ (with analytic Hessians if available, or SR1 Hessians if not) is the recommended approach.

An important question is how Taylor-series based limit state approximations (such as AMV+ and AMV²+) can frequently outperform the best general-purpose optimizers (such as SQP and NIP). The answer likely lies in the exploitation of the structure of the RIA and PMA MPP problems. By approximating the limit state but retaining $\mathbf{u}^T \mathbf{u}$ explicitly in Eqs. 16 and 17, specific problem structure knowledge is utilized in formulating a mixed surrogate/direct approach.

Table III. Analytic bi-level RBDO results, short column test problem.

RBDO Approach	Function Evaluations	Objective Function	Constraint Violation
RIA $\bar{z} \rightarrow p$ x-space AMV+	149	217.1	0.0
RIA $\bar{z} \rightarrow p$ x-space AMV ² +	129	217.1	0.0
RIA $\bar{z} \rightarrow p$ FORM	911	217.1	0.0
RIA $\bar{z} \rightarrow p$ SORM	1204	217.1	0.0
RIA $\bar{z} \rightarrow \beta$ x-space AMV+	72	216.7	0.0
RIA $\bar{z} \rightarrow \beta$ x-space AMV ² +	67	216.7	0.0
RIA $\bar{z} \rightarrow \beta$ FORM	612	216.7	0.0
RIA $\bar{z} \rightarrow \beta$ SORM	601	216.7	0.0
PMA $\bar{p}, \bar{\beta} \rightarrow z$ x-space AMV+	100	216.8	0.0
PMA $\bar{p} \rightarrow z$ x-space AMV ² +	98	216.8	0.0
PMA $\bar{\beta} \rightarrow z$ x-space AMV ² +	98	216.8	0.0
PMA $\bar{p}, \bar{\beta} \rightarrow z$ FORM	285	216.8	0.0
PMA $\bar{p} \rightarrow z$ SORM	306	217.2	0.0
PMA $\bar{\beta} \rightarrow z$ SORM	329	216.8	0.0

4.2. RBDO RESULTS

These reliability analysis capabilities provide a substantial foundation for RBDO formulations, and bi-level and sequential RBDO approaches have been investigated. Both approaches have utilized analytic gradients for z , β , and p with respect to augmented and inserted design variables, and sequential RBDO has additionally utilized a trust-region surrogate-based approach to manage the extent of the Taylor-series approximations. A sample comparison of RBDO performance, taken again from the short column example, is shown in Tables III and IV for bi-level and sequential surrogate-based RBDO, respectively.

Overall, RBDO results for the short column, cantilever, and steel column test problems build on the reliability analysis trends. Basic first-order bi-level RBDO has been evaluated with up to 18 variants (RIA/PMA with different $p/\beta/z$ mappings for MV, x-/u-space AMV, x-/u-space AMV+, and FORM), and fully-analytic bi-level and sequential RBDO have each been evaluated with up to 21 variants (RIA/PMA with different $p/\beta/z$ mappings for x-/u-space AMV+, x-/u-space AMV²+, FORM, and SORM). Bi-level RBDO with MV and AMV are inexpensive but give only approximate optima. These approaches may be useful for preliminary design or for warm-starting other RBDO methods. Bi-level RBDO with AMV+ was shown to have equal accuracy and robustness to bi-level FORM-based approaches and be significantly less expensive on average. In addition, usage of β in RIA RBDO constraints was preferred due to it being more well-behaved and more well-scaled than constraints on p . Warm starts in RBDO were most effective when the design changes were small, with the most benefit for basic bi-level RBDO (with numerical differencing at the design level), decreasing to marginal effectiveness for fully-analytic bi-level RBDO and to relative ineffectiveness

Table IV. Surrogate-based RBDO results, short column test problem.

RBDO Approach	Function Evaluations	Objective Function	Constraint Violation
RIA $\bar{z} \rightarrow p$ x-space AMV+	75	216.9	0.0
RIA $\bar{z} \rightarrow p$ x-space AMV ² +	86	218.7	0.0
RIA $\bar{z} \rightarrow p$ FORM	577	216.9	0.0
RIA $\bar{z} \rightarrow p$ SORM	718	216.5	1.110e-4
RIA $\bar{z} \rightarrow \beta$ x-space AMV+	65	216.7	0.0
RIA $\bar{z} \rightarrow \beta$ x-space AMV ² +	51	216.7	0.0
RIA $\bar{z} \rightarrow \beta$ FORM	561	216.7	0.0
RIA $\bar{z} \rightarrow \beta$ SORM	560	216.7	0.0
PMA $\bar{p}, \bar{\beta} \rightarrow z$ x-space AMV+	76	216.7	2.1e-4
PMA $\bar{p} \rightarrow z$ x-space AMV ² +	58	216.8	0.0
PMA $\bar{\beta} \rightarrow z$ x-space AMV ² +	79	216.8	0.0
PMA $\bar{p}, \bar{\beta} \rightarrow z$ FORM	228	216.7	2.1e-4
PMA $\bar{p} \rightarrow z$ SORM	128	217.2	0.0
PMA $\bar{\beta} \rightarrow z$ SORM	171	216.8	0.0

for sequential RBDO. However, large design changes were desirable for overall RBDO efficiency and, compared to basic bi-level RBDO, fully-analytic RBDO and sequential RBDO were clearly superior.

In second-order bi-level and sequential RBDO, the AMV²+ approaches were consistently more efficient than the SORM-based approaches. In general, sequential RBDO approaches demonstrated consistent computational savings over the corresponding bi-level RBDO approaches, and the combination of sequential RBDO using AMV²+ was the most effective of all of the approaches. With initial trust region size tuning, sequential RBDO computational expense for these test problems was shown to be as low as approximately 40 function evaluations per limit state (35 for a single limit state in short column, 75 for two limit states in cantilever, and 45 for a single limit state in steel column). Finally, second-order RBDO with probability constraints was shown to be more challenging and expensive, but could be more precise in achieving the desired probabilistic performance.

5. Application to MEMS

In this section, we consider the application of DAKOTA's reliability algorithms to the design of micro-electro-mechanical systems (MEMS). In particular, we summarize initial results for one of the applications described in (Adams et al., 2006). These application studies provide essential feedback on the performance of algorithms for real-world design applications, which may contain computational challenges not well-represented in analytically defined test problems.

Pre-fabrication design optimization of microelectromechanical systems (MEMS) is an important emerging application of uncertainty quantification and reliability-based design optimization. Typically crafted of silicon, polymers, metals, or a combination thereof, MEMS serve as micro-scale sensors, actuators, switches, and machines with applications including robotics, biology and medicine, automobiles, RF electronics, and optical displays (Allen, 2005). Design optimization of these devices is crucial since fabrication costs, even for prototypes, can be prohibitive. There is considerable uncertainty in the micromachining and etching processes used to manufacture MEMS and consequently in the behavior of the finished products. RBDO coupled with computational mechanics models of MEMS offers a means to quantify this uncertainty and determine a priori the most reliable and/or robust design that meets performance criteria.

Of particular interest is the design of MEMS bistable mechanisms which toggle between two stable positions, making them useful as micro switches, relays, and nonvolatile memory. We focus on shape optimization of compliant bistable mechanisms, where instead of mechanical joints, material elasticity enables the bistability of the mechanism (Jensen et al., 2001). Figure 1 contains an electron micrograph of a MEMS compliant bistable mechanism in one of its stable positions. One achieves transfer between stable states by applying force to the center shuttle of the device via an electrostatic actuator, heat source, or other means to cause the flexible “legs” (horizontal beams) of the system to buckle through their instability and relax toward the other stable equilibrium.

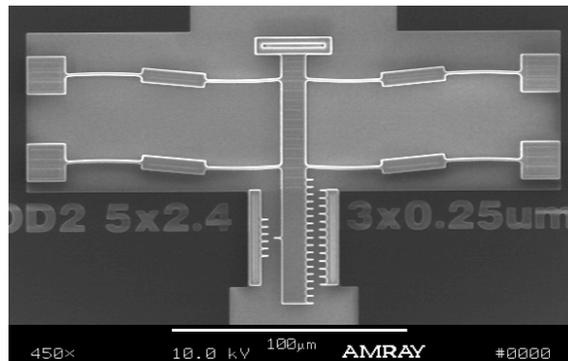


Figure 1. Electron micrograph of MEMS bistable mechanism. Source: J.W. Wittwer, Ph.D. dissertation.

Successful bistable switch actuation in this manner depends on the relationship between force and vertical displacement for the manufactured switch. In Figure 2 we present a schematic of a typical force–displacement curve for a bistable mechanism. The switch characterized by this curve has three equilibria: E_1 and E_3 are stable equilibria whereas E_2 is an unstable equilibrium (arrows indicate stability). A device with such a force–displacement curve could be used as a switch or actuator by setting it to position E_3 as shown in Figure 1 (requiring large force F_{max}) and then actuating by applying the small force F_{min} in the opposite direction to transfer through E_2 toward the equilibrium E_1 . One could utilize this force profile to complete a circuit by placing a switch contact near the displaced position corresponding to maximum (closure) force as illustrated in Figure 2.

The design considered in this work is similar to the electron micrograph in Figure 1, for which design optimization has been considered in (Jensen et al., 2001) and design under uncertainty

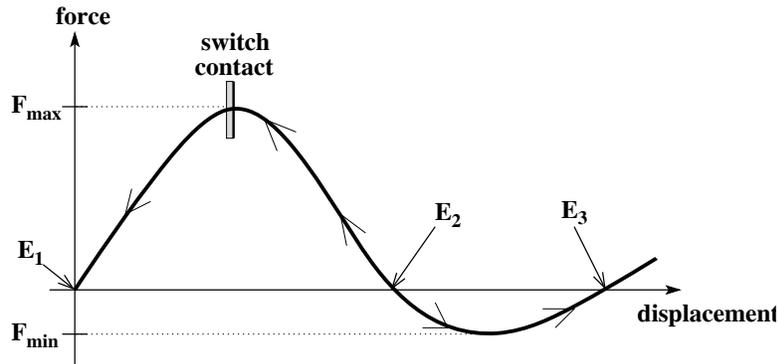


Figure 2. Schematic of force–displacement curve for bistable MEMS mechanism. Arrows indicate stability.

with mean value methods has been investigated in (Wittwer, 2005; Wittwer, 2006). The primary structural difference in (Adams et al., 2006) is in the shape of the legs, and Figure 3 shows a detail of the design of one of these legs.

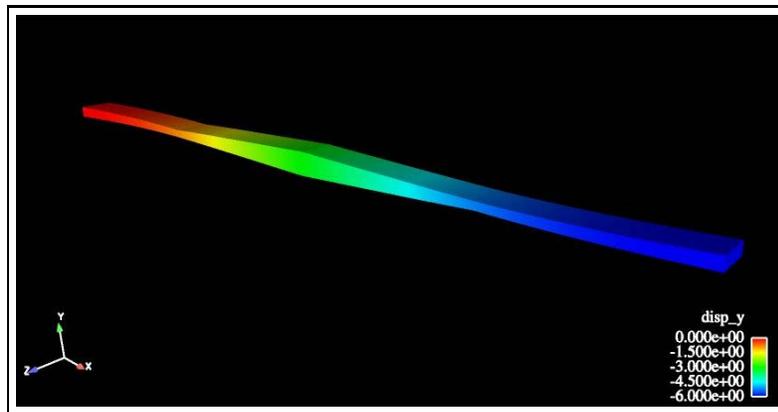


Figure 3. Sample of tapered beam leg for bistable mechanism.

The design criteria used for this bistable switch include

- minimize the magnitude of the force F_{min} required to actuate the switch (drive F_{min} toward zero), while maintaining its bistability ($F_{max} > 0$, $F_{min} < 0$)
- at least $50\mu N$ force at switch contact (to reliably attain closure), but no more than $150\mu N$ (to avoid contact damage)
- point of instability E_2 no more than $8\mu m$
- maximum stress no more than 1200 MPa

The force-displacement profile of bistable MEMS devices is highly sensitive to design geometry, so one can vary manufactured geometry in order to achieve various design criteria. However, due to

manufacturing processes, fabricated geometry can deviate significantly from design-specified beam geometry. As a consequence of photo masks used in the process, fabricated in-plane geometry edges (contributing to widths and lengths) are $0.1 \pm 0.08 \mu m$ less than specified. Uncertainty in the manufactured geometry can lead to substantial uncertainty in the positions of the stable equilibria and in the maximum and minimum force on the force–displacement curve. The manufactured thickness of the device is also uncertain, though this does not contribute as much to variability in the force–displacement behavior. Uncertain material properties such as Young’s modulus and residual stress also influence the characteristics of the fabricated beam. For this application, we consider two uncertain variables: ΔW (edge bias on beam widths, which yields effective manufactured widths of $W_i + \Delta W, i = 0, \dots, n$) and S_r (residual stress in the manufactured device), with distributions shown in Table V.

Table V. Uncertain variables used in RBDO.

variable	mean	std. dev.	distribution
Δw	$-0.2 \mu m$	0.08	normal
S_r	-11 Mpa	4.13	normal

Given 13 geometric design variables \mathbf{d} describing lengths, widths, and orientations of the legs and the two specified uncertain variables \mathbf{x} , we perform a reliability-based design optimization to compute a design that is reliably bistable, but requires minimum force to actuate. The limit state for this problem is

$$g(\mathbf{x}) = F_{min}(\mathbf{x}) \quad (58)$$

and we define failure to be lack of bistability ($F_{min} \geq 0$) and require a reliability index $\beta_{ccdf} \geq 2$. The RBDO problem utilizes an RIA $\bar{z} \rightarrow \beta$ approach:

$$\begin{aligned} \max & & F_{min}(\mathbf{d}, \mathbf{x}) \\ \text{s.t.} & & 2 \leq \beta_{ccdf}(\mathbf{d}, \mathbf{x}) \\ & & 50 \leq F_{max}(\mathbf{d}, \mathbf{x}) \leq 150 \\ & & E_2(\mathbf{d}, \mathbf{x}) \leq 8 \\ & & S_{max}(\mathbf{d}, \mathbf{x}) \leq 1200 \end{aligned} \quad (59)$$

although a PMA $\bar{\beta} \rightarrow z$ approach could also be used. The use of the F_{min} metric in both the objective function and the reliability constraint results in a powerful problem formulation since, in addition to yielding a design with specified reliability, it also produces a robust design. By forcing F_{min} toward zero while requiring two standard deviations of surety, the optimization problem favors designs with less variability in F_{min} . This renders the design performance less sensitive to the uncertainties in the problem.

We solve the optimization problem by applying DAKOTA’s bi-level RBDO approach in combination with mesh generation using CUBIT and finite element analysis using Adagio. Adagio is a quasi-static nonlinear mechanics code, implemented in Sandia National Laboratories’ SIERRA framework of multiphysics codes (Edwards, 2004), that is used to simulate the elastic deformation of

the device through discrete displacement steps to produce a force–displacement curve. We compare three reliability analysis methods for this MEMS application: (1) MVFOSM (no MPP search), (2) AMV+, and (3) FORM. The latter two are advantaged by their ability to provide (semi)analytic derivatives of reliability metrics with respect to design variables for the optimizer (see Section 3.1), whereas the former is much less expensive per reliability analysis but must resort to numerical design derivatives due to the use of σ_g (analytic derivatives of Eq. 2 with respect to \mathbf{d} are impractical to evaluate).

Results for the three methods are presented in Table VI and the optimal force–displacement curves are shown in Figure 4. Optimization with MVFOSM offers substantial improvement over the initial design, yielding a design with a substantially smaller minimum force and tighter reliability constraint β . However, since mean value analyses estimate reliability based solely on evaluations at the means of the uncertain variables, they can yield inaccurate reliability metrics in cases of nonlinearity or nonnormality. In this example, the actual verified reliability of the optimal MVFOSM-based design is only 1.75, less than the prescribed reliability of $\beta = 2$. The optimal designs for the AMV+ and FORM-based RBDO methods were indistinguishable from each other, but relative to MVFOSM-based RBDO, yield a more conservative value of F_{min} due to the improved estimation of β . In each of the three cases, the variability in F_{min} has been reduced from approximately 5.7 to 4.6 μN per (verified) input standard deviation, resulting in designs that are less sensitive to the input uncertainties.

Table VI. RBDO results (MVFOSM and first-order MPP methods) for MEMS bistable mechanism.

lower bound	RBDO metric	upper bound	MVFOSM initial	MVFOSM optimal	AMV+/FORM initial	AMV+/FORM optimal
	F_{min} (μN)		-23.03	-8.08	-23.03	-9.37
2	β		5.66	2.00	4.02	2.00
50	F_{max} (μN)	150	67.35	50.0	67.35	50.0
	E_2 (μm)	8	4.06	3.85	4.06	3.76
	S_{max} (MPa)	1200	396	313	396	323
	Verified β		4.02	1.75		

In Figure 5, we see the results of parameter studies for the metric $F_{min}(\mathbf{d}, \mathbf{x})$ as a function of the uncertain variables \mathbf{x} for two different sets of design variables \mathbf{d} . Since the uncertain variables are both normal, the transformation to u-space used by AMV+ and FORM is linear. The former design variable set corresponds to the optimal values obtained from MVFOSM-based RBDO, and in this case the limit state is relatively linear and well-behaved in the range of interest. First-order probability integrations should be sufficiently accurate. For the second design variable set, however, multiple computational challenges are evident. In this case, the limit state has significant nonlinearity (requiring more sophisticated probability integrations) and its simulation can be seen to be unreliable in the left tail of the edge bias (resulting from too flimsy a structure). This highlights a number of difficulties common in engineering applications: highly nonlinear limit states, nonsmooth and multimodal limit states, and simulation failures caused by, e.g., evaluations in the tails of input

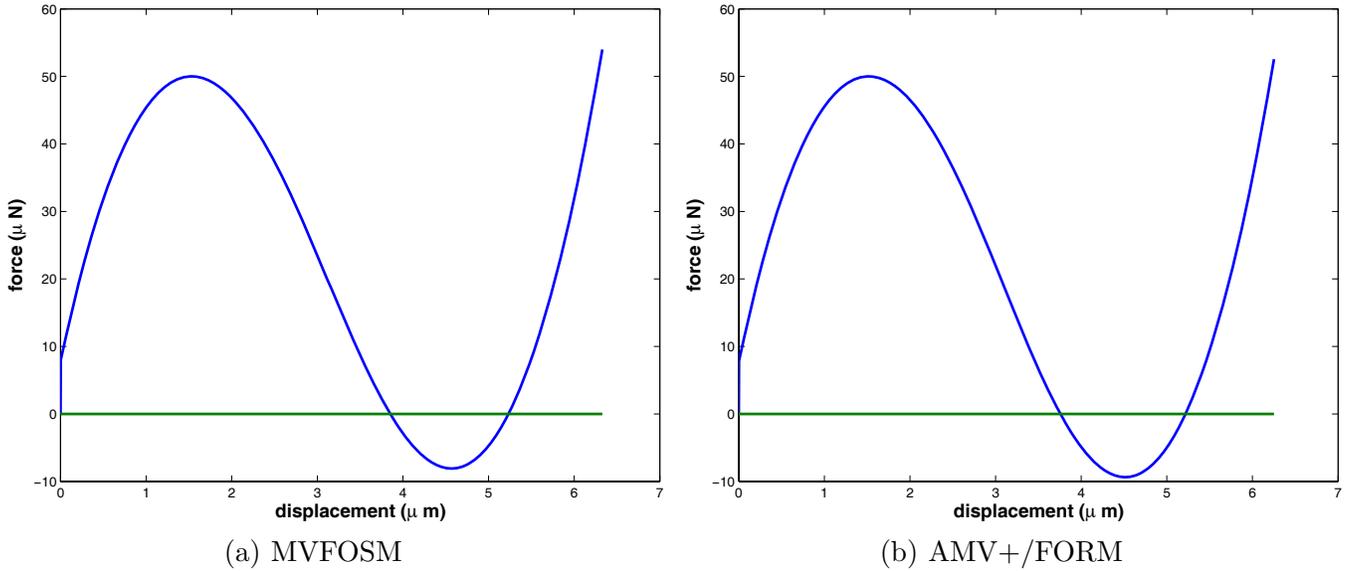


Figure 4. Optimal force-displacement curves resulting from RBDO of MEMS bistable mechanism.

distributions. These difficulties must be mitigated through a combination of algorithm research, problem formulation, and simulation refinement.

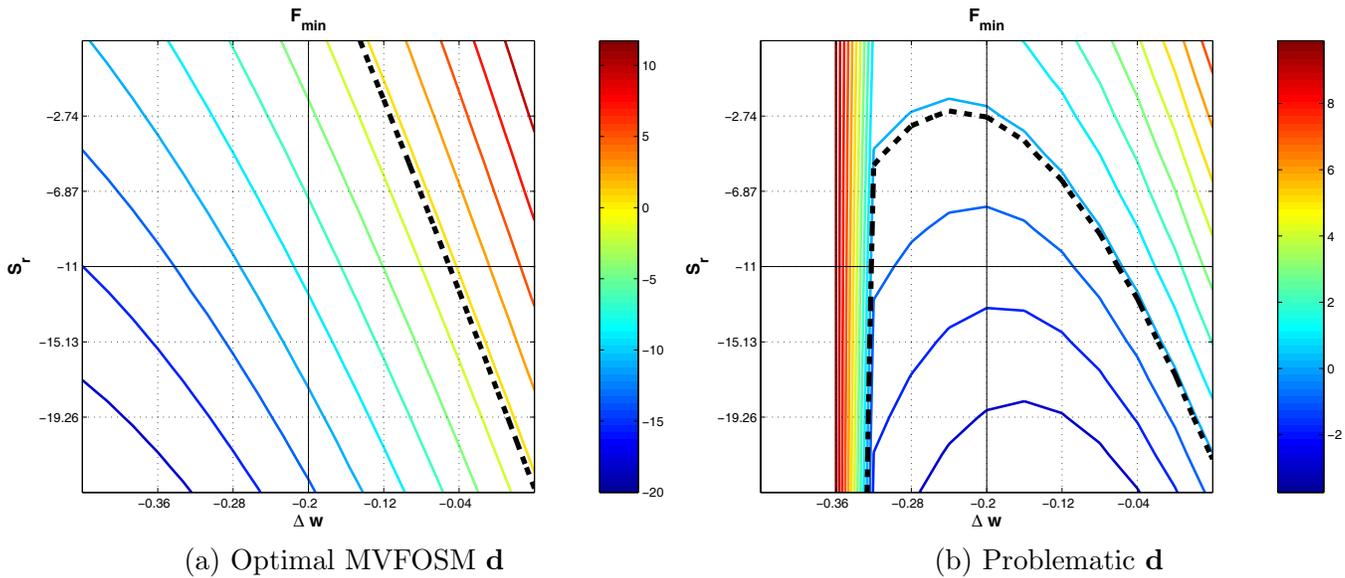


Figure 5. Contour plot of $F_{min}(\mathbf{d}, \mathbf{x})$ as a function of uncertain variables for different design variable sets. Dashed line shows where limit state $F_{min} = 0$.

6. Conclusions

This paper has overviewed recent algorithm research in first and second-order reliability methods. A number of algorithmic variations have been presented, and the effect of different limit state approximations, probability integrations, warm starting, most probable point search algorithms, and Hessian approximations has been discussed. These reliability analysis capabilities provide the foundation for reliability-based design optimization (RBDO) methods, and bi-level and sequential formulations have been presented. These RBDO formulations employ analytic sensitivities of reliability metrics with respect to design variables that either augment or define distribution parameters for the uncertain variables.

Relative performance of these reliability analysis and design algorithms has been measured for a number of benchmark test problems using the DAKOTA software. The most effective techniques in these computational experiments have been AMV²+ for reliability analysis and sequential/surrogate-based approaches for RBDO. Continuing efforts in algorithm research will build on these successful methods through investigation of sequential RBDO with mixed surrogate and direct models (for probabilistic and deterministic components, respectively) and second-order RIA RBDO formulations employing generalized reliability indices.

These reliability analysis and design algorithms are now being applied to real-world applications in the shape optimization of micro-electro-mechanical systems, and initial experiences with this deployment are presented. Issues identified in deploying reliability methods to complex engineering applications include highly nonlinear, nonsmooth/noisy, and multimodal limit states, and potential simulation failures when evaluating parameter sets in the tails of input distributions. To mitigate these difficulties, a combination of continuing algorithm research, enhancements in problem formulation, and refinements to modeling and simulation capabilities is recommended.

7. Acknowledgments

The authors would like to express their thanks to Jonathan Wittwer and Jordan Massad for their assistance in formulating design problems for MEMS and to the Sandia Computer Science Research Institute (CSRI) for support of this collaborative work between Sandia National Laboratories and Vanderbilt University.

References

- Adams, B.M., Eldred, M.S., Wittwer, J.W., and Massad, J.E., Reliability-Based Design Optimization for Shape Design of Compliant Micro-Electro-Mechanical Systems, abstract for *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, VA, Sept. 6-8, 2006.
- Agarwal, H., Renaud, J.E., Lee, J.C., and Watson, L.T., A Unilevel Method for Reliability Based Design Optimization, paper AIAA-2004-2029 in *Proceedings of the 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Palm Springs, CA, April 19-22, 2004.
- Allen, J.J., Micro Electro Mechanical System Design, Taylor and Francis, Boca Raton, 2005.
- Allen, M. and Maute, K., Reliability-based design optimization of aeroelastic structures, *Struct. Multidiscip. O.*, Vol. 27, 2004, pp. 228-242.

- Box, G.E.P. and Cox, D.R., An Analysis of Transformations, *J. Royal Stat. Soc.*, Series B, Vol. 26, 1964, pp. 211-252.
- Breitung, K., Asymptotic approximation for multinormal integrals, *J. Eng. Mech., ASCE*, Vol. 110, No. 3, 1984, pp. 357-366.
- Chen, X., and Lind, N.C., Fast Probability Integration by Three-Parameter Normal Tail Approximation, *Struct. Saf.*, Vol. 1, 1983, pp. 269-276.
- Der Kiureghian, A. and Liu, P.L., Structural Reliability Under Incomplete Probability Information, *J. Eng. Mech., ASCE*, Vol. 112, No. 1, 1986, pp. 85-104.
- Du, X. and Chen, W., Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design, *J. Mech. Design*, Vol. 126, 2004, pp.225-233.
- Edwards, H.C., Sierra Framework for Massively Parallel Adaptive Multiphysics Applications Sandia Technical Report SAND2004-6277C, April 2003, Sandia National Laboratories, July 2005, Albuquerque, NM.
- Eldred, M.S., Giunta, A.A., Wojtkiewicz, S.F., Jr., and Trucano, T.G., Formulations for Surrogate-Based Optimization Under Uncertainty, paper AIAA-2002-5585 in *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA, Sept. 4-6, 2002.
- Eldred, M.S., Giunta, A.A., van Bloemen Waanders, B.G., Wojtkiewicz, S.F., Jr., Hart, W.E., and Alleva, M.P., DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis. Version 3.1 Users Manual. Sandia Technical Report SAND2001-3796, Revised April 2003, Sandia National Laboratories, Albuquerque, NM.
- Eldred, M.S., Agarwal, H., Perez, V.M., Wojtkiewicz, S.F., Jr., and Renaud, J.E., Investigation of Reliability Method Formulations in DAKOTA/UQ, (to appear) *Structure & Infrastructure Engineering: Maintenance, Management, Life-Cycle Design & Performance*, Taylor & Francis Group.
- Eldred, M.S., Bichon, B.J., and Wojtkiewicz, S.F., Jr., Second-Order Reliability Formulations in DAKOTA/UQ, (in review) *Structure & Infrastructure Engineering: Maintenance, Management, Life-Cycle Design & Performance*, special issue on reliability analysis in aerospace systems, Taylor & Francis Group.
- Fadel, G.M., Riley, M.F., and Barthelemy, J.-F.M., Two Point Exponential Approximation Method for Structural Optimization, *Structural Optimization*, Vol. 2, No. 2, 1990, pp. 117-124.
- Gill, P.E., Murray, W., Saunders, M.A., and Wright, M.H., User's Guide for NPSOL 5.0: A Fortran Package for Non-linear Programming, System Optimization Laboratory, Technical Report SOL 86-1, Revised July 1998, Stanford University, Stanford, CA.
- Giunta, A.A. and Eldred, M.S., Implementation of a Trust Region Model Management Strategy in the DAKOTA Optimization Toolkit, paper AIAA-2000-4935 in *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA, September 6-8, 2000.
- Haldar, A. and Mahadevan, S., Probability, Reliability, and Statistical Methods in Engineering Design, 2000 (Wiley: New York).
- Hohenbichler, M. and Rackwitz, R., Sensitivity and importance measures in structural reliability, *Civil Eng. Syst.*, Vol. 3, 1986, pp. 203-209.
- Hohenbichler, M. and Rackwitz, R., Improvement of second-order reliability estimates by importance sampling, *J. Eng. Mech., ASCE*, Vol. 114, No. 12, 1988, pp. 2195-2199.
- Hong, H.P., Simple Approximations for Improving Second-Order Reliability Estimates, *J. Eng. Mech., ASCE*, Vol. 125, No. 5, 1999, pp. 592-595.
- Jensen, B.D., Parkinson, M.B., Kurabayashi, K., Kowell, L.L., and Baker, M.S., Design Optimization of a Fully-Compliant Bistable Micro-Mechanism, *Proc. 2001 ASME Intl. Mech. Eng. Congress and Exposition*, New York, NY, Nov. 11-16, 2001.
- Karamchandani, A. and Cornell, C.A., Sensitivity estimation within first and second order reliability methods, *Struct. Saf.*, Vol. 11, 1992, pp. 95-107.
- Kuschel, N. and Rackwitz, R., Two Basic Problems in Reliability-Based Structural Optimization, *Math. Method Oper. Res.*, Vol. 46, 1997, pp.309-333.
- Meza, J.C., OPT++: An Object-Oriented Class Library for Nonlinear Optimization, Sandia Technical Report SAND94-8225, Sandia National Laboratories, Livermore, CA, March 1994.
- Nocedal, J., and Wright, S.J., Numerical Optimization, Springer, New York, 1999.

- Rackwitz, R., and Fiessler, B., Structural Reliability under Combined Random Load Sequences, *Comput. Struct.*, Vol. 9, 1978, pp. 489-494.
- Rackwitz, R., Optimization and risk acceptability based on the Life Quality Index, *Struct. Saf.*, Vol. 24, 2002, pp. 297-331.
- Rosenblatt, M., Remarks on a Multivariate Transformation, *Ann. Math. Stat.*, Vol. 23, No. 3, 1952, pp. 470-472.
- Sues, R., Aminpour, M. and Shin, Y., Reliability-Based Multidisciplinary Optimization for Aerospace Systems, paper AIAA-2001-1521 in *Proceedings of the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Seattle, WA, April 16-19, 2001.
- Tu, J., Choi, K.K., and Park, Y.H., A New Study on Reliability-Based Design Optimization, *J. Mech. Design*, Vol. 121, 1999, pp.557-564.
- Wang, L. and Grandhi, R.V., Efficient Safety Index Calculation for Structural Reliability Analysis, *Comput. Struct.*, Vol. 52, No. 1, 1994, pp. 103-111.
- Wittwer, J.W., Simulation-Based Design Under Uncertainty for Compliant Microelectromechanical Systems, Ph.D. dissertation, Brigham Young University, April, 2005.
- Wittwer, J.W., Baker, M.S., and Howell, L.L., Robust design and model validation of nonlinear compliant micromechanisms, *J. Microelectromechanical Sys.*, Vol. 15, No. 1, 2006, *to appear*.
- Wojtkiewicz, S.F., Jr., Eldred, M.S., Field, R.V., Jr., Urbina, A., and Red-Horse, J.R., A Toolkit For Uncertainty Quantification In Large Computational Engineering Models, paper AIAA-2001-1455 in *Proceedings of the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Seattle, WA, April 16-19, 2001.
- Wu, Y.-T., and Wirsching, P.H., A new algorithm for structural reliability estimation, *J. Eng. Mech., ASCE*, Vol. 113, 1987, pp. 1319-1336.
- Wu, Y.-T., Millwater, H.R., and Cruse, T.A., Advanced Probabilistic Structural Analysis Method for Implicit Performance Functions, *AIAA J.*, Vol. 28, No. 9, 1990, pp. 1663-1669.
- Wu, Y.-T., Computational Methods for Efficient Structural Reliability and Reliability Sensitivity Analysis, *AIAA J.*, Vol. 32, No. 8, 1994, pp. 1717-1723.
- Wu, Y.-T., Shin, Y., Sues, R., and Cesare, M., Safety-Factor Based Approach for Probability-Based Design Optimization, paper AIAA-2001-1522 in *Proceedings of the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Seattle, WA, April 16-19, 2001.
- Xu, S., and Grandhi, R.V., Effective Two-Point Function Approximation for Design Optimization, *AIAA J.*, Vol. 36, No. 12, 1998, pp. 2269-2275.
- Zou, T., Mahadevan, S., and Rebba, R., Computational Efficiency in Reliability-Based Optimization, *Proceedings of the 9th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability*, Albuquerque, NM, July 26-28, 2004.

Semantic Tolerance Modeling based on Modal Interval Analysis

Yan Wang

NSF Center for e-Design, University of Central Florida
wangyan@mail.ucf.edu

Abstract: A significant amount of research efforts has been given to explore the mathematical basis for 3D dimensional and geometric tolerance representation, analysis, and synthesis. However, engineering semantics is not maintained in these mathematic models. It is hard to interpret calculated numerical results in a meaningful way. In this paper, a new semantic tolerance modeling scheme based on modal interval is proposed to improve interpretability of tolerance modeling. With logical quantifiers, semantic relations between tolerance specifications and implications of tolerance stacking are embedded in the mathematic model. The model captures the semantics of physical property difference between rigid and flexible materials as well as tolerancing intents such as sequence of specification, measurement, and assembly. Compared to traditional methods, the semantic tolerancing allows us to estimate true variation ranges such that feasible and complete solutions can be obtained.

Keywords: 3D Tolerance Modeling, Engineering Design, Interpretability, Semantic Tolerancing, Interval Analysis, Modal Interval

1 Introduction

Tolerance modeling forms an important link between design and manufacturing processes. A significant amount of research efforts has been given to explore the mathematical basis for 3D dimensional and geometric tolerance representation, analysis, and synthesis. Problems of tolerance relations can be mathematically formulated and solved in different ways. The typical methods for analysis include variational estimation, kinematic formulation, statistical approximation, and Monte Carlo simulation. However, current tolerance modeling methods do not represent the semantics of tolerance specifications well.

First, traditional tolerance analysis methods assume objects have rigid geometry. Variance is increasingly “stack-up” as components are assembled. As shown in *Figure 1*, tolerance of assembly is always assumed to be larger than its subassembly. Rigid body tolerance analysis over-estimates variations of flexible materials, such as assemblies containing sheet metal, polymer, and plastic parts, which are common in aerospace, automobile, and electronics industry. For example, an airplane skin can be slightly warped, and yet it can be riveted in place. Similarly,

subassembly components of auto body with much larger variation than the specified can still

achieve the final assembly specification. The conventional addition theorem of variance is no longer valid in these applications. Given the specification of an assembly, unreasonably tight tolerance requirements will be assigned to subassemblies and components during tolerance synthesis, as shown in *Figure 2*. The tolerance allocation based on the rigid body assumption increases manufacturing costs unnecessarily. These methods treat tolerances for rigid and compliant assemblies with the same scheme of +/- range. This does not capture the physical property difference between rigid and flexible materials and implied engineering meanings.

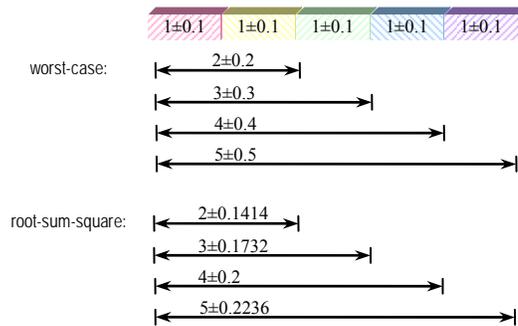


Figure 1. Tolerance ranges are monotonously increasing as assembly is built based on the rigid-body assumption

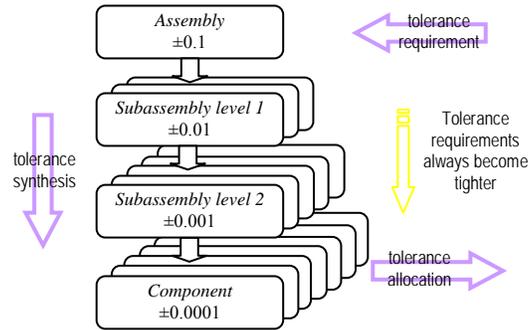


Figure 2. Tolerancing may become so tight that costs increase unnecessarily in flexible assembly based on current rigid-body tolerance synthesis schemes

Second, current tolerance modeling and analysis methods do not maintain the semantics of tolerance specifications. Two types of variation, *priori* and *posteriori*, are not differentiated in current tolerance models. *Priori* variation is predetermined but unknown, such as tolerances of components from suppliers. On the other hand, *posteriori* variation is known and controllable,

such as tolerances of components built in-house. Engineering implication and tolerance allocation strategies are different for two types of variation. Prior variation is not controllable, while posterior variation provides “buffers” in tolerance allocation. Further, how to interpret the numerical outputs with high and low bounds is important to understand the relation between tolerances. Numerical results of current methods are not interpretable. Engineering semantics needs to be maintained during mathematical computation.

Third, accuracy of range estimation is essential in tolerance analysis. Basic questions include *completeness* and *feasibility*. Complete solution includes all possible occurrences, which is to check if an interval includes all possible results. Feasible solution does not include impossible occurrences, which is to check if the interval over estimates the range. Current methods except Monte Carlo simulation with extensive sampling do not always give true range. The worst-case method tends to over estimate because of dependency between variables. Statistical methods do not give true range but statistical intervals. The results from vector and kinematic approaches are numerical estimations from algebraic approximations such as linearization. True range estimation should be both complete and feasible.

Instead of focusing only on mathematic and numerical convenience, a good mathematic model of tolerance should convey the full semantics of size and geometric tolerances and support analysis and synthesis with a simple yet comprehensive structure. Existing research does not concentrate on engineering semantics of tolerance zones. This leads to the problem that numerical solutions are not interpretable.

In this paper, we propose a new scheme to represent and analyze tolerance based on modal interval analysis. Extended from traditional set-based interval, modal interval introduces logical quantifiers and provides interpretation of intervals. Tolerancing semantics thus can be integrated into numerical calculation. In addition to better interpretability, modal interval analysis also provides better variation estimation than traditional interval analysis. The remainder of the paper is organized as follows. Section 2 gives an overview of related work on tolerance modeling and interval analysis, and an introduction to modal interval. Section 3 and 4 present the concept of semantic tolerance modeling and its two basic properties: interpretability and optimality. Section 5 describes analysis methods of the semantic tolerance model.

2 Background

2.1 3D TOLERANCE MODELING

There is plenty of literature on tolerance modeling (Hong and Chang, 2002; ADCATS). We just have a brief overview of 3D geometric tolerance zone representation related to the tolerance semantics. In the variational approaches, tolerance zones are established in 3D Euclidean space by parameter variation of spatial constraints and equations. Requicha (1983) proposed to construct tolerance zones by offsetting the part’s nominal boundaries. Inui *et al.* (1993)

approximate tolerance zone using boundary offset and geometric constraints. Roy and Li (1998; 1999) model tolerance zones of size, flatness, and parallelism in the variational form of plane equation. Teck *et al.* (2001) represent flatness of non-rectangular planar surfaces. Davidson *et al.* (Davidson *et al.*, 2002; Mujezinovic *et al.*, 2004) developed a hypothetical volume-based algebraic model to represent size, form, and orientation tolerances. Bhide *et al.* (2003) extended the method for cylindrical features.

In the statistical approaches (Nigam and Turner, 1995; Gerth, 1997), linear tolerance stack-up can be estimated using root-sum-square methods while non-linear stack-up is approximated using Taylor series. Typically it is assumed that the parameters are independent and the random variables are normally distributed. While the root-sum-square gives optimistic estimation, alternatives were proposed to do adjustment and correction for shifts and drifts (Chase and Greenwood, 1988). Srinivasan and O'Connor (1994) model and analyze tolerance based on statistical tolerance zone in the mean-variance (μ - σ^2) space, which is directly related to process capability indices in industry practices. Zhang *et al.* (1999) apply distribution function zone to tolerance synthesis. Different from other approaches, research in the statistical approach concentrates on dimensional tolerance stack-up and geometric tolerances are not modeled separately.

In the kinematic approaches, geometrical variation and displacement are modeled mathematically in vectors and matrices. Vectorial tolerancing (Wirtz *et al.*, 1993; Martinsen, 1995) models size, form, location, and orientation tolerances in a unified vector format in order to provide an integrated quality control loop. Small displacement torsor method (Bourdet and Ballot, 1995; Giordano and Duret, 1993) approximates rotation and translation displacement in the form of torsors. Matrix representation method (Whitney *et al.*, 1994; Desrochers and Riviere, 1997) models displacement in the form of homogenous transformation matrices. Rivest *et al.* (1994) exploit the kinematic character of the link imposed by a tolerance between the datum and the toleranced feature. Chase *et al.* (Chase *et al.*, 1996; Gao *et al.*, 1998) perform analysis of assembly using small kinematic adjustment between components based on linear approximation of implicit dimensional constraint functions. Joskowicz *et al.* (Joskowicz *et al.* 1997; Sack and Joskowicz, 1998) compute contact tolerance zones of planar parametric parts within configuration space. The kinematic methods distinguish size and each type of geometric tolerances. However, relations between variations are not modeled, and estimation result is hard to interpret.

In the Monte Carlo simulation approach (e.g. Turner and Wozny, 1987; Gao *et al.*, 1995; Ashiagbor *et al.*, 1998), no assumptions on independence and distribution are needed. Based on tolerance response relation, large amount of samples are randomly generated and evaluated in statistical estimation. The drawback is that the computational time for the required sampling process is high if good estimation is needed. It also depends on the pre-assumption of certain statistical distributions for input variables.

The above modeling and analysis methods have been widely accepted and used in commercial software such as Vis VSA® and CE/Tol®. However, it is not easy to interpret the

meanings of the specifications for each type of tolerances in component and assembly. Furthermore, the rigid-body assumption tends to over-estimate the variation of flexible materials.

2.2 TOLERANCE ANALYSIS FOR FLEXIBLE ASSEMBLY

There is relatively little research on tolerance analysis for flexible materials. Takezawa (1980) applied linear regression models to predict auto body panel (sheet metal parts) assembly variation using real production data, and he found variation of assembly could be smaller than individual parts. He concluded that “the conventional addition theorem of variance is no longer valid for deformable sheet metal assemblies”.

Liu and Hu (1997) proposed a linear finite element structural model to predict variation of sheet metal joining based on the concepts of mechanistic variation simulation and influence coefficient. Monte Carlo simulation is used to randomly displace nodes in a finite element model and the variance of the assembly can be estimated (Liu *et al.*, 1996). Long and Hu (1998) extended the method to include the variation of fixtures during assembly operations. Camelio *et al.* (2003) extended the method to multi-station assembly systems with compliant parts. Camelio *et al.* (2004) further applied principle component analysis to simplify covariance matrix in variance computation.

Merkley *et al.* (Merkley *et al.*, 1996; Merkley, 1998) developed a finite element tolerance analysis method for flexible assemblies based on linear elastic contact assumption. Polynomial interpolation is used to model geometric covariance between nodes, and stiffness matrix describes material covariance. Bihlmaier (1999) extended the method to consider autocorrelation in geometric covariance matrices.

The above finite element approaches have been integrated into some commercial software such as vis VSA and CATIA-TAA. However, tradeoff between fidelity and performance is always related to finite element methods. The computation becomes very expensive if the variance estimation involves complex assemblies. In most cases, accurate calculation of structural deformation and stress distribution is not the main purpose of tolerance analysis. Confidence of producibility and associated cost analysis need to be estimated without significant computation.

2.3 INTERVAL ANALYSIS

Interval mathematics is a generalization in which interval numbers replace real numbers, interval arithmetic replaces real arithmetic, and interval analysis replaces real analysis. The real number system \mathbf{R} is geometrically complete for numerical representation, but not practical for digital computing. Not only intervals solve the problem of representation for real numbers on a digital scale, but they are the most suitable way to represent uncertainties and errors in technical constructions, measuring, computations, and ranges of fluctuation and variation.

The set of intervals corresponding to real numbers is $I(\mathbf{R})$. Let $[a] = [\underline{a}, \bar{a}]$, $[b] = [\underline{b}, \bar{b}]$ be real intervals and \circ be one of the four basic arithmetic operations for real numbers, $\circ \in \{+, -, \cdot, /$. The corresponding operations for interval $[a]$ and $[b]$ are defined by

$$[a] \circ [b] = \{x \circ y \mid x \in [a], y \in [b]\}.$$

Interval analysis has been extensively used in reliable computing in computer science. In engineering fields, methods of interval analysis have been used in computer graphics (Mudur and Koparkar, 1984; Toth, 1985; Moore and Wilhelms, 1988; Duff, 1992; Snyder, 1999), robust geometry construction and evaluation (Abrams *et al.*, 1998; Shen and Patrikalakis, 1998; Tuohy *et al.*, 1997; Wallner *et al.*, 2000), set-based modeling (Finch and Ward, 1997), imprecise structural analysis (Rao and Berke, 1997), design optimization (Rao and Cao, 2002), finite-element formulation and analysis (Muhanna and Mullen, 1999; 2001; Muhanna *et al.*, 2004), solving soft geometric constraint and preference (Wang, 2004; Wang and Nnaji, 2006), and worst-case tolerance analysis and synthesis (Yang *et al.*, 2000).

Interval analysis has intrinsic uncertainty and variance properties for tolerance analysis. However, it is based on a worst-case scenario as in traditional linear stack-up methods. The results usually are pessimistic in this variance addition scheme if dependency exists between variables. In contrast, modal interval analysis is an extension of the traditional interval analysis, which differentiates semantics of interval specification in different application situations.

2.4 MODAL INTERVAL ANALYSIS

Modal interval analysis (MIA) (Gardenes *et al.*, 2001; Popova, 2001; Armengol *et al.*, 2001) is a logical and semantic extension of traditional interval analysis. MIA extends real numbers to intervals. Unlike classical interval analysis which identifies an interval by a set of real numbers, MIA identifies the intervals by the set of predicates which is fulfilled by the real numbers.

Given the set of closed intervals of \mathbf{R} , $I(\mathbf{R})$, and the set of logical existential (E or \exists) and universal (U or \forall) quantifiers, a modal interval is defined by a pair:

$$X := (X', Q_X)$$

in which $X' \in I(\mathbf{R})$ and $Q_X \in \{E, U\}$. X' is the classic interval and Q_X is one of the two modalities.

Similar to the way in which real numbers are associated in pairs with same absolute value but opposite $+$ and $-$ signs, modal intervals are associated in pairs too. Each member of a pair is corresponding to the same closed interval of real line, but having opposite modalities of existential or universal. The quantifiers are operators which transform real predicates into interval predicates. They are written as $E(x, X')P(x)$ and $U(x, X')P(x)$, indicating both arguments, the real index x and the interval argument X' . The notations $E(x, X')$ and $U(x, X')$ are interpreted as $(\exists x \in X')$ and $(\forall x \in X')$ respectively.

The canonical notation for modal interval is

$$[a, b] := \begin{cases} ([a, b]', E) & \text{if } a \leq b \\ ([b, a]', U) & \text{if } a \geq b \end{cases}$$

A modal interval $([a, b]', E)$ is called *existential* or *proper* interval whereas $([b, a]', U)$ is called *universal* or *improper* interval. The set of modal intervals is denoted by $I^*(\mathbf{R})$. The modal quantifier Q is associated with every real predicate $P(\cdot)$. For a variable $x \in \mathbf{R}$ and $(X', Q_x) \in I^*(\mathbf{R})$, Q is interpreted by Q_x as

$$Q(x, (X', Q_x))P(x) := Q_x(x, X')P(x).$$

Predicates of modal intervals are defined as the set of real predicates.

$$Pred(X', Q_x) := \{P(\cdot) \in Pred(\mathbf{R}) \mid Q(x, (X', Q_x))P(x)\}.$$

Based on the above semantic extension, basic arithmetic operations of modal interval are defined as follows. For $A = [a_1, a_2]$ and $B = [b_1, b_2]$,

$$A + B = [a_1 + b_1, a_2 + b_2], \quad A - B = [a_1 - b_2, a_2 - b_1]$$

$$A \times B = \begin{cases} [a_1 b_1, a_2 b_2] & (a_1 \geq 0, a_2 \geq 0, b_1 \geq 0, b_2 \geq 0) \\ [a_1 b_1, a_1 b_2] & (a_1 \geq 0, a_2 \geq 0, b_1 \geq 0, b_2 < 0) \\ [a_2 b_1, a_2 b_2] & (a_1 \geq 0, a_2 \geq 0, b_1 < 0, b_2 \geq 0) \\ [a_2 b_1, a_1 b_2] & (a_1 \geq 0, a_2 \geq 0, b_1 < 0, b_2 < 0) \\ [a_1 b_1, a_2 b_1] & (a_1 \geq 0, a_2 < 0, b_1 \geq 0, b_2 \geq 0) \\ [\max(a_1 b_1, a_2 b_2), \min(a_2 b_1, a_1 b_2)] & (a_1 \geq 0, a_2 < 0, b_1 \geq 0, b_2 < 0) \\ [0, 0] & (a_1 \geq 0, a_2 < 0, b_1 < 0, b_2 \geq 0) \\ [a_2 b_2, a_1 b_2] & (a_1 \geq 0, a_2 < 0, b_1 < 0, b_2 < 0) \\ [a_1 b_2, a_2 b_2] & (a_1 < 0, a_2 \geq 0, b_1 \geq 0, b_2 \geq 0) \\ [0, 0] & (a_1 < 0, a_2 \geq 0, b_1 \geq 0, b_2 < 0) \\ [\min(a_1 b_2, a_2 b_1), \max(a_1 b_1, a_2 b_2)] & (a_1 < 0, a_2 \geq 0, b_1 < 0, b_2 \geq 0) \\ [a_2 b_1, a_1 b_1] & (a_1 < 0, a_2 \geq 0, b_1 < 0, b_2 < 0) \\ [a_1 b_2, a_2 b_1] & (a_1 < 0, a_2 < 0, b_1 \geq 0, b_2 \geq 0) \\ [a_2 b_2, a_2 b_1] & (a_1 < 0, a_2 < 0, b_1 \geq 0, b_2 < 0) \\ [a_1 b_2, a_1 b_1] & (a_1 < 0, a_2 < 0, b_1 < 0, b_2 \geq 0) \\ [a_2 b_2, a_1 b_1] & (a_1 < 0, a_2 < 0, b_1 < 0, b_2 < 0) \end{cases} \quad A/B = \begin{cases} [a_1/b_2, a_2/b_1] & (a_1 \geq 0, a_2 \geq 0, b_1 > 0, b_2 > 0) \\ [a_2/b_2, a_1/b_1] & (a_1 \geq 0, a_2 \geq 0, b_1 < 0, b_2 < 0) \\ [a_1/b_2, a_2/b_2] & (a_1 \geq 0, a_2 < 0, b_1 > 0, b_2 > 0) \\ [a_2/b_1, a_1/b_1] & (a_1 \geq 0, a_2 < 0, b_1 < 0, b_2 < 0) \\ [a_1/b_1, a_2/b_1] & (a_1 < 0, a_2 \geq 0, b_1 > 0, b_2 > 0) \\ [a_2/b_2, a_1/b_2] & (a_1 < 0, a_2 \geq 0, b_1 < 0, b_2 < 0) \\ [a_1/b_1, a_2/b_2] & (a_1 < 0, a_2 < 0, b_1 > 0, b_2 > 0) \\ [a_2/b_1, a_1/b_2] & (a_1 < 0, a_2 < 0, b_1 < 0, b_2 < 0) \end{cases}$$

The inclusion relation between modal intervals is defined as $[a_1, a_2] \subseteq [b_1, b_2] \Leftrightarrow (a_1 \geq b_1, a_2 \leq b_2)$. Semantically, $A \subseteq B \Leftrightarrow Pred(A) \subseteq Pred(B)$. If $A \subseteq B$, the implication $Q(x, A)P(x) \Rightarrow Q(x, B)P(x)$ is valid. The “less or equal” relation is defined as $[a_1, a_2] \leq [b_1, b_2] \Leftrightarrow (a_1 \leq b_1, a_2 \leq b_2)$. Some modal interval operations are defined as $Prop([a_1, a_2]) := [\min(a_1, a_2), \max(a_1, a_2)]$, $Impr([a_1, a_2]) := [\max(a_1, a_2), \min(a_1, a_2)]$, and $Width([a_1, a_2]) := |a_1 - a_2|$.

MIA is able to model problems on a logical basis and to obtain the interval functional evaluations for the mathematical model involved. Based on modal interval, we propose a new semantic tolerance modeling scheme, in which the implications of tolerance stacking can be embedded in the tolerance model. Accurate range estimation can be achieved compared to traditional worst-case interval methods.

The purpose of semantic tolerance modeling is to capture logical therefore engineering meanings and implications in mathematical representation, which is to build a bridge between mathematic theory and engineering practice. Semantic tolerance modeling has two important characteristics: (1) *Interpretability*: being able to interpret tolerance intervals during analysis and synthesis processes and to provide the basic understanding of tolerancing semantics; and (2) *Optimality*: being able to analyze tolerance propagation and accumulation so that tolerances can be specified without losing the basic requirements of completeness and feasibility. Interpretability allows tolerance semantics to be embedded in interval results. Optimality assures tightness of variation estimation. The following sections will describe the properties of modal interval representation in semantic tolerancing.

3 Interpretability

The uniqueness of modal interval is the modal semantic extension. If a real relation $z = f(x_1, \dots, x_n)$ is extended to the interval relation $Z = F(f)(X_1, \dots, X_n)$, the interval relation Z is interpretable if there is a semantic relation

$$Q_1(x_1, X_1) \cdots Q_n(x_n, X_n) Q_z(z, F(f)(X_1, \dots, X_n)) z = f(x_1, \dots, x_n).$$

A component x is *uni-incident* in a function $f(\mathbf{X})$ if it occupies only one leaf of the syntax tree for the function. Otherwise, it is *multi-incident*. To reduce the interdependency effect of multi-incident, which usually over estimates interval function ranges, two interval extensions of real function $f(\mathbf{x})$, so-called semantic interval functions, are defined in min-max situation as:

$$f^*(\mathbf{X}) := [\min_{x_p \in X'_p} \max_{x_i \in X'_i} f(x_p, x_i), \max_{x_p \in X'_p} \min_{x_i \in X'_i} f(x_p, x_i)],$$

$$f^{**}(\mathbf{X}) := [\max_{x_i \in X'_i} \min_{x_p \in X'_p} f(x_p, x_i), \min_{x_i \in X'_i} \max_{x_p \in X'_p} f(x_p, x_i)],$$

where (x_p, x_i) is the component splitting corresponding to interval vector $\mathbf{X} = (X_p, X_i)$, with X_p and X_i are sub-vectors containing proper and improper components respectively.

Important properties of interpretability are available and proved.

Theorem 3.1 (Gardenes *et al.*, 2001) Given a continuous function $f: \mathbf{R}^n \rightarrow \mathbf{R}$ and a modal vector $\mathbf{X} \in I^*(\mathbf{R}^n)$, if there exists an interval $F(\mathbf{X}) \in I^*(\mathbf{R})$, then

$$f^*(\mathbf{X}) \subseteq F(\mathbf{X}) \Leftrightarrow U(x_p, X'_p) Q(z, F(\mathbf{X})) E(x_i, X'_i) z = f(x_p, x_i).$$

Theorem 3.2 (Gardenes *et al.*, 2001) Given a continuous function $f : \mathbf{R}^n \rightarrow \mathbf{R}$ and a modal vector $\mathbf{X} \in I^*(\mathbf{R}^n)$, if there exists an interval $F(\mathbf{X}) \in I^*(\mathbf{R})$, then

$$f^{**}(\mathbf{X}) \supseteq F(\mathbf{X}) \Leftrightarrow U(x_i, X'_i)Q(z, Dual(F(\mathbf{X})))E(x_p, X'_p)z = f(x_p, x_i),$$

where Dual operator is defined as $Dual([a, b]) := [b, a]$.

3.1 UNI-INCIDENT INTERPRETATION

Let $f : \mathbf{R}^n \rightarrow \mathbf{R}$ be a rational continuous function. Its modal rational extension $fR : I^*(\mathbf{R}^n) \rightarrow I^*(\mathbf{R})$ is simply replacing the real variables of f with modal interval variables.

Theorem 3.3 (Gardenes *et al.*, 2001) For a modal rational function $fR(\mathbf{X})$, if all arguments of $fR(\mathbf{X})$ are uni-incident, then

$$f^*(\mathbf{X}) \subseteq fR(\mathbf{X}) \subseteq f^{**}(\mathbf{X}).$$

From Theorems 3.1, 3.2, and 3.3, we know modal rational functions of uni-incident variables are interpretable. For example, $f(x, y) = x + y$ is considered for $X' = [1, 3]'$ and $Y' = [2, 5]'$.

$$\begin{aligned} fR([1, 3], [2, 5]) &= [1, 3] + [2, 5] = [3, 8], \\ fR([1, 3], [5, 2]) &= [1, 3] + [5, 2] = [6, 5], \\ fR([3, 1], [2, 5]) &= [3, 1] + [2, 5] = [5, 6], \\ fR([3, 1], [5, 2]) &= [3, 1] + [5, 2] = [8, 3], \end{aligned}$$

have the meanings of

$$\begin{aligned} U(x, [1, 3]')U(y, [2, 5]')E(z, [3, 8]')z &= x + y \text{ or } \forall x \in [1, 3]', \forall y \in [2, 5]', \exists z \in [3, 8]', z = x + y, \\ U(x, [1, 3]')U(z, [5, 6]')E(y, [2, 5]')z &= x + y \text{ or } \forall x \in [1, 3]', \forall z \in [5, 6]', \exists y \in [2, 5]', z = x + y, \\ U(y, [2, 5]')E(x, [1, 3]')E(z, [5, 6]')z &= x + y \text{ or } \forall y \in [2, 5]', \exists x \in [1, 3]', \exists z \in [5, 6]', z = x + y, \\ U(z, [3, 8]')E(x, [1, 3]')E(y, [2, 5]')z &= x + y \text{ or } \forall z \in [3, 8]', \exists x \in [1, 3]', \exists y \in [2, 5]', z = x + y, \end{aligned}$$

respectively.

Different semantics of linear tolerance stack-up in assembly enclosure needs to be differentiated. For example, in Figure 3, dimensions a , b , and c in three components have relation $a + b = c$. According to different assembly sequences or manufacturing needs, we may specify tolerances in different ways. If Part A and B are provided by suppliers and Part C is built in house (Figure 3-b, Case I), the tolerance of c is determined by the tolerances of a and b . In this case, the semantics of “given A and B, C needs to fit A and B” is expressed as $\forall a \in A', \forall b \in B', \exists c \in C', a + b = c$, which is different from the semantics of “given A, B and C need to fit A” when Part A is supplied and Part B and C are built in house (Figure 3-c, Case II). The relations between tolerances should be compatible with the semantics of specifications. In the semantic tolerance model, priori and posteriori tolerances are differentiated. In Case I, a and b

have priori tolerances, while c has a posteriori tolerance. With the modal extension, the semantics of specification sequence and rational can be embedded in the model.

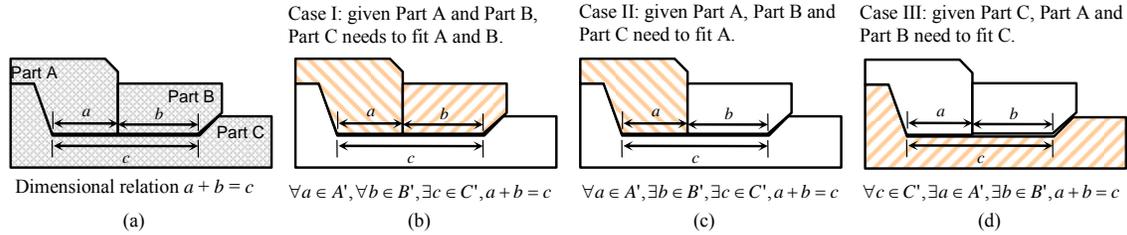


Figure 3. Different types of semantics need to be captured, which are not differentiated in traditional modeling methods

With the differentiation of priori and posteriori tolerances, strategy of tolerance allocation could vary in different scenarios. For example, in Figure 3-b, given two “uncontrollable” dimensions a and b , the “controllable” dimension $c = a + b = [2,5] + [1,3] = [3,8]$. In Figure 3-c, one extra controllable dimension b allows a tighter tolerance of c . $c = a + b = [2,5] + [3,1] = [5,6]$. The tolerance range of c is reduced from 5 to 1, which is smaller than the tolerance range of a . This implies that the principle of selective assembly can be applied to achieve assembly.

3.2 MULTI-INCIDENT INTERPRETATION

Theorem 3.4 (Gardenes *et al.*, 2001) For a modal rational function $fR(\mathbf{X})$, if there are multi-incident improper arguments in $fR(\mathbf{X})$ and \mathbf{XT}^* is obtained from \mathbf{X} , by transforming, for every multi-incident improper component, all incidences but one into its dual, then

$$f^*(\mathbf{X}) \subseteq fR(\mathbf{XT}^*).$$

Theorem 3.5 (Gardenes *et al.*, 2001) For a modal rational function $fR(\mathbf{X})$, if there are multi-incident proper arguments in $fR(\mathbf{X})$ and \mathbf{XT}^{**} is obtained from \mathbf{X} , by transforming, for every multi-incident proper component, all incidences but one into its dual, then

$$f^{**}(\mathbf{X}) \supseteq fR(\mathbf{XT}^{**}).$$

From Theorems 3.1, 3.2, 3.4, and 3.5, modal rational functions of multi-incident variables are interpretable with some modification. For example, $f(x, y) = xy/(x + y)$ is extended to $X = [-1,3]$ and $Y = [15,7]$.

$$fR(\mathbf{X}) = [-1,3] \times [15,7] / ([-1,3] + [15,7]) = [-0.5, 1.5]$$

is not interpretable, whereas

$$\begin{aligned}
fR(\mathbf{XT}^*) &= [-1,3] \times [15,7] / ([-1,3] + [7,15]) = [-1.16667, 3.5], \\
fR(\mathbf{XT}^*) &= [-1,3] \times [7,15] / ([-1,3] + [15,7]) = [-1.07143, 3.21429], \\
fR(\mathbf{XT}^{**}) &= [-1,3] \times [15,7] / ([3,-1] + [15,7]) = [-0.388889, 1.16667], \\
fR(\mathbf{XT}^{**}) &= [3,-1] \times [15,7] / ([-1,3] + [15,7]) = [4.5, -1.5]
\end{aligned}$$

are interpretable. They are interpreted as

$$\begin{aligned}
U(x, [-1,3]') E(y, [7,15]') E(z, [-1.16667, 3.5]') z &= xy / (x + y), \\
U(x, [-1,3]') E(y, [7,15]') E(z, [-1.07143, 3.21429]') z &= xy / (x + y), \\
U(x, [-1,3]') E(y, [7,15]') E(z, [-0.388889, 1.16667]') z &= xy / (x + y), \\
U(x, [-1,3]') U(z, [-1.5, 4.5]') E(y, [7,15]') z &= xy / (x + y)
\end{aligned}$$

respectively.

Combining the first three results, we have

$$U(x, [-1,3]') E(y, [7,15]') E(z, [-0.388889, 1.16667]') z = xy / (x + y),$$

In assembly, parametric relations with multi-incident variables are common. Compared to traditional tolerance modeling, semantic tolerance modeling allows us to interpret explicit algebraic relations with the interpretability properties of modal intervals. Different numerical values can also be selected in order to derive specific semantics.

3.3 RIGIDITY INTERPRETATION

While existential intervals are looked as “fluctuation” or “autonomous” ranges, universal intervals are regard as “regulating” or “feedback” ranges. In material property domain, tolerance range for rigid material is corresponding to existential interval and flexible material is to universal interval.

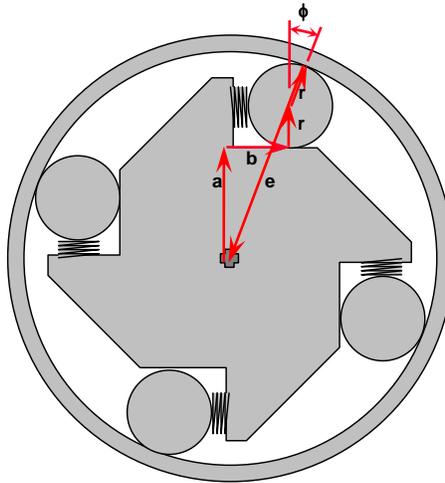


Figure 4. variations of size and geometry, shape deformation, and kinematics form a closed loop in assembly

In the one-way clutch example of Figure 4, the distance vector b , the length of the spring s , and the radius of the ball r satisfy the relation $r+s=b$. If ranges $[5.2,5.7]'$ and $[7.8,8.0]'$ are given to r and b respectively, the range for spring length s can be $[2.1,2.8]'$, as in relation

$$R + S = [5.2,5.7] + [2.8,2.1] = [8.0,7.8] = B .$$

It is interpreted as

$$U(r,[5.2,5.7]')U(b,[7.8,8.0]')E(s,[2.1,2.8]')r + s = b .$$

The spring provides a “cushion” to absorb variance. If a larger range $[7.8,8.5]'$ is allowed for b , no flexible material is required to absorb variance. Rigid material instead of spring for s can be chosen, as in relation

$$R + S = [5.2,5.7] + [2.6,2.8] = [7.8,8.5] = B .$$

It is interpreted as

$$U(r,[5.2,5.7]')U(s,[2.6,2.8]')E(b,[7.8,8.5]')r + s = b .$$

As illustrated in Figure 5, the semantic difference between rigid and flexible material is differentiated by interval modality. Selection of rigid or flexible materials is integrated into algebraic relation.

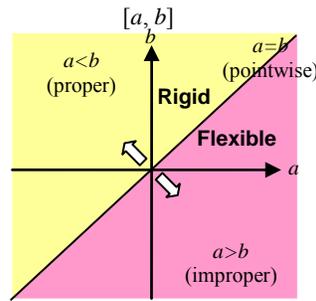


Figure 5. Rigidity diagram

3.4 SEMANTIC TOLERANCING

With modal extension, engineering semantics such as sequences of specification, manufacturing, and assembly, as well as material properties can be captured. Tolerance semantics can be grouped into existential and universal categories, including tolerancing intent, specification precedence and dependency, as well as differentiation of constraint and preference. Taxonomy of specification semantics thus can be developed. Some examples of such semantics are listed in Table 1. Semantic pairs exist in the domains of supply management, manufacturing and assembly sequences, etc.

Table 1. Examples of tolerance semantics

Domain	Existential or Proper category	Universal or Improper category
Supply management	Pre-determined, Uncontrollable, Supplied	Un-determined, Controllable, Built
Manufacturing sequence	Working dimension, Clearance	Balance dimension, Stock removal
Assembly sequence	Place, Virtual condition size	Fit, Bonus tolerance
Material property	Rigid, Wearable	Flexible, Deformable
Process control	Open loop, Manual mode	Closed loop, Auto mode

4 Optimality

For every $\mathbf{X} \in I^*(\mathbf{R}^n)$, if the modal rational extension $fR(\mathbf{X})$ satisfy $f^*(\mathbf{X}) = fR(\mathbf{X}) = f^{**}(\mathbf{X})$, $fR(\cdot)$ is called optimal. In other words, if the evaluation of a modal rational function $fR(\mathbf{X})$ is both complete and feasible, $fR(\cdot)$ is optimal for \mathbf{X} . Optimal functions give tight bounds of complete estimation.

4.1 UNI-INCIDENT OPTIMALITY

Theorem 4.1 (Armengol *et al.*, 2001) If all arguments of $fR(\mathbf{X})$ are uni-incident and they have the same modality,

$$f^*(\mathbf{X}) = fR(\mathbf{X}) = f^{**}(\mathbf{X}).$$

For example, $f(x, y) = (x + y)^2$ is optimal for $X = [1, 3]$ and $Y = [2, 5]$. The true range of the function $R_f = [9, 64]$. The natural extension is $fR([1, 3], [2, 5]) = ([1, 3] + [2, 5])^2 = [9, 64]$. Similarly, $f([3, 1], [5, 2]) = [64, 9]$ is optimal. However, $g(x, y) = x^2 + 2xy + y^2$ is not optimal. $f(x, y)$ is not optimal for $X = [1, 3]$ and $Y = [5, 2]$.

4.2 MULTI-INCIDENT OPTIMALITY

Theorem 4.2 (Armengol *et al.*, 2001) If $fR(\mathbf{X})$ are totally monotonous for all of its multi-incident arguments, and \mathbf{XD} is obtained from \mathbf{X} , by transforming, for every multi-incident component, all incidences into its dual if the corresponding incidence has a mononicity sense contrary to the global one, then

$$f^*(\mathbf{X}) = fR(\mathbf{XD}) = f^{**}(\mathbf{X}).$$

For example, $f(x, y) = xy/(x + y)$ is extended to $X = [1, 3]$ and $Y = [15, 7]$. The partial derivatives of f with respect to x and y are all positive within the domain. The partial derivatives of f with respect to the first incidences of x and y are positive, and negative respect to the second incidences of x and y . Therefore,

$$fR(\mathbf{XD}) = [1, 3] \times [15, 7] / ([3, 1] + [7, 15]) = [0.9375, 2.1]$$

is optimal, compared to $gR(X, Y) = 1/(1/X + 1/Y) = 1/(1/[1, 3] + 1/[15, 7]) = [0.9375, 2.1]$.

4.3 EXAMPLE A: TRUE RANGE ESTIMATION OF ONE-WAY CLUTCH

To illustrate the optimality of modal interval in range estimation, a comparison of MIA method and Direct Linearization Method (DLM) (Chase *et al.*, 1997) (as implemented in CE/Tol[®] package) for the one-way clutch example is made, as shown in *Figure 6* and *Table 2*. Compared to the methods of DLM with Root-Sum-Square (RSS) and Worst-Case (WC), MIA gives accurate estimation of true variation range.

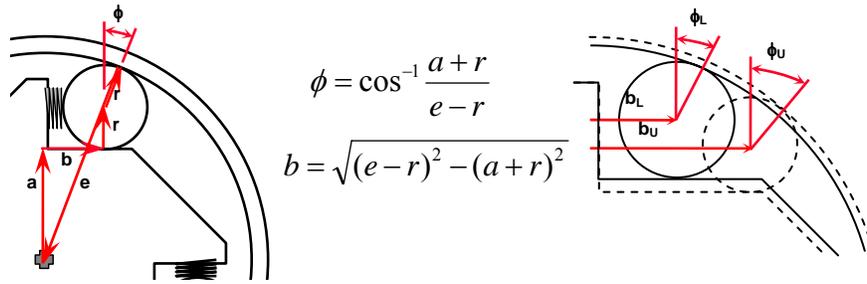


Figure 6. Modal interval makes complex algebraic relations with multi-incident variables interpretable. Interpretations are corresponding to different value sets

Table 2. Result comparison between MIA and DLM method

Input			Output: position of roller (b) True Range is [4.0838,5.4405]		
Hub Height (a)	Ring Radius (e)	Roller Radius (r)	DLM with Root-Sum-Square (as in CE/Tol [®])	DLM with Worst-Case (as in CE/Tol [®])	MIA
[27.595, 27.695]	[50.7875, 50.8125]	[11.42, 11.44]	[4.3585, 5.2625]	[4.1368, 5.4842]	[4.0838, 5.4405]

4.4 EXAMPLE B: HARD DISK TRACKS

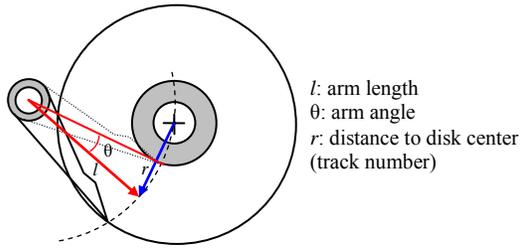
Figure 7 shows an example of hard disk tolerance analysis simulation. To seek tracks, the arm of the hard drive moves certain angles incrementally. Each disk surface may have tens of thousands tracks. Thus, precise movement at high speed is critical to find correct tracks given uncertainty involved in control and geometric variations. A traditional interval model to estimate the distance from each track to disk center is based on

$$R_{k+1} = R_k \cos \frac{\Delta}{2} + \sqrt{4L^2 - R_k^2} \sin \frac{\Delta}{2}$$

for $L = [42.00, 42.02]$ mm, $\Delta = [0.0002, 0.00021]$, and $R_0 = [10.35, 10.37]$ mm. The over-estimation of the range grows as the track number increases. However, the optimal modal interval model based on

$$R_{k+1} = R_k \cos \frac{\Delta}{2} + \sqrt{4L^2 - [Dual(R_k)]^2} \sin \frac{\Delta}{2}$$

gives a tighter range estimate, as compared in Figure 7-d.

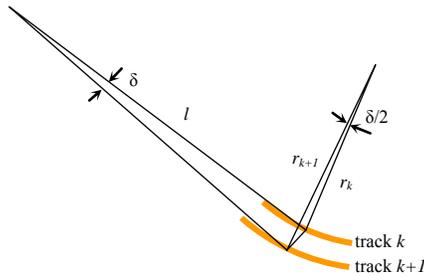


(a) In hard disk, precise arm movement is required to seek tracks

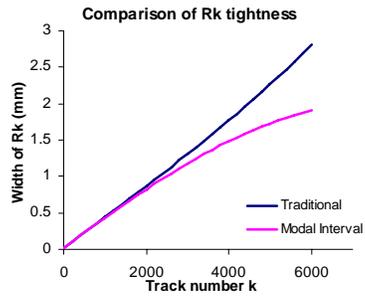
$$r_{k+1} = r_k \cos \frac{\delta}{2} + \sqrt{4l^2 - r_k^2} \sin \frac{\delta}{2}$$

r_k : distance from track k to disk center
 r_{k+1} : distance from track $k+1$ to disk
 δ : arm angle increment for each track

(c) Incremental relation between track distances



(b) Illustration of distance relation between adjacent tracks



(d) Tighter variation estimation based on modal interval compared to traditional worst-case estimation

Figure 7. Modal interval gives interpretable and tighter variation estimation result in track distance simulation

4.5 EXAMPLE C: PROCESS CONTROL SIMULATION

A third example of optimality is a derivative process control simulation, which shows the significant difference between modal interval and traditional interval methods, as compared in Figure 8. With uncertainty involved in parameters, the tooling speed range estimation with respect to time based on MIA optimal extension

$$V(k+1) = V(k) + K_d[V_0 - dual(V(k))] - \frac{1}{S}[dual(V(k)) - V_a]$$

is much better than that of the worst-case traditional interval extension

$$V(k+1) = V(k) + K_d[V_0 - V(k)] - \frac{1}{S}[V(k) - V_a].$$

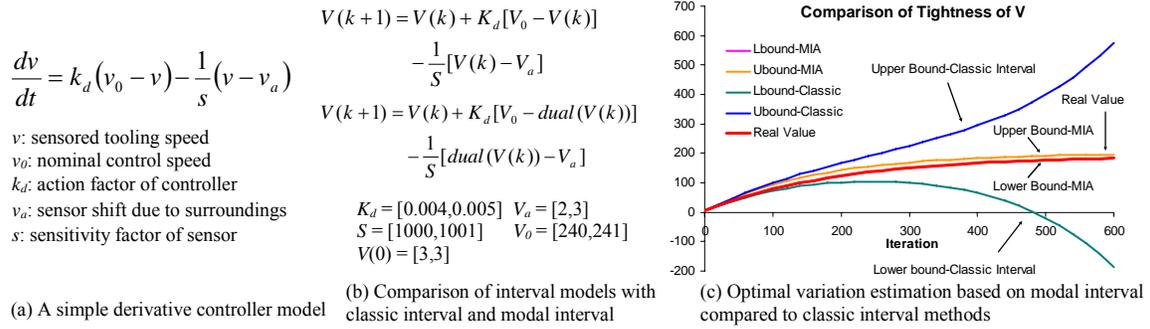


Figure 8. Modal interval shows optimal estimation of variation in a process control simulation

5 Closed-Loop Tolerance Analysis

Besides the semantic completion described in Section 3, MIA has the good property of structural completion. Traditional set-based interval analysis is not complete. The group properties of addition and multiplication operations are lost. There is no interval $[x,y]$ such that $[a,b] + [x,y] = 0$ and the equation $[a,b] + [x,y] = [c,d]$ has an interval solution only when $b - a \leq d - c$. For example, $[1,3] + [x,y] = [2,7]$ has solution $[x,y] = [2,7] - [1,3] = [-1,6]$ instead of $[1,4]$. In contrast, arithmetic operations in MIA are complete.

5.1 CLOSENESS OF MIA ARITHMETIC OPERATIONS

In MIA, it is easy to find true solution for equation $A + X = B$, which is $X = B - dual(A)$, and $AX = B$, which is $X = B / dual(A)$. Thus, $[1,3] + [x,y] = [2,7]$ has the true solution $[x,y] = [2,7] - dual([1,3]) = [2,7] - [3,1] = [1,4]$.

Given that a and b have values from intervals $[2,4]'$ and $[-2,6]'$, finding the interval estimation X for the equation $ax = b$ is interpreted as

$$U(x, X')E(a, [2,4]')E(b, [-2,6]')ax = b.$$

Therefore, X will be the proper interval solution of the equation

$$[4,2] \times X = [-2,6].$$

Thus,

$$X = [-2,6] / dual([4,2]) = [-1,3].$$

The optimality of MIA arithmetic allows us to overcome the over-estimation barrier in worst-case stack-up. True range estimation can be achieved without extensive computation as in simulation approach. In addition, the estimated 3D variation vectors from size, geometry, and

kinematic tolerances such as the one in *Figure 4* should be *closed* in a complete assembly, that is, tolerance ranges R_i in x , y , and z directions should satisfy $f(R_1, R_2, \dots, R_n) = 0$. This constraint in turn helps to estimate ranges more accurately. Traditional methods do not consider the closeness constraint. The closeness of MIA arithmetic operations provides the fundamentals for the soundness of semantic tolerancing.

5.2 TOLERANCE ANALYSIS

Tolerance formulation and numerical methods based on MIA arithmetic operations maintain the completeness of interval computation. During the tolerance and kinematic chain formulation, if explicit functions are available in tolerance analysis, such as in Section 4, accurate and interpretable variation ranges can be estimated. If only implicit functions are available, methods to solve modal interval systems are needed.

An interval system of MIA linear equations $\mathbf{A} \cdot \mathbf{X} = \mathbf{B}$, where $\mathbf{A} = (A_{ij})_{n \times n}$ and $\mathbf{B} = (B_i)_{n \times 1}$, is closely associated with two relations $\mathbf{A} \cdot \mathbf{X} \subseteq \mathbf{B}$ and $\mathbf{A} \cdot \mathbf{X} \supseteq \mathbf{B}$.

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{B} \Leftrightarrow \mathbf{A} \cdot \mathbf{X} \subseteq \mathbf{B} \text{ and } \mathbf{A} \cdot \mathbf{X} \supseteq \mathbf{B}.$$

If a *Jacobi interval operator* is defined as

$$\mathfrak{J}(X_i) := \frac{B_i - \sum_{i \neq j} \text{Dual}(A_{ij}) \times \text{Dual}(X_j)}{\text{Dual}(A_{ii})} \quad (0 \notin A_{ii} \text{ and } i = 1, \dots, n),$$

the following theorem is the foundation of solving MIA linear systems optimally.

Theorem 5.1 (Sainz et al., 2002a; 2002b) (1) If X is a solution to $\mathbf{A} \cdot \mathbf{X} \subseteq \mathbf{B}$, $\mathfrak{J}(\mathbf{X})$ is a solution to $\mathbf{A} \cdot \mathbf{X} \supseteq \mathbf{B}$. (2) If X is a solution to $\mathbf{A} \cdot \mathbf{X} \supseteq \mathbf{B}$, $\mathfrak{J}(\mathbf{X})$ is a solution to $\mathbf{A} \cdot \mathbf{X} \subseteq \mathbf{B}$.

The Jacobi algorithm to solve MIA linear systems is listed in *Figure 9*. By means of the Jacobi interval operator associated with the linear system $\mathbf{A}\mathbf{X} = \mathbf{B}$, it is possible to get a sequence of interval vectors $\mathbf{X}^{(1)} = \mathfrak{J}(\mathbf{X}^{(0)})$, $\mathbf{X}^{(2)} = \mathfrak{J}(\mathbf{X}^{(1)})$, ..., which satisfies

$$\mathbf{X}^{(0)} \supseteq \mathbf{X}^{(1)} \subseteq \mathbf{X}^{(2)} \supseteq \dots \subseteq \mathbf{X}^{(2k)} \supseteq \mathbf{X}^{(2k+1)} \subseteq \dots,$$

such that $\mathbf{X}^{(2k)}$ is a solution of $\mathbf{A} \cdot \mathbf{X} \supseteq \mathbf{B}$, and $\mathbf{X}^{(2k+1)}$ is a solution of $\mathbf{A} \cdot \mathbf{X} \subseteq \mathbf{B}$.

The Jacobi algorithm does not necessarily converge. The sufficient condition for convergence is described in Theorem 5.2.

Theorem 5.2 (Sainz et al., 2002a) For system $\mathbf{AX} = \mathbf{B}$, if $Prop(\mathbf{A})$ is a strictly diagonally dominant interval matrix, $\frac{\sum_{j \neq i} Width(A_{ij})}{Width(A_{ii})} < \alpha < 1$, there exists a limit \mathbf{X}^∞ satisfying $\mathbf{X}^\infty = \mathfrak{I}(\mathbf{X}^\infty)$.

Input: modal interval matrix \mathbf{A} , modal interval vector \mathbf{B}

Output: modal interval vector \mathbf{X} that satisfies $\mathbf{AX} = \mathbf{B}$

1. Initial estimation of $\mathbf{Y}^{(0)}$ such that $Impr(\mathbf{A}) \cdot \mathbf{Y}^{(0)} \subseteq Prop(\mathbf{B})$;
2. $\mathbf{X}^{(0)} = \mathfrak{I}(\mathbf{Y}^{(0)})$, which is the initial solution for $\mathbf{A} \cdot \mathbf{X} \supseteq \mathbf{B}$;
3. Iterate the follows for p times: $\mathbf{X}^{(t)} = \mathfrak{I}(\mathbf{X}^{(t-1)})$.

Figure 9. Jacobi algorithm to solve linear systems of modal intervals [**Error! Bookmark not defined.**]

If \mathbf{A} is not strictly diagonally dominant, general interval methods such as in references (Neumaier, 1990; Hansen, 1992; Ning and Kearfott, 1997) can be used to solve interval linear equations. However, the interpretability is compromised.

When variation functions are nonlinear, a linearization process may be used to reduce the complexity of direct computation of nonlinear functions. This linear approximation changes semantics relation between variables. Again, as a result of linearization, the tolerance interpretability and optimality principles generally do not apply to the numerical results.

5.3 EXAMPLE D: STACKED BLOCK ASSEMBLY – NONLINEAR

A closed vector loop defines relations among size, geometry, and kinematic variations. The sum of vector components in each translational or rotational direction should be equal to zero. *Figure 10* shows an example of stacked block assembly. With known size tolerances, the kinematic variation of the stacked blocks can be calculated with three loops. The parameter values and formulation of loops are listed in *Table 3*.

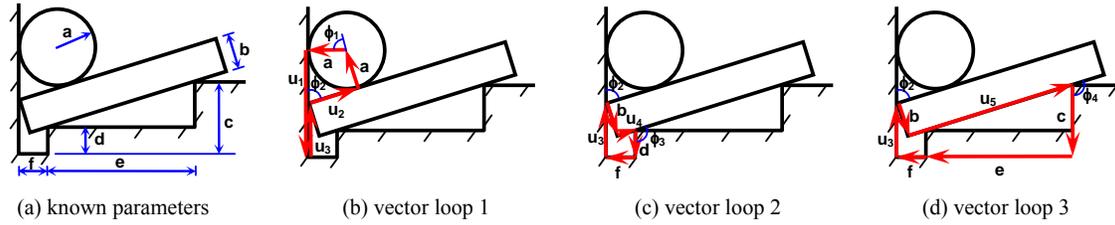


Figure 10. Stacked block assembly

Table 3. The variation formulation of loops

Known size variation	$a = 6.62 \pm 0.2$ $b = 6.805 \pm 0.075$ $c = 10.675 \pm 0.125$ $d = 4.06 \pm 0.15$ $e = 24.22 \pm 0.35$ $f = 3.905 \pm 0.125$
Unknown kinematic variation	$u_1 = 18.7181 \pm ?$ $u_2 = 8.6705 \pm ?$ $u_3 = 10.0477 \pm ?$ $u_4 = 2.1894 \pm ?$ $u_5 = 27.2965 \pm ?$ $\phi_1 = 74.7243 \pm ?$ $\phi_2 = -74.7243 \pm ?$ $\phi_3 = -105.2761 \pm ?$ $\phi_4 = -105.2761 \pm ?$
Loop 1	$\begin{cases} F_1 = u_2 \cos(90 + \phi_2) + a \cos(180 + \phi_2) + a \cos(180 + \phi_1 + \phi_2) = 0 \\ F_2 = u_3 + u_2 \sin(90 + \phi_2) + a \sin(180 + \phi_2) + a \sin(180 + \phi_1 + \phi_2) - u_1 = 0 \\ F_3 = 90 + \phi_2 + 90 + \phi_1 + 90 + 90 - 360 = 0 \end{cases}$
Loop 2	$\begin{cases} F_4 = b \cos(\phi_2) + u_4 \cos(\phi_2 + 90) + d \cos(\phi_2 + 90 + \phi_3) - f = 0 \\ F_5 = u_3 + b \sin(\phi_2) + u_4 \sin(\phi_2 + 90) + d \sin(\phi_2 + 90 + \phi_3) = 0 \\ F_6 = 90 + \phi_2 - 90 + 90 + \phi_3 - 90 + 180 = 0 \end{cases}$
Loop 3	$\begin{cases} F_7 = b \cos(\phi_2) + u_5 \cos(\phi_2 + 90) + c \cos(\phi_2 + 90 + \phi_4) - e - f = 0 \\ F_8 = u_3 + b \sin(\phi_2) + u_5 \sin(\phi_2 + 90) + c \sin(\phi_2 + 90 + \phi_4) = 0 \\ F_9 = 90 + \phi_2 - 90 + 90 + \phi_4 - 90 + 180 = 0 \end{cases}$

To solve nonlinear functions $F(\mathbf{s}, \mathbf{k}) = 0$, where \mathbf{s} is a size variation vector and \mathbf{k} is a kinematic variation vector, linearization process by Taylor's expansion

$$\left[\frac{\partial F_i}{\partial s_j} \right]_{i \times j} \Delta \mathbf{s} + \left[\frac{\partial F_i}{\partial k_i} \right]_{i \times i} \Delta \mathbf{k} = 0$$

with respect to nominal values is conducted before interval linear method is used to estimate the variation. The results from the MIA linearization method and the DLM method (Chase *et al.*, 1997) are compared in Table 4.

Table 4. Comparison of MIA linearization and DLM

MIA Linearization	DLM Worst-Case	DLM Statistical
$\Delta u_1 = [0.5420, -0.5420]$	$\Delta u_1 = [-0.5421, 0.5421]$	$\Delta u_1 = [-0.2998, 0.2998]$
$\Delta u_2 = [0.4672, -0.4672]$	$\Delta u_2 = [-0.3899, 0.3899]$	$\Delta u_2 = [-0.2725, 0.2725]$
$\Delta u_3 = [0.3137, -0.3137]$	$\Delta u_3 = [-0.2942, 0.2942]$	$\Delta u_3 = [-0.1844, 0.1844]$
$\Delta u_4 = [0.2729, -0.2729]$	$\Delta u_4 = [-0.2384, 0.2384]$	$\Delta u_4 = [-0.1411, 0.1411]$
$\Delta u_5 = [0.5209, -0.5209]$	$\Delta u_5 = [-0.5174, 0.5174]$	$\Delta u_5 = [-0.3836, 0.3836]$
$\Delta \phi_1 = [0.0228, -0.0228]$	$\Delta \phi_1 = [-0.8156, 0.8156]$	$\Delta \phi_1 = [-0.4784, 0.4784]$
$\Delta \phi_2 = [0.0228, -0.0228]$	$\Delta \phi_2 = [-0.8156, 0.8156]$	$\Delta \phi_2 = [-0.4784, 0.4784]$
$\Delta \phi_3 = [0.0228, -0.0228]$	$\Delta \phi_3 = [-0.8156, 0.8156]$	$\Delta \phi_3 = [-0.4784, 0.4784]$
$\Delta \phi_4 = [0.0228, -0.0228]$	$\Delta \phi_4 = [-0.8156, 0.8156]$	$\Delta \phi_4 = [-0.4784, 0.4784]$

5.4 EXAMPLE E: STACKED BLOCK ASSEMBLY – LINEAR

Suppose that the limits of angle variation in previous stacked block assembly example are known, the tolerance analysis problem is then reduced to linear equation solving. This linear problem can be solved using the Jacobi algorithm, and the result is interpretable, as listed in Table 5.

Table 5. Linear problem in stacked block assembly

Known size variation	$a = 6.62 \pm 0.2$ $b = 6.805 \pm 0.075$ $c = 10.675 \pm 0.125$ $d = 4.06 \pm 0.15$ $e = 24.22 \pm 0.35$ $f = 3.905 \pm 0.125$
Known kinematic variation	$\phi_1 = 74.7243 \pm 0.4281$ $\phi_2 = -74.7243 \pm 0.4281$ $\phi_3 = -105.2761 \pm 0.4281$ $\phi_4 = -105.2761 \pm 0.4281$
Unknown kinematic variation	$u_1 = 18.7181 \pm ?$ $u_2 = 8.6705 \pm ?$ $u_3 = 10.0477 \pm ?$ $u_4 = 2.1894 \pm ?$ $u_5 = 27.2965 \pm ?$
Linear equations	$\begin{cases} -u_1 + u_2 \sin(90 + \phi_2) + u_3 + a \sin(180 + \phi_2) = 0 \\ u_2 \cos(90 + \phi_2) + a \cos(180 + \phi_2) - a = 0 \\ u_3 + u_4 \sin(\phi_2 + 90) + b \sin(\phi_2) - d = 0 \\ u_4 \cos(\phi_2 + 90) + b \cos(\phi_2) - f = 0 \\ u_5 \cos(\phi_2 + 90) + b \cos(\phi_2) - e - f = 0 \end{cases}$

	$\begin{bmatrix} -1 & [0.2562, 0.2707] & 1 & 0 & 0 & 0 \\ 0 & [0.9626, 0.9667] & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & [0.2562, 0.2707] & 0 & 0 \\ 0 & 0 & 0 & [0.9626, 0.9667] & 0 & 0 \\ 0 & 0 & 0 & 0 & [0.9626, 0.9667] & 0 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{bmatrix} = \begin{bmatrix} [-6.1804, -6.5922] \\ [8.6659, 8.0652] \\ [10.8602, 10.3888] \\ [2.3054, 1.9179] \\ [26.8754, 25.7879] \end{bmatrix}$
Result of Jacobi algorithm	$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{bmatrix} = \begin{bmatrix} [18.7024, 18.7335] \\ [9.0026, 8.34302] \\ [10.2466, 9.85174] \\ [2.39497, 1.98397] \\ [27.9196, 26.6762] \end{bmatrix}$
Interpretation of result	$U(u_1, [18.7024, 18.7335])U(a, [6.42, 6.82])U(b, [6.73, 6.88])U(c, [10.55, 10.8])U(d, [3.91, 4.21])$ $U(e, [23.87, 24.57])U(f, [3.78, 4.03])U(\phi_1, [74.2962, 75.1524])U(\phi_2, [-75.1524, -74.2962])$ $U(\phi_3, [-105.7042, -104.8480])U(\phi_4, [-105.7042, -104.8480])$ $E(u_2, [8.34302, 9.0026])E(u_3, [9.85174, 10.2466])E(u_4, [1.98397, 2.39497])E(u_5, [26.6762, 27.9196])$ $\begin{cases} -u_1 + u_2 \sin(90 + \phi_2) + u_3 + a \sin(180 + \phi_2) = 0 \\ u_2 \cos(90 + \phi_2) + a \cos(180 + \phi_2) - a = 0 \\ u_3 + u_4 \sin(\phi_2 + 90) + b \sin(\phi_2) - d = 0 \\ u_4 \cos(\phi_2 + 90) + b \cos(\phi_2) - f = 0 \\ u_5 \cos(\phi_2 + 90) + b \cos(\phi_2) - e - f = 0 \end{cases}$

6 Conclusion

A semantic tolerance modeling scheme based on modal interval is proposed to enrich the modeling and analysis structure for tolerances such that tolerancing semantics can be embedded in mathematic representation in order to support better design and manufacturing specifications.

The new semantic tolerancing method captures engineering and logic relation between specifications and prevents the degeneracy of engineering semantics during mathematic calculation. Priors and posteriori variations in tolerance specification are differentiated. The model captures the semantics of physical property difference between rigid and flexible materials as well as tolerancing intents such as sequence of specification, measurement, and assembly. Compared to traditional methods, the semantic tolerancing allows us to estimate true variation ranges such that feasible and complete solutions can be obtained.

Future research may include tolerance chain formulation with the consideration of geometric tolerances and interaction between tolerances, optimization approach to solve linear and nonlinear modal interval equations, as well as tolerance synthesis based on global optimization methods of interval analysis.

References

- Abrams, S.L., Cho, W., Hu, C.Y., Maekawa, T., Patrikalakis, N.M., Sherbrooke, E.C., and Ye, X. (1998) Efficient and reliable methods for rounded-interval arithmetic. *Computer-Aided Design*, **30**(8), 657-665
- ADCATS, ADCATS Bibliography for Tolerance Work, <http://adcats.et.byu.edu/WWW/bibliographies/DocList.html>
- Armengol, J., Vehi, J., Trave-Massuyes, L., and Sainz, M.A. (2001) Application of modal intervals to the generation of error-bounded envelopes. *Reliable Computing*, **7**(2), 171-185
- Ashlagbor, A., Liu, H. C., Nnaji, B.O. (1998) Tolerance control and propagation for the product assembly modeler. *International Journal of Production Research*, **36**(1), 75-94
- Bhide, S., Davidson, J.K., and Shah, J.J. (2003) A new mathematical model for geometric tolerances as applied to axes. in *ASME Proceedings of Design Engineering Technical Conferences / Design Automation Conference, Sept 2-6, 2003, Chicago, Illinois*, Paper No. DETC2003/DAC-48736
- Bihlmaier, B.F. (1999) *Tolerance Analysis of Flexible Assemblies Using Finite Element and Spectral Analysis*, M.S. thesis, Department of Mechanical Engineering, Brigham Young University
- Bourdet, P. and Ballot, E. (1995) Geometric behavior for computer aided tolerancing. in *CIRP/JSPE/ASME Proceedings of the 4th CIRP Seminar on Computer Aided Tolerancing, April 5-6, 1995, University of Tokyo, Tokyo, Japan*, pp.143-154
- Camelio, J., Hu, S.J., and Ceglarek, D. (2003) Modeling variation propagation of multi-station assembly systems with compliant parts. *ASME Journal of Mechanical Design*, **125**(4), 673-681
- Camelio, J., Hu, S.J., and Marin, S.P. (2004) Compliant assembly variation analysis using component geometric covariance. *ASME Journal of Manufacturing Science and Engineering*, **126**(2), 355-360
- Chase, K.W. and Greenwood, W.H. (1988) Design issues in mechanical tolerance analysis. *Manufacturing Review*, **1**(1), 50-59
- Chase, K.W., Magleby, S.P., Gao, J., and Sorensen, C.D. (1996) Including geometric feature variations in tolerance analysis of mechanical assemblies. *IIE Transactions*, **28**(10) 795-807
- Chase, K.W., Magleby, S.P., Gao, J. (1997) Tolerance analysis of two- and three- dimensional mechanical assemblies with small kinematic adjustments. in *Advanced Tolerancing Techniques*, ed. by H.C. Zhang, New York: John Wiley & Sons, pp.103-137
- Desrochers, A. and Riviere, A. (1997) A matrix approach to the representation of tolerance zones and clearances. *International Journal of Advanced Manufacturing Technology*, **13**, 630-636
- Davidson, J.K., Mujezinovic, A., and Shah, J.J. (2002) A new mathematical model for geometric tolerances as applied to round faces. *ASME Journal of Mechanical Design*, **124**(4) 609-622

- Duff, T. (1992) Interval arithmetic and recursive subdivision for implicit functions and Constructive Solid Geometry. *Computer Graphics*, **26**(2), 131-138
- Finch, W.W. and Ward, A.C. (1997) A set-based system for eliminating infeasible designs in engineering problems dominated by uncertainty. in *ASME Proceedings of Design Engineering Technical Conference, Sept.14-17, 1997, Sacramento, CA*, paper no. DETC97/DTM-3886
- Gao, J., Chase, K.W., and Magleby, S.P. (1995) Comparison of assembly tolerance analysis by direct linearization and modified Monte Carlo simulation methods, in *ASME Proceedings of the 1995 Design Technical Conference – 21st Design Automation Conference*, DE-Vol.82, pp.353-360
- Gao, J., Chase, K.W., and Magleby, S.P. (1998) General 3-D tolerance analysis of mechanical assemblies with small kinematic adjustments. *IIE Transactions*, **30**(4), 367-377
- Gardenes, E., Sainz, M.A., Jorba, L., Calm, R., Estela, R., Mielgo, H., and Trepal, A. (2001) Modal intervals. *Reliable Computing*, **7**(2), 77-111
- Gerth, R.J. (1997) Tolerance analysis: A tutorial of current practice. in *Advanced Tolerancing Techniques*, ed. by Zhang, H.C., New York: John Wiley & Sons, pp.65-99
- Giordano, M. and Duret, D. (1993) Clearance space and deviation space. in *CIRP Proceeding of the 3rd Seminars on Computer Aided Tolerancing, April 27-28, 1993, Cachan, France*, pp. 179-196
- Hansen, E.R. (1992) Bounding the Solution of Interval Linear Equations. *SIAM Journal on Numerical Analysis*, **29**(5), 1493-1503
- Hong, Y.S. and Chang, T.-C. (2002) A comprehensive review of tolerancing research. *International Journal of Production Research*, **40**(11), 2425-2459
- Inui, M., Otto H., and Kimura, F. (1993) Algebraic interpretation of geometric tolerances for evaluating geometric uncertainties in solid modeling. in *ACM Proceedings of the 2nd Symposium on Solid Modeling and Applications*, pp.377-386
- Joskowicz, L., Sacks, E., and Srinivasan, V. (1997) Kinematic tolerance analysis. *Computer-Aided Design*, **29**(2), 147-157
- Liu, S. and Hu, S. (1997) Variation simulation for compliant sheet metal assemblies using finite element methods. *ASME Journal of Manufacturing Science and Engineering*, **119**(3), 368-374
- Liu, S.C., Hu, S.J., and Woo, T.C. (1996) Tolerance analysis for sheet metal assemblies. *ASME Journal of Mechanical Design*, **118**, 62-67
- Long, Y. and Hu, S. (1998) A unified model for variation simulation of sheet metal assemblies. in *Geometric Design Tolerancing: Theories, Standards, and Applications*, ed. by H.A. Elmaraghy, London: Chapman & Hall, pp.208-219
- Martinsen, K. (1995) Statistical process control using vectorial tolerancing. in *CIRP/JSPE/ASME Proceedings of the 4th CIRP Seminar on Computer Aided Tolerancing, April 5-6, 1995, University of Tokyo, Tokyo, Japan*, pp.195-210

- Merkley, K.G. (1998) *Tolerance Analysis of Compliant Assemblies*, Ph.D. thesis, Department of Mechanical Engineering, Brigham Young University
- Merkley, K.G., Chase, K.W., Perry, E. (1996) An introduction to tolerance analysis of flexible assemblies. in *Proceedings of the 1996 MSC World Users Conference*, Newport Beach, CA, MacNeal_Schwendler Corp.
- Moore, M. and Wilhelms, J. (1988) Collision detection and response for computer animation. *Computer Graphics*, **22**(4), 289-298
- Mudur, S.P. and Koparkar, P.A. (1984) Interval methods for processing geometric objects. *IEEE Computer Graphics & Applications*, **4**(2), 7-17
- Muhanna, R.L. and Mullen, R.L. (1999) Formulation of fuzzy finite-element methods for solid mechanics problems. *Computer-Aided Civil & Infrastructure Engineering*, **14**, 107-117
- Muhanna, R.L. and Mullen, R.L. (2001) Uncertainty in mechanics problems - interval-based approach. *ASCE Journal of Engineering Mechanics*, **127**(6), 557-566
- Muhanna, R.L., Mullen, R.L., and Zhang, H. (2004) Interval finite element as a basis for generalized models of uncertainty in engineering mechanics. in *Proceedings of NSF Workshop on Reliable Engineering Computing (REC'04), September 15-17, 2004, Georgia Institute of Technology, Savannah, GA*, ed. by R.L. Muhanna and R.L. Mullen, pp.353-370
- Mujezinovic, A., Davidson, J.K., and Shah, J.J. (2004) A new mathematical model for geometric tolerances as applied to polygonal faces. *ASME Journal of Mechanical Design*, **126**(3) 504-518
- Neumaier, A. (1990) *Interval Methods for Systems of Equations* (Cambridge: University Press)
- Nigam, S.D. and Turner, J.U. (1995) Review of statistical approaches to tolerance analysis. *Computer-Aided Design*, **27**(1) 6-15
- Ning, S. and Kearfott, R.B. (1997) A Comparison of Some Methods for Solving Linear Interval Equations. *SIAM Journal on Numerical Analysis*, **34**(4), 1289-1305
- Popova, E.D. (2001) Multiplication distributivity of proper and improper intervals. *Reliable Computing*, **7**(2), 129-140
- Rao, S.S. and Berke, L. (1997) Analysis of uncertain structural systems using interval analysis. *AIAA Journal*, **35**(4), 727-735
- Rao, S.S. and Cao, L. (2002) Optimum design of mechanical systems involving interval parameters. *ASME Journal of Mechanical Design*, **124**, 465-472
- Requicha, A.A.G. (1983) Toward a theory of geometric tolerancing. *International Journal of Robotics Research*, **2**(2), 45-60
- Rivest, L., Fortin, C., and Morel, C. (1994) Tolerancing a solid model with a kinematic formulation. *Computer-Aided Design*, **26**(6), 465-476
- Roy, U. and Li, B. (1998) Representation and interpretation of geometric tolerances for polyhedral objects – I Form tolerances. *Computer-Aided Design*, **30**(2), 151-161
- Roy, U. and Li, B. (1999) Representation and interpretation of geometric tolerances for polyhedral objects – II Size, orientation and position tolerances. *Computer-Aided Design*, **31**(4), 273-285

- Sacks, E. and Joskowicz, L. (1998) Parametric kinematic tolerance analysis of general planar systems. *Computer-Aided Design*, **30**(9), 707-714
- Sainz, M.A., Gardenes, E., and Jorba, L. (2002a) Formal solution to systems of interval linear or non-linear equations. *Reliable Computing*, **8**(3), 189-211
- Sainz, M.A., Gardenes, E., and Jorba, L. (2002b) Interval estimations of solution sets to real-valued systems of linear or non-linear equations. *Reliable Computing*, **8**(4), 283-305
- Shen, G. and Patrikalakis, N.M. (1998) Numerical and geometric properties of interval B-Splines. *International Journal of Shape Modeling*, **4**, 31-62
- Snyder, J. (1999) *Generative Modeling for Computer Graphics and CAD: Symbolic Shape Design Using Interval Analysis*, Cambridge: Academic Press
- Srinivasan, V. and O'Connor, M.A. (1994) On interpreting statistical tolerancing. *Manufacturing Review*, **7**(4), 304-311
- Takezawa, N. (1980) An improved method for establishing the process wise quality standard", *Reports of Statistical and Applied Research*, Union of Japanese Scientists and Engineers (JUSE), **27**(3), 63-76
- Teck, T.B., Senthil Kumar, A., and Subramaian, V. (2001) A CAD integrated analysis of flatness in a form tolerance zone. *Computer-Aided Design*, **33**(11) 853-865
- Toth, D.L. (1985) On ray tracing parametric surfaces. *Computer Graphics*, **19**(3), 171-179
- Tuohy, S.T., Maekawa, T., Shen G., and Patrikalakis, N.M. (1997) Approximation of measured data with interval B-Splines. *Computer-Aided Design*, **29**(11), 791-799
- Turner, J.U. and Wozny, M.J. (1987) Tolerances in computer-aided geometric design. *The Visual Computer*, **3**, 214-226
- Wallner, J., Krasauskas, R., and Pottmann, H. (2000) Error propagation in geometric constructions. *Computer-Aided Design*, **32**(11), 631-641
- Wang, Y. (2004) Solving interval constraints in computer-aided design. in *Proceedings of NSF Workshop on Reliable Engineering Computing (REC'04), September 15-17, 2004, Georgia Institute of Technology, Savannah, GA*, ed. by R.L. Muhanna and R.L. Mullen, pp.251-267
- Wang, Y. and Nnaji, B.O. (2006) Solving interval constraints by linearization in computer-aided design. *Reliable Computing*, accepted
- Whitney, D.E., Gilbert, O.L., and Jastrzebski, M. (1994) Representation of geometric variations using matrix transforms for statistical tolerance analysis in assemblies. *Research in Engineering Design*, **6**, 191-210
- Wirtz, A., Gachter, C., and Wipf, D. (1993) From unambiguously defined geometry to the perfect quality control loop. *Annals of the CIRP*, **42**(1) 615-618
- Yang, C.C., Marefat, M.M., and Ciarallo, F.W. (2000) Interval constraint networks for tolerance analysis and synthesis. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **14**, 271-287
- Zhang, C., Luo, J., and Wang, B. (1999) Statistical tolerance synthesis using distribution function zones. *International Journal of Production Research*, **37**(17), 3995-4006

Why are intervals and imprecision important in engineering design?

J. M. Aughenbaugh and C. J. J. Paredis

Systems Realization Laboratory
Georgia Institute of Technology
Atlanta, GA 30308
www.srl.gatech.edu
jasona@gatech.edu or chris.paredis@me.gatech.edu

Abstract: It is valuable in engineering design to distinguish between two different types of uncertainty: inherent variability and imprecision. While variability is naturally random behavior in a physical process or property, imprecision is uncertainty that is due to a lack of knowledge or information. There are many sources of imprecision in design. Sequential decision making introduces imprecision because the results of future decisions are unknown. Statistical data from finite samples of environmental factors are inherently imprecise. Bounded rationality leads to imprecise subjective probabilities. Expert opinions and judgments often are imprecise due to a lack of information or conflict. Behavioral simulations and analysis models are imprecise abstractions of reality. Knowledge of a decision maker's preferences may be imprecise due to bounded rationality or other constraints. Consequently, the engineering design community needs efficient computational methods for interval data and imprecise probabilities in order to support decision making in the design process. This paper introduces these sources and needs, with the aim of forming a foundation for future collaboration with the reliable engineering computing community.

Keywords: imprecision, imprecise probabilities, probability boxes, p-boxes, uncertainty, engineering design, intervals

1. Introduction

The goal of this paper is to introduce the needs of the engineering design community for computations with intervals and imprecise probabilities to the reliable engineering computing community. Earlier work has demonstrated the value of using imprecise probabilities in engineering design (Aughenbaugh and Paredis 2005), the role of imprecise probabilities in applying information economics (Aughenbaugh, Ling et al. 2005), and the elimination of decision alternatives using interval comparisons (Rekuc, Aughenbaugh et al. 2006). However, significant computational challenges are faced in implementing these methods in applied problems.

© 2006 by authors. Printed in USA.

Consequently, the established expertise of the reliable engineering community in these areas could be very valuable in engineering design. By introducing the needs and context of engineering design problems, we hope to foster future collaboration between the design community and the reliable engineering computing community.

Section 2 provides an overview of the design process, including its structure and challenges. The third section describes sources of interval data and imprecise probabilities, together referred to as *imprecision*, in engineering design. The fourth section provides a brief overview of the computational challenges faced in engineering design due to imprecision.

2. The engineering design process

Design is a process of converting information about customer interests and requirements into a specification of a product. This process involves searching through a very large, unstructured space of solutions (Tong and Sriram 1992) based on vague and uncertain knowledge about possible solution alternatives (Gupta and Xu 2002), their physical behavior (Aughenbaugh and Paredis 2004), their cost (Garvey 1999), and the decision maker's preferences (Kirkwood and Sarin 1985; Otto and Antonsson 1992; Carnahan, Thurston et al. 1994; Seidenfeld, Schervish et al. 1995). In order to guide engineers through this process, several approaches have been developed. In this paper, we introduce the general model of *systematic design* described by Pahl and Beitz (1996).

2.1. SYSTEMATIC DESIGN

In systematic design, the design process is broken into four main phases, as summarized in *Table 1*. In the *product planning and clarification of task phase*, a need for a product is determined and described. Product planning is mostly in the domain of corporate strategy and marketing; a company's situation and market condition are analyzed, profitable product ideas sought, and a product proposal made. The next step is to clarify the task by refining the product proposal and creating a detailed requirements list for the product. These requirements tell engineers what a product should be, should not be, and what it must be (at a minimum) in order to be successful. Once a list of requirements and objectives is created, conceptual design can begin.

The conceptual design phase takes the list of requirements and objectives and determines the principle solution structures to be pursued in embodiment design. To some, this is where traditional engineering begins. First, designers distill the problem down to its core, asking *what are we really trying to build?* Then they identify what functions (for example in a car design, functions such as *move person*, *protect person*, *monitor performance*) the design must perform and how these functions interact at a high level, such as transfers of energy, mass, and information. All of this information is combined into a function structure. Next, designers seek to enumerate possible physical implementations, or working principles, for each function. For example, three working principles for the function *mark a piece of paper* could be *deposit*

material by friction (e.g. a pencil), *melt material onto paper* (e.g. laser jet printing), or *burn away material* (e.g. scorching the paper with a laser). Since in general there are multiple functions, each with multiple working principles, they can be combined into an overall product in many different ways, or solution variants. Finally, these solution variants must be evaluated and a principal solution *concept* chosen. This concept forms the foundation for embodiment design.

In embodiment design, designers develop the design concept in more detail by considering additional technical and economic criteria. Essentially, embodiment design takes the working principles and concepts developed in conceptual design and develops an actual design specification, at which point detail design can lead directly into production. During detail design the arrangement, dimensions, materials, and production methods of all parts of the product are finalized and documented.

Table 1. Systematic Design Phases

Phase	Main tasks
Planning and clarifying the task	Investigation into the economic and technical viability of creating a given product, and the definition of the exact requirements of a system and the criteria surrounding its functioning.
Conceptual design	Development of function structure and the evaluation of different solution variants to this problem.
Embodiment design	Conversion of a conceptual working structure to a specification of layout.
Detail design	Finalization of the design and production details.

2.2. PARTITIONING THE DESIGN PROBLEM

Complex problems can rarely, if ever, be solved globally in one step. Most products have reached a level of complexity at which it is infeasible for one engineer or even engineers from one discipline to design them completely. Instead, the design problem must be broken down into smaller chunks that are designed by separate design teams. The solutions to these sub-problems are then synthesized and integrated into a complete design for the overall system. Systematic design is an appropriate approach for designing a product at one level of detail, but it does not address this higher-level process of decomposing a system into subsystems, concurrently designing subsystems, and subsequently integrating subsystem designs into the overall system. A holistic, hierarchical decomposition approach to the design process that addresses these problems is provided by *systems engineering* (Forsberg and Mooz 1992; Buede 2000; Forsberg, Mooz et al. 2000; Blanchard 2004). This paper will not address systems engineering formally. However, it is useful to consider what happens when the design task is broken into sub-problems.

When the design process is sub-divided, it becomes recursive—the overall design process is a sequence of design sub-problems. For example, consider the design of a car. A car can be broken down into many subsystems (such as engine, drivetrain, wheels, chassis, and so on), and each of these subsystems can be broken down into smaller subsystems, as simply shown in Figure 1. In many cases, a different team of engineers will perform the embodiment of each subsystem. Teams may also work on sub-problems concurrently, rather than sequentially. For example, one team may be designing the drivetrain while another team is designing the engine.

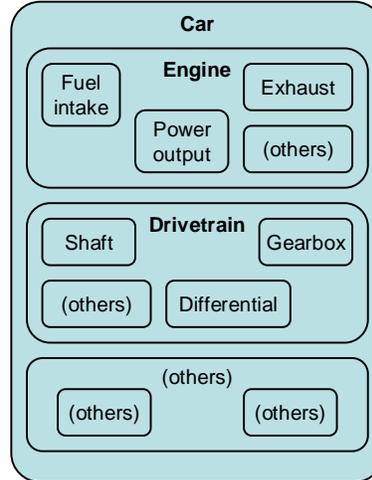


Figure 1. Subsystems of a car

When a team is formed to design the engine, its members first must clarify their task by using their technical expertise to elaborate on the requirements. For example, a particular engine concept is one of the working solutions from the conceptual design (Phase 2) of the car, as shown in Figure 2. Part of this engine design process is subdividing the engine into its subsystems, such as the fuel intake, and so on down to the smallest component of the system. This design process is challenging because the performance of the overall system may be a function of the interactions between sub-systems. Thus, the decisions of one team depend on future decisions and on decisions made concurrently by other design teams. In such situations, engineers can adopt robust design strategies (Chen, Allen et al. 1996) or set-based design approaches (Sobek, Ward et al. 1999; Rekuć, Aughenbaugh et al. 2006). In either case, each team recognizes that the decisions of other teams are uncertain, treating them as random variables, intervals, or sets. The nature of these uncertainties is addressed in Section 3 of this paper. First, the importance of decision making in the design process is discussed.

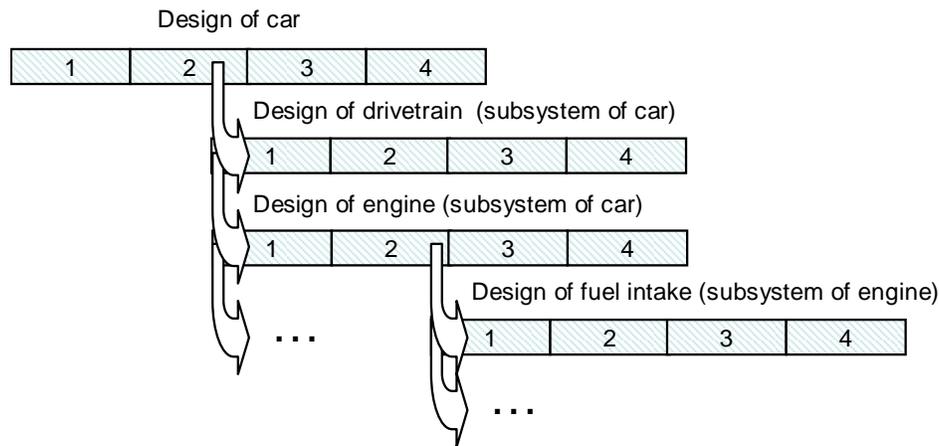


Figure 2. Recursive design process, design phases numbered 1-4

2.3. DECISION-BASED DESIGN

Independent of the design process chosen, designers repetitively must identify problems, search for solutions, evaluate solutions, and choose a final design. Inspired by this process, decision-based design recognizes that the principal role of an engineer in the design process is to make decisions (Mistree, Smith et al. 1990; Hazelrigg 1998; Marston, Allen et al. 2000). This paradigm shifts the emphasis of design research to decision making; one way to improve the design process is to enable engineers to make better decisions.

Engineers must make decisions while subject to many constraints, including limits on human cognitive abilities, or what Herbert Simon describes as bounded rationality (1947). According to Simon, humans cannot simultaneously consider all consequences of every alternative; there is a limit to how much a person can consider at one time. In engineering design, the problems of bounded rationality are exacerbated by the nature of the design process. For example, there are usually multiple people working on a single problem, and these people may be distributed in different geographic locations, organizational divisions, and technical disciplines. Consequently, it is very difficult for the right person to have the right information available at the right time in a format that he or she can comprehend (Cooper 2003). In many cases information, such as future decisions or concurrent decisions made by other design teams, is just inherently unavailable.

Finally, and perhaps most obviously, engineers have finite resources, such as time and money. Consequently, they cannot study every detail of every subsystem extensively. Decisions often are guided with approximate models, expert opinion, rules of thumb, and even pure intuition. One goal of decision-based design is to support these decisions with formal methods. It

has been recommended by some researchers in the engineering design community to base these methods on traditional statistical decision theory (Pratt, Raiffa et al. 1995), in which uncertainty is represented using precise probability distributions. However, it is important in engineering design to distinguish between two different types of uncertainty: inherent variability and imprecision (Parry 1996; Nikolaidis, Chen et al. 2004; Aughenbaugh and Paredis 2005).

2.4. VARIABILITY AND IMPRECISION

Variability, also called aleatory uncertainty (from the Latin *aleator* = dice thrower), is naturally random behavior in a physical process or property (Oberkampf, DeLand et al. 2002; Haukaas 2003). It is also known as objective uncertainty (Ferson and Ginzburg 1996) and irreducible uncertainty (Der Kiureghian 1989). Examples include manufacturing error, errors in communication systems, and radioactive decay. Inherent variability is best represented in stochastic terms, e.g., by a probability density function.

Imprecision, on the other hand, is due to a lack of knowledge or information (Parry 1996) and sometimes is called epistemic uncertainty (from the Greek *episteme* = knowledge), reducible uncertainty (Der Kiureghian 1989) or subjective uncertainty (Ferson and Ginzburg 1996). Imprecision is generally best represented in terms of intervals (Kreinovich, Ferson et al. 1999; Muhanna and Mullen 2004). While some authors doubt the philosophical distinction between aleatory uncertainty and imprecision, such distinctions are useful in practice (Ferson and Ginzburg 1996; Hofer 1996; Winkler 1996; Aughenbaugh and Paredis 2005).

The role of imprecision in engineering design is often overlooked, at least in part due to practical reasons—engineers do not know how to compute and make decisions effectively with imprecise information. They instead assume away or ignore imprecision. Since methods for representing and computing with imprecise information are research topics in the reliable engineering computing and imprecise probability communities, it is important to demonstrate the need for interval and imprecise methods in engineering design to these communities. Ideally, with a new understanding of the needs of engineers, researchers in these areas can help explain these methods to the design community and work with designers to expand these methods to meet the needs of engineering design practice.

3. Sources of imprecision in engineering design decisions

In this paper, the sources of imprecision in engineering design are considered in the context of the simplified design model illustrated in Figure 3. The partitioning of the design problem into sub-problems results in a sequence of decisions (for simplicity, concurrent decisions are ignored), of which one is shown in detail in Figure 3. In this simple example, a designer, or decision maker (hereafter abbreviated as DM), has two decision alternatives. Based on characteristics of the alternatives and environmental factors, the DM performs multiple simulations (S_i) or other analyses (A_i), including eliciting expert opinion, to study the performance of the alternatives.

Performance attributes are then combined or weighted according to the DM's preferences (perhaps according to utility theory), and the most preferred alternative is selected (or alternatively when there are more than two decisions alternatives, the DM can proceed by selecting a set of the more preferred alternatives (Rekuc, Aughenbaugh et al. 2006)).

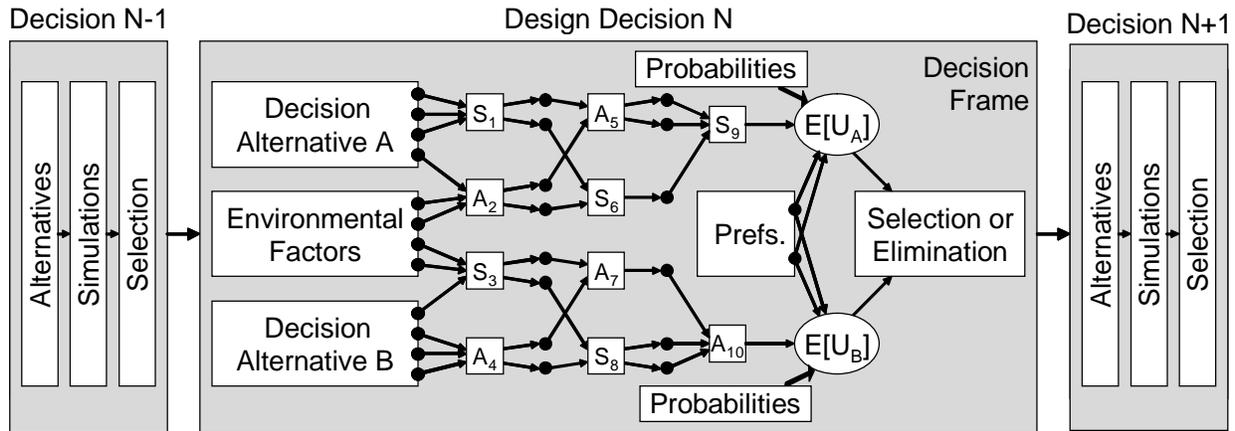


Figure 3. A sequential decision process in simulation-based design.

Almost every aspect of this decision introduces imprecision. More specifically:

- Sequential decision making introduces imprecision because the results of future decisions are unknown.
- Statistical data from finite samples of environmental factors are inherently imprecise.
- Bounded rationality leads to imprecise subjective probabilities.
- Expert opinion and judgments are not precise, due to lack of information or conflict.
- Behavioral simulations and analysis models are imprecise abstractions of reality.
- Preferences may be imprecise due to bounded rationality or non-stationarity.
- Numerical implementation of these models introduces additional imprecision.

In the following sub-sections, we elaborate on how these sources introduce imprecision into the design process.

3.1. SEQUENTIAL DECISION MAKING

As noted earlier, the complexity of the design problem makes it impossible to arrive at an optimal design in one step. Instead, the process is divided into a sequence of decisions. This process is illustrated using a simple design problem with two design variables: vehicle type and engine type. There are two options for vehicle type: car or bike. There are three options for engine type:

gasoline engine, diesel engine, or electric motor. If the DM chooses the design in one step, he or she would choose from the set of six *design alternatives* shown in Figure 4. In the context of this example, each of these design alternatives is a fully detailed design of a final product.

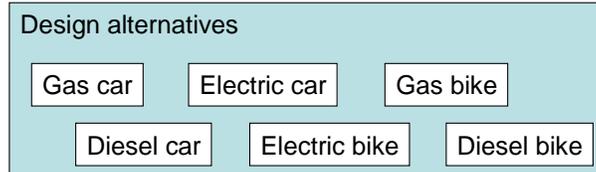


Figure 4. One stage decision

In order to choose the best design out of these six, the DM would need to evaluate and compare all six. While easy in this simple example, it is impractical to enumerate and evaluate all design alternatives by considering all possible combinations of all solution principles for all the subsystems of a complex product. Consequently, the decisions are broken down into sequences to allow for efficient exploration of the design space. For example, in the previous vehicle design example, a DM can follow a sequential approach in which he or she first chooses the vehicle type, and then the engine type, as shown in Figure 5.

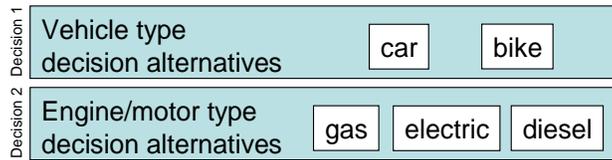


Figure 5. Sequential decisions

Note that it is important here to distinguish clearly between *decision* alternatives and *design* alternatives. A design alternative is one of the possible complete product design specifications (recall Figure 4), while each decision alternative is a specific option for a specific decision and corresponds to a set of design alternatives. For example, when choosing the vehicle type, the DM has two *decision alternatives*: car or bike. Each of these decision alternatives actually corresponds to a *set of design alternatives*, as shown in Figure 6.

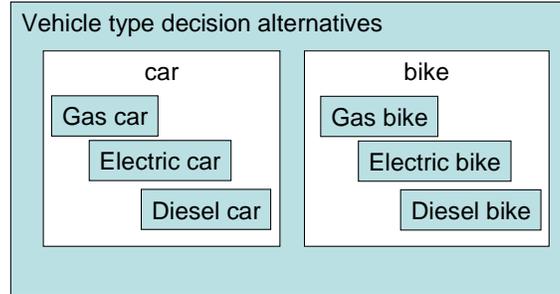


Figure 6. Sets of design alternatives

The choice of decision alternative *car* for vehicle type includes the gas car, diesel car, and electric car design alternatives, because the vehicle type decision will be followed by the engine type decision. Once a decision is made to pursue, for example, a car design rather than a bike, the DM does not need to consider explicitly the design alternatives gas bike, electric bike, and diesel bike; these design alternatives are *eliminated* from consideration.

One limitation of a sequential decision process is that decisions often are coupled. In general, one really needs to know the outcome of future decisions to select the best (or most preferred) decision alternative for the current decision. For example, a fully designed car will have a certain maximum horsepower, but this certain value is unknown when the vehicle type decision is made, because it depends on the future design decision of engine type. The set of car designs in Figure 6 has multiple horsepower maxima, each corresponding to a sub-design (gas car, electric car, and diesel car). Thus, when selecting type *car* rather than *bike*, a DM is not selecting a precisely characterized horsepower, but rather a set or interval of horsepower. In a more complex problem, imprecision will remain once the engine type is chosen because a particular engine type is a set of designs. For example, even if a gas engine is chosen, characteristics such as horsepower, torque, mass, and fuel efficiency will be inherently imprecise because they depend on additional details of the design.

By itself, the inherent existence of sets in sequential decision making demonstrates the need to compute with intervals, sets, or otherwise imprecisely characterized information. However, other sources of imprecision are independent of the existence of sets of design alternatives. These may have different characteristics and may affect the design process differently, as described in the following.

3.2. STATISTICAL DATA

Engineers frequently gather statistical data about environmental or other factors to support design decisions. Such quantitative data gives an illusion of being well-characterized, but actually it is inherently imprecise. Assume one needs to design a pressure vessel, and the vessel will be made

of a new type of steel for which the yield strength X is not well characterized. Engineers have strong theoretical evidence that the material strength is normally distributed, but they do not know the mean μ or variance σ^2 of the distribution. Because the material is new and testing is relatively expensive, DMs have only measured the yield strength in a set Σ of n independent tension tests, where n is a relatively small number due a high cost of testing. These tests can at best give an estimate of the true distribution, so in addition to inherent randomness (irreducible uncertainty), engineers also face imprecision—they cannot characterize the parameters of the random variable precisely.

For example, assume the engineers have a set of 30 material strength measurements. They could use the 30 samples to estimate the true mean and variance of the distribution using standard statistics. However, these estimates ($\hat{\mu}$ and $\hat{\sigma}^2$) are exactly that—*estimates*. The resulting distribution $X \sim N(\hat{\mu}, \hat{\sigma}^2)$ in general is *not* the true distribution. Alternatively, confidence intervals can be constructed on the true mean and variance at the α confidence level as follows, where n is the number of samples and s is the sample standard deviation (Hines, Montgomery et al. 2003):

$$[\underline{\mu}, \bar{\mu}] = \left[\hat{\mu} - t_{\alpha/2, n-1} \frac{s}{\sqrt{n}}, \hat{\mu} + t_{\alpha/2, n-1} \frac{s}{\sqrt{n}} \right] \quad (1)$$

$$[\underline{\sigma}^2, \bar{\sigma}^2] = \left[\frac{(n-1)s^2}{\chi_{\alpha/2, n-1}^2}, \frac{(n-1)s^2}{\chi_{1-\alpha/2, n-1}^2} \right]. \quad (2)$$

The resulting structure:

$$X \sim N([\underline{\mu}, \bar{\mu}], [\underline{\sigma}^2, \bar{\sigma}^2]) \quad (3)$$

is a probability box, or p-box (Ferson and Donald 1998; Aughenbaugh, Ling et al. 2005). All normal distributions with means and variances given by Equations (1) and (2) are contained inside this p-box. Previous work has suggested that accounting for the imprecision in statistical data with p-boxes will lead, on average, to better design decisions for high-risk application (Aughenbaugh and Paredis 2005). However, there are computational challenges for using p-boxes, or more generally imprecise probabilities (Tintner 1941; Hart 1942; Levi 1974; Walley 1991; Weichselberger 2000), in complex engineering problems, as described in the briefly in Section 4 and elaborated on in detail in another paper in this workshop (Bruns, Paredis et al. 2006).

In this section, we focused on statistical data, emphasizing a rather frequentist interpretation of probability. The *frequentist* interpretation is based on the notion of relative frequencies of outcomes. Under a frequentist interpretation, a probability represents the ratio of times that one outcome occurs compared to the total number of outcomes in a series of identical, repeatable, and possibly random trials. In engineering design, events are not always repeatable. Even assuming

some events are essentially repeatable and data can be collected, there is no guarantee that a particular sample is representative of the true relative frequency. Although in theory the relative sample frequency approaches the true relative frequency as the sample size goes to infinity, an infinite sample size is impossible to acquire in practice. Consequently, engineers will always face *imprecision* in their characterizations of the frequentist probabilities. Other times, it is inappropriate to adopt a purely frequentist view of probabilities in engineering design. Often, a *subjective* interpretation is more applicable.

3.3. IMPRECISE SUBJECTIVE PROBABILITIES

Proponents of a *subjective* interpretation of probability assert that there is no such thing as a true or objective probability, but rather probabilities are an expression of belief based on an individual's willingness to bet (de Finetti 1974; Lindley 1982; Winkler 1996). One of the subjectivists' primary arguments against a frequentist perspective is the absence of truly repeatable events, especially in practical problems. For example, the probability that Team A beats Team B in a basketball game has no real meaning under a frequentist interpretation, because that event—that particular game—will occur exactly once. In this context, the notion of a long term frequency, and even random events, is meaningless (de Finetti 1974). However, many people are willing to express their belief of who will win in terms of bets. When framed appropriately, such bets can be taken as subjective probabilities.

We prefer to adopt a loosely subjective interpretation of probability because true relative frequencies cannot be determined with any finite number of data samples, and because a subjective interpretation is applicable to a broader class of problems, as it is not limited to repeatable events. Naturally, subjective probabilities should be consistent with available information, including knowledge about observed relative frequencies (when applicable) and the DM's actual beliefs; such probabilities can be considered *rationalist* subjective probabilities (Walley 1991). Our interpretation is not as strict as the traditional views [see Lindley (1982) for a summary of the strict subjective tradition], because we admit imprecisely known subjective probabilities. The traditional school claims that by definition, subjective probabilities are known to a decision maker, because they *are* his or her beliefs. We prefer an interpretation that acknowledges the practical difficulties in arriving at a precise characterization of such beliefs.

The process of eliciting and assessing an individual's beliefs, or willingness to bet, is resource intensive. Even assuming that precise beliefs—and hence precise probabilities—exist, it will often be impractical to fully characterize them due to constraints such as bounded rationality, time, and computational ability (Weber 1987; Walley 1991; Groen and Mosleh 2005). Consequently, only a partial—and therefore imprecise—characterization of subjective probabilities is normally available.

The notion of imperfectly known probabilities is not new. Decision theory has long differentiated between decision making with known probabilities (*decision making under risk*)

and decision making without knowledge of probabilities (*decision making under uncertainty*) (Knight 1921). Since then, researchers have explored the middle ground of incomplete knowledge of probabilities, such as ordered probabilities (Fishburn 1964) and linear constraints on the probabilities (Kmietowicz and Pearman 1984), in addition to the more general imprecise probabilities. The ability to compute with such uncertainties is crucial to the success of engineering design.

3.4. EXPERT OPINION

A significant source of information in engineering design are experts who use their knowledge and experience to form judgments, beliefs, and estimates (Cooke 1991; Ayyub 2001). Information from expert opinions is inherently imprecise. First, opinions may not always be cited precisely, especially when expressed in linguistic terms, such as *unlikely*, *large*, or *poor*, a case in which fuzzy set theory has a role (Zadeh 1965; Ayyub 2001). Because an opinion about the world is not necessarily the truth of the world, opinions also can differ from person to person. Often, these opinions will conflict. For example, two experts are asked the probability that a quantity X is below 5; that is, $P\{X < 5\}$. The first expert says that $P\{X < 5\} = 0.3$ (and consequently $P\{X \geq 5\} = 0.7$). The second expert states that $P\{X < 5\} = 0.6$ (and consequently $P\{X \geq 5\} = 0.4$). The combination of such evidence, especially when conflicting, is an important research area, often focused on Evidence Theory (Dempster 1967; Shafer 1976; Yager, Kacprzyk et al. 1994; Oberkampf and Helton 2002; Mourelatos and Zhou 2005). Evidence Theory is a general theory that contains both traditional probability theory and possibility theory as special cases (Klir 1992). Consequently, interpretations of and methods for computing with evidence are of significant interest to engineering designers.

3.5. IMPRECISE ANALYSIS MODELS

An important step in decision making and design is to determine the DM's preferences over design alternatives. As illustrated in Figure 3, this involves the application of multiple models: simulation models that predict the performance of the alternatives, models for the uncertain inputs to these behavioral models, and models of the DM's preferences.

Behavioral models predict the performance of design alternatives in terms of attributes that are important to the DM, such as physical behavior, cost, and reliability. Since these models, like all models, are only an abstraction of reality, they are imprecise. Specifically, although the laws of physics are known very precisely, one often makes significant assumptions when applying the laws of physics to complex geometries, or one omits certain known—but less significant—physical phenomena from the model to reduce the complexity.

For example, a model for an internal combustion engine is often abstracted into an algebraic relationship between engine speed and torque. The detailed physical phenomena (including airflow, gas-mixture combustion, friction, and inertia) are reduced into one simple algebraic

relationship. This simple relationship is an idealization that may contain a significant error—the unknown or unmodeled relationships between a variety of parameters that play a role in the engine performance, such as air density, acceleration, or engine temperature. The lack of knowledge of the influence of these parameters on engine performance results in imprecision in the model's results. Since there is no probability distribution associated with such modeling and systematic errors, one cannot express the likelihood of occurrence for a particular error but can at best bound the size of the error, in which case the errors should be represented in terms of interval-based uncertainty.

In addition to the imprecision in the behavioral models themselves, there is often also significant imprecision in the parameter values or inputs to these models. For instance, the air resistance model of a car may include a drag coefficient, which can only be determined precisely through experimentation that is more extensive. Given the limited resources (cost, time, etc.) available for experimentation, the coefficient is only determined up to certain error bounds, which introduces additional imprecision in to the model predictions. There may also be stochastic environmental noise parameters. In this case, the uncertainty in the inputs can be modeled using imprecise probabilities or p-boxes; in addition to the inherent variability of such parameters, they will be imprecisely characterized, as described in the preceding sections for statistical data or subjective probabilities.

3.6. IMPRECISE PREFERENCES

Once the performance attributes of a particular design alternative have been determined, they are combined in a preference model to form a measure (such as expected utility) of the DM's overall preference for the specific alternative, as is illustrated in Figure 3. Keeney and Raiffa (1993) propose a method for developing such a preference model by eliciting preferences with respect to single attributes, expressing the preferences under uncertainty in utils, and then combining the utility functions of the multiple attributes into an overall utility function. Due to resource constraints, such a complete elicitation and precise characterization is unachievable in practice. Instead, the preference model is an imprecise abstraction based on limited preference elicitations. Other literature has examined incomplete or partial information [see (Weber 1987) for a review] in the context of imprecisely characterized preferences (Otto and Antonsson 1992; Carnahan, Thurston et al. 1994; Seidenfeld, Schervish et al. 1995) and unknown weights for tradeoffs between objectives in multi-attribute decision making (Kirkwood and Sarin 1985).

There is also evidence that people cannot express their preferences well in a rational fashion. When presented with choices between which a rational decision maker should be indifferent, even knowledgeable experts with a strong background in decision theory often judge the choices differently (Tversky and Kahneman 1974). This psychological evidence suggests that the environment and manner in which a choice is posed affects the elicited action, and thus choices are not a perfect indication of preference. It is also possible that preferences are non-stationary,

meaning they vary over time. Even if they are reasonably stationary over a relevant time horizon, practical and psychological evidence strongly suggest that preferences can only be modeled imprecisely.

3.7. NUMERICAL CALCULATIONS ARE IMPRECISE

This source of imprecision is probably the most familiar, but possibly the least significant. As is probably known to most readers, the precision of calculations implemented on a digital computer are only precise up to the machine's numerical precision. In practice, modern computers have a very high precision, and this effect is generally not important, especially in comparison to the other sources of imprecision in engineering design. For example, consider the use of a model to calculate some parameter. It often does not matter whether the numerical solution of this model is within 10^{-10} or 10^{-15} of the model's "true" answer, because the model being used is already imprecise; moving to 10^{-15} accuracy just means that one would know the model's wrong answer better; it provides no further insight into the true answer for the real system.

Imprecision also can arise with the use of numerical methods, which are used to approximate analytical solutions when analytical methods are unavailable. Some of these methods are not guaranteed to converge on the exact solution for certain problems, and thus introduce considerable uncertainty that an analyst must explore. Other methods converge on the true solution, but this convergence is not exact in most algorithms; there is usually a tolerance set in them as a stopping criterion. For example, an iterative method may terminate when the solution changes by less than some small amount over several iterations. Consequently, the solution is known imprecisely. While these computational issues are of some interest, it is again believed that the imprecision they introduce generally is inconsequential compared to imprecision from other sources. In order to provide value in engineering design, research in reliable engineering computing must address these more substantial sources of imprecision.

4. Challenges of designing with imprecise information

The presence of imprecision in engineering design decisions obviously demands methods for making decisions and calculating with imprecise information. The goal of this paper is only to explain the context of engineering design and the sources of imprecision in design problems. A companion paper (Bruns, Paredis et al. 2006) in this workshop addresses the challenges of decision making and computing with imprecise information in detail, and a forthcoming conference paper details decision policies for eliminating alternatives in a set-based approach to design (Rekuc, Aughenbaugh et al. 2006). We conclude this paper with a brief overview of the decision-making problem and some references for computations with intervals and imprecise probabilities for completeness.

4.1. CHALLENGES IN DECISION MAKING

In general, there are three possible scenarios of preference between alternatives A and B. Either A is preferred to B, B is preferred to A, or the DM is indifferent between A and B. When utilities are used to reflect preference, these relationships can be determined by the inequality or equalities of the expected utilities (von Neumann and Morgenstern 1944). However, when imprecision exists, the expected utilities are not known precisely and become intervals, as shown in Figure 7. Consequently, comparisons between the alternatives become more complicated.

For example, consider the intervals of expected utility for two alternatives (A and B) shown in Figure 7(a). In this example, the intervals overlap. Since the true expected utility of B can lie anywhere in the given interval, the point labeled b_1 is possible. Similarly, both a_1 and a_2 are possible true values for the expected utility of A. Notice that a_1 is greater than b_1 , but a_2 is less than b_1 . Consequently, the available evidence is *indeterminate*; the DM cannot determine which alternative is the most preferred, nor can the DM determine that he or she is definitely indifferent.

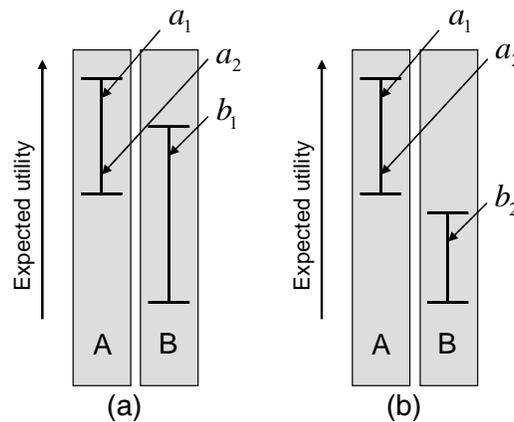


Figure 7. Intervals of expected utility

Given indeterminacy, a DM has two choices: he or she can collect more information in an effort to reduce the imprecision and remove the indeterminacy, or he or she can arbitrarily choose an alternative. *Arbitrary* means not uniquely determined by the DM's preferences, beliefs, and values (Walley 1991), but it does not necessarily imply without guidance or random. Several policies are possible to guide arbitrary choice, including Γ -maximin (Berger 1985) and the Hurwicz-criterion (Arrow and Hurwicz 1972). A Γ -maximin policy says that given indeterminacy in a maximization problem, a DM should select the alternative with the highest lower-bound. This is a conservative policy in that it seeks to mitigate the worst-case. Robust design strategies that choose solutions that are insensitive to imprecision are also applicable at

this stage. If the remaining uncertainty is extreme, it may be valuable to consider an alternative approach such as information gap theory (Ben-Haim 2001).

If a DM elects to continue collecting information, he or she will still need to compare imprecise information, such as intervals. When delaying a decision to collect more information, a designer is effectively adopting a set-based approach to design, in which multiple decision alternatives are considered in parallel. In this process, inferior decision alternatives are eliminated from the set under consideration as soon as they are determined to be less preferable than any other alternative. The simplest case of a clear choice between alternatives for which the DM's preferences are characterized by intervals of expected utility is shown in Figure 7(b). In this case, it does not matter where in the given interval the true expected utility of A falls—it will always be greater than any value in the interval for expected utility of B. This illustrates a situation referred to as *interval dominance*, (For a brief synopsis, see Zaffalon, Wesnes et al. 2003).

While interval dominance is simple to understand and implement, it will rarely be sufficient for eliminating alternatives. Instead, a DM must turn to policies such as *maximality* (Walley 1991) or *E-admissibility* (Levi 1974). The use of these policies is explained and demonstrated in much more detail in the forthcoming conference publication (Rekuc, Aughenbaugh et al. 2006). A summary of the topics is given here.

Consider five decision alternatives whose utility is expressed as a function of a single shared imprecise parameter (such five car designs whose performances all depend on the ambient air temperature) in Figure 8. The intervals for all of these alternatives overlap except for E and D, and hence only D can be eliminated according to interval dominance. Eliminations will have to be made using other criteria.

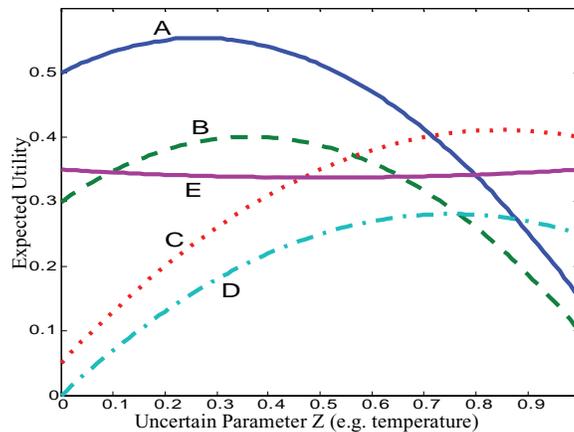


Figure 8. Performance of 5 alternatives influenced by a single uncertain parameter

From Figure 8, one can see that alternative A performs better than alternative B at all temperatures, so A is clearly better than B—an illustration of the criterion of maximality. Similarly, C is always better than D, so D could be eliminated even if it were not interval dominated by E. Consequently, neither alternatives B nor D can be the best decision, so the DM should no longer consider them.

Another way to look at Figure 8 is to ask which alternatives are ever the best, at any temperature. In this case, these are only alternatives A and C. This is an illustration of the E-admissibility criterion, which assumes that eventually all of the imprecision will be eliminated (the temperature will be known exactly), and then only the alternatives that are optimal at some temperature need to be considered. This may be true for some sources of imprecision (such as future decisions), but the DM should carefully consider the tradeoff between the value of obtaining more information and the cost of doing so by applying information economics (Aughenbaugh, Ling et al. 2005). Although the cost of additional investigation is often worth the improved ability to make a more informed decision, the DM will reach a point at which the cost of gathering additional information outweighs the expected benefits. Consequently, imprecision will rarely be eliminated, and the DM must resort to arbitrary choice.

In the case of arbitrary choice, it is desirable that robust alternatives be available. In Figure 8, alternative E is robust to temperature and would be a good arbitrary choice. However, alternative E is eliminated according to E-admissibility since it is never the optimal. Consequently, it appears that maximality is a better criterion than E-admissibility because maximality retains both the possible optimal solutions and the non-dominated robust solutions. In some cases, the imprecision can be reduced through additional analysis and design to the point that the optimal solution can be found, while in other cases a robust or otherwise arbitrary choice will need to be made. The key advantage of considering the alternatives in this manner is that the true optimal solution remains a candidate until late in the process, thus improving the chances of choosing it as the final design. The remaining question is *how can such intervals be calculated, propagated, and compared in a computationally efficient manner?*

4.2. COMPUTATIONAL CHALLENGES

For engineering applications, it is crucial to adopt a mathematical formalism that is convenient and inexpensive for computation and decision making. Various methods have been developed for propagating intervals (Moore 1979; Alefeld and Herzberger 1983; Kearfott and Kreinovich 1996) and imprecise distributions through known algebraic relationships (Springer 1979; Williamson and Downs 1990; Ferson and Ginzburg 1996; Berleant and Zhang 2004; Ferson and Hajagos 2004). However, many engineering models are black boxes—unknown or very complex relationships modeled by simulations or other means—for which algebraic relationships are unavailable. The only existing methods for these types of models are based on brute-force, multi-loop methods that include at least one Monte Carlo sampling loop. These methods are impractical

for engineering design because they are prohibitively expensive in terms of computations. Clearly, the methods that have been found effective for algebraic models must be extended, adapted, or replaced for computations in more general engineering design problems. This problem is formulated in substantially more mathematical detail in the companion paper (Bruns, Paredis et al. 2006).

5. Summary

There are many sources of imprecision in engineering design. The sequential nature of design decisions inherently leads to sets and intervals; probabilities and preferences are not known precisely, and models are imprecise approximations of reality. The presence of imprecision can lead to indeterminacy in decisions when traditional statistical decision theory is applied. Consequently, engineering researchers need to explore and develop new decision theories. The use of intervals and imprecise probabilities to capture a decision maker's state of knowledge also leads to new computational challenges. The potential benefit of using such formalisms is clear, but the feasibility of implementing them efficiently in complex design problems has not been proven. The development and application of efficient algorithms for computations with imprecise structures would help advance the state of engineering design significantly. In order to develop such methods, strong collaboration between the engineering design community and the reliable engineering computing community is needed. As a starting point for such collaboration, this paper has outlined the sources and role of imprecision in engineering design.

6. Acknowledgments

Jason Aughenbaugh is supported under a National Science Foundation Graduate Research Fellowship. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation. This work was supported in part by NSF grant DMI-0522116.

7. References

- Alefeld, G. and J. Herzberger. *Introduction to Interval Computations*. New York, Academic Press, 1983.
- Arrow, K. and L. Hurwicz. An Optimality Criterion for Decision-Making under Uncertainty. *Uncertainty and Expectation in Economics: Essays in Honour of G. L. S. Shackle*. C. F. Carter and J. L. Ford, Blackwell, 1972.
- Aughenbaugh, J. M., J. M. Ling and C. J. J. Paredis. Applying Information Economics and Imprecise Probabilities to Data Collection in Design. *2005 ASME International Mechanical Engineering Congress and Exposition*, Orlando, FL, USA, IMECE2005-81181, 2005.
- Aughenbaugh, J. M. and C. J. J. Paredis. The Role and Limitations of Modeling and Simulation in Systems Design. *2004 ASME International Mechanical Engineering Congress and Exposition*, Anaheim, CA, USA, IMECE2004-5981, 2004.

- Aughenbaugh, J. M. and C. J. J. Paredis. The Value of Using Imprecise Probabilities in Engineering Design. *ASME 2005 DETC DTM*, Long Beach, CA, USA, DETC2005-85354, 2005.
- Ayyub, B. *Elicitation of Expert Opinions for Uncertainty and Risks*. New York, CRC Press, 2001.
- Ben-Haim, Y. *Information Gap Decision Theory*. London, Academic Press, 2001.
- Berger, J. O. *Statistical Decision Theory and Bayesian Analysis*, Springer, 1985.
- Berleant, D. and J. Zhang. Representation and Problem Solving with Distribution Envelope Determination (Denv). *Reliability Engineering & System Safety* **85**(1-3): 153-168, 2004.
- Blanchard, B. S. *Systems Engineering Management*. Hoboken, NJ, John Wiley and Sons, Inc., 2004.
- Bruns, M., C. J. J. Paredis and S. Ferson. Computational Methods for Decision Making Based on Imprecise Information. *Reliable Engineering Computing Workshop*, Savannah, GA, USA, 2006.
- Buede, D. M. *The Engineering Design of Systems: Models and Methods*. New York, John Wiley & Sons, Inc., 2000.
- Carnahan, J. V., D. L. Thurston and T. Liu. Fuzzy Ratings for Multiattribute Design Decision-Making. *Journal of Mechanical Design, Transactions of the ASME* **116**(2): 511, 1994.
- Chen, W., J. K. Allen, D. N. Marvis and F. Mistree. A Concept Exploration Method for Determining Robust Top-Level Specifications. *Engineering Optimization* **26**: 137-158, 1996.
- Cooke, R. M. *Experts in Uncertainty: Opinion and Subjective Probability in Science*, Oxford University Press, 1991.
- Cooper, L. P. A Research Agenda to Reduce Risk in New Product Development through Knowledge Management: A Practitioner Perspective. *Journal of Engineering and Technology Management* **20**: 117-140, 2003.
- de Finetti, B. *Theory of Probability Volume 1: A Critical Introductory Treatment*. New York, Wiley, 1974.
- Dempster, A. P. Upper and Lower Probabilities Induced by a Multivalued Mapping. *The Annals of Statistics* **28**: 325-339, 1967.
- Der Kiureghian, A. Measures of Structural Safety under Imperfect States of Knowledge. *ASCE Journal of Structural Engineering* **115**(5): 1119-1139, 1989.
- Ferson, S. and S. Donald. Probability Bounds Analysis. *International Conference on Probabilistic Safety Assessment and Management (PSAM4)*, New York, NY, Springer-Verlag, 1998.
- Ferson, S. and L. R. Ginzburg. Different Methods Are Needed to Propagate Ignorance and Variability. *Reliability Engineering & System Safety* **54**(2-3): 133-144, 1996.
- Ferson, S. and J. Hajagos. Arithmetic with Uncertain Numbers: Rigorous and (Often) Best Possible Answers. *Reliability Engineering & System Safety* **85**(1-3): 135-152, 2004.
- Fishburn, P. *Decision and Value Theory*. New York, J. Wiley, 1964.
- Forsberg, K. and H. Mooz. The Relationship of Systems Engineering to the Project Cycle. *Engineering Management Journal* **4**(3): 36-43, 1992.
- Forsberg, K., H. Mooz and H. Cotterman. *Visualizing Project Management: A Model for Business and Technical Success*. New York, Wiley, 2000.
- Garvey, P. R. *Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective*. New York, Marcel Dekker, Inc., 1999.
- Groen, F. J. and A. Mosleh. Foundations of Probabilistic Inference with Uncertain Evidence. *International Journal of Approximate Reasoning* **39**(1): 49-83, 2005.

- Gupta, S. K. and C. Xu. Estimating the Optimal Number of Alternatives to Be Explored in Large Design Spaces: A Step Towards Incorporating Decision Making Cost in Design Decision Models. *ASME 2002 DETC CIE*, Montreal, Canada, DETC2002/CIE-34491, 2002.
- Hart, A. G. Risk, Uncertainty and the Unprofitability of Compounding Probabilities. *Studies in Mathematical Economics and Econometrics*. O. Lange, F. McIntyre and T. O. Yntema. Chicago, University of Chicago Press, 1942.
- Haukaas, T. *Types of Uncertainty, Elementary Data Analysis, Set Theory*. University of British Columbia. Vancouver, B.C., 2003.
- Hazelrigg, G. A. A Framework for Decision-Based Design. *Journal of Mechanical Design* **120**(4): 653-658, 1998.
- Hines, W. W., D. C. Montgomery, D. M. Goldsman and C. M. Borror. *Probability and Statistics in Engineering*. Hoboken, NJ, John Wiley & Sons, Inc., 2003.
- Hofer, E. When to Separate Uncertainties and When Not to Separate. *Reliability Engineering & System Safety* **54**(2-3): 113-118, 1996.
- Kearfott, R. B. and V. Kreinovich. *Applications of Interval Computations*. AH Dordrecht, The Netherlands, Kluwer Academic Publishers, 1996.
- Keeney, R. L. and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. New York, NY, Cambridge University Press, 1993.
- Kirkwood, C. W. and R. K. Sarin. Ranking with Partial Information: A Method and an Application. *Operations Research* **33**: 38-48, 1985.
- Klir, G. J. Probabilistic Versus Possibilistic Conceptualization of Uncertainty. *Analysis and Management of Uncertainty: Theory and Applications*. B. M. Ayyub, M. M. Gupta and L. N. Kanal. New York, North-Holland, 1992.
- Kmietowicz, Z. W. and A. D. Pearman. Decision Theory, Linear Partial Information and Statistical Dominance. *Omega* **12**: 391-399, 1984.
- Knight, F. H. *Risk, Uncertainty and Profit*. Boston, Houghton Mifflin, 1921.
- Kreinovich, V., S. Ferson, L. Ginzburg, H. Schulte, M. Barry and H. Nguyen. From Interval Methods of Representing Uncertainty to a General Description of Uncertainty. *International Conference on Information Technology*, Bhubaneswar, India, McGraw-Hill, 1999.
- Levi, I. On Indeterminate Probabilities. *Journal of Philosophy* **71**: 391-418, 1974.
- Lindley, D. V. Subjectivist View of Decision Making. *European Journal of Operational Research* **9**(3): 213, 1982.
- Marston, M., J. K. Allen and F. Mistree. The Decision Support Problem Technique: Integrating Descriptive and Normative Approaches in Decision Based Design. *Engineering Valuation and Cost Analysis* **3**: 107-129, 2000.
- Mistree, F., W. F. Smith, B. A. Bras, J. K. Allen and D. Muster. Decision-Based Design. A Contemporary Paradigm for Ship Design. *1990 SNAME Annual Meeting, Oct 31-Nov 3 1990*, San Francisco, CA, USA, Soc of Naval Architects & Marine Engineers, Jersey City, NJ, USA, 1990.
- Moore, R. E. *Methods and Applications of Interval Analysis*. Philadelphia, Society for Industrial and Applied Mathematics, 1979.
- Mourelatos, Z. and J. Zhou. A Design Optimization Method Using Evidence Theory. *ASME IDETC*, 2005.

- Muhanna, R. L. and R. L. Mullen. Interval Methods for Reliable Computing. *Engineering Design Reliability Handbook*. E. Nikolaidis, D. M. Ghiocel and S. Singhal. New York, NY, CRC Press, 2004.
- Nikolaidis, E., S. Chen, H. Cudney, R. T. Haftka and R. Rosca. Comparison of Probability and Possibility for Design against Catastrophic Failure under Uncertainty. *Journal of Mechanical Design* **126**(3): 386-394, 2004.
- Oberkampf, W. L., S. M. DeLand, B. M. Rutherford, K. V. Diegert and K. F. Alvin. Error and Uncertainty in Modeling and Simulation. *Reliability Engineering and System Safety* **75**(3): 333-357, 2002.
- Oberkampf, W. L. and J. C. Helton. Investigation of Evidence Theory for Engineering Applications. *Non-Deterministic Approaches Forum, 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Denver, CO, 2002.
- Otto, K. N. and E. K. Antonsson. The Method of Imprecision Compared to Utility Theory for Design Selection Problems. *ASME 1993 Design Theory and Methodology Conference*, 1992.
- Pahl, G. and W. Beitz. *Engineering Design: A Systematic Approach*. London, Springer Publishing, 1996.
- Parry, G. W. The Characterization of Uncertainty in Probabilistic Risk Assessment of Complex Systems. *Reliability Engineering and System Safety* **54**(2-3): 119-126, 1996.
- Pratt, J. W., H. Raiffa and R. Schlaifer. *Introduction to Statistical Decision Theory*. Cambridge, MA, The MIT Press, 1995.
- Rekuc, S. J., J. M. Aughenbaugh, M. Bruns and C. J. J. Paredis. Eliminating Design Alternatives Based on Imprecise Information. *Society of Automotive Engineering World Congress*, 2006-01-0272, 2006.
- Seidenfeld, T., M. J. Schervish and J. B. Kadane. A Representation of Partially Ordered Preferences. *The Annals of Statistics* **23**(9): 2168-2217, 1995.
- Shafer, G. *A Mathematical Theory of Evidence*. Princeton, NJ, Princeton University Press, 1976.
- Simon, H. *Administrative Behavior*. New York, The Macmillan Company, 1947.
- Sobek, D. K., II, A. C. Ward and J. Liker. Toyota's Principles of Set-Based Concurrent Engineering. *Sloan Management Review* **40**(2): 67-83, 1999.
- Springer, M. D. *The Algebra of Random Variables*. New York, John Wiley & Sons, 1979.
- Tintner, G. The Theory of Choice under Subjective Risk and Uncertainty. *Econometrica* **9**: 298-304, 1941.
- Tong, C. and D. Sriram. *Artificial Intelligence in Engineering Design: Models of Innovative Design, Reasoning About Physical Systems, and Reasoning About Geometry (Artificial Intelligence in Engineering Design)*. Boston, Academic Press, 1992.
- Tversky, A. and D. Kahneman. Judgment under Uncertainty: Heuristics and Biases. *Science* **185**: 1124-1131, 1974.
- von Neumann, J. and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton, NJ, Princeton University Press, 1944.
- Walley, P. *Statistical Reasoning with Imprecise Probabilities*. New York, Chapman and Hall, 1991.
- Weber, M. Decision Making with Incomplete Information. *European Journal of Operational Research* **28**(1): 44, 1987.
- Weichselberger, K. The Theory of Interval-Probability as a Unifying Concept for Uncertainty. *International Journal of Approximate Reasoning* **24**(2-3): 149, 2000.

- Williamson, R. C. and T. Downs. Probabilistic Arithmetic I: Numerical Methods for Calculating Convolutions and Dependency Bounds". *International Journal of Approximate Reasoning* **4**: 89-158, 1990.
- Winkler, R. L. Uncertainty in Probabilistic Risk Assessment. *Reliability Engineering & System Safety* **54**(2-3): 127-132, 1996.
- Yager, R. R., J. Kacprzyk and M. Fedrizzi. *Advances in the Dempster-Shafer Theory of Evidence*. New York, John Wiley and Sons, 1994.
- Zadeh, L. A. Fuzzy Sets. *Inf. Control* **8**: 338-353, 1965.
- Zaffalon, M., K. Wesnes and O. Petrini. Reliable Diagnoses of Dementia by the Naive Credal Classifier Inferred from Incomplete Cognitive Data. *Artificial Intelligence in Medicine* **29**(1-2): 61-79, 2003.

Computational Methods for Decision Making Based on Imprecise Information

M. Bruns^a, C.J.J. Paredis^a, and S. Ferson^b

*^aSystems Realization Laboratory
Georgia Institute of Technology
mbruns@gatech.edu or chris.paredis@me.gatech.edu*

*^bApplied Biomathematics
scott@ramas.com*

Abstract: In this paper, we investigate computational methods for decision making based on imprecise information in the context of engineering design. The goal is to identify the subtleties of engineering design problems that impact the choice of computational solution methods, and to evaluate some existing solution methods to determine their suitability and limitations. Although several approaches for propagating imprecise probabilities have been published in the literature, these methods are insufficient for practical engineering analysis. The dependency bounds convolution approach of Williamson and Downs and the distribution envelope determination approach of Berleant work sufficiently well only for open models (that is, models with known mathematical operations). Both of these approaches rely on interval arithmetic and are therefore limited to problems with few repeated variables. In an attempt to overcome the difficulties faced by these deterministic methods, we propose an alternative approach that utilizes both Monte Carlo simulation and optimization. The Monte Carlo/optimization hybrid approach has its own drawbacks in that it assumes that the uncertain inputs can be parameterized, that it requires the solution of a global optimization problem, and that it assumes independence between the uncertain inputs.

Keywords: engineering design, probability box, p-box, uncertainty propagation, imprecision, imprecise probability, Monte Carlo, optimization, interval.

1. Introduction

1.1. DESIGN DECISION MAKING

Design is the process of converting information about system requirements into a specification of

2006 by authors. Printed in USA

a system that satisfies those requirements. This set of system specifications constitutes a design solution. The space of possible design solutions is unstructured and effectively infinite both in dimension and size. In order to successfully navigate through the structurally complex design space, it is necessary to proceed systematically.

Decision-based design is a useful paradigm for thinking systematically about the design process (Mistree, Smith et al. 1990; Hazelrigg 1998; Marston, Allen et al. 2000). Designers progress through the design process with the help of basically two mechanisms: the generation of design alternatives and decision making. From the decision-based design perspective, the critical elements of the design process are the decisions. Note that decision-based design is not an *approach* to design—it is a *perspective*. That is, from the decision-based design perspective, decisions should be the focus of the designer. Within this perspective, there still exist many different approaches.

Every decision in the design process must be made under some degree of uncertainty. Uncertainty exists when the decision maker (DM) does not know the outcome of at least one decision alternative definitely. The dilemma that uncertainty poses for decision making is clear: different decision alternatives might be preferable in different possible (but uncertain) states of the world.

1.2. IMPRECISION IN DESIGN

Since uncertainty strongly influences decision making, and therefore design, it is necessary to study the nature of this uncertainty. Uncertainty is typically divided into two components that we call *variability* and *imprecision*. Although some authors question the philosophical validity of this distinction, it has been argued that such a distinction is useful in practice (Ferson and Ginzburg 1996; Hofer 1996; Winkler 1996; Aughenbaugh and Paredis 2005). Variability corresponds to naturally random behavior of a physical system or process. The standard representation of variability is the probability distribution function.

Many of the uncertainties in engineering design are imprecise. Imprecision is uncertainty due to a lack of knowledge or information (Parry 1996). Imprecision is alternatively referred to as incertitude, but to maintain consistency with past research in the engineering design community we use the term “imprecision” in this paper. The standard representation of pure imprecision is the interval (Kreinovich, Ferson et al. 1999; Muhanna and Mullen 2004). Imprecision arises in design from sequential decision-making, statistical data from finite samples, bounded rationality, and many other sources. For a detailed discussion of the sources of imprecision in engineering design, see the companion paper (Aughenbaugh and Paredis 2006).

Traditional decision analysis assumes precise probabilities. That is, it is assumed that all uncertainty is representable as a precise probability distribution. Because of the high degree of imprecision in engineering design, this assumption is not valid. In order to properly account for

imprecise uncertainties in engineering design, alternative representations, methods for propagation, and decision methods must be developed.

1.3. NEED TO PROPAGATE UNCERTAINTY THROUGH PERFORMANCE MODELS

Any method for making decisions under uncertainty must provide three essential tools: (1) a formal representation for uncertain quantities; (2) a method for computing with uncertain quantities; and (3) a decision policy that determines an action under uncertainty. Because of the widespread presence of imprecise uncertainty in engineering design, we seek to develop these three tools for the special case of imprecise probabilities. In particular, we need: (1) a formal representation for imprecise probabilities; (2) a method for computing with imprecise probabilities; and (3) a decision policy that determines the best action given imprecise probabilistic information.

This paper addresses item (2). For insight into the development of representations of imprecise probabilities see (Ferson and Ginzburg 1996; Ferson and Donald 1998; Ferson, Ginzburg et al. 2002). Decision making with imprecise probabilities has been addressed in (Levi 1980; Walley 1991) and with specific emphasis on engineering design in (Aughenbaugh and Paredis 2005; Rekuc, Aughenbaugh et al. 2006).

1.4. SUMMARY OF THE LITERATURE

Several solutions to the problem of computing with imprecise probabilities have been proposed in the literature. Although analytical methods exist for a limited class of operations on *precise* random variables (Springer 1979), no work has been done to extend these methods to accommodate *imprecise* random variables. A completely stochastic alternative involves double-loop sampling. The current state-of-the-art methods numerically compute best-possible bounds on the resultant probability distribution of some function of imprecise random variables. While these methods are efficient and accurate, they are not practical for a large class of engineering design problems. The weaknesses of these methods will be discussed in section 2.6.

Computing with imprecise probabilities is a generalization of the problem of computing the convolution of probability density functions where the probability density functions happen to be imprecise. In this paper, we use the term convolution to mean any operation on some set of random variables. Extensive summaries of analytical methods for computing convolutions of random variables is found in the book by Springer (Springer 1979) and in the thesis of Williamson (Williamson 1989).

The most straightforward approach for propagating imprecise probabilities through mathematical models is double-loop Monte Carlo sampling – this is alternatively called two-dimensional, 2-D, or second-order Monte Carlo. A formal description of double-loop sampling is given in section 3.2, and a good review is found in (Hoffman and Hammonds 1994). Modifications to pure double-loop sampling methods are presented in (Hofer, Kloos et al. 2002;

Helton and Davis 2003). Monte Carlo techniques are easy to implement, but for many complex problems, their computational cost becomes prohibitive.

The first efficient numerical approach to the propagation of uncertain quantities was presented by Williamson and Downs in (Williamson 1989; Williamson and Downs 1990). Williamson's work was motivated by the desire to develop numerical methods for precise probabilistic arithmetic, but his methods are compatible with imprecise probabilistic arithmetic. Williamson's methods are referred to as *dependency bounds convolutions* because they result in bounds on the true probability distribution under any possible dependence relation between the uncertain quantities. Dependency bounds are "best-possible" in the sense that the resultant bounds are guaranteed to contain the true resultant distribution, and any reduction of the bounds results in the possible exclusion of the true distribution. The commercially available software Risk Calc 4.0 (Ferson 2002) provides an implementation of the dependency bounds methods.

A very similar approach was developed independently by Berleant in (Berleant 1993; Berleant and Goodman-Strauss 1998). Both Berleant's approach and Williamson's approach discretize probability distribution functions and use maximization and minimization operations to find the best-possible probability bounds on the resultant quantity. Berleant's approach is implemented in the software Statool (Berleant and Cheng 1998; Berleant, Xie et al. 2003). Berleant calls his approach *distribution envelope determination* or DEnv. Regan, Ferson, and Berleant (Regan, Ferson et al. 2004) have shown that DEnv and dependency bounds convolution are equivalent for cumulative distribution functions on the positive reals.

These two approaches are fully sufficient for the propagation of uncertain quantities through functional relationships given explicitly as a sequence of binary operations, but they are insufficient for the computations in most realistic engineering design problems. In section 2, we present a formal statement of an engineering design problem and explain why the available methods are insufficient.

1.5. MOTIVATION

The presentation of this paper at this conference is motivated by a desire to facilitate collaboration between the design and reliable engineering computing research communities. While much effort has been expended developing algorithms for computing with uncertain information, much of the results of that effort have been inapplicable to realistic engineering design problems. To resolve this impasse, this paper attempts to present a clear, formal description of a general design problem. Our hope is that those in the reliable engineering community will find further motivation for their research and that we, in the engineering design community, will benefit from their technical expertise.

2. Design Computing with Imprecise Uncertainty

In order to understand the computational challenges of using imprecise uncertainties, it is necessary to understand the computations present in the design process. As discussed in section 1, the design process progresses by a sequence of decisions in which the set of design alternatives under consideration is sequentially reduced. We denote a set of design alternatives at step i by \mathbf{D}_i . In the early stages of design, \mathbf{D}_i is complex and poorly defined. Much of design research focuses on developing heuristics for refining \mathbf{D}_i to a mathematically manageable size and structure. In this paper, we are not concerned with such methods. Instead we assume that \mathbf{D}_i is an interval vector (or hypercube) of dimension n , $\mathbf{D}_i \in \mathbb{IR}^n$, where \mathbb{IR}^n is the space of real n -dimensional interval vectors: $\mathbb{IR} \equiv \{[x, \bar{x}] : x, \bar{x} \in \mathbb{R}, x \leq \bar{x}\}$. More specifically, we assume that we can write

$$\mathbf{D}_i = \left[[d_1, \bar{d}_1], [d_2, \bar{d}_2], \dots, [d_n, \bar{d}_n] \right]$$

where each d_k and \bar{d}_k represent the lower and upper bounds of some real, continuous design variable d_k . Similarly, for discrete design variables, d_k and \bar{d}_k correspond to the smallest and largest of the finite set of alternatives. In this context, the set reduction in each design step, $\mathbf{D}_i \rightarrow \mathbf{D}_{i+1}$, corresponds to a decrease in interval width for at least one of the n design variables. This process of sequential width reduction converges to a final decision which specifies a precisely defined (singleton) design alternative, $\mathbf{D}^* = [d_1^*, d_2^*, \dots, d_n^*]$.

Since design computations often involve only a sequence of decisions that are assumed to be decoupled, we focus on the computations involved in a single decision. In the following sections, we will examine in greater detail the mechanics of a single design decision. This involves representing the DM's beliefs and preferences and using performance models to predict how a particular design will satisfy the DM's preferences. This section will close with a precise statement and discussion of the computational problem we hope to solve.

2.1. ELEMENTS OF A DESIGN DECISION PROBLEM

A rational decision should reflect the DM's beliefs and preferences. Given a set of beliefs, preferences, and a set of design alternatives, the DM uses some decision policy to determine the preferred decision alternative. It is important here to differentiate a *decision alternative* and a *design alternative*. A decision alternative is any choice that the DM has available at any step in the sequential design process. A design alternative, on the other hand, is any completely specified design. A single decision alternative might correspond to multiple design alternatives. See the companion paper (Aughenbaugh and Paredis 2006) for a more detailed discussion.

A decision policy can be represented by the expression

$$\mathbf{D}_{i+1} = \pi(\mathcal{B}, \mathcal{P}, \mathbf{D}_i)$$

which can be translated into the decision to eliminate the set $\mathbf{D}_e = \mathbf{D}_i \setminus \mathbf{D}_{i+1}$. Here \mathcal{B} is a functional representation of the DM's belief state, \mathcal{P} is a functional representation of the DM's

preference state, and π is the decision policy. The set of decision alternatives is the set of all the proper subsets of \mathbf{D}_i excluding the null set.

The belief state, \mathcal{B} , is some general multi-valued function that embodies the DM's beliefs about the state of the relevant world at the time of the decision. It is a general but quantifiable measure of the DM's uncertainty about the set of relevant states of affairs. In general, \mathcal{B} is a multi-valued function because of the possibility of imprecision. Realizable relevant states (assumed to be quantifiable) will be represented as scalar vectors, $\mathbf{x} \in \mathbb{R}^n$, where each element, x_j , of \mathbf{x} corresponds to some relevant uncertain quantity. The set of all relevant states of affairs will be denoted by $\mathbf{X} = [X_1, X_2, \dots, X_n]$. In the context of design, \mathbf{X} can be thought of as the set of variables over which the DM has no control. Mirroring the notation for random variables, uppercase is used to emphasize that the actualized relevant state is an uncertain quantity ranging over the space of possible states of affairs. Note that the DM might choose to model any X_j as certain—that is, $X_j = x_j$ is a known quantity. The most common representation of a belief state is a precise probability measure over the sample space of relevant states of affairs. A precise probability measure is a single-valued function $P: \mathbb{R}^n \rightarrow [0,1]$. That is, $P(\mathbf{x}) = p$ such that $p \in [0,1]$.

The preference state, \mathcal{P} , is some general multi-valued function that embodies the DM's preferences about possible consequences of the decision. Like \mathcal{B} , \mathcal{P} can be multi-valued in order to account for imprecision. The uncertain consequences of a decision are dependent on the actual relevant state of affairs $\mathbf{x}^l = [x_1^l, x_2^l, \dots, x_n^l]$ (corresponding to state l) as well as the decision, $\mathbf{D}^k = [d_1^k, d_2^k, \dots, d_m^k]$ (corresponding to decision k), taken. The preference state, \mathcal{P} , at the time of the decision is a measure of the value of a particular consequence to the designer. The most common representation of the preference state is a single-valued utility function $U: \mathbb{R}^k \times \mathbb{R}^l \rightarrow \mathbb{R}$. That is, $U(\mathbf{D}^k, \mathbf{x}^l) = u$. The utility of a of a particular design, \mathbf{D}^k , given some specific outcome, \mathbf{x}^l , is deterministic, but since \mathbf{x}^l is uncertain, the utility of \mathbf{D}^k is also uncertain.

Unlike the uncertain state vector, \mathbf{X} , the design alternative search space, \mathbf{D} , is controlled by the DM. Generally, each d_i in \mathbf{D} might be continuous or discrete and bounded or unbounded. For simplicity, we make the assumption that \mathbf{D} is an interval vector in $\mathbb{I}\mathbb{R}^n$ as was discussed at the beginning of this section.

Finally, the decision policy, π , is a general multi-valued functional mapping from the DM's beliefs and preferences to the set of *non-dominated* decision alternatives \mathbf{D}_{i+1} . A non-dominated decision alternative is an alternative that, given some body of information, the DM cannot rationally eliminate. In classical decision theory, π is “maximize expected utility.” Mathematically, the preferred solution is found as

$$\mathbf{D}^* = \arg \max_{\mathbf{D}^k \in \mathbf{D}_i} \left[\sum_{x_1} \dots \sum_{x_m} u(\mathbf{D}^k, \mathbf{x}) p(\mathbf{x}) \right]$$

or

$$\mathbf{D}^* = \arg \max_{\mathbf{D}^k \in \mathbf{D}_i} \left[\int_{x_1} \dots \int_{x_n} u(\mathbf{D}^k, \mathbf{x}) p(\mathbf{x}) dx_n \dots dx_1 \right]$$

for discrete and continuous problems, respectively. In this case, $\mathbf{D}_e = \mathbf{D}_i \setminus \mathbf{D}^*$.

Two special cases of the general decision problem should be mentioned. Both of these specific cases make assumptions about the uncertainty of the DM's beliefs and preferences. The decision is deterministic when all beliefs and preferences are certain. In this case, the DM can simply maximize the utility over \mathbf{D}_i . The preferred design solution will be $\mathbf{D}^* = \arg \max_{\mathbf{D}^k \in \mathbf{D}_i} (U(\mathbf{D}^k))$. The DM selects the design that necessarily results in the best system

performance. This case is unrealistic since design decisions always involve uncertainty with regards to beliefs and preferences.

The second special case of a general design decision acknowledges the presence of uncertainty, but represents that uncertainty as precise probability distributions. That is the DM's beliefs are purely probabilistic, and his or her preferences are deterministic. Sampling strategies such as Monte Carlo and Latin Hypercube are well-established and frequently-used solutions for propagating precise probabilistic uncertainty (Fishman 1996). The DM is able to make a decision by maximizing the expected utility of the design through stochastic programming. The resulting design solution will be $\mathbf{D}^* = \arg \max_{\mathbf{D}^k \in \mathbf{D}_i} (E[U(\mathbf{D}^k)])$. This case is more realistic than the purely

deterministic solution described above but is still an approximation because the DM is not able to account for imprecise information.

Before we can study computational methods for handling imprecise information, we must first make some simplifying assumptions about the representation of uncertain quantities and the decision and performance models to be used.

2.2. BELIEFS REPRESENTED AS P-BOXES

An uncertain quantity is some event or variable characterized by sets of possible levels of belief. An uncertain quantity is a more general case of a random variable. Whereas a random variable characterizes a quantity by some precise belief function—namely, a probability distribution function, an uncertain quantity assigns a set of belief functions to a single quantity. For instance, consider a bent quarter. I am uncertain about whether it will land heads-up or tails-up, but until I have seen it flipped many times, I am also uncertain about how probable it is that it will land heads-up or tails-up. I believe that the probability of the bent quarter landing heads-up is less than

0.6 and greater than 0.3. My belief state then corresponds to the interval of probability values between 0.3 and 0.6.

The fundamental types of uncertain quantities are intervals and imprecise probabilities. An interval is a connected set of numbers on the real line. Specifically, the real interval $[a, b] = \{x : x \in \mathbb{R} \text{ and } a \leq x \leq b\}$. An interval represents a perfectly imprecise uncertain quantity since no assumptions are made about beliefs between the upper and lower bounds. Under interval uncertainty, the DM believes only the fact that the true value of the quantity is constrained by two bounds. When a DM represents an uncertain quantity as an interval he or she has no beliefs about the likelihood of any value in the interval. This is an extreme case—most often the DM does have some beliefs about likelihoods. For extensive discussions of propagating interval uncertainty, see (Moore 1979; Alefeld and Herzberger 1983; Kearfott and Kreinovich 1996).

An imprecise probability is an interval-valued probability measure assigned to an uncertain event—for instance, my beliefs about the bent quarter. Imprecise probabilities are discussed and justified thoroughly in (Walley 1991). The theory of imprecise probabilities developed by Walley extends the operational definition of subjective probabilities to allow for imprecision. The primary advantage of using imprecise probabilities for representing uncertain beliefs is that they allow for the representation of both variability and imprecision.

The probability box (p-box) is a formalism for representing uncertain quantities (Ferson and Ginzburg 1996; Ferson and Donald 1998). The defining characteristic of a p-box are the probability bounds that define upper and lower limits on cumulative probability over the domain of the uncertain quantity. When defining a p-box formally, there are essentially two structures involved: the p-box proper, and the p-box function. The p-box proper, \boxed{X} , of some uncertain quantity X defines the p-box as a set of distribution functions constrained by probability bounds and the property of being non-decreasing

$$\boxed{X} = \{F_X(x) : \forall x \in \mathbb{R}, \underline{F}_X(x) \leq F_X(x) \leq \bar{F}_X(x)\}$$

where $\underline{F}_X, F_X, \bar{F}_X : \mathbb{R} \rightarrow [0, 1]$, $\underline{F}_X = \underline{P}(X \leq x)$ and $\bar{F}_X = \bar{P}(X \leq x)$ are the lower and upper cumulative probability bounds, and F_X is non-decreasing with x . These probability bound functions are determined by the p-box function. The p-box function is an interval-valued mapping from x to the interval $[0, 1]$. We express the p-box function as

$$F_X^\square(x) = [\underline{F}_X(x), \bar{F}_X(x)]$$

where $\underline{F}_X(x) \leq \bar{F}_X(x)$ for all x . In other discussions, it might be useful to reverse the order of the bounding distributions in the interval above such that $F_X^\square(x) = [\bar{F}_X(x), \underline{F}_X(x)]$. In this case, $\bar{F}_X(x)$ denotes the left bound on the p-box and $\underline{F}_X(x)$ denotes the right bound. In other words, upper and lower are defined with respect to x rather than with respect to cumulative probability. For our purposes, however, it is more convenient to interpret upper and lower with respect to probability.

The p-box is general enough to represent intervals, probability distributions, scalars, as well as imprecise probability distributions. An interval $X = [a, b]$ corresponds to the p-box defined by the probability bounds

$$\underline{F}_X(x) = \begin{cases} 0, & x < b \\ 1, & x \geq b \end{cases}$$

and

$$\bar{F}_X(x) = \begin{cases} 0, & x < a \\ 1, & x \geq a \end{cases}.$$

A normally distributed random variable, $X \sim N(\mu, \sigma)$, corresponds to the p-box containing only one cdf, $\boxed{X} = \{\Phi_{\mu, \sigma}(x)\}$, and the degenerate p-box function with $\underline{F}_X(x) = \bar{F}_X(x) = \Phi_{\mu, \sigma}(x)$ where $\Phi_{\mu, \sigma}(x)$ is the cumulative distribution function of the normal distribution with mean μ and standard deviation σ . A scalar, a , corresponds to the degenerate p-box function with

$$\underline{F}_X(x) = \bar{F}_X(x) = \begin{cases} 0, & x < a \\ 1, & x \geq a \end{cases}.$$

Finally, and most importantly, the p-box can be used to represent imprecise probability distributions such as $X \sim N([\underline{\mu}, \bar{\mu}], [\underline{\sigma}, \bar{\sigma}])$. Here it is known that the uncertain quantity has normal variability with an imprecise mean, $\mu \in [\underline{\mu}, \bar{\mu}]$, and an imprecise variance, $\sigma \in [\underline{\sigma}, \bar{\sigma}]$. This imprecise probability distribution corresponds to the parameterized p-box $\boxed{X}^P = \{F_X(x; \mu, \sigma) = \Phi_{\mu, \sigma}(x) : \mu \in [\underline{\mu}, \bar{\mu}], \sigma \in [\underline{\sigma}, \bar{\sigma}]\}$ where the superscript P denotes that the p-box is parameterized. It is not meaningful to speak of bounding functions for parameterized p-boxes since the parameterized p-box will not contain all non-decreasing functions between its lower and upper bounding functions. Parameterized p-boxes will be discussed in greater detail in section 3.1.

In this paper, the DM's beliefs are modeled as p-boxes. Relating back to previous notation, the DM's beliefs are represented by $\mathcal{B}(\mathbf{x}) = F_{\mathbf{X}}^{\square}(\mathbf{x})$ where $F_{\mathbf{X}}^{\square}(\mathbf{x})$ represents the joint p-box function for the vector of relevant uncertain quantities. A joint p-box is the imprecise equivalent of a joint distribution function for precise probabilities. There are two steps for justifying this representation. First, the theory of imprecise probabilities is the most fully developed model for imprecise uncertainty. Unlike alternative representations of imprecise uncertainty such as possibilities (Dubois and Prade 1988) or fuzzy sets (Zadeh 1965), imprecise probabilities have a clear operational definition. An operational definition is "a rule which indicates how the mathematical notions are intended to be interpreted (Cooke 2004)." The subjective interpretation of probability provides an operational definition in terms of subjective degree of belief expressed

through a willingness to bet (Savage 1972; de Finetti 1980). Walley extends the subjective interpretation to account for the imprecision between minimum selling prices and maximum buying prices of gambles (Walley 1991). For a criticism of uncertainty models without clear operational definitions, see (Cooke 2004).

The second step in the justification for using p-boxes to represent beliefs is that they are intuitive and used in most of the imprecise probability propagation literature for representing imprecise probabilities. Cumulative probabilities are a straightforward way in which to assign definite probabilities to events. As examples of the common use of p-boxes in the literature, see the work of Williamson and Downs (Williamson and Downs 1990), Ferson (Ferson and Ginzburg 1996; Ferson and Donald 1998; Ferson 2002; Ferson and Hajagos 2004), and Berleant (Berleant 1993; Berleant and Goodman-Strauss 1998; Berleant, Xie et al. 2003; Berleant and Zhang 2004).

2.3. UTILITY MODELS REPRESENTED BY BLACK-BOX FUNCTIONS

So far, we have only studied decision policy models in terms of abstract functional mappings from beliefs and preferences to a preferred action. To complete the link from generic decision theory to specific design practice, we must first present and justify several assumptions regarding the mathematical models to be used in design decision making.

For practical reasons, proposed solutions should assume that all mathematical models are black boxes. Although it is not true that engineering models are truly black-boxes, in the sense that nobody knows the mathematical operations inside, it is true that much of engineering practice uses previously developed models as if they were black-boxes. In the future, it is possible that the dependency bounds convolution or the distribution envelope determination methods will be implemented in much of the standard engineering software. At this point in time, however, this is not the case. Although Risk Calc 4.0 (Ferson 2002) and Stool (Berleant, Xie et al. 2003) are useful for propagating imprecise information through algebraic models, much of engineering design practice requires the aid of advanced simulation software such as FLUENT or ANSYS. If the representation of beliefs as p-boxes is to take hold in the engineering design community it is necessary that methods be developed that propagate imprecise beliefs through black-box models developed for advanced software.

2.4. DECISION POLICIES FOR IMPRECISE BELIEFS AND PREFERENCES

A rational DM must choose decision alternatives that maximize his or her utility. In the presence of uncertainty, utility is no longer certain. Therefore, in accordance with the axioms of decision theory, the DM should choose the alternative that maximizes his or her expected utility, $E[U]$. If the DM's uncertainty is all due to variability, maximizing expected utility is sufficient. However, in the previous discussion, it has been argued that the DM's beliefs and preferences are imprecise. The presence of imprecision results in intervals of expected utility, $[\underline{E}[U], \bar{E}[U]]$. While

imprecise beliefs and preferences more accurately reflect the DM's knowledge state, they also complicate considerably the act of decision making. A DM with an imprecise knowledge state needs a more sophisticated decision policy than classical decision theory's prescription of "maximize expected utility." Specifically, imprecise preferences lead to indeterminacy, and indeterminacy results in sets of *non-dominated* decision alternatives. In other words, imprecise preferences result in situations in which rational decision makers cannot choose a single alternative from the set of non-dominated alternatives. Researchers in the imprecise probability community have proposed several decision policies to overcome the indeterminacy in imprecise decision making (Troffaes 2004; Rekuc, Aughenbaugh et al. 2006). Here we limit our discussion to two of these criteria: maximality (Walley 1991) and Γ -maximin (Berger 1985). Any proposed solution to the problem of computing for design decision making with imprecise uncertainty must be compatible with these decision criteria.

To understand the indeterminacy associated with imprecise knowledge better, consider a simple decision problem in which the DM must select a value for a continuous real-valued design variable, d . The DM in this situation can quantify his or her preferences for single values of d with an imprecise expected utility function, $E[U(d)] = [\underline{E}[U(d)], \bar{E}[U(d)]]$. The upper and lower bounds of the DM's utility function are shown in Figure 1.

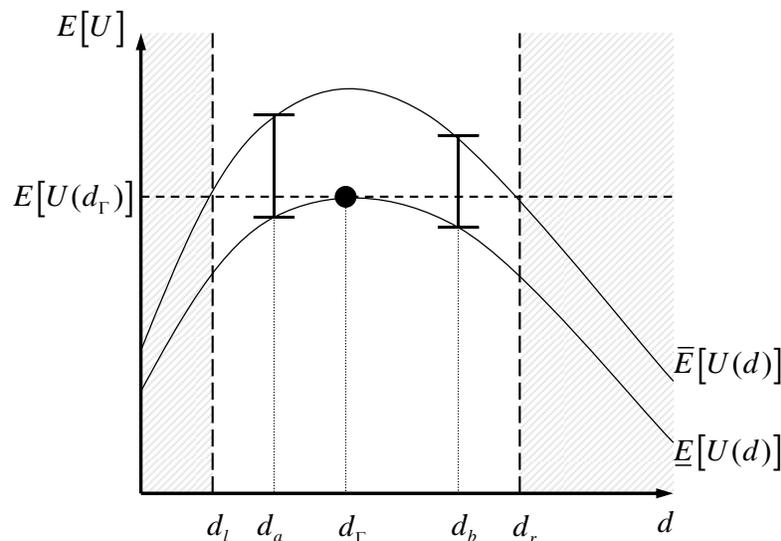


Figure 1. Decision indeterminacy with imprecise utilities.

Which value of d should the DM select? The higher the utility the more preferred the design, but in this example the utility bounds overlap. Consider a comparison between design alternatives d_a and d_b as shown in Figure 1. The actual utility of either of these alternatives could fall anywhere

between their corresponding upper and lower utility bounds, but the DM has no information about where in those bounds. In some actual cases, d_a will be preferable, but in other cases, d_b will more fully satisfy the DM's preferences. We say that d_a and d_b are *pairwise non-dominated*, and the decision between d_a and d_b is indeterminate. In our example, there is a set of dominated design alternatives. All designs between d_l and d_r are non-dominated by every other design alternative in $[d_l, d_r]$. However, all design alternatives outside of this region are pairwise dominated by the design alternative d_T . Therefore, a rational DM will eliminate the set of design alternatives $d < d_l$ and $d > d_r$. However, indeterminacy remains for all designs between these two bounds. In engineering design, indeterminacy is ultimately not an option since a final design for production cannot be imprecisely specified. Therefore, the DM needs a more sophisticated decision policy in order to further distinguish the space of decision alternatives.

Indeed there is no decision policy that is able to identify a *single* rationally preferred solution in the presence of imprecise uncertainty because indeterminacy is inherent in the problem. Maximality can be used to reduce the size of the set of rational decision alternatives—the DM could rationally choose *any* of the alternatives in that set, but none of the decision alternatives in that reduced set is rationally preferable to any of the others in that set given the current knowledge state of the DM. Decision policies for imprecise uncertainty can be grouped into two general strategies: (1) those that seek to minimize the size of the set of non-dominated alternatives through more sophisticated comparisons of alternatives, and (2) those that select a single-valued solution based on some semi-arbitrary decision criterion. While strategies of type (1) are preferable for rational decision making, for practical purposes, the DM may need to employ some strategy of type (2) in order to find a single-valued design solution.

The decision policies that seek to minimize the set of non-dominated alternatives differ in the amount of information they take into account. Generally, as more information is considered, the resultant set of non-dominated alternatives will decrease in size. The *maximality* criterion (Walley 1991) is well-suited for a broad-class of decision problems because it takes into account most of the available relevant information. By introducing differences in expected utility, the DM is able to identify alternatives that are dominated throughout the entire space of possible states of affairs, \mathbf{X} . A strict comparison of utility bounds will lose this additional information. The maximality criterion takes into account shared uncertainty. Shared uncertain variables, z_s , are those uncertain quantities that are independent from the design variables—i.e. no matter what design variable is selected the shared uncertain variable will take the same unknown value. Therefore, when comparing the utility of two designs, the DM should evaluate both utilities at the same values of the shared uncertain variables. The maximality criterion prescribes that the DM eliminate all decision alternatives for which, when compared to some other alternative evaluated at the same values for the shared uncertain variables, the upper bound on their expected difference in utility is strictly less than zero. Formally,

$$\mathbf{D}_e = \left\{ \mathbf{D}^j \in \mathbf{D}_i : (\exists \mathbf{D}^k \in \mathbf{D}_i) \left(\max_{\substack{z_s \in Z_s \\ z_j \in Z_j \\ z_k \in Z_k}} \bar{E} [U(\mathbf{D}^j, z_j, z_s) - U(\mathbf{D}^k, z_k, z_s)] < 0 \right) \right\}$$

where \mathbf{D}^j and \mathbf{D}^k are specific decision alternatives in the set \mathbf{D}_i , Z_s is the set of shared uncertain variables, Z_j is the set of uncertain variables specific to \mathbf{D}^j , and Z_k is the set of uncertain variables specific to \mathbf{D}^k . As an illustration of the use of the maximality criterion, consider again the example in which the DM is trying to select a single value for d . Based on past experience, or some other heuristic, the DM believes that d^* will most likely be the preferred solution. In order to eliminate a larger set of design alternatives, the maximality criterion requires that the DM calculate $\bar{E}[U(d_i, Z_s, Z_i) - U(d^*, Z_s, Z^*)]$ for all $d_i \neq d^*$. A plot of this expected difference in utility is shown in Figure 2.

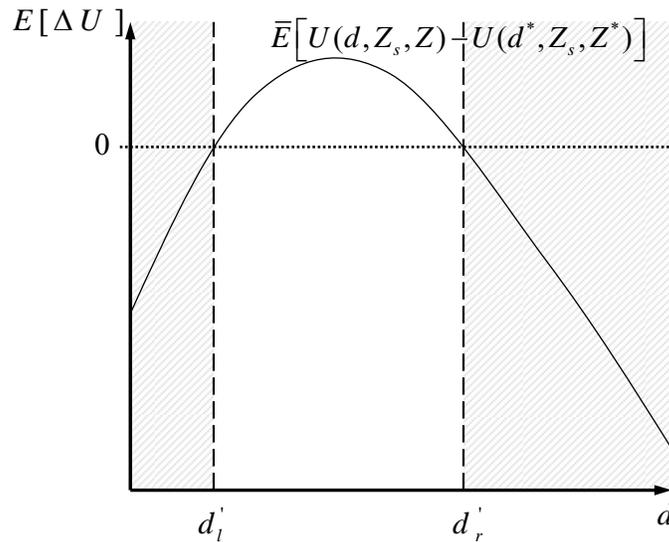


Figure 2. Elimination with the maximality criterion.

For all values of d less than d'_l and greater than d'_r , $\bar{E}[U(d_i, Z_s, Z_i) - U(d^*, Z_s, Z^*)] < 0$. This means that no matter what the actual relevant state of affairs, d^* will outperform those designs, and these regions can be eliminated from consideration. In terms of previous notation, $\mathbf{D}_e = \{d : d < d'_l \text{ and } d > d'_r\}$. While application of the maximality criterion will identify a smaller

set of non-dominated alternatives, the DM will still remain indeterminate between the reduced set of alternatives—in this example the DM is indeterminate between all $d \in [d'_l, d'_r]$. In general, the bounds found through application of the maximality criterion will be tighter than the bounds arising from the application of the interval dominance criterion—that is, $[d'_l, d'_r] \subseteq [d_l, d_r]$ and most often $[d'_l, d'_r] \subset [d_l, d_r]$.

The use of shared uncertain variables is similar to the variance reduction technique of using common random numbers (CRNs) in simulation (Law and Kelton 2000). The goal of a simulation is usually to compare two scenarios or alternative designs by examining the difference in output for different combinations of control parameters. If different random numbers are used in the simulations for the different alternatives, additional noise is introduced into the model. CRNs are used to induce correlation between scenarios, thereby reducing the variances of the results. In engineering design, shared uncertainty is an inherent characteristic of the problem. Therefore, a DM does not have to add the commonality, he or she merely needs to recognize it and take advantage of that additional property when it exists. The maximality criterion is a means of exploiting this inherent commonality. A detailed discussion of shared uncertainty can be found in the Master's Thesis of Rekuc (Rekuc 2005) as well as in (Rekuc, Aughenbaugh et al. 2006).

In order to identify a single-valued decision, the DM must employ some semi-arbitrary decision policy. The most conservative of these types of policies is the Γ -maximin criterion (Berger 1985). Very simply, Γ -maximin prescribes that the DM select the alternative that maximizes the lower bound on expected utility. In other words, the DM selects the best worst case solution. Formally, the Γ -maximin solution is found by the expression

$$\mathbf{D}^\Gamma = \arg \max_{\mathbf{D}^k \in \mathbf{D}_i} \left(\underline{E}_{\mathbf{X}} [U(\mathbf{D}^k, \mathbf{x})] \right)$$

where the subscript, \mathbf{X} , on \underline{E} denotes that the lower expectation is taken over the entire uncertain state space. In Figure 1, the Γ -maximin solution is marked d_Γ . Selecting the Γ -maximin solution assures that in the worst-case actualized state of affairs, d_Γ will outperform any other design alternative operating in its worst-case actualized state of affairs. This is semi-arbitrary because the DM has no rational reason to believe that the worst-case will be actualized, but the DM can still be certain that performance will at least exceed $\underline{E}[U(d_\Gamma)]$.

In the presence of imprecision, the DM will generally need to resort to using some semi-arbitrary decision policy such as Γ -maximin to make a final decision. What value then are the interval dominance and maximality criteria? Should not the DM just compute and select the Γ -maximin solution? The Γ -maximin solution is a function of the body of information available to the DM. Since the design process is not self-contained, this body of information is not static. As the DM progresses through the design process, new information about the structure of the design space and the likelihood of different relevant states of affairs become known. Therefore, the DM should delay making unnecessary (i.e. specific) decisions in the early stages of the design

process. The value of proceeding through the design process with sets of design alternatives is discussed in the set-based design literature (Sobek and Ward 1996; Sobek, Ward et al. 1999; Rekuc, Aughenbaugh et al. 2006). The maximality criterion leads to tight, but rational bounds, on the most-preferred solution and so it is therefore useful in the early stages of the design process.

2.5. PROBLEM STATEMENT

Now that the general issues involved in computing with imprecise information have been explicated, we can now present a concise statement of the problem.

Given:

1. A utility black-box function $U = f(\mathbf{D}^k, \mathbf{x})$ where U is the utility of the design $\mathbf{D}^k \in \mathbb{R}^m$ dependent on some $\mathbf{x} \in \mathbb{R}^n$. Generally, f is an interval-valued mapping $f : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{IR}$ resulting in the lower and upper utilities $\underline{U}(\mathbf{D}^k, \mathbf{x})$ and $\bar{U}(\mathbf{D}^k, \mathbf{x})$.
2. A vector of p-boxes of dimension n , $\underline{\mathbf{X}} = [\underline{X}_1, \underline{X}_2, \dots, \underline{X}_n]$, describing the uncertainty about the relevant state of affairs, \mathbf{x} . This assumes that no *joint* p-box distribution is known which is typically the case in engineering problems. In other words, nothing is known about the dependence relationships between the uncertain quantities.

Find:

1. The lower and upper expected utilities of a design, \mathbf{D}^k , with respect to the vector of uncertain quantities, $\underline{\mathbf{X}}$: $\underline{E}_{\underline{\mathbf{X}}} [U(\mathbf{D}^k, \mathbf{x})]$ and $\bar{E}_{\underline{\mathbf{X}}} [U(\mathbf{D}^k, \mathbf{x})]$.
2. The set of dominated solutions under the maximality criterion:

$$\mathbf{D}_e = \left\{ \mathbf{D}^j \in \mathbf{D}_i : (\exists \mathbf{D}^k \in \mathbf{D}_i) \left(\max_{\substack{z_s \in Z_s \\ z_j \in Z_j \\ z_k \in Z_k}} \bar{E} [U(\mathbf{D}^j, z_j, z_s) - U(\mathbf{D}^k, z_k, z_s)] < 0 \right) \right\}.$$

3. The Γ -maximin solution: $\mathbf{D}^\Gamma = \arg \max_{\mathbf{D}^k \in \mathbf{D}_i} (\underline{E}_{\underline{\mathbf{X}}} [U(\mathbf{D}^k, \mathbf{x})])$.

2.6. INADEQUACIES OF THE AVAILABLE METHODS

In section 1.4, four approaches for propagating uncertainty were discussed—1) distribution convolutions, 2) dependency bound convolutions, 3) distribution envelope, and 4) double-loop sampling. At present, the first three of these approaches are incapable of solving the problem stated in section 2.5. The fourth approach works but is relatively inefficient. Before an alternative method is proposed, it is necessary to explain why the existing methods fail.

The ideal solution would be to formulate and analytically solve the appropriate set of distribution convolutions. Unfortunately, this is practically impossible. The transformation methods described in Springer (Springer 1979) are limited to basic binary algebraic operations for independent variables with a few distribution shapes. Yager's method (Yager 1986) also requires independent distributions, although it can handle arbitrary shapes and operations. All these methods are impossible or very cumbersome for black-box computer models where the functional relationship is not given explicitly as a sequence of binary operations. Analytical methods appear even less tractable in the presence of imprecision where *sets* of distributions must be convolved.

The dependency bounds approach of Williamson and the distribution envelope approach of Berleant are considerably more promising, but they must overcome at least two obstacles before they can be used in engineering design. First, both of these approaches depend strongly on the methods of interval arithmetic for which the presence of repeated variables can result in over-conservative (i.e. not best-possible) solution bounds. While sub-interval reconstitution methods work well for low-dimensional problems (Moore 1979; Ferson and Hajagos 2004; Ferson, Nelsen et al. 2004), they are prohibitively expensive in realistic engineering problems with a large number of imprecise quantities. Second, black-box propagation of intervals is still only workable for quasi-linear problems. Trejo and Kreinovich have developed a randomized algorithm for propagating interval uncertainty through black-box models (Trejo and Kreinovich 2001; Kreinovich and Ferson 2004), but the method assumes that the black-box model is broadly linear in the region of sampling. It is unclear at this point if this black-box method has general applicability towards complex engineering analysis models.

In order for the dependency bounds and distribution envelope methods to be applicable for engineering design, methods for better propagating intervals through black-box models in the presence of many repeated variables need to be developed. If these conditions were met, it would then be necessary to convince the producers of the standard engineering analysis software to incorporate these methods into their products. While this seems possible, and is perhaps the most desirable solution, our concern is more immediate: how can engineers use the tools available to them today to make realistic design decisions under imprecise uncertainty?

One very simple and easy-to-implement approach is a double-loop sampling routine. A formal discussion will be presented in section 3, but double-loop sampling involves random sampling across the two dimensions of an uncertain quantity (Hoffman and Hammonds 1994; Helton and Davis 2003). Since sampling routines only require evaluations at scalar values of the

set of uncertain variables, these approaches meet our requirement of being compatible with black-box utility models. For high-dimensional problems, double-loop sampling can become prohibitively expensive because the sampling in the outer loop does not retain the computational advantages of Monte Carlo simulation. Specifically, the outer loop sampling is not used to determine an expected value but rather the extrema of results of the inner loop. To approximate these bounds accurately an increasingly large number of samples must be taken as the dimensionality of the problem increases. As a possible solution to this, some authors have suggested a sensitivity analysis approach (Hofer, Kloos et al. 2002). In the next section, we present an alternative means of speeding up double-loop sampling in which one of the sampling loops is replaced by an optimization algorithm.

3. Optimizing over Imprecise Distribution Parameters

Of the available methods, double-loop sampling is the only solution convenient for functioning through a black-box utility model. For problems of high-dimensionality, though, double-loop sampling can be prohibitively expensive. In this section, a modification of double-loop sampling is proposed in order to attain a more efficient method for computing with uncertain quantities through black-box utility models.

3.1. PARAMETERIZED P-BOXES

In order to clarify the discussion, it is first necessary to present a simplified representation of the general p-box presented in section 2.2. A parameterized p-box is the set of all possible distributions resulting from some known distribution function with imprecisely known parameters. Formally,

$$\boxed{X}^P = \{F_X(x; \boldsymbol{\theta}) : \boldsymbol{\theta} \in [\underline{\boldsymbol{\theta}}, \bar{\boldsymbol{\theta}}]\}$$

where $F_X(x; \boldsymbol{\theta})$ is non-decreasing with x , and $\boldsymbol{\theta} \in \mathbb{R}^q$ is a vector of distribution parameters that affect the shape or scale of F_X . Imprecision is introduced through uncertainty in the parameters. Specifically, the DM is uncertain of the true values of the distribution parameters except for the fact that they lie within known bounds. That is, for all $\theta_k \in \boldsymbol{\theta}$, $\underline{\theta}_k \leq \theta_k \leq \bar{\theta}_k$.

It is important to emphasize the difference between a parameterized p-box and a general p-box. Similar to a general p-box, a parameterized p-box is a set of non-decreasing probability distribution functions constrained by upper and lower bounds. But unlike a general p-box, a parameterized p-box does not contain all possible non-decreasing distributions lying between its lower and upper bounds. In set notation, if \boxed{X} and \boxed{X}^P share the same bounding functions,

then $\boxed{X}^P \subset \boxed{X}$. To see this, consider a p-box and a parameterized p-box with the same upper and lower bounds.

$$\boxed{X} = \{F_X(x) : \underline{F}_X(x) \leq F_X(x) \leq \bar{F}_X(x)\}$$

where \underline{F}_X is normally distributed with mean $\mu=4$ and standard deviation $\sigma=1$ and \bar{F}_X is normally distributed with $\mu=1$ and $\sigma=1$. A parameterized p-box with identical bounds is

$$\boxed{X}^P = \{F_X(x; \mu, \sigma) : X \sim Normal(\mu = [1, 4], \sigma = 1)\}.$$

Both of these sets of functions are constrained by the bounds \underline{F}_X and \bar{F}_X , but \boxed{X} contains functions not found in \boxed{X}^P as shown in Figure 3.

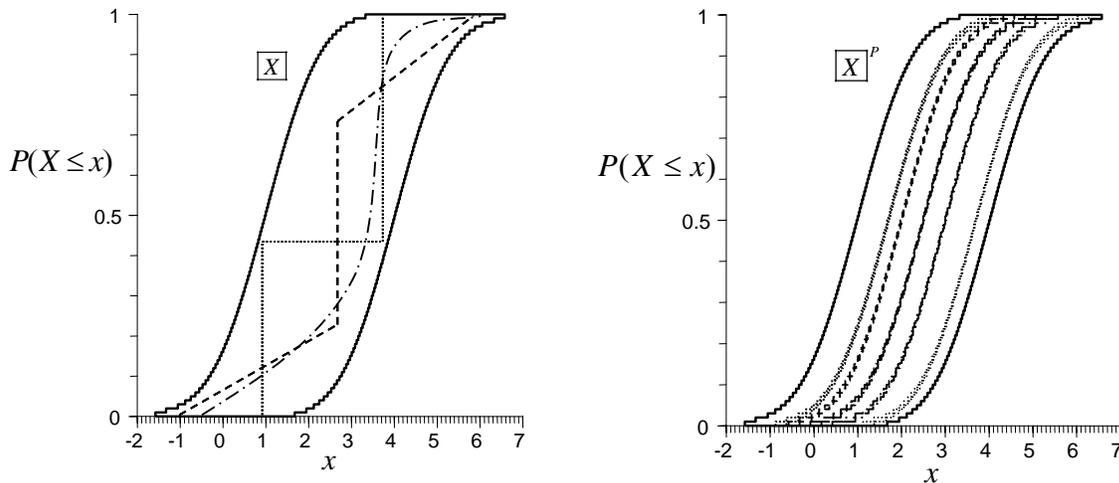


Figure 3. Comparison of general and parameterized p-boxes.

Though less general, a parameterized p-box is in many cases a better representation of the DM’s beliefs about an uncertain quantity. A common example of this arises in statistical parameter estimation where data gives rise to confidence intervals on the true parameter values for some random variable with a known distribution. If the DM’s beliefs cannot be represented as parameterized p-boxes, then double-loop sampling cannot propagate those beliefs.

3.2. DOUBLE-LOOP SAMPLING

Double-loop sampling involves two layers of sampling: one associated with distribution parameters and the other associated with the distributions themselves. In effect, double-loop sampling involves sampling from sampled distributions.

Recall that our problem is to determine $E_{\underline{\mathbf{X}}}[U(\mathbf{D}^k, \mathbf{x})]$ and $\bar{E}_{\underline{\mathbf{X}}}[U(\mathbf{D}^k, \mathbf{x})]$ for a given design, \mathbf{D}^k . The DM has some black-box utility function, $[\underline{U}, \bar{U}] = f(\mathbf{D}^k, \mathbf{x})$, that computes an interval of utility for a given design, \mathbf{D}^k , and a precisely known state of relevant affairs, \mathbf{x} . Assuming that the DM's belief state is representable by a vector of parameterized p-boxes, $\underline{\mathbf{X}}^P$, the DM can determine the upper and lower expected utilities of design \mathbf{D}^k with double-loop sampling. The outer loop will be called the "*parameter*" loop since it involves sampling different values for the set of distribution parameters for all of the uncertain quantities. The inner loop will be called the "*probability*" loop since it involves sampling from precise probability distribution functions.

The first step in double-loop sampling is to define a vector containing all distribution parameters for all of the uncertain quantities. Each $\underline{\mathbf{X}}_j^P \in \underline{\mathbf{X}}^P$ has associated with it a set of imprecise parameters stored in the vector $\boldsymbol{\theta}^j \in [\underline{\boldsymbol{\theta}}^j, \bar{\boldsymbol{\theta}}^j]$. The number of parameters associated with a single uncertain quantity, x_j , is denoted $q_j = \text{length}(\boldsymbol{\theta}_j)$. For notational convenience, it is desirable to combine each of these $q_j \times 1$ vectors into a single vector representing all relevant distribution parameters. This super-vector will be denoted $\boldsymbol{\Theta}$. Also, by extension from the lower and upper bounds of the sub-vectors, lower and upper bounds of the super-vector can be determined. That is, the vector of distribution parameters is constrained such that $\boldsymbol{\Theta} \in [\underline{\boldsymbol{\Theta}}, \bar{\boldsymbol{\Theta}}]$. These parameter bounds are important as they represent all of the imprecision in the DM's belief state. The purpose of the parameter loop is to experiment with these imprecise distribution parameters in order to approximate the smallest and largest utility that the DM should expect for design \mathbf{D}^k .

In the parameter loop, the space of the parameter vector, $\boldsymbol{\Theta}$, is explored by random sampling. Once the DM has defined the elements in $\boldsymbol{\Theta}$, he or she must first randomly select a single point in the space of $\boldsymbol{\Theta}$. This point corresponds to a set of precise distributions for all uncertain quantities and will be denoted $\boldsymbol{\Theta}^a$.

The probability loop uses these precise distribution functions to solve a purely probabilistic sampling problem. Specifically, the probability loop uses Monte Carlo samples from the distributions defined by $\boldsymbol{\Theta}^a$ to compute an expected utility of the design \mathbf{D}^k . The expected utility

of \mathbf{D}^k given some Θ^a is denoted E_a . The process of computing an E_a is repeated for s randomly sampled points, Θ^a , in the parameter space, Θ . That is, the DM computes an E_a corresponding to some Θ^a for $a=1, \dots, s$.

If the sampled parameter vectors sufficiently cover the parameter space, then the largest and smallest values of the set $\{E_a\}$ can be used to approximate the lower and upper expected utilities of the design \mathbf{D}^k . Formally, the lower and upper expected utilities are approximated by the expressions $\underline{E}_{\bar{\mathbf{x}}^p}[U(\mathbf{D}^k, \mathbf{x})] \approx \min_{1 \leq a \leq s} E_a$ and $\bar{E}_{\bar{\mathbf{x}}^p}[U(\mathbf{D}^k, \mathbf{x})] \approx \max_{1 \leq a \leq s} E_a$. A schematic of the double-loop sampling process is sketched in Figure 4.

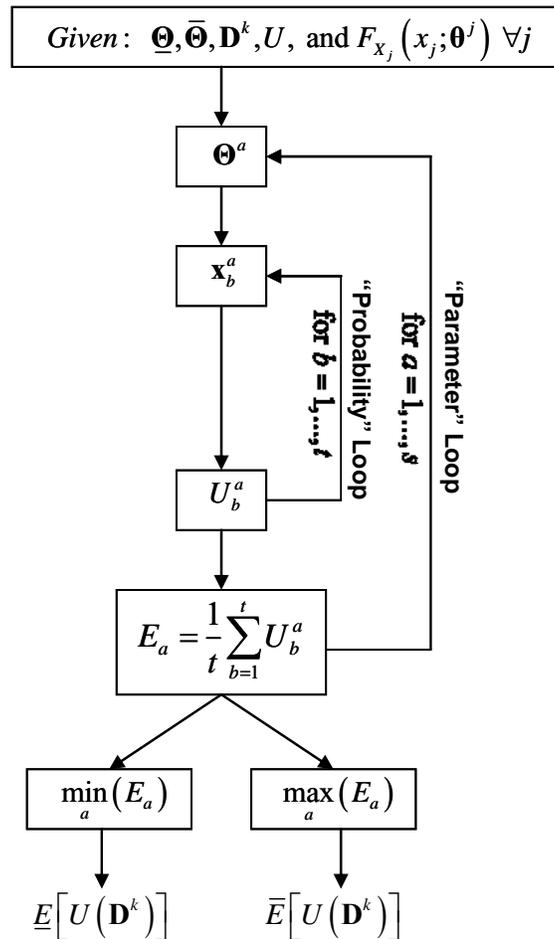


Figure 4. Diagram of double-loop sampling method.

3.3. OPTIMIZING IN THE “PARAMETER” LOOP

In an attempt to overcome the intractability of double-loop sampling for high-dimensional problems, we propose replacing the sampling in the “parameter” loop with an optimization algorithm. The “parameter” loop optimizer is used to locate the points Θ^l and Θ^u in the parameter space that result in the smallest and largest expected utilities, E_l and E_u . These utilities are then used to approximate the lower and upper expected utilities for design \mathbf{D}^k :

$$\begin{aligned} \underline{E}_{\mathbf{x}}^p [U(\mathbf{D}^k, \mathbf{x})] &\approx E_l \\ \bar{E}_{\mathbf{x}}^p [U(\mathbf{D}^k, \mathbf{x})] &\approx E_u. \end{aligned}$$

Essentially, the modified double-loop sampling method is the same as in pure double-loop sampling except that Θ^a is updated intelligently. The modified approach requires the solution of the following two optimization problems:

- (1) minimize $E = f(\Theta) \Rightarrow E_l$
 $\Theta \in [\underline{\Theta}, \bar{\Theta}]$
- (2) maximize $E = f(\Theta) \Rightarrow E_u$
 $\Theta \in [\underline{\Theta}, \bar{\Theta}]$

where $\underline{\Theta}$ and $\bar{\Theta}$ are the upper and lower bounds on the parameter space.

Numerically, solving these optimization problems poses two challenges: the objective function, $E = f(\Theta)$, 1) is approximated non-deterministically and 2) could have local extrema. Different random deviates in the “probability” loop will result in different approximations of E_a for a given vector of parameters, Θ^a . For gradient-based optimizers, this is problematic since the approximation to the objective function will develop sharp local gradients. One possible solution to challenge 1) is to use the same set of random deviates for each step of the optimization algorithm. This of course introduces a bias into the resulting E_l and E_u , but this bias can be made arbitrarily small by increasing the number of “probability” loop samples, t . The second challenge to solving these optimization problems is due to the nature of the true objective function. For realistic engineering problems, $E = f(\Theta)$ is often multi-modal. One possible solution to challenge 2) is to repeat the optimizations from multiple starting points, Θ^1 . Both of these solutions have proven to be effective in the design of an off-road vehicle gearbox (Rekuc, Aughenbaugh et al. 2006).

3.4. WEAKNESSES OF THE ALGORITHM

For many problems, the modified double-loop sampling algorithm described above will more efficiently locate the minimal and maximal sets of distribution parameters. However, the

modified approach retains some of the weaknesses of the pure double-loop sampling and even introduces some new difficulties.

Like pure double-loop sampling, the modified approach assumes a known dependence between the uncertain quantities involved in the computation. If a parameterized joint distribution function of all uncertain quantities were available, it would be compatible with either approach, but for practical problems it is almost never the case that the DM knows a fully characterized joint distribution. The dependency bounds approach, as described in section 1, makes no assumptions about the dependence between uncertain bounds. Indeed, dependency bounds are *best-possible* bounds that contain the results of the computation under any possible case of dependency. For problems in which the computation involves variables with possibly strong but unknown dependency, the methods of Williamson and Berleant maintain a distinct advantage over both the pure and modified double-loop sampling methods.

Also, like pure double-loop sampling, the modified double-loop sampling method might become too computationally expensive for high-dimensional problems. Replacing the “parameter” loop with an optimizer should result in decreased computational cost due to the decreased number of function evaluations, but optimization over a high-dimensional space can itself remain costly. The most that can be claimed of the modified double-loop sampling approach is that it allows for the solution of a wider class of problems than pure double-loop sampling.

Although the modified double-loop sampling method retains some of the weaknesses of pure double-loop sampling, it also introduces an additional difficulty. Specifically, the functions to be optimized, $E = f(\Theta)$, are complex and non-linear and therefore multi-modal. This means that the optimization problems become global optimization problems. Depending on the complexity of the global optimization problem, the modified double-loop sampling method might be computationally infeasible. Although many sophisticated algorithms for solving global optimization problems have been developed (see (Pinter 1996; Horst, Pardalos et al. 2000; Hansen and Walster 2004)), for many problems with relatively few local minima, it is often sufficient to repeat the optimization from multiple starting points.

4. Discussion and Future Work

In this paper, we have introduced and formally described a computational design problem. The goal of this research is to develop computational strategies for propagating imprecise beliefs through design decision models. We argued that the currently available computational methods are unsatisfactory, and an alternative approach was introduced. The main purpose of this paper has been to clarify and communicate the problem, but we do not yet feel that the research question has been satisfactorily answered. Further work remains to be done. In particular, the proposed method needs to be numerically validated, the problem of global optimization needs to be addressed, methods for parameterizing more general p-boxes need to be studied, and means of accounting for unknown dependence need to be developed. While we believe that the

optimization method proposed in this paper is an improvement over the available methods for some classes of design problems, we do not yet feel that we have found a fully satisfactory solution. Hopefully, this paper will lead to development of new alternative strategies for computing with imprecise information.

Numerical validation

Before the advantage of using the modified double-loop sampling method can be confirmed, numerical experiments must first be completed. Essentially, two questions need to be asked: does the modified double-loop sampling method provide results that are reasonably close to the theoretical best-possible bounds, and is the modified double-loop sampling method substantially more computationally efficient than pure double-loop sampling? Some experimentation with these methods has already been carried out (Rekuc, Aughenbaugh et al. 2006), but thorough validation requires a more systematic study.

The proposed experiments will involve three stages. First, in order to assure accuracy, the modified and pure double-loop sampling methods will be compared with the best-possible bounds approaches for a simple sum of p -boxes. It needs to be shown that an arbitrarily small degree of error can be achieved with both of these methods using only a reasonable number of samples. At this stage in the experimentation, a study of the relative efficiency of the two double-loop sampling methods can be conducted. The two sampling methods will be compared in terms of the number of function evaluations required as well as the overall CPU processing time.

The second stage of the proposed numerical validation, will involve another simple algebraic model. The second model will be sufficiently more complex so as to involve several more uncertain quantities as well as repeated variables. The presence of repeated variables results in over-conservative bounds for interval arithmetic operations. Since the best-possible bound methods for propagating imprecise probabilities make use of the operations of interval arithmetic, they result in over-conservative bounds in the presence of repeated variables. This is one aspect in which the double-loop sampling methods have an advantage over the dependency bounds convolution and the distribution envelope approaches. Although Ferson has solved repeated variable problems using subinterval reconstitution within the dependency bounds convolution algorithms (Ferson and Hajagos 2004), it is unclear how efficient this method is for handling problems with a large number of uncertain inputs. The purpose of this second stage is to test the efficiency of the modified double-loop sampling method for problems with a greater number of uncertain quantities. It will also be interesting to compare the accuracy and the efficiency of the modified double-loop sampling method to the dependency bounds convolution with subinterval reconstitution methods.

The third and final stage of the numerical validation of the modified double-loop sampling method will involve applying the method towards a realistic engineering design problem. At this stage, it will be impossible to compare to the dependency bounds convolution approach because of the large number of repeated variables. The previous two stages will test the accuracy of the

modified double-loop sampling as compared to the dependency bounds approach, and the final stage will test to see if the modified double-loop sampling is computationally efficient enough to propagate uncertainty through realistic engineering design problems.

The problem of global optimization

As mentioned previously, some numerical experimentation has been done with the modified double-loop method. Specifically, the method was applied towards the design of a gearbox. The utility curve of the gearbox as a function of the distribution parameters turned out to be multi-modal. A quick fix to this problem was attained by using multiple starting points, and this worked well for the gearbox problem. However, the gearbox utility model is relatively simple compared to other realistic design decision problems. Therefore, it is uncertain whether or not more sophisticated global optimization algorithms will be needed for complex design decision problems. If it is the case that many design utility models have a very large number of modes, then an efficient, reliable global optimization algorithm will need to be found that is compatible with the modified double-loop sampling method.

Parameterizing general p-boxes

Both the pure and modified double-loop sampling methods assume that the DM's imprecise beliefs can be represented as parameterized p-boxes. As was discussed previously, parameterized p-boxes are special cases of general p-boxes. It was argued that parameterized p-boxes arise frequently in practice, but not all realistic belief states can be easily represented as parameterized p-boxes. For instance, Dempster-Shafer structures are general p-boxes that result from the methods of evidence theory (Ferson, Kreinovich et al. 2002; Ferson, Hajagos et al. 2005), but there appears to be no straightforward way in which to model a Dempster-Shafer structure as a distribution with imprecise parameters. If any version of a double-loop sampling method is to be generally applicable, a means of parameterizing more general p-boxes needs to be discovered.

Accounting for known and unknown dependence

One of the primary advantages of the dependency bounds approaches is that they allow for the determination of theoretical best-possible bounds on the resultant p-box under any state of dependence between the uncertain quantities. By comparison, the double-loop sampling methods assume some dependence between the uncertain quantities. This is in violation of the problem statement presented in section 2.5. If a joint distribution is known, then sampling in the "probability" loop can take into account that dependence information by simply sampling from the joint distribution. However, joint distributions are not often known for engineering design problems. It is therefore desirable to further modify the double-loop sampling method such that it provides something similar to best-possible bounds in cases of unknown dependence.

Possibility of alternative approaches

As was discussed in section 1.5, the purpose of this paper has been to pose a problem. While a possible solution to that problem has been presented, obstacles still remain to putting the proposed method into practice. Engineering designers desire a method that allows for reliable, efficient propagation of their imprecise beliefs through complex engineering design models. The dependency bounds convolution and distribution envelope approaches are efficient and provide best-possible bounds, but they are not currently compatible with black-box models. Additionally, these methods face the dilemma of interval arithmetic with repeated variables. The pure and modified double-loop sampling methods discussed in this paper are compatible with black-box models, but they seem to be inefficient for complex engineering design problems. It is our hope that by presenting this problem statement to the reliable engineering computing community, alternative approaches will be suggested.

Acknowledgements

This work was supported in part by NSF grant DMI-0522116. The authors would like to thank Jason Aughenbaugh, Jay Ling, and Rich Malak for their contributions.

References

- Alefeld, G. and J. Herzberger. *Introduction to Interval Computations*. New York, Academic Press, 1983.
- Aughenbaugh, J. M. and C. J. J. Paredis. The Value of Using Imprecise Probabilities in Engineering Design. *ASME DETC 2005*, Long Beach, California, USA, IDETC2005-85354, 2005.
- Aughenbaugh, J. M. and C. J. J. Paredis. Why Are Intervals and Imprecision Important in Engineering Design? *Reliable Engineering Computing Workshop*, Savannah, GA, USA, 2006.
- Berger, J. O. *Statistical Decision Theory and Bayesian Analysis*, Springer, 1985.
- Berleant, D. Automatically Verified Reasoning with Both Intervals and Probability Density. *Interval Computations*(2): 48-70, 1993.
- Berleant, D. and H. Cheng. A Software Tool for Automatically Verified Operations on Intervals and Probability Distributions. *Reliable Computing* 4(1): 71-82, 1998.
- Berleant, D. and C. Goodman-Strauss. Bounding the Results of Arithmetic Operations on Random Variables of Unknown Dependency Using Intervals. *Reliable Computing* 4(2): 147-165, 1998.

- Berleant, D., L. Xie and J. Zhang. Stool: A Tool for Distribution Envelope Determination (Denv), an Interval-Based Algorithm for Arithmetic on Random Variables. *Reliable Computing* **9**(2): 91-108, 2003.
- Berleant, D. and J. Zhang. Representation and Problem Solving with Distribution Envelope Determination (Denv). *Reliability Engineering & System Safety* **85**(1-3): 153-168, 2004.
- Cooke, R. The Anatomy of the Squeezel - the Role of Operational Definitions in Representing Uncertainty. *Reliability Engineering & System Safety* **85**(1-3): 313, 2004.
- de Finetti, B. Foresight. Its Logical Laws, Its Subjective Sources (Translated). *Studies in Subjective Probability*. H. E. Kyburg and H. E. Smokler, E. Krieger Publishing Company, 1980.
- Dubois, D. and H. Prade. *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenum Press, 1988.
- Ferson, S. RAMAS Risk Calc. New York, Lewis Publishers, 2002.
- Ferson, S. and S. Donald. Probability Bounds Analysis. *International Conference on Probabilistic Safety Assessment and Management (PSAM4)*, New York, NY, Springer-Verlag, 1998.
- Ferson, S. and L. Ginzburg. Different Methods Are Needed to Propagate Ignorance and Variability. *Reliability Engineering and System Safety* **54**(2-3), 1996.
- Ferson, S., L. Ginzburg, V. Kreinovich, H. T. Nguyen and S. A. Starks. Uncertainty in Risk Analysis: Towards a General Second-Order Approach Combining Interval, Probabilistic, and Fuzzy Techniques. *2002 IEEE International Conference on Fuzzy Systems: FUZZ-IEEE'02, May 12-17 2002*, Honolulu, HI, United States, Institute of Electrical and Electronics Engineers Inc., 2002.
- Ferson, S. and L. R. Ginzburg. Different Methods Are Needed to Propagate Ignorance and Variability. *Reliability Engineering & System Safety* **54**(2-3): 133-144, 1996.
- Ferson, S. and L. R. Ginzburg. Different Methods Are Needed to Propagate Ignorance and Variability [in Risk Analysis]. *Reliability Engineering & System Safety* **54**(2-3): 133-44, 1996.
- Ferson, S. and J. Hajagos. Arithmetic with Uncertain Numbers: Rigorous and (Often) Best Possible Answers. *Reliability Engineering & System Safety* **85**(1-3): 135-152, 2004.
- Ferson, S., J. Hajagos, D. S. Myers and W. T. Tucker. Constructor: Synthesizing Information About Uncertain Variables, Applied Biomathematics for Sandia National Laboratories, 2005.
- Ferson, S., V. Kreinovich, L. Ginzburg, D. S. Myers and K. Sentz. Constructing Probability Boxes and Dempster-Shafer Structures. Albuquerque, New Mexico, Sandia National Laboratories, 2002.
- Ferson, S., R. B. Nelsen, J. Hajagos, D. Berleant, J. Zhang, W. T. Tucker, L. Ginzburg and W. L. Oberkampf. Dependence in Probabilistic Modeling, Dempster-Shafer Theory, and Probability Bounds Analysis. Albuquerque, Sandia National Laboratories, 2004.

- Fishman, G. S. *Monte Carlo: Concepts, Algorithms, and Applications*. New York, Springer, 1996.
- Hansen, E. and G. W. Walster. *Global Optimization Using Interval Analysis*. New York, Marcel Dekker, Inc., 2004.
- Hazelrigg, G. A. A Framework for Decision-Based Design. *Journal of Mechanical Design* **120**(4): 653-658, 1998.
- Helton, J. C. and F. J. Davis. Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems. *Reliability Engineering & System Safety* **81**: 23-69, 2003.
- Hofer, E. When to Separate Uncertainties and When Not to Separate. *Reliability Engineering & System Safety* **54**(2-3): 113-118, 1996.
- Hofer, E., M. Kloos, B. Krzykacz-Hausmann, J. Peschke and M. Woltereck. An Approximate Epistemic Uncertainty Analysis Approach in the Presence of Epistemic and Aleatory Uncertainties. *Reliability Engineering & System Safety* **77**(3): 229-38, 2002.
- Hoffman, F. O. and J. S. Hammonds. Propagation of Uncertainty in Risk Assessments: The Need to Distinguish between Uncertainty Due to Lack of Knowledge and Uncertainty Due to Variability. *Risk Analysis* **14**(5): 707-712, 1994.
- Horst, R., P. M. Pardalos and N. V. Thoai. *Introduction to Global Optimization*. Dordrecht, Kluwer, 2000.
- Kearfott, R. B. and V. Kreinovich. *Applications of Interval Computations*. AH Dordrecht, The Netherlands, Kluwer Academic Publishers, 1996.
- Kreinovich, V. and S. Ferson. A New Cauchy-Based Black-Box Technique for Uncertainty in Risk Analysis. *Reliability Engineering & System Safety* **85**: 267-279, 2004.
- Kreinovich, V., S. Ferson, L. Ginzburg, H. Schulte, M. Barry and H. Nguyen. From Interval Methods of Representing Uncertainty to a General Description of Uncertainty. *International Conference on Information Technology*, Bhubaneswar, India, McGraw-Hill, 1999.
- Law, A. M. and D. W. Kelton. *Simulation Modeling and Analysis*. St. Louis, McGraw-Hill, 2000.
- Levi, I. *The Enterprise of Knowledge, an Essay on Knowledge, Credal Probability, and Chance*. Cambridge, MA, MIT Press, 1980.
- Marston, M., J. K. Allen and F. Mistree. The Decision Support Problem Technique: Integrating Descriptive and Normative Approaches in Decision Based Design. *Engineering Valuation and Cost Analysis* **3**: 107-129, 2000.
- Mistree, F., W. F. Smith, B. A. Bras, J. K. Allen and D. Muster. Decision-Based Design. A Contemporary Paradigm for Ship Design. *1990 SNAME Annual Meeting, Oct 31-Nov 3 1990*, San Francisco, CA, USA, Publ by Soc of Naval Architects & Marine Engineers, Jersey City, NJ, USA, 1990.
- Moore, R. *Methods and Applications of Interval Analysis*. Philadelphia, Society for Industrial and Applied Mathematics, 1979.

- Muhanna, R. L. and R. L. Mullen. Interval Methods for Reliable Computing. *Engineering Design Reliability Handbook*. E. Nikolaidis, D. M. Ghiocel and S. Singhal. New York, NY, CRC Press: 12.1-12.24, 2004.
- Parry, G. W. The Characterization of Uncertainty in Probabilistic Risk Assessment of Complex Systems. *Reliability Engineering and System Safety* **54**(2-3): 119-126, 1996.
- Pinter, J. D. *Global Optimization in Action*. Dordrecht, Kluwer Academic Publishers, 1996.
- Regan, H. M., S. Ferson and D. Berleant. Equivalence of Methods for Uncertainty Propagation of Real-Valued Random Variables. *International Journal of Approximate Reasoning* **36**(1): 1-30, 2004.
- Rekuc, S. J. *Eliminating Design Alternatives under Interval-Based Uncertainty*. Thesis. Master of Science, Mechanical Engineering, Georgia Institute of Technology, 2005.
- Rekuc, S. J., J. M. Aughenbaugh, M. Bruns and C. J. J. Paredis. Eliminating Design Alternatives Based on Imprecise Information. *Society of Automotive Engineering World Congress*, Detroit, Michigan, 06M-269, 2006.
- Savage, L. J. *The Foundation of Statistics*. New York, Dover Publications, 1972.
- Sobek, D. K., A. Ward and J. K. Liker. Toyota's Principles of Set-Based Concurrent Engineering. *Sloan Management Review*, 1999.
- Sobek, D. K. I. and A. C. Ward. Principles from Toyota's Set-Based Concurrent Engineering Process. *ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Irvine, California, 1996.
- Springer, M. D. *The Algebra of Random Variables*. New York, John Wiley & Sons, 1979.
- Trejo, R. and V. Kreinovich. Error Estimations for Indirect Measurements: Randomized Vs. Deterministic Algorithms for 'Black-Box' Programs. *Handbook on Randomized Computing*. S. Rajasekaran, P. Pardalos, J. Reif and J. Rolim, Kluwer: 673-729, 2001.
- Troffaes, M. C. M. Decision Making with Imprecise Probabilities: A Short Review. *The Society for Imprecise Probability Theory and Applications*. **2**: 4-7, 2004.
- Walley, P. *Statistical Reasoning with Imprecise Probabilities*. London, Chapman and Hall, 1991.
- Williamson, R. C. *Probabilistic Arithmetic*. Thesis., Electrical Engineering, University of Queensland, 1989.
- Williamson, R. C. and T. Downs. Probabilistic Arithmetic I: Numerical Methods for Calculating Convolutions and Dependency Bounds". *International Journal of Approximate Reasoning* **4**: 89-158, 1990.
- Winkler, R. L. Uncertainty in Probabilistic Risk Assessment. *Reliability Engineering & System Safety* **54**(2-3): 127-132, 1996.
- Yager, R. R. Arithmetic and Other Operations on Dempster-Shafer Structures. *International Journal of Man-Machine Studies* **25**: 357-366, 1986.
- Zadeh, L. A. Fuzzy Sets. *Information and Control* **8**: 338-353, 1965.

Sampling Without Probabilistic Model

MICHAEL BEER

*Institut für Statik und Dynamik der Tragwerke, Technische Universität Dresden, 01062 Dresden, Germany,
email: Michael.Beer@tu-dresden.de*

Abstract. In this paper a novel technique for random vector sampling starting from rare data is presented. This model-free sampling technique is developed to operate without a probabilistic model. Instead of estimating a distribution function, the information contained in a given small sample is extracted directly to produce the sampling result as a second sample of considerably larger size that completely reflects the properties of the original small sample. As a further enhancement, the new sampling technique is extended to processing imprecise data.

Model-free sampling can be coupled to stochastic structural analysis and safety assessment by application to input data or to result data. In the case of limited data, for instance, due to a high numerical cost of the underlying computational model, the novel technique can be applied to generate a proper estimation of stochastic structural responses and, thanks to a sound reproduction of distribution tails, of structural reliability. In this context it can provide a basis for increasing the numerical efficiency of Monte Carlo simulations in computational stochastic mechanics.

The usefulness of the model-free sampling technique is underlined by means of numerical examples.

Keywords: Sampling; Monte Carlo simulation; Imprecise data; Fuzzy randomness; Uncertain structural analysis; Safety assessment.

1. Introduction

Simulation techniques often offer the only possibility for solving problems in which random properties must be taken into account. Indeed, Monte-Carlo simulation and further developments thereof have become versatile tools for solving a variety of problems in a wide range of engineering disciplines, see (Schuëller and Spanos, 2001).

An essential precondition for obtaining realistic results from a simulation is the availability of statistically-validated probability distributions for the input variables. The specification of these distributions thus plays an essential role, see (Schuëller, 2001b). For determining reliably parameters and forms of probability distributions, extensive data in the form of samples are required. This enables using well-developed and sophisticated methods of statistical estimation theory and test theory, which operate parametrically or non-parametrically (Mood et al., 1974). Further, the numerical procedure for processing the specified random quantities in structural analysis and safety assessment must be computationally efficient to enable the stochastic analysis of large and nonlinear systems (Schenk et al., 2005; Schenk and Schuëller, 2005; Schuëller et al., 2003). Only if a sufficient amount of structural response data is produced, their stochastic properties can be identified reliably, and failure probabilities can be estimated appropriately with the aid of statistical methods.

In any case, problems may primarily occur in the following three situations. First, the available information is limited in the form of small samples (Problem I). Second, structural response data can only be produced to a limited extent due to a high computational cost in analyzing the underlying structural problem in correspondence with the simulation of random input quantities (Problem II). Third, the sample elements are characterized by uncertainty or imprecision (Problem III). As a result, probability distributions for input variables and structural responses, or probabilities of defined events such as structural failure cannot be specified to a sufficient degree of reliability.

1.1. PROBLEM I – SMALL SAMPLES

In this case statistical estimations and tests based on small samples may yield vague and ambiguous results. For an appropriate level of confidence wide intervals for the estimated values are obtained. The variety of possible probabilistic models can almost not sufficiently narrowed with the aid of tests. This applies, in particular, if the distribution type is not pure in the form of a compound or multimodal distribution, or if general, for example nonlinear, dependencies in multi-dimensional cases are present. The less information the sample contains the more subjectivity is introduced with the specification of a certain probabilistic model. On the other hand, there is no evidence that the information that is actually contained in the sample is extracted completely but only to a certain degree. The results obtained on such basis may vary dramatically. Approaches to remedy this problem aim at determining bounds for the possible range of stochastic models and prognoses. A distinction can be made here between pure probabilistic methods (Deodatis et al., 2003; Papadopoulos et al., 2005; Red-Horse and Benjamin, 2004), which are focused on finding the bounds with different externally applied search strategies, and methods based on extended uncertainty models such as p-box (Berleant and Zhang, 2004), random sets (Hall and Lawry, 2004; Tonon et al., 2000), sets of probability measures (Fetz and Oberguggenberger, 2004), or fuzzy randomness (Möller and Beer, 2004), which cover the possible range of probabilistic models at once and intrinsically contain the search for probabilistic bounds. For this intrinsic search, a generally applicable and numerically efficient optimization algorithm has been developed as modified evolution strategy (Möller et al., 2000), the usefulness of which has already been shown, for example, in safety assessments coupled to a nonlinear structural analysis (Möller et al., 2003). Moreover, the model of fuzzy randomness provides a basis for an evaluation of those problems on several levels of subjective confidence in an encapsulated manner. Further, a variety of methods based on Bayesian theory (Bernardo and Smith, 1994) can be employed if subjective information is available beyond the small sample.

1.2. PROBLEM II – HIGH COMPUTATIONAL COST

The solution to Problem II comprises a wide variety of methods to increase numerical efficiency of stochastic structural analysis and safety assessment (Schuëller, 2001a). The corresponding developments primarily concern enhancements in Stochastic Finite Element Methods and in the numerical simulation of stochastic processes, which have already reached practical relevance in solving engineering problems (Ghanem and Spanos, 1991; Schenk and Schuëller, 2005). Their practical applicability substantially hinges on an efficient representation of the random input quantities. In this context, spectral representations of stochastic processes have attracted considerable attention, which particularly refers to Karhunen-Loève or Polynomial Chaos expansion (Du et al., 2005; Field Jr. and Grigoriu, 2004; Gutierrez and Zaldivar, 2000; Phoon et al., 2005; Schuëller et al., 2003; Spanos and Ghanem, 1989). For reliability analysis, which focuses on rare events,

further efficiency improvements of simulation techniques are pursued. Among several strategies, the class of variance reducing methods such as Importance Sampling and variants thereof (Rackwitz, 2001; Schuëller, 2001a) probably represents the most popular kind of approaches. As these developments are not directly related to the approach pursued in this paper but only act as a motivation, their consideration is not extended at this point.

1.3. PROBLEM III – IMPRECISE DATA

This problem exceeds the limits of traditional methods in uncertainty quantification and processing, and represents a reasearch field of increasing interest. Traditionally, imprecision or uncertainty of sample elements is either neglected totally or taken into account approximately by selecting "probably adverse values" with respect to structural responses and safety measures from a possible value range. However, the actual impact of such a selection of input parameters can generally not be evaluated at the pre-stage of a simulation. On the other hand, the question arises as to how to model that uncertainty or imprecision realistically. It appears, for example, in situations in which the precision of measuring devices is strongly limited, the measuring points cannot be defined precisely (rough surfaces in thickness measurements), the expert evaluations influence the value specification, the measured values are gained under dubious conditions, and linguistic assessments are accounted for. In those cases the data possess random properties and non-random properties simultaneously. A pure probabilistic solution by applying the aforementioned approaches for dealing with limited information in the form of small samples is thus somewhat critical. Only Bayesian methods (Bernardo and Smith, 1994) are capable of incorporating subjective uncertainty, but still in terms of probability, which contradicts the non-random nature of some information. For a more pertinent uncertainty modeling in the case of non-probabilistic phenomena generalized uncertainty models have been developed (Fellin et al., 2005; Helton and Oberkampf, 2004), which are related to or covered by the framework of evidence theory. A comprehensive direct modeling of the imprecision or uncertainty of the individual elements of a random sample can be realized with the aid of a fuzzy randomness approach (Möller and Beer, 2004). Statistical investigations of uncertain or imprecise data and of properties of fuzzy random variables are, to a great extend, in an initial stage of development. Related research in this regard may be found in (Bandemer and Näther, 1992), in (Viertl, 1996), and in (Körner, 1997). These developments concern the analysis of imprecise data, the definition of statistical parameters, and the investigation of statistical laws for fuzzy random variables. Publications discussing the simulation of fuzzy randomness are rare. An approach evaluating fuzzy probability distribution functions on a trajectory-by-trajectory basis is presented in (Sickert et al., 2003). Numerical investigations of statistical properties of fuzzy random variables based on simulation are discussed in (Colubi et al., 2002). However, these methods require prior knowledge about the fuzzy probability distributions or the fuzziness of the realizations to be generated. General techniques for generating fuzzy realizations of fuzzy random variables are not known at the present time. Moreover, the application of traditional sampling methods to the numerical generation of fuzzy realizations encounters considerable difficulties. For instance, the numerical effort for estimating fuzzy parameters and fuzzy probability distributions from fuzzy-valued samples (fuzzy samples) is significantly high, in particular, when interaction between the fuzzy parameters is taken into account. Further, the simulation of fuzzy realizations starting from fuzzy probability distribution functions is not unique. That is, different fuzzy samples may have identical empirical fuzzy probability distribution functions. These conflicts hinder the pursuing of traditional sampling and simulation approaches.

1.4. SOLUTION IDEA

Despite considerable developments in answering the aforementioned three problems, an overall satisfying solution does not exist. In the following an attempt is made to develop a basis for a sampling technique to improve uncertainty processing in structural analysis and safety assessment in those problematic cases. The novel sampling technique is intended to circumvent an explicit specification of a probabilistic model, which avoids an introduction of subjectivity and motivates its denotation as "model-free sampling". Further, it should be capable of attaining appropriate results starting from samples of small size that may consist of uncertain or imprecise data.

The development starts from the basic statistical assumption that all information is contained in the sample. On the basis of a small sample a second sample of considerably larger size is numerically generated that completely reflects the statistical properties and uncertainty characteristics of the original small sample. This sampling technique can be applied to rare input data as well as to rare result data of a stochastic structural analysis and might thus be helpful as a preprocessor or as a postprocessor in combination with established simulation methods in diverse cases to improve estimations of stochastic structural responses and of structural reliability, and to increase the numerical efficiency of the computations. For enhancing the model-free sampling technique to processing imprecise data the generalized uncertainty model fuzzy randomness is taken as a basis, which enables to transfer stochastic uncertainty and non-stochastic uncertainty of the input data completely and simultaneously to the results of structural analysis and safety assessment. Finally, predictions of uncertain stochastic structural responses and of uncertain structural reliability are obtained.

2. Numerical Procedure

The basic concept of the model-free sampling technique is to generate the sampling result directly from a given sample instead of estimating a probability distribution and performing the sampling according to this. The characteristics of a population are described by a sufficiently large sample. As the mathematical model of a distribution function is not employed herein, conventional statistical estimations are dispensed with. The concept of statistical estimation is applied in a generalized sense.

The starting point is the observed sample

$$\underline{S}_0 = \{s_{0,i}, i = 1, \dots, n_0\} \quad (1)$$

of size n_0 as a set of realizations $s_{0,i} = \underline{x}_{0,i}$ in \mathbb{R}^n of the underlying continuous random vector \underline{X}_0 with unknown properties. A second concrete sample

$$\underline{S}_1 = \{s_{1,k}, k = 1, \dots, n_1\} \quad (2)$$

of a considerably larger size $n_1 \gg n_0$ is then sought that represents the original sample \underline{S}_0 "as well as possible". That is, the new sample \underline{S}_1 is expected to exhibit statistical characteristics "comparable" to \underline{S}_0 . This is realized by the following heuristic iterative approach with the superscript ^[1] indicating the iteration step.

1. The starting point is an arbitrary estimate

$$\underline{S}_1^{[0]} = \{ \underline{s}_{1,k}^{[0]}, k = 1, \dots, n_1 \} \tag{3}$$

for the sample \underline{S}_1 . This is broadly specified without consideration of the information contained in the observed sample \underline{S}_0 . All sample elements $\underline{s}_{1,k}^{[0]} = \underline{x}_{1,k}^{[0]}$ of $\underline{S}_1^{[0]}$ should possess the same information content. That is, they should exhibit the same probability density $f_1^{[0]}$ in their immediate surroundings,

$$\int_{\|\underline{\delta}\| \leq \|\underline{\varepsilon}\|} f_1^{[0]}(\underline{s}_{1,p}^{[0]} + \underline{\delta}) d\underline{\delta} = \int_{\|\underline{\delta}\| \leq \|\underline{\varepsilon}\|} f_1^{[0]}(\underline{s}_{1,q}^{[0]} + \underline{\delta}) d\underline{\delta} \tag{4}$$

$$\forall \underline{s}_{1,p}^{[0]}, \underline{s}_{1,q}^{[0]} \in \underline{S}_1^{[0]}, \|\underline{\varepsilon}\| \ll 1.$$

This leads to the specification of $\underline{S}_1^{[0]}$ by continuous uniform distribution over a sufficiently large (physically meaningful), bounded domain $\underline{D} \subset \mathbb{R}^n$ of possible (not excludable) realizations of the random vector \underline{X}_0 represented by \underline{S}_0 ,

$$(\underline{X}_1 \sim U(\underline{D})) \rightarrow \underline{S}_1^{[0]}. \tag{5}$$

2. The sample $\underline{S}_1^{[0]}$ is compared with the observed sample \underline{S}_0 . The purpose of this comparison is to obtain a measure $G^{[0]}$ for the statistical dissimilarity between the samples $\underline{S}_1^{[0]}$ and \underline{S}_0 . For this dissimilarity measure, a real valued function

$$G^{[.]} = g(\underline{S}_0, \underline{S}_1^{[.]}) : (\underline{S}_0, \underline{S}_1^{[.]}) \rightarrow \mathbb{R} \tag{6}$$

is selected which yields a global minimum for $G^{[.]}$ if the samples $\underline{S}_1^{[.]}$ and \underline{S}_0 are "as similar as possible" in a statistical sense. That is, $G^{[.]}$ is intended to be minimal if $\underline{S}_1^{[.]}$ and \underline{S}_0 originate from the same population \underline{X} with probability one,

$$P(\underline{S}_0 \subset \underline{X} \wedge \underline{S}_1^{[.]} \subset \underline{X}) \rightarrow 1 \Rightarrow G^{[.]} \Rightarrow \text{MIN}. \tag{7}$$

Due to the fact that intended application is for samples consisting of imprecise data, established statistical test methods cannot be implemented.

3. The sample $\underline{S}_1^{[0]}$ is modified in such a way that a subset

$$\underline{S}_1^{[0]-} = \{ \underline{s}_{1,k_1}^{[0]-}, \dots, \underline{s}_{1,k_{m_1}}^{[0]-} \} \subset \underline{S}_1^{[0]} \tag{8}$$

of m_1 elements $\underline{s}_{1,k}^{[0]-}$ (stipulated number with $m_1 \ll n_1$) are specified by discrete uniform distribution over the indices k of the elements $\underline{s}_{1,k}^{[0]}$ of $\underline{S}_1^{[0]}$,

$$(X^- \sim U(1, 2, \dots, n_1)) \rightarrow \{k_1, k_2, \dots, k_{m_1}\}, \tag{9}$$

and are removed from $\underline{S}_1^{[0]}$ to obtain the reduced sample

$$\underline{S}_{1,\text{red}}^{[0]} = \underline{S}_1^{[0]} \setminus \underline{S}_1^{[0]-}. \quad (10)$$

As an replacement for the removed elements $\underline{s}_{1,k}^{[0]-}$, a set

$$\underline{S}_1^{[0]+} = \{ \underline{s}_{1,k_1}^{[0]+}, \dots, \underline{s}_{1,k_{m_1}}^{[0]+} \} \quad (11)$$

of m_1 new elements $\underline{s}_{1,k}^{[0]+}$ are generated randomly – again with the aid of a uniform distribution over the domain \underline{D} of possible realizations specified in Step 1,

$$(\underline{X}_1 \sim U(\underline{D})) \rightarrow \underline{S}_1^{[0]+}. \quad (12)$$

Their union with the reduced sample $\underline{S}_{1,\text{red}}^{[0]}$ then yields the modified sample

$$\underline{S}_1^{[1]} = \underline{S}_{1,\text{red}}^{[0]} \cup \underline{S}_1^{[0]+}. \quad (13)$$

Then, the measure value $G^{[1]}$ is computed for the modified sample $\underline{S}_1^{[1]}$.

4. The measure values $G^{[1]}$ and $G^{[0]}$ are compared. If $G^{[1]} \geq G^{[0]}$, it is concluded that the modification in Step 3 has not led to an improved estimation for \underline{S}_1 . The modification is then nullified,

$$\underline{S}_1^{[0]} = \left(\underline{S}_1^{[1]} \setminus \underline{S}_1^{[0]+} \right) \cup \underline{S}_1^{[0]-}, \quad (14)$$

and a repeat modification of $\underline{S}_1^{[0]}$ is carried out according to Step 3. If $G^{[1]} < G^{[0]}$, on the other hand, the modified sample $\underline{S}_1^{[1]}$ yields an improved estimation compared with $\underline{S}_1^{[0]}$. The sample $\underline{S}_1^{[1]}$ is then taken as the basis for the next iteration step and modified anew according to the rules in Step 3 to produce $\underline{S}_1^{[2]}$. Again, the result is assessed. This procedure is repeated with an iteration counter r for successful modifications,

$$\begin{aligned} \underline{S}_1^{[r+1]} &= \{ \underline{s}_{1,1}^{[r+1]}, \dots, \underline{s}_{1,k}^{[r+1]}, \dots, \underline{s}_{1,n_1}^{[r+1]} \} \\ &= \left(\{ \underline{s}_{1,1}^{[r]}, \dots, \underline{s}_{1,k}^{[r]}, \dots, \underline{s}_{1,n_1}^{[r]} \} \setminus \{ \underline{s}_{1,k_1}^{[r]-}, \dots, \underline{s}_{1,k_{m_1}}^{[r]-} \} \right) \\ &\quad \cup \{ \underline{s}_{1,k_1}^{[r]+}, \dots, \underline{s}_{1,k_{m_1}}^{[r]+} \}, \end{aligned} \quad (15)$$

until it is no longer possible to obtain an improvement of \underline{S}_1 beyond $\underline{S}_1^{[r]}$. The dissimilarity measure $G^{[r]}$ then attains its minimum value. As the configuration of \underline{S}_1 that corresponds to the minimum of $G^{[r]}$ can only be realized with probability zero (continuous case), a termination limit is defined for the probability with which an improvement can be obtained. The iteration is terminated if the average success rate of modifications attains a predefined and sufficiently small value. Finally, the sample $\underline{S}_1^{[r^*]}$ obtained from the last successful modification is taken as the sampling result,

$$\underline{S}_1 = \underline{S}_1^{[r^*]}. \quad (16)$$

The random modifications of $\underline{S}_1^{[r]}$ within the iteration ensure that the goal sample \underline{S}_1 is obtained as a random sample in consistency with established sampling principles. By virtue of its general concept the model-free sampling technique is a priori not limited in its applicability.

3. Real-Valued Samples

The model-free sampling technique is developed first, to apply for processing real-valued samples. The samples are deemed real-valued in the sense that their elements are denoted by scalars or vectors consisting of real numbers. This enables assessing the sampling results with the aid of established test methods. In this manner, the effectiveness of the model-free sampling may be evaluated.

3.1. BASIC ASPECTS

The critical point of the proposed technique is to formulate an appropriate function for characterizing the statistical dissimilarity $G^{[r]}$ between the samples $\underline{S}_1^{[r]}$ and \underline{S}_0 in each iteration step r (see Step 2 in Section 2). This function $G^{[r]}$ according to Eq. (6) is required to possess the following four general properties:

1. The measure $G^{[r]}$ and established statistical test methods (homogeneity tests) must lead to basically analogous propositions regarding the statistical dissimilarity between $\underline{S}_1^{[r]}$ and \underline{S}_0 . These propositions must be free of contradictions.
2. The mathematical formulation of the dissimilarity measure $G^{[r]}$ must be extendable to apply for imprecise data in the form of fuzzy-valued samples. That is, the mathematical operations used in the definition of $G^{[r]}$ for the real-valued case must possess appropriate counterparts in fuzzy arithmetics.
3. $G^{[r]}$ is required to decrease at least tendentiously with decreasing statistical dissimilarity between $\underline{S}_1^{[r]}$ and \underline{S}_0 . For samples $\underline{S}_1^{[r]}$ and \underline{S}_0 originating from the same population the measure $G^{[r]}$ should take its global minimum value, see Eq. (7).
4. The mathematical structure of the measure $G^{[r]}$ should be as simple as possible to ensure a fast numerical evaluation and thus to keep the computational cost reasonably low.

To develop a measure $G^{[r]}$ that satisfies these requirements the following theoretical experiment is considered.

According to statistical estimation theory it is assumed that all available information is contained in the observed sample \underline{S}_0 . Then, the best description of \underline{S}_0 is its empirical distribution function $F_{\underline{S}_0}^{(e)}(\underline{x})$, as it is a complete and unique representation of the information in \underline{S}_0 . Moreover, in inferential statistics, the empirical distribution function is one of the most powerful estimators. If this $F_{\underline{S}_0}^{(e)}(\underline{x})$ is taken as the basis for sampling to numerically generate the sample \underline{S}_1 , and no smoothing is applied, the resulting sample \underline{S}_1 and the observed sample \underline{S}_0 possess identical empirical distributions (in the limit),

$$\lim_{n_1 \rightarrow \infty} F_{\underline{S}_1}^{(e)}(\underline{x}) = F_{\underline{S}_0}^{(e)}(\underline{x}). \quad (17)$$

This corresponds to two significant properties of the samples \underline{S}_0 and \underline{S}_1 with respect to each other. First, the positions of the elements of \underline{S}_0 and \underline{S}_1 coincide. Second, each element of \underline{S}_0 has the same number of uniquely assigned elements from the sampling result \underline{S}_1 . In the case of an underlying continuous random variable and an accordingly slightly smoothed empirical distribution $F_{\underline{S}_0}^{(e)}(\underline{x})$, the elements of the sampling result \underline{S}_1 are obtained in a close neighborhood of the elements of \underline{S}_0 with the same assignment property. Sampling results generated in this manner are high quality representations of the underlying sample \underline{S}_0 as may be shown by applying a variety of two-sample tests of homogeneity.

The measure $G^{[.]}$ is thus formulated based on the configuration of the sampling result \underline{S}_1 from the theoretical experiment. This provides two criteria for monitoring the dissimilarity $G^{[.]}$ between $\underline{S}_1^{[.]}$ and \underline{S}_0 , which are defined as an assignment criterion and a distance criterion.

3.2. ASSIGNMENT CRITERION

The assignment criterion evaluates some order in the element configuration in the samples $\underline{S}_1^{[.]}$ and \underline{S}_0 with respect to each other. Each element $\underline{s}_{0,i}$, $i = 1, \dots, n_0$ from sample \underline{S}_0 is supposed to have the same number $n_{\text{ass}}(\underline{s}_{0,i})$ of uniquely assigned elements $\underline{s}_{1,k}^{[.]}$, $k = 1, \dots, n_1$ from sample $\underline{S}_1^{[.]}$. The element assignment is defined on the basis of the Euclidean distance

$$d(\underline{s}_{0,i}, \underline{s}_{1,k}^{[.]}) = \|\underline{s}_{1,k}^{[.]} - \underline{s}_{0,i}\| \quad (18)$$

between the respective elements $\underline{s}_{1,k}^{[.]}$ and $\underline{s}_{0,i}$. For each $\underline{s}_{1,k}^{[.]}$ one assigned element $\underline{s}_{0,i}(\underline{s}_{1,k}^{[.]})$ is determined with

$$\underline{s}_{0,i}(\underline{s}_{1,k}^{[.]}) = \underline{s}_{0,i} \mid d(\underline{s}_{0,i}, \underline{s}_{1,k}^{[.]}) = \min_{i=1, \dots, n_1} [d(\underline{s}_{0,i}, \underline{s}_{1,k}^{[.]})], \quad (19)$$

see Figure 1. If Eq. (19) leads to a multiple assignment of elements $\underline{s}_{0,i}$ to the same $\underline{s}_{1,k}^{[.]}$, which occurs with probability zero in the continuous case but can appear in the numerical procedure due to limited computational precision, the element $\underline{s}_{0,i}$ with the smallest index i is selected for the assignment. The number $n_{\text{ass}}(\underline{s}_{0,i})$ may then be obtained by means of an indicator function,

$$n_{\text{ass}}(\underline{s}_{0,i}) = \sum_{k=1}^{n_1} I(\underline{s}_{0,i}, \underline{s}_{1,k}^{[.]}) \quad (20)$$

$$I(\underline{s}_{0,i}, \underline{s}_{1,k}^{[.]}) = \begin{cases} 1 & \text{if } \underline{s}_{0,i} = \underline{s}_{0,i}(\underline{s}_{1,k}^{[.]}) \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

The target value for the number $n_{\text{ass}}(\underline{s}_{0,i})$ is given by the ratio of the sample sizes n_1 and n_0 ,

$$n_{\text{ass}}^{\text{target}}(\underline{s}_{0,i}) = \frac{n_1}{n_0} \quad (22)$$

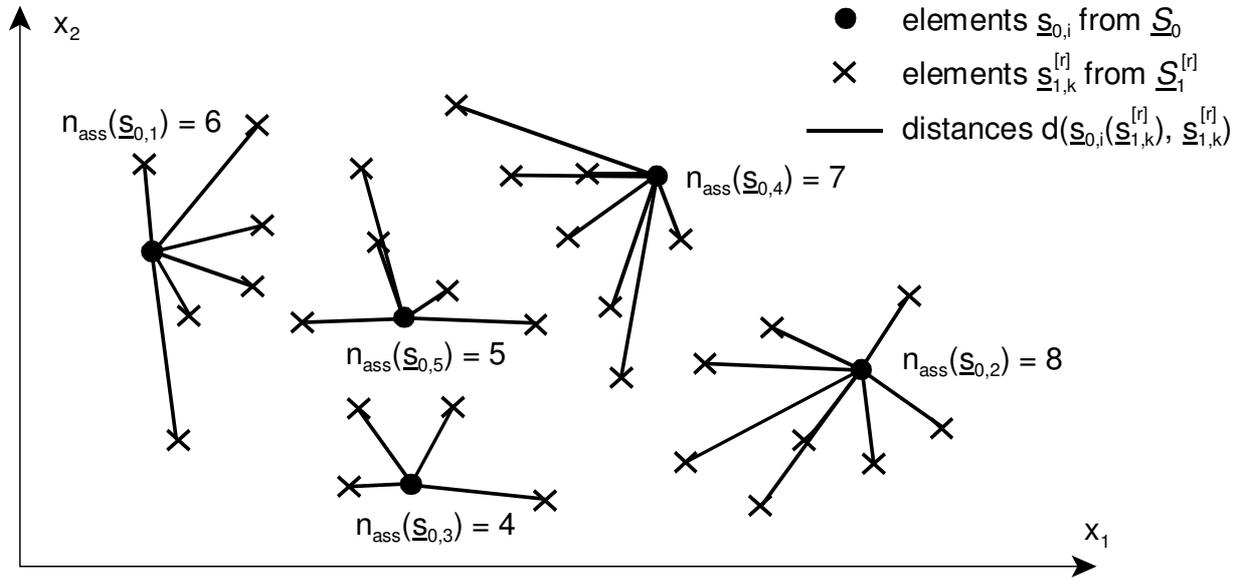


Figure 1. Assignment of sample elements

The assignment criterion is then defined as the total sum of the quadratic differences between the actual numbers $n_{\text{ass}}(s_{0,i})$ and the target value $n_{\text{ass}}^{\text{target}}(s_{0,i})$,

$$C_1^{[.]} = \sum_{i=1}^{n_0} \left(n_{\text{ass}}(s_{0,i}) - \frac{n_1}{n_0} \right)^2 \Rightarrow \text{MIN} . \tag{23}$$

The smallest possible value of $C_1^{[.]}$ depends on the sample sizes n_1 and n_0 . With the parameter

$$a \in \mathbb{N} \mid a \cdot n_0 \leq n_1 < (a + 1) \cdot n_0 \tag{24}$$

this limit is

$$\text{min}_C C_1 = -\frac{1}{n_0} (a \cdot n_0 - n_1)^2 + n_1 - a \cdot n_0 \tag{25}$$

In the special case that the size n_1 of sample $S_1^{[.]}$ is a whole multiple of the size n_0 of S_0 the value $\text{min}_C C_1$ is equal to zero.

3.3. DISTANCE CRITERION

The distance criterion supplements the assignment criterion by additionally evaluating the particular positions of the sample elements $s_{1,k}^{[.]}$ and $s_{0,i}(s_{1,k}^{[.]})$ with respect to each other. The distances between assigned sample elements are supposed to be as small as possible. Specifically,

$$C_2^{[.]} = \sum_{k=1}^{n_1} d \left(\underline{s}_{0,i} \left(\underline{s}_{1,k}^{[.]} \right), \underline{s}_{1,k}^{[.]} \right)^2 \Rightarrow \text{MIN}, \quad (26)$$

with $\underline{s}_{0,i} \left(\underline{s}_{1,k}^{[.]} \right)$ specifying the assignment of $\underline{s}_{1,k}^{[.]}$ to $\underline{s}_{0,i}$ determined with Eq. (19), see Figure 1. The smallest possible value of the distance criterion is zero.

3.4. COMPOSING THE DISSIMILARITY MEASURE

To define the dissimilarity measure $G^{[.]}$ for real-valued samples $\underline{S}_1^{[.]}$ and \underline{S}_0 the assignment criterion according to Eq. (23) and the distance criterion according to Eq. (26) are combined. As a standard formulation, the quantity

$$G^{[.]} = \sqrt{C_1^{[.]} + C_2^{[.]}} \quad (27)$$

is selected. An extension of Eq. (27) by introducing weighting factors for the criteria $C_1^{[.]}$ and $C_2^{[.]}$ has been investigated in several numerical tests; it has not been found particularly effective for improving the simulation results.

3.5. ASSEMBLING THE ITERATION PROCEDURE

The dissimilarity measure $G^{[.]}$ in Eq. (27) is implemented into the numerical procedure according to Steps 1 through 4 in Section 2. Moreover, the number m_1 of elements, see Step 3, which are modified in each iteration step, is not held constant during the iteration but varied frequently by a random selection of m_1 from a predefined range of values $[a, b]$ with the aid of a discrete uniform distribution,

$$(X_{m_1} \sim U(a, a + 1, \dots, b - 1, b)) \rightarrow m_1, \quad a, b \in \mathbb{N}. \quad (28)$$

As an alternative to the random generation of the m_1 new elements with the aid of a uniform distribution according to Eq. (12), the (slightly smoothed) current empirical distribution $F_{\underline{S}_1^{[r]}}^{(\bar{e})}(\underline{x})$ of the sample $\underline{S}_1^{[r]}$ from the last successful modification r can be used for a kind of bootstrap sampling,

$$\left(\underline{X}_1 \sim F_{\underline{S}_1^{[r]}}^{(\bar{e})}(\underline{x}) \right) \rightarrow \underline{S}_1^{[r]+}. \quad (29)$$

In this manner, use is made of the statistical information already gathered in $\underline{S}_1^{[r]}$ during the iteration, which leads to an increase of numerical efficiency. The termination limit in Step 4 is chosen to be 2% and is applied to the moving average of the recent 100 successful iteration steps.

4. Samples of Imprecise Data

4.1. MODELING IMPRECISE DATA

For dealing with imprecise data, we must select a suitable data model that combines the benefits of the well-established probabilistic approach with an appropriate modeling of non-frequentative uncertainty or

imprecision. From the class of available uncertainty models in this context, the concept of fuzzy random variables originally presented in (Kwakernaak, 1978) is selected for further investigation. This model possesses the advantage of simultaneously covering the models of real-valued random variables, intervals, fuzzy sets, rough sets, random sets, and convex models as special cases.

To define a fuzzy random variable the probability space $[\underline{\mathbf{X}}, \mathfrak{G}, P]$ is extended by the dimension fuzziness. If the space of the random elementary events, as in probabilistics, is described by Ω , a fuzzy random vector $\tilde{\underline{\mathbf{X}}}$ on the fundamental set $\underline{\mathbf{X}} = \mathbb{R}^n$ may be defined as the fuzzy result of the mapping

$$\Omega \rightarrow \mathbf{F}(\mathbb{R}^n) \tag{30}$$

where $\mathbf{F}(\mathbb{R}^n)$ is the set of all fuzzy numbers in \mathbb{R}^n . An ordered n-tupel of fuzzy numbers \tilde{x}_i is assigned to each (crisp) elementary event $\omega \in \Omega$. Every n-tupel $\tilde{\underline{x}}(\omega) = (\tilde{x}_1, \dots, \tilde{x}_n) \subseteq \underline{\mathbf{X}}$ is a realization of the fuzzy random vector $\tilde{\underline{\mathbf{X}}}$. Both objective and subjective information are accounted for simultaneously. The theory of fuzzy random variables permits the modeling of uncertain structural parameters which partly exhibit randomness but which cannot be described using real-valued random variables without an element of doubt. The randomness is "disturbed" by a fuzziness component.

A comprehensive discussion on fuzzy randomness particularly with regard to engineering problems may be found in (Möller and Beer, 2004). In this context the concepts of fuzzy structural analysis, see also (Möller et al., 2000), and fuzzy probabilistic safety assessment, see also (Möller et al., 2003), describe the processing of uncertain structural parameters with the aid of numerical procedures. This basis ensures an appropriate evaluation or further processing of the results from model-free sampling of fuzzy random variables within the framework of structural analysis and safety assessment.

The model-free sampling technique is extended to apply for fuzzy samples by implementing the uncertainty model fuzzy randomness into the basic procedure according to Section 2. Due to the generalized character of this uncertainty model, the capability of processing real-valued samples is hereby preserved as a special case.

4.2. EXTENSION OF CRITERIA $C_1^{[.]}$ AND $C_2^{[.]}$

As the starting point for the extension of the model-free sampling technique to processing imprecise data, these data are described with the aid of Fuzzy Set Theory (Zimmermann, 1992). Each imprecise observation, which represents a sample element as a realization of an underlying fuzzy random vector $\tilde{\underline{\mathbf{X}}}$, is modeled as a normalized fuzzy set or fuzzy vector $\tilde{\underline{s}} = \tilde{\underline{x}} \in \mathbf{F}(\mathbb{R}^n)$ with the membership function $\mu_{\tilde{\underline{s}}}(\underline{s})$, see Figure 2. The real-valued samples \underline{S}_0 and \underline{S}_1 from Eqs. (1) and (2) therewith become fuzzy samples,

$$\tilde{\underline{S}}_0 = \{ \tilde{s}_{0,i}, i = 1, \dots, n_0 \}, \tag{31}$$

$$\tilde{\underline{S}}_1 = \{ \tilde{s}_{1,k}, k = 1, \dots, n_1 \}, \tag{32}$$

with the underlying fuzzy random vector $\tilde{\underline{\mathbf{X}}}_0$ for $\tilde{\underline{S}}_0$.

The processing of the fuzzy samples $\tilde{\underline{S}}_0$ and $\tilde{\underline{S}}_1^{[.]}$ within the procedure according to Steps 1 through 4 in Section 2 requires the extension of the dissimilarity measure $G^{[.]}$ and thus of the criteria $C_1^{[.]}$ and $C_2^{[.]}$ to apply for fuzzy vectors $\tilde{s}_{0,i}$ and $\tilde{s}_{1,k}^{[.]}$ as elements of $\tilde{\underline{S}}_0$ and $\tilde{\underline{S}}_1^{[.]}$. As a basis a suitable replacement for the Euclidean

distance $d(\underline{s}_{0,i}, \underline{s}_{1,k}^{[.]})$ in Eq. (18) must be introduced as a distance measure between fuzzy vectors $\tilde{s}_{0,i}$ and $\tilde{s}_{1,k}^{[.]}$. For this purpose the fuzzy vectors $\tilde{s}_{0,i}$ and $\tilde{s}_{1,k}^{[.]}$ are represented with the aid of α -discretization, see Figure 2. For a sufficiently high number of α -levels the fuzzy vectors $\tilde{s}_{0,i}$ and $\tilde{s}_{1,k}^{[.]}$ are completely described by the sets of their α -level sets $\underline{s}_{0,i,\alpha}$ and $\underline{s}_{1,k,\alpha}^{[.]}$, respectively. Specifically, for real numbers $\alpha \in (0, 1]$,

$$\underline{s}_{0,i,\alpha} = \left\{ \underline{s} \in \mathbb{R}^n \mid \mu_{\underline{s}_{0,i}}(\underline{s}) \geq \alpha \right\}, \tag{33}$$

$$\underline{s}_{1,k,\alpha}^{[.]} = \left\{ \underline{s} \in \mathbb{R}^n \mid \mu_{\underline{s}_{1,k}^{[.]}}(\underline{s}) \geq \alpha \right\}, \tag{34}$$

and

$$\tilde{s}_{0,i} = \left\{ \left(\underline{s}_{0,i,\alpha}, \mu_{\underline{s}_{0,i}}(\underline{s}_{0,i,\alpha}) \right) \mid \mu_{\underline{s}_{0,i}}(\underline{s}_{0,i,\alpha}) = \alpha \forall \alpha \in (0, 1] \right\}, \tag{35}$$

$$\tilde{s}_{1,k}^{[.]} = \left\{ \left(\underline{s}_{1,k,\alpha}^{[.]}, \mu_{\underline{s}_{1,k}^{[.]}}(\underline{s}_{1,k,\alpha}^{[.]}) \right) \mid \mu_{\underline{s}_{1,k}^{[.]}}(\underline{s}_{1,k,\alpha}^{[.]}) = \alpha \forall \alpha \in (0, 1] \right\}. \tag{36}$$

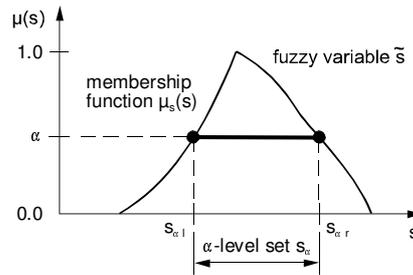


Figure 2. α -discretization of a fuzzy variable

On this basis, the distance $d_F(\tilde{s}_{0,i}, \tilde{s}_{1,k}^{[.]})$ between the fuzzy vectors $\tilde{s}_{0,i}$ and $\tilde{s}_{1,k}^{[.]}$ may be defined by recombining the distances $d_H(\underline{s}_{0,i,\alpha}, \underline{s}_{1,k,\alpha}^{[.]})$ between the associated α -level sets $\underline{s}_{0,i,\alpha}$ and $\underline{s}_{1,k,\alpha}^{[.]}$ (for the same α -level). Specifically, the metric

$$d_F(\tilde{s}_{0,i}, \tilde{s}_{1,k}^{[.]}) = \int_{\alpha=+0}^{\alpha=1} d_H(\underline{s}_{0,i,\alpha}, \underline{s}_{1,k,\alpha}^{[.]}) d\alpha \tag{37}$$

is applied, see (Körner, 1997), which makes use of the Hausdorff metric

$$d_H(\underline{s}_{0,i,\alpha}, \underline{s}_{1,k,\alpha}^{[.]}) = \max \left[d_{H1,i,k}^{[.]}, d_{H2,i,k}^{[.]} \right],$$

$$d_{H1,i,k}^{[.]} \left(\underline{s}_{0,i,\alpha}, \underline{s}_{1,k,\alpha}^{[.]} \right) = \sup_{\underline{s}_0 \in \underline{s}_{0,i,\alpha}} \inf_{\underline{s}_1 \in \underline{s}_{1,k,\alpha}^{[.]}} [d(\underline{s}_0, \underline{s}_1)], \quad (38)$$

$$d_{H2,i,k}^{[.]} \left(\underline{s}_{0,i,\alpha}, \underline{s}_{1,k,\alpha}^{[.]} \right) = \sup_{\underline{s}_1 \in \underline{s}_{1,k,\alpha}^{[.]}} \inf_{\underline{s}_0 \in \underline{s}_{0,i,\alpha}} [d(\underline{s}_0, \underline{s}_1)],$$

between the associated α -level sets $\underline{s}_{0,i,\alpha}$ and $\underline{s}_{1,k,\alpha}^{[.]}$ with $d(\underline{s}_0, \underline{s}_1)$ being the Euclidean distance between crisp elements \underline{s}_0 and \underline{s}_1 from $\underline{s}_{0,i,\alpha}$ and $\underline{s}_{1,k,\alpha}^{[.]}$, respectively, see Figure 3. The outcome of Eq. (38) and hence the distance $d_F(\tilde{s}_{0,i}, \tilde{s}_{1,k}^{[.]})$ from Eq. (37) are crisp values, which can be directly applied in Eqs. (19) and (26) to eventually compute criteria $C_1^{[.]}$ and $C_2^{[.]}$.

The application of criteria $C_1^{[.]}$ and $C_2^{[.]}$ to evaluate the dissimilarity of fuzzy-valued samples enables a consideration of the order in the element configuration and the distance between the respective sample elements. Dissimilarities in the fuzziness of the elements $\tilde{s}_{0,i}$ and $\tilde{s}_{1,k}^{[.]}$, however, are taken into account only to a partial degree. In addition to the criteria $C_1^{[.]}$ and $C_2^{[.]}$, the fuzziness of the realizations provides a basis for a third dissimilarity criterion.

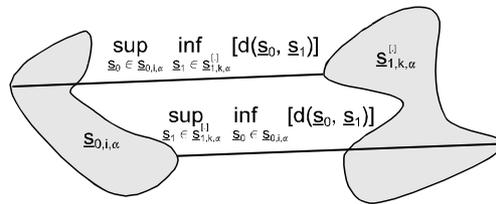


Figure 3. Hausdorff metric applied to α -level sets

4.3. FUZZINESS CRITERION

The fuzziness criterion evaluates the matching in the fuzziness of the respective fuzzy sample elements $\tilde{s}_{0,i}$ and $\tilde{s}_{1,k}^{[.]}$. Fuzzy sample elements that are assigned to each other according to the assignment rule Eq (19) are supposed to exhibit the same fuzziness. For this purpose, the fuzziness of the sample elements is computed with an analog to Shannon’s entropy applied to the membership functions $\mu(\underline{s}_{0,i}) = \mu_{\underline{s}_{0,i}}(\underline{s})$ and $\mu(\underline{s}_{1,k}^{[.]}) = \mu_{\underline{s}_{1,k}^{[.]}}(\underline{s})$ of $\tilde{s}_{0,i}$ and $\tilde{s}_{1,k}^{[.]}$, respectively. For the fuzzy vector \tilde{s} , this uncertainty measure is defined as

$$H_u = -k \cdot \int_{\underline{s}} g(\mu(\underline{s})) d\underline{s},$$

$$g(\mu(\underline{s})) = \mu(\underline{s}) \cdot \ln(\mu(\underline{s})) + (1 - \mu(\underline{s})) \cdot \ln(1 - \mu(\underline{s})). \quad (39)$$

And the fuzziness criterion is

$$C_3^{[.]} = \sum_{k=1}^{n_1} \left(H_u(\tilde{s}_{0,i}(\tilde{s}_{1,k}^{[.]})) - H_u(\tilde{s}_{1,k}^{[.]}) \right)^2 \Rightarrow \text{MIN}. \quad (40)$$

For a "perfect matching", the fuzziness criterion $C_3^{[.]}$ becomes zero.

4.4. PROCEDURE FEATURES FOR IMPRECISE DATA

The generation and the iterative modification of a fuzzy sample $\tilde{S}_1^{[.]}$ require not only determining the position of the sample elements $\tilde{s}_{1,k}^{[.]}$ but also specifying their membership functions $\mu(s_{1,k}^{[.]})$. New fuzzy realizations $\tilde{s}_{1,k}^{[.]}$ are generated in the following three steps.

1. The mean values

$$s_{1,k,\mu=1}^{[.]} = s_{1,k}^{[.]} \in \tilde{s}_{1,k}^{[.]} \mid \mu(s_{1,k}^{[.]}) = 1 \tag{41}$$

of the fuzzy sample elements $\tilde{s}_{1,k}^{[.]}$ (different from the definition of a statistical mean value, see (Zimmermann, 1992)) are specified analogous to the crisp sample elements $s_{1,k}^{[.]}$ of the $S_1^{[.]}$ in Eqs. (5), (12), and (29). That is, the initialization is realized with

$$(\underline{X}_1 \sim U(\underline{D})) \rightarrow \{s_{1,k,\mu=1}^{[0]}, k = 1, \dots, n_1\}, \tag{42}$$

and during the iteration

$$(\underline{X}_1 \sim U(\underline{D})) \rightarrow \{s_{1,k,\mu=1}^{[r]+}, k = k_1, \dots, k_{m_1}\} \tag{43}$$

or, alternatively,

$$\left(\underline{X}_1 \sim F_{\underline{S}_{1,\mu=1}^{[r]}}^{(\bar{e})}(\underline{x})\right) \rightarrow \{s_{1,k,\mu=1}^{[r]+}, k = k_1, \dots, k_{m_1}\} \tag{44}$$

are applied, in which $F_{\underline{S}_{1,\mu=1}^{[r]}}^{(\bar{e})}(\underline{x})$ represents the smoothed empirical distribution of the mean values $s_{1,k,\mu=1}^{[r]}$ in the fuzzy sample $\tilde{S}_1^{[r]}$ in iteration step r .

2. The fuzziness $H_u(\tilde{s}_{1,k}^{[.]})$ is determined by means of a logarithmic normal distribution $F^{(\log)}(H_u)$ estimated from the fuzziness $H_u(\tilde{s}_{0,i})$ of the fuzzy sample elements $\tilde{s}_{0,i}$ in the observed fuzzy sample \tilde{S}_0 ,

$$\{H_u(\tilde{s}_{0,i}), i = 1, \dots, n_0\} \rightarrow F^{(\log)}(H_u), \tag{45}$$

$$(X_H \sim F^{(\log)}(H_u)) \rightarrow \{H_u(\tilde{s}_{1,k}^{[0]}), k = 1, \dots, n_1\}, \tag{46}$$

$$(X_H \sim F^{(\log)}(H_u)) \rightarrow \{H_u(\tilde{s}_{1,k}^{[r]+}), k = k_1, \dots, k_{m_1}\}. \tag{47}$$

3. The shape of the membership function $\mu(s_{1,k}^{[.]})$ is also randomly specified according to the "empirical distribution" of the shape of $\mu(s_{0,i})$ in \tilde{S}_0 . This is realized with the aid of a parametric representation of

the fuzzy sample elements $\tilde{s}_{0,i}$ and $\tilde{s}_{1,k}^{[.]}$ in a zero-mean (in fuzzy terminology, see (Zimmermann, 1992)) form normalized to a unit maximum spread t_r . In general terms,

$$\tilde{t} = \tilde{s} - s_{\mu=1}, \tag{48}$$

$$\tilde{t}^{(n)} = \frac{1}{t_r} \cdot \tilde{t}, \quad t_r = \max_{\underline{t}|\mu(\underline{t})>0} \|\underline{t}\|, \tag{49}$$

$$\mu(\underline{t}^{(n)}) = \mu(p_1, \dots, p_{n_p}). \tag{50}$$

The "empirical distribution" of the shape is then represented by the smoothed joint empirical distribution of the parameters p_1, \dots, p_{n_p} ,

$$\tilde{t}_{0,i} = \tilde{s}_{0,i} - s_{0,i,\mu=1}, \quad i = 1, \dots, n_0, \tag{51}$$

$$\tilde{t}_{0,i}^{(n)} = \frac{1}{t_{r,0,i}} \cdot \tilde{t}_{0,i}, \quad t_{r,0,i} = \max_{\underline{t}|\mu(\underline{t}_{0,i})>0} \|\underline{t}\|, \quad i = 1, \dots, n_0, \tag{52}$$

$$\left\{ \mu(\underline{t}_{0,i}^{(n)}), \quad i = 1, \dots, n_0 \right\} \rightarrow \left(F^{(\bar{e})}(p_1, \dots, p_{n_p}) \right). \tag{53}$$

The random shape of new elements $\tilde{s}_{1,k}^{[.]}$ is determined according to

$$\left(\underline{X}_p \sim F^{(\bar{e})}(p_1, \dots, p_{n_p}) \right) \rightarrow \left\{ \mu(\underline{t}_{1,k}^{(n)[0]}), \quad k = 1, \dots, n_1 \right\} \tag{54}$$

and

$$\left(\underline{X}_p \sim F^{(\bar{e})}(p_1, \dots, p_{n_p}) \right) \rightarrow \left\{ \mu(\underline{t}_{1,k}^{(n)[r]^+}), \quad k = k_1, \dots, k_{m_1} \right\}, \tag{55}$$

respectively. The obtained new normalized fuzzy elements $\underline{t}_{1,k}^{(n)[0]}$ and $\underline{t}_{1,k}^{(n)[r]^+}$ are backtransformed inverse to Eqs. (48) and (49),

$$\tilde{s}_{1,k}^{[0]} = \underline{t}_{1,k}^{(n)[0]} \cdot t'_{r,k} + s_{1,k,\mu=1}^{[0]}, \quad k = 1, \dots, n_1, \tag{56}$$

$$\tilde{s}_{1,k}^{[r]^+} = \underline{t}_{1,k}^{(n)[r]^+} \cdot t'_{r,k} + s_{1,k,\mu=1}^{[r]^+}, \quad k = k_1, \dots, k_{m_1}, \tag{57}$$

with $s_{1,k,\mu=1}^{[0]}$ and $s_{1,k,\mu=1}^{[r]^+}$ from Eqs. (42) and (43) or (44). The spread factors $t'_{r,k}$ are obtained implicitly by the fuzziness $H_u(\tilde{s}_{1,k}^{[0]})$ and $H_u(\tilde{s}_{1,k}^{[r]^+})$, respectively, specified according to Eqs. (46) and (47).

The consideration of fuzzy samples requires incorporating the criterion $C_3^{[.]}$ into the iterative procedure. Tests have shown that it is effective to perform the iteration for fuzzy samples in two parts. In the first part, only the criteria $C_1^{[.]}$ and $C_2^{[.]}$ are satisfied. Subsequently, the obtained element assignment and the mean value positions are frozen. In the second part, criterion $C_3^{[.]}$ is applied in a separate fuzziness iteration. That is, in the second iteration part, only the $H_u(\tilde{s}_{1,k}^{[.]})$ and the shape of the membership functions of the fuzzy sample elements $\tilde{s}_{1,k}^{[.]}$ are adjusted. The iteration termination criterion is also applied separately in both iteration parts.

5. Application to Structural Engineering Problems

5.1. GENERAL APPLICATION SCHEMES

The general concept of model-free sampling provides a beneficial basis for a coupling with structural engineering computations in various ways. Generally, the following application schemes can be pursued – separately or combined.

5.1.1. Processing of input data

Model-free sampling can be applied to crisp or imprecise input data of structural computations, in particular, if the available data are rare, and a probabilistic model is not known to a sufficient degree of confidence. The sampling result $\underline{S}_1^{\text{input}}$ or $\tilde{S}_1^{\text{input}}$ then reflects the stochastic or fuzzy stochastic properties of the input data in the form of a numerically generated data set of crisp or imprecise input vectors for further processing in structural computations. This is equivalent to the result of a Monte Carlo simulation based on a known probabilistic model for the input quantities. The coupling to structural computations can be realized as follows.

- The sampling result $\underline{S}_1^{\text{input}}$ or $\tilde{S}_1^{\text{input}}$ can be directly used for a subsequent stochastic or fuzzy stochastic structural analysis to compute stochastic or fuzzy stochastic structural responses. It represents the input sample, which contains n_1 input vectors $\underline{s}_{1,k}^{\text{input}}$ or $\tilde{s}_{1,k}^{\text{input}}$ for a subsequent n_1 -fold structural analysis. In the case of samples comprising imprecise data, the generalized uncertainty processing algorithms presented in (Möller and Beer, 2004) may be applied. As results fuzzy probabilistic structural responses are obtained, which are characterized by imprecise probability distributions with fuzzy parameters such as a fuzzy mean and a fuzzy variation.
- As a postprocessing attached to a stochastic or fuzzy stochastic structural analysis based on the generated sample $\underline{S}_1^{\text{input}}$ or $\tilde{S}_1^{\text{input}}$, a safety assessment can be carried out by evaluating limit states in the result space, which may be advantageous if limit state surfaces cannot be specified in the input space for some reason. This is simply realized by counting those sample elements in the stochastic or fuzzy stochastic structural responses, which lead to failure according to the defined limit states. The result is obtained as a failure probability or a fuzzy failure probability. As a difference to traditional methods, this procedure can involve imprecise sample elements, which is explained in Section 5.2.
- In contrast to the latter, a safety assessment can also be performed by evaluating limit state surfaces in the input space. This is particularly useful if the underlying structural analysis to produce structural responses is computationally expensive, and the limit state surfaces can be described in the space of the structural input parameters, for example, within the framework of a response surface method. Again, counting of the elements in the failure domain – with an evaluation of imprecise data according to Section 5.2 – yields a failure probability or a fuzzy failure probability.

5.1.2. Processing of result data

The model-free sampling technique can also be used for processing result data from structural computations. This can be instrumental if the set of structural response data is limited and cannot be described with a

probabilistic model on a satisfying confidence level. Also, in the case of imprecise measurements of structural responses this method may be helpful. The stochastic or fuzzy stochastic properties of the structural responses are then described with the aid of the numerical sampling result $\underline{S}_1^{\text{result}}$ or $\tilde{S}_1^{\text{result}}$ of a sufficiently large size n_1 – equivalent to the outcome from a Monte Carlo simulation with a sufficiently high number n_1 of structural analyses. As a prerequisite for obtaining reliable results in this manner, the underlying sample of structural responses must comprise essential information about the properties of the computational model. That is, physical, mechanical, or chemical phenomena that are effective in the underlying structural analysis must be already reflected in the sample of structural responses for being reproduced in a subsequent model-free sampling and thus in the final result $\underline{S}_1^{\text{result}}$ or $\tilde{S}_1^{\text{result}}$. In correspondence with Section 5.1.1, the following two approaches can be pursued for a coupling to structural computations.

- The uncertain structural responses from stochastic or fuzzy stochastic structural analysis can be introduced into model-free sampling to obtain a sufficiently large sample size n_1 for describing crisp or imprecise probability distributions of the responses empirically instead of performing a weak and ambiguous distribution estimation.
- For a safety assessment, limit states in the result space can be directly evaluated with the aid of the sampling result $\underline{S}_1^{\text{result}}$ or $\tilde{S}_1^{\text{result}}$. For the technique of counting fuzzy sample elements in the failure domain, see Section 5.2.

5.2. RELIABILITY ASSESSMENT FOR IMPRECISE DATA

The application of model-free sampling may be particularly useful in reliability assessment as the available data do usually not cover failure domains. It is thus of great interest in this application field to reproduce the tails of the underlying probability distributions to obtain reliable estimations of failure probabilities.

Structural reliability assessment based on model-free sampling is realized as a straightforward extension to traditional methods. The sampling result \underline{S}_1 or \tilde{S}_1 is directly evaluated with regards to the limit states either in the input space or in the result space, see Section 5.1. That is, the structural reliability is determined by counting the sample elements that lead to failure. For dealing with imprecise data, however, this counting needs to be extended in an appropriate manner, see (Möller and Beer, 2004). Due to their fuzziness, some fuzzy sample elements $\tilde{s}_{1,k}$ lie only partly in the failure domain \underline{S}_f , or, in the case of an underlying computational model that involves model uncertainty as fuzziness (Möller et al., 2003), in the fuzzy failure domain \tilde{S}_f . This leads to a fuzzy failure probability \tilde{P}_f . For computing \tilde{P}_f α -discretization is applied again, see Section 4.2. Specifically,

$$\begin{aligned}\tilde{P}_f &= \{(P_{f,\alpha}, \mu(P_{f,\alpha}))\}, \\ P_{f,\alpha} &= [P_{f,\alpha l}, P_{f,\alpha r}], \\ \mu(P_{f,\alpha}) &= \alpha \quad \forall \alpha \in (0, 1].\end{aligned}\tag{58}$$

The interval bounds $P_{f,\alpha l}$ and $P_{f,\alpha r}$ (see Figure 2 for general illustration) are calculated with the aid of indicator functions and particular conditions for evaluating fuzzy realizations, see (Möller and Beer, 2004). Specifically,

$$P_{f,\alpha l} = \frac{1}{n_1} \cdot \sum_{k=1}^{n_1} I_{\alpha l}(\tilde{s}_{1,k}),$$

$$I_{\alpha 1}(\tilde{s}_{1,k}) = \begin{cases} 1 & \text{if } \underline{s}_{1,k,\alpha} \subseteq \underline{S}_{f,\alpha} \\ 0 & \text{otherwise} \end{cases}, \quad (59)$$

and

$$\begin{aligned} P_{f,\alpha r} &= \frac{1}{n_1} \cdot \sum_{k=1}^{n_1} I_{\alpha r}(\tilde{s}_{1,k}), \\ I_{\alpha r}(\tilde{s}_{1,k}) &= \begin{cases} 1 & \text{if } \underline{s}_{1,k,\alpha} \cap \underline{S}_{f,\alpha} \neq \emptyset \\ 0 & \text{otherwise} \end{cases}. \end{aligned} \quad (60)$$

6. Examples

6.1. REAL-VALUED DATA

6.1.1. Sampling

A one-dimensional real-valued sample S_0 of size $n_0 = 200$ is taken as the basis, see Figure 4. This is numerically generated from a compound distribution consisting of two extreme value distributions of Ex-Max type I. The extreme values of the sample S_0 are $\min_{S_0} = 5.1$ and $\max_{S_0} = 21.55$.

An initial estimate $S_1^{[0]}$ is numerically generated according to Eq. (5) by uniformly distributing $n_1 = 10,000$ sample elements $s_{1,k}$ over the interval $D = [0, 25]$, see Figure 4. Then, the iteration Eq. (15) to improve the generalized estimation $S_1^{[0]}$ is started. The number m_1 of modified elements is randomly selected from the interval $[a, b] = [5, 30]$, see Eq. (28), and frequently changed during the iteration. For generating the new elements $s_{1,k}^{[+]}$ the bootstrap-like method of Eq. (29) is applied. After about $r = 4,000$ iteration steps the average success rate starts decreasing distinctly and attains the termination limit in iteration step $r = 4,710$, see Figure 4.

Clearly, there is no visible difference between the empirical distribution functions of the samples S_0 and $S_1 = S_1^{[4,710]}$. Homogeneity tests (Kolmogorov-Smirnov, Mann-Whitney, and chi-squared) yield rejection probabilities of $P < 0.012$ for the H_0 -hypothesis that both samples originate from the same population. The tails of the generated sample S_1 run beyond the extreme values of S_0 with $\min_{S_1} = 3.26$ and $\max_{S_1} = 24.01$. A total of 39 elements $s_{1,k}$ are smaller than $\min_{S_0} = 5.1$ and, and 48 elements $s_{1,k}$ are bigger than $\max_{S_0} = 21.55$. The proportions of S_1 therewith correspond to an extreme value distribution with a thicker tail on the right side than on the left side. Fisher's exact probability test yields a probability of $P = 0.386$ with which the H_0 -hypothesis is not rejected. Further, the sampling result S_1 shows no clumping of the generated sample elements $s_{1,k}$ around the original sample elements $s_{0,i}$, which has been verified by investigating the distribution of the elements $s_{1,k}$ within the "gaps" between the original elements $s_{0,i}$.

Results generated via traditionally estimated probability distributions did not attain the quality level of the present sample S_1 . Kernel-based estimation methods led to samples showing test results comparable to the present approach. Their tails, however, did not run significantly beyond $\min_{S_0} = 5.1$ and $\max_{S_0} = 21.55$ and were pre-determined in their form depending on the (subjectively) selected kernels. The same applies to even generalized bootstrap methods. In contrast to that, the tails of S_1 from model-free sampling are not influenced by subjectivity and obtained in a form with orientation to the structure of the underlying sample S_0 , which possesses significant importance in reliability assessment.

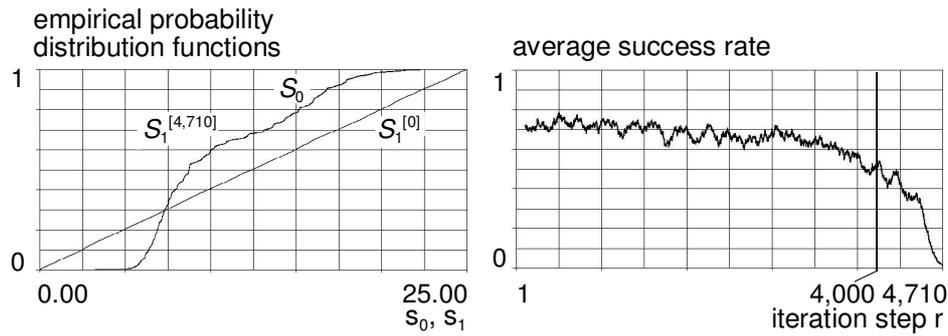


Figure 4. Empirical distribution functions of S_0 , $S_1^{[0]}$, and $S_1^{[4,710]}$; average success rate (last 100 steps) during iteration

6.1.2. Reliability assessment

The reliability assessment is pursued by directly evaluating the sampling result S_1 with respect to a given limit state surface. Since the related procedures are well-known, these are not highlighted in the example. Herein, it is focused on the dependency of the assessment result on the quality of the sampling result.

For demonstration, the observed sample S_0 is interpreted as a possible record of a live load s resulting from road traffic and acting on a structural member of a road bridge. The sampling result S_1 then represents a statistical loading prognosis for future traffic. For defining a limit state surface, the serviceability requirement $s = 22$ is defined.

The empirical failure probability obtained from sample S_0 is $P_f = 0$, whereas the sampling result S_1 yields $P_f = 3.4 \cdot 10^{-3}$. A compound probability distribution estimated from S_0 without additional prior knowledge leads to $P_f = 1.7 \cdot 10^{-3}$. According to the underlying extreme value distribution $P_f = 8.9 \cdot 10^{-3}$ is obtained. These results indicate a good agreement between the prognoses from traditional approaches and from model-free sampling.

6.2. IMPRECISE DATA

6.2.1. Sampling

As a starting point the sample S_0 from Section 6.1 is "fuzzified" to represent an uncertain measurement series, for example, of a live load, see Section 6.1.2. That is, the underlying bimodal distribution from Section 6.1.1 is retained for the mean values $s_{0,i,\mu=1}$. The resulting fuzzy sample \tilde{S}_0 consists of $n_0 = 200$ fuzzy triangular numbers with fluctuating fuzziness $H_u(\tilde{s}_{0,i})$ over the sample elements $\tilde{s}_{0,i}$; for relevant concepts and terminology see (Bandemer and Näther, 1992) and (Möller and Beer, 2004). An initial estimate $\tilde{S}_1^{[0]}$ of size $n_1 = 10,000$ is generated in compliance with Section 4.4 starting from uniformly distributed mean values $s_{1,k,\mu=1}^{[0]}$ and restricting the fuzzy sample elements completely to $\tilde{s}_{1,k}^{[0]} \subseteq [0, 25]$, see Figure 5. Again, the iteration is carried out with a randomly selected number $m_1 \in [5, 30]$ of modified elements. First, the dissimilarity measure $G^{[1]}(C_1^{[1]}, C_2^{[1]})$, see Eq. (27) with the extension from Section 4.2, is minimized in 5,990 iteration steps. The empirical fuzzy probability distributions of \tilde{S}_0 and $\tilde{S}_1^{[5,990]}$ agree very well. However, there is almost no correspondence between the fuzziness $H_u(\tilde{s}_{0,i})$ and $H_u(\tilde{s}_{1,k}^{[5,990]})$ of the respective

fuzzy sample elements, see Figure 5. The subsequent fuzziness iteration (minimization of criterion $C_3^{[1]}$ up to iteration step $r = 16,150$) almost does not affect the empirical distribution, but improves considerably the fuzziness agreement, see Figure 5.

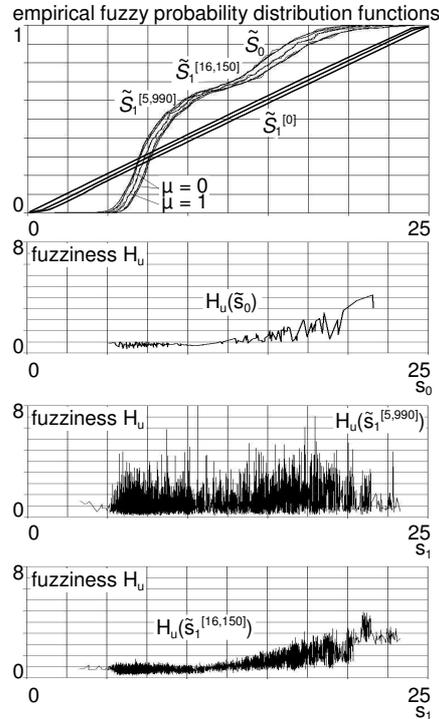


Figure 5. Empirical fuzzy probability distribution functions of \tilde{S}_0 , $\tilde{S}_1^{[0]}$, $\tilde{S}_1^{[5,990]}$, and $\tilde{S}_1^{[16,150]}$; fuzziness H_u of the associated fuzzy sample elements

6.2.2. Reliability assessment

The serviceability requirement $s = 22$ specified in Section 6.1.2 is evaluated with the fuzzy samples \tilde{S}_0 and $\tilde{S}_1 = \tilde{S}_1^{[16,150]}$. The fuzzy failure probability \tilde{P}_f is computed according to Eqs. (58), (59), and (60) with eleven α -levels, see Figure 6. Whereas sample \tilde{S}_0 yields an almost useless result with an overestimated fuzziness, sample \tilde{S}_1 leads to a more meaningful result. The probability values covered by \tilde{P}_f from \tilde{S}_1 again comprise a reasonable range with respect to the results from traditional estimations and from the underlying distribution for the mean values $s_{0,i,\mu=1}$ presented in Sect. 6.1.2.

7. Conclusions

The presented model-free sampling technique may be useful if the data bank comprises, solely, a small sample with uncertain or imprecise elements. It operates free of a probability model, is capable of considering

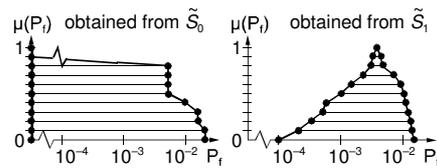


Figure 6. Empirical fuzzy failure probability obtained from \tilde{S}_0 and from \tilde{S}_1

randomness and non-stochastic uncertainty simultaneously, and can be attached to engineering computations that involve uncertainty in various schemes.

Beyond the demonstrated capabilities in the one-dimensional case, promising experiences have already been made in processing vector valued data including nonlinear stochastic dependencies. A further consideration of multidimensional problems for fuzzy valued data is pursued.

Acknowledgements

The author gratefully acknowledges the support of the German Research Foundation (DFG) and of the Alexander von Humboldt-Foundation (AvH).

References

- Bandemer, H. and W. Näther: 1992, *Fuzzy Data Analysis*. Dordrecht: Kluwer Academic Publishers.
- Berleant, D. and J. Zhang: 2004, 'Representation and problem solving with Distribution Envelope Determination (DEnv)'. *Reliability Engineering & System Safety* **85**(1-3), 153-168.
- Bernardo, J. and A. Smith: 1994, *Bayesian Theory*. Chichester New York Brisbane Toronto Singapore: Wiley.
- Colubi, A., C. Fernández-García, and M. A. Gil: 2002, 'Simulation of random fuzzy variables: an empirical approach to statistical/probabilistic studies with fuzzy experimental data'. *IEEE Transactions on Fuzzy Systems* **10**, 384-390.
- Deodatis, G., L. Graham-Brady, and R. Micaletti: 2003, 'A hierarchy of upper bounds on the response of stochastic systems with large variation of their properties: random field case'. *Probabilistic Engineering Mechanics* **18**(4), 365-375.
- Du, S., B. R. Ellingwood, and J. V. Cox: 2005, 'Initialization strategies in simulation-based SFE eigenvalue analysis'. *Computer-aided Civil and Infrastructure Engineering* **20**(5), 304-315.
- Fellin, W., H. Lessmann, M. Oberguggenberger, and R. Vieider (eds.): 2005, 'Analyzing Uncertainty in Civil Engineering'. Berlin Heidelberg New York: Springer.
- Fetz, T. and M. Oberguggenberger: 2004, 'Propagation of uncertainty through multivariate functions in the framework of sets of probability measures'. *Reliability Engineering & System Safety* **85**(1-3), 73-87.
- Field Jr., R. V. and M. Grigoriu: 2004, 'On the accuracy of the polynomial chaos approximation'. *Probabilistic Engineering Mechanics* **19**(1-2), 65-80.
- Ghanem, R. and P. Spanos: 1991, *Stochastic Finite Elements: A Spectral Approach*. New York Berlin Heidelberg: Springer. Revised Edition 2003, Dover Publications, INC., Mineola, New York.
- Gutierrez, E. and J. M. Zaldivar: 2000, 'The application of Karhunen-Loeve, or principle component analysis method, to study the non-linear seismic response of structures'. *Earthquake Engineering & Structural Dynamics* **29**(9), 1261-1286.
- Hall, J. W. and J. Lawry: 2004, 'Generation, combination and extension of random set approximations to coherent lower and upper probabilities'. *Reliability Engineering & System Safety* **85**(1-3), 89-101.

- Helton, J. C. and W. L. Oberkampf: 2004, 'Special Issue on Alternative Representations of Epistemic Uncertainty'. *Reliability Engineering & System Safety* **85**(1-3), 1-369.
- Kwakernaak, H.: 1978, 'Fuzzy random variables – I. Definitions and Theorems'. *Information Sciences* **15**, 1-19.
- Körner, R.: 1997, 'Linear Models with Random Fuzzy Variables'. Ph.D. thesis, Bergakademie Freiberg, Fakultät für Mathematik und Informatik.
- Mood, A., F. Graybill, and D. Boes: 1974, *Introduction to the Theory of Statistics*. New York: McGraw-Hill.
- Möller, B. and M. Beer: 2004, *Fuzzy Randomness - Uncertainty in Civil Engineering and Computational Mechanics*. Berlin: Springer.
- Möller, B., W. Graf, and M. Beer: 2000, 'Fuzzy structural analysis using alpha-level optimization'. *Computational Mechanics* **26**, 547-565.
- Möller, B., W. Graf, and M. Beer: 2003, 'Safety assessment of structures in view of fuzzy randomness'. *Computers and Structures* **81**, 1567-1582.
- Papadopoulos, V., G. Deodatis, and M. Papadrakakis: 2005, 'Flexibility-based upper bounds on the response variability of simple beams'. *Computer Methods in Applied Mechanics and Engineering* **194**(1), 1385-1404.
- Phoon, K., H. Huang, and S. Quek: 2005, 'Simulation of strongly non-Gaussian processes using Karhunen-Loeve expansion'. *Probabilistic Engineering Mechanics* **20**(2), 188-198.
- Rackwitz, R.: 2001, 'Reliability analysis – a review and some perspectives'. *Structural Safety* **23**(4), 365-395.
- Red-Horse, J. R. and A. S. Benjamin: 2004, 'A probabilistic approach to uncertainty quantification with limited information'. *Reliability Engineering & System Safety* **85**(1-3), 183-190.
- Schenk, C. A., H. J. Pradlwarter, and G. I. Schuëller: 2005, 'Non-stationary response of large, non-linear finite element systems under stochastic loading'. *Computers and Structures* **83**(14), 1086-1102.
- Schenk, C. A. and G. I. Schuëller: 2005, *Uncertainty Assessment of Large Finite Element Systems*. Berlin Heidelberg: Springer.
- Schuëller, G.: 2001a, 'Computational Stochastic Mechanics - Recent Advances'. *Computers and Structures* **79**(22-25), 2225-2234.
- Schuëller, G.: 2001b, 'On Computational Procedures for Processing Uncertainties in Structural Mechanics'. In: Z. Waszczyszyn and J. Pamin (eds.): *2nd Europ. Conf. on Computational Mechanics, Cracow*. pp. 1-24. CD-ROM, Doc 608.
- Schuëller, G. and P. Spanos (eds.): 2001, 'Proc. Int. Conf. on Monte Carlo Simulation MCS 2000, Monte Carlo, Monaco, 2000'. Swets & Zeitlinger BV, Lisse, The Netherlands.
- Schuëller, G. I., H. J. Pradlwarter, and C. A. Schenk: 2003, 'A computational procedure to estimate the stochastic dynamic response of large non-linear FE-models'. *Computer Methods in Applied Mechanics and Engineering* **192**(7-8), 777-801.
- Sickert, J.-U., M. Beer, W. Graf, and B. Möller: 2003, 'Fuzzy probabilistic structural analysis considering fuzzy random functions'. In: A. D. Kiureghian, S. Madanat, and J. Pestana (eds.): *9th Int. Conference on Applications of Statistics and Probability in Civil Engineering*. Berkeley, Rotterdam, pp. 379-386, Millpress.
- Spanos, P. D. and R. Ghanem: 1989, 'Stochastic finite element expansion for random media'. *ASCE Journal of the Engineering Mechanics* **115**(5), 1035-1053.
- Tonon, F., A. Bernardini, and A. Mammino: 2000, 'Reliability analysis of rock mass response by means of Random Set Theory'. *Reliability Engineering & System Safety* **70**(3), 263-282.
- Viertl, R.: 1996, *Statistical Methods for Non-Precise Data*. Boca Raton New York London Tokyo: CRC Press.
- Zimmermann, H.-J.: 1992, *Fuzzy set theory and its applications*. Boston London: Kluwer Academic Publishers.

Non-Probabilistic Design Optimization with Insufficient Data using Possibility and Evidence Theories

Zissimos P. Mourelatos¹, Jun Zhou²
Mechanical Engineering Department
Oakland University, Rochester, MI 48309
mourelat@oakland.edu and jzhou@oakland.edu

Abstract: Early in the engineering design cycle, it is difficult to quantify product reliability or compliance to performance targets due to insufficient data or information for modeling the uncertainties. Design decisions are therefore, based on fuzzy information that is vague, imprecise qualitative, linguistic or incomplete. The uncertain information is usually available as intervals with lower and upper limits. In this paper, the possibility and evidence theories are used to account for uncertainty in design with incomplete information. The formal theories to handle uncertainty are first introduced using the theoretical fundamentals of fuzzy measures. The first part of the paper highlights how the possibility theory can be used in design. A computationally efficient and accurate hybrid (global-local) optimization approach is used to calculate the confidence level of “fuzzy” response combining the advantages of the commonly used vertex and discretization methods. A possibility-based design optimization method is proposed where all design constraints are expressed possibilistically. It is shown that the method gives a conservative solution compared with all conventional reliability-based designs obtained with different probability distributions. Also, a general possibility-based design optimization method is presented which handles a combination of random and possibilistic design variables. The second part of the paper describes a design optimization method using evidence theory. The method can be used when limited and often conflicting, information is available from “expert” opinions. A computationally efficient design optimization formulation is presented, which can handle a mixture of epistemic and random uncertainties. It quickly identifies the vicinity of the optimal point and the active constraints by moving a hyper-ellipse in the original design space, using a reliability-based design optimization (RBDO) algorithm. Subsequently, a derivative-free optimizer calculates the evidence-based optimum, starting from the close-by RBDO optimum, considering only the identified active constraints. The computational cost is kept low by first moving to the vicinity of the optimum quickly and subsequently using local surrogate models of the active constraints only. Two numerical examples demonstrate the application of possibility and evidence theories in design and highlight the trade-offs among reliability-based, possibility-based and evidence-based designs.

¹ Associate Professor.

² PH.D. candidate.

© 2006 by authors. Printed in USA

1. INTRODUCTION

Engineering design under uncertainty has recently gained a lot of attention. Uncertainties are usually modeled using probability theory. In Reliability-Based Design Optimization (RBDO), variations are represented by standard deviations which are typically assumed constant, and a mean performance is optimized subject to probabilistic constraints (Tu, Choi and Park, 1999; Liang, Mourelatos and Tu, 2004; Wu, Shin, Sues and Cesare, 2001; Lee, Yang and Ruy, 2002; Youn, Choi and Park, 2001). In general, probability theory is very effective when sufficient data is available to quantify uncertainty using probability distributions. However, when sufficient data is not available or there is lack of information due to ignorance, the classical probability methodology may not be appropriate. For example, during the early stages of product development, quantification of the product's reliability or compliance to performance targets is practically very difficult due to insufficient data for modeling the uncertainties. A similar problem exists when the reliability of a complex system is assessed in the presence of incomplete information on the variability of certain design variables, parameters, operating conditions, boundary conditions etc.

Uncertainties can be classified in two general types; aleatory (stochastic or random) and epistemic (subjective) (Oberkampf, Helton, and Sentz, 2001; Sentz and Ferson, 2002; Klir and Yuan, 1995; Klir and Filger, 1988; Yager, Fedrizzi and Kacprzyk, 1994). Aleatory or irreducible uncertainty is related to inherent variability and is efficiently modeled using probability theory. However, when data is scarce or there is lack of information, the probability theory is not useful because the needed probability distributions cannot be accurately constructed. In this case, epistemic uncertainty, which describes subjectivity, ignorance or lack of information, can be used. Epistemic uncertainty is also called reducible because it can be reduced with increased state of knowledge or collection of more data.

Formal theories to handle uncertainty have been proposed in the literature including evidence theory (or Dempster – Shafer theory) (Klir and Filger, 1988; Yager, Fedrizzi and Kacprzyk, 1994), possibility theory [Dubois and Prade, 1988) and interval analysis (Moore, 1966). Two large classes of fuzzy measures, called belief and plausibility measures, respectively, characterize the mathematical theory of evidence. They are mutually dual in the sense that one of them can be uniquely determined from the other. Evidence theory uses plausibility and belief (upper and lower bounds of probability) to measure the likelihood of events. When the plausibility and belief measures are equal, the general evidence theory reduces to the classical probability theory. Therefore, the classical probability theory is a special case of evidence theory.

Possibility theory handles epistemic uncertainty if there is no conflicting evidence among experts (Klir and Filger, 1988). It uses a special subclass of dual plausibility and belief measures, called possibility and necessity measures, respectively. In possibility theory, a fuzzy set approach is common, where membership functions characterize the input uncertainty (Zadeh, 1965). Even if a probability distribution is not available due to limited information, lower and upper bounds (intervals) on uncertain design variables are usually known. In this case, interval analysis (Moore 1966; Muhanna and Mullen, 2001; Mullen and Muhanna, 1999) and fuzzy set theory (Zadeh, 1965) have been extensively used to characterize and propagate input uncertainty in order to calculate the interval of the uncertain output. An efficient method for reliability estimation with a combination of random and interval variables is presented in (Penmetsa and Grandhi, 2002). However, it is not implemented in a design optimization framework. A few design optimization

studies have been also reported, where some or all of the uncertain design variables are in interval form (Du and Sudjianto, 2003; Rao and Cao, 2002; Gu and batill, 1998).

Optimization with input ranges has also been studied under the term anti-optimization (Elishakoff, Haftka and Fang, 1994; Lombardi and Haftka, 1998). Anti-optimization is used to describe the task of finding the “worst-case” scenario for a given problem. It solves a two-level (usually nested) optimization problem. The outer level performs the design optimization while the inner level performs the anti-optimization. The latter seeks the worst condition under the interval uncertainty (Lombardi and Haftka, 1998). A decoupled approach is suggested in (Lombardi and Haftka, 1998) where the design optimization alternates with the anti-optimization rather than nesting the two. It was mentioned that this method takes longer to converge and may not even converge at all if there is strong coupling between the interval design variables and the rest of the design variables. A “worst-case” scenario approach using interval variables has also been considered in multidisciplinary systems design [(Gu and Batill, 1998; Du and Chen, 2000).

Very recently, possibility-based design algorithms have been proposed (Mourelatos and Zhou, 2005; Choi, Du and Youn, 2004) where a mean performance is optimized subject to possibilistic constraints. It was shown that more conservative results are obtained compared with the probability-based RBDO. A comprehensive comparison of probability and possibility theories is given in (Nikolaidis, Chen, Cudney, Haftka and Rosca, 2004) for design under uncertainty.

Evidence theory is more general than probability and possibility theories, even though the methodologies of uncertainty propagation are completely different (Oberkampf and Helton, 2002; Bae, Grandhi and Canfield, 2004). It can be used in design under uncertainty if limited, and even conflicting, information is provided from experts. Furthermore, the basic axioms of evidence theory allow to combine aleatory (random) and epistemic uncertainty in a straightforward way without any assumptions (Bae, Grandhi and Canfield, 2004). Evidence theory however, has been barely explored in engineering design. One of the reasons may be its high computational cost due mainly to the discontinuous nature of uncertainty quantification. Evidence-based methods have been only recently used to propagate epistemic uncertainty (Bae, Grandhi and Canfield, 2004; Bae, Grandhi and Canfield, 2004) in large-scale engineering systems. Although a computationally efficient method is proposed in (Bae, Grandhi and Canfield, 2004; Bae, Grandhi and Canfield, 2004), the design issue is not addressed. We are aware of only one study which propagates epistemic uncertainty using evidence theory and also performs a design optimization (Agarwal, Renaud, Preston and Padmanabhan, 2004). The optimum design is calculated for multidisciplinary systems under uncertainty using a trust region sequential approximate optimization method with surrogate models representing the uncertain measures as continuous functions.

In this paper, the possibility and evidence theories are used to account for uncertainty in design with incomplete information. The formal theories to handle uncertainty are first introduced using the theoretical fundamentals of fuzzy measures. The first part of the paper highlights how the possibility theory can be used in design. A computationally efficient and accurate hybrid (global-local) optimization approach is presented for calculating the confidence level of “fuzzy” response, combining the advantages of the commonly used vertex and discretization methods. A possibility-based design optimization method is subsequently described where all design constraints are expressed possibilistically. The method gives a conservative solution compared with all conventional reliability-based designs obtained with different probability distributions.

Also, a general possibility-based design optimization method is presented which handles a combination of random and possibilistic design variables.

In the second part of the paper, a computationally efficient design optimization method is proposed based on evidence theory, which can handle a mixture of epistemic and random uncertainties. The method can be used when limited and often conflicting, information is available from “expert” opinions. The algorithm quickly identifies the *vicinity* of the optimal point and the active constraints by moving a hyper-ellipse in the original design space, using an RBDO algorithm. Subsequently, a derivative-free optimizer calculates the evidence-based optimum, starting from the close-by RBDO optimum, considering only the identified active constraints. The computational cost is kept low by first moving to the vicinity of the optimum quickly and subsequently using local surrogate models of the active constraints only.

The paper is organized as follows. Section 2 gives an introduction to fuzzy measures. Section 3 describes the fundamentals of possibility theory based on fuzzy measures as well as some numerical methods for propagating non-probabilistic uncertainty, which are essential in possibility-based design. A detailed formulation of Possibility-Based Design Optimization (PBDO) where design constraints are satisfied possibilistically, is presented in section 4. Section 5 presents a detailed formulation of an Evidence-Based Design Optimization (EBDO) method and its implementation. All principles are demonstrated with examples in section 6. Results are compared among deterministic optimization, RBDO, PBDO and EBDO. Finally, a summary and conclusions are given in section 7.

2. FUZZY MEASURES

The evidence and possibility theories are based on the mathematical foundation of fuzzy measures which provide the foundation of fuzzy set theory. Before we introduce the basics of fuzzy measures, it is helpful to review the used notation on set representation. A universe X represents the entire collection of elements having the same characteristics. The individual elements in the universe X are denoted by x , which are usually called singletons. A set A is a collection of some elements of X . All possible sets of X constitute a special set called the power set $\wp(X)$.

A fuzzy measure is defined by a function $g: \wp(X) \rightarrow [0,1]$ which assigns to each crisp [Ross 1995] subset of X a number in the unit interval $[0,1]$. The assigned number in the unit interval for a subset $A \in \wp(X)$, denoted by $g(A)$, represents the degree of available evidence or belief that a given element of X belongs to the subset A .

In order to qualify as a fuzzy measure, the function g must obey the following three axioms:

Axiom 1 (boundary conditions): $g(\emptyset)=0$ and $g(X)=1$.

Axiom 2 (monotonicity): For every $A, B \in \wp(X)$, if $A \subseteq B$, then $g(A) \leq g(B)$.

Axiom 3 (continuity): For every sequence $(A_i \in \wp(X), i=1,2,\dots)$ of subsets of $\wp(X)$, if either $A_1 \subseteq A_2 \subseteq \dots$ or $A_1 \supseteq A_2 \supseteq \dots$ (i.e., the sequence is monotonic), then

$$\lim_{i \rightarrow \infty} g(A_i) = g(\lim_{i \rightarrow \infty} A_i).$$

A belief measure is a function $Bel: \wp(X) \rightarrow [0,1]$ which satisfies the three axioms of fuzzy measures and the following additional axiom [9]:

$$Bel(A_1 \cup A_2) \geq Bel(A_1) + Bel(A_2) - Bel(A_1 \cap A_2). \tag{1}$$

The axiom (1) can be expanded for more than two sets. For $A \in \wp(X)$, $Bel(A)$ is interpreted as the degree of belief, based on available evidence, that a given element of X belongs to the set A .

A plausibility measure is a function

$$Pl: \wp(X) \Rightarrow [0,1] \tag{2}$$

which satisfies the three axioms of fuzzy measures and the following additional axiom [(Klir and Filger, 1988)

$$Pl(A_1 \cap A_2) \leq Pl(A_1) + Pl(A_2) - Pl(A_1 \cup A_2) \tag{3}$$

Every belief measure and its dual plausibility measure can be expressed with respect to the non-negative function

$$m: \wp(X) \Rightarrow [0,1] \tag{4}$$

such that $m(\emptyset) = 0$ and

$$\sum_{A \in \wp(X)} m(A) = 1 \tag{5}$$

The function m is called Basic Probability Assignment (BPA) due to the resemblance of Eq. (5) with a similar equation for probability distributions. The basic probability assignment $m(A)$ is interpreted either as the degree of evidence supporting the claim that a specific element of X belongs to the set A or as the degree to which we believe that such a claim is warranted. At this point, it should be noted that the BPA is very different from the probability distribution function. Basic probability assignments are defined on sets of the power set (i.e., on $A \in \wp(X)$), whereas the probability distribution functions are defined on the singletons x of the power set (i.e., on $x \in \wp(X)$). Every set $A \in \wp(X)$ for which $m(A) > 0$ is called a focal element of m . Focal elements are subsets of X on which the available evidence focuses; i.e. available evidence exists.

Given a BPA m , a belief measure and a plausibility measure are uniquely determined by

$$Bel(A) = \sum_{B \subseteq A} m(B) \tag{6}$$

and

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B). \tag{7}$$

which are applicable for all $A \in \wp(X)$.

In Eq. (6), $Bel(A)$ represents the total evidence or belief that the element belongs to A as well as to various subsets of A . The $Pl(A)$ in Eq. (7) represents not only the total evidence or belief that the element in question belongs to set A or to any of its subsets but also the additional evidence or belief associated with sets that overlap with A . Therefore,

$$Pl(A) \geq Bel(A). \tag{8}$$

Probability theory is a subset of evidence theory. When the additional axiom of belief measures (see Eq. (1)) is replaced with the stronger axiom

$$Bel(A \cup B) = Bel(A) + Bel(B) \text{ where } A \cap B = \emptyset, \tag{9}$$

we obtain a special type of belief measures which are the classical probability measures. In this case, the right hand sides of Eq. (6) and (7) become equal and therefore,

$$Bel(A) = Pl(A) = \sum_{x \in A} m(x) = \sum_{x \in A} p(x) \quad (10)$$

for all $A \in \wp(X)$, where $p(x)$ is the classical probability distribution function (PDF). Note that the BPA $m(x)$ is equal to $p(x)$. Therefore with evidence theory, we can simultaneously handle a mixture of input parameters. Some of the inputs can be described probabilistically (random uncertainty) and some can be described through expert opinions (epistemic uncertainty with incomplete data). In the second case, the range of each input parameter will be discretized using a finite number of intervals. The BPA value for each interval must be equal to the PDF area within the interval.

It should be noted that according to evidence theory, the $Bel(A)$ and $Pl(A)$ bracket the true probability $P(A)$ [9], i.e.

$$Bel(A) \leq P(A) \leq Pl(A). \quad (11)$$

Evidence obtained from independent sources or experts must be combined. If the BPA's m_1 and m_2 express evidence from two experts, the combined evidence m can be calculated by the following Dempster's rule of combining (Senz and Ferson, 2002)

$$m(A) = \frac{\sum_{B \cap C = A} m_1(B)m_2(C)}{1 - K} \quad \text{for } A \neq \emptyset \quad (12)$$

where

$$K = \sum_{B \cap C = \emptyset} m_1(B)m_2(C) \quad (13)$$

represents the *conflict* between the two independent experts. Dempster's rule filters out any conflict, or contradiction among the provided evidence, by normalizing with the complementary degree of conflict. It is usually appropriate for relatively small amounts of conflict where there is some consistency or sufficient agreement among the opinions of the experts. Yager (Yager, Fedrizzi and Kacprzyk, 2004) has proposed an alternative rule of combination where all degrees of contradiction are attributed to total ignorance. Other rules of combining can be found in (Senz and Ferson, 2002).

The possibility theory is a subcase of the general evidence theory. It can be used to characterize epistemic uncertainty, when incomplete data is available. It applies only when there is *no conflict* in the provided body of evidence. In such a case, the focal elements of the body of evidence are nested and the associated belief and plausibility measures are called consonant. In contrary, when there is conflicting evidence, the belief and plausibility measures are dissonant. A family of subsets of the universal set is nested if they can be ordered in such a way that each is contained within the next. Thus, $A_1 \subset A_2 \subset \dots \subset A_n$ are nested sets. Consonant belief and plausibility measures are usually known as necessity measures n and possibility measures π , respectively. Therefore, if there is no conflicting information, $n(A) = Bel(A)$ and $\pi(A) = Pl(A)$. The necessity and possibility are dual measures, related by

$$n(A) = 1 - \pi(\bar{A}). \quad (14)$$

where \bar{A} is the complement of set A .

3. FUNDAMENTALS OF POSSIBILITY THEORY

This section highlights the fundamentals of possibility theory as it was originally introduced in the context of fuzzy set theory (Zadeh, 1978). In the fuzzy set approach to possibility theory, focal elements are represented by a -cuts of the associated fuzzy set. Focal elements are subsets that are assigned nonzero degrees of evidence. The possibility theory can be used to bracket the true probability based on the fuzzy set approach at various confidence intervals (a -cuts). The advantage of this is that as the design progresses and the confidence level on the input parameter bounds increases, the design need not be reevaluated to obtain the new bounds of the response. Similarly to the probability measures, which are represented by the probability distribution functions, the possibility measures can be represented by the possibility distribution function $r : X \Rightarrow [0,1]$ such that

$$\pi(A) = \max_{x \in A} r(x). \tag{15}$$

It can be shown that possibility measures are formally equivalent to fuzzy sets. In this equivalence, the membership grade of an element x corresponds to the plausibility of the singleton consisting of that x . Therefore, a consonant belief structure is equivalent to a fuzzy set of X .

A fuzzy set is an imprecisely defined set that does not have a crisp boundary. It provides instead, a gradual transition from “belonging” to “not belonging” to the set. A function can be defined such that the values assigned to the elements of the set are within a specified range and indicate the membership grade of these elements in the set. Larger values denote higher degrees of set membership. Such a function is called a membership function and the set defined by it a fuzzy set.

The membership function μ_A by which a fuzzy set A is usually defined has the form $\mu_A : X \rightarrow [0, 1]$ where $[0, 1]$ denotes the interval of real numbers from 0 to 1, inclusive. Given a fuzzy subset A of X with membership function μ_A , Zadeh (Zadeh, 1978) defines a possibility distribution function r associated with A as numerically equal to μ_A , i.e. $r(x) = \mu_A(x)$ for all $x \in X$. Then, he defines the corresponding possibility measure π as

$$\pi(A) = \sup_{x \in A} r(x) \text{ for each } A \in \wp(X). \tag{16}$$

Eq. (16) is equivalent to Eq. (15) when X is finite. In the fuzzy set approach to possibility theory, focal elements are represented by a -cuts of the associated fuzzy set. For the remaining of this discussion, we will follow the fuzzy set approach to possibility theory.

Eq. (11) states that the true probability is bracketed by the belief and plausibility measures. If we know the possibility distribution function $\mu_Y(y)$ of the response Y , then the true probability $P(Y)$ can be also bracketed as

$$n(Y) \leq P(Y) \leq \pi(Y) \tag{17}$$

where the necessity $n(Y)$ and possibility $\pi(Y)$ measures are calculated from Eqs (14) and (16), respectively. The “extension principle” (Klir and Filger, 1988; Yager, Fedrizzi and Kacprzyk, 1994; Ross, 1995) is used to calculate the possibility distribution function $\mu_Y(y)$ of the response.

3.1. FUZZIFICATION PROCESS AND EXTENSION PRINCIPLE

The process of quantifying a fuzzy variable is known as fuzzification. If any of the input variables is imprecise, it is considered fuzzy and must be therefore, fuzzified in order for the uncertainty to be propagated using fuzzy calculus. The fuzzification is done by constructing a possibility distribution, or membership function, for each imprecise (fuzzy) variable. Details can be found in (Ross, 1995). The membership function takes values in the $[0,1]$ interval. Here, we use convex normal possibility distributions to characterize the fuzzy variables. An example of a convex normal triangular possibility distribution is shown in Fig. 1. The point for which the possibility is equal to one is called normal point. The possibility distribution is convex since it is strictly decreasing to the left and right of the normal point. At each confidence level, or α -cut, a set X_α is defined as

$$X_\alpha = \{x : x_L^\alpha \leq x \leq x_R^\alpha, \alpha \in [0,1]\}, \quad (18a)$$

which is a monotonically decreasing function of α ; i.e.

$$\alpha_1 > \alpha_2 \Rightarrow X_{\alpha_1} \subset X_{\alpha_2} \text{ for every } \alpha_1, \alpha_2 \in [0,1]. \quad (18b)$$

Due to the convexity of the possibility distribution function, all sets generated at different α -cuts are nested according to Eq. (18b). Therefore, the convexity and normality of the possibility distribution function satisfies the basic requirement of nested sets (no conflicting evidence) in possibility theory.

After the fuzzification of the imprecise input variables, the “extension principle” is used to propagate the epistemic uncertainty through the transfer function in order to calculate the fuzzy response. The “extension principle” calculates the possibility distribution of the fuzzy response from the possibility distributions of the fuzzy input variables. In particular, given the transfer function $y = f(\underline{x})$, where the output y depends on the N independent fuzzy inputs $\underline{x} = \{x_1, \dots, x_N\}$, the “extension principle” states that the possibility distribution μ_Y of the output is given by

$$\mu_Y[y = f(\underline{x})] = \sup_y \left\{ \min_j \left[\mu_{X_j}(f(\underline{x}_j)) \right] \right\} \quad (19)$$

where “sup” denotes the supremum operator that gives the least upper bound. The above equation can be interpreted as follows. For a crisp value of the output y , there may exist more than one combination of crisp values of input variables \underline{x} resulting in the same output.

The possibility of each combination is given by the smallest possibility value for all fuzzy input variables. The possibility that $y = f(\underline{x})$, is given by the maximum possibility for all these combinations. Note that in probability theory, the probability of an outcome is equal to the product of the probabilities of the constituent events. In fuzzy set theory however, the possibility of an outcome is equal to the minimum possibility of the constituent events.

If the outcome can be reached in many ways, then the outcome probability, in probability theory, is given by the sum of the probabilities of all the ways. In fuzzy theory, the possibility of the outcome is given by the maximum possibility of all the possibilities (Ross, 1995).

The direct (“brute force”) solution of Eq. (19) is practically intractable except for simple cases involving one or two fuzzy variables. The computational effort increases exponentially with increasing number of fuzzy input variables. For this reason, approximate numerical techniques have been proposed, among which the discretization method (Akpan, Rushton and Koko, 2002) and the vertex method (Penmetsa and Grandhi, 2002) are the most popular ones.

In the discretization method, the domain of each fuzzy variable $i; 1 \leq i \leq N$ is discretized with M_i discrete values at each a -cut. Then the output y is evaluated at all possible combinations $\prod_{i=1}^N M_i$ for each a -cut. Subsequently, Eq. (19) is used to calculate the possibility distribution of the output. The range of the output is defined by the minimum and maximum response from all combinations. Although this method can be very accurate, the associated computational cost is practically prohibitive.

In the vertex method, all the binary combinations of only the extreme values of the fuzzy variables at an a -cut are fed into the deterministic transfer function. The bounds of the fuzzy response are then obtained at the a -cut, by choosing the maximum and minimum responses. The procedure is repeated for all a -cuts of interest. The method has the potential to give accurate

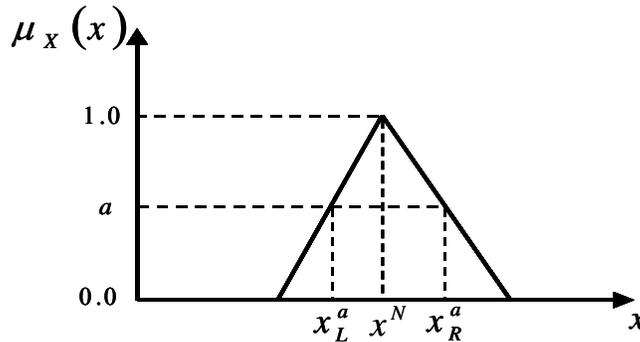


Figure 1. Triangular possibility distribution for a fuzzy variable.

bounds of the response based on the bounded input. However, when the transfer function exhibits minima or maxima within the domain defined by the extreme values of the input variables, the vertex method is inaccurate. This is due to the fact that the function is evaluated only at the binary combinations of the input variable bounds. For a problem with N fuzzy input variables, the required number of function evaluations for the vertex method is $A * 2^N$, where A is the number of a -cuts.

In general, the vertex method is computationally more efficient compared with the discretization method. However, the required computational effort grows exponentially with the number of input fuzzy variables (Ross, 1995). For this reason, most of the reported applications are restricted to very few fuzzy variables (Mullen and Muhanna, 1999; Chen and Rao, 1997; Rao and Sawyer, 1995).

A hybrid (global-local) optimization method has been reported in (Mourelatos and Zhou, 2005], which ensures computational efficiency without loss of accuracy. An optimization algorithm is used to calculate the minimum and maximum values of the response at each a -cut. Because the global minimum and maximum values of the response are needed, a derivative free, global optimizer called DIRECT (DIvisions of RECTangles), is used in order to avoid being trapped at a local optimum and obtain therefore, an inaccurate solution. DIRECT is a modification of the standard Lipschitzian approach that eliminates the need to specify a Lipschitz constant (Jones, Perttunen and Stuckman, 1993). Although global optimizers may get close to the global optimum quickly, it takes them longer to achieve a high degree of accuracy because they usually have a slow rate of convergence. This suggests that the best performance can be obtained by combining DIRECT with a gradient-based local optimizer in a hybrid approach. In this work, DIRECT is first used, followed by a local optimizer based on Sequential Quadratic Programming (SQP). DIRECT provides a converged global optimum based on “loose” convergence criteria. Subsequently, the DIRECT solution is used as starting point for SQP, which identifies the optimum accurately and efficiently.

3.2. A MATHEMATICAL EXAMPLE

The following two-variable, six-hump camel function (Wang, 2003) is used

$$y(x_1, x_2) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4, \quad x_{1,2} \in [-2, 2].$$

to illustrate the accuracy and efficiency of the hybrid optimization method of the previous section and compare it with the vertex and discretization methods. For demonstration reasons, the

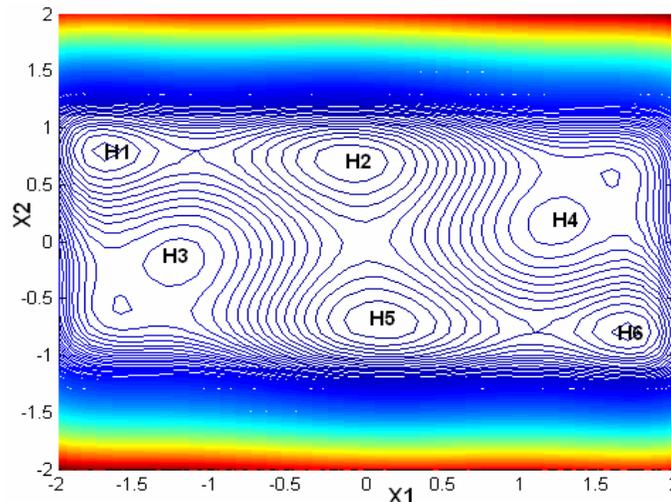


Figure 2. Contour plot for mathematical example.

following simple triangular membership functions are used for the two input variables x_1 and x_2

$$\mu_{x_i}(x_i) = \begin{cases} -\frac{x_i}{2} + 1, & 0 \leq x_i \leq 2 \\ \frac{x_i}{2} + 1, & -2 \leq x_i \leq 0 \end{cases} \quad i = 1, 2.$$

Fig. 2 shows the contour plot of the six hump camel function. The H's indicate all extreme points. Points H2 and H5 with coordinates (0.0898, -0.7127) and (-0.0898, 0.7127) respectively, are two global optima with an equal function value of $y_{\min} = -1.0316$. The calculated membership

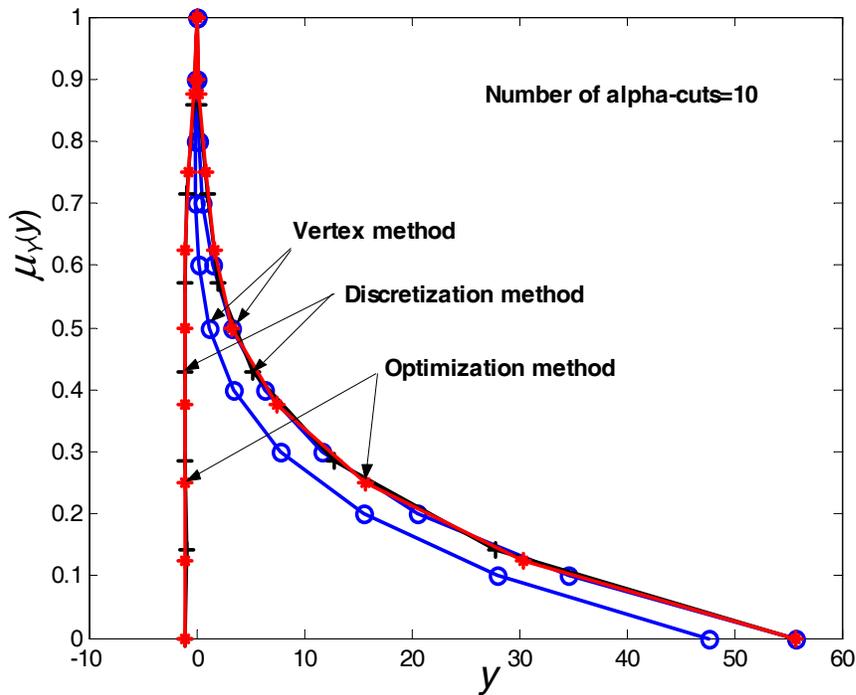


Figure 3. Response membership function for mathematical example.

functions of the response y using the vertex, discretization and hybrid optimization methods are plotted in Fig. 3. Ten α -cuts are used for all three methods. For the discretization method, the range of each input fuzzy variable, at each α -cut, is equally split in 15 divisions. It is known that if the input membership functions are convex normal, the response membership function must also be convex normal. The justification is that when the input uncertainty increases (low α -cut values), the uncertainty of the response must remain the same or increase. As shown in Fig. 3, the response membership function obtained by the vertex method is not convex and therefore, it is wrong.

As explained in section 3.1, the discretization method evaluates the function not only at the upper and lower limits of the input variables at each alpha cut but also between the bounds. Thus,

it can capture the extreme points that might be present in between the upper and lower bounds. At each alpha cut, all combinations are obtained and the minimum and maximum response values are calculated in order to get the response membership function. It is clear that the response becomes more accurate as the number of divisions per alpha cut increases. As shown in Fig. 3, the response membership function calculated with the discretization method, is convex and

Table 1. Accuracy and efficiency comparison of vertex, discretization and hybrid optimization methods

	Vertex	Discretization	Hybrid Optimization
Lower Bound	47.73	-1.01	-1.03
Upper Bound	55.73	55.73	55.73
No of F.E.	4	256	140

normal. The uncertainty decreases as the level of confidence increases (increasing a -cut values). The major disadvantage of this method is that as the number of design variables increases and the number of divisions per a -cut also increases, the method becomes computationally very expensive. In this example, the number of a -cuts is 10 and the number of divisions per a -cut is 15. Therefore, the number of function evaluations is $10 \cdot (15+1)^2 = 2560$. The response membership function of the six hump camel function is also calculated using the proposed hybrid optimization method. The result is identical with that obtained with the discretization method (see Fig. 3).

Table 1 summarizes the lower and upper bound values of the response at the zero a -cut, as calculated by the vertex, discretization and hybrid optimization methods. The vertex method is very efficient but inaccurate. The hybrid optimization method however, has the same accuracy with the “brute force” discretization method but it is much more efficient.

4. POSSIBILITY-BASED DESIGN OPTIMIZATION

In deterministic design optimization, an objective function is minimized subject to satisfying a set of constraints. In Reliability-Based Design Optimization (RBDO), where all design variables are characterized probabilistically, an objective function is usually minimized subject to the probability of satisfying each constraint being greater than a specified high reliability level.

In this section, a methodology is presented on how to use possibility theory in design. We will show that the possibility-based design is conservative compared with all RBDO designs obtained with different probability distributions. In RBDO, some optimality is usually sacrificed in order to accommodate the random uncertainty. The possibility-based design sacrifices a little more optimality in order to accommodate the lack of probability distribution information. It therefore, encompasses all RBDO designs obtained with different distributions.

According to Eq. (11), the probability $P(A)$ of event A is bracketed by the belief $Bel(A)$ and plausibility $Pl(A)$; i.e. $Bel(A) \leq P(A) \leq Pl(A)$. We have also mentioned that for consonant (no conflicting evidence) belief structures, the plausibility measures are equal to the possibility measures, resulting in $\eta(A) \leq P(A) \leq \pi(A)$, where η and π are the necessity and possibility measures, respectively (see Eq. 17). This means that the possibility $\pi(A)$ provides an upper bound to the probability $P(A)$. From the design point of view, we can thus conclude (Klir and

Filger, 1988; Ross, 1995; Zadeh, 1978) that *what is possible may not be probable, and what is impossible is also improbable.*

Note that for an impossible event A , the possibility $\pi(A)$ is zero. If we therefore, make sure that the possibility of violating a constraint is zero, then the probability of violating the same constraint will be also zero. If feasibility of a constraint g is expressed with the positive null form $g \geq 0$, the constraint is always satisfied if

$$\pi(g \leq 0) = 0. \tag{20}$$

The possibility π in Eq. (20) is calculated using Eq. (16). Fig. 4 shows the membership function $\mu_G(g)$ of constraint g . The possibility of set $A = \{g : g_{\min} \leq g \leq g_{\min}^\alpha, \alpha \in [0,1]\}$ is $\pi(A) = \alpha$ and the possibility of set $B = \{g : g_{\max}^\alpha \leq g \leq g_{\max}, \alpha \in [0,1]\}$ is $\pi(B) = 1$. Similarly, the possibility of constraint violation is $\pi(g \leq 0) = \alpha_1$. Eq. (20) can be relaxed as

$$\pi(g \leq 0) \leq \alpha \tag{21}$$

where the a-cut level is small; i.e. $\alpha \ll 1$. Based on Fig. 4, the relation (21) is satisfied if

$$g_{\min}^\alpha \geq 0 \tag{22}$$

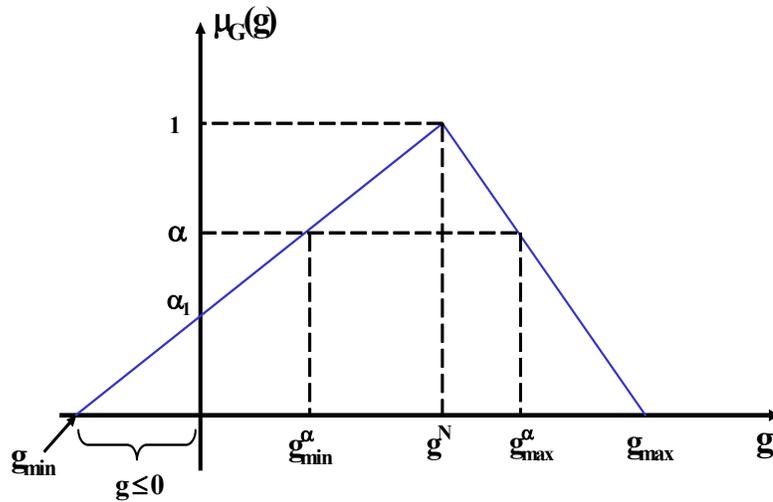


Figure 4. Used notation in possibility-based design optimization

where g_{\min}^α is the global minimum of g at the a -cut. Eq. (22) is analogous to the R-percentile formulation [1] of a probabilistic constraint in RBDO. The possibilistic constraint of Eqs (21) or (22) becomes active if $g_{\max}^\alpha = 0$.

Based on this discussion, a possibility-based design optimization (PBDO) problem can be formulated as

$$\min_{\mathbf{d}, \mathbf{x}^N} f(\mathbf{d}, \mathbf{x}^N, \mathbf{p}^N)$$

$$\begin{aligned} \text{s.t. } & \pi(g_i(\mathbf{d}, \mathbf{X}, \mathbf{P}) \leq 0) \leq \alpha, \quad i = 1, \dots, n \\ & \mathbf{d}_L \leq \mathbf{d} \leq \mathbf{d}_U, \quad \mathbf{x}_L \leq \mathbf{x}^N \leq \mathbf{x}_U \end{aligned} \quad (23)$$

where $\mathbf{d} \in R^k$ is the vector of deterministic design variables, $\mathbf{X} \in R^m$ is the vector of possibilistic design variables, $\mathbf{P} \in R^q$ is the vector of possibilistic design parameters and \mathbf{x}^N and \mathbf{p}^N are the normal point vectors for the possibilistic design variables and parameters, respectively. According to the used notation, a bold letter indicates a vector, an upper case letter indicates a possibilistic variable or parameter and a lower case letter indicates a deterministic variable or a realization of a possibilistic variable or parameter. Feasibility of the i^{th} deterministic constraint is expressed with the positive null form $g_i \geq 0$.

The possibilistic design variables are represented with convex normal possibility distributions (membership functions). Note that they may not be necessarily triangular. The superscript N denotes the normal point of each distribution where the membership function value is equal to one. Subscripts L and U denote lower and upper bounds, respectively. In PBDO, we will assume that the membership functions of the possibilistic design variables have a constant shape and that their normal points are design variables moving within predetermined bounds. This is analogous to RBDO where the PDF of each random design variable stays constant while its mean value is a design variable.

Based on Eq. (22), the PBDO formulation (23) is equivalent to

$$\begin{aligned} & \min_{\mathbf{d}, \mathbf{x}^N} f(\mathbf{d}, \mathbf{x}^N, \mathbf{p}^N) \\ \text{s.t. } & g_{i_{\min}}^\alpha \geq 0 \quad i = 1, \dots, n \\ & \mathbf{d}_L \leq \mathbf{d} \leq \mathbf{d}_U, \quad \mathbf{x}_L \leq \mathbf{x}^N \leq \mathbf{x}_U. \end{aligned} \quad (24)$$

The PBDO formulation (23) or (24) is a double-loop optimization problem where an optimization is performed (inner loop) when the design optimization (outer loop) calls for a possibilistic constraint evaluation. It should be noted that the PBDO optimum at $\alpha=1$ coincides with the deterministic optimum.

4.1. PBDO WITH A COMBINATION OF RANDOM AND POSSIBILISTIC VARIABLES

Reliability-based design optimization (RBDO) provides optimum designs in the presence of only random (or aleatory) uncertainty (Tu, Choi and Park, 1999; Liang, Mourelatos and Tu, 2004; Wu, Shin, Sues and Cesare, 2001). A typical RBDO problem is formulated as (Liang, Mourelatos and Tu, 2004]

$$\begin{aligned} & \min_{\mathbf{d}, \boldsymbol{\mu}_Y} f(\mathbf{d}, \boldsymbol{\mu}_Y, \boldsymbol{\mu}_Z) \\ \text{s.t. } & P(g_i(\mathbf{d}, \mathbf{Y}, \mathbf{Z}) \geq 0) \geq R_i = 1 - p_{f_i}, \quad i = 1, \dots, n \\ & \mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U, \quad \boldsymbol{\mu}_Y^L \leq \boldsymbol{\mu}_Y \leq \boldsymbol{\mu}_Y^U \end{aligned} \quad (25)$$

where $\mathbf{Y} \in R^\ell$ is the vector of random design variables and $\mathbf{Z} \in R^r$ is the vector of random design parameters.

For a variety of practical applications however, there may not be enough information to characterize all design variables and parameters probabilistically. A subset of them can be

therefore, characterized possibilistically using membership functions. A possibility-based design optimization problem with a combination of random and possibilistic (or fuzzy) variables can be formulated as

$$\begin{aligned}
 & \min_{\mathbf{d}, \mathbf{x}^N, \boldsymbol{\mu}_Y} f(\mathbf{d}, \boldsymbol{\mu}_Y, \boldsymbol{\mu}_Z, \mathbf{x}^N, \mathbf{p}^N) & (26) \\
 \text{s.t.} \quad & g_{i_{\min}}^\alpha \geq 0, \quad i = 1, \dots, n \\
 & \mathbf{d}_L \leq \mathbf{d} \leq \mathbf{d}_U, \quad \boldsymbol{\mu}_Y^L \leq \boldsymbol{\mu}_Y \leq \boldsymbol{\mu}_Y^U \\
 & \mathbf{x}_L \leq \mathbf{x}^N \leq \mathbf{x}_U \\
 \text{with} \quad & g_{i_{\min}}^\alpha = \min_{\mathbf{X}} (\beta_i - \beta_{t_i}) \geq 0, \quad i = 1, \dots, n, \\
 & \mathbf{x}_L^\alpha \leq \mathbf{x} \leq \mathbf{x}_U^\alpha, \quad \mathbf{p}_L^\alpha \leq \mathbf{p} \leq \mathbf{p}_U^\alpha \\
 \text{and} \quad & \beta = \min_{\mathbf{U}} \|\mathbf{U}\| \\
 \text{s.t.} \quad & G(\mathbf{U}) = 0
 \end{aligned}$$

where β_i is the target reliability index. Note that \mathbf{x}_L^α and \mathbf{x}_U^α are the lower and upper limits of \mathbf{X} at an α -cut.

Problem (26) represents a triple-loop optimization sequence. The design optimization of the outer loop calls a series of possibilistic constraints in the middle loop. Each possibilistic constraint is in general, a global optimization problem. Finally, each possibilistic constraint is a function of the corresponding reliability index β which represents the third loop of the optimization sequence. For computational purposes, two out of the three nested loops can be easily combined.

5. EVIDENCE-BASED DESIGN OPTIMIZATION (EBDO)

In this section, a methodology is presented on how to use evidence theory in design. We will show that the evidence theory-based design is more conservative compared with all RBDO designs obtained with different probability distributions and less conservative compared with the PBDO design.

If feasibility of a constraint g is expressed with the non-negative null form $g \geq 0$, we have shown that $Bel(g \geq 0) \leq P(g \geq 0) \leq Pl(g \geq 0)$ where $P(g \geq 0)$ is the probability of constraint satisfaction. Therefore,

$$P(g < 0) \leq p_f \text{ is satisfied if } Pl(g < 0) \leq p_f \tag{27}$$

where p_f is the probability of failure which is usually a small prescribed value. The above statement is equivalent to

$$P(g \geq 0) \geq R \text{ is satisfied if } Bel(g \geq 0) \geq R \tag{28}$$

where $R = 1 - p_f$ is the corresponding reliability level.

Hence, an evidence theory-based design optimization (EBDO) problem can be therefore, formulated as

$$\begin{aligned} & \min_{\mathbf{d}, \mathbf{x}^N} f(\mathbf{d}, \mathbf{x}^N, \mathbf{p}^N) \\ \text{s.t. } & Pl(g_i(\mathbf{d}, \mathbf{X}, \mathbf{P}) < 0) \leq p_{f_i}, \quad i = 1, \dots, n \\ & \mathbf{d}_L \leq \mathbf{d} \leq \mathbf{d}_U, \quad \mathbf{x}_L^N \leq \mathbf{x}^N \leq \mathbf{x}_U^N \end{aligned} \tag{29}$$

where $\mathbf{X} \in R^m$ and $\mathbf{P} \in R^q$ are the vectors of uncertain design variables and parameters. The superscript “N” indicates nominal value of uncertain variables or parameters. The uncertainty is provided by expert opinions.

It should be noted that the plausibility measure is used instead of the equivalent belief measure, in Problem (29). The reason is that at the optimum, the failure domain for each active constraint is usually much smaller than the safe domain over the frame of discernment (FD) (domain of all focal elements with nonzero combined BPA; see next section). As a result, the computation of the plausibility of failure is much more efficient than the computation of the belief of safe region.

5.1. ASSESSING *BEL* AND *PL* WITH DEMPSTER-SHAFER THEORY

Evidence theory can quantify epistemic uncertainty, even when the experts provide conflicting evidence. This section shows how to propagate epistemic uncertainty through a given model (transfer function) which is necessary in calculating the plausibility of constraint violation in Problem (29). The uncertainty propagation will be illustrated using the following simple transfer function

$$y = f(a, b) \tag{30}$$

where $a \in A, b \in B$ are two independent input parameters and y is the output. The combined BPA’s for both a and b are obtained from Dempster’s rule of combining of Eq. (12) if multiple experts have provided evidence for either a or b . With combined information for each input parameter, we define a vector $c = [a_{ci}, b_{cj}]$, needed to calculate the output y as

$$C = A \times B = \{c = [a_{ci}, b_{cj}], a_{ci} \in A, b_{cj} \in B\} \tag{31}$$

where subscript c stands for “combined” and i, j indicate focal elements.

Taking advantage of assumed parameter independency, the BPA for c is

$$m_c(h_{ij}) = m(a_{ci})m(b_{cj}) \tag{32}$$

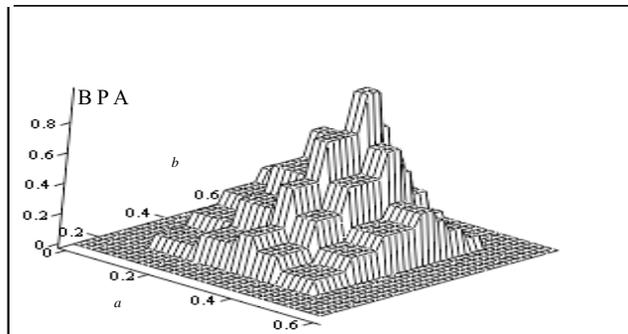


Figure 5. Representative BPA structure for two parameters a and b .

where $h_{ij} = [a_{ci}, b_{cj}]$ and a_{ci}, b_{cj} denote intervals such that $a \in a_{ci}$ and $b \in b_{cj}$. Eq. (32) can be used to calculate the combined BPA structure for the entire domain C . For every $(a,b) \in c \mid c \in C$, needed to evaluate the output y , the combined BPA m_c is used. A representative combined BPA structure is shown in Fig. 5.

The Cartesian product C of Eq. (31) is also called frame of discernment (FD) in the literature. It consists of all focal elements (rectangles in Fig. 5 with nonzero combined BPA) and can be viewed as the finite sample space in probability theory.

If a domain F is defined as

$$F = \{g : g = f(a,b) - y_0 > 0, (a,b) \in c, c = [a_c, b_c] \subset C\} \tag{33}$$

where y_0 is a specified value. According to evidence theory,

$$Bel(F) \leq p_f \leq Pl(F). \tag{34}$$

where $p_f = P(g > 0)$ is the true probability.

The $Bel(F)$ and $Pl(F)$ are calculated using Eqs (6) and (7) where set A is equal to set F of Eq. (33) and B is a rectangular domain (focal element) such that $B \subseteq A$ for Eq. (6) and $B \cap A \neq \emptyset$ for Eq. (7). $B \subseteq A$ means that the focal element must be entirely within the domain $g > 0$ and $B \cap A \neq \emptyset$ means that the focal element must be entirely or partially within the domain $g > 0$ (see Fig. 6). In order to identify if a focal element B satisfies $B \subseteq A$ or $B \cap A \neq \emptyset$, the following minimum and maximum values of g must be calculated

$$[g_{\min}, g_{\max}] = [\min_{\mathbf{x}} g(\mathbf{x}), \max_{\mathbf{x}} g(\mathbf{x})] \tag{35}$$

for $\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U$ where $(\mathbf{x}^L, \mathbf{x}^U)$ defines the focal element domain. For monotonic functions, the vertex method [34] can be used to calculate the minimum and maximum values in Eq. (35) by simply identifying the minimum and maximum values among all vertices of the focal element domain. If for a focal element, g_{\min} and g_{\max} are both positive, the focal element will contribute to the calculation of belief and plausibility. On the other hand, if g_{\min} and g_{\max} are both negative, the focal element will not contribute to the calculation of belief or plausibility. If however, g_{\min} is negative and g_{\max} is positive, the focal element will not contribute to the belief but it will contribute to the plausibility calculation. This is shown schematically in Fig. 6.

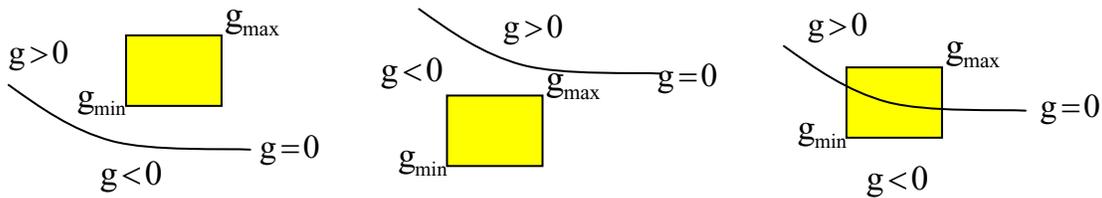


Figure 6. Schematic illustration of focal element contribution to belief and plausibility measures.

In summary the following tasks are performed in order to calculate the belief and plausibility of the failure region:

- 1) For each input parameter, combine the evidence from the experts by combining the individual BPA's from each expert using Dempster's rule of combining (Eq. (12)).
- 2) Construct the BPA structure for the m -dimensional frame of discernment, where m is the number of input parameters. Assuming independent input parameters, Eq. (32) is used.
- 3) Identify the failure region space (set F of Eq. (33)).
- 4) Use Eqs (6) and (7) to calculate the belief and plausibility measures of the failure region. The failure region must be identified only within the frame of discernment. The true probability of failure is bracketed according to Eq. (34).

5.2. IMPLEMENTATION OF THE EBDO ALGORITHM

A computationally efficient solution of Problem (29) is presented here. As a geometrical interpretation of it, we can view the design point (\mathbf{d}, \mathbf{x}) moving within the feasible domain so that the objective f is minimized (see Fig. 7). If the entire FD is in the feasible domain, the constraints are satisfied and are inactive. A constraint becomes active if part of the FD is in the "failure" region so that the plausibility of constraint violation is equal to p_f . In general, Problem (29) represents movement of a hyper-cube (FD) within the feasible domain.

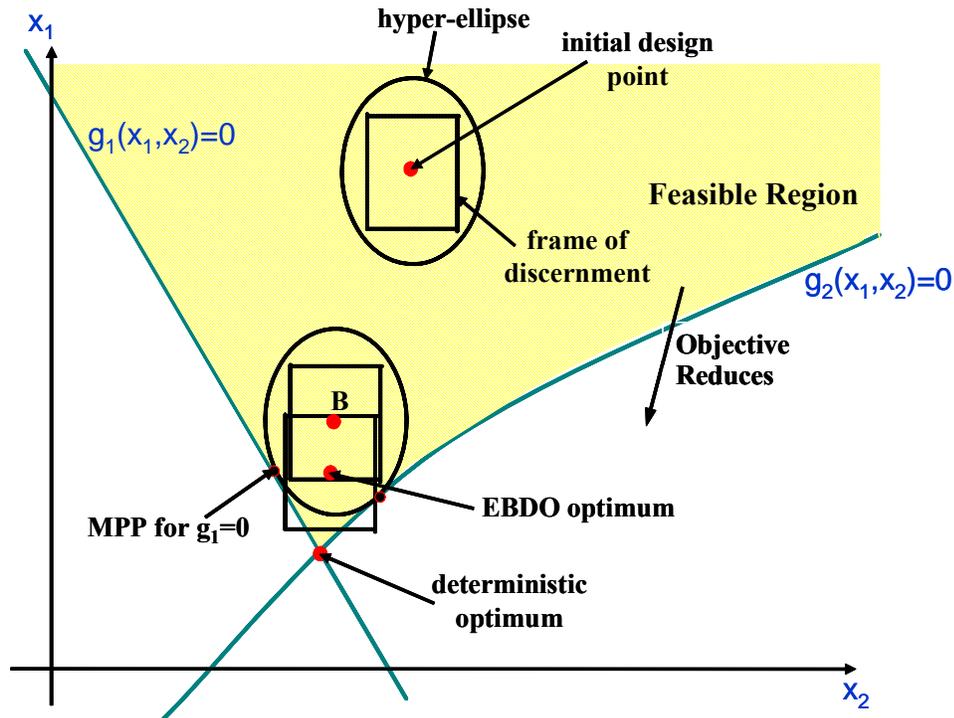


Figure 7. Geometrical interpretation of the EBDO algorithm

In order to save computational effort, the bulk of the FD movement, from the initial design point to the *vicinity* of the optimal point (point B of Fig. 7), can be achieved by *moving a hyper-ellipse which contains the FD*. The center of the hyper-ellipse is the “approximate” design point and each axis is arbitrarily taken equal to three times the standard deviation of a hypothetical normal distribution. This assumes that each dimension of the FD hyper-cube is equal to six times the standard deviation of the hypothetical normal distribution. The hyper-ellipse can be easily moved in the design space by solving a RBDO problem. The RBDO optimum (point B of Fig. 7) is in the vicinity of the solution of Problem (29) (EBDO optimum). The RBDO solution also identifies all active constraints and their corresponding most probable points (MPP’s). The maximal possibility search algorithm (Choi, Du and Youn, 2004) can also be used to move the FD hyper-cube in the feasible domain. It should be noted that the 3-sigma axes hyper-ellipse is arbitrary. The size of the hyper-ellipse is not however, crucial because it is only used to calculate the initial point (point B of Fig. 7) of the EBDO algorithm. The latter calculates the true EBDO optimum accurately. From our experience, a 3 to 4- σ size works fine.

At this point, we generate a *local* response surface of each active constraint around its MPP. In this work, the Cross-Validated Moving Least Squares (CVMLS) [39] method is used based on an Optimum Symmetric Latin Hypercube (OSLH) [40] “space-filling” sampling.

A derivative-free optimizer calculates the EBDO optimum. It uses as initial point the previously calculated RBDO optimum which is close to the EBDO optimum. Problem (29) is solved, considering only the identified active constraints. For the calculation of the plausibility of failure $Pl(g < 0)$ of each active constraint, an algorithm presented in (Mourelatos and Zhou, 2005) is used. It identifies all focal elements which contribute to the plausibility of failure. The computational effort is significantly reduced because accurate local response surfaces are used for the active constraints. The cost can be much higher if the optimization algorithm evaluates the actual active constraints instead of their efficient surrogates (response surfaces). It should be noted that a derivative-free optimizer is needed due to the discontinuous nature of the combined BPA structure. The DIRECT derivative-free, global optimizer is used (Jones, Perttunen and Stuckman, 1993).

6. EXAMPLES

In this section, the possibility-based and evidence-based design algorithms are demonstrated with a cantilever beam example and a pressure vessel example. For both examples, comparisons are made with deterministic design and reliability-based design results. It should be noted that theoretically, the possibility and reliability-based results can not be compared because the possibility and reliability theories are based on different axioms. However for practical purposes, we attempt to compare them by arbitrarily using membership functions which “resemble” the probability density functions used in the reliability-based results.

6.1. A CANTILEVER BEAM EXAMPLE

In this example, a cantilever beam in vertical and lateral bending (Wu, Shin, Sues and Cesare, 2001) is used (see Fig. 8). 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

deterministic design variables. The objective is to minimize the weight of the beam. This is equivalent to minimizing $f = w * t$, assuming that the material density and the beam length are

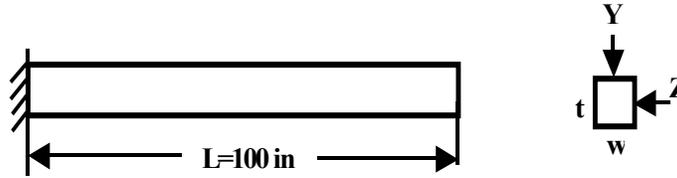


Figure 8. Cantilever beam under vertical and lateral bending

constant.

Two non-linear failure modes are used. The first failure mode is yielding at the fixed end of the cantilever; the other failure mode is that the tip displacement exceeds the allowable value of $D_0 = 2.5''$. The PBDO problem is formulated as,

$$\begin{aligned}
 & \min_{w,t} f = w * t \\
 \text{s.t. } & g_{j_{\min}}^{\alpha} \geq 0 \quad j = 1,2 \\
 & g_1(y, Z, Y, w, t) = y - \left(\frac{600}{wt^2} * Y + \frac{600}{w^2t} * Z \right) \\
 & g_2(E, Z, Y, w, t) = D_0 - \frac{4L^3}{Ewt} \sqrt{\left(\frac{Y}{t^2} \right)^2 + \left(\frac{Z}{w^2} \right)^2} \\
 & 0 \leq w, t \leq 5
 \end{aligned} \tag{40}$$

where g_1 and g_2 are the limit states corresponding to the two failure modes. The design variables w and t are deterministic. In the RBDO study of [2], Y , Z , y and E are normally distributed random parameters with $Y \sim N(1000, 100)$ lb, $Z \sim N(500, 100)$ lb, $y \sim N(40000, 2000)$ psi and $E \sim N(29 * 10^6, 1.45 * 10^6)$ 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 $\beta = 3$ has been used in [2] for both constraints.

For the PBDO case, Y , Z , y and E are possibilistic parameters described with the triangular membership functions $(x^N - 3 * \sigma, x^N, x^N + 3 * \sigma)$ where x^N is the normal point of each variable and σ is the used standard deviation in the RBDO study. The frame of discernment defined by the $(x^N - 3 * \sigma, x^N + 3 * \sigma)$ coordinates is also used in EBDO.

Table 2. Comparison of PBDO, EBDO and RBDO optima for the cantilever beam example

Design Variables	Determ. Optimum	Reliability Optimum	Possibility Optimum		Evidence Optimum	
			$\alpha=0.1$	$\alpha=0$	$p_f = 0.1$	$p_f = 0.0013$
w	2.0470	2.4781	2.5298	2.5901	2.4534	2.5028
t	3.7459	3.8421	4.1726	4.210	3.6162	3.9902
Objective						
f(w,t)	7.6679	9.5212	10.556	10.901	8.8721	9.9868
Constraints						
$g_1(\mathbf{x}) / \bar{y}$	0	0	0	0	0	0.0032
$g_2(\mathbf{x}) / D_0$	0	0.1436	0.15	0.168	0.00428	0.0835

Table 2 compares the deterministic optimization, RBDO, PBDO and EBDO results. The PBDO optimum (objective function) with $a=0$ is higher than the RBDO optimum. Because it represents the worst case design, it provides an upper bound of all RBDO optima obtained with different distributions, as long as these distributions have similar variability ranges (e.g. different beta distributions defined over the same range). For a higher a -cut ($a=0.1$), the PBDO optimum reduces. It should be noted that the PBDO optimum at $a=1$ coincides with the deterministic optimum. The last two rows of Table 2 show the normalized values of the two constraints at the

Table 3. BPA structure for y, Y, Z and E

Z		y (x10³)	
Interval	BPA	Interval	BPA
[200 300]	2.2%	[35 37]	6.1%
[300 400]	13.6%	[37 38]	9.2%
[400 450]	15%	[38 39]	15%
[450 500]	19.2%	[39 40]	19.2%
[500 550]	19.2%	[40 41]	19.2%
[550 600]	15%	[41 42]	15%
[600 700]	13.6%	[42 43]	9.2%
[700 800]	2.2%	[43 45]	7.1%

Y		E (x10⁶)	
Interval	BPA	Interval	BPA
[700 800]	2.2%	[26.5 27.5]	10%
[800 900]	13.6%	[27.5 28.5]	21%
[900 1000]	34.1%	[28.5 29]	13.5%
[1000 1100]	34.1%	[29 29.5]	13.5%
[1100 1200]	13.6%	[29.5 30.5]	21%
[1200 1300]	2.4%	[30.5 31.3]	21%

optimum. The first constraint is normalized by the mean yield strength $\bar{y} = 40000$ and the second constraint is normalized by the allowable tip displacement $D_0 = 2.5$. Although both constraints are active at the deterministic optimum, only the first constraint is active for both the RBDO and PBDO optima.

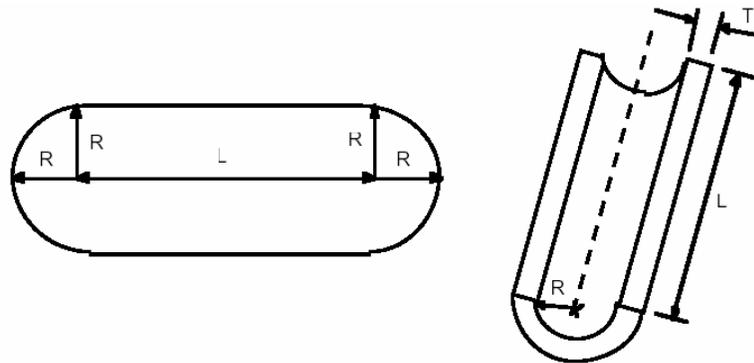
The EBDO problem formulation is the same with Problem (40) but with different constraints. The new constraints are $Pl(g_i < 0) \leq p_f, i=1,2$. The uncertain parameters $\mathbf{P}=[Y,Z,y,E]$ have the

BPA structure of Table 3. The BPA for each interval of an uncertain parameter is assumed to be equal to the area under the PDF used in RBDO, in order to compare the EBDO design with the corresponding RBDO design. This is not how the BPA is obtained in general. As it has been mentioned, expert opinions are used to construct the BPA structure. If however, a random variable or parameter is described probabilistically, equivalent BPA values within specified intervals are calculated as equal to the area under the PDF. In doing so, the evidence theory can be used to handle a mixture of probabilistic and non-probabilistic variables.

The last two columns of Table 2 show the EBDO results for $p_f = 0.1$ and 0.0013 ($\beta = 3$). As expected, the deterministic optimum of 7.6679 is less than the RBDO optimum of 9.5212 which in turn, is less than the EBDO optimum of 9.9868 at $p_f = 0.0013$ ($\beta = 3$). For $p_f = 0.1$, the EBDO optimum reduces. Furthermore, the EBDO optimum of 9.9868 at $p_f = 0.0013$ is better than the worst case PBDO optimum of 10.901 ($a=0$). Although only the first constraint is active for the RBDO and PBDO optima, both constraints are active for the EBDO optima, similarly to the deterministic case.

6.2. A PRESSURE VESSEL EXAMPLE

This example considers the design of a thin-walled pressure vessel (Lewis and Mistree, 1997) which has hemispherical ends as shown in Fig. 9. The design objective is to calculate the radius R , mid-section length L and wall thickness t in order to maximize the volume while avoiding yielding of the material in both the circumferential and radial directions under an internal pressure P . Geometric constraints are also considered. The material yield strength is Y . A safety



factor $SF = 2$ is use

Figure 9. Thin-walled pressure vessel.

The PBDO problem is stated as

$$\begin{aligned} \max_{R_N, L_N, t_N} \quad & f = \frac{4}{3}\pi R_N^3 + \pi R_N^2 L_N \\ \text{s.t.} \quad & g_{j_{\min}}^\alpha \geq 0 \quad j = 1, \dots, 5 \end{aligned}$$

$$\begin{aligned}
 g_1(\mathbf{X}) &= 1.0 - \frac{P(R + 0.5t)SF}{2tY} \\
 \text{where, } g_2(\mathbf{X}) &= 1.0 - \frac{P(2R^2 + 2Rt + t^2)SF}{(2Rt + t^2)Y} \\
 g_3(\mathbf{X}) &= 1.0 - \frac{L + 2R + 2t}{60} \\
 g_4(\mathbf{X}) &= 1.0 - \frac{R + t}{12} \\
 g_5(\mathbf{X}) &= 1.0 - \frac{5t}{R} \\
 0.25 &\leq t_N \leq 2.0 \\
 6.0 &\leq R_N \leq 24 \\
 10 &\leq L_N \leq 48
 \end{aligned}$$

Table 4. BPA structure for R, L, t, P and Y

R	L	t	BPA
$[R_N - 6.0 \ R_N - 4.5]$	$[L_N - 12 \ L_N - 9]$	$[t_N - 0.4 \ t_N - 0.3]$	0.13%
$[R_N - 4.5 \ R_N - 3.0]$	$[L_N - 9 \ L_N - 6]$	$[t_N - 0.3 \ t_N - 0.2]$	2.15%
$[R_N - 3.0 \ R_N]$	$[L_N - 6 \ L_N]$	$[t_N - 0.2 \ t_N]$	47.72%
$[R_N \ R_N + 3.0]$	$[L_N \ L_N + 6]$	$[t_N \ t_N + 0.2]$	47.72%
$[R_N + 3.0 \ R_N + 4.5]$	$[L_N + 6 \ L_N + 9]$	$[t_N + 0.2 \ t_N + 0.3]$	2.15%
$[R_N + 4.5 \ R_N + 6.0]$	$[L_N + 9 \ L_N + 12]$	$[t_N + 0.3 \ t_N + 0.4]$	0.13%

P	Y	BPA
[800 850]	[208000 221000]	0.13%
[850 900]	[221000 234000]	2.15%
[900 1000]	[234000 260000]	47.72%
[1000 1100]	[260000 286000]	47.72%
[1100 1150]	[286000 299000]	2.15%
[1150 1200]	[299000 312000]	0.13%

The EBDO problem formulation is the same but with constraints $Pl(g_j(\mathbf{X}) < 0) \leq p_f \quad j=1, \dots, 5$. For the EBDO case, the uncertainty in design variables $R, L,$ and t and design parameters P and Y are represented with the combined BPA structure of Table 4. To compare results with RBDO, the BPA values of R, L, t, P and Y are taken equal to the area under the PDF of a normal distribution for the intervals shown in Table 4. The

normal distributions for R , L , t , P and Y have standard deviations equal to 1.5, 3, 0.1, 50 and 13000, respectively. The mean values for parameters P and Y are taken equal to 1000 and 260000. The intervals for R , L , t , P and Y extend four standard deviations from each side of the normal point, in an attempt to use a similar variation with the RBDO study. Finally, EBDO and PBDO use the same frame of discernment.

Table 5 compares the deterministic optimization, RBDO, PBDO and EBDO results. Similar conclusions with the previous example are drawn. A reliability index $\beta = 2.0$ ($p_f = 0.0228$) has been used in the RBDO study for all constraints. As expected, the deterministic maximum volume of 22400 is higher than the RBDO volume of 10791 which in turn, is higher than the EBDO volume of 7644. Also, the PBDO optimum of 6132 ($a=0$) which represents the worst case, is the lowest. For comparison purposes, the PBDO and EBDO results are also presented for $a=0.2$ and $p_f=0.0228$, respectively. It is noted that the constraint activity changes among the deterministic, RBDO, PBDO and EBDO optima. Only the third and fourth constraints are active for the deterministic case. However, the second, third and fourth constraints become active at the RBDO and PBDO optima. At the EBDO optimum all constraints are active except the fifth one.

Table 5. Comparison of deterministic, RBDO, PBDO and EBDO optima for vessel example

Design Variables	Deterministic Optimum	Reliability Optimum	Possibility Optimum		Evidence Optimum	
			$a=0.2$	$a=0$	$p_f=0.2$	$p_f=0.0228$
R_N	11.750	8.7244	7.9107	7.0107	8.333	8.1111
L_N	36.000	33.5186	30.3867	30.3867	30.407	26.1852
t_N	0.250	0.269	0.2893	0.2893	0.347	0.3472
Objective						
$-f(R_N, L_N)$	22400	10791	8044	6132	9053	7644
Constraints						
$g_1(\mathbf{x})$	0.8173	0.5003	0.5	0.5	0	0
$g_2(\mathbf{x})$	0.6346	0	0	0	0.0137	0
$g_3(\mathbf{x})$	0	0	0	0	0	0.0183
$g_4(\mathbf{x})$	0	0	0	0	0	0.0118
$g_5(\mathbf{x})$	0.8936	0.6891	0.4325	0.0256	0.9994	0.1038

7. SUMMARY AND CONCLUSIONS

In this paper, the possibility and evidence theories were used to assess design reliability with incomplete information. The possibility theory was viewed as a variant of fuzzy set theory. The different types of uncertainty and formal uncertainty theories were first introduced using the fundamentals of fuzzy measures. Subsequently, the commonly used vertex and discretization methods which are used for propagating non-probabilistic uncertainty were reviewed and compared with a hybrid (global-local) optimization method. It was shown that the hybrid optimization method is very efficient and has the same accuracy with the “brute force” discretization method.

The possibility theory was also used in design. A possibility-based design optimization method was proposed where all design constraints are expressed possibilistically. It was shown that the method gives a conservative solution compared with all conventional reliability-based designs obtained with different probability distributions. A general possibility-based design optimization method was also presented which handles a combination of random and possibilistic design variables.

Furthermore, a computationally efficient design optimization method was described, which can handle a mixture of epistemic and random uncertainties. A mean performance is optimized subject to the plausibility of constraint violation being small. Uncertainty is quantified using “expert” opinions. Two examples demonstrated the proposed possibility-based and evidence-based design optimization methods. It was shown that both the PBDO and EBDO designs are more conservative compared with the RBDO design. However, the EBDO design is usually less conservative compared with the PBDO design.

ACKNOWLEDGEMENT

This study was performed with funding from the General Motors Research and Development Center and the Automotive Research Center (ARC), a U.S. Army Center of Excellence in Modeling and Simulation of Ground Vehicles at the University of Michigan. The support is gratefully acknowledged. Such support does not however, constitute an endorsement by the funding agencies of the opinions expressed in the paper.

REFERENCES

- Agarwal, H., Renaud, J. E., Preston, E. L. and Padmanabhan, D., “Uncertainty Quantification Using Evidence Theory in Multidisciplinary Design Optimization,” *Reliability Engineering and System Safety*, 85, 281-294, 2004.
- Akpan, U. O., Rushton, P. A. and Koko, T. S., “Fuzzy Probabilistic Assessment of the Impact of Corrosion on Fatigue of Aircraft Structures,” Paper AIAA-2002-1640, 2002.
- Bae, H-R, Grandhi, R. V. and Canfield, R. A., “An Approximation Approach for Uncertainty Quantification Using Evidence Theory,” *Reliability Engineering and System Safety*, 86, 215-225, 2004.
- Bae, H-R, Grandhi, R. V. and Canfield, R. A., “Epistemic Uncertainty Quantification Techniques

- Including Evidence Theory for Large-Scale Structures,” *Computers and Structures*, 82, 1101-1112, 2004.
- Chen, L. and Rao, S. S., “Fuzzy Finite Element Approach for the Vibration Analysis of Imprecisely Defined Systems,” *Finite Elements in Analysis and Design*, 27, 69-83, 1997.
- Choi, K. K., Du, L. and Youn, B. D., “A New Fuzzy Analysis Method for Possibility-Based Design Optimization,” 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA 2004-4585, Albany, NY, 2004.
- Du, X. and Chen, W., “An Integrated Methodology for Uncertainty Propagation and Management in Simulation-Based Systems Design,” *AIAA Journal*, 38(8), 1471-1478, 2000.
- Du, X. and Sudjianto, A., “Reliability-Based Design with a Mixture of Random and Interval Variables,” *Proceedings of ASME Design Engineering Technical Conferences*, Paper# DETC2003/ DAC-48709, 2003.
- Dubois, D. and Prade, H., *Possibility Theory*, Plenum Press, New York, 1988.
- Elishakoff, I. E., Haftka, R. T. and Fang, J. “Structural Design under Bounded Uncertainty – Optimization with Anti-Optimization,” *Computers and Structures*, 53, 1401-1405, 1994.
- Gu, X., Renaud, J. E. and Batill, S. M., “An Investigation of Multidisciplinary Design Subject to Uncertainties,” 7th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis & Optimization Symposium, St. Louis, Missouri, 1998.
- Jones, D. R., Perttunen, C. D. and Stuckman, B. E., “Lipschitzian Optimization Without the Lipschitz Constant,” *Journal of Optimization Theory and Applications*, 73(1), 157-181, 1993.
- Klir, G. J. and Filger, T. A., *Fuzzy Sets, Uncertainty, and Information*, Prentice Hall, 1988.
- Klir, G. J. and Yuan, B., *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, 1995.
- Lee, J. O., Yang, Y. O. and Ruy, W. S., “A Comparative Study on Reliability Index and Target Performance Based Probabilistic Structural Design Optimization,” *Computers and Structures*, 80, 257-269, 2002.
- Lewis, K. and Mistree, F., “Collaborative, Sequential and Isolated Decisions in Design,” *Proceedings of ASME Design Engineering Technical Conferences*, Paper# DETC1997/ DTM-3883, 1997.
- 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.
- Lombardi, M. and Haftka, R. T., “Anti-Optimization Technique for Structural Design under Load Uncertainties,” *Computer Methods in Applied Mechanics and Engineering*, 157, 19-31, 1998.
- Moore, R. E., *Interval Analysis*, Prentice-Hall, 1966.
- Mourelatos, Z. P. and Zhou, J., “Reliability Estimation with Insufficient Data Based on Possibility theory”, *AIAA Journal*, 43(8), 1696-1705, 2005.
- Mourelatos, Z. P. and Zhou, J., “A Design Optimization Method using Evidence Theory,” accepted *ASME Journal of Mechanical Design*, December 2005.
- Muhanna, R. L. and Mullen, R. L., “Uncertainty in Mechanics Problems – Interval-Based Approach,” *Journal of Engineering Mechanics*, 127(6), 557-566, 2001.
- Mullen, R. L. and Muhanna, R. L., “Bounds of Structural Response for all Possible Loadings,” *ASCE Journal of Structural Engineering*, 125(1), 98-106, 1999.

- Nikolaidis, E., Chen, S., Cudney, H., Haftka, R. T. and Rosca, R., "Comparison of Probability and Possibility for Design Against Catastrophic Failure Under Uncertainty," *ASME Journal of Mechanical Design*, 126, 2004.
- Oberkampf, W. L. and Helton, J. C., "Investigation of Evidence Theory for Engineering Applications," *AIAA Non-Deterministic Approaches Forum*, AIAA 2002-1569, Denver, CO, April, 2002.
- Oberkampf, W., Helton, J. and Sentz, K., "Mathematical Representations of Uncertainty," *AIAA Non-Deterministic Approaches Forum*, AIAA 2001-1645, Seattle, WA, April 16-19, 2001.
- Penmetsa, R. C. and Grandhi, R. V. "Estimating Membership Response Function using Surrogate Models," Paper AIAA 2002-1234, 2002.
- Penmetsa, R. C. and Grandhi, R. V., "Efficient Estimation of Structural Reliability for Problems with Uncertain Intervals," *Computers and Structures*, 80, 1103-1112, 2002.
- Rao, S. S. and Cao, L., "Optimum Design of Mechanical Systems Involving Interval Parameters," *ASME Journal of Mechanical Design*, 124, 465-472, 2002.
- Rao, S. S. and Sawyer, J. P., "A Fuzzy Finite Element Approach for the Analysis of Imprecisely Defined Systems," *AIAA Journal*, 33, 2264-2370, 1995.
- Ross, T. J., *Fuzzy Logic with Engineering Applications*, McGraw Hill, 1995.
- Sentz, K. and Ferson, S., "Combination of Evidence in Dempster – Shafer Theory," Sandia National Laboratories Report SAND2002-0835, April 2002.
- Sentz, K. and Ferson, S., "Combination of Evidence in Dempster – Shafer Theory," Sandia National Laboratories Report SAND2002-0835, April 2002.
- Tu, J. and Jones, D. R., "Variable Screening in Metamodel Design by Cross-Validated Moving Least Squares Method", *Proceedings 44th AIAA/ASME/ASCE/ AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA-2003-1669, Norfolk, VA, April 7-10, 2003.
- 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.
- Wang, G., "Adaptive Response Surface Method Using Inherited Latin Hypercube Design Points," *ASME Journal of Mechanical Design*, 125, 1-11, 2003.
- 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.
- Yager, R. R., Fedrizzi, M. and Kacprzyk, J. (Editors), *Advances in the Dempster – Shafer Theory of Evidence*, John Wiley & Sons, Inc., 1994.
- Ye, K. Q., Li, W. and A. Sudjianto, "Algorithmic Construction of Optimal Symmetric Latin Hypercube Designs", *Journal of Statistical Planning and Inference*, 90, 145-159, 2000.
- 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, 2001.
- Zadeh, L. A., "Fuzzy Sets as a Basis for a Theory of Possibility," *Fuzzy Sets and Systems*, 1, 3-28, 1978.
- Zadeh, L. A., "Fuzzy Sets," *Information and Control*, 8, 338-353, 1965.

Prediction of uncertain structural responses with fuzzy time series

BERND MÖLLER, UWE REUTER

*Institut für Statik und Dynamik der Tragwerke, Technische Universität Dresden, 01062 Dresden, Germany,
email: moeller@rzs.urz.tu-dresden.de, uwe.reuter@tu-dresden.de*

Abstract. In this paper mathematical methods for prediction of uncertain structural responses with the aid of fuzzy time series are presented. Uncertain measurements of structural loads and responses respectively at equally spaced discrete time points are modeled as fuzzy variables. Hence uncertain measurements over time are considered as time series with fuzzy data. The fuzzy variables are processed on the basis of generally applicable numerical methods for descriptive analysis as well as for stochastic analysis. Algorithms of stochastic analysis are used to forecast fuzzy time series. At this the new fuzzy-ARMA-process is introduced. Forecasts of fuzzy time series provides informationen about future structural responses.

The algorithm of analysis and forecast of fuzzy time series are presented in detail and demonstrated by way of numerical examples.

Keywords: Fuzzy time series; Fuzzy random processes; Fuzzy random variables; forecast

1. Introduction

The prediction of future structural responses is a challenging problem in civil engineering. The knowledge of unknown future impact and future system behavior enables the prediction of such important effects like damage behavior, development of safety level, development of durability or the expected life time of a system. The well established numerical structural analysis and safety assessment however presuppose the knowledge of adequate theoretical models.

As alternative fuzzy time series can be applied. They describe sequences of measurements consisting of imprecise data (Hareter, 2003). The uncertainty of the imprecise data is modeled as fuzziness (Möller and Beer, 2004). Time series with fuzzy data are regarded as realizations of a fuzzy random process, that can be viewed as a random process extended by the dimension fuzziness (Möller et al., 2005). In extension to a random process a fuzzy random process is defined as a sequence of fuzzy random variables. Therein, a fuzzy random variable is declared as set of uncertain realizations (fuzzy variables) in the space of the random elementary events. Each realization of a fuzzy random process then appears as a fuzzy function, which characterizes a sequence of fuzzy variables. In other words time series with fuzzy data can be interpreted as random realizations of an underlying fuzzy random process.

Methods for identification and quantification of the underlying fuzzy random process are presented. A new description of fuzzy variables by so called $l_\alpha r_\alpha$ -discretization has been developed. This description enables prediction without the usually performed defuzzification and refuzzification of fuzzy data. The following types of fuzzy random processes are investigated: fuzzy-AR-processes, fuzzy-MA-processes, fuzzy-ARMA-processes, and fuzzy-white-noise-processes. Strategies for parameter estimation have been

developed that are applicable for stationary and non-stationary fuzzy time series. After parameter estimation the underlying fuzzy random process is known and can be used for forecasting.

The developed theory is demonstrated by way of examples among others the heavy goods vehicle traffic over a bridge is forecasted. Furthermore, on the basis of measured settlements over a period of four years the future settlements for the next three years are predicted with a h -step-forecast.

2. Definition and description of fuzzy time series

Fuzzy time series are interpreted as random realizations of an underlying fuzzy random process. A fuzzy random process $(\tilde{X}_\tau)_{\tau \in \mathbf{T}}$ is defined as a family of fuzzy random variables \tilde{X}_τ with $\tau \in \mathbf{T}$. Thereby \mathbf{T} denotes the space of equidistant points in time. In other words a fuzzy random process $(\tilde{X}_\tau)_{\tau \in \mathbf{T}}$ is defined as the fuzzy result of the mapping

$$\tilde{X}_\tau : \Omega \rightarrow \mathbf{F}(\mathbb{R}) \tag{1}$$

in which Ω denotes the space of the random elementary events ω and $\mathbf{F}(\mathbb{R})$ characterizes the set of all fuzzy numbers on \mathbb{R} . Fuzzy realizations $\tilde{X}_\tau(\omega) = \tilde{x}_\tau$ with $\tau \in \mathbf{T}$ are assigned to each random elementary event $\omega \in \Omega$. Consequently the realizations of a fuzzy random process $(\tilde{X}_\tau)_{\tau \in \mathbf{T}}$ form the fuzzy time series $(\tilde{x}_\tau)_{\tau \in \mathbf{T}}$. A realization $(\tilde{x}_\tau)_{\tau \in \mathbf{T}}$ of a fuzzy random process is plotted in Fig. 1.

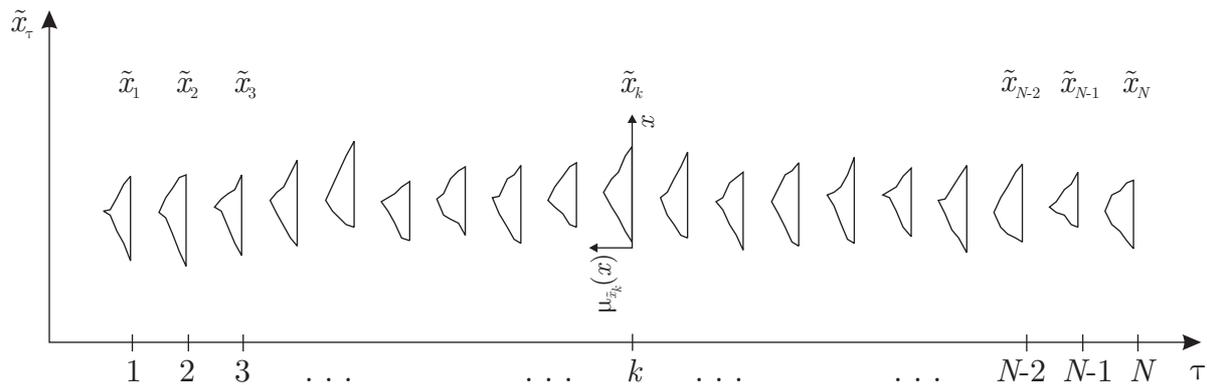


Figure 1. Fuzzy time series as realization of a fuzzy random process

At each specified point $\tau \in \mathbf{T}$ a fuzzy time series specifies a fuzzy variable \tilde{x}_τ in accordance with Eq. 1. A fuzzy variable \tilde{x} is characterized by its membership function $\mu_{\tilde{x}}(x)$. A normalized membership function $\mu_{\tilde{x}}(x)$ is defined by the following equations.

$$0 \leq \mu_{\tilde{x}}(x) \leq 1 \quad \forall x \in \mathbb{R} \tag{2}$$

$$\exists x_l, x_r \text{ mit } \mu_{\tilde{x}}(x) = 1 \quad \forall x \in [x_l; x_r] \tag{3}$$

A fuzzy variable \tilde{x} is referred to as convex if its membership function $\mu_{\tilde{x}}(x)$ monotonically decreases on each side of the maximum value, i.e., if

$$\mu_{\tilde{x}}(x_2) \geq \min [\mu(x_1); \mu(x_3)] \quad \forall x_1, x_2, x_3 \in \mathbb{R} \text{ mit } x_1 \leq x_2 \leq x_3 \quad (4)$$

applies.

A convex fuzzy variable \tilde{x} is referred to as fuzzy number \tilde{x}_Z if its membership function $\mu_{\tilde{x}}(x)$ is at least segmentally continuous and has the functional value $\mu_{\tilde{x}}(x) = 1$ at precisely one of the x values according to Eq. (5).

$$\begin{aligned} x_l = x_r \quad \text{with} \quad x_l = \min [x \in \mathbb{R} | \mu_{\tilde{x}_\tau}(x) = 1] \\ \text{and} \quad x_r = \max [x \in \mathbb{R} | \mu_{\tilde{x}_\tau}(x) = 1] \end{aligned} \quad (5)$$

In the case $x_l < x_r$, the fuzzy variable \tilde{x} is a fuzzy interval \tilde{x}_I . The point x_l is referred to as the peak point of the fuzzy variable.

A convex fuzzy variable \tilde{x} is characterized by a family of α -level sets X_α according to Eq. (6). Each α -level set X_α is a connected interval $[x_{\alpha l}, x_{\alpha r}]$.

$$\tilde{x} = (X_\alpha = [x_{\alpha l}, x_{\alpha r}] | \alpha \in [0, 1]) \quad (6)$$

The number of α -level sets is denoted by n . For $i = 1, 2, \dots, n - 1$ the following holds.

$$0 \leq \alpha_i \leq \alpha_{i+1} \leq 1 \quad (7)$$

$$\alpha_1 = 0 \quad \text{und} \quad \alpha_n = 1 \quad (8)$$

$$X_{\alpha_{i+1}} \subseteq X_{\alpha_i} \quad (9)$$

An example of a convex fuzzy variable \tilde{x} characterized by $n = 4$ α -level sets X_α is shown in Fig. 2.

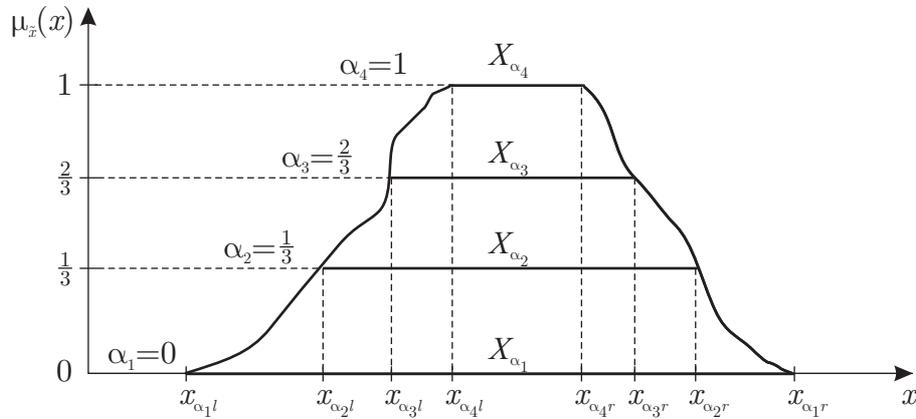


Figure 2. α -discretization of a convex fuzzy variable

In the following the new $l_\alpha r_\alpha$ -discretization is presented. The interval boundaries $[x_{\alpha_i l}, x_{\alpha_i r}]$ of an α -level set X_{α_i} are expressed by Eqs. (10) and (11).

$$x_{\alpha_i l} = x_{\alpha_{i+1} l} - \Delta x_{\alpha_i l} \quad \text{with} \quad \Delta x_{\alpha_i l} = x_{\alpha_i l r} - x_{\alpha_i l l} \quad (10)$$

$$x_{\alpha_i r} = x_{\alpha_{i+1} r} + \Delta x_{\alpha_i r} \quad \text{with} \quad \Delta x_{\alpha_i r} = x_{\alpha_i r r} - x_{\alpha_i r l} \quad (11)$$

The counter $i = 1, 2, \dots, n - 1$ specifies α -level sets with $\alpha < 1$. For $i = 1$ the following equations hold, whereat the term $\Delta x_{\alpha_n l}$ is assigned to the peak point x_l .

$$x_{\alpha_n l} = \Delta x_{\alpha_n l} \quad \text{with} \quad \Delta x_{\alpha_n l} = x_l \quad (12)$$

$$x_{\alpha_n r} = x_{\alpha_n l} + \Delta x_{\alpha_n r} \quad \text{with} \quad \Delta x_{\alpha_n r} = x_r - x_l \quad (13)$$

The terms $\Delta x_{\alpha_i l}$ and $\Delta x_{\alpha_i r}$ are called $l_{\alpha r \alpha}$ -increments. The α -level sets have to fulfill Eq. (14).

$$X_{\alpha_k} \subseteq X_{\alpha_i} \quad \forall \alpha_i, \alpha_k \in [0; 1] \quad \text{with} \quad \alpha_i \leq \alpha_k \quad (14)$$

With Eqs. (10) to (14) the $l_{\alpha r \alpha}$ -discretization is introduced. Fig. 3 illustrates the $l_{\alpha r \alpha}$ -discretization for $n = 4$.

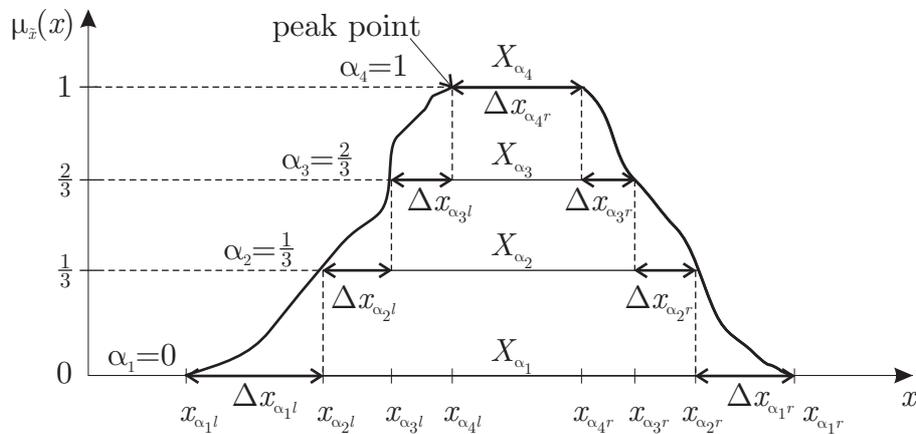


Figure 3. $l_{\alpha r \alpha}$ -Diskretisierung with 4 α -level sets

The $l_{\alpha r \alpha}$ -discretization enables an alternative, discrete representation of a fuzzy variable \tilde{x} in the form of a column matrix introduced by Eq. (15), thereby $\Delta x_1, \Delta x_2, \dots, \Delta x_{2n}$ is a shortened form of $\Delta x_{\alpha_1 l}, \Delta x_{\alpha_2 l}, \dots, \Delta x_{\alpha_1 r}$.

$$\tilde{x} = \begin{bmatrix} \Delta x_{\alpha_1 l} \\ \Delta x_{\alpha_2 l} \\ \vdots \\ \Delta x_{\alpha_n l} \\ \Delta x_{\alpha_n r} \\ \vdots \\ \Delta x_{\alpha_2 r} \\ \Delta x_{\alpha_1 r} \end{bmatrix} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \\ \Delta x_{n+1} \\ \vdots \\ \Delta x_{2n-1} \\ \Delta x_{2n} \end{bmatrix} \quad (15)$$

In context of time series with fuzzy data the following operators are introduced.

The multiplication of a real-valued $[2n, 2n]$ matrix \underline{A} with a fuzzy variable \tilde{x} represented by n α -levels is defined by the operator \odot according to Eqs. (16) and (17). The arithmetic operation is equivalent to the matrix product and results the $l_\alpha r_\alpha$ -increments Δz_j ($j = 1, 2, \dots, 2n$) of the fuzzy result variable \tilde{z} .

$$\underline{A} \odot \tilde{x} = \tilde{z} \tag{16}$$

$$\begin{bmatrix} a_{1,1} & a_{2,2} & \dots & a_{1,2n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{2n,1} & a_{2n,2} & \dots & a_{2n,2n} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_{2n} \end{bmatrix} = \begin{bmatrix} \Delta z_1 \\ \Delta z_2 \\ \vdots \\ \Delta z_{2n} \end{bmatrix} \tag{17}$$

The fuzzy result variable \tilde{z} requires compliance with Eq. (14), so that Eq. (18) must be satisfied for $j = 1, 2, \dots, n - 1, n + 2, \dots, 2n$.

$$\Delta z_j = a_{j,1}\Delta x_1 + \dots + a_{j,2n}\Delta x_{2n} \geq 0 \tag{18}$$

Furthermore a special fuzzy sum and subtraction respectively is required. The operators \oplus and \ominus respectively between two fuzzy variables \tilde{x} and \tilde{y} pursuant to Eq. (19) are introduced as the addition and subtraction respectively of the $l_\alpha r_\alpha$ -increments according to Eq. (19)

$$\tilde{z} = \tilde{x} \oplus \tilde{y} \text{ bzw. } \tilde{z} = \tilde{x} \ominus \tilde{y} \tag{19}$$

The fuzzy result variable \tilde{z} requires compliance with Eq. (14), too. The corresponding conditions are shown in Eq. (20) in which the upper operators are applied for the fuzzy sum and the lower for the fuzzy difference.

$$\Delta z_j = \Delta x_j \pm \Delta y_j \geq 0 \quad \text{for } j = 1, 2, \dots, n - 1, n + 2, \dots, 2n \tag{20}$$

Considering the priority rule (\odot comes before \oplus) a combination of the introduced operators according to Eq. (21) is feasible.

$$\tilde{z} = \underline{A} \odot \tilde{x} \oplus \dots \ominus \dots \oplus \dots \oplus \underline{B} \odot \tilde{y} \tag{21}$$

The fuzzy result variable \tilde{z} also requires compliance with Eq. (14). But only the final $l_\alpha r_\alpha$ -increments Δz_j must be nonnegative, negative intermediate results due the application of the associative law are allowed.

$$\Delta z_j \geq 0 \quad \text{for } j = 1, 2, \dots, n - 1, n + 1, \dots, 2n \tag{22}$$

The demand according to Eq. (22) also represents an boundary condition for the models introduced in the paper.

According to Eq. (1) a fuzzy variable \tilde{x}_τ is interpreted as a random realization of a fuzzy random variable \tilde{X}_τ . Under the assumption of convex fuzzy realizations $\tilde{X}_\tau(\omega) = \tilde{x}$ a fuzzy random variable \tilde{X}_τ is characterized by a family of random α -level sets X_α according to Eq. (23). At this the intervall boundaries $X_{\alpha l}$ and $X_{\alpha r}$ are real-valued random variables.

$$\tilde{X}_\tau = (X_\alpha = [X_{\alpha l}, X_{\alpha r}] \mid \alpha \in [0, 1]) \tag{23}$$

The $l_\alpha r_\alpha$ -discretization enables a new definition of a fuzzy random variable \tilde{X}_τ according to Eq. (24) for $i = 1, 2, \dots, n - 1$.

$$\tilde{X}_\tau = \left(\begin{array}{l} X_{\alpha_i} = [X_{\alpha_{i+1}l} - \Delta X_{\alpha_{i}l}; X_{\alpha_{i+1}r} + \Delta X_{\alpha_{i}r}] \mid \alpha_i \in [0, 1); \\ X_{\alpha_n} = [X_{\alpha_{n}l}; X_{\alpha_{n}l} + \Delta X_{\alpha_{n}r}] \mid \alpha_n = 1 \end{array} \right) \quad (24)$$

In this definition the terms $\Delta X_{\alpha_{i}l}$ and $\Delta X_{\alpha_{i}r}$ are correlated random variables and called random $l_\alpha r_\alpha$ -increments. The $l_\alpha r_\alpha$ -discretization enables an alternative, discrete representation of a fuzzy random variable \tilde{X}_τ in the form of a column matrix introduced by Eq. (25), whereby the real-valued random variables $\Delta X_1, \Delta X_2, \dots, \Delta X_{2n}$ are shortened forms of the random $l_\alpha r_\alpha$ -increments $\Delta X_{\alpha_{1}l}, \Delta X_{\alpha_{2}l}, \dots, \Delta X_{\alpha_{1}r}$.

$$\tilde{X}_\tau = \begin{bmatrix} \Delta X_{\alpha_{1}l} \\ \Delta X_{\alpha_{2}l} \\ \vdots \\ \Delta X_{\alpha_{n}l} \\ \Delta X_{\alpha_{n}r} \\ \vdots \\ \Delta X_{\alpha_{2}r} \\ \Delta X_{\alpha_{1}r} \end{bmatrix} = \begin{bmatrix} \Delta X_1 \\ \Delta X_2 \\ \vdots \\ \Delta X_n \\ \Delta X_{n+1} \\ \vdots \\ \Delta X_{2n-1} \\ \Delta X_{2n} \end{bmatrix} \quad (25)$$

According to Eq. (1) a fuzzy random process $(\tilde{X}_\tau)_{\tau \in \mathbf{T}}$ is defined as a family of fuzzy random variables \tilde{X}_τ . For characterization of a fuzzy random process the first and second order moments of the process – like for random processes – are used. The first order moment is a fuzzy variable, that can be represented by $l_\alpha r_\alpha$ -discretization. The $l_\alpha r_\alpha$ -increments of the fuzzy expected value $E[\tilde{X}_\tau] = \tilde{m}_{\tilde{X}_\tau}$ of a fuzzy random process $(\tilde{X}_\tau)_{\tau \in \mathbf{T}}$ are obtained according to Eq. (26).

$$\begin{aligned} E[\tilde{X}_\tau] = \tilde{m}_{\tilde{X}_\tau} &= \begin{bmatrix} \Delta m_{\alpha_{1}l}(\tau) \\ \vdots \\ \Delta m_{\alpha_{n}l}(\tau) \\ \vdots \\ \Delta m_{\alpha_{1}r}(\tau) \end{bmatrix} \\ &= \begin{bmatrix} \int_0^\infty \Delta x_{\alpha_{1}l} f_{\Delta X_{\alpha_{1}l}}(\Delta x_{\alpha_{1}l}, \tau) d\Delta x_{\alpha_{1}l} \\ \vdots \\ \int_{-\infty}^\infty \Delta x_{\alpha_{n}l} f_{\Delta X_{\alpha_{n}l}}(\Delta x_{\alpha_{n}l}, \tau) d\Delta x_{\alpha_{n}l} \\ \vdots \\ \int_0^\infty \Delta x_{\alpha_{1}r} f_{\Delta X_{\alpha_{1}r}}(\Delta x_{\alpha_{1}r}, \tau) d\Delta x_{\alpha_{1}r} \end{bmatrix} \end{aligned} \quad (26)$$

The functions $f_{\Delta X_{\alpha_{i}l}}(\Delta x_{\alpha_{i}l}, \tau)$ and $f_{\Delta X_{\alpha_{i}r}}(\Delta x_{\alpha_{i}r}, \tau)$ ($i = 1, 2, \dots, n$) are probability density functions of the random $l_\alpha r_\alpha$ -increments $\Delta X_{\alpha_{i}l}(\tau)$ and $\Delta X_{\alpha_{i}r}(\tau)$ of the fuzzy random variable \tilde{X}_τ at time point τ .

Linear dependencies between two fuzzy random variables \tilde{X}_{τ_a} and \tilde{X}_{τ_b} of a fuzzy random process at time points τ_a and τ_b are quantified by the $l_\alpha r_\alpha$ -covariance function $l_r K_{\tilde{X}_\tau}(\tau_a, \tau_b)$ according to Eq. (27).

$$l_r K_{\tilde{X}_\tau}(\tau_a, \tau_b) = \begin{bmatrix} k_{\alpha_1 l}^{\alpha_1 l}(\tau_a, \tau_b) & k_{\alpha_1 l}^{\alpha_2 l}(\tau_a, \tau_b) & \cdots & k_{\alpha_1 l}^{\alpha_1 r}(\tau_a, \tau_b) \\ k_{\alpha_2 l}^{\alpha_1 l}(\tau_a, \tau_b) & k_{\alpha_2 l}^{\alpha_2 l}(\tau_a, \tau_b) & \cdots & k_{\alpha_2 l}^{\alpha_1 r}(\tau_a, \tau_b) \\ \vdots & \vdots & \ddots & \vdots \\ k_{\alpha_1 r}^{\alpha_1 l}(\tau_a, \tau_b) & k_{\alpha_1 r}^{\alpha_2 l}(\tau_a, \tau_b) & \cdots & k_{\alpha_1 r}^{\alpha_1 r}(\tau_a, \tau_b) \end{bmatrix} \quad (27)$$

The elements of the $l_\alpha r_\alpha$ -covariance function $l_r K_{\tilde{X}_\tau}(\tau_a, \tau_b)$ are defined by Eq. 28 where $i, j = 1, 2, \dots, n$.

$$k_{\alpha_j r}^{\alpha_i l}(\tau_a, \tau_b) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (\Delta x_{\alpha_i l} - \Delta m_{\alpha_i l}(\tau_a)) (\Delta x_{\alpha_j r} - \Delta m_{\alpha_j r}(\tau_b)) \dots \dots f(\Delta x_{\alpha_i l}, \Delta x_{\alpha_j r}, \tau_a, \tau_b) d\Delta x_{\alpha_i l} d\Delta x_{\alpha_j r} \quad (28)$$

The $l_\alpha r_\alpha$ -variance $l_r Var[\tilde{X}_\tau] = l_r \sigma_{\tilde{X}_\tau}^2$ corresponds to the diagonal elements of the $l_\alpha r_\alpha$ -covariance function $l_r K_{\tilde{X}_\tau}(\tau_a, \tau_b)$ with $\tau_a = \tau_b = \tau$.

A fuzzy random process is stationary if the $l_\alpha r_\alpha$ -covariance function $l_r K_{\tilde{X}_\tau}(\tau_a, \tau_b)$ does not depend on τ_a and τ_b but just on the time lag $\Delta\tau = \tau_a - \tau_b$ and if the fuzzy expected value $E[\tilde{X}_\tau] = \tilde{m}_{\tilde{X}_\tau}$ is constant over time.

In the following a special case of fuzzy random processes is introduced. The known ARMA model is extended to time series with fuzzy data and results the fuzzy-ARMA-model. A fuzzy random process $(\tilde{X}_\tau)_{\tau \in \mathbf{T}}$ ist called fuzzy-ARMA[p, q]-process if it can be described by Eq. (29).

$$\tilde{X}_\tau = \underbrace{\underline{A}_1 \odot \tilde{X}_{\tau-1} \oplus \dots \oplus \underline{A}_p \odot \tilde{X}_{\tau-p}}_{\text{fuzzy-AR-component}} \oplus \tilde{\mathcal{E}}_\tau \ominus \underbrace{\underline{B}_1 \odot \tilde{\mathcal{E}}_{\tau-1} \ominus \dots \ominus \underline{B}_q \odot \tilde{\mathcal{E}}_{\tau-q}}_{\text{fuzzy-MA-component}} \quad (29)$$

The parameters $\underline{A}_1, \dots, \underline{A}_p$ und $\underline{B}_1, \dots, \underline{B}_q$ are real-valued $[2n, 2n]$ matrices. The factors $\tilde{\mathcal{E}}_\tau$ are elements of a fuzzy-white-noise-process $(\tilde{\mathcal{E}}_\tau)_{\tau \in \mathbf{T}}$ at time point τ and therefore fuzzy random variables. A fuzzy-white-noise-process $(\tilde{\mathcal{E}}_\tau)_{\tau \in \mathbf{T}}$ is characterized by Eqs. (30) to (32).

$$E[\tilde{\mathcal{E}}_\tau] = \tilde{m}_{\tilde{\mathcal{E}}_\tau} = \text{constant} \quad \forall \tau \in \mathbf{T} \quad (30)$$

$$l_r Var[\tilde{\mathcal{E}}_\tau] = l_r \sigma_{\tilde{\mathcal{E}}_\tau}^2 = \text{constant} \quad \forall \tau \in \mathbf{T} \quad (31)$$

$$l_r K_{\tilde{\mathcal{E}}_\tau}(\Delta\tau) = \begin{cases} l_r K_{\tilde{\mathcal{E}}_\tau}(0) & \text{for } \Delta\tau = 0 \\ \underline{0} & \text{for } \Delta\tau \neq 0 \end{cases} \quad (32)$$

3. Parameter estimation

Within the scope of modeling fuzzy time series the parameters $\underline{A}_1, \dots, \underline{A}_p$ and $\underline{B}_1, \dots, \underline{B}_q$ of a fuzzy-ARMA[p, q]-process have to be determined so that the empirical time series is a representative realization. Fundamental condition is the demand of non-negativity of the $l_{\alpha}r_{\alpha}$ -increments Δx_j ($j = 1, 2, \dots, n-1, n+2, \dots, 2n$) of all realizations \tilde{x}_{τ} of the fuzzy-ARMA-process.

The first method is based on the postulation that the differences between the empirical and model characteristics (first and second order moments) are minimal. This condition results in the optimization problem given by Eq. (33), in which \underline{P} is a shortened form of the process parameters $\underline{A}_1, \dots, \underline{A}_p$ and $\underline{B}_1, \dots, \underline{B}_q$

$$\sum_{j=1}^{2n} (\Delta \bar{x}_j - \Delta m_j(\underline{P}))^2 + \sum_{\Delta\tau=-\infty}^{\infty} \sum_{k,l=1}^{2n} \left(\hat{k}_{k,l}(\Delta\tau) - k_{k,l}(\Delta\tau, \underline{P}) \right)^2 \stackrel{!}{=} \min \quad (33)$$

The $l_{\alpha}r_{\alpha}$ -increments $\Delta \bar{x}_i$ of the empirical fuzzy mean value $\tilde{\bar{x}}$ are compared with the $l_{\alpha}r_{\alpha}$ -increments Δm_i of the fuzzy expected value $\tilde{m}_{\tilde{x}_{\tau}}$ as well as the elements $\hat{k}_{k,l}(\Delta\tau)$ of the empirical $l_{\alpha}r_{\alpha}$ -covariance function $l_r \hat{K}_{\tilde{x}_{\tau}}(\Delta\tau)$ with the elements $k_{k,l}(\Delta\tau)$ of the theoretical $l_{\alpha}r_{\alpha}$ -covariance function $l_r K_{\tilde{x}_{\tau}}(\Delta\tau)$. The solution of the minimization problem is found with the aid of the modified evolution strategy by (Möller and Beer, 2004). Constraint of the optimization problem is Eq. (22) for all realizations of the process.

The parameter estimation according to Eq. (33) postulates stationary and ergodic fuzzy time series, otherwise it would be obviously futile to estimate the empirical parameters for each point in time. On this account a second approach for parameter estimation of nonstationary fuzzy time series is presented. The aim is to minimize the mean distance \bar{d}_F between optimal one-step forecasts $\tilde{\hat{x}}_{\tau}(\underline{P})$ and the known fuzzy values \tilde{x}_{τ} of the fuzzy time series for $p < \tau \leq N$ according to Eq. (34). Advantage of this method is the fact, that neither stationary nor ergodic fuzzy time series are presupposed. The approach allows the modeling of nonstationary fuzzy time series with the aid of nonstationary fuzzy random processes without empiric parameters.

$$\bar{d}_F(\underline{P}) = \frac{1}{N-p} \sum_{\tau=p+1}^N d_F \left(\tilde{x}_{\tau}; \tilde{\hat{x}}_{\tau}(\underline{P}) \right) \stackrel{!}{=} \min \quad (34)$$

Depending on the demanded process parameters \underline{P} (i. e. $\underline{A}_1, \dots, \underline{A}_p$ and $\underline{B}_1, \dots, \underline{B}_q$) the optimal one-step-forecasts $\tilde{\hat{x}}_{\tau}(\underline{P})$ are computed for each point in time. The distances d_F between $\tilde{\hat{x}}_{\tau}(\underline{P})$ and the fuzzy values \tilde{x}_{τ} of the fuzzy time series are averaged over time. The minimization of the mean distance \bar{d}_F provides unbiased estimators of the process parameters. The calculation of the optimal one-step forecasts $\tilde{\hat{x}}_{\tau}(\underline{P})$ is given in Section 4. The definition of the distance d_F between two fuzzy variables is given as follows.

According to the metrics introduced in (Körner, 1997) the distance $d_F(\tilde{x}; \tilde{y})$ between fuzzy variables \tilde{x} and \tilde{y} is defined as the integral over the Hausdorff distance $d_H(\cdot; \cdot)$ between the α -level sets X_{α} and Y_{α}

of \tilde{x} and \tilde{y} given by Eq. (35).

$$d_F(\tilde{x}; \tilde{y}) = \int_0^1 d_H(X_\alpha; Y_\alpha) d\alpha \tag{35}$$

The Hausdorff distance $d_H(X_\alpha; Y_\alpha)$ between two non-empty compact sets $X_\alpha; Y_\alpha \subseteq \mathbb{R}$ is defined by Eq. (36).

$$d_H(X_\alpha; Y_\alpha) = \max \left\{ \sup_{x \in X_\alpha} \inf_{y \in Y_\alpha} d_E(x; y); \sup_{y \in Y_\alpha} \inf_{x \in X_\alpha} d_E(x; y) \right\} \tag{36}$$

At this $d_E(x; y)$ is the Euclidean distance between two real-valued variables $x, y \in \mathbb{R}$ according to Eq. (37).

$$d_E(x; y) = |x - y| = \sqrt{(x - y)^2} \tag{37}$$

In the following a third approach for estimation of the parameters $\underline{A}_1, \dots, \underline{A}_p$ and $\underline{B}_1, \dots, \underline{B}_q$ of fuzzy-ARMA-processes is presented. This approach also does not presuppose stationary or ergodic fuzzy time series. The concept is to compare the optimal one-step forecasts $\hat{x}_\tau(\underline{P})$ with the known fuzzy values \tilde{x}_τ of the fuzzy time series for $p < \tau \leq N$ according to Eq. (38). The error E is defined as the square deviation of the forecasted $l_\alpha r_\alpha$ -increments $\Delta \hat{x}_j(\tau, \underline{P})$ ($j = 1, 2, \dots, 2n$) to the known $l_\alpha r_\alpha$ -increments $\Delta x_i(\tau)$ of the fuzzy time series. Advantage of this method is that the solution of the minimization problem can be found with the method of gradients.

$$E = \frac{1}{2} \sum_{\tau=1+p}^N \sum_{i=1}^{2n} (\Delta x_i(\tau) - \Delta \hat{x}_i(\tau, \underline{P}))^2 \stackrel{!}{=} \min \tag{38}$$

After initialization the parameter matrices $\underline{A}_1, \dots, \underline{A}_p$ and $\underline{B}_1, \dots, \underline{B}_q$ are improved with the aid of matrices $\Delta \underline{A}_1, \dots, \Delta \underline{A}_p$ and $\Delta \underline{B}_1, \dots, \Delta \underline{B}_q$ according to Eqs. (39) and (40).

$$\underline{A}_r(\text{new}) = \underline{A}_r(\text{old}) + \Delta \underline{A}_r \quad \text{with } r = 1, 2, \dots, p \tag{39}$$

$$\underline{B}_s(\text{new}) = \underline{B}_s(\text{old}) + \Delta \underline{B}_s \quad \text{with } s = 1, 2, \dots, q \tag{40}$$

The matrices $\Delta \underline{A}_1, \dots, \Delta \underline{A}_p$ and $\Delta \underline{B}_1, \dots, \Delta \underline{B}_q$ are built proportional to the partial derivativs of the error E with respect to the belonging parameter matrices according to Eqs. (41) and (42). The factor η ($\eta > 0$) defines the increment.

$$\Delta \underline{A}_r = -\eta \frac{\partial E}{\partial \underline{A}_r} = -\eta \begin{bmatrix} \frac{\partial E}{\partial a_{1,1}(r)} & \cdots & \frac{\partial E}{\partial a_{1,2n}(r)} \\ \vdots & \ddots & \vdots \\ \frac{\partial E}{\partial a_{2n,1}(r)} & \cdots & \frac{\partial E}{\partial a_{2n,2n}(r)} \end{bmatrix} \tag{41}$$

$$\Delta \underline{B}_s = -\eta \frac{\partial E}{\partial \underline{B}_s} = -\eta \begin{bmatrix} \frac{\partial E}{\partial b_{1,1}(s)} & \cdots & \frac{\partial E}{\partial b_{1,2n}(s)} \\ \vdots & \ddots & \vdots \\ \frac{\partial E}{\partial b_{2n,1}(s)} & \cdots & \frac{\partial E}{\partial b_{2n,2n}(s)} \end{bmatrix} \tag{42}$$

The partial derivatives $\frac{\partial E}{\partial a_{u,v}(r)}$ and $\frac{\partial E}{\partial b_{u,v}(s)}$ of the error E with respect to the single elements of the parameter matrices are defined by Eqs. (43) and (44) ($u, v = 1, 2, \dots, 2n$).

$$\frac{\partial E}{\partial a_{u,v}(r)} = \sum_{\tau=1+p}^N (\Delta x_u(\tau) - \Delta \hat{x}_u(\tau, \underline{P})) \Delta x_v(\tau - r) \quad (43)$$

$$\frac{\partial E}{\partial b_{u,v}(s)} = \sum_{\tau=1+p}^N (\Delta x_u(\tau) - \Delta \hat{x}_u(\tau, \underline{P})) \Delta \hat{\varepsilon}_v(\tau - s) \quad (44)$$

The terms $\Delta \hat{\varepsilon}_v(\tau - s)$ are the $l_\alpha r_\alpha$ -increments of the estimated realizations $\hat{\varepsilon}_\tau$ of the fuzzy-white-noise-process $(\tilde{\varepsilon}_\tau)_{\tau \in \mathbf{T}}$. For each point in time $\tau - s \leq p$ the (not ascertainable) realizations $\Delta \hat{\varepsilon}_v(\tau - s)$ are replaced by the estimated expected value $\hat{E}[\Delta \varepsilon_v]$.

$$\frac{\partial E}{\partial b_{u,v}(s)} = \sum_{\tau=1+p}^N (\Delta x_u(\tau) - \Delta \hat{x}_u(\tau, \underline{P})) \hat{E}[\Delta \varepsilon_v] \quad (45)$$

Constraint of the minimization problem is the demand of non-negativity of the estimated $l_\alpha r_\alpha$ -increments $\Delta \hat{\varepsilon}_j(\tau)$ and furthermore the $\hat{\varepsilon}_\tau$ have to satisfy the conditions of a fuzzy-white-noise-process.

4. Forecast strategies

Goal of forecast is the determination of future fuzzy data \tilde{x}_{N+h} ($h = 1, 2, \dots$) following an observed time series with fuzzy data $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N$. Fundamental precondition for this purpose is the assumption and estimation of an underlying fuzzy random process $(\tilde{X}_\tau)_{\tau \in \mathbf{T}}$. Thus the validity of a forecast is associated with the validity of the postulated fuzzy random process.

Therefor a time series with fuzzy data $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N$ is interpreted as a realization of a fuzzy random process $(\tilde{X}_\tau)_{\tau \in \mathbf{T}}$. Consequently forecast is the estimation of fuzzy variables \tilde{x}_{N+h} belonging to the same realization. Analogical the classical time series analysis (Schlittgen and Streitberg, 2001) a forecasted fuzzy data is regarded as a realization $\vec{\tilde{x}}_{N+h}$ of a fuzzy random forecast process $\vec{\tilde{X}}_{N+h} = \vec{\tilde{X}}_{N+h}(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N)$, at which $\vec{\tilde{X}}_{N+h}$ is a random variable depending on the realizations $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N$ of the fuzzy random variables $\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_N$.

The fuzzy random forecast process $(\vec{\tilde{X}}_\tau)_{\tau \in \mathbf{T}}$ of an underlying fuzzy-ARMA[p, q]-process $(\tilde{X}_\tau)_{\tau \in \mathbf{T}}$ is defined according to Eq. (46) where $h = 1, 2, \dots$.

$$\vec{\tilde{X}}_{N+h} = \underline{A}_1 \odot \vec{\tilde{X}}_{N+h-1} \oplus \dots \oplus \underline{A}_p \odot \vec{\tilde{X}}_{N+h-p} \oplus \tilde{\varepsilon}_{N+h} \ominus \underline{B}_1 \odot \tilde{\varepsilon}_{N+h-1} \ominus \dots \ominus \underline{B}_q \odot \tilde{\varepsilon}_{N+h-q} \quad (46)$$

$$\text{with } \vec{\tilde{X}}_{N+h-u} = \begin{cases} \tilde{x}_{N+h-u} & \text{for } N+h-u \leq N \\ \vec{\tilde{X}}_{N+h-u} & \text{for } N+h-u > N \end{cases}, \quad u = 1, 2, \dots, p$$

$$\text{and } \tilde{\varepsilon}_{N+h-v} = \begin{cases} \tilde{\varepsilon}_{N+h-v} & \text{for } N+h-v \leq N \\ \tilde{\varepsilon}_{N+h-v} & \text{for } N+h-v > N \end{cases}, \quad v = 1, 2, \dots, q$$

Thereby for each point in time $\tau = N + h - u \leq N$ the observed fuzzy variables \tilde{x}_{N+h-u} are inserted for \vec{X}_{N+h-u} . For each point in time $\tau = N + h - v \leq N$ the $\tilde{\mathcal{E}}_{N+h-v}$ are replaced by the realizations $\tilde{\varepsilon}_{N+h-v}$ of the fuzzy-white-noise-process $(\tilde{\mathcal{E}}_\tau)_{\tau \in \mathbf{T}}$.

4.1. OPTIMAL FORECAST

The optimal forecast $\overset{\circ}{\tilde{x}}_{N+h}$ to the time point $\tau = N + h$ is defined as the conditional fuzzy expected value according to Eq. (47).

$$\overset{\circ}{\tilde{x}}_{N+h}(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N) = E[\vec{X}_{N+h} | \tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N] = E[\vec{X}_{N+h}] \quad (47)$$

In the following the optimal forecast of a fuzzy-ARMA-process, which is the underlying fuzzy random process of an observed sequence of fuzzy data, is introduced.

The optimal one-step forecast of a fuzzy-ARMA[p, q]-process $(\vec{X}_\tau)_{\tau \in \mathbf{T}}$ according to Eq. (29) is defined by Eq. (48).

$$\begin{aligned} \overset{\circ}{\tilde{x}}_{N+1} &= \underline{A}_1 \odot \tilde{x}_N \oplus \dots \oplus \underline{A}_p \odot \tilde{x}_{N+1-p} \oplus E[\tilde{\mathcal{E}}_\tau] \ominus \\ &\quad \underline{B}_1 \odot \tilde{\varepsilon}_N \ominus \dots \ominus \underline{B}_q \odot \tilde{\varepsilon}_{N+1-q} \end{aligned} \quad (48)$$

The optimal h -step forecast is obtained by recursive use of the optimal one-step forecast according to Eq. (48). Consequently the forecasted fuzzy data converge with increasing forecast step h on the fuzzy expected value $E[\vec{X}_\tau]$. The optimal h -step forecast of a fuzzy-ARMA[p, q]-process $(\vec{X}_\tau)_{\tau \in \mathbf{T}}$ is defined by Eq. (49).

$$\begin{aligned} \overset{\circ}{\tilde{x}}_{N+h} &= \underline{A}_1 \odot \tilde{x}_{N+h-1} \oplus \dots \oplus \underline{A}_p \odot \tilde{x}_{N+h-p} \oplus E[\tilde{\mathcal{E}}_\tau] \ominus \\ &\quad \underline{B}_1 \odot \tilde{\varepsilon}_{N+h-1} \ominus \dots \ominus \underline{B}_q \odot \tilde{\varepsilon}_{N+h-q} \end{aligned} \quad (49)$$

$$\text{with } \tilde{x}_{N+h-u} = \begin{cases} \tilde{x}_{N+h-u} & \text{für } N+h-u \leq N \\ \overset{\circ}{\tilde{x}}_{N+h-u} & \text{für } N+h-u > N \end{cases}, \quad u = 1, 2, \dots, p$$

$$\text{and } \tilde{\varepsilon}_{N+h-v} = \begin{cases} \tilde{\varepsilon}_{N+h-v} & \text{für } N+h-v \leq N \\ E[\tilde{\mathcal{E}}_\tau] & \text{für } N+h-v > N \end{cases}, \quad v = 1, 2, \dots, q$$

Thereby for each point in time $\tau = N + h - u \leq N$ the optimal forecasts $\overset{\circ}{\tilde{x}}_{N+h-u}$ are inserted for \tilde{x}_{N+h-u} . For each point in time $\tau = N + h - v > N$ the $\tilde{\varepsilon}_{N+h-v}$ are replaced by the fuzzy expected value $E[\tilde{\mathcal{E}}_\tau]$ of the fuzzy-white-noise-process $(\tilde{\mathcal{E}}_\tau)_{\tau \in \mathbf{T}}$.

4.2. FUZZY FORECAST INTERVALS

A fuzzy interval \tilde{x}_I is referred to as fuzzy forecast interval \tilde{x}_{N+h}^κ , if realizations \vec{x}_{N+h} of the fuzzy random forecast process $(\vec{X}_\tau)_{\tau \in \mathbf{T}}$ are contained in \tilde{x}_I with the probability κ . Fuzzy forecast intervals \tilde{x}_{N+h}^κ at time point $\tau = N + h$ of a fuzzy time series $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N$ can be estimated with the aid of monte-carlo-simulation of the fuzzy random forecast process $(\vec{X}_\tau)_{\tau \in \mathbf{T}}$. The monte-carlo-simulation of the fuzzy random forecast process $(\vec{X}_\tau)_{\tau \in \mathbf{T}}$ (with an underlying fuzzy-ARMA[p, q]-process $(\vec{X}_\tau)_{\tau \in \mathbf{T}}$) is obtained by the recursive

procedure according to Eq. (50). In the first step a realization \vec{x}_{N+1} at time point $\tau = N + 1$ of the fuzzy random forecast process $(\vec{X}_\tau)_{\tau \in \mathbf{T}}$ is simulated. The realization \vec{x}_{N+1} of the fuzzy random variable \vec{X}_{N+1} depends on the realization $\tilde{\varepsilon}_{N+1}$ of the fuzzy-white-noise-variable $\tilde{\mathcal{E}}_{N+1}$. The fuzzy variables \tilde{x}_τ and $\tilde{\varepsilon}_\tau$ at time points $\tau \leq N$ are given by the time series.

$$\begin{aligned} \vec{X}_{N+1} &= \underline{A}_1 \odot \tilde{x}_N \oplus \dots \oplus \underline{A}_p \odot \tilde{x}_{N+1-p} \oplus \tilde{\mathcal{E}}_{N+1} \ominus \\ &\quad \underline{B}_1 \odot \tilde{\varepsilon}_N \ominus \dots \ominus \underline{B}_q \odot \tilde{\varepsilon}_{N+1-q} \end{aligned} \quad (50)$$

The fuzzy variable \vec{x}_{N+1} is obtained by monte-carlo-simulation of a realization $\tilde{\varepsilon}_{N+1}$. By use of the obtained fuzzy variable \vec{x}_{N+1} and monte-carlo-simulation of a realization $\tilde{\varepsilon}_{N+2}$ the fuzzy variable \vec{x}_{N+2} is obtained in the next step. A successive computing at time points $\tau = N + 1, N + 2, \dots$ results one potential future gradient of the fuzzy time series $(\tilde{x}_\tau)_{\tau \in \mathbf{T}}$. By repetition of this procedure a number of potential future realizations is obtained.

With the aid of s simulated potential future realizations of a fuzzy time series $(\tilde{x}_\tau)_{\tau \in \mathbf{T}}$ fuzzy forecast intervals \tilde{x}_{N+h}^κ can be estimated as follows. The interval boundaries $\tilde{x}_{\alpha_i l}(N+h)$ and $\tilde{x}_{\alpha_i r}(N+h)$ of the α -level sets $\vec{X}_{\alpha_i}(N+h)$ of the s simulated fuzzy variables \vec{x}_{N+h} are arranged according to size and subscripted according to Eq. (51).

$$\begin{aligned} \tilde{x}_{\alpha_i l}^1(N+h) &\leq \tilde{x}_{\alpha_i l}^2(N+h) \leq \dots \leq \tilde{x}_{\alpha_i l}^s(N+h) \\ \tilde{x}_{\alpha_i r}^1(N+h) &\leq \tilde{x}_{\alpha_i r}^2(N+h) \leq \dots \leq \tilde{x}_{\alpha_i r}^s(N+h) \end{aligned} \quad (51)$$

The interval boundaries $x_{\alpha_i l}^\kappa(N+h)$ and $x_{\alpha_i r}^\kappa(N+h)$ of the α -level sets $X_{\alpha_i}^\kappa(N+h)$ of a fuzzy forecast interval \tilde{x}_{N+h}^κ at time point $\tau = N+h$ can be estimated according to Eq. (52) for a confidence level κ . Eq. (52) is valid for an even number of s .

$$\begin{aligned} x_{\alpha_i l}^\kappa(N+h) &= \begin{cases} \leq \tilde{x}_{\alpha_i l}^1(N+h) & \text{für } a = 0 \\ \tilde{x}_{\alpha_i l}^a(N+h) & \text{für } 0 < a \leq \frac{s}{2} \end{cases} \\ &\quad \text{with } a = \text{int} \left[s \cdot \left(\frac{1}{2} - \frac{\kappa}{2} \right) \right] \\ x_{\alpha_i r}^\kappa(N+h) &= \begin{cases} \tilde{x}_{\alpha_i r}^{b+1}(N+h) & \text{für } \frac{s}{2} \leq b < s \\ \geq \tilde{x}_{\alpha_i r}^b(N+h) & \text{für } b = s \end{cases} \\ &\quad \text{with } b = \frac{s}{2} + \text{int} \left[s \cdot \left(\frac{\kappa}{2} \right) \right] \end{aligned} \quad (52)$$

The interval boundaries $x_{\alpha_i l}^\kappa(N+h)$ and $x_{\alpha_i r}^\kappa(N+h)$ of the α -level sets $X_{\alpha_i}^\kappa(N+h)$ according to Eq. (52) correspond with the lower and upper quantile of the empiric distribution of the interval boundaries. Therewith future realizations \vec{x}_{N+h} of a fuzzy time series $(\tilde{x}_\tau)_{\tau \in \mathbf{T}}$ are contained in the fuzzy forecast interval \tilde{x}_{N+h}^κ with a probability κ .

4.3. FUZZY RANDOM FORECAST

The forecast strategies presented in sections 4.1 and 4.2 provide concrete fuzzy variables and fuzzy intervals. In the following a fuzzy random forecast is presented, which provides estimators for future fuzzy random variables \vec{X}_τ of the fuzzy random forecast process $(\vec{X}_\tau)_{\tau \in \mathbf{T}}$ at time points $\tau = N + h$. Therewith statements about the probability of future fuzzy variables are feasible.

By monte-carlo-simulation of s potential future realizations $(\vec{x}_{N+h} | \tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N)$ of the fuzzy time series $(\tilde{x}_\tau)_{\tau \in \mathbf{T}}$ the fuzzy random variable \vec{X}_{N+h} can be estimated. For characterization of \vec{X}_{N+h} the first and second order moments of the fuzzy random variable \vec{X}_{N+h} are used. With the aid of the simulated fuzzy variables \vec{x}_{N+h}^c ($c = 1, 2, \dots, s$) the estimator of the fuzzy expected value $E[\vec{X}_{N+h}]$ is obtained as the fuzzy mean value \tilde{x}_{N+h} at time point $\tau = N + h$ according to Eq. (53).

$$\hat{E}[\vec{X}_{N+h}] = \tilde{x}_{N+h} = \frac{1}{s} \bigoplus_{c=1}^s \vec{x}_{N+h}^c \tag{53}$$

The fuzzy expected value $E[\vec{X}_{N+h}]$ is identical with the optimal forecast \hat{x}_{N+h} . The $l_\alpha r_\alpha$ -subtraction $\hat{E}[\vec{X}_{N+h}] \ominus E[\vec{X}_{N+h}]$ (and $\tilde{x}_{N+h} \ominus \hat{x}_{N+h}$ respectively) is a measure for the performance of the simulation. With increasing number of s the norm of the empiric $l_\alpha r_\alpha$ -variance ${}_l r Var$ of the $l_\alpha r_\alpha$ -subtraction $\vec{x}_{N+h}(c) \ominus \hat{x}_{N+h}$ ($c = 1, 2, \dots, s$) according to Eq. (54) converges on zero. Consequently, with increasing number s of realizations \vec{x}_{N+h}^c ($c = 1, 2, \dots, s$) the simulation represents the characteristics of the fuzzy random forecast process $(\vec{X}_\tau)_{\tau \in \mathbf{T}}$ superiorly.

$$\lim_{s \rightarrow \infty} |{}_l r Var [\vec{x}_{N+h}(c) \ominus \hat{x}_{N+h} | c = 1, 2, \dots, s]| = 0 \tag{54}$$

By defining a maximal value η for the norm of the empiric $l_\alpha r_\alpha$ -variance ${}_l r Var$ according to Eq. (55) a minimum number s_m of realizations can be obtained. In other words, for a wanted performance η of the simulation a number of s_m realizations is needed.

$$|{}_l r Var [\vec{x}_{N+h}(c) \ominus \hat{x}_{N+h} | c = 1, 2, \dots, s_m]| \leq \eta \tag{55}$$

The elements of the $l_\alpha r_\alpha$ -covariance function ${}_l r K_{\vec{X}_\tau}(\tau_a, \tau_b)$ of the fuzzy random forecast process $(\vec{X}_\tau)_{\tau \in \mathbf{T}}$ are defined by Eq. 56 where $i, j = 1, 2, \dots, n$.

$$\hat{k}_{\alpha_j r}^{\alpha_i l}(\tau_a, \tau_b) = \frac{1}{s} \sum_{c=1}^s [(\Delta \vec{x}_{\alpha_i l}^c(\tau_a) - \Delta \hat{x}_{\alpha_i l}(\tau_a)) \tag{56}$$

$$(\Delta \vec{x}_{\alpha_j r}^c(\tau_b) - \Delta \hat{x}_{\alpha_j r}(\tau_b))] \tag{57}$$

Thereby the terms $\Delta \vec{x}_{\alpha_i l}^c(\tau)$ are the $l_\alpha r_\alpha$ -increments of the simulated fuzzy variables \vec{x}_τ^c at time point $\tau > N$ and the terms $\Delta \hat{x}_{\alpha_i l}(\tau)$ are the $l_\alpha r_\alpha$ -increments of the optimal forecast \hat{x}_τ . The estimator for the $l_\alpha r_\alpha$ -variance ${}_l r Var[\vec{X}_\tau] = {}_l r \sigma_{\vec{X}_\tau}^2$ corresponds the diagonal elements of the estimated $l_\alpha r_\alpha$ -covariance function ${}_l r \hat{K}_{\vec{X}_\tau}(\tau_a, \tau_b)$ with $\tau_a = \tau_b = \tau$.

5. Examples

5.1. EXAMPLE 1

Analysis of time series with fuzzy data is demonstrated by way of heavy goods vehicle traffic over the bridge *Blaues Wunder* in Dresden. Since October 1999 a weight-in-motion measuring point records the entire traffic over the bridge. The data are kindly provided by the highway board department of Dresden. For the projected analysis the measured data for heavy goods vehicle are revised of weekend and holiday data and thereafter fuzzified based on the histograms of each weekday. The time series thus obtained is assumed to be stationary. June 2002 to April 2003 is considered as time period analyzed. An section of the time series is shown in Fig. 4.

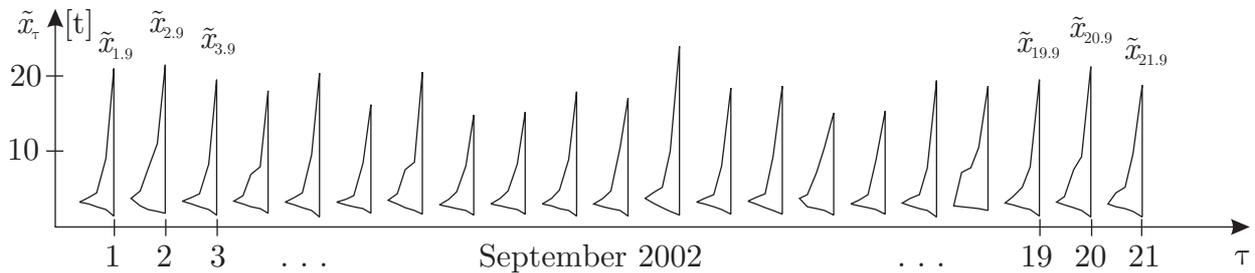


Figure 4. Time series with fuzzy data of heavy goods vehicle traffic over the bridge *Blaues Wunder* in Dresden (section)

The $l_\alpha r_\alpha$ -discretization is applied to α -levels $\alpha_1 = 0.0, \alpha_2 = 0.25, \alpha_3 = 0.5, \alpha_4 = 0.75$ and $\alpha_5 = 1.0$. Fig. 5 shows exemplarily the plot of $l_\alpha r_\alpha$ -increments $\Delta x_{\alpha_i l}$ and $\Delta x_{\alpha_i r}$.

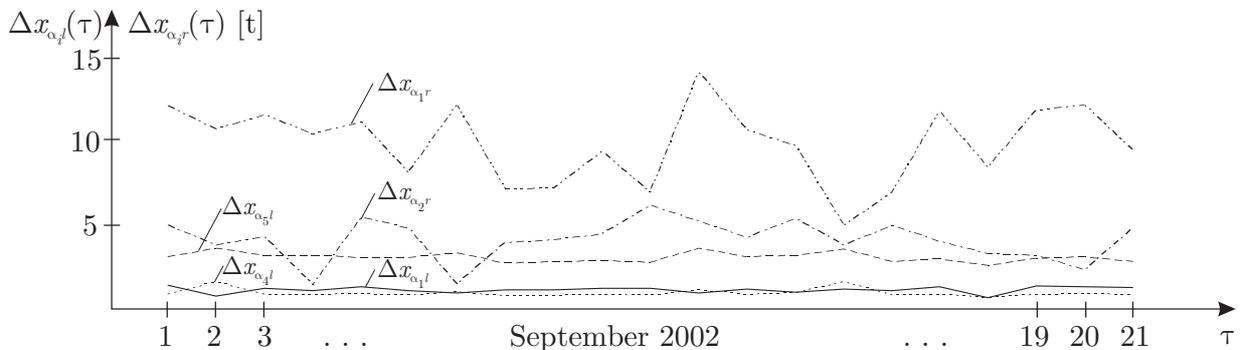


Figure 5. Plot of the $l_\alpha r_\alpha$ -increments

Modeling of this time series with fuzzy data bases on a fuzzy-ARMA [10,0]-process. For estimation of the parameters $\underline{A}_1, \underline{A}_2, \dots, \underline{A}_{10}$ the minimization problem according to Eq. (33) is solved. On this account the empirical fuzzy mean value \tilde{x} (see Fig. 6), the empirical $l_\alpha r_\alpha$ -covariance function, and thus the

empirical $l_{\alpha r \alpha}$ -variance are estimated from the time series under assumption of ergodicity. Consequently, it is demanded that the differences between the empirical and model characteristics (first and second order moments) are minimal.

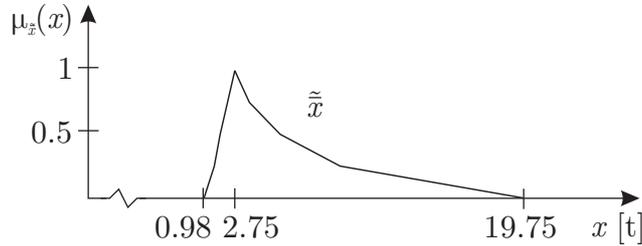


Figure 6. Fuzzy mean value

The solution of the optimization problem yields the following estimators: the process parameters $\underline{A}_1, \underline{A}_2, \dots, \underline{A}_{10}$ as well as the fuzzy expected value $E[\tilde{\mathcal{E}}_\tau]$, the $l_{\alpha r \alpha}$ -variance $l_r Var[\tilde{\mathcal{E}}_\tau]$ and the $l_{\alpha r \alpha}$ -covariance function $l_r K_{\tilde{\mathcal{E}}_\tau}(\Delta\tau)$ as parameters of the fuzzy white noise process $(\tilde{\mathcal{E}}_\tau)_{\tau \in \mathbf{T}}$. With the aid of the estimated underlying fuzzy-ARMA[10,0]-process forecast of the following fuzzy data in May 2003 is feasible. The optimal 1-step-forecast of the fuzzy-ARMA[10,0]-process is given by Eq. (58).

$$\hat{\tilde{x}}_{N+1} = \underline{A}_1 \odot \tilde{x}_N \oplus \dots \oplus \underline{A}_{10} \odot \tilde{x}_{N-9} \oplus E[\tilde{\mathcal{E}}_\tau] \tag{58}$$

A repeated application of Eq. (58) results in the h-step forecast. The forecasted fuzzy data converge on the fuzzy expected value. The resulted fuzzy data in comparison to the real measured data are shown in Fig. 7. The forecast refers to the data for heavy goods vehicle on 12 weekdays in May 2003. The optimal forecasts differ somewhat from the real measured data. Reason for it is, that the analysed fuzzy time series is characterized by a comparatively minor random influence

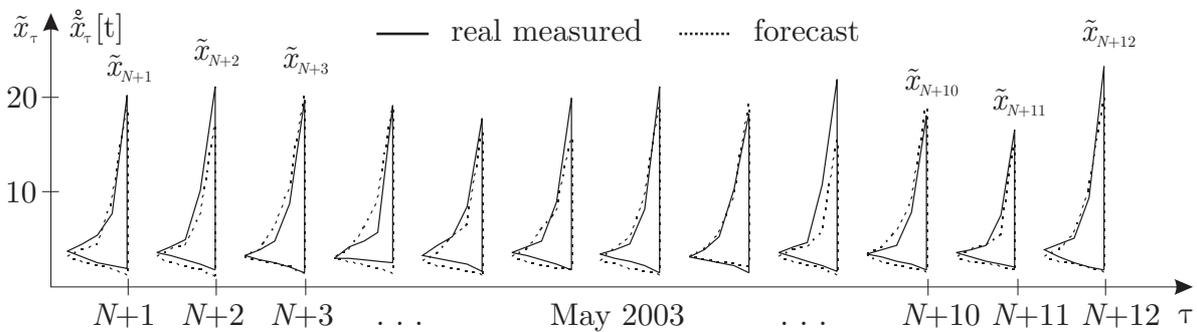


Figure 7. Optimal forecasts in comparison with the measured fuzzy time series

5.2. EXAMPLE 2

Analysis and forecast of nonstationary fuzzy time series is demonstrated by example of extensometer measurements. The given series was measured from 1999 to 2002 and is kindly provided by the EIBS GmbH, Dresden. Table I shows a short section of the measured time series over five days. Three different measuring data exist at each time point. Instead of computing the mean value the measuring difference is considered as uncertainty and modeled as fuzzy variable. The $l_\alpha r_\alpha$ -discretization is realized for $\alpha_1 = 0$ and $\alpha_2 = 1$. Fig. 8 shows the plot of the fuzzy time series.

Table I. Section of extensometer measurements

date	1st meas. [mm]	2nd meas. [mm]	3rd meas. [mm]	mean value [mm]
⋮	⋮	⋮	⋮	⋮
30.05.2000	22.51	22.50	22.52	22.510
27.06.2000	22.50	22.52	22.53	22.517
27.07.2000	22.40	22.40	22.41	22.403
30.08.2000	22.35	22.36	22.35	22,353
27.09.2000	21.72	21.80	21.77	21.763
⋮	⋮	⋮	⋮	⋮

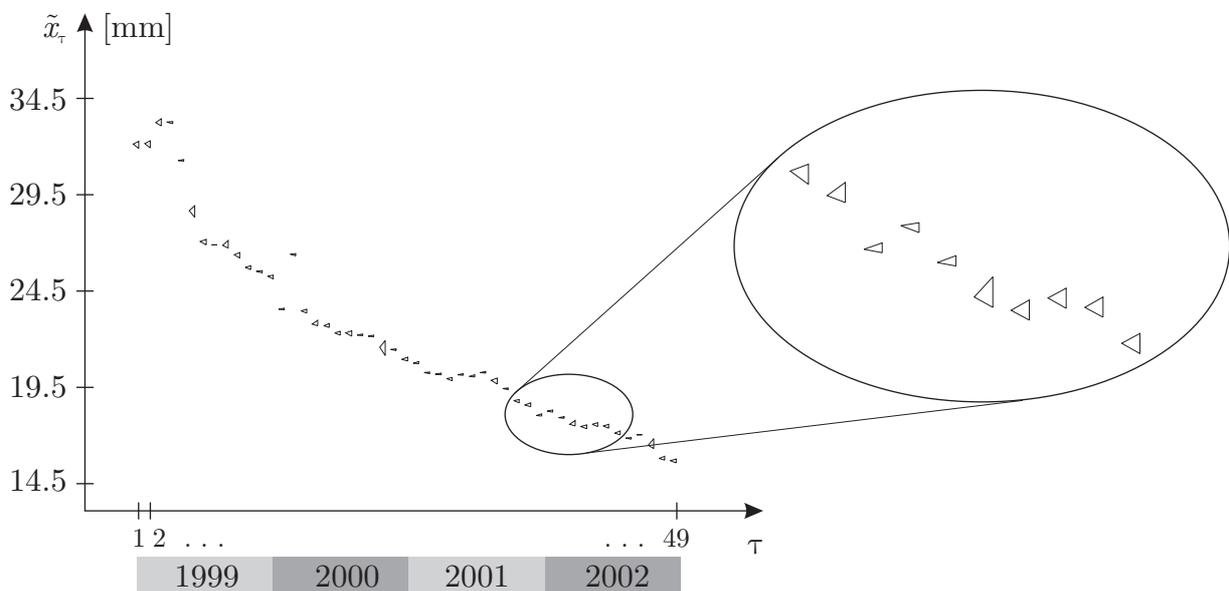


Figure 8. Time series with fuzzified extensometer measurements

The modeling of this fuzzy time series obviously requires a nonstationary fuzzy stochastic process model. The fuzzy time series is specified as nonstationary fuzzy-ARMA-process of the order $p = 10$ and $q = 3$. The estimation of the parameters $\underline{A}_1, \underline{A}_2, \dots, \underline{A}_{10}$ and $\underline{B}_1, \underline{B}_2, \underline{B}_3$ is done with the aid of the optimization problem given by Eq. (34). This procedure yields optimal 1-step-forecasts with a minimized distance to the empirical fuzzy variables in the considered space of time. The result is shown in Fig. 9.

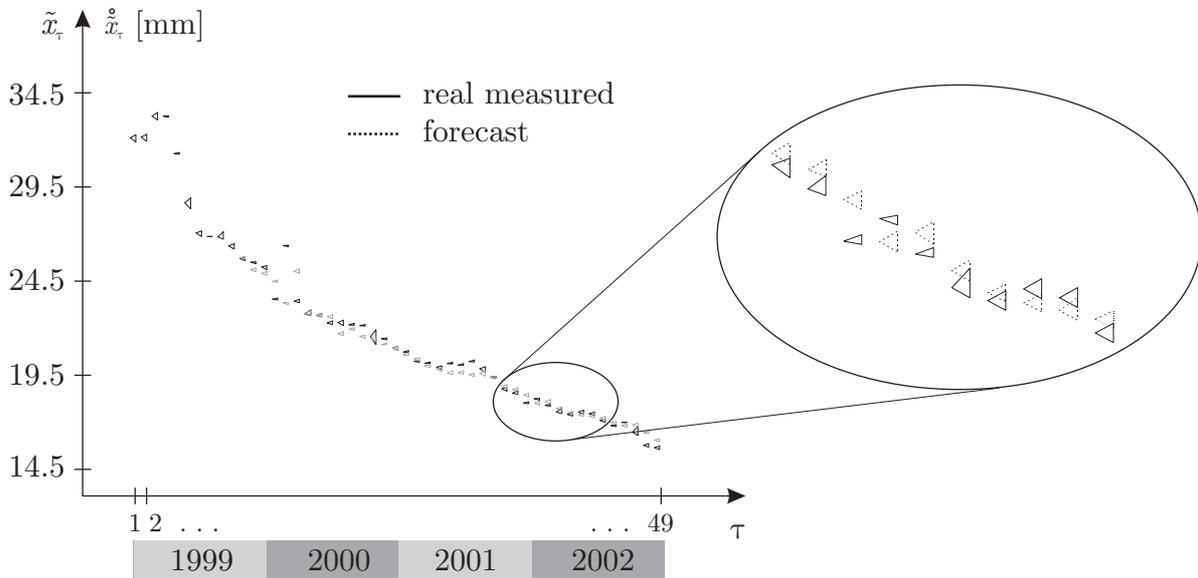


Figure 9. Optimal 1-step-forecasts of the fuzzy time series

For parameter estimation of the underlying fuzzy-ARMA[10,3]-process was based on the empirical fuzzy time series in the space of time from December 1998 until November 2002. The estimated fuzzy stochastic process enables the forecast of future settlements. The optimal long running forecast for the following 37 month is shown in Fig. 10. This is equivalent to a forecasting horizon of 3 years.

With the aid of the fuzzy-ARMA[10,3]-process the estimation of fuzzy forecast intervals is feasible. The fuzzy forecast intervals specify domains in which future realizations are contained with a confidence level κ . Exemplarily the fuzzy forecast intervals with the confidence level 0.95 are shown in Fig. 11.

6. Conclusions

In this paper a new approach for description and modeling of time series with uncertain data is presented. Uncertain data at equally spaced discrete time points are modeled as time series with fuzzy data. In this context a new method for representation of fuzzy data is presented. The $l_{\alpha}r_{\alpha}$ -discretization enables a new statistical evaluation of fuzzy samples. At this the new fuzzy-ARMA-process is introduced. This process enables analysis and forecast of suitable time series with fuzzy data. The fuzzy-ARMA-process is successfully applied to a time series of heavy goods vehicle traffic data and a time series with uncertain extensometer measurements.

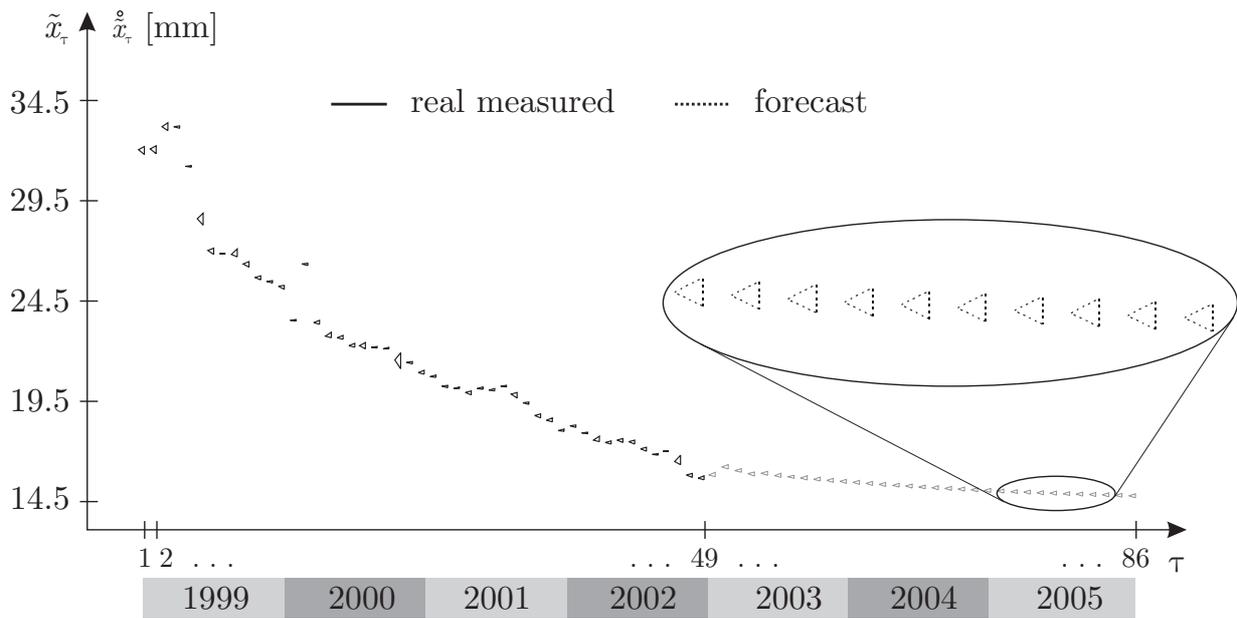


Figure 10. Optimal long running forecast of the fuzzy time series

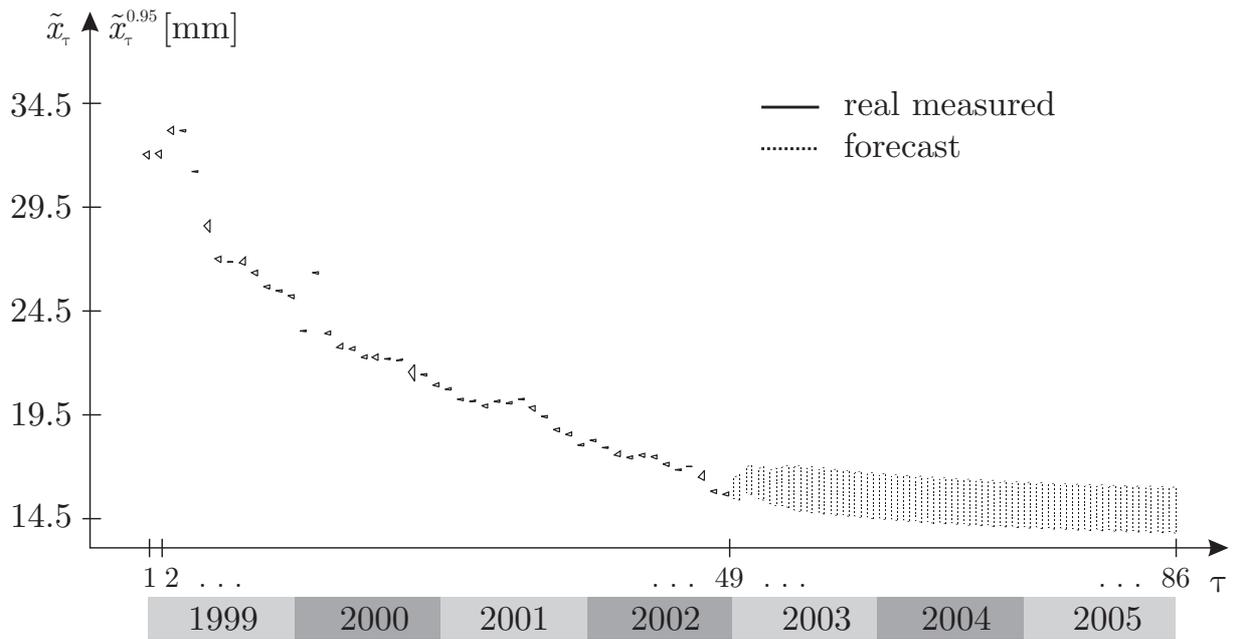


Figure 11. Fuzzy forecast intervals for a confidence level 0.95

Acknowledgements

The authors gratefully acknowledge the support of the German Research Foundation (DFG).

References

- Hareter, D.: *Zeitreihenanalyse mit unscharfen Daten*. Dissertation, TU Wien, Wien, 2003.
- Körner, R.: *Linear Models with Random Fuzzy Variables*. Dissertation, TU Bergakademie Freiberg, Freiberg, 1997.
- Möller, B. and Beer, M.: *Fuzzy Randomness - Uncertainty Models in Civil Engineering and Computational Mechanics*. Springer, Berlin, Heidelberg, New York, 2004.
- Möller, B., Beer, M. and Reuter, U.: Theoretical Basics of Fuzzy Randomness – Application to time series with fuzzy data. In: Augusti, G., Schueller, G.I., and Ciampoli M. (eds.): *Proceedings of the ninth international conference on structural safety and reliability*, Millpress, Rotterdam, CD-ROM, paper 701, pp. 1701–1707, 2005.
- Schlittgen, R. and Streitberg, B.: *Zeitreihenanalyse*. Oldenbourg, München, Wien, 2001.

Boundary Element Analysis of Systems Using Interval Methods

¹B.F. Zalewski, ¹R.L. Mullen and ²R.L. Muhanna

¹ *Department of Civil Engineering
Case Western Reserve University
Cleveland, OH 44106
bxz10@case.edu; rlm@case.edu*

² *Department of Civil and Environmental Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332
rafi.muhanna@gtsav.gatech.edu*

Abstract: In engineering, most governing partial differential equations of physical systems are solved using finite element or finite difference methods. Applications of interval methods have been explored in finite element analysis to model systems with uncertainty in parameters and to account for the impact truncation error on solutions. An alternative to finite element analysis is boundary element method. The boundary element method uses singular functions to reduce the dimension of the domain by transforming the domain variables to variables on the boundaries. In this work, new methods using interval variables are developed to enhance boundary element method for considering impreciseness such as uncertain boundary conditions, truncation errors, integration errors and discretization errors. Exemplars are presented to illustrate the effectiveness and potential of interval approach in boundary element method analysis.

Keyword: boundary element method, interval analysis, truncation error, discretization errors

1. Introduction

Boundary element analysis (BEA) is a method for obtaining approximate solution of partial differential equations. This method requires less meshing than finite element analysis and thus, it is comparatively faster in generating or refining the mesh. BEA is performed by transformation of the domain variables to the variables on the boundaries of the system. The domain transformation is constructed using singular solutions of the governing partial differential equation. Though extensions to non-linear problems can be of the domain, straight forward BEA

© 2006 authors. Printed in USA

formulations apply to linear problems. Then the transformed boundary integral equations are solved using collocation methods, i.e., source points are located sequentially at all boundary nodes that map the domain variables such that they coincide to their values at the nodes.

Errors in BEA can be classified into the following sources:

- 1) Uncertainty in the boundary conditions
- 2) Uncertainty in parameters of the system
- 3) Errors in integration
- 4) Errors in the solution of the resulting linear system of equations
- 5) Discretization errors.

In this paper we will address the use of concepts from interval methods to address all of the above except for the issue of uncertainty in system parameters. If system parameters such as material properties change, one may need to develop a new analytical singular solution. When the boundary conditions are uncertain, the use of intervals to bound this uncertainty leads to a system of linear equations with an interval right hand side. The incorporation of this source of uncertainty can be treated in a manner similar to that used in finite element analysis (Mullen and Muhanna, 1999).

Most boundary element programs use numerical quadrature to integrate terms in the resulting system of linear equations. In some problems, one can perform the integration explicitly; other BEA may require integration that may not be generally performed explicitly. One procedure to overcome this issue is to expand the mapping functions as a series, such as Taylor series expansion. This expansion, in fact, is an approximation of the function in the form of a polynomial, using the function's derivatives evaluated at a point inside the domain of the function. The truncation error is considered as an interval variable obtained from the maximum Taylor series expansion remainder. Then, the BEA is performed in the presence of variation in the corresponding linear system of equations. Based on present error bounds, the enclosure on the bounds of the results is quantified. This procedure can lead to interval bounds on errors due to integration.

Truncation errors in the solution of the resulting system of linear equations can be included in BEA using conventional interval methods for linear equations (Alefeld 1983, Gay 1982, Hansen 1965, Jansson 1991, Moore 1979, Neumaier 1987, 1988, 1990, Rump 1990, Sunaga 1958).

Finally we explore the bounding of discretization errors using local functions that are bounded by interval values. An example for a two dimensional Laplace equation using constant elements is presented. Sharp bounds require a method for solving parametrically constrained systems of linear equations.

2. Boundary Element Analysis of Laplace Equation

2.1. BEA FORMULATION FOR LAPLACE EQUATION

The theory of boundary elements is discussed in the books by Brebbia 1992 and Hartmann 1889. In the following, we will review a two dimension boundary element formulation for Laplace equation.

The Laplace equation is:

$$\begin{aligned} \nabla^2 u &= 0 & \text{in } \Omega \\ u &= \hat{u} & \text{on } \Gamma_1 \\ \frac{\partial u}{\partial n} &= q = \hat{q} & \text{on } \Gamma_2 \end{aligned} \quad (1)$$

where (Ω) is the domain of the system, (Γ) is the boundary of the system and (\hat{u}) and (\hat{q}) are the values at the boundary.

To minimize the error introduced as the exact solution of (u) and (q) is approximated, orthogonalization of Eq. (1) with respect to a test function (w) is performed:

$$\int_{\Omega} \nabla^2 u w d\Omega = \int_{\Gamma_2} (q - \hat{q}) w d\Gamma - \int_{\Gamma_1} (u - \hat{u}) \frac{\partial w}{\partial n} d\Gamma \quad (2)$$

Twice integrating by parts on the left side of Eq. (2) and considering $u^* = w$ and $q^* = \partial u^* / \partial n$ yields:

$$\int_{\Omega} \nabla^2 u^* u d\Omega = - \int_{\Gamma_2} \hat{q} u^* d\Gamma - \int_{\Gamma_1} q u^* d\Gamma + \int_{\Gamma_2} u q^* d\Gamma + \int_{\Gamma_1} \hat{u} q^* d\Gamma \quad (3)$$

or:

$$u(\xi) + \int_{\Gamma_2} u q^* d\Gamma + \int_{\Gamma_1} \hat{u} q^* d\Gamma = \int_{\Gamma_2} \hat{q} u^* d\Gamma + \int_{\Gamma_1} q u^* d\Gamma, \quad \xi \in \Omega \quad (4)$$

where (ξ) is a source point.

The term (u^*) is the fundamental solution satisfying Laplace equation that represents a field generated by a singular source at some point (ξ) . Hence, at a field point (x) , (u^*) must satisfy:

$$\nabla^2 u^* + \delta(x - \xi) = 0 \quad (5)$$

The solution to Eq. (5) for a two-dimensional isotropic domain is:

$$u^* = -\frac{1}{2\pi} \ln(r) \quad (6)$$

$$q^* = -\frac{1}{2\pi r^2} (x - \xi) \cdot n \quad (7)$$

where $r = |x - \xi|$ is the distance between the source point (ξ) and any point of interest (x) . Allowing the boundary to be along (x) and rewriting Eq. (4) before the application of boundary conditions:

$$u(\xi) + \int_{\Gamma_x} q^*(x, \xi) u(x) d\Gamma_x = \int_{\Gamma_x} u^*(x, \xi) q(x) d\Gamma_x, \quad \xi \in \Omega \quad (8)$$

Integrating Eq. (8) such that the source point, (ξ) , is included on the circular boundary of radius (ε) , as $\varepsilon \rightarrow 0$, results in the left side integral vanishing. For constant elements the right side integral results in $-\frac{1}{2}u(\xi)$. Thus, Eq. (8) can be rewritten as:

$$\frac{1}{2}u(\xi) + \int_{\Gamma_x} q^*(x, \xi) u(x) d\Gamma_x = \int_{\Gamma_x} u^*(x, \xi) q(x) d\Gamma_x, \quad \xi \in \Omega \quad (9)$$

2.2. CONSTANT ELEMENT BOUNDARY DISCRETIZATION

Any boundary Γ can be discretized into boundary elements Γ_i consisting of nodes at which a value of either (u) or (q) is known and assumed polynomial shape functions between nodes. In this work, only boundary elements with constant shape functions are used.

These elements contain one node per element, leading to the following discretization:

$$u(x) = u_i \Phi(x) \quad (10)$$

$$q(x) = q_i \Phi(x) \quad (11)$$

where $\{u_i\}$ and $\{q_i\}$ are the vectors of nodal values of (u) and (q) , respectively, at node (i) and $\Phi(x)$ is the vector of constant shape functions. The discretized Eq. (9) is written as:

$$\frac{1}{2}u_i + \sum_{Elements} u_i \int_{\Gamma_x} q^*(x, \xi) \Phi(x) d\Gamma_x = \sum_{Elements} q_i \int_{\Gamma_x} u^*(x, \xi) \Phi(x) d\Gamma_x \quad (12)$$

Eq. (12) is written in a matrix form:

$$Hu = Gq \quad (13)$$

where matrix $[H]$ satisfies the rigid body motion. Eq. (13) is rearranged and solved as:

$$Ax = f \quad (14)$$

The terms of $[H]$ and $[G]$ matrices can either be determined explicitly or are computed numerically, by numerical integration using Taylor series expansion.

3. Taylor Series Expansion

A function can be expressed as a polynomial in terms of its derivatives at some point (a) using Taylor series expansion [Taylor, 1715]:

$$f(x) = \frac{f(a)}{0!} + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots + \frac{f^m(a)}{m!}(x-a)^m \quad (15)$$

where $m \rightarrow \infty$.

If the function has a finite amount of nonzero derivatives, it can be integrated exactly:

$$\int_x f(x)dx = \int_x \left[\frac{f(a)}{0!} + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots + \frac{f^n(a)}{n!}(x-a)^n \right] dx \quad (16)$$

where (n) corresponds to the last nonzero derivative of the function. Since a function $f(x)$ is represented by a polynomial, its integration can be performed:

$$\int_x f(x)dx = \left[f(a)x + \frac{f'(a)}{2}(x-a)^2 + \frac{f''(a)}{6}(x-a)^3 + \dots + \frac{f^n(a)}{(n+1)!}(x-a)^{n+1} \right]_x \quad (17)$$

However, if the function has an infinite amount of nonzero derivatives, integration of the Taylor Series introduces truncation errors, since not all terms in the series can be accounted for.

4. Error Analysis on Taylor Series Expansion

A function can also be expressed using Taylor series expansion with remainder as:

$$f(x) = \frac{f(a)}{0!} + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots + \frac{f^{n-1}(a)}{(n-1)!}(x-a)^{n-1} + R_n \quad (18)$$

where (n) corresponds to the (n^{th}) derivative of the function and R_n is the series remainder as:

$$R_n = \frac{f^n(\zeta)(x-a)^n}{n!} \quad a < \zeta < x \quad (19)$$

Thus, any function can be integrated exactly as:

$$\int_x f(x)dx = \int_x \left[\frac{f(a)}{0!} + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots + \frac{f^{n-1}(a)}{(n-1)!}(x-a)^{n-1} + R_n \right] dx \quad (20)$$

Hence, truncation error can be defined as:

$$\int_x R_n dx = \int_x f(x) dx - \int_x \left[\frac{f(a)}{0!} + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots + \frac{f^{(n-1)}(a)}{(n-1)!}(x-a)^{n-1} \right] dx \quad (21)$$

Integrating Eq. (19) yields:

$$\int_x R_n dx = \frac{f^n(\zeta)(x-a)^{n+1}}{(n+1)!} \Big|_x \quad (22)$$

However, the closed form solution of $\int_x R_n dx$ cannot be obtained since (ζ) is unknown.

The truncation error can be represented by an interval variable. The interval number is a closed set as:

$$\tilde{X} = [\underline{x}, \bar{x}] = \{x \in \mathfrak{R} \mid \underline{x} \leq z \leq \bar{x}\} \quad (23)$$

The maximum truncation error is found:

$$\max \left\{ \int_x R_n dx \right\} = \max \left\{ \frac{f^n(\zeta)(x-a)^{n+1}}{(n+1)!} \Big|_x \right\} \quad (24)$$

The bounds on the truncation error are computed:

$$E = \max \left\{ \frac{f^n(\zeta)(x-a)^{n+1}}{(n+1)!} \Big|_x \right\} [-1, 1] \quad (25)$$

These interval Taylor series expansion bounds are used in order to represent truncation error of $[H]$ and $[G]$ matrices when numerical integration is not used. The approximate terms of the $[H]$ and $[G]$ matrices for an element of length (L) are computed as:

$$\int_x f(x)dx = \int_x \left[\sum_{n=1}^8 \frac{f^{n-1}(a)}{(n-1)!} (x-a)^{n-1} \right] dx = \left[\sum_{n=1}^8 \frac{f^{n-1}(a)}{(n)!} (x-a)^n \right]_0^L \quad (26)$$

5. Interval Boundary Element Formulation

The bounds on the exact value of the non-diagonal terms of $[H]$ and $[G]$ matrices are computed using Eqs. (26) and (25). The diagonal terms of the $[H]$ matrix are computed such that the matrix $[H]$ satisfies the rigid body motion constraint. The diagonal terms of the $[G]$ matrix require special consideration since they contain singular integrals, as the distance $r = |x - \xi|$ vanishes at the node. The approximate value of the diagonal terms is computed using Eq. (26).

Since the function is singular at the node, $\max\{f^n(\zeta)\}$ becomes infinite, Eq. (25) cannot be used to meaningfully determine the error bound. The closed form solution of the improper integral of the diagonal terms of the $[G]$ matrix is found, which is not necessarily in the domain of the actual problem. If the domain of the improper integral is different than that of the problem, the remaining domain is integrated numerically using Eq. (26) and the error found using Eq. (25). If the domain of the improper integral is that of the problem, the difference between the closed form solution and the numerical integration is considered as truncation error.

Interval Boundary Element Analysis using the interval bound on the truncation error is performed as:

$$\tilde{H}\tilde{u} = \tilde{G}\tilde{q} \quad (27)$$

Eq. (14) is rearranged as:

$$\tilde{A}\tilde{x} = \tilde{f} \quad (28)$$

The interval linear system of equation can be solved by Matlab Interval Toolbox [MATLAB 6.5.1], which uses Newton-Krawczyk iteration.

6. Discretization error

In the analysis of the discretization error, we will look for interval bounded unknown functions that will satisfy the continuous problem.

$$\frac{1}{2}u(\xi) + \int_{\Gamma} q^*(x, \xi)u(x)d\Gamma = \int_{\Gamma} u^*(x, \xi)q(x)d\Gamma \quad \xi \in \Gamma \quad (29)$$

The existence and uniqueness of the solution to the above problem for two dimensional Laplace equation when (u) or (q) (but not both) is given is well studied [Friedman 1976]. We will assume that the exact solution to Eq. (29) is $u(x)$ and $q(x)$.

The boundary Γ is subdivided into elements. For each element, we will seek the interval values (\tilde{u}) and (\tilde{q}) that bound the functions (u) and (q) over an element (i) (see Figure 2) such that:

$$\tilde{u}_i \in [\underline{u}_i, \overline{u}_i], \tilde{q}_i \in [\underline{q}_i, \overline{q}_i] \quad \forall \xi \quad \frac{1}{2}u(\xi) + \sum_i \int_{\Gamma_i} q^*(x, \xi)u_i(x)d\Gamma = \sum_i \int_{\Gamma_i} u^*(x, \xi)q_i(x)d\Gamma \quad (30)$$

If (u) or (q) are specified as boundary conditions, the bounds of the function are assumed to be given explicitly. Each term of the summation in Eq. (30) is represented graphically in Figure 1.

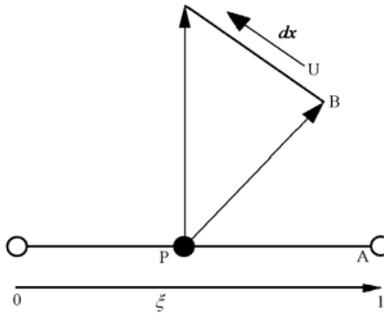


Figure 1. Integration from element B from point P on element A.

The integral of the product will be expanded to the product of two intervals: the interval value of u or q and the interval bounds of the integral of the singular solution over the element for all values of (ξ) . For example:

$$\int_{\Gamma_i} q^*(x, \xi) u_i(x) d\Gamma \subset \int_{\Gamma_i} q^*(x, \xi) d\Gamma \tilde{u} \quad (31)$$

if (q^*) has the same sign over the element. If not, the integration domain is subdivided into portions that have the same sign for (q^*) . Then the integral is replaced by interval bounds.

$$\int_{\Gamma_i} q^*(x, \xi) d\Gamma u_i \subset \tilde{h}_{ji} \tilde{u} \forall \xi \in \Gamma_j \quad (32)$$

Eq. (31) is illustrated in Figure 2 schematically.

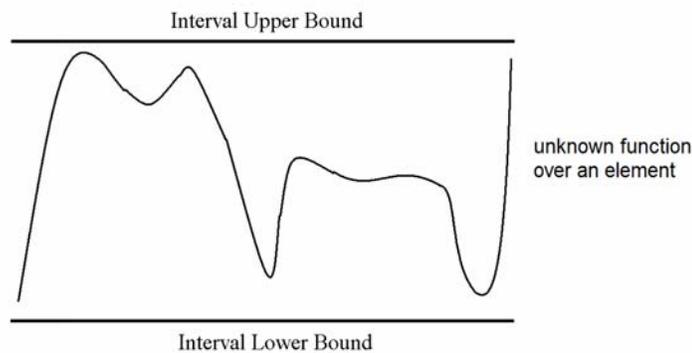


Figure 2. Interval bounds on solution to an element.

Thus, the interval bounds on the solution of Eq. (30) can be expressed as a generalized interval system of linear equations.

For sharp bounds, the parametric dependence of each row of the $[H]$ or $[G]$ matrices on (ξ) must be included in the solution of the interval system.

7. Examples

7.1. EXAMPLE 1

The first example is a demonstration of the interval treatment of uncertain boundary conditions. The unit square domain of the problem as well as the BEA mesh is shown in Figure 3. The left and right hand sides have a zero flux boundary condition while the bottom is between a $[0,1]$ potential and the top is at a $[1,2]$ potential.

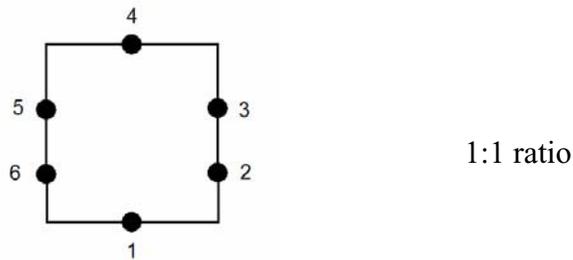


Figure 3. Boundary discretization using six constant boundary elements.

Boundary Conditions: $u_1=[0,1]$, $q_2=0$, $q_3=0$, $u_4=[1,2]$, $q_5=0$, $q_6=0$

The interval bounds are shown and compared with the combinatorial solution (Table 1) for the unknown boundary values. In this solution, the interval solution has significantly larger width compared with the combinatorial solution.

We attribute this over estimation to the fact that right hand side in a boundary element solution includes terms that involve products of the interval boundary conditions with terms from the $[H]$ or $[G]$ matrices. Methods for preserving the parameterization of the right hand side vector need to be explored to provide sharper results.

Node Value	Lower Bound	Combinatorial Lower Bound	Combinatorial Upper Bound	Upper Bound
q1	-2.5770	-2.0763	0.0000	0.5007
q2	0.0922	0.2451	1.2451	1.3981
u3	0.6019	0.7549	1.7549	1.9078
u4	-0.5007	0.0000	2.0763	2.5770
q5	0.6019	0.7549	1.7549	1.9078
q6	0.0922	0.2451	1.2451	1.3981

Table 1. Solutions to Laplace equation with uncertain boundary conditions.

7.2. EXAMPLE 2

The second example uses interval BEA to solve Laplace equation on a 2 x 1 domain using six constant boundary elements with a node located at the mid-point (Figure 4). The sides of the domain have zero flux while the bottom is at zero potential and a potential of 50 is at the top. In this example we will use a four point integration method based on a Taylor series to develop interval terms in the $[H]$ and $[G]$ matrices. The interval system of equations is then solved using Matlab.

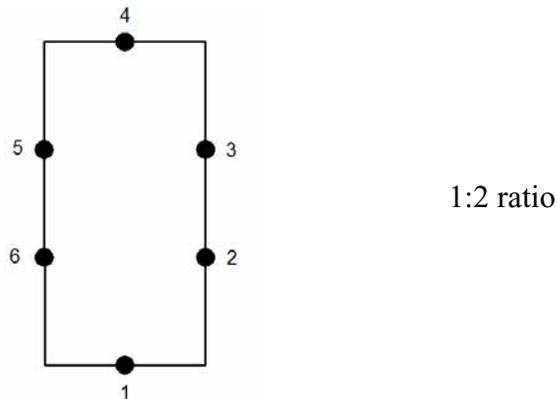


Figure 4. Boundary discretization using six constant boundary elements.

Boundary Conditions: $u_1=0, q_2=0, q_3=0, u_4=50, q_5=0, q_6=0$

The solution obtained by exact integration is shown and compared to the bounds of the solution using the proposed method (Table 2).

Node Value	Lower Bound	Solution with exact integration	Upper Bound
q1	-33.6604	-28.1967	-23.9615
u2	11.1689	11.9357	12.4285
u3	37.5192	38.0643	38.8833
q4	23.4502	28.1967	34.1717
u5	37.5192	38.0643	38.8833
u6	11.1690	11.9357	12.4285

Table 2. Solutions to Laplace equation in presence of truncation error.

The results obtained by the present method shows that the presence of truncation errors in integration as well as in solution of the system of linear equations can be bounded using Interval Boundary Element Analysis.

7.3. EXAMPLE 3

The third example obtains the bounds on discretization error for the BEA of the Laplace equation. We consider a unit domain with zero flux on each side, a zero potential on the bottom and a unit potential on the top. With the coarse meshes used as well as the need to improve the solution of a parameterized system of interval equations, we will present bounds calculated by a “brut force” construction of interval bounds by constructing terms in the $[H]$ and $[G]$ matrices by moving the point (ξ) over the domain of an element to evaluate terms in Eq. (32). Thus, the results represent the potential to efficiently calculate bounds only of an optimal interval solution method to the parametric problem can be developed.

Three different meshes are considered and the solutions in presence of the discretization error are compared.

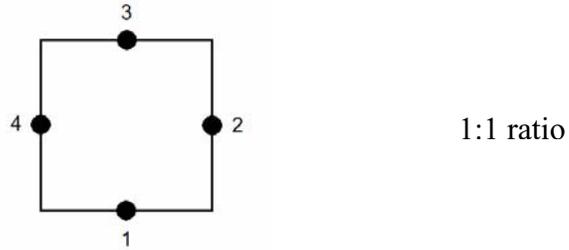


Figure 5. Boundary discretization using four constant boundary elements.

Boundary Conditions: $u_1=0$, $q_2=0$, $u_3=1$, $q_4=0$

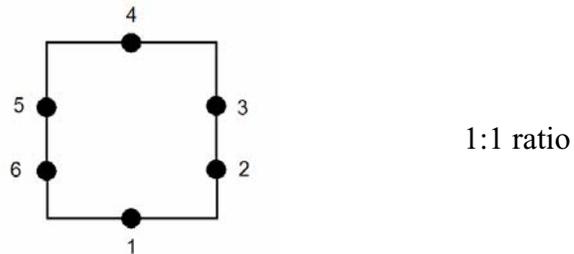


Figure 6. Boundary discretization using six constant boundary elements.

Boundary Conditions: $u_1=0$, $q_2=0$, $q_3=0$, $u_4=1$, $q_5=0$, $q_6=0$

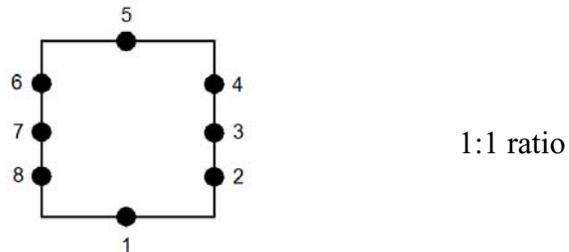


Figure 7. Boundary discretization using eight constant boundary elements.

Boundary Conditions: $u_1=0$, $q_2=0$, $q_3=0$, $q_4=0$, $u_5=1$, $q_6=0$, $q_7=0$, $q_8=0$

The bounds of the interval BEA solution are shown and compared with a conventional BEA solution where the node of each element is located at its mid-point of the element for the three different meshes (Tables 3-5).

Node Value	Lower Bound	Middle Value	Upper Bound	Width	Mid-point Node Solution
q1	-1.9896	-1.2512	-0.5129	1.4768	-1.1746
u2	0.0000	0.5000	1.0000	1.0000	0.5000
q3	0.5129	1.2512	1.98961	1.4768	1.1746
u4	0.0000	0.5000	1.0000	1.0000	0.5000

Table 3. Solutions to Laplace equation in presence of discretization error for a four node mesh.

Node Value	Lower Bound	Central Value	Upper Bound	Width	Mid-point Node Solution
q1	-1.4389	-1.0823	-0.7258	0.7131	-1.0382
u2	-0.0793	0.2431	0.5655	0.6448	0.2451
u3	0.4345	0.7569	1.0793	0.6448	0.7549
q4	0.7258	1.0823	1.4389	0.7131	1.0382
u5	0.4345	0.7569	1.0793	0.6448	0.7549
u6	-0.0793	0.2431	0.5655	0.6448	0.2451

Table 4. Solutions to Laplace equation in presence of discretization error for a six node mesh.

Node Value	Lower Bound	Central Value	Upper Bound	Width	Mid-point Node Solution
q1	-1.2737	-1.0397	-0.8057	0.4680	-1.0161
u2	-0.0731	0.1539	0.3808	0.4539	0.1639
u3	0.2856	0.5000	0.7144	0.4288	0.5000
u4	0.6192	0.8461	1.0731	0.4539	0.8361
q5	0.8057	1.0397	1.2737	0.4680	1.0161
u6	0.6192	0.8461	1.0731	0.4539	0.8361
u7	0.2856	0.5000	0.7144	0.4288	0.5
u8	-0.0731	0.1539	0.3808	0.4539	0.1639

Table 5. Solutions to Laplace equation in presence of discretization error for a eight node mesh.

The bounds on the discretization error are fairly sharp and enclose the exact solution for this problem. In fact, for the edges of the 4 element mesh, the bounds are sharp. In addition, the results show that the width of discretization error bounds reduces with mesh refinement.

8. Conclusion

In this work, new methods are presented to perform boundary element analysis in the presence of the truncation and discretization errors as well as uncertain boundary conditions. The methods rely on interval methods to quantify local errors in BEA. The examples presented demonstrate the potential of interval based boundary element methods to provide reliable engineering computations. Further work is needed to optimally solve the parametric form of the interval equations to advance interval based BEA to a truly reliable and efficient engineering analysis tool.

References

- Alefeld, G. and J. Herzberger. *Introduction to Interval Computations*, Academic Press, New York, NY, 1983.
- Brebbia, C.A. and J. Dominguez. *Boundary Elements: An Introductory Course*, Computational Mechanics, New York, McGraw-Hill, 1992.
- Friedman, Avner. *Partial differential equations*, R. E. Krieger Pub. Co., Huntington, N.Y., 1976.
- Gay, D. M. Solving Interval Linear Equations, *SIAM Journal on Numerical Analysis*, Vol. 19, 4, pp. 858-870, 1982.
- Hansen, E. *Interval arithmetic in matrix computation*, J. S. I. A. M., series B, Numerical Analysis, part I, 2, 308-320, 1965.
- Hartmann, F. *Introduction to boundary elements : theory and applications*, New York, Springer-Verlag, 1989.
- Jansson, C. Interval Linear System with Symmetric Matricies, Skew-Symmetric Matricies, and Dependencies in the Right Hand Side, *Computing*, Vol. 46, pp. 265-274, 1991.
- MATLAB 6.5.1. *Using Matlab Version 6 OEM Manual the Language of Technical Computing*, Mathworks, 2002.
- Moore, R., E, *Interval Analysis*, Prentice-Hall, Inc., Englewood Cliffs, N. J, 1966.
- Mullen, R. L. and R.L. Muhanna. Bounds of Structural Response for All Possible Loadings, "*Journal of Structural Engineering, ASCE*, Vol. 125, No. 1, pp 98-106, 1999.
- Neumaier, A. Overestimation in Linear Interval Equations, *SIAM Journal on Numerical Analysis*, Vol. 24, 1, pp. 207-214, 1987.
- Neumaier, A. Rigorous Sensitivity Analysis for Parameter-Dependent Systems of Equations, *Journal of Mathematical Analysis and Applications*, Vol. 144, pp. 16-25, 1989.
- Neumaier, A. *Interval methods for systems of equations*, Cambridge University Press, 1990.
- Rump, S. M. Rigorous Sensitivity Analysis for Systems of Linear and Nonlinear Equations, *Mathematics of Computations*, Vol. 54, 190, pp. 721-736, 1990.
- Sunaga, T. Theory of interval algebra and its application to numerical analysis, *RAAG Memoirs* 3, pp 29-46, 1958.
- Taylor B. *Methodus incrementorum directa & inversa*, Londini, typis Pearsnianis, 1715.

Reliable Dynamic Analysis of Transportation Systems

Mehdi Modares, Robert L. Mullen and Dario A. Gasparini

Department of Civil Engineering

Case Western Reserve University

Cleveland, OH 44106

E-mail: mxm206@case.edu, rlm@case.edu, dag6@case.edu

Abstract: In transportation engineering, dynamic analysis is an essential procedure for designing reliable systems. However, in current procedures of dynamic analysis for transportation systems, the possible presence of uncertainty in the system's mechanical properties and/or applied forces is not considered. In this work, a new method is developed for the dynamic analysis of continuous uncertain systems subjected to uncertain loads induced by passage of moving vehicles. First, an interval formulation is used to quantify the uncertainty present in the system's mechanical characteristics and/or magnitude of dynamic force. Then, having the interval parameters, the bounds on modal responses of the continuous system are obtained leading to determination of the upper-bounds of total response that may be used for design purposes. An example problem that illustrates the behavior of the method and a comparison with Monte-Carlo simulations are presented.

Keywords: Transportation, Dynamics, Interval, Uncertainty

1. Introduction

In design of transportation facilities, the performance of the system must be guaranteed over its lifetime. Moreover, dynamic analysis is a fundamental procedure for designing reliable systems that are subjected to dynamic forces induced by passage of moving vehicles.

However, in current procedures for dynamic analysis of transportation systems, the possible existence of uncertainty in either mechanical properties of the system or the characteristics of forcing function is generally not considered. These uncertainties can be attributed to physical imperfections, modeling inaccuracies and system complexities.

Although, in a design process, uncertainty is accounted for by a combination of load amplification and strength reduction factors that are based on probabilistic models of historic data, consideration of the effects of uncertainty has been removed from current dynamic analysis of transportation systems.

c 2006 by authors. Printed in USA.

In this work, a new method is developed to perform dynamic analysis of a continuous system subjected to a moving load in the presence of uncertainty in the system's mechanical properties as well as uncertainty in the magnitude of dynamic loads. An interval formulation is used to represent the presence of uncertainty.

Using interval calculation procedures, the upper bounds of system's response are obtained which can be used for reliable design purposes. It is shown that this method can achieve the bounds of dynamic response without Monte-Carlo simulation procedure.

2. Deterministic Dynamic Analysis

The partial differential equation of motion for a flexural beam subjected to a load moving with constant velocity (Figure 1) is:

$$EI \frac{\partial^4 u(x,t)}{\partial x^4} + \bar{m} \frac{\partial^2 u(x,t)}{\partial t^2} = P_0 \delta(x - vt) \quad (0 \leq t \leq \frac{L}{v}) \quad (1)$$

where, E is modulus of elasticity, I is the moment of inertia, u is the displacement, t is time, \bar{m} is mass per unit length, P_0 is the magnitude of load, v is the velocity of the load and δ is the Dirac-delta function.

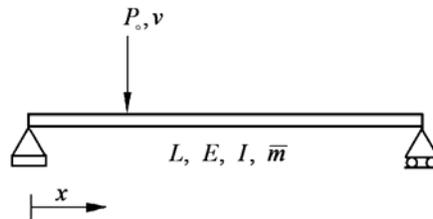


Figure 1. Simply-supported beam with moving load.

Considering free vibration of the system and assuming a harmonic solution of the form: $u(x,t) = \varphi(x)e^{i\omega t}$, in which $\varphi(x)$ a spatial function and ω is the circular natural frequency, the linear eigenvalue problem is:

$$\frac{d^2}{dx^2} \left(EI \frac{d^2 \varphi(x)}{dx^2} \right) = \omega^2 \bar{m} \varphi(x) \quad (2)$$

Applying boundary conditions for the simply-supported flexural Bernoulli beam, ($\varphi(0) = \varphi''(0) = \varphi(L) = \varphi''(L) = 0$), the solution to the characteristic equation for natural circular frequencies and corresponding mass-orthonormalized eigenfunctions (mode shapes) are:

$$\omega_n = n^2 \pi^2 \sqrt{\frac{EI}{\bar{m}L^4}} \quad (3)$$

$$\varphi_n(x) = \sqrt{\frac{2}{\bar{m}L}} \sin\left(\frac{n\pi x}{L}\right) \quad (4)$$

where, n is the mode number.

The solution for the forced vibration may be expressed as:

$$u(x,t) = \sum_{n=1}^{\infty} y_n(t) \varphi_n(x) \quad (5)$$

Where, $y_n(t)$ are the modal coordinates.

Substituting Eq. (5) in the governing equation, Eq. (1), premultiplying by $\varphi_n(x)$, integrating over the domain, decoupling and adding modal damping ratio (ζ_n), the modal equation becomes:

$$\ddot{y}_n(t) + 2\zeta_n \omega_n \dot{y}_n(t) + \omega_n^2 y_n(t) = \int_0^L \varphi_n(x) P_0 \delta(x-vt) dx \quad (6)$$

or:

$$\ddot{y}_n(t) + 2\zeta_n \omega_n \dot{y}_n(t) + \omega_n^2 y_n(t) = \Gamma_n \sin\left(\frac{n\pi v}{L} t\right) = P_0 \sqrt{\frac{2}{\bar{m}L}} \sin\left(\frac{n\pi v}{L} t\right) \quad (7)$$

where, $\Gamma_n = P_0 \sqrt{2/\bar{m}L}$ is the modal participation factor.

Defining a scaled generalized modal coordinate:

$$d_n(t) = \frac{y_n(t)}{\Gamma_n} \quad (8)$$

Eq. (7) is rewritten in terms of the scaled modal coordinate, $d_n(t)$, as:

$$\ddot{d}_n(t) + 2\zeta_n \omega_n \dot{d}_n(t) + \omega_n^2 d_n(t) = \sin\left(\frac{n\pi v}{L} t\right) \quad \left(0 \leq t \leq \frac{L}{v}\right) \quad (9)$$

For each decoupled generalized modal equation, the maximum modal coordinate is obtained from the response spectrum (maximum ratio of dynamic to static response) for modal frequency and assumed modal damping ratio (Figure 2).

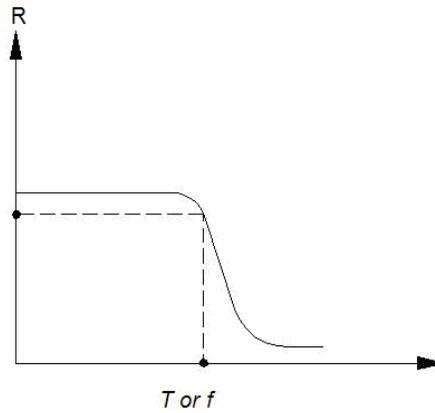


Figure 2. A generic response spectrum.

Then, the maximum modal displacement response is obtained as the multiplication of the maximum modal coordinate, modal participation factor, and mode shape as:

$$u_{n,\max} = (d_{n,\max})(\Gamma_n)(\varphi_n(x)) = (d_{n,\max})\left(\frac{2P_0}{mL}\right)\sin\left(\frac{n\pi x}{L}\right) \quad (10)$$

Finally, the total displacement response is obtained using superposition of modal maxima. The superposition can be performed by considering Square Root of Sum of Squares (SRSS) of modal maxima as (Rosenblueth 1962):

$$u_{\max} = \sqrt{\sum_{n=1}^{\infty} u_{n,\max}^2} \quad (11)$$

For practical purposes, the infinite series must be truncated. For systems with different patterns of load and boundary conditions, the same procedures can be used.

3. Interval Variables

The concept of interval numbers has been originally applied in the error analysis associated with digital computing. Quantification of the uncertainties introduced by truncation of real numbers in numerical methods was the primary application of interval methods (Moore 1966).

A real interval is a closed set defined by extreme values as (Figure 3):

$$\tilde{Z} = [z^l, z^u] = \{z \in \mathfrak{R} \mid z^l \leq z \leq z^u\} \tag{12}$$

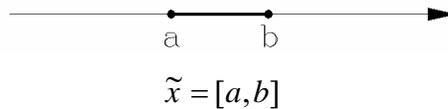


Figure 3. An interval variable.

In this work, the symbol ($\tilde{\cdot}$) represents an interval quantity. One interpretation of an interval number is a random variable whose probability density function is unknown but non-zero only in the range of interval.

Another interpretation of an interval number includes intervals of confidence for α -cuts of fuzzy sets. The interval representation transforms the point values in the deterministic system to inclusive set values in the system with bounded uncertainty.

4. Interval Dynamic Analysis

The partial differential equation of motion for a flexural beam subjected to a load moving with constant velocity with interval uncertainty in modulus of elasticity and magnitude of load is:

$$\tilde{EI} \frac{\partial^4 u(x,t)}{\partial x^4} + \tilde{m} \frac{\partial^2 u(x,t)}{\partial t^2} = \tilde{P}_0 \delta(x-vt) \quad \left(0 \leq t \leq \frac{L}{v}\right) \tag{13}$$

where, $\tilde{E} = [E^l, E^u]$ and $\tilde{P}_\circ = [P_\circ^l, P_\circ^u]$.

Then, the interval eigenvalue problem becomes:

$$\frac{d^2}{dx^2} \left(\tilde{E} I \frac{d^2 \varphi(x)}{dx^2} \right) = \tilde{\omega}^2 \bar{m} \varphi(x) \quad (14)$$

Applying boundary conditions, the solution for natural circular frequencies and corresponding mode shapes are:

$$\tilde{\omega}_n = n^2 \pi^2 \sqrt{\frac{\tilde{E} I}{\bar{m} L^4}} \quad (15)$$

$$\varphi_n(x) = \sqrt{\frac{2}{\bar{m} L}} \sin\left(\frac{n\pi x}{L}\right) \quad (16)$$

Eq. (15) can be rewritten as:

$$\tilde{\omega}_n = n^2 \pi^2 ([\sqrt{E^l}, \sqrt{E^u}]) \sqrt{\frac{I}{\bar{m} L^4}} \quad (17)$$

This shows that the lower bound of modulus of elasticity (or in general stiffness) yields the lower bound of natural circular frequency and similarly, the upper bound of modulus of elasticity yields the upper bound of natural circular frequency. This leads to an evident realization of monotonic behavior of natural circular frequencies due to variation in stiffness in continuous dynamic systems.

In discrete systems, because of the complexity of the eigenvalue problem, this realization is not straightforward. Modares and Mullen (2004) proved this monotonic behavior of natural frequencies in discrete systems using monotonicity of eigenvalues for symmetric matrices subjected to non-negative definite perturbations.

The interval modal coordinate is determined using the excitation response spectrum evaluated for the corresponding interval of natural circular frequency and assumed modal damping ratio (Figure 4).

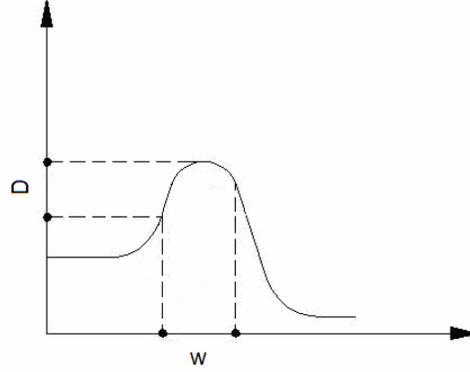


Figure 4. Determination of \tilde{d}_n corresponding to a $\tilde{\omega}_n$ for a generic response spectrum

Having the interval modal coordinate, the maximum (upperbound) modal coordinate $d_{n,\max}$ is determined as:

$$d_{n,\max} = \max(\tilde{d}_n) \quad (18)$$

The interval modal participation factor is:

$$\tilde{\Gamma}_n = \tilde{P}_o \sqrt{\frac{2}{mL}} \quad (19)$$

Therefore, the maximum modal coordinate is:

$$\Gamma_{n,\max} = \max(\tilde{\Gamma}_n) = (P_o^u) \sqrt{\frac{2}{mL}} \quad (20)$$

Then, the maximum modal displacement response is obtained as the multiplication of maximum modal coordinate, maximum modal participation factor and mode shape as:

$$u_{n,\max} = (d_{n,\max})(\Gamma_{n,\max})(\varphi_n(x)) = (d_{n,\max})\left(\frac{2P_o^u}{mL}\right) \sin\left(\frac{n\pi x}{L}\right) \quad (21)$$

Finally, the total displacement response is obtained using superposition of modal maxima. Using SRSS, the total response is:

$$u_{\max} = \sqrt{\sum_{n=1}^{\infty} u_{n,\max}^2} \quad (22)$$

5. Numerical Example

The example obtains the bounds on dynamic mid-span displacement for a continuous flexural simply-supported beam with interval uncertainty in the modulus of elasticity and magnitude of moving load.

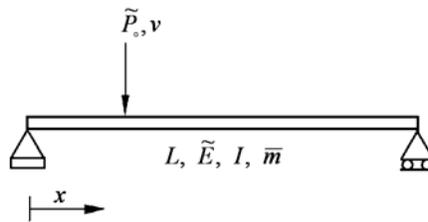


Figure 5. Flexural beam with uncertainty in modulus of elasticity and magnitude of moving load.

The beam's length is $L = 200 \text{ ft}$, mass is $\bar{m} = 1 \text{ kips/g}$ per foot, the moment of inertia is $I = 700 \text{ ft}^4$, assumed modal damping ratio $\zeta = 1\%$, and uncertain modulus of elasticity is $E = ([0.9, 1.1])576000 \text{ kips/ft}^2$. The moving load's velocity is $v = 55 \text{ mph}$, and its parametric uncertain magnitude of load is $\tilde{P}_0 = [0.9, 1.1]P_0$.

5.1. SOLUTION

The problem is solved using the present method and the results are compared with Monte-Carlo simulation solution using bounded uniformly distributed random variables in 10000 simulations.

The solution for bounds on modal natural circular frequencies is summarized in table (1).

	Lower Bound <i>Present Method</i>	Lower Bound <i>Monte-Carlo Simulation</i>	Upper Bound <i>Monte-Carlo Simulation</i>	Upper Bound <i>Present Method</i>
$\frac{\omega_n}{(n^2)}$	1.41717	1.41718	1.56673	1.56675

Table1. Bounds on Natural Circular frequencies

The response spectrum for the first (fundamental) mode is obtained and shown in figure (6).

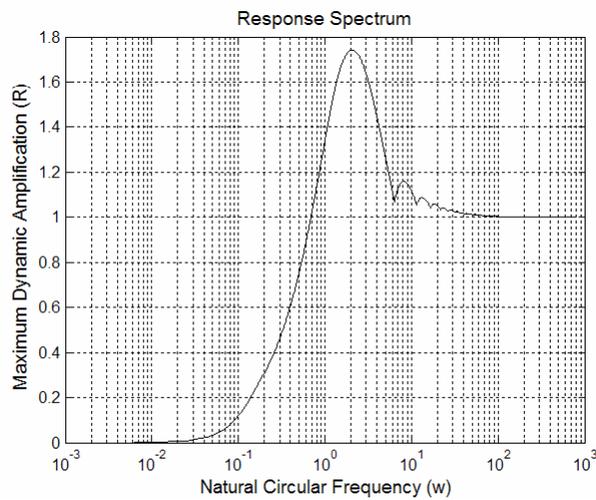


Figure 6. Response spectrum for fundamental mode of the example problem.

The upperbounds the mid-span displacement response for the fundamental mode is summarized in table (2).

	Upper Bound <i>Monte-Carlo Simulation</i>	Upper Bound <i>Present Method</i>
$\frac{u_1}{P_0}$	8.06557e-004	8.12128e-004

Table2. Upper bounds of displacement response

The first-mode beam response is depicted in figure (7).

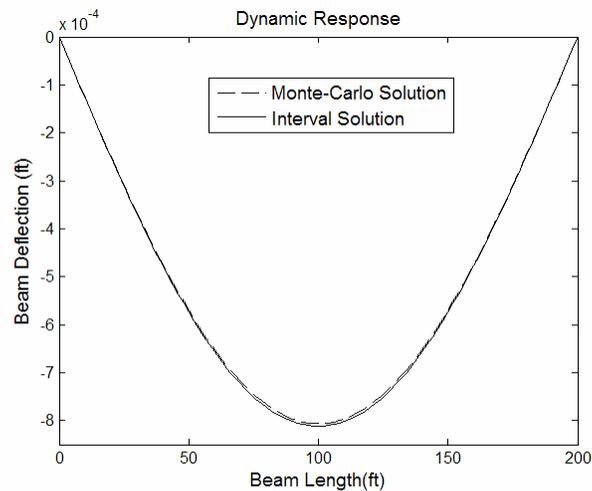


Figure 7. Beam deflection for the fundamental mode response of the example problem.

The results show that using the proposed method, the system's physics is preserved and also, the obtained sharp solutions are upper-bounds to solutions obtained by methods that produce inner-bound results such as Monte-Carlo simulation.

6. Conclusions

A new method for dynamic analysis of transportation systems with uncertainty in the mechanical characteristics of the system as well as the properties of the moving load is developed.

This computationally efficient method shows that implementation of interval analysis in a continuous dynamic system preserves the problem's physics and the yields sharp and robust results. This may be attributed to completeness of the closed-form solution in continuous dynamic systems.

The results show that obtaining bounds does not require expensive stochastic procedures such as Monte-Carlo simulations.

The simplicity of the proposed method makes it attractive to introduce uncertainty in analysis of continuous dynamic systems.

References

- Biot, M. A. Vibrations of Buildings during Earthquake, *Ph.D. Thesis* Aeronautics Department, Calif. Inst. of Tech., Pasadena, California, U.S.A. 1932.
- Clough, R.W. and J. Penzien. Dynamics of Structures. McGraw-Hill, New York 1993.
- Modares, M. and R. L. Mullen. Free Vibration of Structures with Interval Uncertainty. *9th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability* 2004.
- Moore, R. E. Interval Analysis. Prentice Hall, Englewood, NJ 1966.
- Muhanna, R. L. and R. L. Mullen. Uncertainty in Mechanics Problems-Interval-Based Approach. *Journal of Engineering Mechanics* June-2001, pp.557-566 2001.
- Neumaier, A. Interval Methods for Systems of Equations. Cambridge University Press, Cambridge 1990.
- Rosenblueth, E. and J. I. Bustamente. Distribution of Structural Response to Earthquakes *Proc. ASCE*, Vol. 88, No. EM 3 p. 75 1962.
- Veletsos, A. S. and N. M. Newmark, N. M. Natural Frequencies of Continuous Flexural Members. *Trans. ASCE*, Vol. 122, p. 249 1957.

Geometric Uncertainty in Truss Systems: An Interval Approach

Rafi L. Muhanna¹, Ayse Erdolen¹, and Robert L. Mullen²

¹Center for Reliable Engineering Computing (REC)
Department of Civil & Environmental Engineering,
Georgia Institute of Technology, Savannah, GA 31407, USA
Rafi.muhanna@gtsav.gatech.edu, aerdolen@gtsav.gatech.edu

²Center for Reliable Engineering Computing (REC)
Department of Civil Engineering, Case Western Reserve University,
Cleveland, OH, USA, rlm@cwru.edu

Abstract: In this work, geometric uncertainty due to fabrication errors and/or thermal changes in engineering systems is addressed. The system components' deviation from the nominal dimensions (missfitting) are introduced as intervals. Such geometric uncertainty is converted into an equivalent nodal load uncertainty. In the case of elastic truss systems the Interval Finite Element formulation leads to a linear interval system of equations with interval right hand side. An exact enclosure on the final system geometry is obtained. Results are illustrated in example problems.

Keywords: geometric uncertainty, interval finite elements, fabrication errors

1. Introduction

Engineering systems are usually designed with a pre-described geometry in order to meet the intended function for which they are designed. However, due to fabrication errors and/or thermal changes, the dimensions of system components will deviate from their nominal values creating a missfitting problem during the manufacturing/construction process. In engineering practice, such a fabrication deviation is defined in a form of maximum allowable tolerance for individual components or for the completed system after the assemblage. Usually, the design and manufacturing processes of mechanical components require a complete definition of geometry of these components, however the definition of the geometries of the components are only

© 2006 by authors. Printed in USA.

considered complete if tolerances are included in the design. Thus, the proper design should be a "completely toleranced" design, which means that geometries of the geometrical elements of a workpiece are completely defined and toleranced (Henzold, G., 1995). More information about the allowable values of fabrication tolerances can be found in the publications of the American National Standards Institute (ANSI) or other similar international organizations such as the International Organisation for Standardization (ISO).

State of the art technologies are striving for higher performance, higher efficiency and greater reliability. To achieve such goals, the analysis and design procedures have to account for all possible factors that could affect the product. Tolerances represent one of the main sources of uncertainty that should be accounted for.

Tolerances, usually, are defined as absolute deviation from the nominal values. Thus, including the tolerance, in the analysis and design, as a possible value within a given interval that possesses known bounds might be a realistic or natural way of representing such type of uncertainty.

In the present work, tolerances (geometrical uncertainty) will be introduced as interval values i.e., the true value is known to lie between two bounding values, but the exact value is unknown.

During the last decade, Interval Finite Element Methods (IFEM) have been developed in a number of works to handle uncertainty in structural mechanics, as an example we can mention a few such as the works of Koyluoglu, Cakmak and Nielsen (1995), Rao, S.S. and Sawyer, J.P. (1995), Rao, S.S. and Berke L (1997), Rao, S.S. and Li Chen (1998), Nakagiri and Suzuki (1999) Muhanna and Mullen (1995), Muhanna and Mullen (1999), Mullen and Muhanna (1999); Corliss, G., C. Foley, and R. B. Kearfott (2004); Popova, E.D, M. Datcheva, R. Iankov, and T. Schanz (2003), Neumaier and Pownuk (2004); Muhanna, Mullen, and Zhang (2005). The accounted for uncertainty in these works has included load, stiffness, and element cross sectional area. However, the uncertainty in the components' length has not been addressed.

In the present work we introduce a new formulation for geometric uncertainty due to fabrication errors and/or thermal changes in engineering system components with an application to elastic truss systems.

The formulation will be presented in section 2. Sample calculations are given in section 3. In this paper, boldface will denote interval quantities (interval number, interval vector, interval matrix). All interval quantities are implicitly real interval quantities. Non-boldface will denote real (deterministic) quantities. For an interval quantity \mathbf{x} or \mathbf{A} , the notation x and A is used to denote a generic (arbitrary) element $x \in \mathbf{x}$ and $A \in \mathbf{A}$.

2. Formulation

The present formulation will be developed for the case of truss structures. A *truss* is a structure composed of straight bars connected at their points of intersection by means of momentless joints called pins or hinges (frictionless joints). All loadings are assumed to be applied only at these points of intersection. Thus each straight bar is subjected only to axial force, not to shear forces, bending nor twisting moments.

Due to fabrication errors and/or thermal changes certain bars could have improper length. In practice, the bar is forced into its position between two joints by applying some initial extension or compression. Under such a condition, some axial forces are introduced in the bars in the absence of external loads. The solution of such a problem in the absence of uncertainty is well known in the text books of structural engineering. However, based on the engineering practice, the length of the truss bar is introduced as a random value that is equal to the nominal value plus/minus a tolerance. That means, the bar length can have any value between two bounds, namely $L_o - \delta L$ and $L_o + \delta L$, where L_o is the bar nominal length and δL is the given tolerance. In this study we will incorporate uncertainty in the bar length as the range between the lower and upper bounds on the nominal length of the bar.

$$L \in \mathbf{L}, \quad \mathbf{L} \equiv [\underline{L}, \bar{L}] := \{L \in R \mid \underline{L} \leq L \leq \bar{L}\} \quad (1)$$

$$\mathbf{L} = [L_o - \delta L, L_o + \delta L] \quad (2)$$

The formulation includes two steps and the results of the two steps are then superimposed. Since it is required that all bars have to fit the nominal pre-described geometry, then if a bar is longer than its nominal length it should be compressed to fit into its position between two joints. So when the bar is released it will apply equal and opposite compressive forces on its joints, and if the bar is shorter it will apply a tensile forces. The axial forces developed in all bars due to initial extension/compression or temperature changes can be determined and then can be used to calculate the nodal forces within the finite element context. By doing that, the geometric uncertainty in the bar's length is converted into an equivalent load uncertainty. The interval axial force for a typical bar element will be given by

$$\mathbf{F} = EA \frac{\delta L}{L_o} \quad (3a)$$

Where $\delta L = [-\delta L, +\delta L] = [\underline{\delta L}, \overline{\delta L}]$ is the interval deviation from the nominal value of the bar's length, E is the modulus of elasticity, and A is the cross sectional area of the bar. In the case of a temperature change of the interval amount δT the interval force will be given by

$$F = EA\alpha\delta T \tag{3b}$$

Where $\delta T = [-\delta T, +\delta T] = [\underline{\delta T}, \overline{\delta T}]$ is the interval of the temperature change, and α is the coefficient of thermal expansion.

The combination of fabrication errors and temperature changes can be analyzed using the sum of equivalent forces.

To illustrate how the above mentioned procedure can be applied, let us consider a typical truss bar element as shown in figure 1. According to finite element formulation (Bathe, Gallagher, Zienkiewicz and Taylor) the nodal forces induced by a given bar due fabrication error or temperature change can be determined as

$$P_o = \begin{pmatrix} F_{1x} \\ F_{1y} \\ F_{2x} \\ F_{2y} \end{pmatrix} = F \begin{pmatrix} c \\ s \\ -c \\ -s \end{pmatrix} = EA \frac{\delta L}{L_o} \begin{pmatrix} c \\ s \\ -c \\ -s \end{pmatrix} \tag{4}$$

Where P_o is the interval vector of nodal forces obtained as a result of the missfitting problem, $c = \cos\varphi$, and $s = \sin\varphi$. In the absence of external loading the final interval finite element system of equations can be given by

$$KU = MF \tag{5}$$

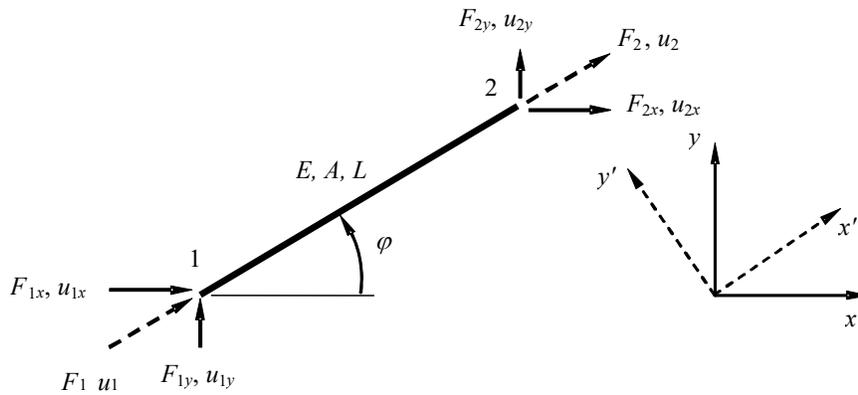


Figure 1. Local (x', y') and global (x, y) coordinate systems for a truss bar element

Where K is the stiffness matrix of the system and U is the vector of interval displacements, M is a matrix that relates the system's degrees of freedom with the elements loads, the complete derivation of this matrix can be found in Mullen and Muhanna 1999, and F is the interval vector of elements' fabrication errors or temperature changes. Equation (5) is an interval linear system, where only the right hand side is interval and an exact enclosure can be obtained. This enclosure represents the final deformed geometry of the truss due to missfitting.

To obtain the final internal force in each bar, first we need to calculate the internal force in each bar due to the nodal forces P_o using the following equation

$$S_i = K_i L_i U \quad (6)$$

where S_i is the interval force of the i th bar of the truss, K_i is i th element stiffness matrix, and L_i is a Boolean matrix with 1 and 0 entries, and secondly the obtained force should be added to that force given in equation 1 or 2, depending on the case under consideration, i.e. fabrication error or temperature change. In the next section we will introduce some example problems.

3. Examples

Numerical solution a one-bay truss (6 elements) shown in Figure 2. The following data: $E=200$ GPa, the same cross-sectional area for all members $A=0.01 \text{ m}^2$, the same fabrication error for all members $\delta L = [-0.001, 0.001]$ is assumed. The results for displacement of all nodes (upper and lower bounds) are given in Table 1.

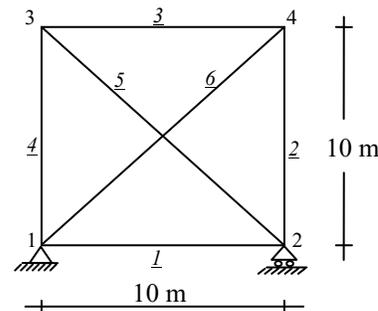


Figure 2. One-Bay Truss

Table 1. One Bay Truss (6 Elements) Nodes Displacement

<i>Node</i>	$U_x (m)$	$U_y (m)$
1	[0, 0]	[0, 0]
2	[-0.00150,0.00150]	[0, 0]
3	[-0.003414, 0.003414]	[-0.00150,0.00150]
4	[-0.003207,0.003207]	[-0.00150,0.00150]

4. Conclusions

A method for the analysis of structures and mechanical components with geometric uncertainties given in the form of dimensional tolerances is presented. This method is based on the use of interval computations in the context of a finite element analysis. The geometric uncertainties are transformed to equivalent uncertain loads. The need for maintaining parametric relationships within the interval formulations is ensured by an element-by-element approach to the formulations. Exact enclosure on the deformed geometry is obtained. Example calculations are presented that show the sharpness of the interval calculations.

References

- Bathe, K. (1996). *Finite Element Procedures*, Printice Hall, Englewood Cliffs, New Jersey 07632, New Jersey.
- Computer Science and Mathematics in Architecture and Civil Engineering, Weimar, 2003,
- Corliss, G., C. Foley, and R.~B. Kearfott., Formulation for Reliable Analysis of Structural Frames, In R. L. Muhanna and R. L. Mullen, editors, Proceedings of NSF workshop on Reliable Engineering Computing}, Savannah, Georgia, September 2004, USA.
- Gallagher, R. H. (1975). *Finite Element Analysis Fundamentals*, Prentice Hall, Englewood Cliffs. Germany.
- Henzold, Georg, Handbook of Geometrical Tolerancing, Design, Munufacturing and Inspection. John Wiley & Sons Ltd., England, 1995.

- Hibbeler, R. C. "Structural Analysis", Prentice Hall, 2006.
- IKM2003: Digital Proceedings of 16th International Conference on the Applications of ISO (International Organisation for Standardization): *Guide to the Expression of Uncertainty in Measurements*. Printed in Switzerland, 1995.
- Koyluoglu, U. and Elishakoff, I. (1998). "A Comparison of Stochastic and Interval Finite Elements Applied to Shear Frames with Uncertain Stiffness Properties," *Computers and Structures*, Vol. 67, No. 1-3, pp.91-98.
- Koyluoglu, U., Cakmak, S., Ahmet, N., and Soren R. K. (1995). "Interval Algebra to Deal with Pattern Loading and Structural Uncertainty," *Journal of Engineering Mechanics*, November, 1149-1157.
- Muhanna, R. L. and Mullen, R. L.(1995). "Development of Interval Based Methods for Fuzziness in Continuum Mechanics," *Proc., ISUMA-NAFIPS'95*, September 17-20, 145-150.
- Muhanna, R. L. and Mullen, R. L. (1999). "Formulation of Fuzzy Finite Element Methods for Mechanics Problems," *Computer-Aided Civil and Infrastructure Engineering (previously Microcomputers in Civil Engineering)*, Vol.14, pp 107-117.
- Muhanna, R.L. R.L. Mullen, Z. Hao:2005, 'Penalty-Based Solution for the Interval Finite-Element Methods'. *Journal of Engineering Mechanics*, 1102-1111.
- Mullen, R. L. and 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*, August 7-9, 310-313.
- Mullen, R. L. and Muhanna, R. L (1999). "Bounds of Structural Response for All Possible Loadings," *Journal of Structural Engineering, ASCE*, Vol. 125, No. 1, pp 98-106.
- Nakagiri, S. and Suzuki, K. (1999). "Finite Element Interval analysis of external loads identified by displacement input uncertainty," *Comput. Methods Appl. Mech. Engrg.* 168, pp. 63-72.
- Neumaier, A., and A. Pownuk. ,Linear Systems with Large Uncertainties, with Applications to Truss Structures, 2005. <http://www.mat.univie.ac.at/~neum/ms/linunc.pdf>
- Popova, E.~D., M. Datcheva, R. Iankov, T. Schanz., Mechanical Models with Interval Parameters
- Rao, S. S., Berke, L. (1997). "Analysis of Uncertain Structural Systems Using Interval Analysis," *AIAA Journal*, Vol. 35, No. 4, pp 727-735.
- Rao, S. S., LI Chen, (1998). "Numerical Solution of Fuzzy Linear Equations In Engineering Analysis," *Int. J. Numer. Meth. Engng.* Vol. 43, pp 391-408.
- Rao, S. S., Sawyer, P. (1995). "Fuzzy Finite Element Approach for Analysis of Imprecisely Defined Systems," *AIAA Journal*, Vol. 33, No. 12, pp 2364-2370.

Prediction of Deflection for Prestressed Concrete Girders

Using a Bayesian Approach

X.J. Chen^{1,2}, C.W. Shen¹, and L. J. Jacobs²

¹ *School of Transportation,
Wuhan University of Technology,
Wuhan ,Hubei, P.R.China 430063*

² *School of Civil & Environmental Engineering,
Georgia Institute of Technology,
Atlanta ,GA 30332
email: xc11@mail.gatech.edu*

Abstract: To control the alignment of prestressed concrete bridge during construction, it is reasonable to treat the prediction of the deflection as an uncertainty. This paper presents a Bayesian updating approach to predict the deflection of prestressed concrete girders. A prior distribution is developed by using Monte Carlo stimulation with a proposed deterministic model under a variety of prestressing levels, material properties and environmental conditions. Then the posterior distribution is obtained by updating the prior distribution based on a limited number of initial measurements, thus greatly reducing the uncertainty of the deflection prediction. The method is applied to predict the camber of two actual prestressed T girders, and the predictions are satisfied.

Keywords: PC, deflection, Bayesian approach, prediction

1. Introduction

In order to obtain a good alignment design for a particular prestressed concrete (PC) bridge, the deflection is a very important aspect in its construction control. It is obvious that the deflection of PC girders varies with the different cases because of various random effects; these effects include

© 2006 by authors. Printed in USA.

concrete strength, elastic modulus, creep and shrinkage as well as section properties. All these make the deterministic prediction of the deflection unrealistic. It is therefore reasonable to treat the prediction of the deflection as an uncertainty. This paper provides an assessment of the variability of the deflection for PC girders with a Bayesian updating approach (Bazant and Wittman, 1987; Zhang and Du, 1994). Based on a deterministic model for deflection, the numerical calculation is repeated many times using Monte Carlo simulation (Li Jihua, 1988), the results of which is used as a prior distribution. With a limited number of measurements available, a posterior distribution can be obtained by updating the prior distribution and a more believable long-term prediction for the deflection of the PC girders can be assessed.

2. Deterministic model for time-dependent deflection

Time-stepping approaches for deformation calculations based on the principle of superposition have appeared in many literatures. For the case of changing stress, the stress history can be divided into several sections $[t_j, t_{j+1}]$, the stress is assumed as a series of stress increments $\Delta\sigma(t_j)$ applied at times t_j , then the creep strains can be expressed as follows based on the principle of superposition:

$$\varepsilon(t_{i+1}) = \sigma_0 \Phi(t_{i+1}, t_0) + \sum_{j=1}^i \Delta\sigma(t_j) \Phi(t_{i+1}, t_j) \quad (1)$$

The Eq.(1) can also be written in a form of a section curvature as follows:

$$\psi(t_{i+1}) = \frac{M_0}{I} \Phi(t_{i+1}, t_0) + \sum_{j=1}^i \frac{\Delta M(t_j)}{I} \Phi(t_{i+1}, t_j) \quad (2)$$

where $\varepsilon(t_{i+1})$ and $\psi(t_{i+1})$ are the total strain and section curvature at time t_{i+1} respectively; σ_0 and $\Delta\sigma(t_j)$ are the instantaneous stress at time t_0 and the increment at time t_j ; M_0 and $\Delta M(t_j)$ are the instantaneous moment at time t_0 and the increment at time t_j respectively; I is the moment inertia of the section; and $\Phi(t_{i+1}, t_j)$ is a creep function for creep strain at time t_{i+1} . Among all the creep and shrinkage models, ACI Committee 209, CEB-FIP (MC78, MC90), BP2 (Bazant and Lisa, 1980) are often recommended. Although the BP2 is the most complex in form, it takes the effects of aggregation into account and it is adopted in this paper. The deflection then can be calculated by many approaches (Glali and Azarnejad, 1996) such as the finite element method. For a simply

supported beam, the deflection f at the mid span can be accurately calculated using the following equation:

$$f = \frac{1^2}{96} [\psi_{E1}(t, \tau) + 10\psi_M(t, \tau) + \psi_{E2}(t, \tau)] \quad (3)$$

where ψ is the section curvature. The subscripts E1 and E2 mean both the end sections of beam, and M the mid section.

Table 1. Parameter statistical distributions for prestressed concrete girder

Variables	Mean	Std. Dev.	Cov.
C50 concrete strength (MPa)	39.9088	4.9088	—
Unit weight of concrete (kg/m ³)	2400	80	—
permissible prestress σ_k (MPa)	1.00 σ_k	—	0.055
Prestressing aere A_p (cm ²)	1.01176 A_p	—	0.0125
Moment inertia I (cm ⁴)	1.006 I	0.0107	—
Location of duct d (cm)	1.00 d	1.20	—
Volume surface ratio V/S	1.00 V/S	—	0.0528
Coarse aggregate W_1 (kg)	1.00 W_1	—	0.100
Fine aggregate W_2 (kg)	1.00 W_2	—	0.100
Cement W_3 (kg)	1.00 W_3	—	0.050
Water W_4 (kg)	1.00 W_4	—	0.050
Humidity	70%	—	13.3%

Based on the deterministic approaches mentioned above, Monte Carlo simulation can be used. The basic idea of a Monte Carlo analysis is repeatedly to simulate random input parameters. These statistical parameters are listed in Table 1 based on related information. (Zhao and Jin, 2000). All variables are considered to be normally distributed for simplicity.

3. Bayesian approach for deflection prediction

According to the Bayesian formula, if the initial probability $P(X_{ik})$ of all hypotheses X_{ik} are

known, the posterior probability $P(X_{ik})$ then could be obtained in conjunction with a set of limited measurements S_M which were taken during the early life of the girder. As stated in the former section, the probability obtained by Monte Carlo stimulation can be taken as a prior probability:

$$P'(X_{ik}) = C \cdot L(S_M | X_{jk}) \cdot P(X_{ik}) \quad (4)$$

where X_{ik} represents the predicted deflection at time t_i on the k^{th} Monte Carlo run; S_M is a set of measured deflection; C is a normalizing constant and $L(\cdot)$ represents the likelihood function, which means the likelihood of obtaining the measured values. Assuming that the deflection is normally distributed, $N(\mu_{ik}, \sigma_i)$ and the statistical independence is appropriate, then,

$$L(S_j | X_{jk}) = \prod_{j=1}^m p_j(S_j | X_{jk}) \quad (5)$$

in which,

$$p_j(S_j | X_{jk}) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left[-\frac{1}{2}\left(\frac{S_j - \mu_{jk}}{\sigma_j}\right)^2\right] \quad (6)$$

$$\text{let } w_k = \exp\left[-\sum_{j=1}^m \frac{1}{2}\left(\frac{S_j - \mu_{jk}}{\sigma_j}\right)^2\right] \quad (7)$$

replacing them into Eq.(3) and since C should ensure the total probability to be unity, the probability of the deflection X_{ik} appeared at the time t_i in the k^{th} Monte Carlo run becomes,

$$P'(X_{ik}) = \frac{\sum w_k P(X_{ik})}{\sum w_k} \quad (8)$$

The mean and standard deviation of the posterior distribution for the deflection at time t_i can be further written as

$$\bar{X}_i' = \frac{\sum w_k X_{ik}}{\sum w_k} \quad (9)$$

$$V'_i = \sqrt{\frac{\sum w_k (X_{ik} - \bar{X}'_i)^2}{\sum w_k}} \quad (10)$$

4. Experiments on T girders

The cambers of two of prestressed concrete T girders were measured during the early phase of the bridge. Both of the girders (T1 and T2) are the same in section and span with a height and a span 1.68m and 30m respectively, prestressing strand 270(low relaxation), the total prestressing steel area 0.0028m^2 , permissible prestress $0.75R_y^b$, concrete C50. Area and moment of inertia of the section are 0.615m^2 and 0.2115m^4 . The prestressing ages for T1 and T2 girders are 5 and 4 days, respectively. The camber measurement was taken before sunrise so as to reduce the effect of the temperature.

Table 2. The mean and standard deviation of the girders at loading age of 100 days

State	Number of samples used	T1		T2	
		mean	std. dev.	mean	std. dev.
Before updating	0	2.8742	0.4873	2.9816	0.5095
After updating	5	2.8679	0.2057	3.0616	0.2146
	15	2.9211	0.1268	3.0273	0.1367

Numerical calculations were conducted with the Bayesian updating algorithm outlined above using only the five data values at the early period. The results for both T1 and T2 girders are shown in Figure 1 and 2. These figures show the camber (prior) mean of the girders based on the Monte Carlo simulation and the updated (posterior) mean based on the limited measured data and their 95% confidence limits are also indicated. In both experiments, the measured data produce a significant narrowing of the confidence limit band as shown in Table 2, which demonstrates an improvement in the confidence of long-term prediction. It is also noticed that the later measured data are all fallen within the narrowed limit band which verifies the confidence of the proposed method. From the comparison of the two girders, the deviation of initial measured data has great effects on the long-term prediction, so correctly measured data should be ensured.

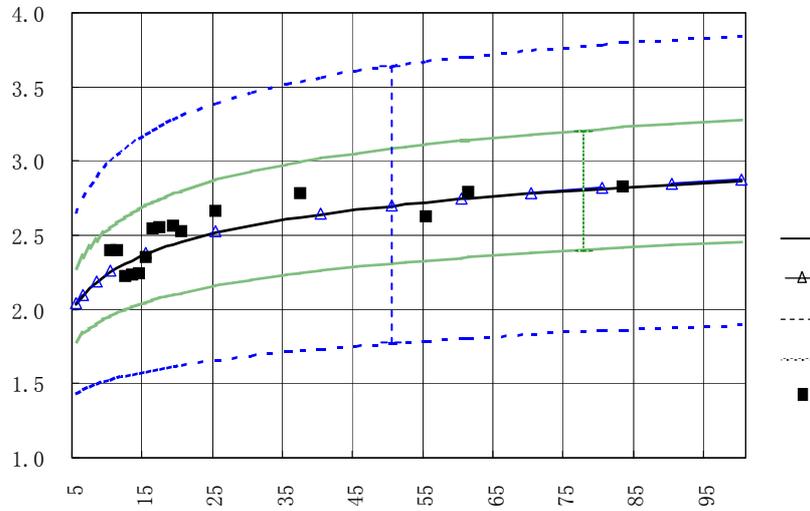


Figure 1. Camber of T1 at midspan (cm)

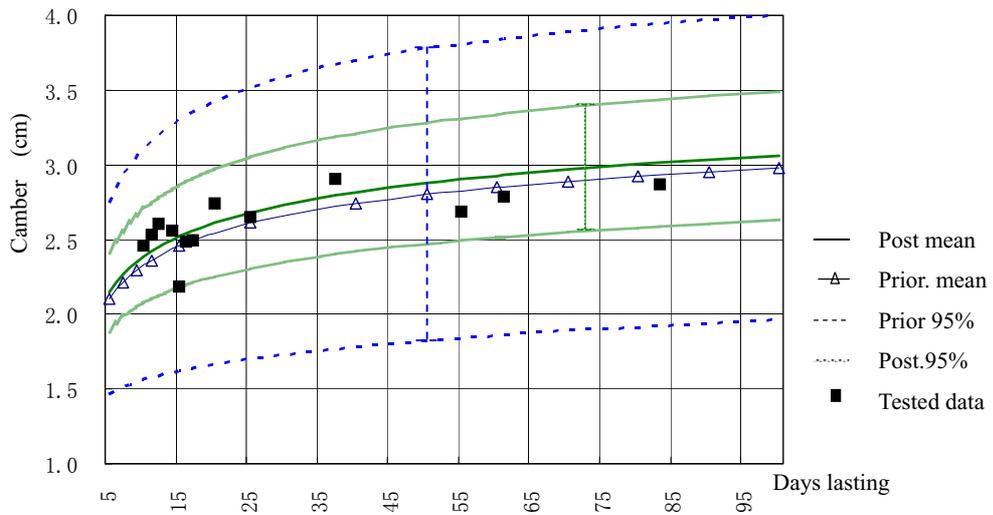


Figure 2. Camber of T2 at midspan (cm)

5. Conclusion

This paper presents a method to reduce the uncertainties in the long-term prediction of a prestressed concrete girder. By updating the prior distribution based on a limited number of measurements in the early stage of the girder, the uncertainty of its deflection prediction can be greatly reduced. Two practical experiments show that the results are accurate. It is also noticed that the deviation of the measurements has great effects on the long-term prediction.

Reference

- Bazant, Z.P., Kim, J.F., Wittman, F.H. and Alou, F., Statistical Extrapolation of Shrinkage Data-Part II: Bayesian Updating. *ACI Materials Journal*, 83(2): 83-91, 1987
- Zhang Yaoting and Du Jingsong, Bayesian Analysis, Likelihood Ratio and Information, *Applied Mathematics*, JCU 9B: 105-108, 1994
- Li Jihua, *Reliability Mathematics*, Construction Industry Publishing Company of China, Beijing, 1988.(in Chinese)
- Bazant, Z.P., Lisa Panula, Creep and shrinkage characterization for analyzing prestressed concrete structure, *PCI Journal*, 25(3):87-123, 1980
- Glali, A., Azarnejad, A., Deflection Prediction of Members of Any Concrete Strength. *ACI Structural Journal*, 96(5): 807-816, 1996
- Zhao Guofan, Jin Weiliang and Gong Jinxin, *Theory of Structural Reliability*, Construction Industry Publishing Company of China, Beijing, 2000.(in Chinese)
- Specification for Design on RC and PC Highway Bridges and Culverts (JTJ023-85)*, Ministry of Communications of China, Beijing, 1985. (in Chinese)

■ NSF WORKSHOP ON RELIABLE ENGINEERING COMPUTING ■
MODELING ERRORS AND UNCERTAINTY IN ENGINEERING COMPUTATIONS

PROCEEDINGS: AUTHOR INDEX

A		N	
Adams	267	Neumaier	113
Alexander	189	Neumann	75
Araiza	229	Nooner	65
Aughenbaugh	319		
B		O	
Beer	369	Ocloo	189
Bichon	267	Orshansky	197
Bonev	245	Osorio	179
Bruns	341		
C		P	
Ceberio	127	Paluri	169
Chavez	39	Paredis	319, 341
Chen	477	Popova	245
Chessa	229		
Chopra	127	R	
D		Reuter	419
Daumas	39	Romero	179
F		S	
Ferson	115, 127, 341	Santoro	39
G		Schön	75
Gasparini	457	Solin	229
Gianchandani	189	Stadtherr	155
Gupte	189	Svasek	213
H		T	
Hickey	91	Tonon	1
Hu	65	V	
K		Vainstein	25, 179
Kreinovich	115, 127, 197, 229	W	
Kubal	169	Wang, W.	197
Kutterer	75	Wang, Y.	293
L		Wittenberg	91
Lin	155	X	
M		Xiang	127, 197, 229
Marte	179	Y	
Modares	457	Yankov	245
Moeller	419	Z	
Mourelatos	391	Zalewski	439
Muhanna	229, 439, 469	Zhou	391
Mullen	439, 457, 469		
Murguia	127		