PREFACE

Goals and purposes of the special issue

During the last decades, computer assisted modeling and analysis of various industrial processes have gained importance. Computers help to reduce the design and development time for new products and to substitute low cost virtual tests for expensive experiments on real life prototypes. However, the results are often unreliable due to errors that are generated either by the underlying computer arithmetic or by inaccuracy resulting from idealizing the mathematical model of the process considered. In this special issue, we focus on verified methods as a means of solving such problems.

A method is called *verified* if it guarantees the correctness of its output. In this context, intervals and Taylor models are widely used approaches to verifying results obtained on a computer. For example, the former provides a (multidimensional) box described in terms of floating point arithmetic which is guaranteed to contain the exact result. Besides, these methods are able to allow for uncertainty in parameters, which helps to generate more realistic mathematical models or to take into account measurement errors.

The goal of this special issue is to make verified methods known to a broader circle of researchers and industry representatives working on modeling, analysis, and design of systems with uncertainties. For this purpose, we outline their potential by presenting selected applications in medicine and engineering. The contents of the special issue reflect and supplement the presentations which were given at the minisymposium *Validated Methods: Applications to Modeling, Analysis, and Design of Systems in Medicine and Engineering* during the 15th European Conference on Mathematics for Industry *ECMI* in July 2008.

This issue, which is formed in collaboration with scientists from Germany, France, the USA, Hungary, and Bulgaria, consists of 10 articles. The first two focus on theoretical aspects of verified computations, introducing the reader to the field and describing the recent progress. The remaining articles illuminate different application aspects of the described techniques with regard to real life problems.

In their paper, G.-Tóth and Kreinovich describe recent advances in the field of verified multi-objective optimization and provide an algorithmic analysis of the computability of Pareto sets. Rauh *et al.* focus on the developments in the area of differential-algebraic equations and present a new verified approach to solution of initial value problems with algebraic constraints.

The authors of this issue discuss mathematical system models which describe static as well as dynamic processes. These models are given by sets of algebraic equations, ordinary differential equations, and differential-algebraic equations. In all cases, we have to deal with uncertainties in system parameters which can be bounded empirically.

The rest of the issue highlights advantages of the use of verified methods for systems with bounded uncertainties from an application-oriented point of view. The articles are arranged with respect to the problems they solve. Tasks in engineering are presented first, after which applications in biomechanics and medicine are described.

The article by Merlet is an overview of problems and tasks in robotics that can be successfully solved by interval analysis although usual numerical methods might fail. Pepy *et al.* present recent results in the areas of verified robot localization and tracking as well as simultaneous localization and mapping. A further task considered in this paper is robust path planning. Rauh *et al.* apply verified techniques to controller design and develop routines for verified sensitivity analysis. Moreover, they analyze controllability, observability, and asymptotic stability of dynamical systems. A nonlinear four-dimensional model of an uncertain wastewater treatment process is analyzed from the verified point of view by Dimitrova and Krastanov. The article by Auer and Luther concludes this part of the issue by describing an integrated environment for the verification of the kinematics and dynamics of mechanical systems as well as its new uses for sensitivity analysis and simulation of closed-loop systems.

This article also builds a bridge to the last part of the issue that discusses the application of verified methods to biomechanics and medicine. Tändl *et al.* describe MOBILE, a modeling and simulation tool for mechanical systems based on usual floating point arithmetics, and its uses for accurate bone motion reconstruction from marker trajectories. SMARTMOBILE was built on top of MOBILE, which presupposes that non-verified biomechanical models and methods developed with the latter tool can be verified with the help of the former. The article by Tändl *et al.* does not contain verification aspects, parts of which are mentioned in the article by Auer and Luther. However, this paper presents results on motion reconstruction along with their *validation* against real patient data. An uncertain model of human blood cell dynamics is analyzed from the point of view of verification by Freihold and Hofer. The issue is concluded with a paper by Enszer and Stadtherr, in which they study the impact of infections within a population using epidemiological models with uncertainties.

For all applications presented here, the use of verified techniques provides essential information about parameter dependency, robustness, and safety of technical and medical systems in terms of guaranteed bounds for the quantities of interest. In contrast to non-verified techniques, which in most cases rely on grid-based or stochastic procedures for simulation and estimation, interval methods, Taylor model approaches, and other techniques allow us to verify the worst-case influence of bounded uncertainties on mathematical system models. In that sense they supplement traditional numerical techniques for the modeling, analysis, and design of real life systems. We hope that this special issue will succeed in addressing industry and applied research representatives and increase the degree of popularity and understanding of verified methods as well as corresponding software libraries.

Main theoretical concepts and the notation used

To simplify the readability of this issue, we recapitulate the basic principles of verified computations using the example of interval arithmetic. First, elementary operations in this arithmetic are described. Then, a basic interval algorithm for solving initial value problems is outlined to give an impression of the difference to floating point analogues. Further, we provide a list of libraries that automatize different aspects of verification. Finally, the notation commonly used by the contributors is provided.

An interval $[\underline{x}\ ;\ \overline{x}]$, where $\underline{x}$ is the lower, $\overline{x}$ the upper bound, is defined as $[\underline{x}\ ;\ \overline{x}] = \{x \in \mathbb{R} | \underline{x} \leq x \leq \overline{x}\}$. For any operation $\circ = \{+, -, \cdot, /\}$ and intervals $[\underline{x}\ ;\ \overline{x}]$, $[\underline{y}\ ;\ \overline{y}]$, the corresponding interval operation can be defined as $[\underline{x}\ ;\ \overline{x}] \circ [\underline{y}\ ;\ \overline{y}] = [\min(\underline{x} \circ \underline{y}, \underline{x} \circ \overline{y}, \overline{x} \circ \underline{y}, \overline{x} \circ \overline{y})\ ;\ \max(\underline{x} \circ \underline{y}, \underline{x} \circ \overline{y}, \overline{x} \circ \underline{y}, \overline{x} \circ \overline{y})]$. Note that the result of an interval operation is also an interval. Every possible combination $x \circ y$ with $x \in [\underline{x}; \overline{x}]$ and $y \in [\underline{y}; \overline{y}]$ lies inside this interval. For division of intervals, $0 \notin [\underline{y}\ ;\ \overline{y}]$ is assumed usually.

To be able to work with this definition on a computer using a finite precision arithmetic, the concept of machine intervals is necessary. The machine interval has floating point numbers as lower and upper bound. To obtain the corresponding machine interval for the real interval $[\underline{x}\ ;\ \overline{x}]$, the lower bound is rounded down to the largest representable machine number equal or less than $\underline{x}$, and the upper bound is rounded up to the smallest machine number equal or greater than $\overline{x}$. These notions can be extended further to define interval vectors and matrices.

Consider an algorithm for solving the initial value problem $\dot{x}(t) = f(x(t))$, $x(t_0) \in [x_0]$, where $t \in [t_0\ ;\ t_n] \subset \mathbb{R}$ for some $t_n > t_0$, $f \in C^{p-1}(\mathcal{D})$ for some $p > 1$, $\mathcal{D} \subseteq \mathbb{R}^m$ is open, $f : \mathcal{D} \mapsto \mathbb{R}^m$, and $[x_0] \subset \mathcal{D}$. The problem is discretized on a grid $t_0 < t_1 < \cdots < t_n$ with $h_{k-1} = t_k - t_{k-1}$. Denote the solution with the initial condition $x(t_{k-1}) = x_{k-1}$ by $x(t; t_{k-1}, x_{k-1})$ and the set of solutions $\{x(t; t_{k-1}, x_{k-1}) \mid x_{k-1} \in [x_{k-1}]\}$ by $x(t; t_{k-1}, [x_{k-1}])$. The goal is to find interval vectors $[x_k]$ for which the relation $x(t_k; t_0, [x_0]) \subseteq [x_k]$, $k = 1, \ldots, n$ holds.

The (simplified) k-th time step of the algorithm[a] consists of two stages:

1. *Proof of existence and uniqueness.* Compute a step size h_{k-1} and an *a priori* enclosure $[\tilde{x}_{k-1}]$ of the solution such that

[a] As implemented, for example, in VNODE.

(i) $x(t; t_{k-1}, x_{k-1})$ is guaranteed to exist for all $t \in [t_{k-1}; t_k]$ and all $x_{k-1} \in [x_{k-1}]$,

(ii) the set of solutions $x(t; t_{k-1}, [x_{k-1}])$ is a subset of $[\tilde{x}_{k-1}]$ for all $t \in [t_{k-1}; t_k]$.

2. *Computation of the solution.* Compute a tight enclosure $[x_k] \subseteq [\tilde{x}_{k-1}]$ of the solution of the initial value problem such that $x(t_k; t_0, [x_0]) \subseteq [x_k]$. The prevailing algorithm is as follows:
2.1. Choose a one-step method

$$x(t; t_k, x_k) = x(t; t_{k-1}, x_{k-1}) + h_{k-1}\varphi(x(t; t_{k-1}, x_{k-1})) + z_k,$$

where $\varphi(\cdot)$ is an appropriate method function, and z_k is the local error which takes into account discretization effects. The usual choice for $\varphi(\cdot)$ is a Taylor series expansion.
2.2. Find a guaranteed enclosure for the local error z_k. For the Taylor series expansion of order $p-1$, this enclosure is obtained as $[z_k] = h_{k-1}^p f^{[p]}([\tilde{x}_{k-1}])$, where $f^{[p]}([\tilde{x}_{k-1}])$ is an enclosure of the p-th Taylor coefficient of the solution over the state enclosure $[\tilde{x}_{k-1}]$ determined in Stage 1.
2.3. Compute a tight enclosure of the solution. If mean-value evaluation for computing the enclosures of the ranges of $f^{[i]}([x_k])$, $i = 1, \ldots, p-1$, instead of the direct evaluation of $f^{[i]}([x_k])$ is used, tighter enclosures can be obtained.

The theoretical notions outlined above are implemented in various libraries. We name some of them without the ambition to provide a complete list. First, there are tools for working with arithmetic operations and standard functions such as sine or cosine in a guaranteed way. Interval arithmetic is implemented in PROFIL/BIAS, FILIB++, C-XSC, INTLAB. LIBAFFA is a library for affine arithmetic, whereas COSY implements Taylor models.

Second, there are libraries for verified solution of systems of algebraic, differential or differential-algebraic equations. C-XSC TOOLBOX offers a general means of solving different classes of systems as well as an implementation in C-XSC. For solving initial value problems in interval arithmetic, there exist packages such as AWA, VNODE, and recently developed VALENCIA-IVP. In the framework of Taylor models, the solver COSY VI was developed.

Almost all of the above mentioned solvers need a means of computing Taylor coefficients or (high order) partial derivatives automatically. Some of them, for example, COSY VI, use the facilities provided by the basis arithmetic in COSY. Interval implementations do not possess this facility in general; external tools are necessary in this case. For this purpose, libraries such as FADBAD++, CPPAD, or ADOL-C can be employed.

For an overview of verified software libraries, application-oriented references, and current research activities, the reader is referred to the web page `http://www.cs.utep.edu/interval-comp/main.html`.

Basic notation

As a conclusion to this editorial, we provide a set of notations which the contributors used as far as possible throughout this special issue. As a general convention, small letters are reserved for scalars and vectors (e.g., a, x) while capital letters (e.g., A) for matrices.

Specification of interval variables

$[x]$	Interval enclosure of a real vector $x \in \mathbb{R}^n$
$\underline{x} = \inf\{[x]\}$	Infimum (lower bound) of the interval vector $[x] = [\underline{x}\ ;\ \overline{x}]$, i.e., $\underline{x}_i = \inf\{[x_i]\}$
$\overline{x} = \sup\{[x]\}$	Supremum (upper bound) of the interval vector $[x] = [\underline{x}\ ;\ \overline{x}]$, i.e., $\overline{x}_i = \sup\{[x_i]\}$
$[x] \circ [y]$	Interval operations for scalar operands $[x]$, $[y]$, $\circ \in \{+, -, \cdot, /\}$

$$\begin{aligned}[z] &:= [x] \circ [y] := \{x \circ y \mid x \in [x], y \in [y]\} \\ &= \{z \mid \min(\underline{x} \circ \underline{y}, \underline{x} \circ \overline{y}, \overline{x} \circ \underline{y}, \overline{x} \circ \overline{y}) \leq z \leq \max(\underline{x} \circ \underline{y}, \underline{x} \circ \overline{y}, \overline{x} \circ \underline{y}, \overline{x} \circ \overline{y})\}\end{aligned}$$

For the division of intervals, the case $0 \in [y]$ needs special treatment.

$\operatorname{diam}\{[x]\}$	Diameter of an interval vector $[x]$, defined component-wise, i.e., $\operatorname{diam}\{[x]\} := \overline{x} - \underline{x}$
$\operatorname{mid}\{[x]\}$	Midpoint of an interval vector $[x]$, defined component-wise, i.e., $\operatorname{mid}\{[x]\} := \frac{1}{2}(\underline{x} + \overline{x})$
$\operatorname{rad}\{[x]\}$	Radius of an interval vector $[x]$, defined component-wise, i.e., $\operatorname{rad}\{[x]\} := \frac{1}{2}(\overline{x} - \underline{x})$
$\operatorname{vol}\{[x]\}$	Pseudo volume of an interval box $[x]$ with $\operatorname{vol}\{[x]\} := \prod_{i=1}^{n} \operatorname{diam}\{[x_i]\}$, $x \in \mathbb{R}^n$
$V_f([x])$	Range of the function $f : \mathbb{R}^m \mapsto \mathbb{R}^n$ with $m, n \geq 1$ over the interval $[x] \in \mathbb{IR}^m$, i.e., $V_f([x]) := \{f(x) \mid x \in [x]\}$
$f([x])$	Any interval evaluation of the function f over the interval $[x]$, e.g. obtained by direct application of the interval operations defined above

Special sets and operators

$\mathbb{R}$	Set of real numbers
$\mathbb{IR}$	Set of all scalar real intervals, i.e., $\mathbb{IR} := \{[x] \mid \underline{x} \leq \overline{x} \text{ for all } \underline{x}, \overline{x} \in \mathbb{R}\}$
$\emptyset$	Empty set
$\subseteq$	Subset, i.e., $[a] \subseteq [b]$ means $x \in [b]$ for all $x \in [a]$
$\subset$	*True* subset, i.e., $[a] \subset [b]$ means $[a] \subseteq [b]$ and $[a] \neq [b]$
$\in$	Element of a set
$\cup$	Union operator for sets
$\sqcup$	Convex hull operation for intervals, i.e., $[x] \sqcup [y] := \left[\min\{\underline{x}, \underline{y}\}\ ;\ \max\{\overline{x}, \overline{y}\}\right]$, defined component-wise
$\cap$	Intersection operator for sets and intervals, i.e.,

$$[x] \cap [y] := \begin{cases} \left[\max\{\underline{x}, \underline{y}\}\ ;\ \min\{\overline{x}, \overline{y}\}\right] & \text{for } \max\{\underline{x}_i, \underline{y}_i\} \leq \min\{\overline{x}_i, \overline{y}_i\} \\ & \text{and all } i = 1, \ldots, n, \\ \emptyset & \text{otherwise} \end{cases}$$

Andreas Rauh
Chair of Mechatronics
University of Rostock, Germany

Ekaterina Auer
Faculty of Engineering, INKO
University of Duisburg-Essen, Germany

Eberhard P. Hofer
Institute of Measurement, Control,
and Microtechnology
University of Ulm, Germany

Wolfram Luther
Faculty of Engineering, INKO
University of Duisburg-Essen, Germany

September 2009