Report from Dagstuhl Seminar 11371

Uncertainty modeling and analysis with intervals: Foundations, tools, applications

Edited by

Isaac Elishakoff[1], Vladik Kreinovich[2], Wolfram Luther[3], and Evgenija D. Popova[4]

1 **Florida Atlantic University – Boca Raton, US**
2 **University of Texas – El Paso, US,** vladik@utep.edu
3 **Universität Duisburg-Essen, DE,** luther@inf.uni-due.de
4 **Bulgarian Academy of Sciences, BG,** epopova@bio.bas.bg

Abstract

This report documents the program and the results of Dagstuhl Seminar 11371 "Uncertainty modeling and analysis with intervals – Foundations, tools, applications", taking place September 11-16, 2011. The major emphasis of the seminar lies on modeling and analyzing uncertainties and propagating them through application systems by using, for example, interval arithmetic.

Seminar 11.–16. September, 2011 – www.dagstuhl.de/11371

1998 ACM Subject Classification G.1.0 General Computer arithmetic; Error analysis; Interval arithmetic; G.4 Mathematical Software Algorithm design and analysis; Verification; I.6.4 Model Validation and Analysis.

Keywords and phrases Uncertainty modeling – Propagation of uncertainty through and validation for computational models using interval arithmetic – Imprecise probabilities, Sensitivity analysis – Applications to structures, mechatronics, bioinformatics and finance

Digital Object Identifier 10.4230/DagRep.1.9.26

Edited in cooperation with Martin Fuchs (CERFACS – Toulouse, FR)

1 Executive Summary

Isaac Elishakoff
Vladik Kreinovich
Wolfram Luther
Evgenija Popova

Verification and validation (V&V) assessment of process modeling and simulation is increasing in importance in various areas of application. They include complex mechatronic and biomechanical tasks with especially strict requirements on numerical accuracy and performance. However, engineers lack precise knowledge regarding the process and its input data. This lack of knowledge and the inherent inexactness in measurement make such general V&V cycle tasks as design of a formal model and definition of relevant parameters and their ranges difficult to complete.

To assess how reliable a system is, V&V analysts have to deal with uncertainty. There are two types of uncertainty: aleatory and epistemic.

Aleatory uncertainty refers to variability similar to that arising in games of chance. It cannot be reduced by further empirical study. Epistemic (reducible) uncertainty refers to the incertitude resulting from lack of knowledge. An example is the absence of evidence about the probability distribution of a parameter. In this situation, standard methods for modeling measurement uncertainty by using probability distributions cannot be applied. Here, interval methods provide a possible solution strategy.

Another option, mostly discussed in the context of risk analysis, is to use interval-valued probabilities and imprecisely specified probability distributions. The probability of an event can be specified as an interval; probability bounds analysis propagates constraints on a distribution function through mathematical operations. In a more general setting, the theory of imprecise probabilities is a powerful conceptual framework in which uncertainty is represented by closed, convex sets of probability distributions. Bayesian sensitivity analysis or Dempster-Shafer theory are further options.

A standard option in uncertainty management is Monte Carlo simulation. This is a universal data-intensive method that needs random number generators, distributions, dependencies, and a mathematical model (but not a closed analytic solution) to provide accurate results. Compared to interval methods, it yields less conservative bounds, which, however, might fail to contain the exact solution. As an implementation of convolution in probability theory, Monte Carlo methods are complementary to interval approaches.

Additionally, they play an important role in probability bounds analysis, Dempster-Shafer theory, and further approaches combining probabilistic and interval uncertainties.

The goal of this seminar is to promote and accelerate the integration of reliable numerical algorithms and statistics of imprecise data into the standard procedures for assessing and propagating uncertainty. The main contributions of this seminar were

- Expressing, evaluating and propagating measurement uncertainties; designing efficient algorithms to compute various parameters, such as means, median and other percentiles, variance, interquantile range, moments and confidence limits; summarizing the computability of such statistics from imprecise data.
- New uncertainty-supporting dependability methods for early design stages. These include the propagation of uncertainty through dependability models, the acquisition of data from similar components for analyses, and the integration of uncertain reliability and safety predictions into an optimization framework.
- Modeling and processing applications from the areas of robust geometrical design, financial simulation and optimization, robotics, mechatronics, reliability and structural safety, bioinformatics and climate science with uncertain input parameters and imprecise data.
- Discussing software for probabilistic risk and safety assessments working with real numbers, intervals, fuzzy numbers, probability distributions, and interval bounds on probability distributions that combines probability theory and interval analysis and makes the newest techniques such as interval Monte Carlo method, probability bounds analysis and fuzzy arithmetic available.
- Promoting a new interval standard for interval arithmetic as explained in the P1788 draft: "This standard specifies basic interval arithmetic operations selecting and following one of the commonly used mathematical interval models and at least one floating-point type defined by the IEEE-754/2008 standard. Exception conditions are defined and standard handling of these conditions are specified. Consistency with the model is tempered with practical considerations based on input from representatives of vendors and owners of existing systems".

2 Table of Contents

3 Overview of Talks

The seminar was attended by 33 participants from 8 countries who gave 34 talks. To stimulate debate and cross-fertilization of new ideas we scheduled a mixture of tutorials, contributed talks, a meeting of the IEEE P1788 working group, and software demonstrations. The seminar started with a series of talks aimed at providing a suitable level of introduction to the main areas of discussion and providing a leveling ground for all participants.

The format of the seminar was then a series of contributed presentations on the variety of the seminar topics mentioned above. A lively discussion on the current state of the interval standardization was initiated by the talk on the hot topic of decorated intervals on Tuesday afternoon and continued during the meeting of the IEEE P1788 working group on Thursday afternoon. A session on software tools, held on Wednesday, was followed by software demonstrations on Thursday evening. There was much time for extensive discussions in between the talks, in the evenings, and during the excursion on Wednesday afternoon. The seminar had generally a very open and constructive atmosphere. As a result of the seminar there will be a special issue published in a leading journal that will not only publish papers presented at the seminar, but also provide a roadmap for the future directions of the uncertainty modeling.

3.1 Application of Verified Methods to Solving Non-smooth Initial Value Problems in the Context of Fuel Cell Systems

Ekaterina Auer (Universität Duisburg-Essen, DE)

Joint work of Auer, Ekaterina; Rauh, Andreas

In many engineering applications, there is a need to choose mathematical models that depend on non-smooth functions. For example, the models for friction or, more broadly, contact dynamics, are not even continuous in general. There are also less obvious situations that call for non-smooth functions, for instance, when naturally arising conditions such as non-positivity of variables have to be taken into account. The task becomes especially difficult if such functions appear on the right side of an initial value problem (IVP). Here, even the definition of the solution depends on the application at hand [2]. Since uncertainty in parameters obstructs many non-smooth tasks additionally, verified methods might prove themselves to be more useful than those from the usual numerics. Besides, they guarantee the correctness of the result within the limitations of a particular model.

The development of verified methods for IVPs with non-smooth right sides has got relatively few attention throughout the last three decades. To our knowledge, there exist no modern publicly available implementation at the moment. In [6], Rihm proposes a suitable definition and a method to enclose the solution to IVPs changing their right sides in dependence on a certain algebraic function. In [1, 4, 5], the authors propose algorithms for systems switching their representation according to graphs containing different ordinary differential equations as vertices and logical conditions as edges. Additionally, a lot of research has been done on generalizing the notion of a derivative for non-smooth functions in the area of verified optimization [3, 7].

In this talk, we give a short overview of the already existing methods for solving IVPs with non-smooth right sides. Next, we develop a generalized derivative definition for a certain type of continuous functions and its possible modification for bounded non-continuous functions of the same type. Then we use this definition inside the algorithm of the verified solver ValEncIA-IVP. We chose this solver because it needs only first order derivatives of the right side of an IVP. Finally, we demonstrate the applicability of our method in the context of modeling and simulation of high temperature fuel cells.

References

1 A. Eggers, M. Fränzle, and C. Herde. Application of Constraint Solving and ODE-Enclosure Methods to the Analysis of Hybrid Systems. In *Numerical Analysis and Applied Mathematics 2009*, volume 1168, pages 1326–1330. American Institute of Physics, 2009.
2 A. Filippov. *Differential Equations With Discontinuous Righthand Sides*. Kluwer Academic Publishers, 1988.
3 H. Munoz and R. B. Kearfott. Slope Intervals, Generalized Gradients, Semigradients, Slant Derivatives, and Csets. *Reliable Computing*, 10:163–193, 2004.
4 N. Nedialkov and M. von Mohrenschildt. Rigorous Simulation of Hybrid Dynamic Systems with Symbolic and Interval Methods. In *Proceedings of the American Control Conference Anchorage*, 2002.
5 A. Rauh, C. Siebert, and H. Aschemann. Verified Simulation and Optimization of Dynamic Systems with Friction and Hysteresis. In *Proceedings of ENOC 2011*, Rome, Italy, July 2011.
6 R. Rihm. Enclosing solutions with switching points in ordinary differential equations. In *Computer arithmetic and enclosure methods. Proceedings of SCAN 91*, pages 419–425. Amsterdam: North-Holland, 1992.
7 M. Schnurr. *Steigungen höherer Ordnung zur verifizierten globalen Optimierung.* PhD thesis, Universität Karlsruhe, 2007.

3.2 Fuzzy Probabilities and Applications in Engineering

Michael Beer (University of Liverpool, GB)

A key issue in computational engineering disciplines is the realistic numerical modeling of physical and mechanical phenomena and processes. This is the basis to derive predictions regarding behavior, performance, and reliability of engineering structures and systems. In engineering practice, however, the available information is frequently quite limited and of poor quality. A solution to this conflict is given with imprecise probabilities, which involve both probabilistic uncertainty and non-probabilistic imprecision. An entire set of plausible probabilistic models is considered in one analysis. This leads to more realistic results and helps to prevent wrong decisions. In this context fuzzy probabilities and their application in engineering were discussed in the presentation. Usefulness and benefits were demonstrated by means of various practical examples from different engineering fields.

3.3 Asymptotic Stabilization of a Bioprocess Model Involving Uncertainties

Neli Dimitrova (Bulgarian Academy of Sciences, BG)

The dynamic modeling of anaerobic digestion has recently become an active research area.

This is due to the fact that a mathematical model of the bioreactor can be used as a powerful tool to simulate different operating, control and optimization strategies.

One of the main drawbacks in the modeling and control of the anaerobic digestion lies in the difficulty to monitor on-line the key biological variables of the process and in estimating the expressions of the bacterial growth rates.

Thus developing control systems only based on simple measurements and minimal assumptions on the growth rates that guarantee stability of the process is of primary importance.

We consider the following model of a biological digestion process

$$\begin{aligned}
\frac{ds_1}{dt} &= u(s_1^i - s_1) - k_1\mu_1(s_1)x_1 \\
\frac{dx_1}{dt} &= (\mu_1(s_1) - \alpha u)x_1 \\
\frac{ds_2}{dt} &= u(s_2^i - s_2) + k_2\mu_1(s_1)x_1 - k_3\mu_2(s_2)x_2 \\
\frac{dx_2}{dt} &= (\mu_2(s_2) - \alpha u)x_2
\end{aligned} \tag{1}$$

with output

$$Q = k_4\mu_2(s_2)x_2,$$

where the phase variables s_1, s_2 and x_1, x_2 represent substrate and biomass concentrations respectively, $\mu_1(s_1)$ and $\mu_2(s_2)$ are bacterial growth rate functions, s_1^i and s_2^i are input substrate concentrations, u is the dilution rate (control input), Q is the methane flow rate, α is a homogeneity parameter and k_j, $j = 1, 2, 3, 4$, are coefficients.

This model has been investigated in [1], [2], where some control strategies are proposed and their robustness is illustrated mainly by simulation studies.

The present talk is an overview of authors' results on global asymptotic stabilizability of the dynamic model (1) by a feedback control law.

Practical applications impose the following requirements on the feedback law: dependance on online measurable variables, and robustness under uncertainties in the model coefficients and growth rate functions.

Let the growth rate functions satisfy the following *general assumption*: $\mu_j(s_j)$ is defined for $s_j \in [0, +\infty)$, $\mu_j(0) = 0$, $\mu_j(s_j) > 0$ for $s_j > 0$; $\mu_j(s_j)$ is continuously differentiable and bounded for all $s_j \geq 0$, $j = 1, 2$.

The first result concerns the so-called adaptive asymptotic stabilization of the control system (1).

We extend the system by the differential equation

$$\frac{d\beta}{dt} = -C(\beta - \beta^-)(\beta^+ - \beta)k_4\,\mu_2(s_2)\,x_2\,(s - \bar{s}), \tag{2}$$

where C, β^- and β^+ are appropriate positive constants, and $\bar{s}$ is a previously chosen operating point, representing the biological oxygen demand (which is on-line measurable). Under some additional (practically meaningful) conditions we have shown in [3] that the feedback control law

$$u \equiv k(s_1, x_1, s_2, x_2, \beta) = \beta\, k_4\, \mu_2(s_2)\, x_2 = \beta\, Q$$

stabilizes asymptotically the extended closed-loop system (1)–(2) to an equilibrium point, corresponding to the value $\bar{s}$, for each starting point $(s_1(0), x_1(0), s_2(0), x_2(0)) \geq 0$. The proposed feedback is robust with respect to model uncertainties only in the case of uncertain growth rates: we assume that instead of the exact functions $\mu_1(s_1)$ and $\mu_2(s_2)$ we know bounds for them, i. e. $\mu_1(s_1) \in [\mu_1(s_1)] = [\mu_1^-(s_1),\ \mu_1^+(s_1)]$, $\mu_2(s_2) \in [\mu_2(s_2)] = [\mu_2^-(s_2),\ \mu_2^+(s_2)]$. If any $\mu_j(s_j) \in [\mu_j(s_j)]$, $j = 1, 2$, satisfies the above general assumption, it is shown in [4] that the global stabilizability of the closed-loop system is retained.

Uncertainties with respect to the four coefficients k_j, $j = 1, 2, 3, 4$, are not considered.

Further recent results in [4] extend and improve the above studies. First we avoid the auxiliary differential equation (2), since it cannot be interpreted in terms of process dynamics; second, we assume that not only the growth rates, but also the coefficients k_j are unknown but bounded within compact intervals: $k_j \in [k_j] = [k_j^-,\ k_j^+]$, $j = 1, 2, 3, 4$.

Denote by $s^{i-} = \frac{k_2^-}{k_1^+} s_1^i + s_2^i$ a lower bound for the input biological oxygen demand. Let be $\beta \in \left(\frac{k_3^+}{s^{i-} \cdot k_4^-},\ +\infty \right)$. Take any value of $k_j \in [k_j]$, $j = 1, 2, 3, 4$, and define $\bar{s}_\beta = s^i - \frac{k_3}{\beta k_4}$. It is then proved in [4] that the feedback control law $k(s_1, x_1, s_2, x_2) = \beta\, k_4\, \mu_2(s_2)\, x_2 = \beta\, Q$ stabilizes asymptotically the closed-loop system to an equilibrium point, corresponding to $\bar{s}_\beta$ for each starting point $(s_1(0), x_1(0), s_2(0), x_2(0)) \geq 0$.

An important practical problem is the stabilization of the model (1) to a point, where maximum methane flow rate Q is achieved. This problem is also solved in [3] and [4] by designing an extremum seeking model-based algorithm. Computer simulations illustrate the theoretical studies.

References

1 L. Maillert, O. Bernard, and J.-P. Steyer. Robust regulation of anaerobic digestion processes. *Water Science and Technology*, 48(6):87–94, 2003.

2 F. Grognard and O. Bernard. Stability analysis of a wastewater treatment plant with saturated control. *Water Science and Technology*, 53:149–157, 2006.

3 N. Dimitrova and M. Krastanov. *Modeling, Design, and Simulation of Systems with Uncertainties*, volume 3 of *Mathematical Engineering*, chapter Nonlinear adaptive control of a bioprocess model with unknown kinetics, pages 275–292. Springer, 2011.

4 N. Dimitrova and M. Krastanov. On the asymptotic stabilization of an uncertain bioprocess model. To appear in Lecture Notes in Computer Science, 2012.

3.4 Robust optimization for aerospace applications

Martin Fuchs (Cerfacs – Toulouse, FR)

Many modern applications require a profound treatment of uncertainties. Two of the most critical issues to be dealt with are lack of statistical information and the well-known curse of dimensionality. One of the concerned fields is optimization for aerospace applications, where computational black box models include many uncertain parameters with little data and little information about statistical correlations given. The major research goal is to achieve optimal solutions that can be qualified as robust by accounting for the uncertainties in the high dimensional parameter space.

In lower dimensions, there are several tools to handle lack of information reliably, e.g., p-boxes, Dempster-Shafer structures, or possibility distributions. However, in higher dimensions it may require an intrusive implementation to be efficient to propagate uncertainties through a function. If the uncertainties are propagated through a black box function simulation techniques are often preferred, but they may fail to be reliable in many cases, see [1]. Sensitivity analysis can help to reduce the dimensionality at additional computational cost. The clouds formalism, see [5], combines concepts of intervals, fuzzy sets, and probability theory, in order to deal with both incomplete and higher dimensional information in a reliable and computationally tractable fashion.

Our approach first uses clouds to determine a polyhedral representation of the uncertainties. In other words, we describe the set, in which we search for worst-case scenarios, as a polyhedron. Methods to generate this polyhedron already exist, see [3]. In the second step we solve an optimization problem subject to polyhedral constraints to actually find the worst-case scenario. Our approach to the solution of the problem is inspired by the simulation based Cauchy deviates method for interval uncertainty, see [4]. It turns out to be computationally very attractive and it can be easily parallelized.

The new methods are employed in the context of robust optimization. The worst-case analysis of the previous steps becomes a constraint of an optimization problem formulation, see [2]. The objective is to find an optimum that is safeguarded against uncertain perturbations. To this end we determine the worst-case objective function, i.e., we propagate the uncertainties through the objective function. Thus the extra computational effort to account for robustness amounts to extra objective function evaluations. Hence it is important to use only very few evaluations as the total budget of evaluations in the optimization phase of real-life applications is typically very limited. This gives a new point of view of the Cauchy deviates method. Numerical tests are presented for applications from space system design and aircraft wing shape optimization.

References

1 S. Ferson, L. Ginzburg, and R. Akcakaya. Whereof one cannot speak: When input distributions are unknown. Web document, http://www.ramas.com/whereof.pdf, 1996.

2 M. Fuchs and A. Neumaier. Autonomous robust design optimization with potential clouds. *International Journal of Reliability and Safety*, 3(1/2/3):23–34, 2009.

3 M. Fuchs and A. Neumaier. Potential based clouds in robust design optimization. *Journal of Statistical Theory and Practice*, 3(1):225–238, 2009.

4 V. Kreinovich and R. Trejo. *Handbook of Randomized Computing*, chapter Error estimations for indirect measurements: randomized vs. deterministic algorithms for 'black-box' programs, pages 673–729. Kluwer, 2001.

5 A. Neumaier. Clouds, fuzzy sets and probability intervals. *Reliable Computing*, 10(4):249–272, 2004.

3.5 Verified Solution of Finite Element Models for Truss Structures with Uncertain Node Locations

Jürgen Garloff (HTWG Konstanz, DE)

URL http://www-home.htwg-konstanz.de/~garloff/

In our talk, we consider finite element models for mechanical truss structures where all of the physical model parameters are uncertain but bounded and are represented by intervals. In this case the application of the finite element method results in a system of parametric linear equations where the parameters vary within given intervals. We aim at finding tight bounds for the solution set of such a parametric system, the so-called parametric solution set. We first consider the case that the parameter dependency is rational and briefly report on a combination of software for the parametric residual iteration written by E. Popova in a *Mathematica* environment and our own software for the tight enclosure of the range of multivariate polynomials over a box.

Then not just the material parameters and applied loads, but also the positions of the nodes are assumed to be inexact and are represented by intervals, a case which does not seem to have previously been considered in the literature. In civil engineering, these uncertainties are often due to imperfections of the fabrication process. The application of the mentioned software for the enclosure of the parametric solution set results in intervals which are too wide for practical purposes. To contract the obtained intervals we employ interval pruning techniques. In the case of a statically indeterminate truss structure, the resulting intervals for the node displacements are still wider than we would like. Therefore, we employ a monotonicity analysis for all the parameters to provide tight guaranteed enclosures.

3.6 Interval Linear Programming: Foundations, Tools and Challenges

Milan Hladík (Charles University – Prague, CZ)

Many practical problems can be formulated in terms of linear programming, and in the others linear programming is sometimes used as a auxiliary technique. Interval liner programming studies such problems that are subject to uncertainties and there are given lower and upper bounds for uncertain quantities. This approach is superior the standard sensitivity analysis since it takes into account more complex perturbations of input data. Contrary to the stochastic programming approach one does not have to care about distributions of uncertain parameters; lower and upper estimates are enough.

We present a survey on interval linear programming according to the very recent paper [1]; just in press. We give a brief overview on the basic problems concerning feasibility, unboundedness and optimality for both weak and strong case, where *weak* means the property for some realization of interval data, and *strong* means validity for each realization. In particular, we list the time complexities (polynomial vs. NP-hard) of these problems. Then we turn our attention to the two fundamental problems studied. The first one is a problem of determining the optimal value range. Depending on the form of the interval linear program, some of the bounds can be computed by an ordinary linear program, but the others are NP-hard and only a formula using exponential number of ordinary linear programs is known. The second fundamental problem, which is very difficult and still challenging, is to find a tight enclosure of the optimal solution set. This becomes easy in the case of the so called basis stability, i.e., there is a basis optimal for each realization of interval data. Checking basis stability is also a computationally hard problem, but there are quite strong sufficient conditions that may be utilized. Eventually, we state some open problems.

References

1 Milan Hladík. *Linear Programming – New Frontiers in Theory and Applications*, 2nd chapter Interval linear programming: A survey. Nova Science Publisher, New York, 2011.

3.7 Intervals, Orders, and Rank

Cliff Joslyn (Pacific Northwest National Lab. – Seattle, US)

Joint work of Joslyn, Cliff; Hogan, Emilie

Intervals are ordered
Orders have intervals
Orders have ranks
Ranks are intervals

Intervals have order relations defined on them as an important operation. Also, partially ordered sets (posets) and lattices have intervals within them as important sub-structures. In traditional interval analysis, the set on which the intervals are drawn is the real numbers, a special ordered set which is a total order; and the ordering relation used between these real intervals is the "strong" interval order (one interval being entirely below another), in the context of an overall Allen's algebra.

But in our work in semantic databases and ontology management, it is the more general cases which are demanded. Specifically: 1) the intervals in question are valued in a finite, bounded, generally partially ordered set; and 2) when real intervals ARE used, the conjugate endpoint product order and subset orders are far preferable to the standard strong order (which isn't even really an ordering relation anyway). The attendant issues have implications for the foundations of interval analysis which we seek to explore with the group.

Depending on time, structure, and the interests of attendees, we can go into more or less depth on the following.

We begin by describing our use of large, finite, bounded posets to represent taxonomic semantic data structures for applications such as ontology clustering and alignment. We then consider the challenges presented by their layout and display.

Our first challenge, the vertical layout of nodes, we have been working on for a while. We observe that rank in posets is best considered as being valued on integer intervals. These integer-valued rank intervals can themselves in turn be ordered (in the endpoint product order), so that an iterative operation is available. Repeated application serves to identify a privileged embedding of the poset to a total preorder preferred to reflect the underlying partial order. We have results about how the height, width, and dimension of the poset changes in repeated application, and prove that we do achieve a final embedding of the original poset to a total preorder. In the process, results about measures of gradedness of posets are also motivated.

Our second challenge we are just beginning to explore, but we will present some preliminary ideas for discussion. We seek to simplify the display of large posets (actually lattices in the first treatment) when only a subset of nodes are specified by a user. A tremendous reduction in complexity can be available when (poset) intervals among the target nodes are identified which are disjoint (pairwise or moreso). The underlying mathematical representation suggested is the graph of the intersection structure of poset intervals, that is, a generalization of an interval graph to poset intervals. Cliques in this graph determine the number of total meets and joins which need to be displayed in the reduced visualization, and thus the amount of compression achievable. But, this approach requires us to have the ordering relation between pairs of poset intervals, that is, to develop an Allen's algebra generalized to poset intervals.

3.8 Interval Computations – Introduction and Significant Applications

Ralph Baker Kearfott (Univ. of Louisiana – Lafayette, US)

In this tutorial, we first outline the historical motivations and early work in interval arithmetic, then review basic interval arithmetic operations and their properties, including advantages and pitfalls. We conclude with a variety of examples of successful application of interval arithmetic.

Historical motivation and work includes

1931 — Classical analysis: Rosaline Cecily Young developed interval arithmetic to handle analysis of one-sided limits (where $\liminf f \neq \limsup f$) [15].

1951 — Roundoff error analysis: Paul S. Dwyer developed interval arithmetic in the chapter on roundoff error analysis in his numerical analysis text [3].

1956 — Calculus of Approximations: Warmus and Steinhaus developed interval arithmetic to provide a sound theoretical backing to numerical computation [14].

1958 — Automatic error analysis: Teruro Sunaga developed interval arithmetic [11].

1959 — Automatic error analysis: Ray Moore developed interval arithmetic in a report and dissertation to which most modern work on the subject can be traced [5].

Advantages of interval arithmetic include the ability to quickly compute mathematically rigorous bounds on roundoff error and on ranges of functions, where computation of the exact range is NP-hard. Disadvantages are that these bounds may be unusably pessimistic, unless special algorithms are designed.

Current successful significant applications include the following:

A filter in branch and bound methods in leading commercial software, such as [9] (and others).

Constraint solving and constraint propagation, as in [8, §14] and numerous other works.

Verified solution of ODEs, as in [1], [4], etc.

Computer-aided proofs, as in [13], [12].

Chemical engineering, as in [10].

PDE problems such as structural analysis with uncertainties [6], [7] or analysis of photonic crystals [2].

Numerous others.

References

1 Martin Berz. COSY INFINITY web page, 2000. cosy.pa.msu.edu.
2 W. Dörfler, A. Lechleiter, M. Plum, G. Schneider, and C. Wieners. *Photonic Crystals: Mathematical Analysis and Numerical Approximation.* Oberwolfach Seminars. Birkhauser Verlag AG, 2011.
3 P. S. Dwyer. Computation with approximate numbers. In P. S. Dwyer, editor, *Linear Computations*, pages 11–35, New York, 1951. Wiley & Sons Inc.
4 Youdong Lin, Joshua A. Enszer, and Mark A. Stadtherr. Enclosing all solutions of two-point boundary value problems for ODEs. *Computers in Chemical Engineering*, 2008. (in press).
5 R. E. Moore and C. T. Yang. Interval analysis I. Technical Document LMSD-285875, Lockheed Missiles and Space Division, Sunnyvale, CA, USA, 1959.
6 R. L. Muhanna and R. L Mullen. Uncertainty in mechanics problems — interval-based approach. *Journal of Engineering Mechanics*, 127(6):557–566, 2001.
7 Rafi L. Muhanna, Hao Zhang, and Robert L. Mullen. Interval finite elements as a basis for generalized models of uncertainty in engineering mechanics. *Reliable Computing*, 13:173–194, 2007.
8 Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. In A. Iserles, editor, *Acta Numerica 2004*, pages 271–369. Cambridge University Press, 2004.
9 Nikolaos V. Sahinidis. BARON: A general purpose global optimization software package. *J. Global Optim.*, 8(2):201–205, 1996.
10 Mark A. Stadtherr. Interval analysis: Application to chemical engineering design problems. In Arieh Iserles, editor, *Encyclopedia of Optimization.* Kluwer Academic Publishers, 2001.
11 Teruo Sunaga. Theory of interval algebra and its application to numerical analysis. *RAAG Memoirs*, 2:29–46, 1958.
12 George G. Szpiro. *Kepler's Conjecture: How Some of the Greatest Minds in History Helped Solve One of the Oldest Math Problems in the World.* E-Book, June 2003.
13 Warwick Tucker. A rigorous ODE solver and Smale's 14th problem. *Found. Comput. Math.*, 24:53–117, 2002.
14 M. (Mieczysław) Warmus and H. Steinhaus. Calculus of approximations. *Bulletin de l'Academie Polonaise des Sciences, Cl. III*, IV(5):253–259, February 1956.
15 R. C. Young. The algebra of many-valued quantities. *Math. Ann.*, 104:260–290, 1931.

3.9 Integration of Interval Contractors in Hierarchical Space Decomposition Structures

Stefan Kiel (Universität Duisburg-Essen, DE)

Hierarchical spatial data structures can be used for decomposing geometric objects into simpler primitives [5]. Often used primitives are axis-aligned boxes. Intervals are a natural choice for representing them. Furthermore, interval arithmetic (IA) [1] offers us a way to construct a verified decomposition enclosing the object and to cope with uncertainties in the original model.

However, classical IA often suffers from overestimation which might make object enclosures too wide. Recently, we have presented the framework UniVerMeC (Unified Framework for Verified Geometric Computations) [3] that allowed us to employ more sophisticated arithmetics like affine arithmetic or Taylor models to reduce the overestimation. Another way to tighten enclosures is to use contractors which identify parts of the decomposition disjoint with the object.

In this talk, we will discuss how UniVerMeC helps us to integrate arbitrary interval contractors into the trees. This is a direct extension of our recent approach [4] which was limited formerly to implicit linear interval estimations [2]. The direct integration into the trees lets an algorithm take advantage of contractors' properties without being aware of their actual use. This decoupling allows us to add or remove different contractors easily. In contrast to domain decompositions used, for example, in global optimization, interval trees have to cover the whole area by nodes properly. That is, we cannot dispose parts not containing the object but have to cover them with WHITE nodes indicating that they are empty. This makes employing interval contractors in trees complicated. We will show how to simplify this task by introducing special inversion nodes into the standard set. This does not change existing tree algorithms because nodes of the new type can be converted exactly into a set of standard nodes.

References

1 G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press, 1983.

2 K. Bühler. Implicit linear interval estimations. In *Proceedings of the 18th spring conference on Computer graphics*, page 132. ACM, 2002.

3 E. Dyllong and S. Kiel. A comparison of verified distance computation between implicit objects using different arithmetics for range enclosure. In *Proceedings of SCAN'2010*. Submitted.

4 S. Kiel. Verified spatial subdivision of implicit objects using implicit linear interval estimations. In *Curves and Surfaces 2011*, volume 6920 of *LNCS*. Springer, 2011. To appear.

5 H Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, San Francisco, 2006.

3.10 Degree-Based (Interval and Fuzzy) Techniques in Math and Science Education

Olga M. Kosheleva (University of Texas – El Paso, US)

In education, evaluations of the student's knowledge, skills, and abilities are often subjective. Teachers and experts often make these evaluations by using words from natural language like "good", "excellent". Traditionally, in order to be able to process the evaluation results, these evaluations are first transformed into exact numbers. This transformation, however, ignores the uncertainty of the original estimates. To get a more adequate picture of the education process and education results, it is therefore desirable to transform these evaluations into intervals – or, more generally, fuzzy numbers.

We show that this more adequate transformation can help on all the stages of the education process: in planning education, in teaching itself, and in assessing the education results.

Specifically, in planning education and in teaching itself, interval and fuzzy techniques help us:

- better plan the order in which the material is presented and the amount of time allocated for each topic;
- interval and fuzzy techniques help us find the most efficient way of teaching inter-disciplinary topics;
- these techniques also help to stimulate students by explaining historical (usually informal) motivations – often paradox-related motivations – behind different concepts and ideas of mathematics and science.

In assessment, interval and fuzzy techniques help:

- to design a better grading scheme for test and assignments, a scheme that stimulates more effective learning,
- to provide a more adequate individual grading of contributions to group projects – by taking into account subjective estimates of different student distributions (and the uncertainty of these estimates), and
- to provide a more adequate description of the student knowledge and of the overall teaching effectiveness.

The talk summarizes, combines, and expands on the ideas and results, some of which published in journals and conference proceedings. These published papers also contain additional technical details and practical examples of using these ideas.

3.11 A Comparison of Different Kinds of Multiple Precision and Arbitrary Precision Interval Arithmetics

Walter Krämer (Universität Wuppertal, DE)

Joint work of Krämer, Walter; Blomquist, Frithjof; Hofschuster, Werner
URL http://www2.math.uni-wuppertal.de/org/WRST/index_de.html

The current version of the C++ class library C-XSC for verified numerical computing offers quite a lot of different interval data types with different properties. We compared multiple precision data types like staggered precision data types (unevaluated sums of floating-point numbers) as well as arbitrary precision types based on arrays of integers and integer operations. In some respects staggered numbers and operation are restricted by properties of the underlying basic floating-point data type (IEEE double precision) whereas arbitrary precision numbers are only limited by the memory resources available.

We presented some preliminary execution time comparisons and gave some advice when it is appropriate to use a specific multiple/arbitrary precision C-XSC data type. We also discussed the availability of some underlying external packages restricting the use of C-XSC's arbitrary precision data types on some platforms. Several source code examples were presented to demonstrate the ease of use of the data types (due to operator and function name overloading) and the power of the different multiple/arbitrary precision C-XSC packages.

Further features of C-XSC have been discussed in the talk "C-XSC – Overview and new developments" presented by Michael Zimmer during the course of this Dagstuhl seminar. Please refer to the corresponding abstract included in this document.

Keywords: Multiple precision, arbitrary precision, staggered data types, C-XSC, MPFR, MPFI, interval computations.

References

1 E. Adams and U. Kulisch. *Scientific Computing With Automatic Result Verification.* Academic Press, Inc., 1993.
2 F. Blomquist, W. Hofschuster, and W. Krämer. A modified staggered correction arithmetic with enhanced accuracy and very wide exponent range. In *Lecture Notes in Computer Science LNCS 5492*, A. Cuyt, W. Krämer, W. Luther, and P. Markstein (Eds.), pages 41–67, Springer, 2009.
3 L. Fousse, G. Hanrot, V. Lefèvre, P. Pelissier, and P. Zimmermann. MPFR: A multiple-precision binary floating-point library with correct rounding. *ACM Transactions on Mathematical Software*, 33(2), Article 13, 2007.
4 M. Grimmer, K. Petras, and N. Revol. Multiple precision interval packages: Comparing different approaches. In *Numerical Software with Result Verification, Lecture Notes in Computer Science*, Vol. 2991, R. Alt, A. Frommer, R.B. Kearfott, W. Luther (Eds.), pages 64–90, Springer, 2004.
5 W. Hofschuster and W. Krämer. C-XSC 2.0: A C++ library for extended scientific computing. In *Numerical Software with Result Verification, Lecture Notes in Computer Science*, Volume 2991, Springer, pages 15–35, 2004.
6 R. Klatte, U. Kulisch, A. Wiethoff, C. Lawo, and M. Rauch. *C-XSC – A C++ Class Library for Extended Scientific Computing.* Springer, 1993.
7 W. Krämer. Multiple precision computations with result verification. In *Scientific Computing With Automatic Result Verification*, Academic Press, pages 325–356, 1993.
8 W. Krämer, U. Kulisch, and R. Lohner. *Numerical Toolbox for Verified Computing II – Advanced Numerical Problems (draft)*, chapter 7 Multiple-precision arithmetic using integer operations, pages 210–251, 1998. http://www2.math.uni-wuppertal.de/wrswt/literatur/tb2.ps.gz
9 R. Lohner. Interval arithmetic in staggered correction format. In *Scientific Computing With Automatic Result Verification*, Academic Press, pages 301–342, 1993.
10 N. Revol and F. Rouillier. Motivations for an arbitrary precision interval arithmetic and the MPFI library. *Reliable Computing*, 11:275–290, 2005.
11 F. Blomquist, W. Hofschuster, and W. Krämer. C-XSC-Langzahlarithmetiken für reelle und komplexe Intervalle basierend auf den Bibliotheken MPFR und MPFI. Preprint 2011/1, Universitaet Wuppertal, 2011.

3.12 Towards Optimal Representation and Processing of Uncertainty for Decision Making, on the Example of Economics-Related Heavy-Tailed Distributions

Vladik Kreinovich (University of Texas – El Paso, US)

Uncertainty is usually gauged by using standard statistical characteristics: mean, variance, correlation, etc. Then, we use the known values of these characteristics (or the known bounds on these values) to select a decision. Sometimes, it becomes clear that the selected characteristics do not always describe a situation well; then other known (or new) characteristics are proposed. A good example is description of volatility in finance: it started with variance, and now many descriptions are competing, all with their own advantages and limitations.

Another good example is the case of heavy-tailed distributions frequently occurring in economics and finance: for these distributions, variance is infinite and thus, we cannot use variance to describe deviations from the mean.

In such situations, a natural idea is to come up with characteristics tailored to specific application areas: e.g., select the characteristic that maximize the expected utility of the resulting risk-informed decision making. As a case study, we found optimal characteristics for measures of deviation and dependence in financial applications – where, for heavy-tailed distributions, traditional variance and correlation cannot be used.

With the new characteristics, comes the need to estimate them when the sample values are only known with interval uncertainty. Algorithms originally developed for estimating traditional characteristics can often be modified to cover new characteristics.

3.13 From Processing Interval-Valued Data to Processing Fuzzy Data: A Tutorial

Vladik Kreinovich (University of Texas – El Paso, US)

Some quantities y are difficult (or impossible) to measure or estimate directly. A natural solution is to measure them indirectly, i.e., to measure auxiliary quantities $x_1, \ldots, x_n$ which are related to y by a known dependence $y = f(x_1, \ldots, x_n)$, and then use the results $\widetilde{x}_i$ of these measurements to compute an estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ for y. Measurements are never absolutely accurate; as a result, the measurement results $\widetilde{x}_i$ are, in general, different from the actual (unknown) values x_i of the corresponding quantities. Hence, the estimate $\widetilde{y}$ is, in general, different from the desired value y. How can we estimate the difference $\Delta y = \widetilde{y} - y$? In some cases, we know the probabilities of different measurement errors $\Delta x_i = \widetilde{x}_i - x_i$; however, often, we do not know these probabilities, we only know the upper bounds Δ_i on the measurement errors – upper bounds provided by the manufacturers. In this case, after we know the measurement result $\widetilde{x}_i$, we know that the actual value x_i belongs to the interval $[\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$, and we can use interval computations to find the interval of possible values of y.

Often, in addition to the guaranteed bounds Δ_i on the measurement errors, we also have expert estimates on these bounds which hold only with some degree of certainty – and which are described by words from natural language like "small", "most probably", etc,. Fuzzy techniques were specifially designed to process such estimates. The main idea is that, for each "fuzzy" (natural-language) property like "small", and for each real value x, we describe the degree $d(x)$ to which x satisfies this property – e.g., by polling experts. Once we know the degrees $d_i(x_i)$ to which each x_i satisfies the corresponding expert property, we need to combine these degrees into a degree with which all the values x_i satisfy their properties. We show how this need leads to a complex formula called Zadeh's extension principle, and how from the computational viewpoint, it means that for each real value $\alpha \in [0, 1]$, to find all the values for which $d(y) \geq \alpha$ ("α-cut" of y), we can apply interval computations to the corresponding α-cuts of x_i. Thus, from the computational viewpoint, processing such fuzzy inputs can be (and usually is) reduced to interval computations.

3.14 Generating a Minimal Interval Arithmetic Based on GNU MPFR

Vincent Lefèvre (ENS – Lyon, FR)

Searching for the hardest-to-round cases for the correct rounding of some function f on an interval I in a fixed precision can be done efficiently by first approximating the considered function by a polynomial, on which specific algorithms are then applied. One also needs to determine an enclosure of the range $f(I)$, more precisely the exponent range.

Our implementation currently uses Maple and the intpakX interval arithmetic package in order to compute both the exponent range and the polynomial approximation. But Maple/intpakX has various drawbacks.

The GNU MPFR library has since been available and could be used for our computations in arbitrary precision. But we need an interval arithmetic on top of it. As reliability matters more than performance in this context, we seek to implement a minimal interval arithmetic by generating code on the fly using MPFR. The implementation should be as simple as possible so that it could easily be checked and/or proved formally.

3.15 IPPToolbox – a package for imprecise probabilities in R

Philipp Limbourg (Universität Duisburg-Essen, DE)

Joint work of Limbourg, Philipp; Rebner, Gabor; Auer, Ekaterina; Luther, Wolfram

A lot of researchers working on Dempster-Shafer, imprecise probabilities etc., e. g. in systems reliability engineering, structural reliability and similar fields. While the theory stems from the 60s & 70s, it was originally mainly used for expert systems & automatic reasoning. Imprecise probabilities were brought into uncertainty modelling practice by Sandia laboratories in US (2004).

However, most people use their own, proprietary codes. This talk presents an R and Matlab package – the IPP toolbox, an open-source package for imprecise probability calculations. The package includes R help and two examples (Fflood, Reliability). Available on CRAN (package "ipptoolbox"), it is the predecessor of the DSI toolbox Application areas: 100+ downloads of Matlab version (R figures not known, at CRAN), applied at Electricité de France R&D, e. g. flood modelling, event trees, design optimization.

3.16 Constrained Intervals and Interval Spaces

Weldon A. Lodwick (University of Colorado, US)

Constrained intervals, intervals as a mapping from [0,1] to linear functions with non-negative slopes, and arithmetic on constrained intervals generate a space that turns out to be a cancellative Abelian monoid albeit with a richer set of properties than standard interval arithmetic. This means that not only do we have the classical embedding as developed by H.

Radström and S. Markov but directly the properties of the subset of these polynomials. We study a little of the geometry of the embedding of intervals into a quasi-vector space and some of the properties of the mapping of constrained intervals into a space of polynomials.Thus, there are two parts to this talk. (1) The representation of intervals as linear polynomials with non-negative slopes. (2) The algebraic structure of this new representation. The geometry is mentioned in passing as a way to visualized the embedded space and will not be discussed further. The theoretical reason for considering a new representation of intervals is to have a formalization in (a subset of) polynomial space with the view to evaluate expressions (functions) of intervals. The theoretical reason for considering the algebraic structure of the embedding into a space with inverses is to solve equations. We only look at additive inverses in this presentation.

3.17 Verification and Validation Requirements in Biomechanical Modeling and Simulation

Wolfram Luther (Universität Duisburg-Essen, DE)

Joint work of Luther, Wolfram; Auer, Ekaterina; Chuev, Andrey; Cuypers, Roger; Kiel, Stefan
URL http://www.scg.inf.uni-due.de/fileadmin/Veroeffentlichungen/papereuromech.pdf

We give an overview on how accurate and verified methods can be employed for several biomechanical processes. Our focus is on a broad field of patient specific preoperative surgical planning based on the superquadric (SQ) geometrical modeling. We show how to use verified tools to efficiently and reliably implement important parts of the processing pipeline. Furthermore, we describe our current activities to verify SQ-based operations numerically and to validate the models and their parameters by various measurement methods. For this purpose, we replace or combine data types, algorithms and tools with their verified versions where possible. Here, we might consider using C-XSC (Krämer) instead of plain C++, INTLAB (Rump) in addition to Matlab, DSI (Dempster Shafer tool with intervals) instead of IPP (the Imprecise Probabilities toolbox, Limbourg, Rebner) or employing stochastic data types as in Cadna++ (Lamotte et al.) or the static program analyzer Fluctuat (CEA).

Recently, we introduced four classes for the use in V&V assessment, from lowest to highest certification standard. A process implementation that relies on standard floating-point or fixed-point arithmetic with unverified results belongs to Class 4. If the system is to qualify for Class 3, the numerical implementation of the process needs to employ at least standardized IEEE 754-2008 floating-point arithmetic. Furthermore, sensitivity analysis has to be carried out for uncertain parameters and uncertainty propagated throughout the subsystems using various methods. Additionally, a priori/posteriori error bounds should be provided for important sub-processes, condition numbers computed, failure conditions identified. To belong to Class 2, relevant subsystems have to be implemented using tools with result verification or with an accompanying computation of reliable error bounds. The tools should use language extensions, the convergence of numerical algorithms must be proved via existence theorems, analytical solutions, computer-aided proofs or fixed-point theorems. In Class 1, uncertainty is quantified and propagated throughout the process using interval computing. Model parameters are optimized by calibration. The whole system is verified using tools with result verification. Basic numeric algorithms and (special) functions are certified. Alternatively, real number algorithms, analytical solutions or computer-aided existence proofs are used.

We managed to classify several biomechanical processes. Recently, a dynamical gait simulation based on motion tracking under uncertainty in parameters was described. Processes from a recently completed project PROREOP have been used to perform elements of V&V assessment during the designing step.

Our analysis shows that such assessment should begin with the specification of the process and its sub-processes, the design of the building-blocks and their software modules, the definition of interfaces and data flows, and, finally, the selection and adaptation of appropriate data types and algorithms. PROREOP aimed at developing and evaluating a highly interactive prosthesis planning tool that allowed surgeons to assess 3D imaging data and to use geometrical, mechanical, kinematical and material/surface-specific bone features of the patient as primary sources for their decisions.

In our talk, we focus on an example addressing superquadric (SQ) bone/prosthesis modeling, prosthesis fitting into the medullary space of the routed femoral shaft and the total hip arthroplasty (THA), which was broadly discussed in the PhD-thesis of the fourth author. The following hardware and software blocks are analyzed: Data acquisition by MRI and CT imagery, bone and muscle segmentation (Class 3), SQ modeling with an in/out decision algorithm, distance computation between convex SQs (both Class 1), compound model optimization (Class 3), K-segments algorithms (Class 2), feature extraction with various validation approaches, verified distance computation between compound models (both Class 1), and pose computation (Class 3).

3.18 Enclosing solutions of initial-value problems with large uncertainty

Arnold Neumaier (Universität Wien, AT)

Joint work of Neumaier, Arnold; Fazal, Qaisra

We consider the enclosure of initial-value problems for ordinary differential equations where the initial condition has a large uncertainty. Uncertainties are described at each time by means of enclosing ellipsoids. Our new approach combines

- work by Chernousko for linear ODEs, who derived for this class differential equations for the parameters of the enclosing ellipsoid,
- work by Kühn based on defect estimates and curvature bounds,
- new results on conditional differential inequalities for validating error bounds, and
- global optimization techniques for verifying the assumptions needed to apply the conditional differential inequalities.

The approach was implemented in Matlab, using automatically generated AMPL files as an interface to optimization algorithms. For simple examples, the performance was essentially the same as the state-of-the-art packages VSPODE, VNODE-LP, and VALENCIA-IVP, while for highly nonlinear problems, it gave much superior error bounds.

3.19 Characterizing AE Solution Sets to Parametric Linear Systems

Evgenija D. Popova (Bulgarian Academy of Sciences, BG)

Consider linear systems whose input data are linear functions of uncertain parameters varying within given intervals. Such systems are common in many engineering analysis or design problems, control engineering, robust Monte Carlo simulations, etc., where there are complicated dependencies between the model parameters which are uncertain. Various solution sets to a parametric linear system can be defined depending on the way the parameters are quantified by the existential and/or the universal quantifiers.

We are interested in an explicit description of the so-called AE parametric solution sets (where all universally quantified parameters precede all existentially quantified ones) by a set of inequalities not involving the parameters. The problem is related to quantifier elimination where Tarski's general theory is EXPSPACE hard and a lot of research is devoted to special cases with polynomial-time decidability.

In this talk we present how to obtain an explicit description of AE parametric solution sets by combining a modified Fourier-Motzkin-type elimination of existentially quantified parameters with the elimination of the universally quantified parameters.

Some necessary (and sufficient) conditions for existence of non-empty AE parametric solution sets are discussed, as well as some properties of the AE parametric solution sets.

Explicit description of particular classes of AE parametric solution sets (tolerable, controllable, any 2D) is presented.

Numerical examples illustrate the solution sets and their properties. A comparison to results obtained by quantifier elimination demonstrates the advantage of the presented approach.

3.20 What you always wanted to know about decorated intervals

John D. Pryce (Cardiff University, GB)

Work on an Interval Arithmetic Standard has been under way since 2008, in IEEE Working Group P1788. Early on, we chose a "silent" paradigm for function evaluation, namely interval evaluation of a library function partly or wholly outside its domain returns an enclosure of those values that are defined (e.g. 1/[0,1] gives [1,infinity); sqrt([-2,-1]) gives the empty set), instead of throwing an exception as in current interval systems.

This raises the question of how to record that such an exceptional event has occurred. This is necessary, since various important interval algorithms need to determine rigorously whether a function, given by an expression, has properties such as being defined, or defined and continuous, everywhere on a box. Examples are branch-and-bound search methods to "pave" space regions defined by inequalities; or validated ODE methods that apply Brouwer's fixed-point theorem. In current interval systems, such information can only be determined to a limited extent, and by clumsy ad hoc means.

Summarising discussions over the past 15 months, this talk aims to show why and how it is feasible to record such data automatically by suitably enhancing the interval versions of library functions. The data might be stored in global flags similar to the IEEE floating point flags "zerodivide", etc.; or locally by attaching it as "decorations" to computed intervals. It explains why we favour the latter. It presents the Neumaier-Hayes idea of arranging decoration information in a linearly ordered sequence of values that can be considered to go from "best" to "worst", and compares several such schemes.

It discusses some problematic and contentious issues such as the status of intersection/union operations. It outlines the proposed Compressed Intervals, which gain speed at the cost of limited decoration support. They use the same space (typically 16 bytes) as undecorated intervals, and suffice for the applications mentioned above.

3.21 Verified Parameter Estimation for the Thermal Behavior of High-Temperature Fuel Cells

Andreas Rauh (Universität Rostock, DE)

Joint work of Rauh, Andreas; Dötschel, Thomas; Auer, Ekaterina; Aschemann, Harald

The thermal behavior of high-temperature solid oxide fuel cell (SOFC) systems is characterized by the interaction of different physical and electro-chemical processes. In particular, these processes take place in the anode and cathode gas manifolds as well as in the interior of the fuel cell stack module.

The chemical reactions of hydrogen at the anode and oxygen at the cathode have to be enabled to generate electricity and process heat simultaneously. The prerequisite for these reactions is both ion conduction through the electrolyte and electron interchange between the electrodes [1]. The fuel cell stack module itself consists of a thermal insulation and an assembly of anode-electrolyte-cathode elements, which are electrically connected in series.

One possible future application of fuel cells is the use in decentralized supply systems for process heat and electricity. For this type of application, it is necessary to operate SOFCs with a time-varying electric load. It is desired to implement operating strategies for variable electric loads with smallest possible battery buffers acting as electric load shaping devices. This leads to the necessity for advanced control strategies minimizing the influence of electric load variations on the resulting changes of the cell temperature and its local distribution.

The decoupling of electric load variations from the thermal behavior of the SOFC is crucial from an application point of view: Mechanical strain introduced by large spatial gradients in the temperature distribution within the fuel cell stack module along with local over-temperatures may lead to an accelerated degradation and — in the worst case — to the destruction of the fuel cell materials. Moreover, the maximization of the efficiency of a high-temperature SOFC is commonly linked to an increase of the overall temperature level in the interior of the stack module [2]. This fact imposes further demands on the accuracy of the mathematical system model used for control synthesis.

The reliable operation of SOFC systems by means of accurate control strategies requires a sufficient knowledge of the ongoing physical processes. Therefore, these processes are expressed in terms of a mathematical system model with explicitly given bounds for the uncertain quantities. This system model has to be usable for the online analysis and prediction of the influence of electric load variations and for its compensation in real time by means of nonlinear feedback control strategies. The same holds for the online estimation and identification of non-measured internal process variables with the help of state and disturbance observers.

The real-time applicability of the mathematical system models can be achieved by using control-oriented descriptions which take into account the dominating dynamic effects and spatial variations of process variables such as temperatures and partial pressures of the gas fractions. These control-oriented descriptions commonly replace sets of (nonlinear) partial differential equations by finite-dimensional sets of ordinary differential equations and algebraic relations [4].

To parameterize these low-dimensional, control-oriented system models, identification routines are employed. The parameter identification is based on experimental data gathered from a test rig available at the Chair of Mechatronics at the University of Rostock. Classical floating point approaches for parameter identification (which are based on local optimization procedures) cannot be used to incorporate uncertainty in measured data directly in the identification procedure.

For this reason, a first version of a verified interval-based parameter identification routine is presented in this contribution. This routine is applied to determine those parameter ranges which are consistent with measured data under explicit consideration of their uncertainty. The interval-based identification routine is a generalization of a verified optimization procedure developed by the authors for the design of reliable optimal controllers for uncertain continuous-time and discrete-time processes [3]. Moreover, the identification results obtained by the above-mentioned interval-based routine are compared with a classical floating point implementation for the thermal subsystem of the SOFC currently involving a total number of 26 a-priori unknown or uncertain parameters.

A simplified thermal system model is given by the scalar nonlinear differential equation

$$\dot{\vartheta}_{FC} = \frac{1}{c_{FC}\ m_{FC}} \left[\frac{1}{R_A}(\vartheta_A - \vartheta_{FC}) - (c_{N_2}\zeta_{N_2,C} + c_{O_2}\zeta_{O_2}) \cdot \dot{m}_{CG} \cdot (\vartheta_{FC} - \vartheta_{CG,in}) \right.$$
$$\left. - (c_{H_2}\dot{m}_{H_2} + c_{H_2O}\dot{m}_{H_2O} + c_{N_2}\dot{m}_{N_2,A}) \cdot (\vartheta_{FC} - \vartheta_{AG,in}) + \frac{\Delta H_m(\vartheta_{FC})\dot{m}_{H_2}}{M_{H_2}} \right] \ ,$$

which describes the temperature ϑ_{FC} in the fuel cell stack. This temperature is assumed to be homogeneously distributed. It depends on the mass flow rate $\dot{m}_{CG}$ of cathode gas (CG) as well as the stoichiometrically balanced mass flow $\dot{m}_{H_2}$ of hydrogen, $\dot{m}_{H_2O}$ of vaporized water, and $\dot{m}_{N_2,A}$ of nitrogen at the anode. The inlet temperature of the cathode gas is given by $\vartheta_{CG,in}$, the corresponding anode inlet temperature is denoted by $\vartheta_{AG,in}$. Second-order polynomial approximations $c_x(\vartheta_{FC}) = \sum_{i=0}^{2} \alpha_{x,i}\vartheta_{FC}^i$ are used to describe the temperature dependencies of all heat capacities, $x \in \{N_2, O_2, H_2, H_2O\}$, as well as the reaction enthalpy $\Delta H_m(\vartheta_{FC})$. According to Fig. 1, the estimation of the parameters of the thermal system model by a verified optimization routine leads to a system parameterization that is asymptotically stable beyond the range of the temperature values used for the parameter identification. In contrast, the simulated system output shows an unstable time response if the parameters are identified on the basis of a purely non-verified local optimization procedure.

Finally, an outlook has been given on how to employ the uncertain dynamic system model in real time for robust online control. This control framework will be designed by using interval arithmetic techniques guaranteeing both accuracy and stability in a rigorous way. For that purpose, the control synthesis is based, for example, on sliding mode techniques to guarantee stability of the system dynamics in spite of parameter uncertainties [5].

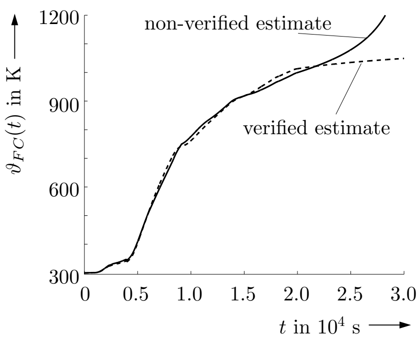

Figure 1 Comparison of simulated temperatures for verified and non-verified parameter identification

References

1 R. Bove and S. Ubertini (Eds.). *Modeling Solid Oxide Fuel Cells.* Springer-Verlag, Berlin, 2008.

2 A. Gubner. Non-isothermal and dynamic SOFC voltage-current behavior. In *Solid Oxide Fuel Cells IX (SOFC-IX): Vol. 1, Cells, Stacks and Systems*, S. C. Singhal and J. Mizusaki, Eds. The Electrochemical Society, pages 814–826, 2005.

3 A. Rauh, J. Minisini, E. P. Hofer, and H. Aschemann. Robust and optimal control of uncertain dynamical systems with state-dependent switchings using interval arithmetic. *Reliable Computing*, 15(4):333–344, 2011.

4 A. Rauh, T. Dötschel, H. Aschemann. Experimental parameter identification for a control-oriented model of the thermal behavior of high-temperature fuel cells. In *Proc. of IEEE Intl. Conference on Methods and Models in Automation and Robotics MMAR 2011*, Miedzyzdroje, Poland, 2011.

5 V. Utkin. *Sliding Modes in Control and Optimization.* Springer–Verlag, Berlin, Heidelberg, 1992.

3.22 Verified Add-ons for the DSI toolbox

Gabor Rebner (Universität Duisburg-Essen, DE)

The Dempster-Shafer with Intervals (DSI) toolbox [1], which is based on the imprecise probability toolbox (IPP) [5], is an extension for MATLAB that provides algorithms for verified computing with basic probability assignments. Throughout this talk, we use the term verification in its narrow sense of referring to a mathematical proof for correctness of a result obtained by a computer calculation. One problem of computer calculations is the appearance of rounding errors, which are unavoidable because of the finite nature of floating-point arithmetic. Even more important is to allow for uncertainty in a simulation model. Verified methods offer a solution in these cases, motivating us to use them in combination with probabilistic methods. In this talk, we present a new verified implementation of Markov set chains (MSC) [4], a C-XSC [3] interface to MATLAB and an improvement in the evaluation of monotonic and non-monotonic system functions in DSI. Furthermore we demonstrate the ability of DSI to handle uncertainty under Dempster-Shafer theory in a software presentation.

This talk is structured as follows. First, we review the DSI toolbox and its ability to handle uncertain data in a verified way. Next, we discuss our verified implementation of MSC. MSC extend classic Markov chains by defining uncertain transition matrices and initial vectors. For example, a nuclear power plant cannot be modeled by Markov chains because of the deterministic behavior of the applied transition matrices. By utilizing MSC, we are able to describe the system's operation time and environmental influences in terms of a nondeterministic transition matrix. Besides, the use of verified algorithms allows us to cope with the limitations of floating point arithmetic. Such algorithms provide an enclosure which is guaranteed to contain the exact result.

We illustrate the functionality of our new implementation using a close-to-life example. In the next part of our contribution, we present an implementation of a C-XSC to MATLAB interface. The goal of this add-on is to make algorithms written in C-XSC accessible in DSI. To avoid conversion errors, we use the MATLAB build-in interface to access MATLAB memory directly. We discuss this implementation by considering the example of the error function. Finally, we demonstrate the ability of DSI to sample monotonic and non-monotonic functions in a verified way. In order to minimize the overestimation and the computation time, we implemented a monotonicity test using automatic differentiation provided by INTLAB [6]. To compute a tight enclosure of the solution space of a function which is monotonic and contains exclusively basic arithmetic operations $\{+, -, *, /\}$ and their compositions, we utilize floating point arithmetic with directed rounding. Otherwise, we have to use interval arithmetic to get a verified enclosure of the solution. We close our talk by giving illustrative examples of sampling $f(x) = x \cdot x - x$ with x uniform distributed in two to three.

In the course of the software presentation, we demonstrate the ability of DSI to handle basic probability assignments (BPA) [2] in a verified way.

This demonstration is split into two parts. First we show the option to define cumulative distribution functions with uncertain bounds. As an example we use the triangle distribution with the uncertain mean, the lower, and the upper bound. The second part of our demonstration deals with (non-)monotonic function propagation. We take the function $f(x) = \sin(x^2) + x^3$ with x triangularly distributed and having uncertainties in the bounds and the mean. Furthermore, we illustrate the ability of DSI to detect erroneous user inputs. Finally, we demonstrate the C-XSC to MATLAB interface.

References

1 E. Auer, W. Luther, G. Rebner, and P. Limbourg. A Verified MATLAB Toolbox for the Dempster-Shafer Theory. In M. Arnaud, editor, *Proceedings of the Workshop on the Theory of Belief Functions*, Brest, April 2010. http://bfas.iutlan.univ-rennes1.fr/belief2010/html/papers/p83.pdf http://udue.de/DSI.

2 S. Ferson, V. Kreinovich, L. Ginzburg, D.S. Myers, and K. Sentz. Constructing probability boxes and Dempster-Shafer structures. Sand Report SAND2002-4015, Sandia National Laboratories, 2003.

3 R. Hammer, M. Hocks, U. Kulisch, and D. Ratz. *C++ Toolbox for Verified Computing.* Springer-Verlag Berlin Heidelberg New York, 1995.

4 D. Hartfiel. *Markov Set-Chains.* Lecture notes in mathematics; 1695. Springer, 1998. ISBN 3-540-64775-9.

5 P. Limbourg. Imprecise Probability Propagation Toolbox (IPP Toolbox). http://www.uni-due.de/il/ipptoolbox.php.

6 S. Rump. INTLAB – INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. http://www.ti3.tu-harburg.de/rump/.

3.23 Refining Abstract Interpretation-based Approximations with Constraint Solvers

Michel Rueher (Université Nice Sophia Antipolis, FR)

Joint work of Ponsini, Olivier; Michel, Claude; Rueher, Michel
Main reference O. Ponsini, C. Michel, M, Rueher, "Refining Abstract Interpretation-based Approximations with Constraint Solvers,".
URL http://hal.archives-ouvertes.fr/hal-00623274/fr/

Programs with floating-point computations are tricky to develop because floating-point arithmetic differs from real arithmetic and has many counterintuitive properties. A classical approach to verify such programs consists in estimating the precision of floating-point computations with respect to the same sequence of operations in an idealized semantics of real numbers.

Tools like Fluctuat –based on abstract interpretation– have been designed to address this problem. However, such tools compute an over-approximation of the domains of the variables, both in the semantics of the floating-point numbers and in the semantics of the real numbers. This over-approximation can be very coarse on some programs. We show that constraint solvers over floating-point numbers and real numbers can significantly refine the approximations computed by Fluctuat.

Keywords: Program verification; Floating-point computation; C programs; Abstract interpretation-based approximation; Interval-based constraint solvers over real and floating-point numbers

3.24 Constraint Programming over Continuous Domains

Michel Rueher (Université Nice Sophia Antipolis, FR)

In this tutorial, we first recall the basics of the constraint programming framework, a *branch & prune* schema which is best viewed as an iteration of two steps:

1. *Pruning the search space*
2. *Making a choice to generate two (or more) sub-problems*

The pruning step is based on partial consistencies. It *reduces an interval* when it can prove that the upper bound or the lower bound does not satisfy some constraint. We outline the intuitions behind the most common partial consistencies and we analyze the relationship between these consistencies and interval arithmetic techniques. We detail some critical implementation issues of Hull-consistency, Box-consistency and Quad–consistency.

The branching step *splits the interval* associated to some variable in two intervals (often with the same width). We give a short overview of the most effective search heuristics used in this process.

To conclude, we illustrate the powerful *refutation capabilities* of local consistencies on two applications: *boosting OBR* [2] in global optimisation; *refining approximations* (see http://hal.archives-ouvertes.fr/hal-00623274/fr/) in program verification.

References

1 F. Benhamou and L. Granvilliers. *Handbook of Constraint Programming*, chapter Continuous and Interval Constraints. Elsevier 2004.
2 A. Goldsztejn, Y. Lebbah, C. Michel, and M. Rueher. Capabilities of constraint programming in safe global optimization. *Reliable computing* 15(1):47–59, 2011.
3 R. B. Kearfott. Interval analysis: Unconstrained and constrained optimization. *Encyclopedia of Optimization*, pages 1727–1730, 2009.
4 P. Van-Hentenryck, D. McAllester, and D. Kapur. Solving polynomial systems using branch and prune approach. *SIAM Journal on Numerical Analysis*, 34(2):797–827, 1997.

3.25 Managing uncertainty and discontinuous condition numbers in finite-precision geometric computation

Peihui Shao (Université de Montréal, CA)

Joint work of Shao, Peihui; Stewart, Neil

Roundoff and representation errors in geometric computation may be modelled as epistemic or aleatory processes. In either case, if it is required that the computed result have correct topological form, then problems such as computing regularized Boolean operations are fundamentally ill-conditioned. In this paper it is shown that if we wish to drive the process through a discontinuity in the condition number, and if we wish to prove rigorous theorems, then use of an exception-handling mechanism analogous to those used in programming languages cannot be avoided.

The above observations show that traditional approaches to proving robustness, when computation is done using ordinary IEEE floating-point arithmetic, are inappropriate. We discuss the nature, in the context described above, of the theorems that should be proved, and we give a very simple result that illustrates our approach.

3.26 Reliable Kinetic Monte Carlo Simulation based on Random Set Sampling

Yan Wang (Georgia Institute of Technology, US)

The kinetic Monte Carlo (KMC) method has been widely used in simulating rare events such as chemical reactions or phase transitions. The lack of complete knowledge of transitions and the associated rates is one major challenge for accurate KMC predictions. In this work, a reliable KMC (R-KMC) mechanism is developed to improve the robustness of KMC results, where propensities are interval estimates instead of precise numbers and sampling is based on random sets instead of random numbers. A multi-event algorithm is developed for event selection, and the system time is advanced based on best- and worst-case scenarios. The weak convergence of the multi-event algorithm towards traditional KMC is demonstrated with an interval master equation.

References

1 Y. Wang. Reliable kinetic Monte Carlo simulation based on random set sampling. In *Proc. 2011 ASME International Design Engineering Technical Conferences & The Computer and Information in Engineering Conference*, Aug. 29-31, 2011, Washington, DC, Paper No.DETC2011-48575.

3.27 The General Interval Power Function

Jürgen Wolff von Gudenberg (Universität Würzburg, DE)

Joint work of Heimlich, Oliver; Nehmeier, Marco; Wolff von Gudenberg, Jürgen

The purpose of this article is to find a general power function for use in interval arithmetic, particularly with regard to the upcoming interval arithmetic standard which is being developed by the IEEE working group P1788.

Apparently, in the history of mathematics exponentiation has sometimes been used more or less as an abbreviatory notation and several definitions from different backgrounds, i.e., algebra and analysis, have been combined in people's minds.

For the set of real numbers these definitions agree on many points, but not on all. Points at issue are treated differently depending on the context, which poses problems searching for a common standard.

One contentious issue with the definition of general exponentiation is the assignment of real result to powers with negative base and rational exponent. We discuss three variants each of which has its merits and drawbacks.

The interval extensions are presented for each of the three variants, and efficient algorithms are implemented in INTLAB.

Finally, we recommend two of the three variants for interval libraries.

3.28 C-XSC – Overview and new developments

Michael Zimmer (Universität Wuppertal, DE)

C-XSC is a powerful C++ library for verified numerical computing. It provides many useful data types, among them real and complex intervals, (interval) vectors and matrices and a long accumulator for the computation of dot product expressions in high accuracy. Also included is a toolbox with implementations of many useful verified algorithms. In the past few years, their have been some major extensions to the C-XSC library, especially with regard to the use of C-XSC in High Performance Computing. In our presentation, we gave a short overview of its features and then focused especially of the new developments in recent versions.

The first addition is the ability to compute dot products and dot product expressions in K-fold double precision. The desired precision can be changed at runtime, so that the user can choose between higher accuracy or higher speed for each computation. The second major addition is the optional support of the BLAS library for all vector-vector, matrix-vector and matrix-matrix products in double precision. The use of these algorithms can drastically increase the performance of such operations.

Another major addition are new data types for sparse matrices and vectors. These allow to work with such vectors and matrices in a very efficient way, both in terms of memory consumption and computing speeds. The new data types provide an easy to use through the use of operator overloading. The sparse data types are based on widely used data structures which makes it easy to write interfaces to other sparse matrix software.

Also among the new features are new data types for multiple and arbitrary precision arithmetic, which were covered by the talk of Walter Krämer during the course of this Dagstuhl seminar.

The last presented new feature was the drastically improved thread safety of C-XSC, which makes it easy for the user to parallelize C-XSC programs for multicore machines, for example by using OpenMP. Finally, the talk also give a small outline of future developments of the C-XSC library.

During this Dagstuhl seminar, Gabor Rebner presented a talk about add-ons for the DSI toolbox using a self build interface between Matlab/Intlab and C-XSC using Matlabs Mex compiler to make use of the error function implemented in C-XSC. In a future work, a full interface between Matlab/Intlab and C-XSC might be possible.

In an additional software presentation at this Dagstuhl seminar, the installation of C-XSC on an example system (64 bit Macbook Pro running Mac OS X Lion) was demonstrated. Many of the options for compilation and optimization for the compilation of the core library and of C-XSC programs were covered in detail during this presentation. It should be stressed that a default installation of C-XSC only requires to start an installation script, accept the license by typing `yes` and then hitting enter 9 times to accept all the default settings.

Special thanks to Frithjof Blomquist as well as to some of our recent Master/Diploma students (Falko Sieg, Sascha Habicht, Frank Roitzsch, Christian Doescher, Daniel Kreuer, Michael Hirdes and Daniel Dakowski) for contributing to the development of C-XSC.

References

1 Weblink to C-XSC:
`http://www2.math.uni-wuppertal.de/wrswt/xsc/cxsc_new.html`

2 G. Alefeld and J. Herzberger. *Introduction to Interval Computations.* Academic Press, 1983.

3 L. S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R. C. Whaley. An updated set of basic linear algebra subprograms (BLAS). *ACM Trans. Math. Soft.*, 28(2):135–151, 2002.

4 A. Cuyt, W. Krämer, W. Luther, and P. Markstein. Numerical validation in current hardware architectures. In*Lecture Notes in Computer Science LNCS 5492*, Springer, 2009.

5 R. Hammer, M. Hocks, U. Kulisch, and D. Ratz. *Numerical Toolbox for Verified Computing I: Basic Numerical Problems.* Springer, 1993.

6 W. Hofschuster and W. Krämer. C-XSC 2.0: A C++ library for extended scientific computing. In *Numerical Software with Result Verification, Lecture Notes in Computer Science*, Volume 2991/2004, Springer, pages 15–35, 2004.

7 W. Krämer, M. Zimmer, and W. Hofschuster. Using C-XSC for high performance verified computing Accepted for publication (PARA2010).

8 U. Kulisch. *Computer Arithmetic and Validity: Theory, Implementation, and Applications.* De Gruyter Studies in Mathematics, 2008.

9 R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis.* SIAM, Philadelphia, 2009.

10 S. M. Rump. INTLAB – Interval Laboratory. Developments in Reliable Computing, pages 77–104, 1999.
11 S. M. Rump. Verification methods: Rigorous results using floating-point arithmetic *Acta Numerica*, 2010.
12 M. Zimmer, W. Krämer, G. Bohlender, and W. Hofschuster. Extension of the C-XSC library with scalar products with selectable accuracy. *Serdica Journal of Computing*, 4(3):349–370, 2010.
13 M. Zimmer, W. Krämer, and W. Hofschuster. Sparse matrices and vectors in C-XSC. *Reliable Computing*, 14:128–160, 2010.

4 Schedule

Monday, September 12, 2011

9:00–9:30: Welcome
9:30–10:00/10:15–10:45: *Ralph Baker Kearfott*, Interval Computations – Introduction and Significant Applications
10:45–11:15: *Olga M. Kosheleva*, Degree-Based (Interval and Fuzzy) Techniques in Math and Science Education
11:15–12:15: *Michel Rueher*, Constraint Programming over Continuous Domains
14:00–14:30: *Wolfram Luther*, Verification &Validation Requirements in Biomechanical Modeling and Simulation
14:30–15:00: *Andreas Rauh*, Verified Parameter Estimation for the Thermal Behavior of High-Temperature Fuel Cells
15:30-15:30: *Ekaterina Auer*, Application of Verified Methods to Solving Non-smooth Initial Value Problems in the Context of Fuel Cell Systems
16:00–17:00: *Vladik Kreinovich*, Tutorial: Foundations in Fuzzy Modeling
17:00–17:30: *Michael Beer*, Fuzzy Probabilities and Applications in Engineering
17:30–18:00: *Weldon A. Lodwick*, Constrained Intervals and Interval Spaces

Tuesday, September 13, 2011

9:00–9:30: *Jürgen Garloff*, Verified Solution of Finite Element Models for Truss Structures with Uncertain Node Locations
9:30–10:00: *Milan Hladik*, Interval Linear Programming: Foundations, Tools and Challenges
10:15–10:45: *Ralph Baker Kearfott*, A Review of Techniques for Handling Model Uncertainty in Interval-Based Global Optimization Procedures
10:45–11:15: *Evgenija D. Popova*, Characterizing AE Solution Sets to Parametric Linear Systems
11:15–11:45: *Olivier Mullier*, Under-approximation of the Range of Vector-Valued Functions having Different Dimensions for Domain and Codomain
11:45–12:15: *Nathalie Revol*, Solving and Verifying Efficiently the Solution of a Linear System
14:00–14:30: *Martin Fuchs*, Robust Optimization for Aerospace Applications
14:30–15:00: *Neli Dimitrova*, Asymptotic Stabilization of a Bioprocess Model Involving Uncertainties
15:00–15:30: *Jean-Luc Lamotte*, Analysis of Electronic Circuit with Point of View of Uncertainty
16:00–16:30: *John D. Pryce*, Decorations for Dummies (1)

16:30–17:00: *Heinrich Rommelfanger*, Describing Vague Data by Fuzzy Intervals - Intelligent Ways for Solving Real-World Decision Problems and for Saving Information Costs
17:00–17:30: *Yan Wang*, Reliable Kinetic Monte Carlo Simulation based on Random Set Sampling

Wednesday, September 14, 2011

9:00–9:30: *Walter Krämer*, A Comparison of Different Kinds of Multiple Precision and Arbitrary Precision Interval Arithmetics
9:30–10:00: *Vincent Lefèvre*, Generating a Minimal Interval Arithmetic Based on GNU MPFR
10:15–11:15: *Philipp Limbourg and Gabor Rebner*, IPP Toolbox and Verified Add-ons for the DSI toolbox
11:15–11:45: *Jürgen Wolff von Gudenberg*, The General Interval Power Function
11:45–12:15: *Michael Zimmer*, C-XSC - Overview and New Developments
13:45–21:00: Excursion to Trier (guided tour, sight seeing, winery, dinner and wine tasting)

Thursday, September 15, 2011

9:00–9:30: *Cliff Joslyn and Emilie Hogan*, Intervals, Orders, and Rank
9:30–10:00: *Peihui Shao and Neil Stewart*, Managing Epistemic Uncertainty when Condition Numbers are Discontinuous
10:15–10:45: *Stefan Kiel*, Integration of Interval Contractors in Hierarchical Space Decomposition Structures
10:45–11:15: *Götz Alefeld*, Error Bounds for Nonlinear Complementary Problems with Band Structure
11:15–11:45: *Michel Rueher*, Using CSP Refutation Capabilities to Refine AI-based Approximations
14:00–14:30: *Vladik Kreinovich*, Towards Optimal Representation and Processing of Uncertainty for Decision Making, on the Example of Economics-Related Heavy-Tailed Distributions
14:30–15:00: *John D. Pryce*, Decorations for Dummies (2)
15:00–18:00: P 1788 Interval Standard Group *(N. Revol et al.)*
19:30–20:30: Software Presentation

Friday, September 16, 2011

9:00–9:30: *S. Rump*, Error Estimation of Floating-Point Summation and Dot Product
9:30–10:00: *Arnold Neumaier and Qaisra Fazal*, Enclosing Dynamical Systems with Large Initial Uncertainties
10:30–11:15: Strategy discussion
11:15–12:00: Closing session

Acknowledgements

We would like to thank the staff of Schloss Dagstuhl for their help in organizing this seminar and for the excellent facilities. Thanks go to Martin Fuchs for his help in collecting abstracts of the talks and other related materials for these proceedings.

Participants

- Götz Alefeld
KIT – Karlsruhe Institute of Technology, DE
- Ekaterina Auer
Universität Duisburg-Essen, DE
- Michael Beer
University of Liverpool, GB
- Neli Dimitrova
Bulgarian Acad. of Sciences, BG
- Martin Fuchs
Cerfacs – Toulouse, FR
- Jürgen Garloff
HTWG Konstanz, DE
- Milan Hladík
Charles University – Prague, CZ
- Cliff Joslyn
Pacific Northwest National Lab. – Seattle, US
- Ralph Baker Kearfott
Univ. of Louisiana – Lafayette, US
- Stefan Kiel
Universität Duisburg-Essen, DE
- Olga M. Kosheleva
University of Texas – El Paso, US
- Walter Krämer
Universität Wuppertal, DE
- Vladik Kreinovich
University of Texas – El Paso, US
- Jean-Luc Lamotte
UPMC – Paris, FR
- Vincent Lefevre
ENS – Lyon, FR
- Philipp Limbourg
Universität Duisburg-Essen, DE
- Weldon A. Lodwick
University of Colorado, US
- Wolfram Luther
Universität Duisburg-Essen, DE
- Olivier Mullier
Supélec – Gif-sur-Yvette, FR
- Arnold Neumaier
Universität Wien, AT
- Evgenija D. Popova
Bulgarian Acad. of Sciences, BG
- John D. Pryce
Cardiff University, GB
- Andreas Rauh
Universität Rostock, DE
- Gabor Rebner
Universität Duisburg-Essen, DE
- Nathalie Revol
ENS – Lyon, FR
- Heinrich Rommelfanger
Univ. Frankfurt am Main, DE
- Michel Rueher
Université de Nice, FR
- Siegfried M. Rump
TU Hamburg-Harburg, DE
- Peihui Shao
Université de Montréal, CA
- Neil Stewart
Université de Montréal, CA
- Yan Wang
Georgia Inst. of Technology, US
- Jürgen Wolff von Gudenberg
Universität Würzburg, DE
- Michael Zimmer
Universität Wuppertal, DE