THEMATIC SECTION:
RESULT-VERIFYING COMPUTING

Introduction

Andreas Frommer, Bruno Lang
Applied Computer Science and Scientific Computing,
University of Wuppertal,
D-42097 Wuppertal, Germany;
{*frommer,lang*}*@math.uni-wuppertal.de*

Suppose that after a lengthy computation the computer gives you some real number, and you want to be *sure* that this number is the correct result, or at least close to it. Or you need to know whether the problem (a nonlinear system, a differential equation, etc.) has a solution at all and where the solution(s) may be, or how much the results can vary if there are tolerances in the input.
In this thematic section we focus on result-verifying algorithms based on interval arithmetic, a computational tool that might help you to answer such questions in an *automated* way. Some central ideas underlying interval arithmetic can be explained on a few pages; we will do so in the following. The obvious approaches will, however, often yield results that cannot be used in practice. Then more sophisticated techniques are required. We will give a few hints on how such techniques work. The interested reader is referred to books, e.g. [3, 5] or the "classics" [1, 4], for a broader introduction to interval arithmetic.

The Case Studies

The case studies in this thematic section highlight a few applications of interval-based computations, coming from such diverse areas as the design of structures ranging from particle accelerators to space telescopes to reinforced concrete beams to computer chips; the analysis and robust design of hybrid systems and chemical processes; the evaluation of special functions; and combinatorial optimization.

Basic Concepts of Interval Arithmetic

In the following, (non-empty, real, compact) intervals will be denoted as

$$[a] = [\underline{a}, \overline{a}] = \{\tilde{a} \in \mathbb{R} : \underline{a} \le \tilde{a} \le \overline{a}\},$$

where $\underline{a} \le \overline{a}$, and $\mathbb{IR}$ is the set of all such intervals. The addition of two intervals is defined as

$$[a] + [b] = \{\tilde{a} + \tilde{b} : \tilde{a} \in [a], \tilde{b} \in [b]\}$$

and analogously for subtraction, multiplication, and division (provided that $0 \notin [b]$). Similarly, the standard functions are extended to intervals via

$$\exp([a]) = \{\exp(\tilde{a}) : \tilde{a} \in [a]\},$$

etc. Thus the result of an interval operation contains all possible outcomes if the respective operation is applied to arbitrary numbers from the argument intervals. For continuity reasons the results of the interval operations are again intervals. The above definitions allow to embed $\mathbb{R}$ into $\mathbb{IR}$ by identifying the real numbers with "point" (or "degenerate") intervals $a \equiv [a, a] = \{a\}$.
One important feature of the interval operations is that their results can be determined with just a few operations involving the interval bounds. For example,

$$\begin{aligned} [a] + [b] &= [\underline{a} + \underline{b}, \overline{a} + \overline{b}], \\ [a] \cdot [b] &= [\min S, \max S], \\ &\qquad \text{where } S = \{\underline{a}\underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b}\}, \\ \exp([a]) &= [\exp(\underline{a}), \exp(\overline{a})]. \end{aligned}$$

For non-monotonic functions like the sine function the situation is slightly more complex, the result depending on which multiples of $\pi/2$ are contained in the argument interval.
In order to make interval arithmetic amenable to computers, one obstacle must be overcome: The result of an interval operation needs not be representable in a given floating-point format, even if the operands are. Here one makes use of the fact that modern processors (e.g. those conforming to the IEEE Standard 754) give control over the rounding of non-representable numbers. In the case of interval multiplication, the lower bound of the product is obtained by computing the four elements of S with downward rounding (toward $-\infty$) and taking their minimum. To obtain the upper bound, one *recomputes* the four numbers with upward rounding (toward $+\infty$) and takes their maximum. This *outward rounding* guarantees that the exact result of the interval operation is contained in the computed result.
Interval arithmetic provides a straight-forward way for computing enclosures for the range of a function φ over an interval vector (or "box") $[\boldsymbol{x}] = ([\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n])^T \in \mathbb{IR}^n$: Replace each variable x_i with the respective interval $[\underline{x}_i, \overline{x}_i]$ and perform each operation that occurs during the evaluation of φ as an interval operation. For example, let

$$\varphi(\boldsymbol{x}) = x_1 \cdot x_2 - x_1$$

and $[\boldsymbol{x}] = ([-1, 1], [0, 1])^T$. Then

$$\varphi([\boldsymbol{x}]) = [-1, 1] \cdot [0, 1] - [-1, 1] = [-2, 2],$$

which indeed contains the range $[-1, 1]$ of φ over $[\boldsymbol{x}]$. This procedure is called the "natural interval evaluation" of φ.

Given the ability to compute enclosures for the range, the following simple "branch–and–bound" procedure can be used to find the solutions of a nonlinear system $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$ with $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$. Starting with a box $[\boldsymbol{x}]^0 \subset \mathbb{IR}^n$, we compute enclosures $[f_i]$ for the ranges of the functions f_i over $[\boldsymbol{x}]^0$. If $0 \notin [f_i]$ for some i then $[\boldsymbol{x}]^0$ cannot contain a solution of the system, and can be discarded ("bound"). Otherwise $[\boldsymbol{x}]^0$ is subdivided into smaller boxes and the same procedure is applied recursively to these ("branch"), until the remaining boxes are "small enough". At the end we obtain a list of small boxes with the *guarantee* that these boxes cover all solutions of the system within the initial box $[\boldsymbol{x}]^0$. It is, however, not guaranteed that each result box does indeed contain a solution.

For many of the boxes even this proof can be *automated*, again with the aid of interval arithmetic. According to Brouwer's fixed point theorem any continuous function $\boldsymbol{g}$ that maps a non-empty, convex, and compact set C into itself must have a fixed point $\boldsymbol{x}^* \in C$. Now define $\boldsymbol{g}$ in such a way that its fixed points are zeros of $\boldsymbol{f}$. Letting $[\boldsymbol{g}]$ denote an enclosure for the range of $\boldsymbol{g}$ over a box $[\boldsymbol{x}]$, the condition $[\boldsymbol{g}] \subseteq [\boldsymbol{x}]$ can be checked automatically, and it ensures the existence of a zero of $\boldsymbol{f}$ in $[\boldsymbol{x}]$.

More Sophisticated Techniques

A naive use of the methods described in the previous section may lead to results of limited applicability.

Natural interval evaluation typically *over-estimates* the range of the function, and the result interval may be substantially larger than the true range. This is mainly due to the *dependency problem*, i.e., multiple occurrences of the same variable in an expression are treated as independent quantities with the same range of variation. To reduce the over-estimation one resorts to different representations of the function. Let φ be continuously differentiable and $\widetilde{\boldsymbol{x}}$ be some point in $[\boldsymbol{x}]$. Then the mean value theorem states that for each $\boldsymbol{x} \in [\boldsymbol{x}]$ there exists a $\boldsymbol{\xi}$ between $\widetilde{\boldsymbol{x}}$ and $\boldsymbol{x}$ such that

$$\varphi(\boldsymbol{x}) = \varphi(\widetilde{\boldsymbol{x}}) + \nabla\varphi(\boldsymbol{\xi}) \cdot (\boldsymbol{x} - \widetilde{\boldsymbol{x}}).$$

Therefore the "centered form"

$$[\varphi] = \varphi(\widetilde{\boldsymbol{x}}) + [\nabla\varphi] \cdot ([\boldsymbol{x}] - \widetilde{\boldsymbol{x}})$$

also yields an enclosure for the range of φ over $[\boldsymbol{x}]$, where $[\nabla\varphi]$ denotes some enclosure for the range of the *gradient* of φ over $[\boldsymbol{x}]$. Such an enclosure again can be obtained in an automated way, e.g., by combining Automatic Differentiation [2] and interval arithmetic. Centered forms can provide substantially sharper enclosures for the range, in particular for small boxes. Other methods for reducing the over-estimation include higher-order Taylor expansion (e.g., "Taylor arithmetic") and "affine" arithmetic, to name only two.

The basic branch–and–bound algorithm described above is by far too inefficient to be useful in practice. It must be complemented with acceleration techniques that allow cutting off parts of the current box (without losing solutions) before subdividing it. To give one example, the *Krawczyk operator* is defined by

$$[\boldsymbol{k}] = \widetilde{\boldsymbol{x}} - \mathbf{R} \cdot \boldsymbol{f}(\widetilde{\boldsymbol{x}}) + (\mathbf{I} - \mathbf{R} \cdot [\mathbf{J}]) \cdot ([\boldsymbol{x}] - \widetilde{\boldsymbol{x}}).$$

Here $\widetilde{\boldsymbol{x}}$ is some point in $[\boldsymbol{x}]$, $[\mathbf{J}]$ is an enclosure for the Jacobian $\mathbf{F}'$ over $[\boldsymbol{x}]$, and $\mathbf{R}$ is an arbitrary matrix. Then each zero of $\boldsymbol{f}$ that is contained in $[\boldsymbol{x}]$ is also contained in $[\boldsymbol{x}] \cap [\boldsymbol{k}]$. In particular, $\boldsymbol{f}$ cannot have a zero in $[\boldsymbol{x}]$ if $[\boldsymbol{x}] \cap [\boldsymbol{k}] = \emptyset$. Moreover, if $[\boldsymbol{k}]$ is contained in $[\boldsymbol{x}]$ and $\mathbf{R}$ is non-singular, then the existence of a zero of $\boldsymbol{f}$ in $[\boldsymbol{x}]$ is guaranteed. There are other, more powerful, operators achieving similar effects.

In addition to the "standard" interval arithmetic described so far, extensions have been defined to cover infinite intervals, partially defined functions, etc.

Applying Interval Methods

Interval versions of known algorithms have been developed, or novel methods have been devised, for linear and nonlinear systems of equations, eigenvalue problems, unconstrained and constrained global optimization, ordinary and partial differential equations, numerical integration, geometric problems, and many more. The added value of using these methods is the guaranteed correctness of the results, as compared to standard floating-point algorithms. To facilitate the use of these methods, several programming languages such as C, C++, Fortran, and Pascal, have been extended to provide interval support.

The interested reader is referred to the repository `http://www.cs.utep.edu/interval-comp/` maintained by Vladik Kreinovich. There one can find pointers to current publications and available software.

References

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

[2] A. Griewank. *Evaluating Derivatives—Principles and Techniques of Automatic Differentiation*. SIAM, Philadelphia, PA, 2000.

[3] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag, London, UK, 2001.

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

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

Long-Term Stability of Large Particle Accelerators

Kyoko Makino, Martin Berz,
Youn-Kyung Kim, and Pavel Snopok
Department of Physics and Astronomy
Michigan State University
East Lansing, MI 48820;
{*makino,berz,kimyounk*}*@msu.edu*
snopok@pa.msu.edu

Large Particle Accelerators

High energy particle accelerators constitute the largest scientific instruments currently in use. Their purpose is to accelerate subatomic particles to energies more than a thousand times their rest mass and bring them to collision. In many cases, a layout of two counter-rotating rings is being used, in which particles travel in high-quality vacuum and are held in orbit by record strength superconducting magnets. Since in each turn, only a very small fraction of the particles actually collide, this design allows the re-use of those particles that do not produce collisions. The approach is used in the Tevatron at Fermilab with nearly 7km circumference, which is the accelerator currently achieving the highest energies, and also in the new LHC accelerator under construction at CERN with a circumference of about 27km.

Computational Challenges

The computational efforts necessary for understanding the dynamics of these particles are daunting: they are kept in orbit for minutes to hours, which at a speed very near that of light requires stable motion for about 10^8 or 10^9 revolutions. In each of these, particles are affected by several thousand control magnets. Thus, attempts at direct numerical integration of orbits have to resort to various approximations and the selection of small subsets of particle coordinates.

It has proven useful to simulate the dynamics with the use of so-called transfer maps which describe the relationship of final coordinates on initial coordinates for one revolution via a Taylor expansion. Traditionally, formulas for maps were computed by hand using methods of perturbation theory for each type of magnetic element being employed [1, 2, 3]. The use of formula manipulators has pushed this technique from the previous order three to order five [4], leading to explicit formulas extending over many tens of thousands of lines of computer code.

High-Order Maps and Normal Forms

The development of the differential algebraic method [5, 6] made it possible to straightforwardly extend these orders even beyond ten, at which level the accuracy of the approach reaches machine precision. Recent enhancements of the method [7] are also able to determine rigorous interval bounds of the remainder errors, and advances in the field of verified integration of ODEs have allowed the far-reaching suppression of the so-called wrapping effect problem [8].

However, further analysis requires an additional tool, the method of normal forms, developed to arbitrary order in [9, 10, 6]. In the resulting normal form coordinates, the motion follows nearly perfect circles around a fixed point. The measure of "non-circularity" is on the one hand an indication of non-integrability [11]. On the other hand, it allows the computation of stability times; because if circularity is preserved up to an error Δ, then it apparently requires at least

$$N = \frac{r_2 - r_1}{\Delta}$$

revolutions to migrate from the region inside radius r_1 to the region outside r_2.

Figure 1 shows an example of a normal form defect function. While having very small function values, already in this two-dimensional projection of the six dimensional function, many local minima are visible.

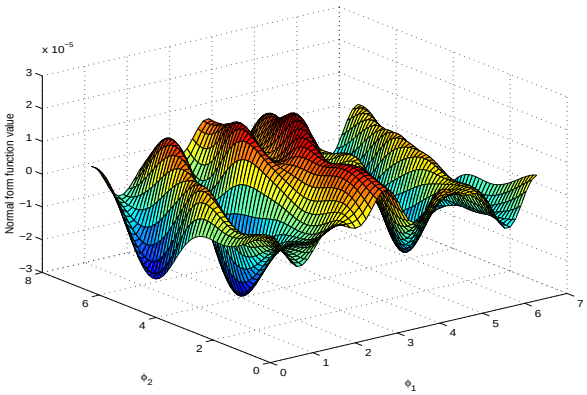


Figure 1: Projection of the normal form defect function. Dependence on two angle variables for fixed radii.

Taylor Models and Verified Range Bounding

Stability can thus be decided by determining a rigorous upper bound of the so-called normal form defect Δ. Verified computational methods provide various tools for rigorous global optimization (see for example [12]). For the problem of the bounding of the normal form defect, however, the methods are not directly applicable because of far reaching overestimation due to the

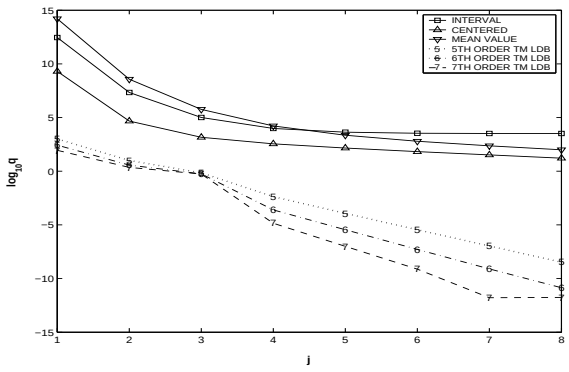


Figure 2: Relative overestimation of various verified bounding methods for a sample normal form defect function in the domain $0.1 \cdot (1 + [-2^{-j}, 2^{-j}])^6$.

Region	Bound	Stable Turns
$[0.2, 0.4] \cdot 10^{-4}$	$0.859 \cdot 10^{-13}$	$2.328\,3 \cdot 10^{8}$
$[0.4, 0.6] \cdot 10^{-4}$	$0.587 \cdot 10^{-12}$	$3.407\,2 \cdot 10^{7}$
$[0.6, 0.9] \cdot 10^{-4}$	$0.616 \cdot 10^{-11}$	$4.870\,1 \cdot 10^{6}$

Table 1: Global bounds obtained for three regions in normal form space for the Tevatron. Also computed are the guaranteed minimum stable turns for each of the regions.

dependency problem. The recently developed Taylor model methods (see [13] and references therein) allow to suppress much of this problem [14] by representing a function by its Taylor polynomial and a rigorous enclosure for the error of this approximation. By doing so, the bulk of the functional dependency is described by the Taylor polynomial, which is not subject to the effects of the dependency problem.

To illustrate this, we compare the performance of Taylor models (TM) of orders 5, 6, and 7 with the commonly used centered form (CF), the mean value form (MF), and plain interval evaluation (I). In Figure 2, we show the results for the domains $D = 0.1 \cdot (1 + [-2^{-j}, 2^{-j}])^6$. The bounding of the polynomials is performed using the LDB bounder [15]. It is seen that for $j = 7$, the 7th order TM method outperforms CF by around 14 orders of magnitude.

Results

Utilizing a verified branch-and-bound optimizer based on Taylor models, we calculate the normal form defect for the Tevatron accelerator for various annular regions corresponding to the actual locations of the beam. Table 1 shows the obtained normal form defect bounds and the resulting stability times, proving the stability of the Tevatron for nearly 300 million turns.

References

[1] K. L. Brown, R. Belbeoch, and P. Bounin. First- and second- order magnetic optics matrix equations for the midplane of uniform-field wedge magnets. *Review of Scientific Instruments*, 35:481, 1964.

[2] H. Wollnik. *Optics of Charged Particles*. Academic Press, Orlando, Florida, 1987.

[3] A. J. Dragt. Lectures on nonlinear orbit dynamics. In *1981 Fermilab Summer School*, volume 87. AIP Conference Proceedings, 1982.

[4] M. Berz and H. Wollnik. The program HAMILTON for the analytic solution of the equations of motion in particle optical systems through fifth order. *Nuclear Instruments and Methods*, A258:364–373, 1987.

[5] M. Berz. Differential algebraic description of beam dynamics to very high orders. *Particle Accelerators*, 24:109, 1989.

[6] M. Berz. *Modern Map Methods in Particle Beam Physics*. Academic Press, San Diego, 1999. Also available at http://bt.pa.msu.edu/pub.

[7] M. Berz and K. Makino. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4(4):361–369, 1998.

[8] K. Makino and M. Berz. Suppression of the wrapping effect by Taylor model based validated integrators. Submitted. Also MSUHEP-040910, available at http://bt.pa.msu.edu/pub.

[9] E. Forest, M. Berz, and J. Irwin. Normal form methods for complicated periodic systems: A complete solution using Differential algebra and Lie operators. *Particle Accelerators*, 24:91, 1989.

[10] M. Berz. High-order computation and normal form analysis of repetitive systems. In: M. Month (Ed), *Physics of Particle Accelerators*, volume 249, page 456. American Institute of Physics, New York, 1991.

[11] Henri Poincare. *New Methods of Celestial Mechanics*, volume 1-3. American Institute of Physics, New York, 1893/1993.

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

[13] K. Makino and M. Berz. Taylor models and other validated functional inclusion methods. *International Journal of Pure and Applied Mathematics*, 6,3:239–316, 2003. available at http://bt.pa.msu.edu/pub.

[14] K. Makino and M. Berz. Efficient control of the dependency problem based on Taylor model methods. *Reliable Computing*, 5(1):3–12, 1999.

[15] M. Berz, K. Makino, and Y.-K. Kim. Long-term stability of the tevatron by validated global optimization. *Nuclear Instruments and Methods*, in print, 2005.

Robust Design of a Deployable Space Telescope

Jean-Pierre Merlet
INRIA
06902 Sophia-Antipolis, France;
Jean-Pierre.Merlet@sophia.inria.fr

Problem Description

Current space telescopes use a two-mirrors (primary, secondary) architecture. The mechanical structure connecting the primary and secondary mirrors is passive although it is exposed to large disturbances. For example the telescope cannot be launched in its final configuration and the deployment mechanism induces significant uncertainties in the geometry of the telescope. Furthermore a space telescope is exposed to large thermal variations that modify its geometry. Computer image enhancement may partially correct the errors due to the modification of the geometry of the telescope, but this method has almost reached its limit. Another approach is to strengthen the structure of the telescope but this increases its inertia, thereby inducing a higher space fuel consumption (available only in very limited amount) to change the pointing direction.
We have been approached by a space telescope company to address this problem.

Active Telescope

To reduce the inertia and to improve the positioning accuracy we have proposed the use of an *active* deployable mechanism (Figure 1). During the launch the secondary mirror is supported by the primary mirror. Six articulated legs attached on a crown surrounding the primary mirror are connected to the secondary mirror. A spreadband mechanism in each leg is coiled during the launch and uncoiled to a fixed length in space, allowing the secondary mirror to reach approximately its nominal position. Six linear actuators located on the crown allow to move the foot of the spreadband along a fixed direction. Such a structure is known in mechanism theory under the name *parallel structure*. It can be shown that by controlling the motion $\boldsymbol{\rho}$ of the linear actuators it is possible to control the position and orientation of the secondary mirror with respect to the primary one. More precisely let $\boldsymbol{x}$ be a vector of parameters describing the position/orientation of the secondary mirror; then a relation $\boldsymbol{\rho} = \boldsymbol{f}(\boldsymbol{x})$ may be established. The possible motions of the actuators are limited and verify $\rho_{\min} \leq \boldsymbol{\rho} \leq \rho_{\max}$ where $\rho_{\min}, \rho_{\max}$ are constants.

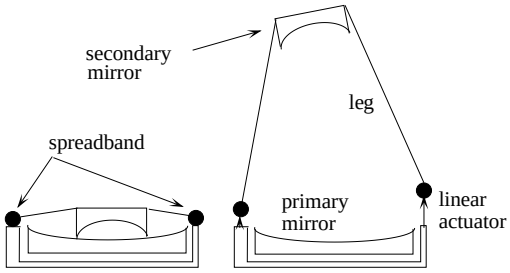


Figure 1: The deployable telescope (only 2 legs are represented) in its launch (left) and in space configuration.

These motions are measured by sensors but these measurements are afflicted with a uniform noise $\Delta\rho_i$ lying in the range $[-\Delta\rho^M, \Delta\rho^M]$, where the maximal sensor error $\Delta\rho^M$ is known. Hence if ρ_i^{m} is a measurement of ρ_i we have $\rho_i = \rho_i^{\mathrm{m}} + \Delta\rho_i$. The measurement errors imply that we cannot control exactly the position of the mirror. The positioning error $\boldsymbol{\Delta x}$ of the mirror is linearly related to the sensors' error by $\boldsymbol{\Delta\rho} = \mathbf{J}(\boldsymbol{x}) \cdot \boldsymbol{\Delta x}$, where $\mathbf{J}$ is a 6×6 matrix called the *inverse Jacobian*, that is a function of $\boldsymbol{x}$ and of the geometry of the mechanism. An analytical formulation of $\mathbf{J}$ is known but its inverse is very complex.
The interest of such a structure is that its inertia is reduced to a minimum (the spreadbands have a very low inertia, even when fully deployed), while the positioning accuracy of the secondary mirror with respect to the primary may be very high if the mechanism is properly designed, ensuring high quality images.

The Design Problem

Parallel structures are known to be very effective but also to have very sensitive performances with respect to their geometry. The requirements that have to be fulfilled are as follows:

- *workspace constraint*: being given bounds on the possible deviation of the location of the secondary mirror from its nominal position $\boldsymbol{x}^{\mathrm{n}}$, the limited motion of the actuator should allow to bring back the mirror close to $\boldsymbol{x}^{\mathrm{n}}$. The possible locations of the secondary mirror define the *workspace* $\mathcal{W}$ of the mechanism
- *accuracy constraint*: when under control, the telescope will receive a requested position $\boldsymbol{x}^{\mathrm{r}}$ for the secondary mirror, and the final position $\boldsymbol{x}^{\mathrm{f}}$ of the secondary mirror should be such that $|x_i^{\mathrm{r}} - x_i^{\mathrm{f}}| \leq \epsilon_i$ where ϵ_i is a predefined threshold

We have to determine the design parameters of the mechanism so that the workspace and accuracy constraints are satisfied. These parameters are the location of the attachment points of the legs on the primary and

secondary mirrors ($3 \times 6 \times 2 = 36$ unknowns), the length L of the legs (that are supposed to be identical) and the motion limits $\rho_{\min}, \rho_{\max}$ of the actuators (which are identical). Hence we have a total of 39 design parameters (note that due to the geometry of the robot and of the task all the design parameters can be bounded). A lower limit $\Delta\rho^M_{\text{low}}$ for the sensor error is given and we have just to verify that for any $\boldsymbol{x}$ in $\mathcal{W}$ the positioning errors Δx_i are smaller than ϵ_i.

We have however to deal with another problem: the design parameters of a mechanism are never exactly respected when the mechanism is built because of the manufacturing tolerances. Still it is critical to guarantee that the accuracy and workspace constraints will be satisfied for the real telescope. For that purpose we have designed a method for determining the design parameters that satisfy the workspace and accuracy constraint as ranges which have a width at least equal to the manufacturing tolerances.

A Design Methodology with Interval Analysis

Basically we have to find a set of design parameters $\boldsymbol{p}$ such that for all $\boldsymbol{x} \in \mathcal{W}$ we have

$$\begin{aligned} \rho_{\min} - \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}) &\leq \mathbf{0} \\ \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}) - \rho_{\max} &\leq \mathbf{0} \end{aligned} \tag{1}$$

and such that for all $\boldsymbol{x} \in \mathcal{W}$ all the solutions $\Delta\boldsymbol{x}$ of the linear systems

$$\mathbf{J}(\boldsymbol{p}, \boldsymbol{x}) \cdot \boldsymbol{\Delta x} = \boldsymbol{\Delta \rho} \tag{2}$$

have each component Δx_i lower than ϵ_i in absolute value.

Interval analysis is an appropriate method to find an inner approximation of the solutions of a set of inequalities such as (1) [1]. For an n-dimensional vector $\boldsymbol{p}$ the result will be provided as a list of n-dimensional *boxes*, each edge of a box representing the values of one component of $\boldsymbol{p}$. For any point in a box the inequalities will be satisfied for all $\boldsymbol{x} \in \mathcal{W}$. The algorithm may be designed in such a way that none of the boxes in the list will have a width that is lower than the manufacturing tolerances.

Dealing with the accuracy constraint is more demanding. Basically we have a set of linear systems $\mathbf{J}(\boldsymbol{p}, \boldsymbol{x}) \cdot \boldsymbol{\Delta x} = \boldsymbol{\Delta \rho}$, where the right hand-side term has a fixed value (the extremal values of the sensor errors) and we have to verify that all solutions $\boldsymbol{\Delta x}$ of these systems are lower than a given threshold. When $\boldsymbol{p}, \boldsymbol{x}$ have interval values the matrix $\mathbf{J}$ has also interval coefficients and the linear system is called a *linear interval system*. Bounding the solutions of such system is a well known problem in interval analysis [2, 3]. But the usual methods of interval analysis do not take well into account that the coefficients of $\mathbf{J}$ are not independent (they are all functions of $\boldsymbol{x}$). We have thus developed a specific Gaussian elimination scheme and pre-conditioning of $\mathbf{J}$ that allows one to get sharper bounds on the solution $\boldsymbol{\Delta x}$.

Experimental Verification

To test the concept and the design methodology it has been decided to design a reduced scale (1/5 of the size of the real telescope) prototype. The design algorithm has been run for about 40 hours on a cluster of 20 computers to determine all possible design solutions. As a very large number of solutions have been obtained the company has added another requirement to choose the final design solution. Namely it was requested that the stiffness of the mechanism in its nominal position should be the highest possible. If k is the axial stiffness of the spreadband, then the stiffness matrix of the mechanism is obtained as $k\mathbf{J}^{\mathbf{T}}\mathbf{J}$. After an examination of the design solutions it was found out that a few of them were exhibiting better stiffness. We have thus chosen one of them to design the prototype.

Tests have then been performed in the clean room of a satellite manufacturer company. The first measurement tests were aimed at measuring the differences between the theoretical design solution and the prototype. It was found out that there were significant differences (although well within the manufacturing tolerances). The second series of tests has allowed to verify that the workspace of the mechanism was satisfactory: a series of extreme disturbances were simulated for the telescope and it was verified that the mechanism was able to correct all of them. The final series of tests was aimed at qualifying the positioning accuracy of the mechanism. A photogrammetry device was used to measure the absolute accuracy of the mechanism in various positions. It was found out that the positioning errors were well within the expected ranges for all the test positions.

Hence these series of tests have shown that the design methodology was indeed robust: in spite of significant differences between the prototype and its theoretical model there was an exact accordance with the expected workspace and accuracy performances and the measured ones.

References

[1] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag, 2001.

[2] R. Moore. *Methods and Applications of Interval Analysis*. SIAM Studies in Applied Mathematics, 1979.

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

Reliable Optimal Shear Force Design of Reinforced Concrete Beams

András Erik Csallner
Department of Computer Science,
University of Szeged,
H-6725 Szeged, Hungary;
csallner@jgytf.u-szeged.hu

András Balázs Kocsis
KÉSZ Ltd.,
H-6721 Szeged, Hungary;
kocsisa@kesz.hu

Tibor Csendes
Institute of Informatics,
University of Szeged,
H-6720 Szeged, Hungary;
csendes@inf.u-szeged.hu

Beams and Shear Reinforcement

Concrete beams occur in several kinds of engineering structures, e.g., in our houses we live and work in. A beam has to be designed to endure all the loads and effects that may occur during the construction and usage. From all the effects the most important is shear force because if a structure is under-designed for this then the result is usually a sudden implosion without any prior clear indication.
Concrete itself can resist compression to a reasonable level but it usually cannot absorb tension. Therefore in most of the cases steel reinforcement has to be applied. To resist shear force two kinds of reinforcement can be used. The first class consists of the stirrups and the second of the bent-up bars (inclined shear reinforcement). Stirrups are easier to mount since they are simple steel frames orthogonally embracing the core of the concrete beam and the stress-bars, which run longitudinally in the beams. But shear forces are not developed perpendicularly to the direct axis of the beam hence only a certain proportion of the stirrups' steel is exploited to absorb these effects. Bent-up bars can be inclined to desired angles but they are not only more expensive to be mounted but accordant to the European Standards at least fifty percent of the shear forces has to be absorbed by stirrups.

Overestimating Step Functions

As an input for a concrete beam design problem we have usually a load function which is either given analytically or graphically. From this the shear force function can be derived to determine the shear reinforcement. Unfortunately, the shear reinforcement always consists of separate steel parts. So even if the minimally necessary amount of steel can be calculated in each point along the beam the placement of the separate parts remains a question. However, this problem can be solved using step functions. The basic idea for the solution is the following:

1. Determine an optimal overestimating step function for the shear force function.
2. Calculate a minimal steel reinforcement for the step function.

The second stage of this method means a series of small finite optimization problems, the solution of which is a well-defined engineering problem, so we will not discuss this part here. The first stage, however, needs thorough examination.
It is obvious that the optimality of the step function highly depends on the width of the steps. If this width tends to zero, the overestimation tends to zero, as well. In practice the width of the steps is bounded from below, and this bound can be calculated from the kind of reinforcement used. In some cases the shear force function is piecewise linear. In this case the step function can be determined analytically. In the general case, however, it is far from being evident to find an optimal overestimating step function using just standard floating-point arithmetic. Here a step function can be calculated using interval algorithms and computations in the following way.
A simple approach is to subdivide the length of the beam into equally sized intervals and to calculate the upper bounds of the shear force function's range over these intervals. One way is to join some neighboring intervals to obtain an overestimating step function with wide enough steps. Another but more sophisticated solution is to find intervals with locally minimal or maximal shear force function upper bounds and try to find an optimal overestimating step function by laying line segments to these optima and calculating the optimal steps of this linear relaxation.
Several algorithms are under investigation and the participants of this project are working on finding the most convenient and robust way to be able to provide an optimal overestimator to any kind of shear force functions.

Acknowledgements

This work is supported by the University of Szeged, by the KÉSZ Ltd., and by the grants OTKA T046822, and OTKA T048377.

Result-Verifying Timing Analysis of Computer Chips

Michael Orshansky
Electrical and Computer Engineering,
The University of Texas at Austin,
Austin, TX 78712,USA;
orshansky@mail.utexas.edu

Vladik Kreinovich
Computer Science,
University of Texas at El Paso,
El Paso, TX 79968, USA;
vladik@utep.edu

Decreasing Clock Cycle: A Practical Problem

In chip design, one of the main objectives is to decrease the chip's clock cycle. It is therefore important to estimate the clock cycle at the design stage.

A computer chip must correctly perform billions of operations per second. We must guarantee that every elementary operation performed by the chip actually finishes within one clock cycle. The clock cycle of a chip is thus constrained by the maximum path delay over all the circuit paths $D \stackrel{\rm def}{=} \max(D_1, \ldots, D_N)$, where D_i is the delay along the path corresponding to the i-th elementary operation. Each path delay D_i is the sum of the delays d_{ik} corresponding to the logic gates and wires along this path. Each of these delays, in turn, depends on several factors x_j, such as the variation caused by the manufacturing process and the environmental design characteristics (e.g., variations in temperature and in supply voltage).

The difference Δx_j between the actual and the nominal values of factors x_j is usually small. So, one typically ignores quadratic (and higher order) terms in the Taylor expansion of the dependence of d_{ik} on Δx_j and assumes that the dependence of each delay d_{ik} on these differences can be described by a linear function. As a result, each path delay $D_i = \sum\limits_k d_{ik}$ is also linear in Δx_j, hence $D = \max\limits_i \left(a_i + \sum\limits_{j=1}^{n} a_{ij} \cdot \Delta x_j \right)$ for some coefficients a_i and a_{ij}.

Traditional Statistical Approach and Need for Result-Verifying Methods

In the traditional statistical timing analysis approach, we typically assume that the factors x_j are independent normally distributed random variables with known mean and variance. Based on this assumption, we can use Monte-Carlo or analytical techniques and come up with a value D_0 such that $D \le D_0$ with probability $\ge 1 - \varepsilon$ for some given small $\varepsilon > 0$.

The fundamental assumption behind these techniques is that the complete probabilistic descriptions are readily available. In a practical setting of cutting-edge IC design, the full probabilistic information about parameter uncertainty is not available, especially, at the ramp-up phase of the industrial manufacturing. For example, the uncertainty about supply voltage is most typically represented by the range information; the corresponding probability distribution is usually unknown. Similarly, we do not know the probability distributions corresponding to other sources of on-chip uncertainty such as temperature. Traditional statistical techniques cannot meaningfully handle such a realistic scenario.

Result-Verifying Methods

Usually, we know the intervals $[\underline{x}_j, \overline{x}_j]$ that are guaranteed to contain the (unknown) actual values of x_j. By using interval computations, we can compute an interval $[\underline{D}, \overline{D}]$ that is guaranteed to contain the actual maximum path delay D, and use the resulting guaranteed upper bound $\overline{D}$ as the clock cycle. Interval-related techniques – that take into consideration the linear dependence of D_i on Δx_j – are actually used in estimating the upper bounds $\overline{D}$. Verified bounds for more realistic models, in which the dependence of D_i on Δx_j is non-linear, can be obtained as well.

In addition to the guaranteed upper bound $\overline{D}$, it is desirable to provide a quantile bound D_0 such that $D \le D_0$ with probability $\ge 1 - \varepsilon$. To compute this bound, we use new result-verifying techniques that take into account not only the intervals $[\underline{x}_j, \overline{x}_j]$ but also known partial information about the probability distributions of x_j (e.g., moments of x_j).

For details, see [1, 2, 3] and references therein.

References

[1] M. Orshansky. Increasing circuit performance through statistical design techniques. In D. Chinnery and K. Keutzer, editors, *Closing the Gap Between ASICs and Custom*. Kluwer, Dordrecht, 2002.

[2] M. Orshansky and K. Keutzer. A general probabilistic framework for worst case timing analysis. In *Proc. ACM Design Automation Conference DAC'2002*, pages 556–561, New Orleans, Louisiana, USA, Dec. 2002.

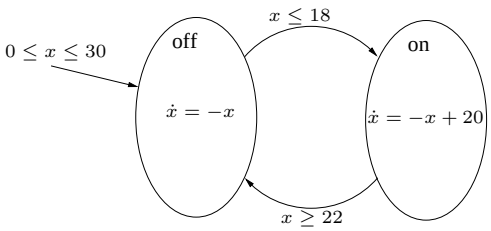
[3] M. Orshansky, W.-S. Wang, M. Ceberio, and G. Xiang. Interval-based robust statistical techniques for non-negative convex functions, with application to timing analysis of computer chips. In *Proc. ACM Symposium on Applied Computing SAC'06*, pages 1629–1633, Dijon, France, Apr. 2006.

Verification of Hybrid Systems

Stefan Ratschan
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
D-66123 Saarbrücken, Germany
stefan.ratschan@mpi-inf.mpg.de [1]

Problem Description

Imagine a thermostat controlling the heating of a room as follows: The heating can either be on or off. In either mode, the room temperature evolves according to the corresponding differential equation shown in the figure below. If the temperature drops below a value less or equal to 18 degrees, the heating switches on, if it reaches a value greater or equal to 22 degrees it switches off. We assume that at time zero the room temperature is between 0 and 30.



This is an example of a hybrid system—a dynamical system that shows both continuous and discontinuous state and evolution. In general, the state of a hybrid system can also be dis-continuously reset to a new value (that might or might not depend on the current value), and—in addition to differential equations—the continuous evolution can also be described by differential inequalities. All these conditions can be non-linear.

Such hybrid systems are an important tool for modeling embedded systems, where a digital controller acts on its environment. Modern machinery often contains thousands of embedded digital computing devices and it is essential to be able to reason about the resulting systems.

Usually one wants to prove properties such as safety (does the system always stay within a certain set of safe states, that is, does it never reach a certain set of unsafe states?), or stability (does the system in every case reach a certain set of target states and stay there?). In this article we will describe an algorithm that can prove the safety of hybrid systems.

[1] This work was partly supported by the German Research Council (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS). See `www.avacs.org` for more information.

Here one can observe the following problem characteristics:

- Hybrid systems are usually non-deterministic: the constraints defining the initial values, the evolution, and the unsafe values of a hybrid system do not have unique but, in general, uncountably many solutions, resulting in uncountably many trajectories that one might have to check.
- Manual rounding error analysis for hybrid systems is extremely difficult, due to the fact that even if the components of a hybrid system (the differential equations/inequalities, the conditions for switching between them) are well-conditioned, this does not imply a well-conditioned overall problem.
- Hybrid systems are often used to model safety-critical technical systems like cars or trains, where failure can result in loss of human life and large monetary costs.

The appeal of using result-verifying computing in this context lies in the fact that it can directly address these characteristics: intervals can be used to represent the sets of real numbers arising from non-determinism, and outward rounding guarantees that—independent from the problem conditioning—the computed result is provably correct. Such an approach has been pioneered by S. Kowalewski and O. Stursberg [3, e.g.,]. Here we present a newer, more general approach.

Constraint Propagation Based Abstraction Refinement

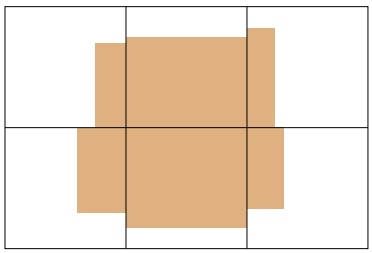
The approach decomposes—for each of the discrete modes—the state-space (which we assume to be bounded) into finitely many hyper-rectangles (*boxes*). Then it marks boxes that might contain an initial/unsafe point as initial/unsafe, and marks pairs of boxes through which a trajectory from an initial to an unsafe state might run, using edges of a directed graph. The result can be viewed as finite directed graph for which some nodes are marked as being initial or unsafe. If this graph (the *abstraction*) cannot be traversed from an initial to an unsafe node then the original hybrid system is safe. If the abstraction can be traversed in such a way, then this might be either due to a corresponding trajectory in the original system or due to the over-approximation introduced by the abstraction process. To exclude the second case, one can then refine the abstraction by splitting boxes into more pieces and re-computing the abstraction.

In our approach [2] we compute and refine abstractions using a constraint solver [1] based on interval constraint propagation techniques (a certain way of result-verifying computation with intervals). This solver can

take as input a constraint in a very general form (formally speaking: a formula in the first-order predicate language over the real-numbers), and a box $[\boldsymbol{b}]$, and returns a sub-box $[\boldsymbol{b}']$ of $[\boldsymbol{b}]$ such that $[\boldsymbol{b}] \setminus [\boldsymbol{b}']$ does not contain any solution of the input constraint (i.e., no solutions are lost when replacing $[\boldsymbol{b}]$ by $[\boldsymbol{b}']$).

Using such constraints (in which the differentiation symbol is not allowed), one can formulate necessary conditions for the fact that a box contains an initial/unsafe element, and for the fact that a trajectory is possible from one point of the state space to the other. Then we mark graph nodes as initial/unsafe if the constraint solver cannot disprove the corresponding condition, and put an edge between nodes if the constraint solver cannot disprove the condition that there may be a trajectory between the two corresponding boxes.

Moreover, in order to deal with the curse of dimensionality that can be experienced by excessive box splitting when refining the abstraction, in addition, we remove parts from boxes for which these conditions show that they cannot be on a trajectory from an initial to an unsafe state (e.g., replace the lined boxes in the figure below by the shaded ones). This allows us, in some cases, to reflect more information about the hybrid system in the abstraction without introducing new boxes.



The constraints that we currently use are based on certain re-formulations of the mean-value theorem, and—in the linear case—on the explicit solution of differential equations. One of the advantages of using a general constraint language with a corresponding solver is that the user is able to formulate new, problem-specific, constraints that can then be used in the verification process.

Conclusion

The verification of hybrid systems has high practical importance due to the fact that hybrid systems are widely used to model embedded computing devices. Here, result-verifying computation offers a solution to the fact that hybrid systems are usually nondeterministic, that their error analysis is extremely difficult, and that hybrid systems often occur in a safety critical context. According algorithms can try to limit the curse of dimensionality using interval constraint propagation techniques, as implemented in existing constraint solvers.

References

[1] S. Ratschan. RSOLVER. `http://rsolver.sourceforge.net`, 2004. Software package.

[2] S. Ratschan and Z. She. Safety verification of hybrid systems by constraint propagation based abstraction refinement. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control*, volume 3414 of *LNCS*. Springer, 2005.

[3] O. Stursberg and S. Kowalewski. Analysis of controlled hybrid processing systems based on approximation by timed automata using interval arithmetic. In *Proceedings of the 8th IEEE Mediterranean Conference on Control and Automation (MED 2000)*, 2000.

Robust Design of Dynamic Systems

Wolfgang Marquardt, Martin Mönnigmann
Process Systems Engineering,
RWTH Aachen University,
D-52056 Aachen, Germany;
{*marquardt,moennigmann*}*@lpt.rwth-aachen.de*

Thomas Beelitz, Bruno Lang, Paul Willems
Applied Computer Science and Scientific Computing,
University of Wuppertal,
D-42097 Wuppertal, Germany;
{*beelitz,lang,willems*}*@math.uni-wuppertal.de*

Christian H. Bischof
Scientific Computing and CCC,
RWTH Aachen University,
D-52056 Aachen, Germany;
bischof@sc.rwth-aachen.de

Problem Description

Dynamic systems, such as chemical processes, often can be modeled as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}), \tag{1}$$

where $\boldsymbol{x}$ and $\boldsymbol{p}$ denote the systems' state variables and parameters, respectively. Finding optimal steady states then corresponds to an optimization problem

$$\min_{\boldsymbol{x},\boldsymbol{p}} \phi(\boldsymbol{x}, \boldsymbol{p}) \quad \text{subject to} \quad \begin{aligned} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}) &= \boldsymbol{0}, \\ \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{p}) &\leq \boldsymbol{0}. \end{aligned} \tag{2}$$

Here, ϕ is a cost function, and the equations and inequalities characterize steady states and operational constraints, respectively. Unfortunately, the optimization problem (2) does not take the dynamics of the system (1) into account, and therefore its solution may be instable and thus not usable in practice.
Our aim is to choose the parameters $\boldsymbol{p}$ in such a way that the cost function is minimized while avoiding the "critical points" where stability gets lost, i.e., points where the matrix $\partial \boldsymbol{f}/\partial \boldsymbol{x}$ has a zero or purely imaginary eigenvalue (saddle–node or Hopf bifurcation, respectively). Since the parameters p_i can be set only up to certain tolerances Δ_i, a whole *box* $[\boldsymbol{p}] = [\boldsymbol{p}^* - \Delta, \boldsymbol{p}^* + \Delta]$ around the prospective point of operation must be free from critical points to ensure safe operation. This amounts to keeping a prescribed distance from the critical points.

Staying Away from Critical Points

One approach to optimal process design is to first optimize the parameters and then check for stability by exploring the vicinity of the solution, e.g., with continuation methods. Besides a waste of resources in case of instable optima this approach cannot guarantee robustness even if carried out by an experienced engineer.
Our *hybrid* approach aims at avoiding these two problems. Robustness is addressed already in the optimization phase. To this end we monitor the eigenvalues of $\partial \boldsymbol{f}/\partial \boldsymbol{x}$ at the points $\boldsymbol{p}^k$ that are generated during the solution of the optimization problem (2). Thus we can detect when a manifold of critical points has been crossed. If this is the case then additional "normal vector" constraints are added to the optimization problem, which ensure that only points $\boldsymbol{p}$ with a prescribed distance to this manifold will be considered later on. Then the optimization is resumed with these additional constraints until an optimum is found or another manifold of critical points is crossed, in which case further normal vector constraints are added.

Verification of Robustness

The optimization with normal vector constraints cannot completely exclude that there are critical points in the vicinity of the computed optimum $\boldsymbol{p}^*$, for two reasons. First, a manifold may have been crossed twice, which can go unnoticed, and second, the critical points may be slightly off the optimizer's path. Therefore in a second stage our new result-verifying nonlinear solver SONIC is used to guarantee the robustness of the solution. More precisely, we check if neither the "augmented" system

$$\boldsymbol{f} = \boldsymbol{0}, \quad (\partial \boldsymbol{f}/\partial \boldsymbol{x}) \cdot \boldsymbol{v} = \boldsymbol{0}, \quad \|\boldsymbol{v}\|^2 = 1$$

characterizing saddle–node bifurcations nor a similar system for Hopf bifurcations has a solution in the box $[\boldsymbol{p}^* - \Delta, \boldsymbol{p}^* + \Delta]$. Some of SONIC's features are tailored particularly to augmented systems, but the tool turned out to be efficient in solving nonlinear systems and optimization problems from other areas as well.
Note that the use of a result-verifying constrained optimizer from the beginning would render the second stage superfluous, but the hybrid approach is more cost-effective. A more detailed description of the new method and additional references may be found in [1].
With our approach, an optimal robust design could be obtained for a simple fermenter model in a fully automated way, taking a few seconds for the (non-rigorous) optimization and a few minutes for the rigorous stage.

References

[1] M. Mönnigmann, W. Marquardt, C. H. Bischof, T. Beelitz, B. Lang, and P. Willems. A hybrid approach for efficient robust design of dynamic systems. Preprint BUW-SC 04/9, 2004. `http://www.math.uni-wuppertal.de/SciComp/Preprints`.

Reliable Multiprecision Evaluation of Special Functions

Stefan Becuwe[1], Annie Cuyt
Departement Wiskunde en Informatica,
Universiteit Antwerpen,
Middelheimlaan 1, B-2020 Antwerpen, Belgium;
{*stefan.becuwe, annie.cuyt*}*@ua.ac.be*

Introduction

Special functions are pervasive in all fields of science and industry. The most well-known application areas are in physics, engineering, chemistry, computer science and statistics. Because of their importance, several books and a large collection of papers have been devoted to algorithms for the numerical computation of these functions.

Virtually all present-day computer systems, from personal computers to the largest supercomputers, implement the IEEE 64-bit floating-point arithmetic standard, which provides 53 binary or approximately 16 decimal digits accuracy. For most scientific applications, this is more than sufficient. However, for a rapidly expanding body of applications, 64-bit IEEE arithmetic is no longer sufficient. These range from some interesting new mathematical investigations to large-scale physical simulations performed on highly parallel supercomputers. Moreover in these applications, portions of the code typically involve numerically sensitive calculations, which produce results of questionable accuracy using conventional arithmetic. These inaccurate results may in turn induce other errors, such as taking the wrong path in a conditional branch. Such blocks of code benefit enormously from a combination of reliable numeric techniques and the use of high-precision arithmetic. Indeed, the aim of reliable numeric techniques is to deliver, together with the computed result, a guaranteed upper bound on the total error or, equivalently, to compute an enclosure for the exact result.

Instead of high accuracy, some applications only require a very modest but guaranteed number of significant digits. These applications can profit from a reliable implementation with scalable precision. For instance, in electromagnetic simulation models the required accuracy is usually in the order of only 2 to 3 significant digits.

Up to this date, even environments such as Maple, Mathematica, MATLAB and libraries such as IMSL, CERN and NAG offer no routines for the reliable evaluation of special functions. The following quotes concisely express the need for new developments in the evaluation of special functions:

- *"Algorithms with strict bounds on truncation and rounding errors are not generally available for special functions. These obstacles provide an opportunity for creative mathematicians and computer scientists."* Dan Lozier, general director of the DLMF project, and Frank Olver [2].
- *"The decisions that go into these algorithm designs — the choice of reduction formulae and interval, the nature and derivation of the approximations — involve skills that few have mastered. The algorithms that MATLAB uses for gamma functions, Bessel functions, error functions, Airy functions, and the like are based on Fortran codes written 20 or 30 years ago."* Cleve Moler, founder of MATLAB [5].

Implementing a Function Library

The realization of a machine implementation of a function $f(x)$ is a three-step process.

1. For a given argument x, the evaluation $f(x)$ is often reduced to the evaluation of f for another argument $\tilde{x}$ lying within specified bounds and for which there exists an easy relationship between $f(x)$ and $f(\tilde{x})$. For instance, for the exponential function in a base β implementation,

$$\exp(x) = \beta^k \exp(\tilde{x}),$$
$$\tilde{x} = \mathrm{mod}(x, \ln \beta), \qquad |\tilde{x}| \leq \ln \frac{\beta}{2}.$$

Although the given argument x is known exactly, because it is a given floating-point number, usually the reduced argument $\tilde{x}$ cannot be computed exactly, but is subject to a rounding error. The issue of argument reduction is a topic in its own right and mostly applies to only the simplest transcendental functions such as the elementary functions.

2. After the reduced argument is determined, the mathematical model F for f is constructed and a truncation error

$$\frac{|f(\tilde{x}) - F(\tilde{x})|}{|f(\tilde{x})|}$$

comes into play, which needs to be bounded.

3. When implemented, in other words evaluated as $\mathtt{F}(\tilde{x})$, this mathematical model is also subject to a rounding error

$$\frac{|F(\tilde{x}) - \mathtt{F}(\tilde{x})|}{|f(\tilde{x})|},$$

which needs to be controlled.

[1]The author is supported by the Institute for the Promotion of Innovation through Science and Technology in Flanders.

Finally the effect of switching from the argument x to the reduced argument $\tilde{x}$ must be taken into account. This introduces a final additional error.

Toolkit for a Reliable Library

The technique to provide a floating-point model $F(x)$ of a function $f(x)$ differs substantially when going from a fixed finite precision context to a finite multiprecision context. In the former, the aim is to provide an optimal mathematical model, valid on a reduced argument range and requiring as few operations as possible. Here, "optimal" means that, in relation to the model's complexity, the truncation error is as small as it can get. The total relative error should not exceed a prescribed threshold, round-off error and possible argument reduction effect included. In the latter, the goal is to provide a more generic technique, from which an approximant yielding the user-defined accuracy, can be deduced *at runtime*. Hence best approximants are not an option since these models have to be recomputed every time the precision is altered and a function evaluation is requested. At the same time the generic technique should generate an approximant of as low complexity as possible.

We aim, on the one hand, at a generic technique suitable for use in a multiprecision context, which on the other hand is efficient enough to compete with the traditional hardware algorithms. We also want our implementation to be reliable, in the sense that a sharp interval enclosure for the requested function evaluation is returned without any additional cost.

Besides series representations, continued fraction representations of functions can be very helpful in the multiprecision context. A lot of well-known constants in mathematics, physics and engineering, as well as elementary and special functions enjoy very nice and rapidly converging continued fraction representations. In addition, many of these fractions are limit-periodic, meaning that the partial numerators and denominators converge.

It is well-known that the tail or rest term of a convergent Taylor series expansion converges to zero. It is less well-known that the tail of a convergent continued fraction representation does not need to converge to zero; it does not even need to converge at all. In order to develop a useful continued fraction technique, we first need to obtain sharp a priori truncation error estimates for a general class of continued fractions, taking into account that a suitable approximation of the disregarded continued fraction tail may speed up the convergence of the continued fraction approximants. Hence the truncation error estimate needs to be valid for use with nonzero continued fraction tail estimates. Such estimates are developed in the framework of this project [3]. The rounding error involved can subsequently be bounded by a classical result obtained in [4].

Special Function Coverage

The implementation will be made available in two forms: as a C/C++ library and as a Maple library. Both will make use of the fully IEEE 754-854 compliant multiprecision library `MpIeee`, in which the user can select the base β, precision t and exponent range $[L, U]$ of the computations. We aim, for the evaluation of all functions, at a relative error $|f(x) - \mathtt{F}(x)|/|f(x)|$ bounded above by 1 ULP (Unit-in-the-Last-Place) or $\beta^{-(t-1)}$.

Which special functions will be supported? Among the special functions that enjoy rapidly converging limit-periodic continued fraction representations are the ones listed in the table below. In the column marked AS, we indicate whether the standard work [1] contains at least one continued fraction representation for the function in question. The second column, marked CFHB, tells us whether our new handbook [3] contains a useful continued fraction representation for the purpose. In the third column we have added for which special functions an implementation is (✓) or will be (✓*) available soon.

	AS	CFHB	
elementary functions	✓	✓	✓
$\psi_1(z)$, $\psi_2(z)$		✓	✓*
$\gamma(a, z)$		✓	✓
$\Gamma(a, z)$	✓	✓	✓
$\mathrm{erf}(z)$	✓	✓	✓
$\mathrm{erfc}(z)$		✓	✓
$C(z)$, $S(z)$ (Fresnel)		✓	
$E_n(z)$	✓	✓	✓
${}_2F_1(a, b; c; z)$		✓	✓
${}_1F_1(a; b; z)$		✓	✓
$J_\nu(z)$ (Bessel)	✓	✓	✓*
$I_\nu(z)$ (Bessel)		✓	✓*
$I_x(a, b)$ (beta)	✓	✓	✓*

References

[1] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, volume 55 of *NIST*. 1964.

[2] B. A. Cipra. A new testament for special functions? *SIAM News*, 31(2), 1998.

[3] A. Cuyt, V. Petersen, B. Verdonk, H. Waadeland, W. B. Jones, and C. Bonan-Hamada. *Handbook of Continued Fractions for Special Functions*. Kluwer Academic Publishers, 2006.

[4] W. B. Jones and W. J. Thron. Numerical stability in evaluating continued fractions. *Math. Comp.*, 28:795–810, 1974.

[5] C. Moler. The tetragamma function and numerical craftmanship. *MATLAB News & Notes*, 2002.

On Verification of Ill-posed Optimization Problems

Christian Jansson
Institute for Reliable Computing,
TU Hamburg-Harburg,
D-21071, Germany;
jansson@tu-harburg.de

Ill-conditioned or ill-posed optimization problems occur rather frequently, for example by using relaxation techniques for solving combinatorial optimization problems. In this case rounding errors may produce erroneous results, although this deterministic method should compute the exact solution in a finite number of steps. Neumaier and Shcherbina [2004] [1] present an innocent-looking linear integer problem where the commercial, state-of-the-art solvers CPLEX, BONSAIG, GLPK, XPRESS, XPRESS-MP/INTEGER, and MINLP failed. The reason is that the ill-conditioned linear relaxations are not solved rigorously. Several semidefinite relaxations of Graph Partitioning, Quadratic Assignment and Max Cut Problems are ill-posed (see Gruber and Rendl [2002]). Several other problems become ill-posed due to the modelling (for example problems with redundant constraints, identically zero variables, and free variables transformed to variables bounded on one side).
In [1] it is shown how verified results can be obtained by rigorously bounding the optimal value of semidefinite relaxations, even in the ill-posed case. All rounding errors due to floating point arithmetic are taken into account. Numerical results with up to thousands of constraints and variables are presented there. In the following, we shortly describe the rigorous lower bound of the optimal value for *semidefinite programming problems* in *block diagonal form*:

$$\begin{aligned} &f_p^* := \min \sum_{j=1}^{n} \langle \mathbf{C}_j, \mathbf{X}_j \rangle \text{ s.t.} \\ &\sum_{j=1}^{n} \langle \mathbf{A}_{ij}, \mathbf{X}_j \rangle = b_i,\ i = 1, \ldots, m, \mathbf{X}_j \succeq 0, \end{aligned} \tag{1}$$

where $\boldsymbol{b} \in \mathbb{R}^m$ and $\mathbf{C}_j, \mathbf{A}_{ij}, \mathbf{X}_j \in S^{s_j}$, the linear space of real symmetric $s_j \times s_j$ matrices. By $\langle ., . \rangle$ we denote the usual inner product on the linear space of symmetric matrices, which is defined as the trace of the product of two matrices. $\mathbf{X} \succeq 0$ means that $\mathbf{X}$ is positive semidefinite. Hence, $\succeq$ denotes the *Löwner partial order* on this linear space. It is easy to see that many convex optimization problems can be formulated as semidefinite programming problems, for example linear programming or second order cone programming.

[1] All references can be found in [1]

Let $\tilde{\boldsymbol{y}} \in \mathbb{R}^m$ be an approximately computed Lagrange multiplier of the semidefinite program, and assume that the following *primal boundedness qualification* holds valid: there are simple nonnegative (possibly infinite) bounds $\overline{x}_j$ such that an optimal solution $(\mathbf{X}_j)$ satisfies

$$\lambda_{\max}(\mathbf{X}_j) \leq \overline{x}_j \text{ for } j = 1, \ldots, n, \tag{2}$$

where $\lambda_{\max}$ denotes the largest eigenvalue. Let

$$\mathbf{D}_j := \mathbf{C}_j - \sum_{i=1}^{m} \tilde{y}_i \mathbf{A}_{ij} \text{ and } \underline{d}_j \leq \lambda_{\min}(\mathbf{D}_j). \tag{3}$$

Assume that for $j = 1, \ldots, n$ the defect $\mathbf{D}_j$ has at most l_j negative eigenvalues. Then it can be proved that

$$f_p^* \geq \boldsymbol{b}^T \tilde{\boldsymbol{y}} + \sum_{j=1}^{n} l_j \overline{x}_j \min\{0, \underline{d}_j\} =: \underline{f}_p^*. \tag{4}$$

This rigorous lower bound can easily be computed by using interval arithmetic and a verification method for eigenvalues.
In the following table, we display some numerical results for ill-posed relaxations of Graph Partitioning which are given by Gruber and Rendl [2002] (n: number of vertices of the graph, t: time required for computing the approximate value $\tilde{f}_p^*$ with the semidefinite solver SDPT3, t_1: time for computing the rigorous lower bound, and $\mu(\tilde{f}_p^*, \underline{f}_p^*)$: relative error). For these ill-posed problems with up to 600 constraints and 180000 variables one can see that the additional time t_1 for the rigorous lower bound is negligible compared to the time required for the approximations, and that the relative errors are small.

n	t	t_1	$\mu(\tilde{f}_p^*, \underline{f}_p^*)$
200	8.81	0.19	6.86788e-008
400	41.27	0.89	3.82904e-007
600	131.47	2.69	1.05772e-006

In a recent paper, Ordóñez and Freund [2003] stated that 71% of the lp-instances in the NETLIB Linear Programming Library are ill-posed. This library contains many industrial problems. For rigorous numerical results of this test suite see Jansson and Keil [2004].
Summarizing, the theory described in [1] facilitates cheap and rigorous lower and upper bounds for the optimal value of convex optimization problems. These bounds can be used for verification of combinatorial and global optimization problems.

References

[1] C. Jansson. Termination and Verification for Ill-posed Semidefinite Programming Problems, 2005, submitted. `http://optimization-online.org/DB_HTML/2005/06/1150.html`