SCAN'2012

September 23–29
Novosibirsk, Russia

15th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetics and Verified Numerics

Book of Abstracts

15th GAMM–IMACS International Symposium on Scientific Computing, Computer Arithmetics and Verified Numerics

SCAN'2012

Novosibirsk, Russia
September 23–29, 2012

Book of Abstracts

Institute of Computational Technologies
Novosibirsk, 2012

SCIENTIFIC COMMITTEE

- Götz Alefeld (Karlsruhe, Germany)
- Jean-Marie Chesneaux (Paris, France)
- George F. Corliss (Milwaukee, USA)
- Tibor Csendes (Szeged, Hungary)
- Andreas Frommer (Wuppertal, Germany)
- R. Baker Kearfott (Lafayette, USA)
- Walter Krämer (Wuppertal, Germany)
- Vladik Kreinovich (El Paso, USA)
- Ulrich Kulisch (Karlsruhe, Germany)
- Wolfram Luther (Duisburg, Germany)
- Svetoslav Markov (Sofia, Bulgaria)
- Günter Mayer (Rostok, Germany)
- Jean-Michel Muller (Lyon, France)
- Mitsuhiro Nakao (Fukuoka, Japan)
- Michael Plum (Karlsruhe, Germany)
- Nathalie Revol (Lyon, France)
- Jiří Rohn (Prague, Czech Republic)
- Siegfried Rump (Hamburg, Germany)
- Sergey P. Shary (Novosibirsk, Russia)
- Yuri I. Shokin (Novosibirsk, Russia)
- Wolfgang V. Walter (Dresden, Germany)
- Jürgen Wolff von Gudenberg (Würzburg, Germany)
- Nobito Yamamoto (Tokyo, Japan)

WEB-SITE

`http://conf.nsc.ru/scan2012`

ORGANIZERS

- Institute of Computational Technologies SD RAS, http://www.ict.nsc.ru
- Novosibirsk State University, http://www.nsu.ru
- Novosibirsk State Technical University, http://www.nstu.ru
- "Scientific Service" Ltd.

SPONSOR

Russian Foundation for Basic Research (RFBR)
http://www.rfbr.ru

ORGANIZING COMMITTEE

- Sergey P. Shary shary@ict.nsc.ru
- Irene A. Sharaya sharaya@ict.nsc.ru
- Yuri I. Molorodov yumo@ict.nsc.ru
- Svetlana V. Zubova zub@bionet.nsc.ru
- Andrei V. Yurchenko
- Vladimir A. Detushev
- Dmitri Yu. Lyudvin

E-MAIL

scan2012@ict.nsc.ru

Preface

This volume contains peer refereed abstracts of the 15th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Verified Numerical Computations, Novosibirsk, September 23–29, 2012.

This conference continues the series of international SCAN symposia initiated by University of Karlsruhe, Germany, and held under the joint auspices of GAMM and IMACS. SCAN symposia have been held in many cities across the world:

Karlsruhe, Germany (1988)
Basel, Switzerland (1989)
Albena-Varna, Bulgaria (1990)
Oldenburg, Germany (1991)
Vienna, Austria (1993)
Wuppertal, Germany (1995)
Lyon, France (1997)
Budapest, Hungary (1998)
Karlsruhe, Germany (2000)
Paris, France (2002)
Fukuoka, Japan (2004)
Duisburg, Germany (2006)
El Paso, Texas, USA (2008)
Lyon, France (2010)

SCAN'2012 strives to advance the frontiers in verified numerical computations, interval methods, as well as their application to computational engineering and science. Topics of interest include, but are not limited to:

- theory, algorithms, and arithmetics for verified numerical computations
- hardware and software support, programming tools for verification
- symbolic and algebraic methods for numerical verification
- verification in operations research, optimization, and simulation
- verified solution of ordinary differential equations
- computer-assisted proofs and verification for partial differential equations
- interval analysis and its applications
- supercomputing and reliability
- industrial and scientific applications of verified numerical computations

We want to thank all contributors and participants of symposium SCAN'2012. Without their active participation, we would not have succeeded.

Local Organizers

Contents

Solvability of systems of interval linear equations via the codifferential descent method

Todor Angelov

Saint-Petersburg State University
35, Universitetskii prospekt
198504 Saint-Petersburg, Russia
angelov.t@gmail.com

Keywords: linear interval equations, solvability, nonsmooth analysis, codifferential calculus

A system of linear interval equations

$$\boldsymbol{A}x = \boldsymbol{b} \tag{1}$$

is considered in the works [1-3]. Here $\boldsymbol{A} = (\boldsymbol{a}_{ij})$ is an interval $m \times n$-matrix, and $\boldsymbol{b} = (\boldsymbol{b}_i)$ is an interval m-vector.

We need the following definitions: $\boldsymbol{a} = [\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}] = \{\, x \in \mathbb{R} \mid \underline{\boldsymbol{a}} \leq x \leq \overline{\boldsymbol{a}}\}$, $\operatorname{mid} \boldsymbol{a} = \frac{1}{2}(\overline{\boldsymbol{a}} + \underline{\boldsymbol{a}})$, $\operatorname{rad} \boldsymbol{a} = \frac{1}{2}(\overline{\boldsymbol{a}} - \underline{\boldsymbol{a}})$, and $\langle \boldsymbol{a} \rangle = \max\{0, \underline{\boldsymbol{a}}, -\overline{\boldsymbol{a}}\}$.

By the (weak) *solution set* to a system of linear interval equations (1), we mean the set

$$\Xi(\boldsymbol{A}, \boldsymbol{b}) = \{x \in \mathbb{R}^n \mid Ax = b \text{ for some } A \in \boldsymbol{A},\, b \in \boldsymbol{b}\}\,,$$

constructed of all possible solutions of the systems $Ax = b$ with $A \in \boldsymbol{A}$ and $b \in \boldsymbol{b}$ [2,3].

Statement [1]**.** The expression

$$\operatorname{Uni}(x, \boldsymbol{A}, \boldsymbol{b}) = \min_{1 \leq i \leq m} \left\{ \operatorname{rad} \boldsymbol{b}_i - \left\langle \operatorname{mid} \boldsymbol{b}_i - \sum_{j=1}^{n} \boldsymbol{a}_{ij} x_j \right\rangle \right\}$$

defines the functional $\operatorname{Uni} : \mathbb{R}^n \to \mathbb{R}$, such that the membership of a vector $x \in \mathbb{R}^n$ in the solution set $\Xi(\boldsymbol{A}, \boldsymbol{b})$ of the system of linear interval equations $\boldsymbol{A}x = \boldsymbol{b}$ is equivalent to nonnegativity of the functional Uni in x,

$$x \in \Xi(\boldsymbol{A}, \boldsymbol{b}) \quad \Longleftrightarrow \quad \operatorname{Uni}(x, \boldsymbol{A}, \boldsymbol{b}) \geq 0.$$

Consider a locally Lipschitz function f defined on an open set $X \subset \mathbb{R}^n$.

Definition [4]**.** A function $f : X \to \mathbb{R}$ is called codifferentiable at a point $x \in X$ if there exist compact convex sets $\underline{d}f(x) \subset \mathbb{R}^{n+1}$ and $\overline{d}f(x) \subset \mathbb{R}^{n+1}$ such that the following expansion holds

$$f(x+\Delta) = f(x) + \max_{[a,v]\in \underline{d}f(x)} \{a + (v,\Delta)\} + \min_{[b,w]\in \overline{d}f(x)} \{b + (w,\Delta)\} + o(x,\Delta),$$

where $\dfrac{o(x,\Delta)}{\|\Delta\|} \longrightarrow 0$ as $\|\Delta\| \to 0$, $a, b \in \mathbb{R}$, $v, w \in \mathbb{R}^n$.

The pair $Df(x) = [\underline{d}f(x), \overline{d}f(x)]$ is called the codifferential of f at x. A function f is called continuously codifferentiable at a point $x \in X$ if it is codifferentiable in a neighborhood of x and if there exists a codifferential mapping Df which is Hausdorff continuous at the point x.

It turns out that most known nonsmooth functions, as well as the functional Uni, are continuously codifferentiable. The codifferential mapping has the property to identify sets of points of nondifferentiability. Note that Uni is multi-extremal, and its graph is constructed of a finite number of hyperplanes. In general, the local minima points of $-$ Uni lie on intersections of these hyperplanes, which appear to be sets of nondifferentiability of $-$ Uni. This allows the codifferential descent method [4] to reach the local minima points of $-$ Uni in one or a small amount of iterations. Also, the proposed method has the property to "jump out" of local minima points and descent further.

In comparison, Shary in his paper [1] proposes a solution to the optimization of Uni, based on the fact that Uni is concave in every orthant of $\mathbb{R}^n$. Therefore, the localizations of Uni can be studied by means of tools of convex analysis.

References:

[1] S.P. Shary, Solvability of interval linear equations and data analysis under uncertainty, *Automation and Remote Control*, 73 (2012), No. 2, pp. 310–322.

[2] S.P. Shary, *Finite-dimensional Interval Analysis*, Novosibirsk, 2011. Electronic book, accessible at http://www.nsc.ru/interval/Library/InteBooks (in Russian)

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

[4] V.F. Demyanov, A.M. Rubinov, *Constructive Nonsmooth Analysis*, Verlag Peter Lang, Frankfurt-am-Main, 1995.

Uses of verified methods for solving non-smooth initial value problems

Ekaterina Auer and Stefan Kiel

Faculty of Engineering, INKO
University of Duisburg-Essen
D-47048 Duisburg, Germany
{auer, kiel}@inf.uni-due.de

Keywords: IVP, ODE, non-smooth systems, interval methods

Many system types from the area of engineering require mathematical models involving non-differentiable or discontinuous functions [1]. The non-smoothness can be obvious, such as that in commonly used models for friction or contact. There are also more obscure cases occurring, for example, in computer-based simulations where `if-then-else` or similar conditions are used on model variables. The task of finding reliable solutions becomes especially difficult if non-smooth functions appear on the right side of an initial value problem (IVP). On the one hand, such system models are often sensitive to round-off errors. On the other hand, their parameters might be uncertain due to impreciseness in measurements or lack of knowledge. Therefore, interval methods represent a straightforward choice for verified analysis of such systems. They guarantee the correctness of results obtained on a computer and can represent purely epistemic bounded uncertainty in a natural way.

However, the application of the existing interval methods to real-life scenarios is challenging since they might provide overly conservative enclosures of exact solutions. Even in the case of simple jump discontinuities, where the solution is not differentiable in just several switching points, the accuracy is poor and, consequently, the resulting enclosures might be too wide [4]. This is probably the reason for the relatively little attention the non-smooth problems have got in the last decades whereas verified solution of smooth IVPs has been extensively explored. For example, there exists no publicly available verified implementation of a non-smooth IVP solver at the moment to our knowledge. Nonetheless, meaningful outcomes can still be obtained as is demonstrated in this talk for several examples.

In our contribution, we identify important types of non-smooth application along with their corresponding solution definitions first. Second, we provide an overview of the existing techniques for verified enclosure of exact solutions to

non-smooth IVPs [2,3,4] and assign a suitable solution method to each of the application types mentioned above. After that, we focus our considerations on a special case in which the switching points are known a priori in a certain sense. For this situation, we describe a simple method to solve non-smooth IVPs using basically the same techniques as in the smooth case. Finally, we demonstrate the applicability of the method using several examples.

References:

[1] A. Filippov, *Differential Equations With Discontinuous Righthand Sides*, Kluwer Academic Publishers, 1988.

[2] 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.

[3] 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.

[4] R. Rihm, Enclosing solutions with switching points in ordinary differential equations, In *Computer arithmetic and enclosure methods. Proceedings of SCAN 91*, Amsterdam: North-Holland, 1992, pp. 419–425.

Interval of uncertainty in the solution of inverse problems of chemical kinetics

Fayruza Badrtdinova

Birsk State Socially-pedagogical Academy,
International, 10, 452453, Birsk, Russia
fairusa85@mail.ru

Keywords: interval of uncertainty, chemical kinetics

The inverse problem of chemical kinetics is a problem of identifying reactions and the rate constants, as well as the other kinetic parameters associated with these reactions. Solving this problem is often obstructed with ambiguity upon the estimation of specific kinetic parameters. Such an ambiguity reflects the nature of a kinetic model that describes only some features of chemical reactions in a certain area of the reaction. In the inverse problem of chemical kinetics, being a problem of identifying reaction factors, it is necessary to evaluate the uncertainty limits for the kinetic parameter estimates. For this purpose, we suggest using a method that is based on the Kantorovich idea [1], when only knowledge of maximal experimental errors is used. Each measured value is considered to be an interval $[k]$ that is a set of all possible values k bounded by inequalities $k^- \leq k \leq k^+$ [2]. Under this assumption, each kinetic parameter can be estimated by a region, whose every point is a result of a numerical simulation of the reaction. Considering all these regions together we obtain a multidimensional area that consists of the points that represent a valid set of the kinetic parameters.

The parameters of chemical kinetics k are found from the differential equations by solving the inverse problem. Depending on the type of the experiment, the system of differential equations of chemical kinetics has different forms:

1) non-steady state experiment $dx/dt = f_1(x, y, k)$, $dy/dt = f_2(x, y, k)$,

2) quasi-steady state experiment $dx/dt = f_1(x, y, k)$, $f_2(x, y, k) = 0$,

3) equilibrium $f_1(x; y; k) = 0$, $f_2(x; y; k) = 0$,

where x is the vector of the measurable compounds; y is the vector of the compounds that cannot be measured.

There are but a few methods for determining uncertainty ranges. For example, the direct search method, which is the simplest but the slowest. Its drawback is that the function to be minimized has to be calculated many times.

In this study, we consider a method based on L.V. Kantorovich's idea. The following problem is set: for each constant, the uncertainty range (more exactly, its boundaries) has to be found. To find the range for the constant k_j, we need to determine $\min k_j$ and $\max k_j$ under the condition that the restrictions

$$|x_{exp} - x_{calc}| \leq \varepsilon \tag{1}$$

are satisfied.

During the search, the direct kinetic problem is solved with a certain set of constants, simultaneously checking that the concentrations found satisfy the inequality (1). If the concentration values computed satisfy the inequality, then the given set of constants belongs to the desired uncertainty range. To find a boundary of the desired region d_j, a certain set of constants has to be taken, all constants being fixed except one, for example k_j. The set of constants is determined from a solution of the inverse problem.

The following algorithm for finding the uncertainty range by constant k_j is considered.

Some value of the constant that satisfies (1) is considered as the initial approximation. Such a value can be found by minimization of a criterion that takes into account the discrepancy between calculations and measurements. Let us assume that an initial point $(k_1^0, ..., k_m^0)$ is found and an initial step h_0 is chosen. To find the desired range for the j-th constant, we determine $\max k_j$ (the algorithm for finding $\min k_j$ is the same, but the step must be taken with a 'minus' sign. By adding the h_0 step to k_j^0, we obtain the following set of constants: $(k_1^0, ..., k_j^0 + h_0, ..., k_m^0)$. Now we solve the direct problem with the available set of constants and check the consistency of inequality (1). If the inequality is satisfied, the point $k_j^0 + h_0$ belongs to the desired range, and we continue to move to the right if we are searching for $\max k_j$. If inequality (9) is not satisfied at the point $k_j^0 + h_0$, we decrease the step twofold, $h_1 = \frac{h_0}{2}$, and add it to k_j^0 to obtain a new set of constants $(k_1^0, ..., k_j^0 + h_1, ..., k_m^0)$. We solve the direct problem with the resulting set of constants and check the consistency of inequality (1). The process is repeated until the step with the required accuracy is obtained. In such a way, the boundaries of the uncertainty range are determined. A similar search procedure is used for the other rate constants.

References:

[1] L.V. Kantorovich, On some new approaches to computational methods and observation processing, *Siberian Mathematical Journal*, 3 (1962), No. 5, pp. 701-709. (in Russian)

[2] A.I. Khlebnikov, On the uncertainty center method, *Journal of Analytical Chemistry*, 51 (1996), No. 3, pp. 321-322.

Software package for investigation of dynamic properties of control systems under interval uncertainty

Mamurjon Bazarov, Laziz Otakulov, Kadir Aslonov

210100, Navoi state mining institute, Navoi, Uzbekistan
mamurjon@mail.ru

Keywords: program system, interval uncertainty, parametrical identification, control systems of technological objects

Algorithmization of methods of interval analysis encounters essential difficulties caused by the fact that the existing computer hardware and software do not completely match specific requirements of interval computations. We mean specific computer interval arithmetic with directed rounding, evaluation of interval functions and some analytical transformations of the expressions.

Experts engaged in design of automatic control systems deal with mathematical models in the form of differential equations systems and/or block diagrams with transfer functions. They usually involve one or more parameters of the object and its regulator. Such mathematical models are known to be called as "parametric". In the program system INTAN-1 (INTerval ANalysis –1) developed by our team, automatic control systems under interval uncertainty of parameters can be analyzed providing that, on entry, data and constraints are considered precisely known, while the values of parameters of the automatic control systems have interval uncertainty.

The program system INTAN-1 consists, basically, of three main parts that are responsible for

- identification of the control objects with interval parameters,
- the analysis of automatic control systems under interval uncertainty,
- computing (interval) parameters of the regulator.

All the other blocks of the system are either auxiliary or utility programs. The structure of INTAN-1 can be represented in a vivid way through a block diagram.

At the start of the program system run, the block "Head program" carries out editing of the input data and checking their correctness. If an error in the input record is detected, then a corresponding diagnostic message in outputted. In case of success, further performance is launched, that is, forming input and target data, processing the current stage and analyzing the result. Then the name of the next program unit is determined, it is loaded into the memory, the system directs the control to it, and informs the user about the details of the program execution.

The block "Program toolkit" consists of the solvers that perform the main work during the solution of the problem. The short description of these blocks is given below.

The block "Auxiliary programs" includes the computational procedures for testing regularity of interval matrices, testing positive definiteness of interval matrices, etc. These are necessary for the analysis of automatic control systems under interval uncertainty.

The block "Algebraic equations solvers" includes MATLAB® implementations of interval Gauss-Seidel method, subdifferential Newton method as well as some other popular techniques for the solution of various problems that arise in connection with linear and nonlinear algebraic equations.

The basic operations used in the analysis of automatic control systems under interval uncertainty are implemented in the block "Analysis of interval automatic control systems".

The end-user, during the work with our program system, forms the initial information on the problem under solution. Then the system analyzes the information, constructs an interval model of the problem that supplements the input information, carries out analytical transformations (if necessary), performs interval expansions of the expressions, computes interval extensions of the functions, and, finally, produces a solution to the problem.

On constructing nonpolynomial spline formulas

Irina Burova

Mathematics and Mechanics Faculty, St. Petersburg State University
Universitetsky prospekt, 28, 198504, Peterhof, St. Petersburg, Russia
BurovaIG@mail.ru

Keywords: spline

Let m, l, s, n, p be integer nonnegative numbers, $l \geq 1, s \geq 1$, $p \leq s + l - 2$, $m = s + l$, $\{x_k\}$ be a mesh of ordered nodes, $a = x_0 < \ldots < x_{k-1} < x_k < x_{k+1} \ldots < x_n = b$, and the function $u \in C^m[a, b]$. We suppose that φ_j, $j = 1, \ldots, m$, is a Chebyshev system on $[a, b]$, in which case the functions $\varphi_j \in C^m[a, b]$, $j = 1, \ldots, m$, are strictly monotone and nonzero within $[a, b]$. The basic functions $\omega_j(x)$, for which supp $\omega_j = [x_{j-s}, x_{j+l}]$, $j = 1, \ldots, m$, are assumed to be valid, can be defined from the system of equations

$$\sum_{k=j-l+1}^{j+s} \omega_k(x)\varphi_i(x_k) = \sum_{k=1}^{m} c_{ik}\varphi_k(x), \quad i = 1, 2 \ldots, m,$$

where $c_{ik} = 0$ if $i \neq k$ and $c_{kk} = 1$. Next, we use $\omega_j(x)$ in the Lagrange interpolation problem or in the least squares problem. If we take $c_{ik} \neq 0$, then it is possible to construct nonpolynomial basic splines with required characteristics (e.g., smoothness).

For example, let us discuss how to construct trigonometrical basic splines of the minimal defect ($\omega_j \in C^1[a, b]$) with three mesh intervals in support.

Let supp $\omega_j = [x_{j-1}, x_{j+2}]$. If $x \in [x_j, x_{j+1}]$, then we find $\omega_j(x)$ from the following system:

$$\begin{cases} \omega_{j-1}(x) + \omega_j(x) + \omega_{j+1}(x) = 1, \\ \sin(x_{j-1})\omega_{j-1}(x) + \sin(x_j)\omega_j(x) + \sin(x_{j+1})\omega_{j+1}(x) = c_{10}\sin(x) + c_{01}\cos(x), \\ \cos(x_{j-1})\omega_{j-1}(x) + \cos(x_j)\omega_j(x) + \cos(x_{j+1})\omega_{j+1}(x) = c_{02}\sin(x) + c_{20}\cos(x). \end{cases}$$

If $[x_{j-1}, x_j]$, then we find $\omega_j(x)$ from the system

$$\begin{cases} \omega_{j-2}(x) + \omega_{j-1}(x) + \omega_j(x) = 1, \\ \sin(x_{j-2})\omega_{j-2}(x) + \sin(x_{j-1})\omega_{j-1}(x) + \sin(x_j)\omega_j(x) = c_{10}\sin(x) + c_{01}\cos(x), \\ \cos(x_{j-2})\omega_{j-2}(x) + \cos(x_{j-1})\omega_{j-1}(x) + \cos(x_j)\omega_j(x) = c_{02}\sin(x) + c_{20}\cos(x), \end{cases}$$

and if $[x_{j+1},\, x_{j+2})$, then we find $\omega_j(x)$ from the system

$$\begin{cases} \omega_j(x)+\omega_{j+1}(x)+\omega_{j+2}(x)=1, \\ \sin(x_j)\omega_j(x)+\sin(x_{j+1})\omega_{j+1}(x)+\sin(x_{j+2})\omega_{j+2}(x)=c_{10}\sin(x)+c_{01}\cos(x), \\ \cos(x_j)\omega_j(x)+\cos(x_{j+1})\omega_{j+1}(x)+\cos(x_{j+2})\omega_{j+2}(x)=c_{02}\sin(x)+c_{20}\cos(x). \end{cases}$$

We find the values of the parameters c_{01}, c_{10}, c_{02}, c_{20} from $\omega_j \in C^1(\mathbb{R}^1)$, thus obtaining $c_{02}=-c_{01}=\cos(h/2)\sin(h/2)$, $c_{10}=c_{20}=\cos^2(h/2)$, $h=(b-a)/n$. Then, on $[x_j, x_{j+1}]$, we have $\omega_{j-1}(x)=(\cos(x-jh-h)-1)/(2(\cos(h)-1))$,

$$\omega_j(x)=\frac{\cos(h)-\cos(x-jh-h/2)\cos(h/2)}{\cos(h)-1},\ \omega_{j+1}(x)=\frac{\cos(x-jh)-1}{2(\cos(h)-1)}.$$

Hence,

$$\omega_j(x)=\begin{cases} \frac{\cos(x-jh+h)-1}{2(\cos(h)-1)}, & x\in[x_{j-1},x_j), \\ \frac{\cos(h)-\cos(x-jh-h/2)\cos(h/2)}{\cos(h)-1}, & x\in[x_j,x_{j+1}), \\ \frac{\cos(x-jh-2h)-1}{2(\cos(h)-1)}, & x\in[x_{j+1},x_{j+2}]. \end{cases}$$

If $\{\varphi_j\}$ are $\{1, \sin(kx), \cos(kx)\}$, $k=1,2$ ($\omega_j \in C^2$) or $k=1,2,3$ ($\omega_j \in C^3$), then the problem is more complex, but, nevertheless, it can be easily solved using Maple$^{\text{TM}}$ [1, 2].

Suppose that we are interested in the value of a physical quantity $\tilde{u}(x)$ that is difficult or impossible to measure directly. To find the value of $\tilde{u}(x)$, several other quantities $u(x_0)+K_1(h)u'(x_0), \ldots, u(x_n)+K_1(h)u'(x_n)$, $K_1(h)=tg(h/2)$ are measured, and then we reconstruct the value of $\tilde{u}(x)\approx u(x)$:

$$\tilde{u}(x)=\sum_{k=j-1,j,j+1}\left(u(x_k)+K_1(h)u'(x_k)\right)\omega_k(x),\quad x\in[x_j,x_{j+1}].$$

We take the values $X_k=[x_k-d_k, x_k+d_k]$, $X=[xl, xp]$, $x\in X$, $U_k=[u_k-t_k, u_k+t_k]$, $U'_k=[u'_k-s_k, u'_k+s_k]$, $d_k>0$, $t_k>0$, $s_k>0$ and estimate $\tilde{u}(X)$ using interval computations ([3]).

References:

[1] I.G. Burova, About trigonometric splines construction. *Vestnik St. Petersburg University: Mathematics. Series I*, 2 (2004), pp. 7–15.

[2] I.G. Burova, T.O. Evdokimova, About smooth trigonometric splines of the third order, *Vestnik St. Petersburg University: Mathematics. Series I*, 4 (2004), pp. 12–23.

[3] G. Alefeld, J. Herzberger, *Introduction to Interval Computations*, Academic press, Tokyo, Toronto, 1983.

On the OLS set in linear regression with interval data

Michal Černý and Miroslav Rada

Department of Econometrics, University of Economics, Prague
Winston Churchill Square 4
13067 Prague, Czech Republic
cernym@vse.cz, miroslav.rada@vse.cz

Keywords: possibilistic interval regression, OLS set, zonotope

Consider the linear regression model $y = X\beta + \varepsilon$, where y denotes the vector of (observations of) output data, β denotes the vector of regression parameters, X denotes the matrix of (observations of) input data, which is assumed to have full column rank, and ε denotes the vector of disturbances. Assume that y, the vector of observations of the output variable, ranges over an interval vector $\boldsymbol{y}$. Using well-known ordinary least squares (OLS) estimator, we define the *OLS set* as $\{(X'X)^{-1}X'y : y \in \boldsymbol{y}\}$. The OLS set consists of all OLS-estimates of regression parameters of the regression model as the vector of observations ranges over $\boldsymbol{y}$.

For a user of the regression model it is essential to have a suitable description of the OLS set.

The OLS set is a zonotope in the parameter space. We present a method for construction of vertex description of the OLS set, inequality description of the OLS set and computation of volume of the OLS set. The method, called "Reduction-and-Reconstruction-Recursion", is a uniform approach to the three problems. While it runs in exponential time in the general case (which is not surprising as the computation of volume of a zonotope is a $\#\boldsymbol{P}$-hard problem*), in a fixed dimension (= number of regression parameters) it is a polynomial-time method. We further discuss complexity-theoretic properties of the method

*Unlike the class $\boldsymbol{NP}$ of the *decision* problems, the problem class $\#\boldsymbol{P}$ contains the *function* or precisely *counting* problems associated with problems in $\boldsymbol{NP}$.

in the general case and compare it with other known methods for enumeration of facets, enumeration of vertices and computation of volume of a zonotope.

In general, the OLS set is a polytope, which is complex from the combinatorial point of view (i.e., with respect to the number of facets and vertices). Hence it makes sense to seek for reasonably simple approximations. Using interval arithmetic, construction of the interval enclosure is trivial. We show a method for finding an ellipsoidal approximation of the Löwner-John type. We present an adaptation of Goffin's Algorithm (a version of the shallow-cut ellipsoid method) for construction of an ellipsoidal enclosure. In particular, given $\epsilon > 0$ fixed, in polynomial time we construct an ellipse $\mathcal{E}(E, s)$ such that $\mathcal{E}(d^{-2} \cdot E, s) \subseteq OLS \subseteq \mathcal{E}((1+\epsilon)E, s)$, where $\mathcal{E}(E, s)$ is the ellipse $\{x : (x-s)'E^{-1}(x-s) \leq 1\}$ with E is positive definite, OLS is the OLS set and d is dimension (number of regression parameters).

References:

[1] D. Avis, K. Fukuda, Reverse search for enumeration, *Discrete Applied Mathematics*, 65 (1996) 21–46.

[2] J. L. Goffin, Variable metric relaxation methods. Part II: The ellipsoid method, *Mathematical Programming*, 30 (1984) 147–162.

[3] M. Grötschel, L. Lovász, A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer, Germany, 1993.

[4] L. J. Guibas, A. Nguyen, L. Zhang, Zonotopes as Bounding Volumes, *Proceeding SODA '03 Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms,* SIAM, Pennsylvania, 2003.

[5] H. Kashima, K. Yamasaki, A. Inokuchi, H. Saigo, Regression with interval output values, *19th International Conference on Pattern Recognition ICPR 2008,* Tampa, USA, 2008, pp. 1–4.

[6] S. Schön, H. Kutterer, Using zonotopes for overestimation-free interval least-squares — some geodetic applications, *Reliable Computing*, 11 (2005), 137–155.

[7] H. Tanaka, J. Watada, Possibilistic linear systems and their application to the linear regression model, *Fuzzy Sets and Systems*, 27 (1988) 275–289.

[8] G. Ziegler, *Lectures on Polytopes*, Springer, Germany, 2004.

A statistical inference model for the dynamic range of LTI systems

Alexandre Chapoutot, Laurent-Stéphane Didier and Fanny Villers

ENSTA ParisTech – 32 bd Victor, 75739 Paris Cedex 15, France
UPMC-LIP6 – 4, place Jussieu, 75252 Paris Cedex 05, France
UPMC-LPMA – 4, place Jussieu, 75252 Paris Cedex 05, France
alexandre.chapoutot@ensta-paristech.fr
laurent-stephane.didier@upmc.fr
fanny.villers@upmc.fr

Keywords: range estimation, linear time invariant filters, extreme value theory, time series

Control-command and signal filtering algorithms are two main components of embedded software. These algorithms are usually described by linear time-invariant (LTI) systems which have good properties and are well understood mathematically. In automotive domain, in order to increase performance of the implementation of such algorithms, e.g., to reduce execution time or memory consumption, the use of fixed-point arithmetic is almost unavoidable. Nevertheless at the design level, these algorithms are studied and defined using floating-point arithmetic. As the two arithmetics have very different behaviors, we need tools to transform with strong guaranties floating-point programs into numerically equivalent programs using fixed-point arithmetic. This conversion requires two steps. The range estimation deals with the integer part of the targeted fixed point representation while the accuracy estimation allows to define the fractional part. In this work we are considering range estimation methods.

The range estimation of LTI systems is an important research field in which two kinds of methods exist. The *static methods* based on interval [5] or affine [6] arithmetics and the *dynamic methods* based on statistical tools. This work is focused on the second kind of methods. In both cases, the first step in the fixed-point conversion is the computation of the dynamic range of each variable in the program which is a mandatory information to determine the fixed-point format. A few statistical models exist for this task, e.g., the previous work [1,3,4]. In particular, the *Generalized Extreme Value (GEV) Distribution* [2], used in [4] and in a restricted form in [3], seems very promising as it can be used to infer minimal and maximal values of each variable in function of a user parameter. It defines the probability that these values may be exceeded during the execution of the program.

The use of the GEV distribution shows good results in practice, especially for LTI systems. In this approach, several simulations of the studied systems are performed using random input. For each simulation the maximum is kept. Because dealing with minimum is similar, we focus our study on the maxima. They appear to belong to a GEV distribution. However, in this model, it is not taken into account that each simulation is producing a time series. In this work we show that data produced by LTI systems can be modelized trough an *autoregressive model* (AR). This property can be used in order to show that the distribution of inner variables maxima of LTI systems is a GEV distribution.

References:

[1] W. Sung and K.-I. Kum, Simulation-base word-length optimization method for fixed-point digital signal processing systems, *IEEE Transactions on Signal Processing*, 43 (1995), No. 12, pp. 3087–3090.

[2] S. Coles, *An Introduction to Statistical Modeling of Extreme Values*, Springer, 2001.

[3] E. Özer, A. P. Nisbet, and D. Gregg, A stochastic bitwidth estimation technique for compact and low-power custom processors, *ACM Transaction on Embedded Computing Systems*, 7 (2008), No. 3, pp. 1–30.

[4] A. Chapoutot, L.-S. Didier, and F. Villers, Range estimation of floating-point variable in Simulink models, In *Numerical Software Verification (NSV-II)*, 2009.

[5] R.E. Moore, R.B. Kearfott, M.J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[6] L. de Figueiredo and J. Stolfi, Self-validated numerical methods and applications, In *Brazilian Mathematics Colloquium Monograph*, 1997.

Interval-based robustness of linear parameterized filters

Alexandre Chapoutot and Thibault Hilaire and Philippe Chevrel

ENSTA ParisTech – 32 bd Victor, 75739 Paris Cedex 15, France
LIP6 – 4, place Jussieu, 75252 Paris Cedex 05, France
IRCCyN – 1, rue de la Noë, BP 92 101, 44321 Nantes Cedex 3, France
alexandre.chapoutot@ensta-paristech.fr
thibault.hilaire@lip6.fr
philippe.chevrel@mines-nantes.fr

Keywords: linear filters, interval arithmetic, sensitivity analysis

Introduction. This article deals with the resilient implementation of parametrized linear filters (or controllers), *i.e.* with realizations that are robust with respect to their fixed-point implementation.
The implementation of a linear filter/controller in an embedded device is a difficult task due to numerical deteriorations in performances and characteristics. These degradations come from the quantization of the embedded coefficients and the roundoff occurring during the computations.

As mentioned in [1], there are an infinity of equivalent possible algorithms to implement a given transfer function h. To cite a few of them, one can use direct forms, state-space realizations, ρ-realizations, etc. Although they do not require the same amount of computation, all these realizations are equivalent in infinite precision, but they are no more in finite precision. The *optimal realization problem* is then to find, for a given filter, the most resilient realization.

We here consider an extended problem with filters those coefficients depend on a set $\boldsymbol{\theta}$ of parameters that are not exactly known during the design. They are used for example in automotive control, where a very late fine tuning is required.

Linear parametrized filters. Following [3], we denote $\boldsymbol{Z}(\boldsymbol{\theta})$ the matrix containing all the coefficients used by the realization, $h_{\boldsymbol{Z}(\boldsymbol{\theta})}$ the associated transfer function and $\boldsymbol{\theta}^\dagger$ the quantized version of $\boldsymbol{\theta}$. $\boldsymbol{Z}^\dagger(\boldsymbol{\theta}^\dagger)$ is then the set of the quantized coefficients, *i.e.* the quantization of coefficients $\boldsymbol{Z}(\boldsymbol{\theta}^\dagger)$ computed from the quantized parameters $\boldsymbol{\theta}^\dagger$. The corresponding transfer function is denoted $h_{\boldsymbol{Z}^\dagger(\boldsymbol{\theta}^\dagger)}$.

Performance Degradation Analysis. The two main objectives of this article are to evaluate the impact of the quantization of $\boldsymbol{\theta}$ and $\boldsymbol{Z}(\boldsymbol{\theta})$ on the filter performance and to estimate the parameters $\boldsymbol{\theta}$ that give the worst transfer function error in the set of possible parameters $\boldsymbol{\Theta}$.

For that purpose, there are mainly two kinds of tools to study the degradation of filter performance due to the quantization effect: *i)* use a sensitivity measure (with respect to the coefficients) based on a first order approximation and a statistical quantification error model; *ii)* use interval tool, based on transfer function with interval coefficients. In both cases, we seek the maximal distance between the exact transfer function $h_{\boldsymbol{Z}(\boldsymbol{\theta})}$ and the quantized one $h_{\boldsymbol{Z}^\dagger(\boldsymbol{\theta}^\dagger)}$. For that purpose, we can use the L_2-*norm i.e.*, $\| g \|_2 \triangleq \sqrt{\frac{1}{2\pi}\int_0^{2\pi} | g(e^{j\omega}) |^2 \, d\omega}$ or the *Maximum norm i.e.*, $\| g \|_\infty \triangleq \max_{\omega\in[0,2\pi]} | g(e^{j\omega}) |$.

The measure of the degradation of the finite precision implementation is then given by $\| h_{\boldsymbol{Z}(\boldsymbol{\theta})} - h_{\boldsymbol{Z}^\dagger(\boldsymbol{\theta}^\dagger)} \|_\diamond$, with $\diamond \in \{2, \infty\}$. So the worst-case parameters $\boldsymbol{\theta}_0$ can be found by solving:

$$\arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \| h_{\boldsymbol{Z}(\boldsymbol{\theta})} - h_{\boldsymbol{Z}^\dagger(\boldsymbol{\theta}^\dagger)} \|_\diamond \quad . \tag{2}$$

Since $\boldsymbol{\Theta}$ is an interval vector, we denote $[h]$ the interval transfer function. With an interval approach, we can define the following constrained global optimization problem:

$$\text{Maximize} \| [h]^\dagger_{\boldsymbol{Z}^\dagger(\boldsymbol{\theta}^\dagger)} - [h]_{\boldsymbol{Z}(\boldsymbol{\theta})} \|_\diamond \quad \text{subject to} \quad \boldsymbol{\theta} \in \boldsymbol{\Theta} \ . \tag{3}$$

Note that in both cases, the evaluation of the norms can be done in interval with $\omega \in [0, 2\pi]$.

We will present the solutions of this problem using interval optimization methods [2] and we will compare them with the statistical sensitivity approach.

References:

[1] H. Hanselmann, Implementation of digital controllers – a survey, *Automatica*, 23 (1987), No. 1, pp. 7–32.

[2] E.R. Hansen and G.W. Walster, *Global Optimization Using Interval Analysis*, Pure and Applied Mathematics, Marcel Dekker, 2004.

[3] T. Hilaire, P. Chevrel, and J.F. Whidborne, A unifying framework for finite wordlength realizations, *IEEE Trans. on Circuits and Systems*, 54 (2007), No. 8, pp. 1765–1774.

Acceleration of the computational convergence of extended interval Newton method for a special class of functions

Chin-Yun Chen

Department of Applied Mathematics, NCYU,
No 300, University Rd,
Chiayi 600, TAIWAN
cychen09@gmail.com

Keywords: interval arithmetic, enclosure methods, verified bounds, extended interval Newton method, monosplines, Peano kernels

Interval Newton method (cf. [1,2,3]) is a practical tool for enclosing a unique simple zero $x^* \in [x]$ of a smooth function $f \in C^1[x]$ in an interval domain $[x]$ such that the width of the enclosure $[x^*]$ satisfies a given error tolerance. In this case, the interval Newton method has a quadratic convergence.

In case of existing more zeros or a multiple zero in the given domain $[x]$, interval Newton method can be extended to enclose all the zeros according to a requested resolution (cf. [2,3]). The extension is based on the extended interval division, namely, the division by an interval that contains 0. The extended interval Newton method (XIN) has a linear convergence and its performance depends on the chosen definition for the underlying interval division, cf. [4]. An effective algorithm for XIN that is based on the precise quotient set [5] is suggested in [4]. It has superior effectiveness when the midpoint of $[x]$ happens to be a zero of f and f is not too flat in a neighborhood of the midpoint $\mathrm{mid}([x])$. If $f(\mathrm{mid}([x])) = 0$ and f is flat in a neighborhood of $\mathrm{mid}([x])$ then the algorithm in [4] could be superior in efficiency. In the other cases, it is comparable to the algorithms for XIN that are based on the supersets of the precise quotient set.

One problem of the zero-finding by XIN is that there could be a cluster of redundant intervals produced for a multiple zero, where those intervals could be adjacent or nearby disjoint, which depends on the chosen algorithm for XIN, cf. [4]. For a pure zero-finding task, the situation of redundancy could be detected by extra inspection or attention. However, if the information of the zeros is to be used for further automatic computation, the redundant intervals

can lead to unsatisfactory numerical results. To overcome this problem, extra attention to the properties of the function f should be paid.

This work uses the algorithm in [4] to find all the zeros of Peano monosplines. By Peano monosplines, we mean the Peano kernels regarding the quadrature rules that are constructed for proper (Riemann-)integrals. They generally possess more than one multiple zero in their domains; moreover, their zeros are generally required for deriving reliable bounds of Peano error constants. In this work, the properties of Peano monosplines as well as the computational techniques that are useful for the performance of XIN are discussed. Numerical results are then given for Peano monosplines regarding different quadrature rules to demonstrate the improvements in the computational convergence of XIN.

References:

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

[2] U.W. Kulisch, *Computer Arithmetic and Validity — Theory, Implementation, and Applications*, de Gruyter, Berlin, 2008.

[3] R.E. Moore, R.B. Kearfott, M.J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[4] C.-Y. Chen, Extended interval Newton method based on the precise quotient set, *Computing*, 92 (2011), No. 4, pp. 297-315.

[5] U.W. Kulisch, Arithmetic operations for floating-point intervals, as Motion 5 accepted by the IEEE Standards Committee P1788 as definition of the interval operations (2009), see [6].

[6] J. Pryce (ed), P1788: IEEE standard for interval arithmetic version 02.2, http://grouper.ieee.org/groups/1788/email/pdfOWdtH2mOd9.pdf (2010)

[7] C.-Y. Chen, A performance comparison of the zero-finding by extended interval Newton method for Peano monosplines, preprint.

Numerical comparison of some verified approaches for approximate integration

Chin-Yun Chen

Department of Applied Mathematics, NCYU,
No 300, University Rd,
Chiayi 600, TAIWAN
cychen09@gmail.com

Keywords: interval arithmetic, enclosure methods, verified bounds, numerical integration

Approximate computation of the definite integral $I(f) = \int_B f(\vec{x})\, d\vec{x}$ over an n-dimensional interval $B \in I\mathbb{R}^n$, $n \in \mathbb{N}$, is an essential task in different fields of science and engineering. Traditional approaches for the numerical integration $I(f) \approx S(f) = \sum_{i=1}^{N} w_i f(\vec{x}_i)$ generally use the null rules for error estimation, i.e. $E(f) := I(f) - S(f) \approx \hat{S}(f) - S(f)$, where $S(\cdot)$ and $\hat{S}(\cdot)$ are integration rules with $\deg \hat{S} > \deg S$ and $\deg S := \max\{n \in \mathbb{N} \,|\, S(x^n) = I(x^n)\}$. Different from the traditional approaches, verified integrators enclose the discretization error $E(f)$ reliably; more precisely, $I(f) \in [I(f)] = [S(f)] + [E(f)]$. Due to this difference, the numerical integrators that are based on interval arithmetic can be superior in efficiency to the conventional approaches, especially when oscillating integrands are considered, cf. [1,2]. Moreover, verified integrators also can be relatively effective when conventional integrators encounter difficulties, cf. [1,2].

In the literature, there are different verified approaches discussed for the numerical integration $I(f) \approx I(p)$ or $I(f) \approx S(f)$, where p is an interpolation polynomial of f. The approximation $I(f) \approx I(p)$ is considered for example by the Taylor model method in [3], where p is a Taylor polynomial of f. Those verified approaches differ in the approximation rules, the ways of error estimation, and/or the adaptive strategies. All their efforts mainly focus on reducing the width of error enclosures. The methods of error estimation that are considered in verified integrators include the derivative free method for analytic functions (cf. [1,4,5]), the Taylor model method by one or more higher (partial) derivatives of a fixed order (cf. [3]), the classical error bounds of the highest orders (cf. [6]), and the adaptive error estimation by the Peano-Sard kernel method (cf. [2,7,8]). The importance of the Peano-Sard kernel method is that it supplies multiple error estimates for each integration rule of a higher degree, which can be realized by interval arithmetic for sufficiently smooth integrands.

It is known that the classical error bounds of the highest orders, depending on the functional behavior of the integrands, are not always practical for error estimation, cf. [7,8,9]. This work gives numerical comparison of some verified approaches for approximate integration that do error estimation by the Taylor model method in [3], the derivative free method in [4] and the Peano-Sard kernel method in [2,8].

References:

[1] K. Petras, Self-validating integration and approximation of piecewise analytic functions, *J. Comput. Appl. Math.*, 145 (2002), pp. 345–359.

[2] C.-Y. Chen, Bivariate product cubature using Peano kernels for local error estimates, *J. Sci. Comput.*, 36 (2008), No. 1, pp. 69–88.

[3] M. Berz, K. Makino, New methods for high-dimensional verified quadrature, *Reliable Computing*, 5 (1999), pp. 13–22.

[4] K. Petras, Principles of verified numerical integration, *J. Comput. Appl. Math.*, 199 (2004), pp. 317–328.

[5] M. C. Eiermann, Automatic, guaranteed integration of analytic functions, *BIT*, 29 (1989), pp. 270–282.

[6] U. Storck, An adaptive numerical integration algorithm with automatic result verification for definite integrals, *Computing*, 65 (2000), pp. 271–280.

[7] C.-Y. Chen, Computing interval enclosures for definite integrals by application of triple adaptive strategies, *Computing*, 78 (2006), No. 1, pp. 81–99.

[8] C.-Y. Chen, On the properties of Sard kernels and multiple error estimates for bounded linear functionals of bivariate functions with application to non-product cubature, *Numer. Math.*, accepted.

[9] C.-Y. Chen, Verified computed Peano constants and applications in numerical quadrature, *BIT*, 47 (2007), No. 2, pp. 297–312.

Numerical probabilistic analysis under aleatory and epistemic uncertainty

Boris S. Dobronets and Olga A. Popova

Siberian Federal University
79, Svobodny Prospect
660041 Krasnoyarsk, Russia
BDobronets@sfu-kras.ru, olgaarc@yandex.ru

Keywords: numerical probabilistic analysis, epistemic uncertainty, second order histogram

Many important practical problems involve different uncertainty types. In this paper, we consider Numerical Probabilistic Analysis (NPA) for problems under so-called *epistemic uncertainty* that characterizes a lack of knowledge about a considered value. Generally, epistemic uncertainty may be inadequate to "frequency interpretation", typical for classical probability and for uncertainty description in traditional probability theory. Instead, epistemic uncertainty can be specified by a "degree of belief". Alternative terms to denote epistemic uncertainty are "state of knowledge uncertainty", "subjective uncertainty", "irreducible uncertainty". Sometimes, processing epistemic uncertainty may require the use of special methods [1].

In our work, we develop a technique that uses Numerical Probabilistic Analysis for decision making under epistemic uncertainty of probabilistic nature. One more application of Numerical Probabilistic Analysis is to solve various problems with stochastic data uncertainty.

The basis of NPA is numerical operations on probability density functions of the random values. These are operations "$+$", "$-$", "$\cdot$", "$/$", "$\uparrow$", "max", "min", as well as binary relations "$\leq$", "$\geq$" and some others. The numerical operations of the histogram arithmetic constitute the major component of NPA. It is worthwile to note that the idea of numerical histogram arithmetic has been first implemented in the work [2].

Notice that the density function can be a discrete function, a histogram (piecewise constant function), and a piecewise-polynomial function.

Next, we consider the concepts of natural, probabilistic and histogram extensions of function. We outline the numerical algorithms for constructing such extension for some classes of function [3].

Using the arithmetic of probability density functions and probabilistic extensions, we can construct numerical methods that enable us solving systems of linear and nonlinear algebraic equations with stochastic parameter [4].

To facilitate more detailed description of the epistemic uncertainty, we introduce the concept of *second order histograms*, which are defined as piecewise histogram functions [5]. The second order histograms can be constructed using experience and intuition of experts.

Relying on specific practical examples, we show that the use of the second order histograms may prove very helpful in decision making. In particular, we consider risk assessment of investment projects, where histograms of factors such as Net Present Value (NPV) and Internal Rate of Return (IRR) are computed.

References:

[1] L.P. Swiler, A.A. Giunta, Aleatory and epistemic uncertainty quantification for engineering applications, *Sandia Technical Report*, SAND2007-2670C.

[2] V.A. Gerasimov, B.S. Dobronets, and M.Yu. Shustrov, Numerical operations of histogram arithmetic and their applications, *Automation and Remote Control*, 52 (1991), No. 2, pp. 208–212.

[3] B.S. Dobronets, O.A. Popova, Numerical probabilistic analysis and probabilistic extension, *Proceedings of the XV International EM2011 Conference*, O. Vorobyev, ed., SFU, RIFS, Krasnoyarsk, 2011, pp. 67–69.

[4] B.S. Dobronets, O.A. Popova, Numerical operations on random variables and their application, *Journal of Siberian Federal University. Mathematics & Physics*, 4 (2011), No. 2, pp. 229–239.

[5] B.S. Dobronets, O.A. Popova, Histogram time series, *Proceedings of the X International FAMES2011 Conference*, O. Vorobyev, ed., RIFS, SFU, KSTEI, Krasnoyarsk, 2011, pp. 127–130.

Model predictive control of discrete linear systems with interval and stochastic uncertainties

Vladimir V. Dombrovskii and Elena V. Chausova

Tomsk State University
36, Lenin ave.
634050 Tomsk, Russia
dombrovs@ef.tsu.ru, chau@ef.tsu.ru

Keywords: linear dynamic system, interval uncertainty, stochastic uncertainty, model predictive control, convex optimization, linear matrix inequalities

The work examines the problem of model predictive control for an uncertain system containing both interval and stochastic uncertainties. We consider a linear dynamic system described by the following equation:

$$x(k+1) = \left(A_0(k) + \sum_{j=1}^{n} A_j(k) w_j(k)\right) x(k) + \left(B_0(k) + \sum_{j=1}^{n} B_j(k) w_j(k)\right) u(k), \quad k = 0, 1, 2, \ldots. \quad (1)$$

Here, $x(k) \in \mathbb{R}^{n_x}$ is the state of the system at time k ($x(0)$ are assumed to be available); $u(k) \in \mathbb{R}^{n_u}$ is the control input at time k; $w_j(k), j = 1, \ldots, n$, are independent white noises with zero mean and unit variance; $A_j(k) \in \mathbb{R}^{n_x \times n_x}$, $B_j(k) \in \mathbb{R}^{n_x \times n_u}, j = 0, \ldots, n$, are the state-space matrices of the system.

The elements of the state-space matrices are known not exactly, and we have only the intervals of their possible values:

$$A_j(k) \in \boldsymbol{A}_j, \quad B_j(k) \in \boldsymbol{B}_j, \quad j = 0, \ldots, n, \quad k \geq 0, \quad (2)$$

where $\boldsymbol{A}_j \in \mathbb{IR}^{n_x \times n_x}, \boldsymbol{B}_j \in \mathbb{IR}^{n_x \times n_u}, j = 0, \ldots, n$; $\mathbb{IR}$ is the set of the real intervals $\boldsymbol{x} = [\underline{x}, \overline{x}]$, $\underline{x} \leq \overline{x}$, $\underline{x}, \overline{x} \in \mathbb{R}$.

Model predictive control [1] involves the on-line solution of an optimization problem to determine, at each time instant, a fixed number of optimal future control inputs. Although more than one control move is calculated only the first one is implemented. At the next sampling time, the state of the system is measured, and the optimization is repeated.

Allowing for two uncertainty types (interval and stochastic) present in the system (1), we consider the following performance objective:

$$\min_{u(k+i|k),\, i=0,\ldots,m-1,} \quad \max_{\substack{A_j(k+i)\in \boldsymbol{A}_j,\, B_j(k+i)\in \boldsymbol{B}_j,\\ j=0,\ldots,n,\, i\geq 0,}} J(k),$$

where

$$J(k) = \mathsf{E}\left\{ \sum_{i=0}^{\infty} \left(x(k+i|k)^T Q x(k+i|k) + u(k+i|k)^T R u(k+i|k) \right) \,\middle|\, x(k) \right\}.$$

$\mathsf{E}\{\cdot|\cdot\}$ denotes the conditional expectation; $Q \in \mathbb{R}^{n_x \times n_x}, Q = Q^T \geq 0$, $R \in \mathbb{R}^{n_u \times n_u}, R = R^T > 0$, are given symmetric weighting matrices; $u(k+i|k)$ is the predictive control at time $k+i$ computed at time k, and $u(k|k)$ is the control move implemented at time k; $x(k+i|k)$ is the state of the system at time $k+i$ derived at time k by applying the sequence of predictive controls $u(k|k), u(k+1|k), \ldots, u(k+i-1|k)$ on the system (1), and $x(k|k)$ is the state of the system measured at time k; m is the number of control moves to be computed, $u(k+i|k) = 0$ for all $i \geq$; $A > 0$ $(A \geq 0)$ means that A is a positive definite (semi-definite) matrix.

We compute the optimal control according to the linear state-feedback law:

$$u(k+i|k) = F(k)x(k+i|k), \quad i \geq 0, \tag{3}$$

where $F(k) \in \mathbb{R}^{n_u \times n_x}$ is the state-feedback matrix at time k.

We solve the above problem by using linear matrix inequalities [2], as this has been done in [1]. At each time instant k, we solve an eigenvalue problem in order to calculate the state-feedback matrix $F(k)$ in the control law (3) which minimizes the upper bound on $J(k)$. As a result, we get the optimal robust control strategy providing the system with stability in the mean-square sense.

References:

[1] M.V. Kothare, V. Balakrishnan, M. Morari, Robust constrained model predictive control using linear matrix inequality, *Automatica*, Vol. 32 (1996), No. 10, pp. 1361–1379.

[2] S. Boyd, L. Ghaoui, E. Feron, V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, 1994. (Studies in Applied Mathematics, vol. 15)

Numerical verification and experimental validation of sliding mode control design for uncertain thermal SOFC models

Thomas Dötschel[1], Andreas Rauh[1], Ekaterina Auer[2], and Harald Aschemann[1]

[1]Chair of Mechatronics, University of Rostock
D-18059 Rostock, Germany
{Thomas.Doetschel,Andreas.Rauh,Harald.Aschemann}@uni-rostock.de

[2]Faculty of Engineering, INKO, University of Duisburg-Essen
D-47048 Duisburg, Germany
Auer@inf.uni-due.de

Keywords: interval-based sliding mode control, numerical verification, experimental validation, real-time implementation

The dynamics of high-temperature solid oxide fuel cell (SOFC) systems can be mainly described by their thermal, fluidic, and electro-chemical behavior. In modeling for control purposes, it is essential to focus especially on the thermal subsystem which represents the most dominant system part. The admissibility of control strategies for SOFCs is usually characterized by limitations on the maximum fuel cell temperature and on the spatial and temporal variation rates of the internal stack temperature distribution. These constraints are introduced to minimize mechanical strain due to different thermal expansion coefficients of the stack materials and to reduce degradation phenomena of the cell materials.

Control-oriented models for the thermal behavior of SOFC systems are given by ordinary differential equations (ODEs). They can be derived from the first law of thermodynamics for nonstationary processes and represent integral balances of the inflow and outflow of energy, which determine the internal energy. In addition, the internal energy can be directly linked to the temperature of the stack module. The preceding fundamental modeling procedure can be modified to account for the spatial temperature distribution in the interior of a stack module by means of a finite volume semi-discretization. The corresponding nonlinear system models describe, firstly, the transient behavior during the heating phase, secondly, the influence of variable electrical loads during usual system

operation, and, finally, the transient cooling process during the shutdown phase of the system.

The parameter intervals and non-verified parameter estimates that have been identified by the procedures presented in [2] provide the basis for the design of robust controllers. To obtain such a controller, we use an extension of classical sliding mode control making use of a suitable Lyapunov function to stabilize the system dynamics despite possible bounded uncertainty in the system parameterization and a-priori unknown disturbances. A first simulation study was published in [1].

In this contribution, we extend our considerations in such a way that the enthalpy flow of the cathode gas into the stack module is defined as a control input for the thermal behavior. This enthalpy flow can be influenced by manipulating the air mass flow as well as the temperature difference between the supplied air in the preheating unit and the inlet elements of the fuel cell stack module. If the above-mentioned sliding mode control procedure is employed to determine the enthalpy flow, further physical restrictions have to be accounted for. These restrictions result from the admissible operating ranges of both the valve for the air mass flow and the temperature of the preheating unit. Moreover, the variation rate of the temperature difference between the preheating unit and the stack module's inlet elements has to be restricted to prevent damages due to thermal stress. These feasibility constraints are taken into account using an appropriate cost function which is evaluated along with the design criteria for the guaranteed stabilizing interval-based sliding mode controller.

Employing the results for the interval-based verified parameter identification, we present both numerical simulations and experimental results, the latter validating the control procedures for the SOFC test rig which is available at the Chair of Mechatronics at the University of Rostock.

References:

[1] T. Dötschel, A. Rauh, and H. Aschemann, Reliable Control and Disturbance Rejection for the Thermal Behavior of Solid Oxide Fuel Cell Systems, *Proc. of Vienna Conference on Mathematical Modelling*, Vienna, Austria, 2012, `http://www.IFAC-PapersOnLine.net` (accepted).

[2] A. Rauh, T. Dötschel, E. Auer, and H. Aschemann, Interval Methods for Control-Oriented Modeling of the Thermal Behavior of High-Temperature Fuel Cell Stacks, *Proc. of 16th IFAC Symposium on System Identification SysID 2012*, Brussels, Belgium, 2012.

Limitations of complex interval Gauss-Seidel iterations

Vadim S. Dronov

Altai State University
61, Lenin str.
656049 Barnaul, Russia
planeswalker@rambler.ru

Keywords: interval analysis, verified computing

A necessity to solve computational problems in case of incomplete and uncertain input data has been one of the main reasons for emerging the interval analysis. Nowadays, interval methods are well-developed for the data described by real intervals. However, some practical problems bring to life models with the similar type of uncertainty (bounded within some intervals), with the data being complex. Good examples are meso-mechanic algorithms in physics [1], estimation of dynamic functions (like heat transfer function in [2]), and so on. Consequently, the computation methods for complex-valued models is a major issue.

In this work, we are trying to generalize Gauss-Seidel iteration method from real intervals to complex intervals and show their possible limitations.

Our basic interval object is a set of circular complex intervals $\langle c, r\rangle = \{\, x \in \mathbb{C} : |x - c| \leq r \,\}$. There are no "standard definition" for a complex interval, and different tasks require different basic objects. Basic system is the system of linear equations $\boldsymbol{A}x = \boldsymbol{b}$, where $\boldsymbol{A}$ is an interval $n \times n$-matrix, $\boldsymbol{b}$ is an interval vector.

We work with one kind of solutions sets, i.e. the so-called united solutions set:

$$\Xi_{uni}(\boldsymbol{A}, \boldsymbol{b}) = \{\, x \in \mathbb{C}^n \mid (\exists A \in \boldsymbol{A})(\exists b \in \boldsymbol{b})(Ax = b)\},$$

Statement. Classic Gauss-Seidel iteration method can be generalized for the complex case with minimal problems. (This require replacement of real interval operations by complex ones, and replacing the exact intersection of circular intervals by hull of them only).

Theorem 1. Complex analogues of Gauss-Seidel iterations method still function, i.e. do not deteriorate outer estimation of solutions set at any step and still produce outer estimate of solution set as a result.

The efficiency of interval Gauss-Seidel iterations in the real case is limited by Neumaier theorem [3], which states that the method works only with the so-called interval H-matrices. The complex case also has limitations, which we formulate below.

In the sequel, an interval $n \times n$-matrix will be called *circular trace dominant matrix* (CTD-matrix), if, for any n-dimensional non-zero interval vector $\boldsymbol{u}$ with $\operatorname{mid} \boldsymbol{u} = 0$, the condition

$$\left| \sum_{i \neq j} \boldsymbol{a}_{ij} \boldsymbol{u}_j \right| < |\boldsymbol{a}_{ii} \boldsymbol{u}_i|$$

is true for every i.

Theorem 2. If, in the system of equations $\boldsymbol{A}x = 0$, the matrix A is not an CTD-matrix, then there exists a starting interval $\boldsymbol{x}$ of any width that cannot be improved by Gauss-Seidel iterations.

Definition. We call the interval $n \times n$-matrix $\boldsymbol{A}$ *strongly different* from CTD-matrices, with difference coefficient τ, if a vector $\boldsymbol{U}$, $\operatorname{mid} \boldsymbol{U} = 0$, exists for which $|\sum_{i \neq j} \boldsymbol{a}_{ij} \boldsymbol{U}_j| > \tau |\boldsymbol{a}_{ii} \boldsymbol{U}_i|$.

Statement. If, in the system $\boldsymbol{A}x = \boldsymbol{b}$, the matrix $\boldsymbol{A}$ is strongly different from CTD-matrix, and the difference coefficient is large enough, there exists a starting interval approximation of any width that are "improvement-resistant" for Gauss-Seidel iterations.

The main difference with the real case, however, is the following theorem, that substantially narrows the applicability of the Gauss-Seidel iterations in case of complex intervals:

Theorem 3. The class of CTD-matrices is empty.

References:

[1] O. Dessombz, F. Thouverez, J.-P. Laine, L. Jezequel, Analysis of mechanical systems using interval computations applied to finite elements methods, *Journal of Sound and Vibration*, 2001, No. 5, pp. 949–968.

[2] Y. Candau, T. Raissi, N. Ramdani, L. Ibos, Complex interval arithmetic using polar form, *Reliable Computing*, 12 (2006), No. 1, pp. 1–20.

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

Endpoint and midpoint interval representations – theoretical and computational comparison*

Tomáš Dzetkulič

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8, Czech Republic
dzetkulic@cs.cas.cz

Keywords: interval arithmetic, interval format, high precision

In classical interval analysis [3], a real value x is in a digital computer represented by an interval $x \in [x_{lo}, x_{hi}]$ where x_{lo} and x_{hi} are two floating point numbers. There are further possible representations of the value of x using two or three floating point numbers:

- $x \in [x_{mid} - e, x_{mid} + e]$ using two floating point numbers x_{mid} and e,
- $x \in [x_{mid} + e_{lo}, x_{mid} + e_{hi}]$ using three floating point numbers x_{mid}, e_{lo} and e_{hi}.

To motivate our work, let us consider an example where $x = 1/15$. Using the classical interval format, the tightest possible interval that contains x using standard double precision floating point format [2] is

$$[6.6666666666666657415 \times 10^{-2}, 6.6666666666666796193 \times 10^{-2}].$$

The width of this interval is approximately 1.387779×10^{-17}. Using the alternative representation with $x_{mid} \doteq 6.6666666666666657415 \times 10^{-2}$, we can obtain $e = e_{hi} \doteq 9.252 \times 10^{-19}; e_{lo} = 0$. With either one of the alternative representations, we achieve an order of magnitude tighter enclosure of the actual value of x. Moreover, if we allow midpoint to lie outside the interval, we can set e_{lo} close to e_{hi} to get an interval width less than 10^{-30} using just three floating point numbers.

*This work was supported by Czech Science Foundation grant 201/09/H057, Ministry of Education, Youth and Sports project number OC10048 and long-term financing of the Institute of Computer Science (RVO 67985807). The author would like to thank Stefan Ratschan for a valuable discussion and helpful advice.

For midpoint intervals, the optimal error can be estimated based on the work of Dekker [1]. Intervals of the form $[x_{mid} - e, x_{mid} + e]$ with such optimal error estimation were used in [5] but no theoretical comparison with classical interval analysis was given. On the other hand, theoretical comparison in [4] is based on suboptimal error estimation. In our work we compare midpoint and endpoint intervals using the optimal error estimation. Moreover, we introduce intervals of the form $[x_{mid} + e_{lo}, x_{mid} + e_{hi}]$ and we show that, in case of narrow intervals, both alternative forms provide tighter enclosures compared to the classical interval form. We also compare all interval representations using computational benchmarks.

References:

[1] T.J. Dekker, A floating-point technique for extending the available precision, *Numerische Mathematik*, 18 (1971/72), pp. 224–242.

[2] *IEEE standard for binary floating-point arithmetic*, (Technical Report IEEE Std 754-1985), The Institute of Electrical and Electronics Engineers, 1985.

[3] R.E. Moore, R.B. Kearfott, M.J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[4] S.M. Rump, Fast and parallel interval arithmetic, *BIT*, 39, pp. 534–554.

[5] A. Wittig, M. Berz, Rigorous high precision interval arithmetic in COSY INFINITY, *Proceedings of the Fields Institute*, 2009.

Rigorous computation with function enclosures in Chebyshev basis*

Tomáš Dzetkulič

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2
182 07 Prague 8, Czech Republic
dzetkulic@cs.cas.cz

Keywords: initial value problem, rigorous integration, Chebyshev basis

When rigorously computing with a real continuously differentiable function, a Taylor polynomial is commonly used to replace the actual function. The Taylor polynomial remainder is bounded to create a conservative enclosure of the function. One of the applications of such a rigorous function enclosure lies in verified algorithms for integration of nonlinear ordinary differential equations [4].

In this paper, we present a multivariable function enclosure using the Chebyshev polynomial instead of the Taylor polynomial. Since the Chebyshev series converge faster for all analytic functions compared to the Taylor series, our function enclosures approximate real analytic functions with tighter remainder intervals.

In the existing works on Chebyshev polynomials [1,2], only operations with functions in one variable are described. In [1], the function approximation is stored in the form of function values in the Chebyshev nodes. The authors use non-rigorous methods to compute coefficients of Chebyshev polynomials, and no enclosure of the exact function value is guaranteed. On the other hand, the authors in [2] use rigorous methods, but only addition, multiplication and composition of one variable functions are presented.

We present an efficient algorithm for rigorous addition, substraction, multiplication, division, composition, integration and derivative of multi-variable Chebyshev function enclosures. Our publicly available implementation [3] supports function enclosures based on both Taylor and Chebyshev polynomials and allows their comparison. Computational experiments with the initial value

*This work was supported by Czech Science Foundation grant 201/09/H057, Ministry of Education, Youth and Sports project number OC10048 and long-term financing of the Institute of Computer Science (RVO 67985807). The author would like to thank Stefan Ratschan for a valuable discussion and helpful advice.

problem of ordinary differential equations show that the approach is competitive with the best publicly available verified solvers.

References:

[1] Z. Battles and L. N. Trefethen, An extension of MATLAB to continuous functions and operators, *SIAM J. Sci. Comput.*, 25 (2004), No. 5, pp. 1743–1770.

[2] N. Brisebarre and M. Joldeş, Chebyshev interpolation polynomial-based tools for rigorous computing, In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation ISSAC'10*, ACM, New York, 2010, pp. 147–154.

[3] T. Dzetkulič, `http://odeintegrator.sourceforge.net`, 2012, software package *ODEIntegrator*.

[4] K. Makino and M. Berz, Rigorous integration of flows and ODEs using Taylor models, *Symbolic Numeric Computation*, 2009, pp. 79–84.

Solving the Table Maker's Dilemma by reducing divergence on GPU

Pierre Fortin and Mourad Gouicem and Stef Graillat

UPMC Univ Paris 06 and CNRS UMR 7606, LIP6,
4 place Jussieu, F-75252, Paris cedex 05, France
{pierre.fortin, mourad.gouicem, stef.graillat}@lip6.fr

Keywords: Table Maker's Dilemma, Graphical Processing Unit, correct rounding, elementary functions

The IEEE 754-2008 standard recommends correctly rounding elementary functions. However, these functions are transcendental and their results can only be approximated with error $\epsilon > 0$. If $\circ_p$ is a rounding function at precision p, there may exist some arguments x, called (p, ϵ) *hard-to-round* arguments, such that $\circ_p(f(x) - \epsilon) \neq \circ_p(f(x) + \epsilon)$, inducing an uncertainty on the rounding of $f(x)$. Finding an error ε such that there are no (p, ε) hard-to-round arguments is known as the Table Maker's Dilemma (TMD).

There exist two major algorithms to solve the TMD for elementary functions which are Lefvre's and SLZ algorithms [2, 3]. The most computationally intensive step of these algorithms is the (p, ϵ) *hard-to-round* argument search since its complexity is exponential in the size of the targeted format. It takes for example several years of computation to get all of them for the classic exponential function in double precision and the same holds for all other classical elementary functions. Hence, getting (p, ϵ) *hard-to-round* arguments is a challenging problem. In order to obtain these (p, ϵ) *hard-to-round* arguments for larger formats (extended precision, quadruple precision), the implemented algorithms should be able to efficiently operate on petaflops systems. In the long term, we would expect to require the correct rounding of some functions in the next versions of the IEEE 754 standard, which will allow to completely specify all the components of numerical software.

High-performance computing systems increasingly rely on many-core chips such as Graphical Processing Units (GPU), which present a partial SIMD execution (Single Instruction Multiple Data). However, when the control flows of the threads on a SIMD unit diverge, the execution paths are serialized. Hence, in order to efficiently use GPU, one has thus to avoid divergence, i.e. manage to have regular control flow within each group of threads executed on the same

SIMD unit.

This work is a first step for solving the TMD on many-core architectures. We focused on Lefèvre's algorithm [2] as it is efficient for double precision. Also, it is embarrassingly parallel and fine-grained which makes it suitable for GPU. We first deployed this algorithm on GPU using the most efficient (to our knowledge) implementation techniques [5]. Then we redesigned it using the concept of continued fractions. This made it possible to obtain a better understanding of Lefèvre's algorithm and a new algorithm which is much more regular. More precisely, we strongly reduce two major sources of divergence of Lefèvre's algorithm: loop divergence and branch divergence. Compared to the reference implementation of Lefèvre's algorithm on a single high-end CPU core, the deployment of Lefèvre's algorithm on an NVIDIA Fermi GPU offers a speedup of 15x whereas the new algorithm enables a speedup of 52x.

References:

[1] J.M. Muller, N. Brisebarre, F. de Dinechin, C.P. Jeannerod, V. Lefèvre, G. Melquiond, N. Revol, D. Stehlé, S. Torres, *Handbook of Floating-point Arithmetic*, Birkhauser, 2009.

[2] V. Lefèvre, New Results on the Distance Between a Segment and $\mathbb{Z}^2$. Application to the Exact Rounding, *Proceedings of the 17th IEEE Symposium on Computer Arithmetic*, 2005, pp. 68–75.

[3] D. Stehlé, V. Lefèvre, Paul Zimmermann, Searching worst cases of a one-variable function using lattice reduction, *IEEE Transactions on Computers*, 54 (2005), pp. 340–346.

[4] A. Ziv, Fast evaluation of elementary mathematical functions with correctly rounded last bit, *ACM Trans. Math. Softw.*, 17 (1991), pp. 410–423.

[5] P. Fortin, M. Gouicem, S. Graillat, Towards solving the Table Maker's Dilemma on GPU, *Proceedings of the 20th International Euromicro Conference on Parallel, Distributed and Network-based Processing*, 2012, pp. 407–415.

Efficient angle summation algorithm for point inclusion test and its robustness

Stepan Gatilov

Novosibirsk State University
2, Pirogova st.
630090 Novosibirsk, Russia
stgatilov@gmail.com

Keywords: point inclusion, point-in-polygon, angle summation, winding angle, bounding volume hierarchy, numerical stability

The point inclusion test is a classical problem of computational geometry. The problem statement is: given a two-dimensional domain bounded by a piecewise smooth Jordan curve, determine whether a certain point belongs to it. The boundary curve is comprised of a sequence of smooth curvilinear edges. It should be noted that the classical problem definition considers only a polygonal boundary described by sequence of its vertex points.

An excellent survey of the classical point-in-polygon methods is given in [1]. In the conclusion it advises to "avoid the angle summation test like the plague" due to its high constant factor in the time complexity. The most notable other methods are the ray intersection test and the test based on barycentric coordinates. The robustness of these methods is studied in [2]. The barycentric coordinate test is shown to be unstable when the point lies on a diagonal of the polygon. The ray intersection test potentially can fail when the ray passes through a vertex, although this instability can be completely eliminated for the classical problem definition.

The geometric data in computer aided design is often imprecise. The boundary is represented by individual edges and the incident vertices of any two consecutive edges may differ up to the tolerance value. This gap between incident vertices renders the ray intersection and barycentric coordinate tests unstable. However, the angle summation method continues to be backwards stable.

The purpose of this work is twofold: first, analyze the stability of the angle summation algorithm; and second, introduce a preprocessing optimization for the many-points-and-one-domain scenario.

The numerical behavior of the angle summation algorithm is analyzed. To calculate the angle between vectors, the FPATAN x87 instruction (atan2) is used for maximum accuracy. Line segments and circular arcs are considered as the edges. It is proven that the answer of the point inclusion query must be correct given that the point is far enough from the boundary. The answer remains correct even if the edges are perturbed slightly, potentially introducing gaps between incident vertices.

With some preprocessing, the subsequent angle summation queries can be accelerated significantly as follows. An axis-aligned bounding box is calculated for each edge, and a bounding volume hierarchy (BVH) is constructed from all of them. Angle summation queries are processed by traversing the BVH recursively. The winding angle is calculated instantly for a BVH node if the point lies outside of the corresponding box.

Given that all the boxes are tight, the algorithm works in $O(K \log \frac{n}{K} + KT)$ time, where T is the amount of time required to calculate the winding angle for a single edge, and $2\pi K$ is the "absolute winding angle" of the boundary (assuming $K \leq n$). The absolute winding angle is a sum of unsigned winding angles for all the infinitesimal pieces of the boundary. It is supposed that often $K \ll n$ in practice. For instance, for any convex domain $K = 1$, which means that queries takes optimal $O(\log n + T)$ time. Some upper bounds for K are given.

References:

[1] E. Haines, Point in polygon strategies, *Graphics Gems IV*, ed. Paul Heckbert, Academic Press, 1994, pp. 24–46.

[2] S. Schirra, How reliable are practical point-in-polygon strategies? *Proceedings of the 16th Annual European Symposium on Algorithms*, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 744–755.

Subinterval analysis. First results

Alexander Harin

Modern University for the Humanities
32, Nizhegorodskaya str.
109029 Moscow, Russia
aaharin@yandex.ru

Keywords: incomplete information, large databases, Internet, economics

The report is devoted to subinterval analysis as a new direction, branch of interval analysis. Subinterval analysis or subinterval weights analysis was founded in [1]. It deals usually with weights as whole characteristics of subintervals.

1. Subinterval arithmetic

Suppose a finite quantity or function $w(x_k)$ is defined on an interval $\boldsymbol{X}_{Total}$ and is known within the accuracy of adjacent subintervals $\{\boldsymbol{X_s}\} : s = 1, 2, ..., S : 1 < S < \infty$, $\underline{X_s} < \underline{X_s + 1}$, of $\boldsymbol{X}_{Total} \equiv \boldsymbol{X}_{1..S}$. At that, many characteristics, such as moments (mean, dispersion, etc.) of $w(x_k)$ are the interval ones.

Let us define the weight of $\boldsymbol{X_s}$ as

$$wht\, X_s \equiv w_s \equiv \sum_{x_k \in \boldsymbol{X_s},\ x_k \notin \boldsymbol{X_{s+1}}} w(x_k)$$

Subinterval arithmetic calculate and rigorously evaluate characteristics of quantities, intervals and subintervals, e.g., by the "Ring of formulas" for widths $wid\, M_{Total}$ of interval $\boldsymbol{M_{Total}}$ of mean of $w(x_k)$

$$wid\, M_{Total} = \sum_{s=1}^{S} wid\, X_s\ w_s = wid\, X_{1..S} - \sum_{s=1}^{S} wid\, X_s \sum_{n=1,...S|n\neq s} w_n =$$

$$= wid\, X_{Total} - \sum_{s=1}^{S} w_s \sum_{n=1,...S|n\neq s} wid\, X_n$$

2. Subinterval analysis of inexact information

2.1. Decision making

If the width and weight of any subinterval cannot be less than nonzero values, then nonzero ruptures exist between the interval $\boldsymbol{M_{Total}}$ of mean value and the bounds of $\boldsymbol{X}_{Total}$ (see [1]). These ruptures explain basic utility paradoxes.

2.2. Global optimization

An analog of Lipschitz's condition may be defined for weights of elementary subintervals and subboxes $\boldsymbol{X_{Elem,s}}$, $\boldsymbol{X_{Elem,t}}$: $\boldsymbol{X_{Elem,s}} \cap \boldsymbol{X_{Elem,t}} \neq \emptyset$

$$|wht\, X_{Elem,s} - wht\, X_{Elem,t}| \leq \Delta_{wht} \equiv \Delta_w$$

allowing discontinuity of the function and revealing new relations.

3. Subinterval analysis of exact but incomplete information

3.1. Theorem of interval character of incomplete knowledge

If a finite nonnegative quantity is exactly known everywhere except two points, the distance between them is nonzero and the values of the quantity in them may vary not less than over a nonzero interval, then any moment of the quantity is known within the accuracy not better than a nonzero interval.

This theorem extends essentially the realm of interval analysis applications.

4. Subinterval approximation of exact information through time, space, ...

4.1. Large databases

A ternary subinterval one-dimentional picture, image needs only 2 numbers as the coordinates of two intersections of 3 subintervals. The picture of N-dimensional plot of 1000^N bytes needs only 2^N bytes.

5. Applications: Accounting. Macroeconomics. Economics. Population analysis. Recognition. Internet.

Accounting is a natural application of time subintervals as months, quarters for gain, profit, etc. Audit incomplete knowledge can be processed by subinterval analysis. Macroeconomics is a natural application of space subintervals as town, sity, province, state, etc. Populations subintervals as sex, age, profession, wage, etc. may be used. Subinterval images and pictures may be used for preliminary analysis and recognition and greatly accelerate them in large databases. Internet is a prospective field for subinterval analysis also.

References:

[1] A.A. Harin, About possible additions to interval arithmetic, *Proceedings of the X International FAMES2011 Conference*, Krasnoyarsk, 2011, `http://eventology-theory.ru/0-lec/X-fames2011-24-October.pdf`, pp. 356–359 (in Russian).

Theorem on interval character of incomplete knowledge. Subinterval analysis of incomplete information

Alexander Harin

Modern University for the Humanities
32, Nizhegorodskaya str.
109029 Moscow, Russia
aaharin@yandex.ru

Keywords: interval analysis, incomplete information, durable processes

Theorem. If a finite nonnegative quantity is defined on a finite segment and is exactly known everywhere except two points, the distance between these points is nonzero and the values of the quantity in these points may vary not less than over a nonzero interval, then any moment of the quantity, including the mean and the dispersion of the quantity, is known within the accuracy not better than another nonzero interval.

The proof is quite simple (see [1]), but the theorem enlarges the interval analysis to the fields of exact but incomplete knowledge, of planning and control of durable measurements, researches, business, work and other processes, etc.

Analysis example 1. At equal widths $wid\, X_s = wid\, X_1$ of subintervals $\boldsymbol{X_s}$, of a total interval $\boldsymbol{X_{Total}} \equiv \boldsymbol{X_{1..S}}$ we obtain from Novoselov formula (see [1]), for the width $wid\, M_{Total}$ of the interval $\boldsymbol{M_{Total}}$ of the total mean value

$$wid\, M_{Total} = \sum_{s=1}^{S} wid\, X_s\; w_s = wid\, X_1 = \frac{wid\, X_{1..S}}{S} \equiv \frac{wid\, X_{Total}}{S}$$

To prove rigorously this simple but, strictly speaking, not obvious conclusion we do not need any information about weights of subintervals or of interval.

Analysis example 2. Assume the width $wid\, X_{First} = 2$ and the weight $w_{First} = 0.7$ of only a single or first subinterval $\boldsymbol{X_{First}} = [2, 4]$ of a total interval $\boldsymbol{X_{Total}} = [A, B] = [0, 10]$ are known (see Fig. 1). Then from Ring of formulas (see [1]) for the interval $\boldsymbol{M_{Total}}$ of the total mean value

$$w_{First} wid\, X_{First} \le wid\, M_{Total}$$

$$wid\, M_{Total} \le wid\, X_{Total} - w_{First}(wid\, X_{Total} - wid\, X_{First})$$

$$\underline{M_{Total}} \ge \underline{X_{Total}} + wid\,(\underline{X_{First}} - \underline{X_{Total}})\; w_{First} = 0 + 2 \times 0.7 = 1.4$$

$$\overline{M_{Total}} \le \overline{X_{Total}} - wid\,(\overline{X_{Total}} - \overline{X_{First}})\; w_{First} = 10 - 6 \times 0.7 = 5.8$$

$$2 \times 0.7 = 1.4 \le wid\, M_{Total} \le 10 - 0.7 \times 8 = 4.4$$

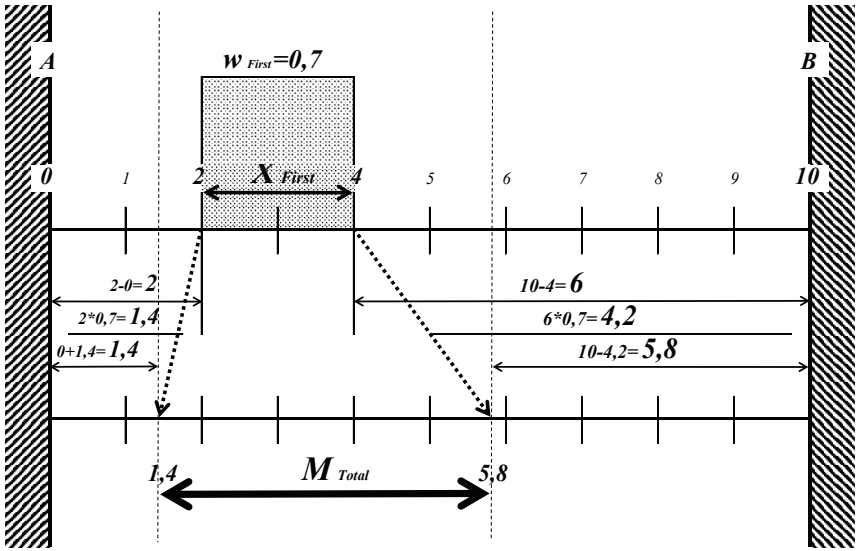

Figure 1: An illustrative example of calculations of the interval M_{Total} of mean value with the help of the only (or the first) measurement.

Note, although we use the incomplete information, all evaluations of the both examples are rigorous and exact as usually in the interval analysis.

References:

[1] A.A. Harin, Theorem of interval character of incomplete knowledge. Its applications to planning of experiments, *Moscow Science Review*, Vol. 16 (2011, No. 12), pp. 3–5 (in Russian).

Arithmetic and algebra of mapped regular pavings

Jennifer Harlow[1], Raazesh Sainudiin[1,2] and Warwick Tucker[3]

[2]Laboratory for Mathematical Statistical Experiments and
[1]Department of Mathematics and Statistics,
University of Canterbury, Christchurch, NZ
[3]Department of Mathematics,
Uppsala University, Uppsala, Sweden
raazesh.sainudiin@gmail.com

Keywords: finite rooted binary trees, tree arithmetic and algebra

A regular paving [1,2] is a finite succession of bisections that partition a root box $\boldsymbol{x}$ in $\mathbb{R}^d$ into sub-boxes using a tree-based data structure. Such trees are also known as plane binary trees [3] or finite rooted binary trees [4]. Here we extend regular pavings to mapped regular pavings which allow us to map sub-boxes in a regular paving of $\boldsymbol{x}$ to elements in some set $\mathbb{Y}$. Arithmetic operations defined on $\mathbb{Y}$ can be extended point-wise in $\boldsymbol{x}$ and carried out in a computationally efficient manner using $\mathbb{Y}$-mapped regular pavings of $\boldsymbol{x}$. The efficiency is due to recursive algorithms on the finite rooted binary trees that are closed under union operations. We provide a novel memory-efficient arithmetic over mapped partitions based on regular pavings and develop an inclusion algebra based on intervals in a complete lattice $\mathbb{Y}$ over a dense class of such partitions of $\boldsymbol{x}$ based on finite rooted binary trees.

Some application of mapped regular pavings include (i) computationally efficient representations of radar-observed flight co-trajectories over a busy airport that is endowed with arithmetic for pattern-recognition [5], (ii) averaging of histograms in multi-dimensional nonparametric density estimation, (iii) arithmetic over a class of simple functions that are dense for continuous real-valued functions, (iv) arithmetic over an inclusion algebra of interval-valued functions to enclose locally Lipschitz real-valued functions, (v) obtaining the marginal density by integrating along any subset of its coordinates, (vi) obtaining the conditional function by fixing the values in the domain on a subset of coordinates and (vii) producing the domain with the highest coverage region.

More generally, the mapped regular pavings allow any arithmetic defined over elements in a general set $\mathbb{Y}$ to be extended to $\mathbb{Y}$-mapped regular pavings.

The properties of such approximations and arithmetic operations are theorized, implemented and demonstrated with examples.

References:

[1] H. Samet, *The design and analysis of spatial data structures*, Addison-Wesley, Boston, 1990.

[2] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer-Verlag, London, 2001.

[3] R. P. Stanley, *Enumerative Combinatorics, Vol. 2*, Cambridge University Press, Cambridge, 1999.

[4] J. Meier, *Groups, graphs and trees: an introduction to the geometry of infinite groups*, Cambridge University Press, Cambridge, 2008.

[5] G. Teng, K. Kuhn, and R. Sainudiin, Statistical regular pavings to analyze massive data of aircraft trajectories, *Journal of Aerospace Computing, Information and Communication*, (2012), accepted for publication.

Verified computation of symmetric solutions to continuous-time algebraic Riccati matrix equations

Behnam Hashemi

Department of Mathematics, Faculty of Basic Sciences
Shiraz University of Technology, Shiraz, Iran
hashemi@sutech.ac.ir

Keywords: interval arithmetic, enclosure methods, Fréchet derivative

Our goal is to develop methods based on interval arithmetic which provide guaranteed error bounds for solutions of the continuous-time algebraic Riccati equation (CARE)

$$R(X) = A^T X + XA - XSX + Q = 0, \tag{1}$$

where A, S and Q are given matrices in $\mathbb{R}^{n\times n}$ and $X \in \mathbb{R}^{n\times n}$ is the unknown solution.

The severe disadvantage of the standard Krawczyk operator for the particular equation (1) is that its computation needs a total cost of $\mathcal{O}(n^6)$. An interval Newton algorithm has been used in [2] for enclosing a symmetric solution to the CARE (1) with a similar cost of $\mathcal{O}(n^6)$. The following theorem is the main theoretical basis for our modified Krawczyk operator that is more efficient to implement.

Theorem 1 [1]. *Assume that $f : D \subset \mathbb{C}^N \to \mathbb{C}^N$ is continuous in D. Let $\check{x} \in D$ and $\boldsymbol{z} \in \mathbb{IC}^n$ be such that $\check{x} + \boldsymbol{z} \subseteq D$. Moreover, assume that $\mathcal{P} \subset \mathbb{C}^{n\times n}$ is a set of matrices containing all slopes $P(\check{x}, y)$ for $y \in \check{x} + \boldsymbol{z} =: \boldsymbol{x}$. Finally, let $R \in \mathbb{C}^{n\times n}$. Denote $\mathcal{K}_f(\check{x}, R, \boldsymbol{z}, \mathcal{P})$ the set*

$$\mathcal{K}_f(\check{x}, R, \boldsymbol{z}, \mathcal{P}) := \{-Rf(\check{x}) + (I - RP)z : P \in \mathcal{P}, z \in \boldsymbol{z}\}. \tag{2}$$

Then, if $\mathcal{K}_f(\check{x}, R, \boldsymbol{z}, \mathcal{P}) \subseteq \operatorname{int} \boldsymbol{z}$, the function f has a zero x^* in

$$\check{x} + \mathcal{K}_f(\check{x}, R, \boldsymbol{z}, \mathcal{P}) \subseteq \boldsymbol{x}.$$

Moreover, if $\mathcal{P}$ also contains all slope matrices $P(y, x)$ for $x, y \in \boldsymbol{x}$, then this zero is unique in $\boldsymbol{x}$.

Suppose that the closed-loop matrix $A - SX$ is nondefective. Therefore, it satisfies the following spectral decomposition

$$A - SX = V\Lambda W \quad \text{with} \quad \begin{array}{l} V, \Lambda, W \in \mathbb{C}^{n\times n}, \quad VW = I, \\ \Lambda = Diag(\lambda_1, \lambda_2, \cdots, \lambda_n) \ \text{ diagonal.} \end{array}$$

In general the following identity holds

$$r'(x) = (V^{-T} \otimes W^T) \cdot \Big(I \otimes [W(A - SX)W^{-1}]^T + [V^{-1}(A - SX)V]^T \otimes I\Big) \cdot (V^T \otimes W^{-T}),$$

where $\otimes$ stands for the Kronecker product of matrices. Hence, an approximate inverse for $r'(\boldsymbol{X})$ is

$$R = (V^{-T} \otimes W^T) \cdot \Delta^{-1} \cdot (V^T \otimes W^{-T}), \tag{3}$$

where $\Delta = I \otimes \Lambda^T + \Lambda^T \otimes I$. For any matrix $X \in \mathbb{C}^{n\times n}$ and any vector $z \in \mathbb{C}^{n^2}$ we have

$$\big(I_{n^2} - R(I_n \otimes (A - SX)^T + (A - SX)^T \otimes I_n)\big)\, z = (V^{-T} \otimes W^T)\ \Delta^{-1}\ \Omega\ (V^T \otimes W^{-1})z, \tag{4}$$

where $\Omega = \Delta - I_n \otimes \left(W(A - SX)W^{-1}\right)^T - \left(V^{-1}(A - SX)V\right)^T \otimes I_n$.

Theorem 2. *Suppose that S and the solution X for the CARE (1) are both symmetric matrices. The interval arithmetic evaluation of the Fréchet derivative of $R(X)$ contains slopes $P(y, x)$ for all $x, y \in \boldsymbol{x}$.*

Formula (4) together with the above theorem are what we need for enclosing the set $\{(I - RP)z : P \in \mathcal{P}, z \in \boldsymbol{z}\}$, as a part of our modified Krawczyk operator $\mathcal{K}_r(\check{x}, R, \boldsymbol{z}, \mathcal{P})$ defined in (2). Here, $r := r(x)$ denotes the vector form of the CARE (1) and R denotes our factorized preconditioner (3). Note that Ω is close to a diagonal matrix and also the multiplication by Δ^{-1} can be done cheaply via Hadamard division. In addition, the first term $-Rr(\check{x})$ in (2), where R is defined by (3), can be enclosed in a similar fashion. An important point is the use of formula $\mathrm{vec}(ABC) = (C^T \otimes A)\mathrm{vec}(B)$, where vec(.) denotes the operator of stacking the columns of a matrix into a long vector. As a result, our algorithm needs only $\mathcal{O}(n^3)$ arithmetic operations. It is mainly based on matrix-matrix multiplications and therefore can be implemented very *efficiently* in Level 3 BLAS. Numerical results will be reported at the conference.

References:

[1] A. Frommer, B. Hashemi, Verified computation of square roots of a matrix, *SIAM Journal on Matrix Analysis and Applications*, 31 (2009), No. 3, pp. 1279–1302.

[2] W. Luther, W. Otten, Verified calculation of the solution of algebraic Riccati equation, *Developments in Reliable Computing* (T. Cendes, Ed.), Kluwer, 1999, pp. 105–119.

Computing interval power functions

Oliver Heimlich and Marco Nehmeier
and Jürgen Wolff von Gudenberg

Institute for Computer Science, University of Würzburg
D 97074 Würzburg, Germany
wolff@informatik.uni-wuerzburg.de

Keywords: interval arithmetic, elementary functions, power function

We can distinguish between four variants of the general real power function x^y depending mainly on the domain. For strictly positive values of x, e.g., powers with arbitrary y can be computed without problems, whereas adding the powers of 0 infiltrates the problem of determining 0^0. In the history of mathematics we can find quite a few papers that support the opinion that $0^0 = 0$, but also many others that support $0^0 = 1$. The decision for one of these alternatives can not be taken without regarding the specific context. If there is no such context, we propose to define that x^y is undefined for $(0, 0)$. For negative x, it certainly makes sense to allow integer exponents only and thus leading to a discrete domain. Nevertheless the semantics is clearly and uniquely defined. This variant has another advantage, it equals exactly the evaluation of the complex variant applied to real inputs. Finally we discuss the variant which is defined for rational exponents with odd denominators. This variant may have some applications in interval analysis, because the domain is dense in the corresponding contiguous interval.

In this talk we discuss those four variants and try to solve the contentious issues depending on the context. We start with a detailed analysis of the behaviour when x or y approach $\pm\infty$ or ± 0 or when x approaches 1. With this information interval versions of each variant can be computed by efficient algorithms. For the positive case, e.g., we developed some improvements to IntLab that reduce the runtime by 40%. For the other variants algorithms are based on the positive version. It is, however, strange to define a function that is meant for an interval extension on a discrete grid. The algorithms are accompanied by a rigorous treatment of rounding errors.

Last but not least we test our implementation with respect to accuracy and speed. The former tests mainly use the multiprecision interval library MPFI, some extension of its functionality is needed. The efficiency tests compare the runtime with several other well known libraries: C-XSC, filib++ and Boost as well as IntLab.

Computing reverse interval power functions

Oliver Heimlich and Marco Nehmeier
and Jürgen Wolff von Gudenberg

Institute for Computer Science, University of Würzburg
Am Hubland
D 97074 Würzburg, Germany
nehmeier@informatik.uni-wuerzburg.de

Keywords: interval arithmetic, elementary functions, reverse functions, power function

There are a few difficulties with the inversion of interval functions. Plain transformations may create results that are either of low quality, i.e., by far overvalue the correct answer, or are wrong.

In this context "reverse operations" [2] act as an effective solution for the problems encountered: A single operation shall compute an interval containing solutions to basic equations, which comprise intervals, interval operations and optional interval constraints.

> For a (partial) binary arithmetic operation $\circ$ there are two *binary reverse operations* on intervals, $\circ_1^- : \overline{\mathbb{IR}} \times \overline{\mathbb{IR}} \to \mathcal{P}(\mathbb{R})$ and $\circ_2^- : \overline{\mathbb{IR}} \times \overline{\mathbb{IR}} \to \mathcal{P}(\mathbb{R})$, defined by
>
> $$\circ_1^-(\mathbf{y}, \mathbf{z}) = \{x \in \mathbb{R} \mid \text{there exists } y \in \mathbf{y} \text{ with } x \circ y \in \mathbf{z}\} \text{ and}$$
> $$\circ_2^-(\mathbf{x}, \mathbf{z}) = \{y \in \mathbb{R} \mid \text{there exists } x \in \mathbf{x} \text{ with } x \circ y \in \mathbf{z}\}$$
>
> with $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \overline{\mathbb{IR}}$.

Note that in principle we have

$$\circ_1^-(\mathbf{y}, \mathbf{z}) = \mathbf{z}^{1/\mathbf{y}}$$
$$\circ_2^-(\mathbf{x}, \mathbf{z}) = \log_{\mathbf{x}} \mathbf{z}$$

The details are analyzed in the paper. It turns out that already for the most restricted domain 8 groups of inverse images are necessary depending on the overlapping relation [1].

Inverse images of the other variants are more complicated and require distinction between a lot more cases. With an application or extension of the overlapping relation we show that 26 cases are sufficient. However, in most cases the reverse interval operations produce results which can simply be computed as the hull of one or two intervals which are possibly to be intersected with the subset of even or odd integral numbers. But, for the first reverse function there are even some cases where the result is a union of infinitely many and possibly disjoint intervals.

The algorithm works as follows: At first, an enclosure of the union of the many intervals is computed, which, when intersected with $\mathbf{x}$, already produces an enclosure of the result. Each boundary of this enclosure is sharp if, and only if, it is part of the union of many intervals. Thus, the result's boundaries can further be optimized if they are not part of the reverse operation's result. At this point, the algorithm utilizes that the relevant part of the inverse image of $\mathbf{z}$ consists of individual lines which are parallel to the x axis. The idea behind the algorithm is illustrated graphically.

References:

[1] M. Nehmeier, J. Wolff von Gudenberg, Interval comparisons and lattice operations based on the interval overlapping relation, In *Proceedings of the World Conference on Soft Computing 2011 (WConSC'11)*, San Francisco, CA, USA, 2011.

[2] A. Neumaier, Vienna proposal for interval standardization, Final version, December 19, 2008, `www.mat.univie.ac.at/ neum/ms/1788.pdf`.

New directions in interval linear programming

Milan Hladík

Charles University, Faculty of Mathematics and Physics
Malostranské nám. 25
118 00, Prague, Czech Republic
University of Economics, Faculty of Informatics and Statistics
nám. W. Churchilla 4
13067, Prague, Czech Republic
hladik@kam.mff.cuni.cz

Keywords: linear programming, interval equations, interval inequalities

Linear programming is undoubtedly one of the most frequently used techniques in problem solving. Since real life data are often not known precisely due to measurement errors, estimations and other kinds of uncertainties, we have to reflect it in the theory of linear programming. Modeling such uncertainties by intervals gives rise to the research area called interval linear programming [1,2]. Herein, we suppose that interval domains of uncertain quantities are a priori given, and the aim is to calculate verified results giving rigorous enclosures (or other types of answers) valid for all possible realizations of interval data.

There are many problems regarding interval linear programming, such as verifying feasibility, (un)boundedness or optimality for some or for all realizations of interval quantities; some of them are polynomially solvable, but the others are NP-hard. However, there are two main directions of determining (enclosing) the optimal value range and the optimal solution set. While the former was intensively studied in the past and many results concerning computational complexity and methods are available, there is still lack of theory and practical methods for the latter. Rigorously and tightly enclosing the optimal solution set is the most challenging problem in this subject. Traditional approaches were based on the so called basis stability, meaning that there is a basis being optimal for each realization of intervals. Under basis stability, the problems can be solved very efficiently. Checking this property may be computationally expensive in general, but strong sufficient conditions exist. The problem is, however, that in many situations (e.g. under basis degeneracy), the problem is not basis stable even for tiny intervals. Overcoming the non-basis stability is remaining to be an important but difficult problem.

In the talk, we survey the known results and present recent developments as well. We focus on the computational complexity, methods and other aspects of enclosing the optimal value range and the optimal solution set. We discuss applications of this technique in diverse areas. Besides many real-world optimization problems (in economics, environmental management, logistics, ...), interval linear programming may also serve as a supporting tool for linear relaxation in constraint programming and global optimization, in matrix games with inexact data or in statistics in linear regression on uncertain data by using L_1 or L_∞ norm. Sensitivity analysis, frequently used in economical operations research, can be extended from the traditional one-parameter case to the case with multiple parameters situated in diverse positions. Eventually, we mention some open problems and challenges for the future research.

References:

[1] M. Fiedler, J. Nedoma, J. Ramík, J. Rohn, and K. Zimmermann, *Linear Optimization Problems with Inexact Data*, Springer, New York, 2006.

[2] M. Hladík, *Interval Linear Programming: A Survey*, in Z.A. Mann, editor, *Linear Programming — New Frontiers in Theory and Applications*, chapter 2, pages 85–120, Nova Science Publishers, New York, 2012.

Computing enclosures of overdetermined interval linear systems

Jaroslav Horáček[1,2] and Milan Hladík[1,2]

[1] Charles University, Faculty of Mathematics and Physics
Malostranské nám. 25, 118 00, Prague, Czech Republic

[2] University of Economics, Faculty of Informatics and Statistics
nám. W. Churchilla 4, 13067, Prague, Czech Republic

horacek@kam.mff.cuni.cz, hladik@kam.mff.cuni.cz

Keywords: interval linear systems, enclosure methods, overdetermined systems

Real-life problems can be described by different means: difference and differential equations, linear and nonlinear systems, etc. The various descriptions can be often transformed to each other using only linear equalities (or inequalities). That is why interval linear systems are still in the focus of researchers. By an interval linear system, we mean a system $\boldsymbol{A}x = \boldsymbol{b}$, where $\boldsymbol{A}$ is an interval $m \times n$ matrix and $\boldsymbol{b}$ is an $m-$dimensional interval vector. We will now consider a special class of these systems called *overdetermined systems*. They are the systems for which $m > n$ holds. Simply said, they have more equations than variables.

When we talk about interval linear systems, it is necessary to mention, what we mean by the solution of these systems. The solution set Σ of an interval linear system $\boldsymbol{A}x = \boldsymbol{b}$ is an accumulation of all solutions of all instances of this interval system. We get an instance of an interval system, when we independently pick the values from all interval coefficients of the system thus obtaining a point real system. In other words,

$$\Sigma = \{x \mid Ax = b \text{ for some } A \in \boldsymbol{A}, b \in \boldsymbol{b}\}.$$

In what follows, we consider Σ, not the least squares or other approximation of the solution set. If no instance of the interval system has a solution, we call this system *unsolvable*. We are interested in the tightest possible n-dimensional box (aligned with axes) that encloses the solution set of an interval system. It is also called *interval hull* of the solution set. Finding it is an NP-hard problem, so it is often sufficient to find an as narrow as possible n-dimensional box containing the hull that is called an *interval enclosure* of the solution set.

Square systems (those for which $m = n$ holds) can possess some advantageous properties. Their matrix $\boldsymbol{A}$ can be diagonally dominant, positive definite, M-matrix and many more. And we know that our algorithms behave well in these cases. Unfortunately, overdetermined systems do not posses any of these properties. That is why it is sometimes more difficult to solve these systems. However, we can use some earlier designed numerical methods and adapt them to be suitable for computing with intervals.

Here we would like to present the summary of the methods applicable to the overdetermined interval linear systems. They are Gaussian elimination, classical iterative methods, the Rohn method, supersquare and subsquare methods or linear programming.

After introducing each method, we would like to talk about the comparison of all the mentioned methods based on extensive numerical testing for random matrices. We also would like to point out discovered properties of the methods. Some methods fail if the radii of interval coefficients of a system exceed some limits. Some of them, despite their simplicity (supersquare and subsquare methods) return surprisingly narrow results. Another important property of some methods (Gaussian elimination, subsquare methods) is that they can quickly determine, whether the system is unsolvable. This can be useful in applications (system validation, technical computing) where we do care if the systems are solvable or unsolvable.

[1] E.R. Hansen, G.W. Walster, Solving overdetermined systems of interval linear equations, *Reliable Computing*, 12(2006), No. 3, pp.239–243.

[2] J. Horáček, *Overdetermined systems of interval linear equations*, master thesis (in Czech), Charles University in Prague, Department of Applied Mathematics, Prague, 2011.

[3] R.E. Moore, R.B. Kearfott, M.J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

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

[5] J. Rohn, Enclosing solutions of overdetermined systems of linear interval equations, *Reliable Computing*, 2(1996), No. 2, pp.167–171.

Sardana: an automatic tool for numerical accuracy optimization

Arnault Ioualalen[1,2,3], Matthieu Martel[1,2,3]

[1]Univ. Perpignan Via Domitia, Digits, Architectures et Logiciels Informatiques, F-66860, Perpignan, France
[2]Univ. Montpellier II, Laboratoire d'Informatique Robotique et de Microélectronique de Montpellier, UMR 5506, F-34095, Montpellier, France
[3]CNRS, Laboratoire d'Informatique Robotique et de Microélectronique de Montpellier, UMR 5506, F-34095, Montpellier, France
arnault.ioualalen@univ-perp.fr, matthieu.martel@univ-perp.fr

Keywords: numerical accuracy, abstract interpretation, code synthesis.

On computers real numbers are approximated by floating-point numbers defined by the IEEE754 formats [1]. For most computations these formats are precise enough even though the induced approximation errors inherently. However in some cases the accuracy of the calculation is critical and the user needs to certify that his program will always yield an accurate enough output for every valid input. As it is impossible to check the validity of a calculation for every inputs, static analyzers such as Fluctuat [4] or Astrée [2] rely on an interval or relational representation of the inputs, combined to abstract interpretation.
Sardana is a tool designed to automatically rewrite numerical computations performed in floating-point arithmetics in order to optimize their accuracy. Sardana works directly on the source code of a LUSTRE program such as the ones used in real avionic software and produces a new source code as well as an absolute bound of error which is less than the original one. To achieve this Sardana uses: (*i*) Interval analysis, to handle large sets of inputs and not only isolated traces, (*ii*) A novel intermediate representation of program called APEG [5] which allows us to manipulate many transformed versions of the initial program in a compact way, (*iii*) A local search heuristic [5] to synthesize from an APEG a new version of the program, (*iv*) And abstract interpretation [3] to enforce the validity of our analysis of the accuracy.

The first challenge is how to transform a program into a more accurate one. As there is an exponential number of ways to write an arithmetic expression (e.g. a simple sum of n terms), we cannot exhaustively generate all possible transformations. This problem is closely related to the *phase ordering problem* of compilers. We use abstract interpretation to narrow down this search space

while allowing to represent in an abstract way as many transformed versions as possible of the initial program. Our structure of Abstract Program Expression Graph (APEG) is built from the syntactic tree of the source code, and is a compact and efficient way to handle multiple versions of a program without duplication and exponential growth of the structure. As there are many transformations which involve only the same part of the program, APEGs merge them locally into one equivalence class without duplicating the rest of the structure. Also, we introduce the concept of abstraction boxes into APEGs, which are defined by an operator and a set of sub-expressions. Each abstraction box allows to represent the exponential number of expressions that can be synthesized with the given operator over the set of sub-expressions of the box.

Next, Sardana has to extract from an APEG a program which has a better numerical accuracy. We use a limited depth search strategy with memorization. Intuitively we select the best way to evaluate an expression by considering only the best way to evaluate its sub-expressions. To accurately calculate both rounding errors and floating-point values, Sardana uses the GMP and MPFR libraries. Sardana is also able to manipulate any floating-point IEE754 format and fixed-point arithmetic.
Several experimental results have been obtained on various benchmarks and real-case applications, such as: summation (results are 50% closer to the real values), polynomial functions like Taylor expansions (20% to 30% more accurate), and real avionic codes (10% more accurate half the time). Finally, Sardana provides a graphical interface allowing the user to specify the analyzer parameters easily and analyze the results in a user friendly way.

References:

[1] ANSI/IEEE, *IEEE Standard for Binary Floating-point Arithmetic*, Std 754-2008 edition, 2008.

[2] J. Bertrane, P. Cousot, R. Cousot, J. Feret, L. Mauborgne, A. Miné, and X. Rival, Static analysis and verification of aerospace software by abstract interpretation, *AIAA Infotech@Aerospace*, 2010.

[3] P. Cousot and R. Cousot, Abstract Interpretation: A unified lattice model for static analysis of programs by construction of approximations of fixed points, *Principles of Programming Languages*, pp. 238–252, 1977.

[4] D. Delmas, E. Goubault, S. Putot, J. Souyris , K. Tekkal , and F. Vedrine, Towards an industrial use of FLUCTUAT on safety-critical avionics software, *Formal Methods for Industrial Critical Systems (FMICS)*, 2009.

[5] A. Ioualalen and M. Martel, A new abstract domain for the representation of mathematically equivalent expressions, *Static Analysis Symposium (SAS)*, 2012.

Interval analysis and robotics

Luc Jaulin

Labsticc, IHSEV, ENSTA-Bretagne
2, rue François Verny,
29200, Brest, France
luc.jaulin@ensta-bretagne.fr

Keywords: interval arithmetic, contractors, robotics

When dealing with complex mobile robots, we often have to solve a huge set of nonlinear equations. They may be related to some measurements collected by sensors, to some prior knowledge on the environment or to the differential equations describing the evolution of the robot. For a large class of robots these equations are uncertain, enclose many unknown variables, are strongly nonlinear and should be solved very quickly. Hopefully, the number of these equations is generally much larger than the number of variables. We can assume that the system to be solved has the following form

$$\left\{ \begin{array}{c} f_i\left(x, y_i\right) = 0, \\ x \in \mathbb{R}^n, \quad y_i \in [y_i] \subset \mathbb{R}^{p_i}, \\ i \in \{1, \ldots, m\}. \end{array} \right. \tag{1}$$

The vector $x \in \mathbb{R}^n$ is the vector of unknown variables, the vector $y_i \in \mathbb{R}^{p_i}$ is the ith data vector (which is approximately known) and $f_i : \mathbb{R}^n \times \mathbb{R}^{p_i} \to \mathbb{R}$ is the ith function. The box $[y_i]$ is a small box of $\mathbb{R}^n$ that takes into account some uncertainties on y_i. Here, we assume that the number of equations m is much larger that the number of unknown variables n (otherwise, the method will not be able to provide accurate results). Typically, we could have $n = 1000$ and $m = 10000$. In order to provide a fast polynomial algorithm able to find a box $[x]$ that encloses all feasible x, we shall associate, to each equation $f_i\left(x, y_i\right) = 0$, a contractor $C_i : \mathbb{IR}^n \to \mathbb{IR}^n$ that narrows the box $[x]$ without removing any value for x consistent with the ith equation. Such a contractor can be obtained using interval computations [1]. Then we iterate each contractor until no more contraction can be performed. An illustration of the procedure is Figure 1, where the sequence of contractors C_1, C_2, C_3, C_1, C_2, $C_3 \ldots$ is applied. Note that the first contractor C_1 was able to contract the initial box $[x] = [-\infty, \infty]^2$ to the box containing the thick circle.

As an example, we shall consider the SLAM (Simultaneous localization and map building) problem asking whether it is possible for an autonomous robot

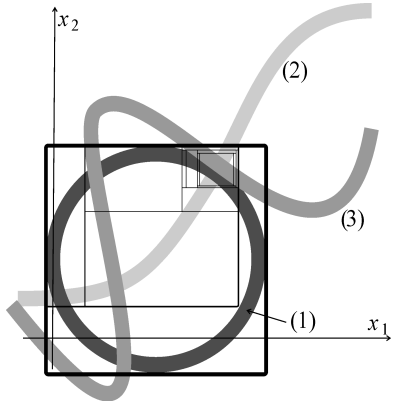

Figure 1: Illustration of the propagation process

to move in an unknown environment and build a map of this environment while simultaneously using this map to compute its location. It is shown in [2] that the general SLAM problem can be cast into the form (1). The corresponding system is strongly nonlinear and classical non-interval methods cannot to deal with this type of problems in a reliable way. The efficiency of the approach will be illustrated on a two-hour experiment where an actual underwater robot is involved. This four-meter long robot build by the GESMA (Groupe d'étude sous-marine de l'Atlantique) is equipped with many sensors (such as sonars, Loch-Doppler, gyrometers, ...) which provide the data. The algorithm is able to provide an accurate envelope for the trajectory of the robot and to compute sets which contain some detected mines in less than one minute. Other examples involving underwater robots and sailboat robots will also be presented.

References:

[1] R.E. Moore, R.B. Kearfott, M.J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[2] L. Jaulin, A nonlinear set-membership approach for the localization and map building of an underwater robot using interval constraint propagation, *IEEE Transaction on Robotics*, 25 (2009), No. 1, pp. 88–98.

Using interval branch-and-prune algorithm for lightning protection systems design

Maksim Karpov

Department of Computer Software, Ivanovo State Power University
34, Rabfakovskaya st.
153003 Ivanovo, Russia
karpov@igt.ispu.ru

Keywords: interval analysis, enclosure methods, lightning protection

The external lightning protection system (LPS) is intended to intercept direct lightning flashes to a structure, including flashes to the side of the structure. The probability of structure penetration by a lightning current is considerably decreased by the presence of a properly designed air-termination system. It interacts with lightning. The form of protection zones and protection of objects and structures depend on its configuration. In the design practice, horizontal sections of protection zone, made at a certain height (usually the highest building is used), are commonly used for checking the safety of objects. The facility is considered to be protected if it is totally covered with these sections. Otherwise, it is necessary to determine the total unprotected area. Knowing the contours at several levels makes it possible to check whether a structure of complicated form is completely inside the protected volume.

Such section is constructed as a group of protection zones sections, which are formed by individual rods as well as by their interactions (pair, triple, multiple). The shape of the section is described by linear and nonlinear constraints. It depends on the applied model of lightning attraction to ground objects. Geometrically, there is a collection of planar closed objects with complicated form. The boundary of the region (outer boundaries and holes) consists of end-connected curves where each point shares only two edges.

Usually sections are based on the geometric modeling kernel, which incorporates low-level data structures and algorithms to support mixed-dimensional geometric modeling. For converting objects that enclose an area into a region, we use Boolean operations and algorithms selecting closed contours. This operations take a long time to complete. Furthermore, we often have to deal with errors in constructions on the kernel side.

We propose a method for computing inner and outer approximations of unprotected region by interval pavings. In this paper we shall consider a covering method that provides a tight piecewise linear interval enclosure of the region. The method is based on the branch-and-prune algorithm suggested in [2], and it generates a covering of the solution set by a collection of smaller and smaller boxes which give increasingly accurate information about the location of the boundary of the region. The proposed new method can be used to speed up geometric computations for lightning protection systems design.

Further details will be considered too, namely, the possibility (and usability) of preconditioning for improving the result and performance of the subdivision algorithm as well as the possibility of parallelizing the method.

We are going to present and discuss numerical results produced by our technique.

References:

[1] IEC 62305-3 Protection against lightning, Part 3: Physical damages and life hazard in structures, *International Electrotechnical Commission*, 2006.

[2] A. Neumaier, The enclosure of solutions of parameter-dependent systems of equations, In *Reliability in Computing*, (ed. by R.E. Moore), Acad. Press, San Diego, 1988, pp. 269–286.

An algorithm to reduce the number of dummy variables in affine arithmetic

Masahide Kashiwagi

Faculty of Science and Engineering
3-4-1 Ookubo, Shinjuku-ku
Tokyo 169-8555, Japan
kashi@waseda.jp

Keywords: affine arithmetic

Affine arithmetic (AA) is an extension of interval arithmetic. In AA, quantities are represented by affine forms:

$$a_0 + a_1\varepsilon_1 + a_2\varepsilon_2 + \cdots + a_n\varepsilon_n$$

where ε_i are dummy variables which satisfy $-1 \leq \varepsilon_i \leq 1$ and express the relation between quantities written in affine form. In AA, number of ε gradually increases and that makes calculation slower.

In this paper, we propose an algorithm to reduce the number of εs .

Note that we should apply the algorithm to as many affine variables on memory as possible simultaneously. Application to small affine variables is not effective.

Consider p affine variables which have q dummy εs:

$$\begin{aligned} &a_{10} + a_{11}\varepsilon_1 + \cdots + a_{1q}\varepsilon_q \\ &a_{20} + a_{21}\varepsilon_1 + \cdots + a_{2q}\varepsilon_q \\ &\qquad\vdots \\ &a_{p0} + a_{p1}\varepsilon_1 + \cdots + a_{pq}\varepsilon_q \end{aligned}$$

we can reduce the number of ε by 'intervalize' several εs. Let S be a index set of εs which we want to erase, we can erase εs by substituting as follows:

$$\begin{aligned} \sum_{i\in S} a_{1i} &\rightarrow (\sum_{i\in S} |a_{1i}|)\varepsilon_{q+1} \\ &\vdots \\ \sum_{i\in S} a_{pi} &\rightarrow (\sum_{i\in S} |a_{pi}|)\varepsilon_{q+p} \end{aligned}$$

Here, p number of new εs are added in order to represent the generated interval.

In the following, we consider to reduce number of εs to r . We select $q-(r-p)$ εs which have small 'intervalize penalty' and intervalize these εs, then we can reduce the number of εs to $(r-p)+p=r$:

$$\sum_{i\notin S} a_{1i} + (\sum_{i\in S} |a_{1i}|)\varepsilon_{q+1}$$

$$\vdots$$

$$\sum_{i\notin S} a_{pi} + (\sum_{i\in S} |a_{pi}|)\varepsilon_{q+p}$$

Now, we will show how to select εs whose 'intervalize penalty' are small. Let vectors $v_1, v_2, \cdots, v_q \in \mathbb{R}^p$ be $v_i = (a_{i1}, \cdots, a_{ip})^T$.

Definition 1 (Penalty Function) *For vector $v = (a_1, \cdots, a_p)^T$ we define penalty function P as follows:*

- *When $a_1 = a_2 = \cdots = a_p = 0$, we define $P(v) = 0$.*
- *Otherwise, let a_s, a_t be the first and second values in the order of absolute values $|a_i|$. That is, $|a_s| \geq |a_t| \geq |a_i| \quad (i \neq s, t)$ hold. Then we define $P(v) = \dfrac{|a_s| \cdot |a_t|}{|a_s| + |a_t|}$.*

We should choose $q-(r-p)$ number of ε in ascending order of the value $P(v_i)$. Note that the penalty function has the following property.

Theorem 1 (Property of Penalty function) *Let $v = (a_1, \cdots, a_p)^T \in \mathbb{R}^p$ and norm of $\mathbb{R}^p$ be maximum norm. Let $L \subset \mathbb{R}^p$ be a line segment defined by*

$$(a_1 \cdots a_p)^T \varepsilon \quad (-1 \leq \varepsilon \leq 1)$$

and let $B \subset \mathbb{R}^p$ be a hyper-rectangular defined by

$$(a_1\varepsilon_1, \cdots, a_p\varepsilon_p)^T \quad (-1 \leq \varepsilon_i \leq 1).$$

Then Housedorff distance between L and B becomes $H(L, B) = 2P(v)$.

That is, $P(v)$ is the maximum distance between L (a line segment defined by original v) and B (hyper-rectangular generated by intervalization of L). We can say that the smaller $P(v)$ is, the smaller increase of range by intervalization.

References:

[1] M.V.A. Andrade, J.L.D. Comba and J. Stolfi, Affine arithmetic, *INTERVAL'94*, St. Petersburg (Russia), March 7-10, 1994.

Uniform second-order polynomial-time computable operators and data structures for real analytic functions

Akitoshi Kawamura[1], Norbert Müller[2],
Carsten Rösnick[3], Martin Ziegler[3]

[1]Tokyo University and [2]Universität Trier and [3]TU Darmstadt

Keywords: computable analysis, complexity theory, analytic functions

Recursive Analysis is the theory of real computation by approximation up to guaranteed prescribable absolute error. Initiated by Alan Turing, it formalizes verified numerics in unbounded precision [1,7] in the common framework of the Theory of Computation [2]. More precisely, a function $f : [0,1] \to \mathbb{R}$ is called *computable* iff a Turing machine can, upon input of every sequence $a_m \in \mathbb{Z}$ with $|x - a_m/2^{m+1}| \leq 2^{-m}$ for $x \in [0,1]$, output a sequence $b_n \in \mathbb{Z}$ with $|f(x) - b_n/2^{n+1}| \leq 2^{-n}$. Any such f is necessarily continuous. More generally, the *Type-2 Theory of Effectivity* [9] studies, compares, and combines so-called representations, that is, encodings for separable metric spaces like $C[0,1]$. Refining mere computability, real complexity theory investigates the running time in terms of the output precision n; see, e.g., [5] and the references therein. Asymptotic growth, polynomial in n, is generally considered practical. Concerning operators and functionals on $C[0,1]$, recall the following strong, nonuniform lower bounds relative to the millennium problem and its strengthenings:

- $\mathrm{Max}(f) := \big([0,1] \ni x \mapsto \max\{f(y) : 0 \leq y \leq x\}\big) \in C[0,1]$ is polynomial-time computable for every polynomial-time computable $f \in C[0,1]$ iff $\mathcal{P} = \mathcal{NP}$; cmp. [5, Theorem 3.7].
- $\int f := \big([0,1] \ni x \mapsto \int_0^x f(y)\,dy\big) \in C[0,1]$ is polynomial-time computable for every polynomial-time computable $f \in C[0,1]$ iff $\mathcal{FP} = \#\mathcal{P}$; cmp. [5, Theorem 5.33].
- The (unique local) solution $u() =: \mathrm{Solve}(f)$ to the ordinary differential equation $u'(t) = f\big(t, u(t)\big)$, $u(0) = 0$, is polynomial-time computable for every polynomial-time computable *Lipschitz*-continuous f iff $\mathcal{P} = \mathcal{PSPACE}$; cmp. [3, Theorem 3.2].

Restricted to functions $f : [0,1] \to \mathbb{R}$ analytic on some complex open neighbourhood of $[0,1]$ on the other hand, the above operators $\mathrm{Max}, \int, \mathrm{Solve}$ have been shown to map polynomial-time computable arguments to polynomial-time

computable values; cmp. [6] and the references therein. However these results are nonuniform, too, in referring to the dependence on n only while regarding arguments f as arbitrary but fixed. We strengthen the latter by presenting and analyzing algorithms receiving both n and f as inputs. More precisely, consider the following three data structures representing a real analytic $f : [0,1] \to \mathbb{R}$:

$\tilde{\alpha}$: As a finite list $\big(M, (x_m), (a_{m,j}), (L_m), (A_m)\big)$ of dyadic centers $x_m \in \mathbb{D} \cap [0,1]$ $(1 \leq m \leq M)$, binary integer bounds A_m, and inverse radii $L_m \in \mathbb{N}$ in unary such that the intervals $\big[x_m - 1/(4L_m), x_m + 1/(4L_m)\big]$ cover $[0,1]$, together with (programs computing the) power series coefficients $a_{m,j} = f^{(j)}(x_m)/j!$ of f around x_m satisfying $|a_{m,j}| \leq A_m \cdot L_m^j$.

$\tilde{\beta}$: A program computing f, together with an integer L in unary such that f is complex analytic even on (an open neighbourhood of) the closed rectangle $\overline{R}_L := \{x + iy \mid -\frac{1}{L} \leq y \leq \frac{1}{L}, -\frac{1}{L} \leq x \leq 1 + \frac{1}{L}\}$ and a binary integer upper bound B to $|f|$ on said $\overline{R}_L$.

$\tilde{\gamma}$: A program evaluating $f|_{\mathbb{D}}$, together with integers A (in binary) and K (in unary) such that $|f^{(j)}(x)| \leq A \cdot K^j \cdot j!$ holds for all $0 \leq x \leq 1$.

We prove them mutually second-order [4] polynomial-time equivalent; and we devise second-order polynomial-time algorithms on them for i) evaluation, ii) addition, iii) multiplication, iv) differentiation, v) integration, and vi) maximization. These may help to improve the mere computability of Bloch's Constant [8] to an algorithm actually calculating some new digits of it.

References:

[1] F. Bornemann, D. Laurie, S. Wagon, J. Waldvogel, *The SIAM 100-Digit Challenge: A Study in High-Accuracy Numerical Computing*, 2004.

[2] M. Braverman, S.A. Cook, Computing over the reals: foundations for scientific computing, *Notices of the AMS*, 53 (2006), No. 3, pp. 318–329.

[3] A. Kawamura, Lipschitz continuous ordinary differential equations are polynomial-space complete, *Computational Complexity*, 19 (2010), No. 2, pp. 305–332.

[4] A. Kawamura, S.A. Cook, Complexity theory for operators in analysis, *Proc. 42nd Ann. ACM Symp. on Theory of Computing*, pp. 495–502; full version to appear in *ACM Transactions in Computation Theory*.

[5] K.-I. Ko, *Computational Complexity of Real Functions*, Birkhäuser, 1991.

[6] N.T. Müller, Constructive aspects of analytic functions, *Informatik-Berichte FernUniversität Hagen*, 190 (1995), pp. 105–114.

[7] N.T. Müller, The `iRRAM`: exact arithmetic in `C++`", *Springer LNCS*, 2064 (2001), pp. 222–252.

[8] R. Rettinger, Bloch's constant is computable, *Journal of Universal Computer Science*, 14 (2008), No. 6, pp. 896–907.

[9] K. Weihrauch, *Computable Analysis*, Springer, 2000.

On rigorous upper bounds to a global optimum

Ralph Baker Kearfott

Department of Mathematics
University of Louisiana at Lafayette
U.L. Box 4-1010
Lafayette, LA 70504-1010 USA
`rbk@louisiana.edu`

Keywords: global optimization, verified bounds, local optimizers

In mathematically rigorous complete search in global optimization, a sharp upper bound on the global optimum is important for the overall efficiency of the branch and bound process. Local optimizers, using floating point arithmetic, often compute a point close to an actual global optimizer. However, particularly with many equality constraints or active inequality constraints, methods for using this approximate local optimizer to obtain a mathematically rigorous upper bound on the global optimum fail. On the other hand, there are various such techniques. Several of these are:

Verify feasibility of a reduced system: We identify equality constraints and active inequality constraints. Provided the total number m of such constraints is less than the number of variables n, we identify a subspace of dimension m in which the m values of the m constraints are sensitive, then apply an m-dimensional interval Newton method within this subspace to prove existence of a feasible point within a small box. This is the technique espoused in [2] or [1], [§5.2.4].

Use the Kuhn–Tucker or Fritz John conditions: Perform an interval Newton method in the $m_1 + m_2 + n + 1$-dimensional space defined by the Fritz John conditions (variables and multipliers), where m_1 and m_2 are the numbers of equality and inequality constraints. This can prove existence of a critical point within a small box surrounding an approximate optimum.

Relax the equality constraints to inequality constraints: This is the approach followed, say, in [3]. Although a slightly different problem is being solved, a point strictly interior to the feasible region can be found, and a simple interval evaluation at that point can be used to verify feasibility.

The preceding verification techniques all must start with a point that is approximately feasible or approximately optimal; the technique is then either applied directly to that point or a small box is constructed around the point, within which feasibility can be verified. Some ways of obtaining such a point are:

Use a local (floating point) optimizer (such as IPOPT [4]);

Use a generalized Newton method to project onto the feasible set (that is, apply Newton's method with the pseudo-inverse of the Jacobian matrix of the constraints);

Use specialized techniques to project onto or into the feasible set, starting with an approximate feasible point or approximate optimizing point.

We present our experience and summarize the advantages and pitfalls of each of these techniques.

References:

[1] R. B. Kearfott, *Rigorous Global Search: Continuous Problems* (Nonconvex optimization and its applications, Vol. 13), Kluwer Academic Publishers, Norwell, MA, USA, and Dordrecht, The Netherlands, 1996.

[2] R. B. Kearfott, On proving existence of feasible points in equality constrained optimization problems, *Math. Program.*, 83 (1998), No. 1, pp. 89–100.

[3] J. Ninin, *Optimisation Globale Basée sur l'Analyse d'Intervalles: Relaxations Affines et Techniques d'Accélération.* Ph.D. dissertation, Université de Toulouse, Toulouse, France, December 2010.

[4] A. Wächter, `https://projects.coin-or.org/Ipopt` (homepage of IPOPT).

Bounding optimal value function in linear programming under interval uncertainty

Oleg V. Khamisov

Institute of Energy Systems SD RAS
130, Lermontov str.
644033 Irkutsk, Russia
khamisov@isem.sei.irk.ru

Keywords: parametric linear programming, optimal value function, convex and concave support functions

We consider the optimal value function of parametric linear programming problem

$$f(c, A, b) = \min\{c^T x : Ax \le b,\ \underline{x} \le x \le \overline{x}\}$$

where $c, \underline{x}, \overline{x} \in \mathbb{R}^n$, A is $m \times n$ matrix, $b \in \mathbb{R}^m$. We assume that coefficients of c, A and b vary within the prescribed intervals

$$\underline{c}_j \le c_j \le \overline{c}_j, \quad j = 1, \ldots, n,$$
$$\underline{a}_{ij} \le a_{ij} \le \overline{a}_{ij}, \quad i = 1, \ldots, m,\ j = 1, \ldots, n,$$
$$\underline{b}_i \le b_i \le \overline{b}_i, \quad i = 1 \ldots, m,$$

$\underline{x}$ and $\overline{x}$ are fixed. Optimal value function $f(c, A, b)$ is in general nonsmooth and nonconvex. The problem is to find bounds $\underline{f}$ and $\overline{f}$ such that

$$\underline{f} \le f(c, A, b) \le \overline{f}.$$

To do this we consider auxiliary problems of minimizing and maximizing $f(c, A, b)$. A branch and bound type global optimization approach is suggested. It is based on concepts of convex and concave support functions [1]. Illustrative numerical examples are presented.

References:

[1] O.V. Khamisov, On application of convex and concave support functions in nonconvex problems, *Manuscript of Institute of Operations Research, University of Zurich*, (1998), 16 p.

An environment for verified modeling and simulation of solid oxide fuel cells

Stefan Kiel[1], Ekaterina Auer[1], and Andreas Rauh[2]

[1]Faculty of Engineering, INKO
University of Duisburg-Essen
D-47048 Duisburg, Germany
{kiel, auer}@inf.uni-due.de

[2]Chair of Mechatronics
University of Rostock
D-18059 Rostock, Germany
andreas.rauh@uni-rostock.de

Keywords: global optimization, parallelization, GPU, SOFC, UniVerMeC

A major goal of a current joint project between the Universities of Rostock and Duisburg-Essen is to design and develop robust and accurate control strategies for solid oxide fuel cells (SOFCs). For this purpose, system models based on ordinary differential equations (ODEs) are being developed [2]. Unlike most state-of-the-art models, they can be used to devise control laws for SOFCs which are valid not only for fixed but also for nonstationary operating points. To allow users to employ the new models and techniques easily in combination with different verified tools, we implement the environment VeriCell. It features an intuitive graphic interface for construction of SOFC models from predefined building blocks and is based on the framework UniVerMeC [1] which provides a unified access to various verified arithmetics and algorithms. New SOFC component models can be added to VeriCell as they are being developed, for which purpose a plug-in based interface is adopted.

In this talk, we present the environment with the focus on efficient implementation of verified optimization routines for parameter identification in SOFC systems. The task is to minimize a quadratic cost function which contains the solution to the initial value problem (IVP) for the above-mentioned ODEs as one of its constituent parts. At the moment, the exact solution to the IVP is approximated by the explicit Euler method [3]. The cost function is complex in practice since it is composed of many summands (the number of which is proportional to the number of measurements) and is strongly influenced by cancelation.

The ODE-based model takes into account preheated air and fuel gas supplied to the SOFC system as well as the corresponding reaction enthalpies. The parameters of interest describe the thermal resistances of the stack materials, the dependency of the heat capacities on the temperature, and the heat produced during the exothermic electrochemical reactions in each individual fuel cell.

Important aspects in solving this task are to increase the model accuracy and to reduce computing times. In the first case, the use of verified IVP solvers such as VNODE-LP instead of the Euler approximation is necessary. In the second case, the employment of the GPU might be profitable, along with the ordinary multi-kernel parallelization. In this talk, we show what steps are necessary to be able to identify parameters of SOFC models of different dimensions using parallelization techniques and the GPU, highlighting in the latter case the questions of accurate implementation, efficient memory use, and correct choice of the working precision. These issues are demonstrated on examples modeled and simulated in VeriCell, which gives an overview of its main features.

References

[1] E. Dyllong and S. Kiel, A comparison of verified distance computation between implicit objects using different arithmetics for range enclosure, *Computing*, 2011.

[2] A. Rauh, T. Dötschel, and H. Aschemann, Experimental parameter identification for a control-oriented model of the thermal behavior of high-temperature fuel cells, In *CD-Proc. of IEEE Intl. Conference MMAR 2011*, Miedzyzdroje, Poland, 2011.

[3] A. Rauh, T. Dötschel, E. Auer and H. Aschemann, Interval methods for control-oriented modeling of the thermal behavior of high-temperature fuel cell stacks, In *Proc. of SysID 2012* (accepted).

Use of Grothendieck's inequality in interval computations: quadratic terms are estimated accurately modulo a constant factor

Olga Kosheleva and Vladik Kreinovich

University of Texas at El Paso, El Paso, TX 79968, USA
olgak@utep.edu, vladik@utep.edu

Keywords: enclosure methods, Grothendieck inequality, feasible algorithms

In interval computations, one of the most widely used methods of efficiently computing an enclosure $\boldsymbol{Y}$ the range $\boldsymbol{y} = f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ of a given function $f(x_1, \ldots, x_n)$ on a given box $\boldsymbol{x} = \boldsymbol{x}_1 \times \ldots \times \boldsymbol{x}_n$ is the Mean Value (MV) method: $\boldsymbol{Y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n) + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(\boldsymbol{x}) \cdot [-\Delta_i, \Delta_i]$, where $\widetilde{x}_i$ is a midpoint of the i-th interval, Δ_i is its radius, and the ranges of the derivatives $f_{,i} \stackrel{\text{def}}{=} \frac{\partial f}{\partial x_i}$ can be estimated, e.g., by using straightforward interval computations; see, e.g., [5]. This method has excess width $O(\Delta^2)$, where $\Delta \stackrel{\text{def}}{=} \max \Delta_i$.

Can we come up with more accurate enclosures? We cannot get too drastic an improvement, since even for quadratic functions $f(x_1 \ldots, x_n)$, computing the interval range is NP-hard (see, e.g., [4,7]) – and therefore (unless P=NP), a feasible algorithm with excess width $O(\Delta^{2+\varepsilon})$ is impossible. What we can do is try to decrease the overestimation of the quadratic term. It turns out that such a possibility follows from an inequality proven by A. Grothendieck in 1953 [2].

Specifically, the MV method is based on the 1st order Mean Value Theorem (MVT): $f(\widetilde{x} + \Delta x) = f(\widetilde{x}) + \sum f_{,i}(\widetilde{x} + \eta) \cdot \Delta x_i$ for some $\eta_i \in [-\Delta_i, \Delta_i]$ [3]. Instead, we propose to estimate the range by adding estimates for ranges of linear, quadratic, and cubic terms in the 3rd order MVT: $f(\widetilde{x} + \Delta x) = f(\widetilde{x}) + \sum f_{,i}(\widetilde{x}) \cdot \Delta x_i + \sum f_{,ij}(\widetilde{x}) \cdot \Delta x_i \cdot \Delta x_j + \sum f_{,ijk}(\widetilde{x} + \eta) \cdot \Delta_i \cdot \Delta_j \cdot \Delta_k$. The range of the cubic term is estimated via straightforward interval computations; the resulting estimate is of order $O(\Delta^3)$. The range of the linear term $f(\widetilde{x}) + \sum f_{,i}(\widetilde{x}) \cdot \Delta x_i$ can be explicitly described as $[\widetilde{y} - \Delta, \widetilde{y} + \Delta]$, where $\widetilde{y} \stackrel{\text{def}}{=} f(\widetilde{x})$ and $\Delta = \sum |f_{,i}(\widetilde{x})| \cdot \Delta_i$. So, the remaining problem is: how accurately can we find the range $[-Q, Q]$

of the quadratic term $\sum_{i,j=1}^{n} a_{ij} \cdot \Delta x_i \cdot \Delta x_j$ (where $a_{ij} \stackrel{\text{def}}{=} f_{,ij}(\widetilde{x})$), on the box $[-\Delta_1, \Delta_1] \times \ldots \times [-\Delta_n, \Delta_n]$.

By re-scaling, we conclude that Q is equal to the maximum of the function $B(z) \stackrel{\text{def}}{=} \sum_{i,j=1}^{n} b_{ij} \cdot z_i \cdot z_j$ (where $b_{ij} \stackrel{\text{def}}{=} a_{ij} \cdot \Delta_i \cdot \Delta_j$), over values $z_i \in [-1, 1]$. Grothendieck's inequality enables us to estimate the maximum Q' of the related bilinear function $b(z,t) \stackrel{\text{def}}{=} \sum_{i,j=1}^{n} b_{ij} \cdot z_i \cdot t_j$ when $z_i, t_j \in \{-1, 1\}$: namely, we can feasibly compute Q'' for which $K_G^{-1} \cdot Q'' \le Q' \le Q''$, where $K_G \in [1, 1.782]$ (see, e.g., [1,6]). One can easily see that Q' is equal to the maximum of $b(z,t)$ when $z_i, t_j \in [-1, 1]$. Since $B(z) = b(z,z)$, we have, $Q \le Q'$; on the other hand, since $b(z,t) = B((z+t)/2) - B((z-t)/2)$, we have $Q' \le 2Q$. Thus, $Q'/2 \le Q \le Q'$ and so, $\dfrac{Q''}{2K_G} \le Q \le Q''$.

Hence, by computing Q'', we can feasibly estimate the quadratic term Q accurately modulo a small constant factor $2K_G \le 3.6$.

References:

[1] N. Alon, A. Naor, Approximating the cut-norm via Grothendieck's inequality, *SIAM J. Comp.*, 35 (2006), No. 4, pp. 787–803.

[2] A. Grothendieck, Résumé de la théorie métrique des produits tensoriels topologiques, *Boll. Soc. Mat. São-Paulo*, 8 (1953), pp. 1–79.

[3] O. Kosheleva, How to explain usefulness of different results when teaching calculus: example of the Mean Value Theorem, *J. Uncertain Systems*, 7 (2013), to appear.

[4] V. Kreinovich, A. Lakeyev, J. Rohn, P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1998.

[5] R.E. Moore, R.B. Kearfott, M.J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[6] G. Pisier, Grothendieck's theorem, past and present, *Bulletin of the American Mathematical Society*, 49 (2012), No. 2, pp. 237–323.

[7] S.A. Vavasis, *Nonlinear Optimization: Complexity Issues*, Oxford University Press, N.Y., 1991.

On boundedness and unboundedness of polyhedral estimates for reachable sets of linear systems

Elena K. Kostousova

Institute of Mathematics and Mechanics,
Ural Branch of the Russian Academy of Sciences,
16, S. Kovalevskaja street, 620990, Ekaterinburg, Russia
kek@imm.uran.ru

Keywords: interval analysis, linear differential systems, reachable sets, polyhedral estimates, parallelepipeds

The problem of constructing trajectory tubes (in particular, reachable tubes which describe a dynamic of reachable sets) is an essential theme in control theory [1]. Since the practical construction of these tubes may be cumbersome, the different numerical methods are devised for this cause. Among them the techniques were developed for estimating reachable sets by domains of some fixed shape such as ellipsoids, parallelepipeds, zonotopes. In particular, box-valued estimates may be constructed by means of interval calculations. But such estimates can turn out to be rather conservative and even unbounded due to the wrapping effect [2,3] known in interval analysis. To make possible exact representations of reach sets A.B. Kurzhanski proposed to use families of fixed shape estimates [1,4] and, moreover, families of so called tight estimates [4]. We expanded this approach to polyhedral (parallelepiped-valued) estimates. The family $\mathfrak{P}$ of outer polyhedral estimates of reachable sets for linear differential systems with parallelepiped-valued uncertainties in initial states and additive inputs may be introduced [5]. These estimates are determined by a given dynamics of orientation matrices $P(t) \in \mathbb{R}^{n\times n}$ (this function is the parameter of the family) and by corresponding parameterized differential equations which describe the dynamics of centers and "semi-axis" values of parallelepipeds. Considering different types of the orientation matrix dynamics $P(\cdot)$ we obtain several subfamilies $\mathfrak{P}^i \in \mathfrak{P}$ of the estimates with different properties, in particular, subfamilies $\mathfrak{P}^3$ and $\mathfrak{P}^1$ of tight and touching [6] (tight in n specific directions) estimates (both ensure the exact representations of reachable sets through intersections of their units). Box-valued estimates may be attributed to the subfamily $\mathfrak{P}^2$ of estimates with constant orientation matrices. In fact, the orientation matrix $V = P(0)$ at the initial time is the parameter of all mentioned subfamilies $\mathfrak{P}^i$.

The paper presents our recent results on studying the properties of boundedness and unboundedness at the infinite time interval of outer polyhedral estimates for reachable sets of systems with stable matrices. The mentioned properties are determined by interaction of three factors: the matrix V, the real Jordan matrix for system's matrix and the properties of the bounding sets for uncertainties. The results of this interaction are different for different subfamilies $\mathfrak{P}^i$. We formulate the corresponding criteria for boundedness / unboundedness of estimates from $\mathfrak{P}^1$ and $\mathfrak{P}^2$ (see [7] for some of them), including characterizing the possible degree of the growth of estimates in terms of the exponents. Then we present new results concerning the subfamily $\mathfrak{P}^3$ of tight estimates. In particular, it turns out that for two-dimensional systems all estimates from $\mathfrak{P}^3$ are bounded and in addition they turn out to be orthohogonal parallelepipeds. This is unlike to two other cases mentioned above because there are two-dimensional systems for which all estimates from $\mathfrak{P}^1$ and $\mathfrak{P}^2$ are unbounded (these systems are of different kinds for $\mathfrak{P}^1$ and $\mathfrak{P}^2$). The results of numerical simulations are presented.

The work was supported by the Program of the Presidium of the Russian Academy of Sciences "Dynamic Systems and Control Theory" under support of the Ural Branch of RAS (project 12-P-1-1019), by the State Program for Support of Leading Scientific Schools of Russian Federation (grant 2239.2012.1) and by the Russian Foundation for Basic Research (grant 12-01-00043).

References:

[1] A.B. Kurzhanski, I. Valyi, *Ellipsoidal Calculus for Estimation and Control*, Birkhäuser, Boston, 1997.

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

[3] A.N. Gorban, Yu.I. Shokin, V.I. Verbitskii, Simultaneously dissipative operators and the infinitesimal wrapping effect in interval spaces, *Vychisl. Tekhnol.*, 2 (1997), No. 4, pp. 16–48.

[4] A.B. Kurzhanski, P. Varaiya, On ellipsoidal techniques for reachability analysis, Parts I, II, *Optim. Methods Softw.*, 17 (2002), No. 2, pp. 177–237.

[5] E.K. Kostousova, Outer polyhedral estimates for attainability sets of systems with bilinear uncertainty, *Prikl. Mat. Mekh.*, 66 (2002), No. 4, pp. 559–571 (Russian); translation in *J. Appl. Math. Mech.*, 66 (2002), No. 4, pp. 547–558.

[6] E.K. Kostousova, State estimation for dynamic systems via parallelotopes: optimization and parallel computations, *Optim. Methods Softw.*, 9 (1998), No. 4, pp. 269–306.

[7] E.K. Kostousova, On the boundedness of outer polyhedral estimates for reachable sets of linear systems. *Zh. Vychisl. Mat. Mat. Fiz.*, 48 (2008), No. 6, pp. 974–989 (Russian); translation in *Comput. Math. Math. Phys.*, 48 (2008), No. 6, pp. 918–932.

Arbitrary precision real interval and complex interval computations

Walter Krämer

Department of Mathematics and Computer Science
University of Wuppertal
Wuppertal, Germany
kraemer@math.uni-wuppertal.de

Keywords: arbitrary precision, complex interval arithmetic, complex interval functions, extended interval Newton method

The design and development of two new software libraries for arbitrary precision real interval and complex interval computations are discussed. These libraries provide a comprehensive set of basic operations and mathematical functions. Their comfortable usage (due to C++ operator and function overloading) is demonstrated on challenging examples like an extended interval Newton method to automatically bound all zeros of a given function. The derivatives are computed via algorithmic differentiation. The libraries are open source and freely available on the net.

References:

[1] W. Krämer and F. Blomquist, Arbitrary precision complex interval computations in C-XSC. In: *Parallel Processing and Applied Mathematics* (Roman Wyrzykowski , Jack Dongarra , Konrad Karczewski und Jerzy Wasniewski, Eds.), Springer Verlag, 2012. *Lecture Notes in Computer Science*, Volume 7204, pp. 457–466.

Decision making under interval uncertainty

Vladik Kreinovich

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

Keywords: interval uncertainty, decision making, utility theory, p-boxes, modal intervals, symmetries, control

To make a decision, we must:

- find out the user's preference, and
- help the user select an alternative which is the best – according to these preferences.

A general way to describe user preferences is via the notion of *utility* (see, e.g., [7]): we select a very bad alternative A_0 and a very good alternative A_1; utility $u(A)$ of an alternative A if then defined as the probability p for which A is equivalent to the lottery in which we get A_1 with probability p, and A_0 otherwise. One can prove that utility is determined uniquely modulo linear re-scaling (corresponding to different choices of A_0 and A_1), and that the utility of a decision with probabilistic consequences is equal to the expected utility of these consequences.

Once the utility function $u(d)$ is elicited, we select the decision $d_{\rm opt}$ with the largest utility $u(d)$. Interval techniques can help in finding the optimizing decision; see, e.g., [4].

Often, we do not know the exact probability distribution, so we need to extract, from the sample, the characteristics of a distribution which are most appropriate for decision making. We show that, under reasonable assumptions, we should select moments and cumulative distribution function (cdf). Based on a finite sample, we can only find bounds on these characteristics, so we need to deal with bounds (intervals) on moments [6] and bounds on cdf [1] (a.k.a. p-boxes).

Once we know intervals $[\underline{u}(d), \overline{u}(d)]$ of possible values of utility, which decision shall we select? We can simply select a decision d_0 which *may* be optimal,

i.e., for which $\overline{u}(d_0) \geq \max\limits_{d} \underline{u}(d)$, but there are usually many such decisions; which of them should be select? It is reasonable to assume that this selection should not depend on linear re-scaling of utility; under this assumption, we get Hurwicz optimism-pessimism criterion $\alpha \cdot \overline{u}(d) + (a - \alpha) \cdot \underline{u}(d) \to \max$ [7]. The next question is how to select α: interestingly, e.g., too optimistic values ($\alpha > 0.5$) do not lead to good decisions.

In some situations, it is difficult to elicit even interval-valued utilities. In many such situations, there are reasonable symmetries which can be used to make a decision; see, e.g., [5]. We show how this method works on the example of selecting a location for a meteorological tower [3].

Finally, while optimization problems are ubiquitous, sometimes, we need to go beyond optimization: e.g., we need to make sure that the system is *controllable* for all disturbances within a given range. In such problems, modal intervals [2] naturally appear. In more complex situations, we need to go beyond modal intervals, to more general Shary's classes.

References:

[1] S.Ferson, V.Kreinovich, J.Hajagos, W.Oberkampf, L.Ginzburg, *Experimental Uncertainty Estimation and Statistics for Data Having Interval Uncertainty*, Sandia National Laboratories, 2007, Publ. 2007-0939.

[2] E. Gardeñes et al., Modal intervals, *Reliable Computing*, 7 (2001), pp. 77–111.

[3] A. Jaimes, C. Tweedie, V. Kreinovich, M. Ceberio, Scale-invariant approach to multi-criterion optimization under uncertainty, with applications to optimal sensor placement, in particular, to sensor placement in environmental research, *International Journal of Reliability and Safety*, 6 (2012), No. 1–3, pp. 188–203.

[4] R.E. Moore, R.B. Kearfott, M.J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[5] H.T. Nguyen, V. Kreinovich, *Applications of Continuous Mathematics to Computer Science*, Kluwer, Dordrecht, 1997.

[6] H.T. Nguyen, V. Kreinovich, B. Wu, G. Xiang, *Computing Statistics under Interval and Fuzzy Uncertainty*, Springer Verlag, 2012.

[7] H. Raiffa, *Decision Analysis*, McGraw-Hill, Columbus, Ohio, 1997.

Excluding regions using Sobol sequences in an interval branch-and-bound method

Bartłomiej Jacek Kubica

Institute of Control and Computation Engineering (of WUT)
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
bkubica@elka.pw.edu.pl

Keywords: interval methods, exclusion regions, Sobol sequences, underdetermined systems, nonlinear equations

Interval branch-and-prune (b&p) methods (also called by a more general term, branch-and bound [2]) are commonly used to solve systems of nonlinear equations and several other problems. Their main drawbacks (as for most combinatorial approaches) are high computational cost and high memory space requirements, in the pessimistic case. This computational burden can be avoided by choosing proper heuristics and policies to adapt the process for a specific problem. Hence, any improvements or accelerations to the process are very worthwhile.

The paper is going to consider a preprocessing step of the b&p method, in which some infeasible regions are removed from further search. For the system of equations $f(x) = 0$, $x \in \boldsymbol{x} \subseteq \mathbb{R}^n$ we can remove any box $\boldsymbol{z} \subseteq \boldsymbol{x}$ such that $f_i(z) > \varepsilon$ or $f_i(z) < -\varepsilon$ for all $z \in \boldsymbol{z}$ (i.e., $\mathsf{f}_i(\boldsymbol{z}) \subseteq [\varepsilon, +\infty)$ or $\mathsf{f}_i(\boldsymbol{z}) \subseteq (-\infty, -\varepsilon]$) and an arbitrary equation i and some $\varepsilon > 0$.

Tools that are used to solve the above problem include:

- simple computations of interval extension of functions,
- solving the interval tolerance problem (see, e.g., [6]) for the linearized problem,
- applying ε-inflation [2] to enlarge the infeasible box, being removed.

Such a procedure – simple and well-known – does not specify how to choose initial regions for removal, which is crucial for efficiency. These regions can be constructed around some "seeds" scattered around the problem domain. The "seeds" can be chosen randomly, but a better approach is to use a deterministic low-discrepancy sequence [1], e.g., the Sobol sequence [8], also called the LP_τ sequence. Points of this sequence are distributed in a very regular way over the

search domain and the method remains deterministic (hence easy to investigate). Also, there are efficient algorithms to generate such sequences [8].

According to the author's observations, it seems most efficient to choose n seeds, i.e., as many as the number of variables, independently of the number of equations.

The considered approach seems particularly useful for underdetermined systems, where the solutions are not isolated points, but belong to a continuous set. For such systems, we cannot verify the uniqueness of a solution (as, e.g., in [7]) and – on the other hand – deleting infeasible regions may result in boxes in which segments of the solution manifold are easy to verify.

Thanks to this approach, speedups of the rate 30-50% are obtained, at least, for some problems. The paper is going to present a few variants of the method and its cooperation with the equations systems solver developed in [3]–[5]. Computational experiments for examples of underdetermined and well-determined systems will be considered. Parallelization of the method will also be investigated.

References:

[1] M. Drmota, R.F. Tichy, *Sequences, Discrepancies and Applications*, Springer-Verlag, Berlin, Heidelberg, 1997.

[2] R.B. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer Academic Publishers, Dordrecht, 1996.

[3] B.J. Kubica, Interval methods for solving underdetermined nonlinear equations systems, *Reliable Computing*, 15 (2011), pp. 207–217.

[4] B.J. Kubica, Intel TBB as a tool for parallelization of an interval solver of nonlinear equations systems, *Tech. rep.* no 09-02, ICCE WUT, 2009.

[5] B.J. Kubica, Tuning the multithreaded interval method for solving underdetermined systems of nonlinear equations, *PPAM 2011 Proceedings*, *LNCS*, accepted.

[6] S.P. Shary, *Finite-dimensional Interval Analysis*, XYZ, 2010 (in Russian).

[7] H. Schichl, A. Neumaier, Exclusion regions for systems of equations, *SIAM Journal of Numerical Analysis*, 42 (2004), pp. 383–408.

[8] Sobol sequence generator, `http://web.maths.unsw.edu.au/~fkuo/sobol/`.

Interval methods for computing various refinements of Nash equilibria

Bartłomiej Jacek Kubica and Adam Woźniak

Institute of Control and Computation Engineering (of WUT)
ul. Nowowiejska 15/19,
00-665 Warsaw, Poland
bkubica@elka.pw.edu.pl

Keywords: interval methods, game theory, solution concepts, strong Nash equilibria

The game theory has numerous applications in many branches of theoretical and applied science. One of the basic solution concepts for non-cooperative games is the idea of a Nash equilibrium [5]. It can be defined as a situation (an assignment of strategies to all players), when it is not beneficial to any of the players to change their strategy unless others will do so. Such points, however, have several drawbacks – both theoretical (rather strong assumptions about the players' knowledge and rationality) and practical (they can be Pareto-inefficient).

Hence, several "refinements" to the notion have been introduced, including epsilon-equilibrium, strong Nash equilibrium, manipulated Nash equilibrium and many others.

On the other hand, computing any kind of these solutions can be a hard problem. In particular, very few computational methods exist for continuous games.

In our previous paper [3] we proposed an interval algorithm to compute Nash equilibria of a continuous non-cooperative game. In [4] it was shown that the interval branch-and-bound (b&b) method can be used to compute the enclosure of any set of points that fulfill a given condition, described, by some kind of a predicate formula (see also [2]). But, as all refinements of the Nash equilibrium can be described this way, computing all of them should be possible, using a version of the b&b framework.

The paper is going to investigate interval algorithms for computing other solution concepts for continuous games. Data structures and parallelization issues will be considered. In particular, the concept of strong Nash equilibrium [1] and some its modifications are going to be analyzed.

As an example, we consider a simple and interesting pursuit game, developed by Steinhaus [6,7]. Some variants and modifications of the game (including an increased number of players) are going to be presented, too.

References:

[1] R.J. Aumann, S. Hart (eds.) *Handbook of Game Theory with Economic Applications*, Vol. 1, North-Holland, 1992, Chapter 4.

[2] V. Kreinovich, B.J. Kubica, From computing sets of optima, Pareto sets and sets of Nash equilibria to general decision-related set computations, *Journal of Universal Computer Science*, 16 (2010), pp. 2657–2685.

[3] B.J. Kubica, A. Woźniak, An interval method for seeking the Nash equilibria of non-cooperative games, *LNCS*, 6068 (2010), pp. 446–455.

[4] B.J. Kubica, A class of problems that can be solved using interval algorithms, *SCAN 2010 Proceedings*, *Computing*, 94 (2012), pp. 271–280.

[5] J.F. Nash, Equilibrium points in n-person games, *Proceedings of National Association of Science*, 36 (1950), pp. 48-49.

[6] H. Steinhaus, Definitions for a theory of games and pursuit, *Naval Research Logistics Quarterly*, 7 (1960), pp. 105–107.

[7] H. Steinhaus, O grach (swobodnie), [Games, an informal talk], *Studia Filozoficzne*, 5 (1969), pp. 3–13 (in Polish).

Interval approach to identification of parameters of experimental process model

Sergey I. Kumkov[1] and Yuliya V. Mikushina[2]

[1]Institute of Mathematics and Mechanics UrB RAS
16, S. Kovalevskaya str., 620219 Ekaterinburg, Russia
[2]Institute of Organic Synthesis UrB RAS
20, S. Kovalevskaya str., 620990 Ekaterinburg, Russia
kumkov@imm.uran.ru

Keywords: interval identification, parameters, model, experimental process

The work considers an application of the general interval analysis methods (e.g., [1]) to a special practical problem of parameters identification of a real experimental chemical process [2]. In the process, concentration $S(t)$ of peroxide H_2O_2 is measured versus the time t of the decomposition catalytic reaction for various nano-catalysts. Two possible models of the process are investigated.

Model 1. The experimental process is described by the function $S(t, C, \alpha, BG) = C\exp(\alpha t) + BG$. The vector of parameters to be identified is three-dimensional: $C > 0$ is the initial value of concentration; $\alpha < 0$ is the activity coefficient of the first approximation model; $BG > 0$ is a background value.

Model 2. Here, the describing function is $S(t, C, \alpha, \beta, BG) = C\exp(\alpha t + \beta t^2) + BG$, where, in comparison with Model 1, the coefficient β of activity of the second approximation is introduced ($\beta < 0$). So, the vector of parameters to be identified is four-dimensional.

The following input data are given [2]: the sample of noised measurements $\{t_k, S_k = S(t_k)\}$, $k = 1, 4$; it is assumed that values t_k are known exactly, but measurements S_k are noised with the total additive errors bounded by modulus as $|e_k| \le e_{\max} = 0.035$. The experiments have been performed very carefully, with very clear reactants, and small actual measuring errors. As a consequence, the measurements were not distorted by jerks and there are no outliers in the sample. The results of measuring the process for three various catalysts (1 – 3) are given in Fig.1. To show the trends of the processes in Experiments 2 and 3, the samples are approximated (black curves) by the standard regression method

using Model 1. For the Experiment 1, the uncertainty intervals H_k of the length $2e_{\max}$ are drawn around each measurement: $H_k = [S_k - e_{\max}, S_k + e_{\max}]$.

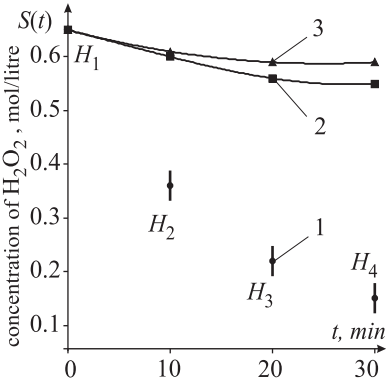

Figure 1. Decomposition of H_2O_2 on bronzes (kinetic curves)

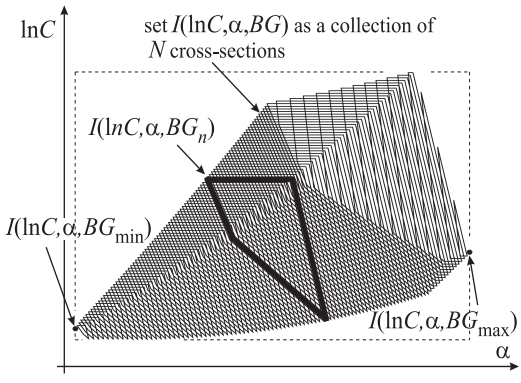

Figure 2. The form of the solution: informational set $I(\ln C,\alpha,BG)$ of parameters for Model 1

The problem of identification is formulated as follows: *it is necessary to identify (to construct) the set of admissible values of model parameters consistent with the given input data.*

We consider the main idea and procedures of the elaborated algorithms for Model 1. The following procedures are performed. By shifting the background parameter BG to the left-hand side and by standard logarithmic operation, the initial function is transformed to the following function $y = \ln\big(S(t) - BG\big)$ with linear dependence on the parameter α and a new parameter $\ln C$: $y(t, \ln C, \alpha) = \ln\big(S(t) - BG\big) = \ln C + \alpha t$; note that the central term in this expression will be an interval for each t_k. Some reasonable *a priori* interval of the parameter BG is introduced with a grid $\{BG_m, m = 1, M\}$. Application of algorithms [3] to constructing the informational set $I(\ln C, \alpha, BG_m)$ for each node BG_m (together with adjusting the position of the grid, its step, and number of nodes) gives the whole desired informational set $I(\ln C, \alpha, BG)$ as a collection (Fig.2) of its cross-sections $\big\{I(\ln C, \alpha, BG_n)\big\}, n = 1, N$ over all *admissible* nodes N of the adjusted grid, i.e., nodes, for which the cross-section is not empty. For Model 2, the algorithms are similarly repeated for two grids in parameters BG and β. Note that the elaborated approach is significantly faster and more accurate than ones based on application of parallelotopes [1].

References:

[1] L. Jaulin, M. Kieffer, O. Didrit, E. Walter, *Applied Interval Analysys*, Springer, London, 2001.

[2] L.A. Petrov, Yu.V. Mikushina, et. al., Catalytic activity of oxide polycristal and nano-size tungsten bronzes produced by electrolysis of molten salts, *Izv. Acad. Nauk, ser. Chemical*, 2011, No. 10, pp. 1951–1954.

[3] S.I. Kumkov, Procession of experimental data on ionic conductivity of molten electrolyte by the interval analysis methods, *Rasplavy*, 2010, No. 3, pp. 86–96.

The libieee754 compliance library for the IEEE 754-2008 standard

Olga Kupriianova and Christoph Lauter

UPMC - LIP6 - Équipe PEQUAN
4, place Jussieu, F - 75252 Paris Cedex 05, France
olga.kupriianova@lip6.fr, christoph.lauter@lip6.fr

Keywords: reliable floating-point arithmetic, correct rounding, heterogeneous floating-point operations, IEEE 754-2008

In 1985, the IEEE 754 Standard for Binary Floating-Point Arithmetic [2] provided concise means to achieve portability and provability of Floating-Point (FP) programs. The high level of achieved reliability was the key to its widespread adoption.

In 2008, a revised version, the IEEE 754-2008 Standard for Floating-Point Arithmetic [1], was published. This revision reinforced the reproducibility aspects of the standard and added a few new operations and concepts, such as decimal arithmetic, heterogeneous operations or fused-multiply-and-add (FMA).

As of today, the IEEE 754-2008 standard has already been accepted as the preferred FP Arithmetic system. For instance, the P1788 working group* recognized it as a base for standardized Interval Arithmetic.

However, IEEE 754-2008 is currently not completely supported by Programming Languages like C99, nor by Operating Systems, such as GNU/Linux. In C99, some operations are missing and some are only partly compliant with the standard. For instance, decimal-to-binary conversion in *scanf* commonly implements correct rounding only for *round-to-nearest* mode or FMA might incorrectly round twice. Complete IEEE 754-2008 compliance is available only on Intel-compatible processors, through a closed-source library provided by Intel†.

For Open Source IEEE 754-2008 compliance, this work proposes the libieee754 library. The library implements all the 354 operations IEEE 754-2008 mandates for Binary FP Arithmetic in both binary32 and binary64 formats. While the library is reasonably fast, speed was not the main purpose but 100% standard compliance. All operations support all rounding modes and set

*cf. http://grouper.ieee.org/groups/1788/

†cf. http://software.intel.com/sites/products/documentation/hpc/composerxe/en-us/2011Update/cpp/lin/cref_cls/common/cppref_libbfp754_ovrvw.htm

all flags as required by IEEE 754-2008. The library is reentrant as there is no global state other than the global state foreseen by the standard.

The functions in `libieee754` performing correctly rounded conversion from arbitrary length decimal character sequences to the binary FP formats should be highlighted. They support all rounding modes and does not perform any dynamic memory allocation. While the algorithms found in the literature [3] resume to one-step correctly rounded decimal-to-binary conversion with unknown memory consumption limits, the novel algorithm implemented in `libieee754` performs decimal-to-binary conversion indirectly in three steps: first, convert from decimal to binary and round to a floating-point midpoint, second, exactly convert the binary midpoint back to decimal and third, round correctly. This allows memory consumption to be known beforehand, avoiding any dynamic memory allocation.

The algorithm for decimal-to-binary conversion set aside, the most important difficulty when designing `libieee754` was with the rounding-mode, which cannot be queried by the library code, and with the IEEE 754 flags. Each FP operation needed hence to be chosen with 4 rounding modes and possible side-effects on flags in mind.

The `libieee754` library was completely proven on paper and extensively tested. The proofs are available for reference. Future work is supposed to bring formal proofs, in a system such as Coq [4].

In the future, `libieee754` is supposed to be extended with respect to the binary128 format, decimal FP Arithmetic and the optional parts of the IEEE 754-2008 Standard. Additionally, the library's code base should be extended to allow for compilation on systems where no hardware floating-point support is available and where a complete emulation of all floating-point operations using integer instructions will be needed.

References:

[1] *IEEE Standard for Floating-Point Arithmetic, IEEE Std 754™-2008*, IEEE, New York, NY, USA, 2008.

[2] *IEEE Standard for Binary Floating-Point Arithmetic, IEEE Std 754-1985*, IEEE, New York, NY, USA, 1985.

[3] M. Hack, On intermediate precision required for correctly-rounding decimal-to-binary floating-point conversion, *Proc. of RNC 6*, 2004, pp.113–134.

[4] G. Melquiond, Floating-point arithmetic in the Coq system, *Proc. of RNC 8*, 2008, pp. 93–102.

Monotone and convex interpolation by weighted quadratic splines

Boris I. Kvasov

Institute of Computational Technologies SD RAS
6, Lavrentiev ave.
630090 Novosibirsk, Russia
kvasov@ict.nsc.ru

Keywords: shape-preserving interpolation, weighted C^1 quadratic splines, adaptive choice of shape control parameters, weighted B-splines and control point approximation

We are interested in numerically fitting a curve through a given finite set of points $P_i = (x_i, f_i)$, $i = 0, \ldots, N+1$, in the plane, with $a < x_0 < x_1 < \cdots < x_{N+1} = b$. These points can be thought of as coming from the graph of some function f defined on $[a, b]$. We are particularly interested in algorithms which preserve local monotonicity and convexity of the data (or function). Here, we shall focus only on those algorithms which use C^1 piecewise quadratic interpolants.

Monotone and convex local C^1 quadratic splines with perhaps one additional knot in each subinterval between data points were considered by L.L. Schumaker [7] (see also [2,5], and references therein). In these interactive algorithms the location of additional knots allows the user to adjust spline to the data and to take full advantage of the flexibility which quadratic splines permit. Some improvements of these algorithms were suggested in [1,4]. Very similar algorithms were also obtained in [9].

In contrast with the previously published algorithms for shape preserving quadratic splines which rely on local schemes, our algorithms are based on global weighted C^1 quadratic splines. Such splines generalize global quadratic splines introduced by Yu.N. Subbotin [8] and are similar to weighted C^1 cubic splines [6]. We let the additional knots be the midpoints in each subinterval to have actually their optimal location.

While there are many methods available for the solution of the shape-preserving interpolation problem (see a very detailed literature review in [3]), the shape parameters are mainly viewed as an interactive design tool for manipulating shape of a spline curve.

The main challenge of this paper is to present algorithms that select shape control parameters (weights) automatically. We give two such algorithms: one to preserve the data monotonicity and other to retain the data convexity. These algorithms based on the sufficient conditions of monotonicity and convexity for C^1 quadratic splines and adapt the spline curve to the data geometric behavior. The main point, however, is to determine whether the error of approximation remains small under the proposed algorithms. To this end we prove two theorems to estimate error bounds. We show that by using special choice of shape parameters one can rise the order of approximation. We construct also weighted B-splines and consider control point approximation. Recurrence relations for weighted B-splines offer valuable insight into their geometric behavior.

References:

[1] R.A. DeVore, Z. Yan, Error analysis for piecewise quadratic curve fitting algorithms, *Comput. Aided Geom. Des.*, 3 (1986), No. 1, pp. 205–215.

[2] T.A. Foley, Local control of interval tension using weighted splines, *Comput. Aided Geom. Des.*, 3 (1986), No. 1, pp. 281–294.

[3] T.N.T. Goodman, Shape preserving interpolation by curves, In: J. Levesley, I. Anderson, J. Mason (eds.) *Algorithms for Approximation IV*, University of Huddersfield, 2002, pp. 24–35.

[4] M.H. Lam, Monotone and convex quadratic spline interpolation, *Virginia Journal of Science*, 41 (1990), No. 1, pp. 3–13.

[5] D.F. McAllister, J.A. Roulier, Interpolation by convex quadratic splines, *Math. Comput.*, 17 (1980), pp. 238–246.

[6] K. Salkauskas, C^1 splines for interpolation of rapidly varying data, *Rocky Mountain Journal of Mathematics*, 14 (1984), No. 1, pp. 239–250.

[7] L.L. Schumaker, On shape preserving quadratic spline interpolation, *SIAM J. Numer. Anal.*, 20 (1983), No. 4, pp. 854–864.

[8] S.B. Stechkin, Yu.N. Subbotin, *Splines in Computational Mathematics*, Nauka, Moscow, 1976 (in Russian).

[9] V.T. Voronin, Construction of shape preserving splines, *Preprint 404, Computing Center, Siberian Branch of USSR Academy of Sciences*, Novosibirsk, 1982, 27 pp. (in Russian).

On unboundedness of generalized solution sets for interval linear systems

Anatoly V. Lakeyev

Institute of Systems Dynamics and Control Theory SB RAS
134, Lermontov ave., 664033 Irkutsk, Russia
lakeyev@icc.ru

Keywords: interval linear systems, NP-hardness

We consider systems of linear interval equations of the form

$$\boldsymbol{A}x = \boldsymbol{b},$$

where $\boldsymbol{A} = [\underline{A}, \overline{A}]$ is an interval $m \times n$-matrix, $\boldsymbol{b} = [\underline{b}, \overline{b}]$ is an interval m-vector, and $x \in \mathbb{R}^n$. The interval matrix and the interval vector are traditionally understood as the sets

$$\boldsymbol{A} = \{\, A \in \mathbb{R}^{m\times n} \mid \underline{A} \le A \le \overline{A} \,\}, \qquad \boldsymbol{b} = \{\, b \in \mathbb{R}^m \mid \underline{b} \le b \le \overline{b} \,\}$$

(by $\mathbb{R}^{m\times n}$ from now on we denote the set of $m \times n$-matrices). It is also assumed that $\underline{A} \le \overline{A}$, $\underline{b} \le \overline{b}$, and the inequalities between the matrices and the vectors are understood elementwise and coordinatewise, respectively.

Following the papers [1], we suppose that an $m \times n$-matrix $\Lambda = (\lambda_{ij})$, $\lambda_{ij} \in \{-1, 1\}$, $i = \overline{1,m}$, $j = \overline{1,n}$ and an m-vector $\beta = (\beta_1, \ldots, \beta_n)^\top$, $\beta_i \in \{-1, 1\}$, $i = \overline{1,m}$ are given along with the interval $m \times n$-matrix $\boldsymbol{A}$ and the interval m-vector $\boldsymbol{b}$. The matrix $\boldsymbol{A} = (\boldsymbol{a}_{ij})$ is decomposed into the two matrices $\boldsymbol{A}^\exists = (\boldsymbol{a}^\exists_{ij})$ and $\boldsymbol{A}^\forall = (\boldsymbol{a}^\forall_{ij})$ so that

$$\boldsymbol{a}^\exists_{ij} = \begin{cases} \boldsymbol{a}_{ij}, & \text{if } \lambda_{ij} = 1, \\ 0, & \text{if } \lambda_{ij} = -1, \end{cases} \qquad \boldsymbol{a}^\forall_{ij} = \begin{cases} 0, & \text{if } \lambda_{ij} = 1, \\ \boldsymbol{a}_{ij}, & \text{if } \lambda_{ij} = -1. \end{cases}$$

Similarly, let us decompose the vector $\boldsymbol{b} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m)^\top$ into the two vectors

$$\boldsymbol{b}^\exists = (\boldsymbol{b}^\exists_1, \ldots, \boldsymbol{b}^\exists_m)^\top \qquad \text{and} \qquad \boldsymbol{b}^\forall = (\boldsymbol{b}^\forall_1, \ldots, \boldsymbol{b}^\forall_m)^\top$$

such that

$$\boldsymbol{b}^\exists_i = \begin{cases} \boldsymbol{b}_i, & \text{if } \beta_i = 1, \\ 0, & \text{if } \beta_i = -1, \end{cases} \qquad \boldsymbol{b}^\forall_i = \begin{cases} 0, & \text{if } \beta_i = 1, \\ \boldsymbol{b}_i, & \text{if } \beta_i = -1. \end{cases}$$

It is furthermore obvious that $\boldsymbol{A} = \boldsymbol{A}^{\forall} + \boldsymbol{A}^{\exists}$, $\boldsymbol{b} = \boldsymbol{b}^{\forall} + \boldsymbol{b}^{\exists}$.

Definition (S.P.Shary [1])**.** For a given quantifier matrix Λ and a quantifier vector β, the generalized AE-solution set of the type $\Lambda\beta$ is

$$\Xi_{\Lambda,\beta}(\boldsymbol{A},\boldsymbol{b}) = \big\{\, x \in \mathbb{R}^n \mid (\,\forall A' \in \boldsymbol{A}^{\forall}\,)(\,\forall b' \in \boldsymbol{b}^{\forall}\,) \\ (\,\exists A'' \in \boldsymbol{A}^{\exists}\,)(\,\exists b'' \in \boldsymbol{b}^{\exists}\,)(\,(A'+A'')\,x = b'+b''\,)\,\big\}. \quad (1)$$

The main purpose of our paper is to inquire into the algorithmic complexity of the problem that arises in connection with these sets:

Problem. Find out whether the set (1) is unbounded or not.

In the rest of the paper, for the two $m \times n$-matrices $A = (a_{ij})$ and $B = (b_{ij})$, by $A \circ B$ we will denote their Hadamard product $A \circ B = (a_{ij}b_{ij})$. Using the well-known Oettli-Prager theorem, it is possible to obtain Oettli-Prager-type description of the generalized solution sets.

For any given Λ and β, the equality

$$\Xi_{\Lambda,\beta}(\boldsymbol{A},\boldsymbol{b}) = \{\, x \in \mathbb{R}^n \mid |A_c x - b_c| \le (\Lambda \circ \Delta)|x| + \beta \circ \delta \,\},$$

holds, where $A_c = \frac{1}{2}(\underline{A} + \overline{A})$, $\Delta = \frac{1}{2}(\overline{A} - \underline{A})$, $b_c = \frac{1}{2}(\underline{b} + \overline{b})$, $\delta = \frac{1}{2}(\overline{b} - \underline{b})$. Using this description, we obtain the following

Proposition. The set $\Xi_{\Lambda,\beta}(\boldsymbol{A},\boldsymbol{b})$ is unbounded if and only if for some $y \in Q = \{\, x \in \mathbb{R}^n \mid x_i \in \{-1,1\},\ i = \overline{1,n}\}$ there exists a solution to the following system of linear inequalities (where $T_y = \operatorname{diag}\{y_1,\ldots,y_n\}$)

$$\begin{cases} -(\Lambda \circ \Delta)T_y x - \beta \circ \delta \le A_c x - b_c \le (\Lambda \circ \Delta)T_y x + \beta \circ \delta, \quad T_y x \ge 0, \\ -(\Lambda \circ \Delta)T_y z \le A_c z \le (\Lambda \circ \Delta)T_y z, \quad T_y z \ge 0, \quad \sum_{i=1}^{i=n} y_i z_i \ge 1. \end{cases} \quad (2)$$

Theorem. If the functions Λ, β are easily computable and 1-saturate (the definition can be found in [2]) then the Problem 1 is NP-complete.

In particular, it follows from the theorem that if $P \neq NP$ then there does not exist a criterion of unboundedness of the AE-solution set which is better than checking solvability for 2^n systems of the form (2).

References:

[1] S.P. Shary, A new technique in systems analysis under interval uncertainty and ambiguity, *Reliable Computing*, 8 (2002), No. 5, pp. 321–418.

[2] A.V. Lakeyev, Computational complexity of estimation of generalized solution sets for interval linear systems, *Computation Technologies*, 8 (2003), No. 1, pp. 12–23.

There's no reliable computing without reliable access to rounding modes

Christoph Lauter and Valérie Ménissier-Morain

LIP6 – UPMC, 4 place Jussieu, 75252 Paris Cedex 5, France
Christoph.Lauter@lip6.fr, Valerie.Menissier-Morain@lip6.fr

While approximate answers are accepted for pure Floating-Point Arithmetic (FPA), Interval Arithmetic (IA) is supposed to give reliable results. Indeed IA never lies as it provides lower and upper bounds that provably encompass the true result. Basic IA achieves this enclosure property by taking all Floating-Point (FP) roundings into account, rounding lower bounds down ($\bigtriangledown$) and upper bounds up ($\bigtriangleup$), or inflating the round-to-nearest result by a machine epsilon [6]. E.g. interval addition $[a, b] + [c, d]$ is implemented as $[\bigtriangledown(a + c), \bigtriangleup(b + d)]$.

However, basic IA often cannot be used as such [4,5]. First, each basic IA operation uses both directed rounding modes (RM), hence requiring at least one RM change. As this is an expensive operation on most processors requiring for instance a pipeline flush, it should be avoided as often as possible. Second, basic IA provides the elementary operations such as addition and multiplication only, whereas most modern scientific computing needs high-level operations such as matrix and vector addition and multiplication or linear system solving.

In the world of pure FPA, all these operations are available in fast and highly tuned math libraries. The Intel Math Kernel Library (MKL) [1] is one of the most advanced and widely used libraries for this purpose. In a decade of existence, with a whole team working on it, it has reached significant maturity.

MKL did have high-level IA, particularly linear solvers, between 2005 and 2008 [2]. Then this part suddenly disappeared. Nowadays MKL provides FPA only and implementations for IA would have to go through the same decade of difficulties MKL has gone through to get from basic IA to high-level operations.

Recent papers and software tools such as Intlab therefore try to reuse the FPA in MKL for IA by applying high-level reasoning on the code [4,5]. For instance, for a matrix-matrix-product, MKL with the RM set to round-down for all operations, should enable us to compute a matrix that is a lower bound for the exact matrix product. By clever rewrites of IA formulas and a small number of RM changes before calls to MKL, interval enclosures for IA operations can hence be computed. Inflating the round-to-nearest result is not possible for matrices as there is no "machine epsilon" for whole matrix operations.

Here is where the trouble arrives: the reliability of the IA results boils down to setting the RM for all subsequent FP operations *reliably*. Indeed suppose we work in Matlab/Intlab (for other tools, like Maple, Mathematica, it is similar), we have a mix of C code, MKL and specific Matlab or Intlab code.

For C code, `fesetround` exists. Matlab uses it, too. However e.g. printing instructions might affect the RM again. How a RM change is propagated from one thread or node of a cluster to all others is unspecified in the C standard.

In MKL the RM can be specified only in the VML (Vector Math Library) part and any multi-threading and clustering behavior is not documented. Further MKL executes for the same function different codes depending on word length, the processor vendor or the possibility to use the x87 co-processor or the SSE2 instruction set. In the generic code the internal computations are essentially performed in extended precision and then converted back to double or single. There is no known guarantee that the result actually is a reliable bound.

Moreover as mentioned in a March 2012 message on the `reliable_computing` mailing list by Frédéric Goualard, the RM can change independently of the one specified by the programmer and obviously independently of the prerequisites of other libraries. The quality of the final result is seriously compromised.

With such a mess, how can IA be called *reliable*? We cannot know in each piece of code what will be the RM. So what do we know for a mix of codes?

We are thus calling for *reliable* support for setting the RM and clear documentation in all the tools mentioned, as they are MKL, Matlab, Intlab, Maple, Gap. Otherwise publishing papers on *reliable* IA seems to be a waste of time.

References:

[1] *Intel® Math Kernel Library 10.3 – Documentation*, `http://software.intel.com/sites/products/documentation/hpc/mkl/mklman/index.htm`.

[2] *Intel® Math Kernel Library – reference Manual*, `http://www.nsc.ru/interval/MKLmanual.pdf`, September 2007.

[3] *MATLAB, version 7.14 (R2012a)*, The MathWorks Inc., Natick, Massachusetts, 2012, `http://www.mathworks.fr/help/techdoc/`.

[4] S.M. Rump, INTLAB – INTerval LABoratory, In *Developments in Reliable Computing* (Tibor Csendes, ed.), Dordrecht, 1999, pp. 77–104.

[5] C.R. Milani, M. Kolberg and L.G. Fernandes, Solving dense interval linear systems with verified computing on multicore architectures, In *Proc. of the 9th Intern'l Conf. on HPC for Comput'l Science*, 2011, pp. 435–448.

[6] W. Hofschuster and W. Krämer, *FI_LIB, eine schnelle und portable Funktionsbibliothek für reelle Argumente und reelle Intervalle im IEEE-double-Format*, Tech. Rep. 98/7, Univ. Karlsruhe, 1998.

A framework of high precision eigenvalue estimation for selfadjoint elliptic differential operator

Xuefeng Liu and Shin'ichi Oishi

Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan
xfliu.math@gmail.com

Keywords: eigenvalue problem, finite element method, homotopy method, Lehmann-Goerisch's theorem

Based on several fundamental preceding research results[1–4], this talk aims to propose a framework to provide high precision bounds for the leading eigenvalues of selfadjoint elliptic differential operator over polygonal domain:

$$-\text{div}\ (a\nabla u) + cu = \lambda u \text{ in } \Omega, \quad u = 0 \text{ on } \partial\Omega \tag{1}$$

where $a \in C^1(\Omega)$ and $c \in L_\infty(\Omega)$. The proposed framework has the following features:

- the domain of eigenvalue problem in consideration can be of free shape, which is because the finite element method with nice flexibility is successfully adopted in bounding the eigenvalues [2];
- it can deal with general selfadjoint elliptic operator, where the homotopy method [1] plays an important role;
- the obtained eigenvalue bounds have high precision, which is due to Lehmann-Goerisch's theorem [3,4] and well constructed approximating base function.

The eigenvalue problem (1) is solved by considering the weak formulation:

$$\text{Find } u \in H_0^1(\Omega), \lambda \in R, \text{ s.t. } (a\nabla u, \nabla v) + (cu, v) = \lambda(u, v), \forall v \in H_0^1(\Omega)\ , \tag{2}$$

where $H_0^1(\Omega)$ is a kind of Sobolev function space. Let us denote the eigenvalues by $\lambda_1 \leq \lambda_2 \leq \cdots$. The high precision bounds for the leading eigenvalues $\{\lambda_i\}$

are obtained in three steps.

Step 1: the base eigenvalue problem $-\Delta u = \lambda u$ is solved approximately in a certain finite element space with approximate eigenvalues as

$$\lambda_{1,h} \leq \lambda_{2,h} \leq \cdots \leq \lambda_{n,h} \, ,$$

and an error estimation for the approximate eigenvalues is given as below [2],

$$\frac{\lambda_{i,h}}{1 + M_h \lambda_{i,h}} \leq \lambda_i \leq \lambda_{i,h}, \quad (i = 1, \cdots, n) \, , \tag{3}$$

where M_h is computable quantity depending on domain shape and mesh size.

Step 2: the eigenvalue bounds for general elliptic operator in consideration is obtained by applying the homotopy method [2], which estimates the eigenvalue variation in transforming the base problem $-\Delta u = \lambda u$ to the one wanted. If the domain is convex, this step can be simplified by extending the result of (3).

Step 3: the Lehmann-Goerisch's theorem [3,4] is applied to sharpen the bounds along with proper selection of base function to approximate the eigenfunction. To deal with the domain of free shape, the singular base function corresponding to the singular part of eigenfunction, and Bezier patch over triangulation of domain are used.

In the talk, we will also illustrate several examples to demonstrate the efficiency of our proposed framework.

References:

[1] M. Plum, Bounds for eigenvalues of second-order elliptic differential operators, *The Journal of Applied Mathematics and Physics (ZAMP)*, 42 (1991), No. 6, pp. 848–863.

[2] X. Liu, S. Oishi, Verified eigenvalue evaluation for Laplacian over polygonal domain of arbitrary shape, submitted to *SIAM Journal on Numerical Analysis*, 2012.

[3] N.J. Lehmann, Optimale eigenwerteinschließungen, *Numerische Mathematik*, 5 (1963), No. 1, pp. 246–272.

[4] H. Behnke, F. Goerisch, Inclusions for eigenvalues of selfadjoint problems, *Topics in Validated Computations* (ed.J. Herzberger), North-Holland, Amsterdam, 1994, pp. 277–322.

Comparisons of implementations of Rohn modification in PPS-methods for interval linear systems

Dmitry Yu. Lyudvin, Sergey P. Shary

Institute of Computational Technologies SD RAS
6, Lavrentiev ave.
630090 Novosibirsk, Russia
lyudvin@ngs.ru, shary@ict.nsc.ru

Keywords: interval analysis, interval linear algebraic system, method of partitioning of the parameter set, Rohn modification

We consider interval linear systems of the form $\boldsymbol{A}x = \boldsymbol{b}$ with interval matrices $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$ and interval right-hand side vectors $\boldsymbol{b} \in \mathbb{IR}^n$. The interval system is understood as a family of point linear systems $Ax = b$ with $A \in \boldsymbol{A}$ and $b \in \boldsymbol{b}$. The solution set of the interval linear system is defined as the set $\Xi(\boldsymbol{A}, \boldsymbol{b}) = \{\, x \in \mathbb{R}^n \mid (\exists\, A \in \boldsymbol{A})(\exists\, b \in \boldsymbol{b})(\, Ax = b\,)\,\}$, formed by solutions to all the point systems $Ax = b$ with $A \in \boldsymbol{A}$ and $b \in \boldsymbol{b}$. We are interested in the optimal enclosure of the solution set to the interval linear system, i.e. the least inclusive interval vector that contains the solution set.

For the solution of the above problem, we use the parameter partitioning methods or, shortly, PPS-methods developed in [1, 2]. The essence of PPS-methods is sequential refining the estimates of the solution set through adaptive partitioning of the interval parameters of the system under solution.

The purpose of the present work was to compare various implementations of PPS-methods that use

1) Rohn's technique for eliminating unpromising vertex combinations;
2) estimate monotonicity test, with respect to the components of the matrix and the right-hand side vector of the system;
3) various enclosure methods for interval linear systems;
4) various ways of processing the so-called working list, in which the results of the partition of the interval linear system are stored.

Special attention is paid to the modification based on Rohn's technique, which is the most complex, laborious, but the most efficient one for the systems of moderate dimensions.

J. Rohn revealed that, if the matrix $\boldsymbol{A}$ is regular, then both minimal and maximal component-wise values of the points from the solution set are attained at the set of no more than 2^n so-called extreme solutions to the equation $|(\operatorname{mid}\boldsymbol{A})x - \operatorname{mid}\boldsymbol{b}| = \operatorname{rad}\boldsymbol{A} \cdot |x| + \operatorname{rad}\boldsymbol{b}$ [3]. Our INTLAB code `linppse` [4] implements a modification of the general PPS-methods based on this result [2]. We have carried out numerical tests and examined the efficiency of the algorithm depending on the properties of the interval matrix of the system.

Also, we have investigated various versions of PPS-methods, which used, as procedures for computing basic enclosures, Krawczyk method, modified Krawczyk method with epsilon-inflation, interval Gauss method, interval Gauss-Seidel iteration, Hansen-Bliek-Rohn procedure, `verifylss` procedure from the toolbox INTLAB. Experimental results demonstrated that, amongst the above listed techniques, Hansen-Bliek-Rohn procedure with preliminary preconditioning is the best enclosure for PPS-methods.

Based on numerical experiments, we elaborate practical recommendations on how to optimize, within the PPS-methods, processing the working list (of "systems-descendants"). Finally, we present the results of comparisons between two computer codes for computing optimal enclosures of the solution set to interval linear systems, namely, our `linppse` [4] and `verintervalhull` from Rohn's VERSOFT package [5].

References:

[1] S.P. Shary, A new class of algorithms for optimal solution of interval linear systems, *Interval Computations*, 4 (1992), No. 2, pp. 18–29.

[2] S.P. Shary, Parameter partition methods for optimal numerical solution of interval linear systems, *Computational Science and High-Performance Computing III. The 3rd Russian-German advanced research workshop, Novosibirsk, Russia, 23–27 July 2007* (E. Krause, Yu.I. Shokin, M. Resch, N.Yu. Shokina, eds.), Springer, Berlin-Heidelberg, 2008, pp. 184–205.

[3] M. Fiedler, J. Nedoma, J. Ramik, J. Rohn, K. Zimmermann, *Linear optimization problems with inexact data*, Springer, New York, 2006.

[4] *The program for the optimal (exact) componentwise estimation of the united solution set to interval linear system of equations*, `http://www.nsc.ru/interval/Programing/MCodes/linppse.m`

[5] *Verification software in MATLAB/INTLAB*, `http://www.cs.cas.cz/rohn/matlab`

Componentwise inclusion for solutions in least squares problems and underdetermined systems

Shinya Miyajima

Faculty of Engineering, Gifu University
1-1 Yanagido, Gifu-shi, Gifu 501-1193, Japan
miyajima@gifu-u.ac.jp

Keywords: least squares problems, underdetermined systems, minimal 2-norm solution, numerical enclosure, verified error bound

In this talk, we are concerned with the accuracy of numerically computed results for solutions in least squares problems

$$\min_{x\in\mathbb{R}^n} \|b - Ax\|_2, \quad A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m, \tag{1}$$

and minimal 2-norm solutions in underdetermined systems

$$Ax = b, \quad A \in \mathbb{R}^{n\times m},\ b \in \mathbb{R}^n, \tag{2}$$

where $m \geq n$ and A has full rank. The problems (1) and (2) arise in many applications of scientific computations, e.g. linear and nonlinear programming [1], statistical analysis, signal processing, computer vision [2] and so forth. It is well known (e.g. [3,4]) that the solutions in (1) and (2) can be written as A^+b, where A^+ denotes the pseudo-inverse of A.

We consider in this talk numerically enclosing A^+b, specifically, computing error bounds for $\tilde{x}$ using floating point operations, where $\tilde{x}$ denotes a numerical result for A^+b. It is well known (e.g. [4]) that A^+b in (1) and (2) can be computed by solving the augmented linear systems

$$\begin{pmatrix} A & -I_m \\ O_n & A^T \end{pmatrix}\begin{pmatrix} x \\ w \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} A^T & -I_m \\ O_n & A \end{pmatrix}\begin{pmatrix} w \\ x \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix}, \tag{3}$$

respectively, where I_m and O_n denote the $m \times m$ identity matrix and the $n \times n$ zero matrix, respectively, since these systems imply $x = A^+b$. The INTLAB [5] function `verifylss` encloses A^+b in (1) and (2) by enclosing solutions in (3), supplies *componentwise* error bounds, and requires $\mathcal{O}((m+n)^3)$ operations.

The VERSOFT [6] routine `verlsq` returns the enclosure of A^+b in (1) and (2) by computing an interval matrix including A^+ and gives *componentwise* error bounds. The author [7] has proposed algorithms for enclosing A^+b in (2), which gives *normwise* error bounds. In this algorithm, (3) is not utilized, i.e. (2) is directly considered, so that the computational cost of this algorithm is not $\mathcal{O}((m+n)^3)$ but $\mathcal{O}(m^2n)$ operations. Recently Rump [8] proposed fast algorithms for enclosing A^+b in (1) and (2), which return *normwise* error bounds.

The purpose of this talk is to propose algorithms for enclosing A^+b in (1) and (2) which supply *componentwise* error bounds and are as fast as the algorithms in [8]. These algorithms do not assume but prove A to have full rank. We prove that the obtained error bounds by the proposed algorithms are equal or smaller than those by the algorithms in [8], and finally compare the proposed algorithms with `verifylss`, `verlsq` and the algorithms in [7,8] through some numerical results.

References:

[1] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.

[2] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, Bundle adjustment-a modern synthesis, *Lect. Notes Comput. Sc.*, 1883 (2000), pp. 153–177.

[3] G.H. Golub, C.F. Van Loan, *Matrix Computations, third ed.*, The Johns Hopkins University Press, Baltimore and London, 1996.

[4] N.J. Higham, *Accuracy and Stability of Numerical Algorithms, second ed.*, SIAM Publications, Philadelphia, 2002.

[5] S.M. Rump, INTLAB – INTerval LABoratory, in *Developments in Reliable Computing* (T. Csendes, ed.), Kluwer Academic Publishers, Dordrecht, 1999, pp. 77–104.

[6] J. Rohn, VERSOFT: Verification software in MATLAB / INTLAB, `http://uivtx.cs.cas.cz/~rohn/matlab/`.

[7] S. Miyajima, Fast enclosure for solutions in underdetermined systems, *J. Comput. Appl. Math.*, 234 (2010), pp. 3436–3444.

[8] S.M. Rump, Verified bounds for least squares problems and underdetermined linear systems, *SIAM J. Matrix Anal. Appl.*, 33 (2012), pp. 130–148.

Verified computations for all generalized singular values

Shinya Miyajima

Faculty of Engineering, Gifu University
1-1 Yanagido, Gifu-shi, Gifu 501-1193, Japan
miyajima@gifu-u.ac.jp

Keywords: singular values, generalized singular values, verified bounds

A matrix factorization having great importance in numerical linear algebra is the singular value decomposition (SVD) (e.g. [1]), which is based on the following theorem:

Theorem 1 *Let $A \in \mathbb{R}^{m\times n}$ be given and $q := \min(m,n)$. There exist orthogonal $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ such that*

$$U^T AV = \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_q),$$
$$\sigma_1 \geq \cdots \geq \sigma_{r^*} > \sigma_{r^*+1} = \cdots = \sigma_q = 0, \quad r^* = \mathrm{rank}(A).$$

The nonnegative real numbers σ_i, $i = 1, \ldots, q$ are called the singular values of A, which play important roles in application areas. It is well known that σ_i^2 are the eigenvalues of the symmetric pencils $A^TA - \lambda I_n$ and $AA^T - \lambda I_m$, where I_n denotes the $n \times n$ identity matrix.

Van Loan [2] generalized the SVD. This generalization is called the generalized singular value decomposition (GSVD) and based on the following theorem:

Theorem 2 *Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ and $B \in \mathbb{R}^{p\times n}$ be given and $q := \min(p,n)$. There exist orthogonal $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{p\times p}$ and a nonsingular $X \in \mathbb{R}^{n\times n}$ such that*

$$U^T AX = \Sigma_A = \mathrm{diag}(c_1, \ldots, c_n), \quad V^T BX = \Sigma_B = \mathrm{diag}(s_1, \ldots, s_q),$$
$$0 \leq c_1 \leq \cdots \leq c_n \leq 1, \quad 1 \geq s_1 \geq \cdots \geq s_{r^*} > s_{r^*+1} = \cdots = s_q = 0,$$
$$r^* = \mathrm{rank}(B), \quad c_i^2 + s_i^2 = 1, \ i = 1, \ldots, q.$$

The quotients $\mu_j = c_j/s_j$, $j = 1, \ldots, r^*$ are called the generalized singular values of A and B. Note that μ_j^2 are the eigenvalues of the symmetric pencil $A^TA - \lambda B^TB$. Although more general definition of the GSVD can be found in

[3], in this talk, we define the GSVD by Theorem 2 for simplicity. The GSVD is a tool used in many applications, such as damped least squares, least squares with equality constraints, certain generalized eigenvalue problems and weighted least squares [2].

In this talk, we consider computing verified bounds of *all* the singular values and generalized singular values. For the singular values, Oishi [4] first proposed such an algorithm utilizing numerical full SVD. Recently Rump [5] proposed a fast algorithm which utilizes not the full SVD but the eigen-decomposition. The VERSOFT [6] routine `versingval` encloses all the singular values utilizing an augmented matrix. For the generalized singular values, an algorithm for computing verified bounds of c_j, s_j and j-th columns of U, V and X for *specified* $j \in \{1, \ldots, r^*\}$ has been proposed in [7]. As long as the author know, on the other hand, an algorithm giving verified bounds of μ_j for *all* $j = 1, \ldots, r^*$ has not been known.

The purpose of this talk is to propose algorithms for computing verified bounds of *all* the singular values or generalized singular values. For the singular values, we propose an algorithm which is faster than the algorithms in [4,5] and `versingval`. We extend this algorithm to the generalized singular values and propose two algorithms. The first and second algorithms are applicable if $B^T B$ and $A^T A$ are nonsingular, respectively. We do not assume but prove these nonsingularities during the executions of these algorithms. Numerical results show the properties of the proposed algorithms.

References:

[1] G.H. Golub, C.F. Van Loan, *Matrix Computations, third ed.*, The Johns Hopkins University Press, Baltimore and London, 1996.

[2] C.F. Van Loan, Generalizing the singular value decomposition, *SIAM J. Numer. Anal.*, 13 (1976), No. 1, pp. 76–83.

[3] C.C. Paige, M.A. Sauders, Towards a generalized singular value decomposition, *SIAM J. Numer. Anal.*, 18 (1981), No. 3, pp. 398–405.

[4] S. Oishi, Fast enclosure of matrix eigenvalues and singular values via rounding mode controlled computation, *Linear Algebra Appl.*, 324 (2001), pp. 133–146.

[5] S.M. Rump, Verified bounds for singular values, in particular for the spectral norm of a matrix and its inverse, *BIT Numer. Math.*, 51 (2011), No. 2, pp. 367–384.

[6] J. Rohn, VERSOFT: Verification software in MATLAB / INTLAB, `http://uivtx.cs.cas.cz/~rohn/matlab/`.

[7] G. Alefeld, R. Hoffmann, G. Mayer, Verification algorithms for generalized singular values, *Math. Nachr.*, 208 (1999), pp. 5–29.

Information support of scientific symposia

Yurii I. Molorodov

Institute of Computational Technologies SB RAS
6, Lavrentiev ave.
630090 Novosibirsk, Russia
yumo@ict.sbras.ru

Keywords: information systems, interval arithmetic, engineering

In information support of science, an important point is organization of regular meetings and discussions of researchers working in specific fields. In particular, this is critical for scientific computing, computer arithmetic, and verified numerical methods, where one should have assess to the achievements, to see trends and to predict the prospects of this area of knowledge. That is the purpose of the current 15'th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Verified Numerical Computations, which will be held in Novosibirsk on September 23–29, 2012.

The events, such as SCAN'2012, are preceded by a large amount of preparatory work performed by the organizers proper and many other people [1, 2]. The first step is to initiate the conference, formulate its goals and objectives, its scope, determine its time and venue, form an organization team. The competence of the program committee is to identify the "content" of the conference, the specificity of the submissions to be presented and discussed. These committees are responsible for the overall success of the conference.

The purpose of the second stage is the notification of all potentially interested individuals and organizations about the forthcoming conference, its scope, venue, format and dates, conditions of participation. To do this, the organizers use a wide range of various means: putting the information onto electronic bulletin boards devoted to the relevant topics, direct mailing, printing and distributing leaflets, etc. At this stage, the availability of information plays a crucial role, so that it becomes necessary to maintain a web-site of the conferences that publishes and updated promptly all the information, including news and ads (in our case, this is `http://conf.nsc.ru/scan2012`).

At the next stage, the organizers analyze and process the input information. In its flow, two major components can be identified: applications from potential participants, and abstracts of the submissions. A preliminary qualitative and

quantitative assessments is made in order to determine a general outline of the forthcoming meeting. The place of each submitted abstract within the program of the meeting is determined.

At the beginning of the fourth stage, after the pre-appointed time elapses, the organizers stop receiving abstracts and turn to their analysis. Peer-reviewing of the submissions is usually performed by a Program Committee, consisting of experts in the field, that evaluate the submissions according to several criteria: originality of the results, the quality of presentation, relevance of the work, and others. Often, within the overall scope of the conference, there exist several different branches, and the corresponding submissions are to be presented and discussed separately. It is the task of the program committee to co-ordinate such branches and conduct the overall scientific policy. As the result of this stage, a pool of accepted submissions is formed that can be a basis for compiling a preliminary working program of the meeting. At the end of the fourth stage, a preliminary program of the meeting is prepared as well as the overall activity plan, which are published on the conference website. Also, the organizing committee makes and prints the volume of abstracts to be distributed among the conference participants during the on-desk registration.

The main part of the conference begins with registration of the arrived participants. The organizers should be aware in advance of their intention to stay in hotels of a class and provide the opportunity. At this stage, a large number of various problems may occur. Much of them should be predicted and prevented at the preparatory stages, although they cannot be totally eliminated.

After completion of the main stage, the final part of the scientific meeting comes, when the organizers should summarize the overall results of the conference and make them publicly available for future use. This traditionally amounts to publication of the conference proceedings, either in paper or electronic form. For SCAN'2012, the proceedings will be published in the open electronic journal Reliable Computing [3].

References:

[1] A.M. Fedotov, A.E. Guskov, Yu.I. Molorodov, *Conference Information system*, `http://www/sbras.ru/ws/`

[2] A.E. Guskov, *Semantic Web: Theory and Practice*, LAP – LAMBERT Academic Publishing, 2005.

[3] Reliable Computing (an open electronic journal), `http://interval.louisiana.edu/reliable-computing-journal`

Towards an efficient implementation of CADNA in the BLAS: example of the routine DgemmCADNA

Sethy Montan[1,2], Jean-Marie Chesneaux[2], Christophe Denis[1], Jean-Luc Lamotte[2]

[1]EDF R&D - Département SINETICS
1, Avenue du général de Gaulle, 92141 Clamart Cedex – France
[2]Laboratoire d'Informatique de Paris 6 - Université Pierre et Marie Curie,
4 place jussieu, 75252 Paris Cedex 05 – France
sethy.montan@edf.fr

Keywords: CADNA, discrete stochastic arithmetic, CESTAC, BLAS

Several approximations occur during a numerical simulation : physical phenomena are modelled using mathematical equations, continuous functions are replaced by discretized ones and real numbers are replaced by finite-precision representations (floating-point numbers). The use of the IEEE-754 arithmetic generates round-off errors at each elementary arithmetic operation. By accumulation, these errors can affect the accuracy of computed results, possibly leading to partial or total inaccuracy. The effect of these rounding errors can be analyzed and studied by some methods like forward/backward analysis, interval arithmetic or stochastic arithmetic (which is implemented in the CADNA validation tool).

A numerical verification of industrial codes, such those that are developed at EDF R&D –the French provider of electricity–, is required to estimate the precision and the quality of computed results, even more for code running in HPC environments where millions instructions are performed each second. These programs usually use external libraries (MPI, BLACS, BLAS, LAPACK) [1]. In this context, it is required to have a tool as nonintrusive as possible to avoid rewriting the original code. In this regard, the CADNA library appears to be one of the promising approach for industrial applications.

The CADNA library, developed by the Laboratoire d'Informatique de Paris 6, enables us to estimate round-off error propagation using a probabilistic approach in any simulation program (written in C/C++ or Fortran) and to control its numerical quality by detecting numerical instabilities that may occur at run time [2]. CADNA implements Discrete Stochastic Arithmetic which is

based on a probabilistic model of round-off errors (this arithmetic is defined with the CESTAC Method). CADNA provides new numerical types, the so-called stochastic types, on which round-off errors can be estimated. However, a problem remains: stochastic types are not compatible with the aforementioned libraries. It is, therefore, necessary to develop some extensions for these external libraries.

We are interested in an efficient implementation of the BLAS routine xGEMM compatible with CADNA. We have called this new routine *DgemmCADNA*. The BLAS (Basic Linear Algebra Subprograms) are routines that provide standard building blocks for performing basic vector and matrix operations and xGEMM is the routine which goal is to perform matrix multiplication [5]. The implementation of a basic algorithm for matrix product compatible with stochastic types leads to an overhead greater than 1000 for a matrix of 1024*1024 compared to the standard version and commercial versions of xGEMM. This overhead is due to the use of stochastic types, the rounding mode which changes randomly at each elementary operation ($\times, /, +, -$), and a non optimized use of the memory (cache and TLB misses).

We will present different solutions to reduce this overhead and the results we have obtained. In order to improve the hierarchical memory usage, special data structures (Block Data Layout) are used. This allows us to improve the memory performance to reduce cache and TLB misses. A new implementation of CESTAC Method has been introduced to reduce the overhead due to the random rounding mode. Finally, we have obtained an overhead about 25 compared to GotoBLAS in a sequential mode.

We will also present, briefly, new extensions for CADNA : CADNA_MPI and CADNA_BLACS which allow to use stochastic data in programs using the communications standard routines (MPI or BLACS).

References:

[1] Ch. Denis and S. Montan, Numerical verification of industrial numerical codes, *ESAIM Proceedings*, 35 (March 2012), pp. 107–113.

[2] F. Jézéquel, J.-M. Chesneaux, and J.-L. Lamotte, A new version of the CADNA library for estimating round-off error propagation in Fortran programs, *Computer Physics Communications*, 181, No. 11 (2010), pp. 1927–1928.

[3] K. Goto, and R.A. Van De Geijn, High-performance implementation of the level-3 BLAS, *ACM Transactions on Mathematical Software (TOMS)*, 35, No. 1 (2008), (14 pages).

[4] N.J. Higham, *Accuracy and Stability of Numerical Algorithms, 2nd ed.*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.

[5] *Basic Linear Algebra Technical Forum Standard*, August 2001.

Verification methods for linear systems on a GPU

Yusuke Morikura[1], Katsuhisa Ozaki[2] and Shin'ichi Oishi[3]

[1]Graduate School of Fundamental Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan
[2]Department of Mathematical Sciences, Shibaura Institute of Technology
307 Fukasaku, Minuma-ku, Saitama-shi, Saitama 337-8570, Japan
[3]Faculty of Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan
m.myusuke@suou.waseda.jp

Keywords: linear systems, verified numerical computations, a priori error estimate, GPGPU

This talk discusses a verification method for linear systems:

$$Ax = b$$

where A is a real $n \times n$ dense matrix and b is a real n-vector. The verification method means a method which outputs an error bound between a numerical solution and an exact solution by floating-point computations. The aim of this talk is to propose a verification method for linear systems suited for a GPU.

The GPU is used for not only acceleration of building of images but also for numerical computations since the computational performance is very high. Recently, useful toolboxes and libraries for GPGPU (General-Purpose Computation on Graphics Processing Unit) have been developed, for example, MATLAB Parallel Computing Toolbox, JACKET, MAGMA and so forth.

Several verification methods for linear systems have been developed for a dense coefficient matrix [1, 2, 3]. Many verification methods require switches

of rounding modes defined by the IEEE 754 standard [4]. However, since the GPU (Graphics Processing Unit) does not have no dynamically configurable rounding mode [5], the methods of [1, 3] cannot be implemented on the GPU straightforwardly. To overcome the problems, we first improved Ogita-Rump-Oishi's error estimation [2] by using new floating-point error estimations by Rump [6]. Our algorithm does not switch rounding modes, namely, it works only in default rounding mode on GPGPU (rounding to nearest).

Next, an amount of device (GPU) memory is little compared to that of host (CPU) memory in many cases. For example, Tesla C2070 by NVIDIA Corporation has 6 Gbytes memory although CPU has much more working memory (recently, amount of memory installed for a CPU can be over 48 Gbytes). Therefore, we apply blockwise computations to reduce the amount of working memory of GPU. Data transfer of block matrices from CPU to GPU and from GPU to CPU is required for blockwise computations and its transfer speed is slow due to low bandwidth. However, numerical results illustrate that computational times using blockwise computation are only 10.7 percent slower than that by using no blockwise computation. Therefore, blockwise computation does not significantly slow down the computational performance. The error bound by our algorithm is twice or three times better than that by [2].

References:

[1] S. Oishi, S. M. Rump, Fast verification solutions of matrix equations, *Numer. Math.*, 90 (2002), No. 4, pp. 755–773.

[2] T. Ogita, S. M. Rump, S. Oishi, *Verified Solutions of Linear Systems without Directed Rounding, Technical Report 2005–04*, Advanced Research Institute for Science and Engineering, Waseda University, Tokyo, Japan, 2005.

[3] T. Ogita, S. Oishi, Fast verification method for large-scale linear systems, *IPSJ Transactions*, 46, No. SIG10(TOM12) (2005), pp. 10–18.

[4] *ANSI/IEEE Std 754–1985 : IEEE Standard for Binary Floating-Point Arithmetic*, New York, 1985.

[5] NVIDIA, CUDA C Programming Guide, `http://developer.nvidia.com/nvidia-gpu-computing-documentation`

[6] S.M. Rump, Error estimation of floating-point summation and dot product, *BIT Numerical Mathematics*, 52 (2012), No. 1, pp. 201–220.

Approach based on instruction selection for fast and certified code generation

Christophe Mouilleron[1,2,3], Amine Najahi[1,2,3], Guillaume Revy[1,2,3]

[1]Univ. Perpignan Via Domitia, DALI, F-66860, Perpignan, France
[2]Univ. Montpellier II, LIRMM, UMR 5506, F-34095, Montpellier, France
[3]CNRS, LIRMM, UMR 5506, F-34095, Montpellier, France
{amine.najahi,christophe.mouilleron,guillaume.revy}@univ-perp.fr

Keywords: fixed-point arithmetic, automatic code generation, instruction selection, numerical certification

Floating-point arithmetic [1] has become ubiquitous in the specification and implementation of programs, including those targeted at embedded systems. However, for the sake of chip area or power consumption constraints, some of these embedded systems are still shipped with no floating-point unit. In this case, only integer arithmetic is available at the hardware level. Hence, to run floating-point programs, we need either to use a library that emulates floating-point arithmetic in software (such as the FLIP* library), or to rewrite the programs to rely on fixed-point arithmetic [2]. Both approaches require the design of fixed-point routines, which appears to be a tedious and error prone task, especially since it is partly done by hand. Thus, one of the current challenges is to design automatic tools to generate fixed-point programs as fast as possible while satisfying some accuracy constraints. In this sense, we have developed the CGPE† software tool, dedicated to the generation of fast and certified codes for evaluating bivariate polynomials in fixed-point arithmetic. This tool, based on the generation of several fast evaluation codes combined with a systematic numerical verification step, is well suited for VLIW integer processors using only binary adders and multipliers. We propose here an extension of CGPE, which consists in adding a step based on instruction selection [4, §8.9] to improve the speed and the accuracy of the generated codes for more advanced architectures.

Given an instruction set architecture, *instruction selection* is the compilation process that aims at finding a sequence of instructions implementing "at best" a given program. It works on a target-independent intermediate representation of this program, represented as a tree or a directly acyclic graph (DAG), and

*Floating-point Library for Integer Processors (see `http://flip.gforge.inria.fr`).
†Code Generation for Polynomial Evaluation (see `http://cgpe.gforge.inria.fr` and [3]).

is usually used to optimize the code size or latency on the target architecture, while no guarantee is provided concerning the accuracy of the generated code. The general problem of instruction selection has been well studied and, even though it has been proven to be NP-complete even for simple machines in the case of DAGs [5], several algorithms exist to tackle this problem (see [5] and the references therein).

In the context of CGPE, where we represent polynomial evaluation expressions with DAGs, we can benefit from this work on instruction selection by combining it with the numerical verification step already implemented. The interest of our new approach is twofold. First it is much more flexible than writing a generation algorithm for each available processor. Indeed, it mainly needs to work on the DAG representation of the expression to be implemented, which is independent of the target architecture, and thus it makes easier to handle various architectures shipping different kind of instructions. Second it allows us to generate automatically codes optimized for a given target and satisfying various criteria like accuracy and performance, as well as code size or number of operators. This approach has been validated on the evaluation of polynomials, where it allows us to write efficient codes using at best some advanced architecture features such as the presence of a fused-multiply-add operator.

References:

[1] *IEEE Standard 754-2008 – IEEE Standard for Floating-Point Arithmetic*, 2008.

[2] D. Menard, D. Chillet, and O. Sentieys, Floating-to-fixed-point conversion for digital signal processors, In *EURASIP Journal on Applied Signal Processing*, 2006, pp. 1–15.

[3] C. Mouilleron and G. Revy, Automatic generation of fast and certified code for polynomial evaluation, In *Proc. of the 20th IEEE Symposium on Computer Arithmetic* (E. Antelo, D. Hough, and P. Ienne, editors), IEEE, 2011, pp. 233–242.

[4] A.V. Aho, R. Sethi, and J.D. Ullman, *Compilers: Principles, Techniques, and Tools*, Addison-Wesley, Boston, 1986.

[5] D.R. Koes and S.C. Goldstein, Near-optimal instruction selection on DAGs, In *Proc. of the 6th IEEE/ACM international Symposium on Code Generation and Optimization*, ACM, New York, 2008, pp. 45–54.

JInterval library: principles, development, and perspectives

Dmitry Nadezhin[1] and Sergei Zhilin[2]

[1]Oracle Labs, Zelenograd, Russia
[2]Altai State University, 61, Lenin ave., 6560049, Barnaul, Russia
dmitry.nadezhin@oracle.com, sergei@asu.ru

Keywords: interval computations, Java, library

`JInterval` [1] was started in 2008 as a research project to develop a Java library for interval computations. The library is intended mainly for developers who create Java-based applied software. The design of the `JInterval` library was guided by the following requirements ordered by descending priority:

1. *The library must be clear and easy to use.* No matter how wonderful a software tool is, it will be hardly accepted by developers if it is not transparent and easy to use.

2. *The library should provide flexibility in the choice of interval arithmetic for computations.* The user must be able to choose interval arithmetic (classical, Kaucher, complex rectangular, complex circular, etc.) and to switch one arithmetic to another if they are compatible. Syntactic differences between the use of this or that arithmetic should be minimized.

3. *The library should provide flexibility in extending its functionality.* The library must be layered functionally. Four layers should be defined: interval arithmetic operators, elementary interval functions, interval vector and matrix operations, and, finally, high-level interval methods, such as solvers of equations, optimization procedures, etc. Architecture of the library must allow for extensions at every layer, starting from the bottom one.

4. *The library should provide flexibility in choosing precision of interval endpoints and associated rounding policies.* The choice of interval endpoints representation and the rounding mode could allow the user to tune accuracy and speed of computation depending on the problem he solves.

5. *The library must be portable.* Cross-platform portability of the library is one of its major strengths, being a key distinction over its closest competitors. To a large extent, this requirement is ensured by the choice of the Java technology built on the principle "write once, run anywhere". However, the design must adhere to certain restrictions on practical implementation of this requirement.

6. *The library should provide high performance.* In the era of multicore and multiprocessor systems, a prerequisite for high performance is the ability to use the library safely in a multithreaded environment.

Achieving the required flexibility leads to widening the scope of the library, which results in a vast and obscure design, contrary to the simplicity requirement. To avoid this contradiction, and to preserve clarity of the library, the overall architecture needs to be transparent and consistent. This is done due to appropriate design decisions. Methods for interval classes, regardless of interval arithmetic and of the internal representation of intervals are unified. Intervals are considered as immutable objects. The user is provided with a simple interface to manage rounding policy and interval endpoints representation.

At the moment, `JInterval` provides a user with several interval arithmetics (classical real, extended Kaucher, complex rectangular, complex circular, complex sector, complex ring), interval elementary functions, interval vector and matrix operations, as well as a few methods for inner and outer estimation of the solution sets to interval linear systems.

A number of applications have been built using `JInterval library`. A collection of plugins is developed for the data mining platform KNIME. The collection include interval regression builder, outlier detector, ILS solver, etc. Another example is mobile applications, where `JInterval` is used for position uncertainty modeling in hybrid navigation.

The experience of `JInterval` implementation and usage taught us several lessons, and further development of `JInterval` will be governed by the following principles:

1. Java language has a lot of advantages, but its syntax is not expressive enough for computational programming. Scala language (fully compliant with JVM) is considered as a basic language for a new `JInterval` implementation.

2. Presently, `JInterval` is not compliant with the project of interval arithmetic standard IEEE P1788. A new implementation will be adjusted for P1788.

3. To achieve high performance, `JInterval` will be equipped (using Java Native Interface) with optional plugins for machine-dependent implementation of high precision arithmetic and interval linear algebra algorithms.

4. For applied software developers, a rich content of the fourth layer of the library (high-level interval analysis methods) is one of the most valuable issues. Therefore the replenishment of `JInterval` with solvers of algebraic and differential equations, interval optimizers, etc., remains the foreground task.

References:

[1] Java Library for Interval Computations, `http://jinterval.kenai.com`.

Verified integration of ODEs with Taylor models

Markus Neher

Karlsruhe Institute of Technology
Institute for Applied and Numerical Mathematics
Kaiserstr. 89-93
76049 Karlsruhe, Germany
markus.neher@kit.edu

Keywords: ODEs, initial value problems, Taylor models

Verified integration methods for ODEs are methods that compute rigorous bounds for some specific solution or for the flow of some initial set of a given ODE. For almost fifty years, interval arithmetic has been used for calculating bounds for solutions of initial value problems. The origin of these methods dates back to Moore [5]. The most well-known interval method is the QR method due to Lohner [2], implemented in the AWA software package.

Unfortunately, interval methods sometimes suffer from overestimation. Pessimistic bounds are caused by the dependency problem, that is the lack of interval arithmetic to identify different occurrences of the same variable, and by the wrapping effect, which occurs when intermediate results of a calculation are enclosed into intervals.

Overestimation is a particular concern in the verified solution of initial value problems for ODEs. While it may sometimes be possible to reduce dependency by skillful reformulation of the given equations or by evaluating all function expressions by centered forms, the wrapping effect is more difficult to prevent. Interval methods usually compute enclosures of the flow at intermediate time steps of the integration domain. When the flow is a nonconvex set and is bounded by some convex interval, overestimation is inevitable.

For improving bounds, Taylor models have been developed as a combination of symbolic and interval computations by Berz and his group since the 1990s. In Taylor model methods, the basic data type is not a single interval, but a Taylor model $\mathcal{U} := p_n + \boldsymbol{i}$ consisting of a multivariate polynomial p_n of order n and some remainder interval $\boldsymbol{i}$. In computations that involve $\mathcal{U}$, the polynomial part is propagated by symbolic calculations where possible, and is thus hardly affected by the dependency problem or the wrapping effect. Only the interval

remainder term and polynomial terms of order higher than n, which are usually small, are bounded using interval arithmetic.

Besides reducing dependency, Taylor model methods for ODEs also benefit from their capability to represent non-convex sets. This is an intrinsic advantage over interval methods for enclosing the flows of nonlinear ODEs, especially in combination with large initial sets or with large integration domains [1, 3, 4, 6].

In our talk, we analyze Taylor model methods for the verified integration of ODEs and compare these methods with interval methods.

References:

[1] M. Berz, K. Makino, Suppression of the wrapping effect by Taylor model-based verified integrators: Long-term stabilization by shrink wrapping, *Int. J. Diff. Eq. Appl.*, 10 (2005), pp. 385–403.

[2] R. Lohner, Enclosing the solutions of ordinary initial- and boundary-value problems, In *Computer arithmetic: Scientific Computation and Programming Languages* (E. Kaucher, U. Kulisch, Ch. Ullrich, eds), Teubner, Stuttgart, 1987, pp. 255–286.

[3] K. Makino, M. Berz, Suppression of the wrapping effect by Taylor model-based verified integrators: Long-term stabilization by preconditioning, *Int. J. Diff. Eq. Appl.*, 10 (2005), pp. 353–384.

[4] K. Makino, M. Berz, Suppression of the wrapping effect by Taylor model-based verified integrators: The single step, *Int. J. Pure Appl. Math.*, 36 (2006), pp. 175–197.

[5] R.E. Moore, *Interval Analysis*, Prentice Hall, Englewood Cliffs, N.J., 1966.

[6] M. Neher, K.R. Jackson, N.S. Nedialkov, On Taylor model based integration of ODEs, *SIAM J. Numer. Anal.*, 45 (2007), pp. 236–262.

Searching solutions to the interval multi-criteria linear programming problem

Sergey I. Noskov

Irkutsk University of Railway Communications
15, Chernyshevskogo str., 664074 Irkutsk, Russia
noskov_s@irgups.ru

Keywords: interval, multi-criteria task, linear programming

A multi-criteria linear programming problem (MLP) is one of the classical problem statements in the theory of decision making, and its formulation has the form:

$$Cx \to \max_{x \in X}, \quad X = \{\, x \in \mathbb{R}^n \mid Ax \le b,\ x \ge 0\}. \tag{1}$$

Here, in contrast to the usual linear programming problem (LP), C is a matrix with the dimension $l \times n$, and not a vector, A is a constraint matrix with the dimension $m \times n$. Thus, the multi-criteria problem (1) involves the maximization, on a polyhedron, l linear criteria at the same time, as distinct from the ordinary LP problem. Note that the normal form of (1) can be easily transformed to the canonical from. The constraint $x \ge 0$ is also easy to implement.

As a rule, the traditional solution of the problem (1) does not exist, that is, the point $x \in X$, such that $Cx \ge Cy$ for all $y \in X$ and $y \ne x$, is absent. In the case where the decision maker (DM) does not have a priori information on the relative importance of various criteria, the solution of (1) is understood as the so-called Pareto set. Denote it as $N \subset X$. The solution $x \in N$ is called Pareto solution (non-dominated, unimprovable), if it can not be improved with respect to any criterion without worsening the value of at least one of the remaining criterion. Or, formally,

$$x \in N \iff (\forall y \in X,\, y \ne x)\neg((Cy \ge Cx) \wedge (\exists i\ C^i y > C^i x)),$$

where C^i is the i-th row (i-th criterion) of the matrix C.

The problem of constructing the Pareto set in the MLP problem has been extensively covered in the literature, and one of the best publications on the

subject is the article of P.L. Yu and M. Zeleny [1]. They have derived and theoretically substantiated a number of methods for constructing Pareto sets of vertices $N^{ex} \subset N$ and the whole Pareto set. In particular, the so-called multicriteria simplex method is developed in [1] for the construction of the set N^{ex}. It is based on a fundamental theorem whose formulation is given below.

Theorem [1]. The set N^{ex} is connected, $x^0 \in N \Leftrightarrow \omega = 0$; $x^0 \in D \Leftrightarrow \omega > 0$. Here, $D = X \backslash N$, and ω is a solution of the LP problem

$$\omega = \max \sum_{i=1}^{l} e_i, \;\; \widetilde{X} = \{ (x, l) \in \mathbb{R}^{n+l} \mid x \in X, \; Cx - e \geq Cx^0, \; e \geq 0 \}. \qquad (2)$$

The essence of the algorithm for constructing the set N^{ex} that is described in [1], is as follows. We start from searching the first Pareto vertex x^1. To do this, it is sufficient to solve the LP problem with the objective function

$$\sum_{i=1}^{l} \lambda_i C^i x \to \max_{x \in X}, \;\; \lambda > 0.$$

After that, all the neighbouring vertices for point x^1 are being checked to be Pareto ones by solving the problem (2). Those who really prove to be Pareto vertices, are included in N^{ex}, then we test their adjacent vertices, etc.

It should be noted that in [1] shows (see, for example, Theorem 3.1 in [1]), a set of simple sufficient conditions for belonging to some arbitrary point $y \in X$ set D, which greatly facilitates the search. Note that in [2] is a simple way to spot the characterization of the Pareto set N.

We now pose the problem (1) somewhat differently, namely, we assume that both the constraint matrix, right-hand side and the criterion matrix are interval (the scalar formulation has long been solved by various approaches). The above raises a number of natural questions. Is the problem formulation correct? If so, what is meant by Pareto solution and Pareto vertex in this case? What will be multi-criteria simplex method? And a number of extremely interesting related questions.

References:

[1] L. Yu, M. Zeleny, The set of all nondominated solutions in linear cases and multycriteria simplex method, *J. of Math. Anal. and Applic.*, 45 (1975), No. 2, pp. 430–468.

[2] S.I. Noskov, The problem of uniqueness of Pareto-optimal solutions in the problem of linear programming with a vector criterion function, *Modern technologies. System analysis. The simulation*, Special issue, 2011, pp. 283–285.

Verified solutions of sparse linear systems

Takeshi Ogita

Division of Mathematical Sciences, Tokyo Woman's Christian University
2–6–1 Zempukuji, Suginami-ku, Tokyo 167–8585, Japan
ogita@lab.twcu.ac.jp

Keywords: sparse linear systems, floating-point arithmetic, verified numerical computations

To solve linear systems is ubiquitous since it is one of the basic and significant tasks in scientific computing. Floating-point arithmetic is widely used for this purpose. Since it uses finite precision arithmetic and numbers, rounding errors are included in computed results. To guarantee the accuracy of the results, there are methods so-called verified numerical computations based on interval arithmetic. Excellent overviews can be found in [6] and references cited therein.

Let A be a real $n \times n$ matrix, and b a real n-vector. Let $\kappa(A) = \|A\|_2 \cdot \|A^{-1}\|_2$ be the condition number of A, where $\|\cdot\|_2$ stands for the spectral norm. Throughout the talk we assume for simplicity that IEEE standard 754 binary64 (formerly, double precision) floating-point arithmetic is used. Let $\mathbf{u}$ denote the rounding error unit of floating-point arithmetic, which is equal to 2^{-53}.

We are concerned with practically proving the nonsingularity of A (if A is nonsingular) and then obtaining a forward error bound of an approximate solution $\widetilde{x}$ of a linear system $Ax = b$ to the exact solution $x^* = A^{-1}b$ such that $|x_i^* - \widetilde{x}_i| \leq \epsilon_i$ for $1 \leq i \leq n$ by the use of verified numerical computations. For this purpose estimating $\|A^{-1}\|$ is essential for some matrix norm.

For dense linear systems there are several efficient methods for this purpose (e.g. [1,4]). For sparse systems things are much different; Fast and efficient verification for large sparse linear systems is still difficult in terms of both computational complexity and memory requirements except a few cases where it is known in advance or to be proved that A belongs to a certain special matrix class, e.g. diagonally dominant and M-matrix (see, e.g. [3]). Moreover, a super-fast verification method proposed in [7] is applied to the case where A is sparse, symmetric and positive definite. However, to our knowledge, few methods are known in case of A being a general sparse matrix except methods by Rump [5]. Thus the verification for sparse systems of linear (interval) equations is known as one of the important open problems posed by Neumaier in

Grand Challenges and Scientific Standards in Interval Analysis [2]. Moreover, Rump [6] formulated the following challenge:

> Derive a verification algorithm which computes an inclusion of the solution of a linear system with a general symmetric sparse matrix of dimension 10000 with condition number 10^{10} in IEEE 754 double precision, and which is no more than 10 times slower than the best numerical algorithm for that problem.

In the present talk we try to partially solve the problem for symmetric but not necessarily positive definite input matrices, and also to a certain extent for nonsymmetric matrices. Namely, we assume that A is large, e.g. $n \geq 10000$, and sparse, possibly $\kappa(A) > 1/\sqrt{\mathbf{u}}$.

We survey some existing verification methods for sparse linear systems. After that, we propose new verification methods. Numerical results are also presented.

References:

[1] A. Neumaier, A simple derivation of the Hansen-Bliek-Rohn-Ning-Kearfott enclosure for linear interval equations, *Reliable Computing*, 5 (1999), pp. 131–136, and Erratum, *Reliable Computing*, 6 (2000), p. 227.

[2] A. Neumaier, Grand challenges and scientific standards in interval analysis, *Reliable Computing*, 8 (2002), pp. 313–320.

[3] T. Ogita, S. Oishi, Y. Ushiro, Fast verification of solutions for sparse monotone matrix equations, *Computing Suppl.*, 15 (2001), pp. 175–187.

[4] S. Oishi, S.M. Rump, Fast verification of solutions of matrix equations, *Numer. Math.*, 90 (2002), No. 4, pp. 755–773.

[5] S.M. Rump, Validated solution of large linear systems, *Computing Suppl.*, 9 (1993), pp. 191–212.

[6] S.M. Rump, Verification methods: Rigorous results using floating-point arithmetic, *Acta Numerica*, 19 (2010), pp. 287–449.

[7] S.M. Rump, T. Ogita, Super-fast validated solution of linear systems, *J. Comp. Appl. Math.*, 199, No. 2 (15 February 2007), pp. 199–206.

Error estimates with explicit constants for Sinc quadrature and Sinc indefinite integration over infinite intervals

Tomoaki Okayama

Graduate School of Economics, Hitotsubashi University
2-1 Naka, Kunitachi, Tokyo 186-8601, Japan
tokayama@econ.hit-u.ac.jp

Keywords: Sinc numerical methods, infinite interval, error estimates

The *Sinc quadrature* has been known as an efficient numerical integration formula for definite integrals, $\int_a^b f(x)\,\mathrm{d}x$, if the following conditions are met: (i) $(a,\,b) = (-\infty,\,\infty)$, and (ii) $|f(x)|$ decays exponentially as $x \to \pm\infty$. In other cases, users should employ an appropriate variable transformation $x = \psi(t)$, i.e., the given integral is transformed as $\int_a^b f(x)\,\mathrm{d}x = \int_{-\infty}^{\infty} f(\psi(t))\psi'(t)\mathrm{d}t$, so that those two conditions are met. Stenger [2] considered the following cases:

1. $(a,\,b) = (-\infty,\,\infty)$, and $|f(x)|$ decays algebraically as $x \to \pm\infty$,
2. $(a,\,b) = (0,\,\infty)$, and $|f(x)|$ decays algebraically as $x \to \infty$,
3. $(a,\,b) = (0,\,\infty)$, and $|f(x)|$ decays (already) exponentially as $x \to \infty$,
4. The interval $(a,\,b)$ is finite,

and gave the concrete transformations for each case:

$$\begin{aligned}
\psi_{\mathrm{SE1}}(t) &= \sinh(t),\\
\psi_{\mathrm{SE2}}(t) &= \mathrm{e}^t,\\
\psi_{\mathrm{SE3}}(t) &= \operatorname{arcsinh}(\mathrm{e}^t),\\
\psi_{\mathrm{SE4}}(t) &= \frac{b-a}{2}\tanh(t/2) + \frac{b+a}{2},
\end{aligned}$$

which are called the *Single-Exponential (SE) transformations*. Takahasi–Mori [3] have proposed the following improved transformations:

$$\begin{aligned}
\psi_{\mathrm{DE1}}(t) &= \sinh[(\pi/2)\sinh t],\\
\psi_{\mathrm{DE2}}(t) &= \mathrm{e}^{(\pi/2)\sinh t},\\
\psi_{\mathrm{DE3}}(t) &= \mathrm{e}^{t-\exp(-t)},\\
\psi_{\mathrm{DE4}}(t) &= \frac{b-a}{2}\tanh(\pi\sinh(t)/2) + \frac{b+a}{2},
\end{aligned}$$

which are called the *Double-Exponential (DE) transformations*. Error analyses of them have been given [2,4] in the following form:

$$|\mathrm{Error(SE)}| \le C\mathrm{e}^{-\sqrt{2\pi d\alpha N}}, \quad |\mathrm{Error(DE)}| \le C\mathrm{e}^{-\pi dN/\log(8dN/\alpha)},$$

where α denotes the decay rate of the integrand, and d indicates the width of the domain in which the transformed integrand is analytic, and C is a constant independent of N. In view of the inequalities, we notice that the accuracy of the approximation can be guaranteed if the constant C is explicitly given in a computable form. In fact, the explicit form of C has been revealed in the case 4 (the interval is finite) [1], and the result was used for verified automatic integration [5].

The purpose of this study is to reveal the explicit form of C's in the remaining cases: 1–3 (the interval is infinite), which enables us to bound the errors. Numerical experiments that confirm the results will be shown in this talk.

In addition to the *Sinc quadrature* described above, the similar results can be given for the *Sinc indefinite integration* for indefinite integrals $\int_a^\xi f(x)\,\mathrm{d}x$, which will also be reported in this talk. For this (indefinite) case, a new variable transformation $\psi_{\mathrm{DE3}\ddagger}(t) = \log(1+\mathrm{e}^{\pi \sinh t})$ is proposed for the DE transformation in the case 3, so that its inverse can be written with elementary functions.

References:

[1] T. Okayama, T. Matsuo and M. Sugihara, Error estimates with explicit constants for Sinc approximation, Sinc quadrature and Sinc indefinite integration, *Mathematical Engineering Technical Reports 2009-01*, The University of Tokyo, 2009.

[2] F. Stenger, *Numerical Methods Based on Sinc and Analytic Functions*, Springer-Verlag, New York, 1993.

[3] H. Takahasi and M. Mori, Double exponential formulas for numerical integration, *Publications of the Research Institute for Mathematical Sciences*, Kyoto University, 9 (1974), pp. 721–741.

[4] K. Tanaka, M. Sugihara, K. Murota and M. Mori, Function classes for double exponential integration formulas, *Numerische Mathematik*, 111 (2009), pp. 631–655.

[5] N. Yamanaka, T. Okayama, S. Oishi and T. Ogita, A fast verified automatic integration algorithm using double exponential formula, *Nonlinear Theory and Its Applications*, IEICE, 1 (2010), pp. 119–132.

On methodological foundations of interval analysis of empirical dependencies

Nikolay Oskorbin and Sergei Zhilin

Altai State University, 61, Lenin ave., 6560049, Barnaul, Russia
osk46@mail.ru, sergei@asu.ru

Keywords: methodology, experimental data processing, interval observations, inconsistent data

We consider methodological issues of the usage of interval analysis as a method for mathematical modeling of real-world processes and experimental data processing.

Let for a process described by a linear dependence $y = x\beta^*$ with output variable $y \in \mathbb{R}$, input variables $x \in \mathbb{R}^p$ and unknown true values of parameters $\beta^* \in \mathbb{R}^p$, we have a set of interval observations $\{(\mathbf{Y}_j, \mathbf{X}_j) \mid j = 1, \ldots, N\}$. The problem of estimation of the process parameters is reduced to finding the united solution set $B(N)$ of the interval linear system $\mathbf{Y} = \mathbf{XB}$. The set of possible parameters values $B(N)$ is also called the information set. If underlying assumptions about the structure of dependence and validity of interval observations are strongly fulfiled, the inclusion $\beta^* \in B(N) \neq \varnothing$ holds. This inclusion is a fundamental foundation of reliability of the constructed parameters estimates.

This interval approach to modeling of processes is developed by a number of authors and competes with probabilistic approach on efficiency of estimates in a number of applications. Using the interval approach benefits the simplicity and reliability of data and knowledge, flexibility in employment of a priori information, possibility of state estimation, forecasting and choosing control actions for a modeled process. There are applications of the interval approach to the modeling of nonlinear processes and processes with an inner noise.

Essential difficulties arise in a case when we are not sure about the underlying assumptions of the method, and hence there is no good cause to state $\beta^* \in B(N)$ even if $B(N) \neq \varnothing$. The fulfilment of assumptions cannot be verified using only existing data and knowledge. An analogous problem situation often takes place when statistical probabilistic methods are used for data analysis.

When looking for a way out of the situation, it is necessary to take into account the following principles.

1. It is impossible to obtain reliable estimates of process parameters using an inconsistent set of data and knowledge about the process.

2. None of the inner mathematical needs can be a ground for any kind of modifications of analyzed data and knowledge.

In the authors' opinion, the methodologically correct way out of the impasse involves a discovering of inconsistencies in the data and knowledge and their elimination after appraisal by application domain experts.

The proposed way is implemented in a case when $B(N) = \varnothing$ [1–3]. Widening of some or, in general, all interval variables allows us to obtain an information set $B(N,k) \supset B(N)$ which is determined by a set k of expansion coefficients for interval variables. The expanded set $B(N,k)$ is formed by elementary information portions $B_j(N,k), B(N,k) = \cap_{j=1}^{N} B_j(N,k)$. Choosing k we can obtain $B(N,k) \neq \varnothing$ and detect portions which need domain expert's appraisal.

Besides, to discover inconsistencies one can

- estimate an informational value of each portion of data and knowledge against the selected basic set;
- relate the volume of $B(N,k)$ to the value of N;
- investigate the dynamics of the volume of $B(N,k)$ depending on N.

Implementation of the proposed approach demands on the development of suitable mathematical tools and accumulation of experience in specific applications. We show model and real-world case studies to illustrate the approach.

The authors wish to express their gratitude to Professor S.P. Shary for his initiative to prepare this talk.

References:

[1] N.M. Oskorbin, A.V. Maksimov, S.I. Zhilin, Construction and analysis of empirical dependencies using uncertainty center method, *Transactions of Altai State University*, 1 (1998), pp. 35–38, (In Russian).

[2] S.I. Zhilin, Simple method for outlier detection in fitting experimental data under interval error, *Chemometrics and Intellectual Laboratory Systems*, 88 (2007), No. 1, pp. 60–68.

[3] S.P. Shary, Solvability of interval linear equations and data analysis under uncertainty *Automation and Remote Control*, 73 (2012), No. 2, pp. 310–322.

Performance comparison of accurate matrix multiplication

Katsuhisa Ozaki[1] and Takeshi Ogita[2]

[1]Shibaura Institute of Technology
307 Fukasaku, Minumaku, Saitama-shi, Saitama 337-8570, Japan
[2]Tokyo Woman's Christian University
2-6-1 Zempukuji, Suginami-ku, Tokyo 167-8585, Japan
ozaki@sic.shibaura-it.ac.jp

Keywords: accurate computations, matrix multiplication

This talk discusses accurate numerical algorithms for matrix multiplication. Accurate matrix multiplication is useful for verified numerical computations, especially for verified solutions of systems of equations including proofs of matrix non-singularity and Krawczyk's method (See, for example, Chapter 4 in [1], Section 4 in [2] and Section 10 in [5]). Let A be an m-by-n matrix and B be an n-by-p matrix with floating-point entries as defined by the IEEE 754-2008 standard, respectively. If the matrix multiplication AB is evaluated by floating-point arithmetic, then an inaccurate result may be obtained due to accumulation of rounding errors. The aim is to develop an algorithm outputting a computed result C such that

$$|C - AB| \leq u|AB|, \tag{1}$$

where u is the relative rounding error unit, for example, $u = 2^{-53}$ for binary64. The inequality implies that C is as accurate as if AB were first evaluated exactly and the results were rounded to the nearest floating-point numbers componentwise. To achieve (1), a simple way is to apply an accurate summation algorithm, such as the algorithm proposed by Rump-Ogita-Oishi in [3], for each dot product in matrix multiplication since a dot product can be transformed into a sum of $2n$ floating-point numbers by so-called error-free transformations. The above-mentioned algorithm is called Algorithm-A in this abstract.

Recently, an error-free transformation of matrix multiplication [4] is developed by the authors. It transforms a product of two floating-point matrices into an unevaluated sum of floating-point matrices, namely

$$AB = \sum_{i=1}^{q} C^{(i)}, \quad q \in \mathbb{N},$$

where each $C^{(i)}$ is an m-by-p floating-point matrix. By using this transformation and the accurate summation algorithm given in [3], it is possible to develop an algorithm which achieves (1). Namely, the error-free transformation is first applied. Next, the accurate summation algorithm [3] is applied to the sum of matrices componentwise. The above-mentioned algorithm is called Algorithm-B in this abstract.

First, we compare computational performance and efficiency of parallelization of the two algorithms by numerical examples. If there is not much difference in the order of magnitude among elements in the same row of A and those in the same column of B, then Algorithm-B is much faster than Algorithm-A. Otherwise, Algorithm-A is faster than Algorithm-B.

A drawback of Algorithm-B is to require a large amount of working memory. To overcome this problem, we develop a new algorithm which reduces the amount of working memory by block matrix computations and reuse of working memory. We incorporate these approaches into Algorithm-B. The proposed algorithm is called Algorithm-C in this abstract. It is shown by numerical examples that such approaches for saving working memory are efficient and do not slow down the computational performance significantly. For example, if the required working memory for Algorithm-C is reduced into 1/5 of that for Algorithm-B, then Algorithm-C is only 20 % slower than Algorithm-B in the numerical examples.

References:

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

[2] A. Frommer, B.Hashemi, Verified error bounds for solutions of Sylvester matrix equations, *Linear Algebra and Its Applications*, 436, No. 2 (2012), pp. 405–420.

[3] S. M. Rump, T. Ogita, S. Oishi, Accurate Floating-Point Summation Part II: Sign, K-fold Faithful and Rounding to Nearest, *SIAM J. Sci. Comput.*, 31, No. 2 (2008), pp. 1269–1302.

[4] K. Ozaki, T. Ogita, S. Oishi, S. M. Rump, Error-free transformation of matrix multiplication by using fast routines of matrix multiplication and its applications, *Numerical Algorithms*, 59, No. 1 (2012), pp. 95–118.

[5] S. M. Rump, Verification methods: Rigorous results using floating-point arithmetic, *Acta Numerica*, 19 (2010), pp. 287–449.

Interval methods for global unconstrained optimization: a software package

Valentin N. Panovskiy

Moscow Aviation Institute NRU
4, Volokolamskoe shosse
125993 Moscow, Russia
mai@mai.ru

Keywords: interval arithmetic, range dichotomy, cutoff of virtual values, stochastic cutoff, changing directions, unconstrained global optimization

This work is devoted to the interval methods for the solution of unconstrained global optimization problems. Use of interval analysis as a base component of the methods gives essential advantages (e.g., less requirements to the problem statement [1]).

Method of range dichotomy and cutoff methods use the approach previously elaborated in [4]. The feature of our technique is that it does not use subdivision of the function domain, and only divided the range of values. According to the terminology from [4], all the components of the domain are "mute" in this case. Our methods use a construction called invertor that requires, on input, the objective function, a target interval, a box and an accuracy parameter. The invertor returns a set of boxes on which the objective function returns an interval which belongs to the target interval or has non-empty intersection with it (in this case, the width of the box must satisfy an accuracy constraint).

On the first step of the method of range dichotomy, we evaluate the range of the function over the search area and consider it as the target interval. Further on each iteration the target interval is bisected. Then we apply the invertor to the first part. If the invertor returns a nonempty set, we check the accuracy condition. In case of its failure, the first part is considered as a new target interval, and a new iteration of the method starts. Otherwise, the returned set has a box that contains the global minima of function. If the invertor returns an empty set, the second part is considered as a new target interval, and the method begins new iteration.

Strategies for the cutoff methods are similar to the strategy for the range dichotomy method. The only distinction is the presence of a tightening stage,

which comes before the iterative part. After the range evaluation, we apply the operator of compression to it, which deletes a part of virtual values from the evaluation. Then the method works as the previous method. The stage described tries to reduce the target interval. This accelerates convergence.

Strategy of the changing directions method consists of constant analysis of the best box and all the potential best boxes stored in the memory. To explain the work of the method, it is necessary to introduce the concept of the double buffer. We will consider the double buffer as a set of the ordered pairs "box"$\mathbf{x}$"enclosure of the range". One box is considered better than another, if it has a smaller lower boundary and width of its range enclosure. On each iteration, the best box, which becomes target box, is selected from the double buffer. Further, the target box is divided at its midpoint. Then we evaluate the range of newly organized boxes and restructure the double buffer. This method works while there is at least one box in the double buffer that do not meet the accuracy requirement.

The methods were not only theoretically substantiated and have their convergence proved. Besides, they were tested on benchmarks of unconstrained global optimization problems (global minimization of the Schwefel's, Griewank's, Ackley's functions, etc.). All the methods compute the box which contains the point of global minima or is close enough to it according to accuracy specification.

The methods have been implemented, using C#, as a software package that can solve unconstrained global optimization problem and automatically analyze the efficiency of the methods.

References:

[1] L. Jaulin, M. Kieffer, *Applied Interval Analysis*, Springer-Verlag, London, 2001.

[2] R.E. Moore, R.B. Kearfott, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[3] S.P. Shary, *Finite-dimensional Interval Analysis*, Novosibirsk, XYZ, 2012. (in Russian)

[4] S.P. Shary, A surprising approach in interval global optimization, *Reliable Computing*, 7 (2001), No. 6, pp. 497–505.

[5] V.N. Panovskiy, Application of the interval analysis for the search of the global extremum of functions, *Trudy MAI*, No. 51 (2012), `http://www.mai.ru/science/trudy/published.php?eng=Y&ID=28953`

Application of redundant positional notations for increasing arithmetic algorithms scalability

Anatoly V. Panyukov

South Ural State University
76, Lenin ave., 454080 Chelyabinsk, Russia
a_panyukov@mail.ru

Keywords: integer arithmetic, positional notation, hybrid scalability

For algorithmic analysis of large scale unstable problems (considered e.g. in [1]), the library "Exact computation" [2–4] provides helpful instruments for distributed computing environment. Further increasing of effectiveness of such software is possible for heterogeneous computing environment that allows one to parallelize execution of local arithmetic operations over a large number of threads. Application of redundant positional notations is also an effective approach for increasing arithmetic algorithms scalability.

References:

[1] A.V. Panyukov, V.A. Golodov, Calculating of pseudo-solution of linear equation systems with interval uncertainty of coefficients, *Algorithmic Analysis of Unstable Problems: Abstracts of the International Conference, 2011, October 31 – November 5, Yekaterinburg*, Institute of Mathematics and Mechanics of Ural Branch of Russian Academy of Sciences, 2011, pp. 262–263 (in Russian).

[2] A.V. Panyukov, V.V. Gorbik, Using massively parallel computations for absolutely precise solution of the linear programming problems, *Automation and Remote Control*, 73 (2012), No. 2, pp. 276–290.

[3] A.V. Panyukov, V.V. Gorbik, Exact and guaranteed accuracy solutions of linear programming problems by distributed computer systems with MPI, *Tambov University Reports. Series: Natural and Technical Sciences*, 15 (2010), No. 4, pp. 1392–1404.

[4] V.A. Golodov, Distributed symbolic rational calculation on x86 and x64 CPUs, *Proceedings of International Conference "Parallel Computing Technologies", Novosibirsk, March 26 – March 30, 2012)*, South Ural State University Press, Chelyabinsk, p. 774 (in Russian).

Computing the best possible pseudo-solutions to interval linear systems of equations

Anatoly V. Panyukov[1] and Valentin A. Golodov[2]

South Ural State University
76, Lenin ave., 454080 Chelyabinsk, Russia
[1]a_panyukov@mail.ru, [2]avaksa@gmail.com

Keywords: interval uncertainty, interval linear equations, linear programming, massively parallel computations, pseudo-solution, tolerable solution set

We consider solution of the linear equations set $Ax = b$ under interval uncertainty of its elements, which can belong to the interval $n \times n$-matrix A and interval right-hand side n-vector b. That is, we only know that $a_{ij} \in \boldsymbol{a}_{ij} = [\underline{a}_{ij}, \overline{a}_{ij}]$ and $b_i \in \boldsymbol{b}_i = [\underline{b}_i, \overline{b}_i]$ for all $i, j = 1, 2, \ldots, n$.

As a solution to the disturbed linear systems, we consider a point from the *tolerable solution set* $\Xi_{tol}(\boldsymbol{A}, \boldsymbol{b}) = \{\, x \in \mathbb{R}^n \mid (\forall A \in \boldsymbol{A})(\, Ax \in \boldsymbol{b})\}$. A substantial contribution to the theory of the tolerable solution set and tolerance problem has been made by J. Rohn [1] and S. Shary [2].

For real-life situations, we often have $\Xi_{tol}(\boldsymbol{A}, \boldsymbol{b}) = \varnothing$. By parity of reasoning, we introduce pseudo-solution concept [3] for interval linear equation systems. Let $\boldsymbol{b}(z) = [\,\underline{b} - z|\underline{b}|, \overline{b} + z|\overline{b}|\,]$, $z > 0$, then we denote $z^* = \inf\{z \mid \Xi_{tol}(\boldsymbol{A}, \boldsymbol{b}(z)) \neq \varnothing\}$. *Pseudo-solution* of the original system $Ax = b$, by definition, is an inner point of tolerable solution set $\Xi_{tol}(\boldsymbol{A}, \boldsymbol{b}(z^*))$. Extending Rohn's representation of the tolerable solution set [1], we deduce

Theorem 3 *A solution* x^{+^*}, $x^{-^*} \in \mathbb{R}^n$, $z^* \in \mathbb{R}$ *to the linear programming problem*

$$
\begin{gathered}
z \to \min_{x^+,\ x^-,\ z}, \\
\sum_{j=1}^{n} (\underline{a}_{ij} x_j^+ - \overline{a}_{ij} x_j^-) \geq \underline{b}_i - z|\underline{b}_i|,\ \ i = 1, 2, \ldots, n, \\
\sum_{j=1}^{n} (\overline{a}_{ij} x_j^+ - \underline{a}_{ij} x_j^-) \leq \overline{b}_i + z|\overline{b}_i|,\ \ i = 1, 2, \ldots, n, \\
x_j^+, x_j^-, z \geq 0,\ \ j = 1, 2, \ldots, n,
\end{gathered}
\tag{1}
$$

exists, and $x^* = x^{+^*} - x^{-^*}$ *is a pseudo-solution to the linear equations set* $\boldsymbol{A}x = \boldsymbol{b}$.

Linear programming problem (1) is strongly degenerate, and solving it with the use of the standard floating point data types is impossible, since cycling is not efficiently eliminated by known anticycling tools under approximate computations. The cycling and accuracy problems can be solved by using symbolic rational-fractional computations [5]. To accelerate the computations, we may avail ourselves of massively parallel computations [6].

In our talk, we discuss the solutions for a number of the above problems, present a new theory and techniques elaborated in the course of our research.

The work is performed under support of Russian foundation for basic research (project No 10-07-96003-r_ural_a).

References:

[1] J. Rohn, Inner solutions of linear interval systems, in *Interval Mathematics 1985*, K. Nickel, ed., Springer Verlag, New York, 1986, pp. 157–158.

[2] S.P. Shary, Solving the linear interval tolerance problem, *Mathematics and Computers in Simulation*, 39 (1995), pp. 53–85.

[3] V.Ya. Arsenin, On ill-posed problems, *Russian Math. Surveys*, 31 (1976), No. 6, pp. 93–107.

[4] A.V. Panyukov, V.A. Golodov, Calculating of pseudo-solution of linear equation systems with interval uncertainty of coefficients, *Algorithmic Analysis of Unstable Problems: Abstracts of the International Conference, October 31 – November 5, 2011, Yekaterinburg*, Institute of Mathematics and Mechanics of Ural Branch of Russian Academy of Sciences, 2011, pp. 262–263 (in Russian).

[5] V.A. Golodov, Distributed symbolic rational calculation on x86 and x64 CPUs, *Proceedings of International Conference "Parallel Computing Technologies", Novosibirsk, March 26 – March 30, 2012)*, South Ural State University Press, Chelyabinsk, p. 774 (in Russian).

[6] A.V. Panyukov, V.V. Gorbik, Using massively parallel computations for absolutely precise solution of the linear programming problems, *Automation and Remote Control*, 73 (2012), No. 2, pp. 276–290.

Properties and estimations of parametric AE-solution sets

Evgenija D. Popova

Institute of Mathematics and Informatics, BAS
Acad. G. Bonchev str., block 8
1113 Sofia, Bulgaria
epopova@math.bas.bg

Keywords: linear systems, dependent data, AE-solution set, characterization, estimations, tolerable solution set, controllable solution set

Consider linear systems $A(p)x = b(p)$, where the elements of the matrix and right-hand side vector are linear functions of uncertain parameters varying within given intervals, $p_i \in [p_i]$, $i = 1, \ldots, k$. 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 the parametric AE-solution sets, which are defined by universally and existentially quantified parameters, and the former precede the latter. For two disjoint sets of indexes $\mathcal{E}$ and $\mathcal{A}$, such that $\mathcal{E} \cup \mathcal{A} = \{1, \ldots, k\}$,

$$\begin{aligned}\Sigma^p_{AE} &= \Sigma(A(p_{\mathcal{A}}, p_{\mathcal{E}}), b(p_{\mathcal{A}}, p_{\mathcal{E}}), [p]) \\ &:= \{x \in \mathbf{R}^n \mid (\forall p_{\mathcal{A}} \in [p_{\mathcal{A}}])(\exists p_{\mathcal{E}} \in [p_{\mathcal{E}}])(A(p)x = b(p))\}.\end{aligned}$$

Parametric AE-solution sets generalize the parametric united solution set and the nonparametric AE-solution sets.

In this talk we present three types of characterizations for the parametric AE-solution sets: set-theoretic characterization, characterization in form of interval inclusions and characterization by Oettli-Prager-type absolute-value inequalities. The focus of the characterizations is on how to obtain explicit description of a parametric AE-solution set in the form of Oettli-Prager-type inequalities. The description is explicit for some classes of parametric AE-solution sets and in the general case can be obtained by a Fourier-Motzkin-type algorithmic procedure eliminating the existentially quantified parameters.

The characterizations of parametric AE-solution sets inspire proving various properties of these solution sets and designing some numerical methods for their

outer or inner estimation. We will present some important inclusion relations between classes of parametric AE-solution sets, where the relations are determined by the type of the parameter dependencies. Various other properties like the shape of a parametric AE-solution set and some criteria for nonempty and bounded solution set will be also discussed. Special consideration is provided for the parametric tolerable solution set

$$\begin{aligned}\Sigma^p_{tol} &= \Sigma(A(p_{\mathcal{A}}), b(p_{\mathcal{E}}), [p]) \\ &:= \{x \in \mathbf{R}^n \mid (\forall p_{\mathcal{A}} \in [p_{\mathcal{A}}])(\exists p_{\mathcal{E}} \in [p_{\mathcal{E}}])(A(p_{\mathcal{A}})x = b(p_{\mathcal{E}}))\}\end{aligned}$$

and for the parametric controllable solution set

$$\begin{aligned}\Sigma^p_{cont} &= \Sigma(A(p_{\mathcal{E}}), b(p_{\mathcal{A}}), [p]) \\ &:= \{x \in \mathbf{R}^n \mid (\forall p_{\mathcal{A}} \in [p_{\mathcal{A}}])(\exists p_{\mathcal{E}} \in [p_{\mathcal{E}}])(A(p_{\mathcal{E}})x = b(p_{\mathcal{A}}))\}.\end{aligned}$$

Some numerical methods for outer and inner estimations of parametric AE-solution sets will be also presented. The properties of these methods for estimating the parametric tolerable and the parametric controllable solution sets are compared. We show that in some cases the parametric approach provides a more efficient solution for some nonparametric problems than the existing nonparametric approaches. Numerical examples accompanied by graphic representations will illustrate the solution sets and their properties or the numerical methods and their properties. Some of the properties (or methods) are new, most of them generalize known properties (or methods) for nonparametric AE-solution sets, studied by I. Sharaya, S. Shary and others. Presenting some first results about parametric AE-solution sets, the talk will also outline some open problems and directions of possible further research.

An interval approach to recognition of numerical matrices

Alexander Prolubnikov

Omsk State University
55-A, Mira ave.
644077 Omsk, Russia
a.v.prolubnikov@mail.ru

Keywords: pattern recognition, outer estimation

In our work, we propose a new interval approach to recognition of numerical matrices.

The registration of data by technical means is often complicated by measurement errors or noise that interfere the registration process. If it is known that the data presented in the form of a matrix are distorted by noise or registered with errors from a given set of pattern matrices, a common problem is to recognize the patterns under specified constraints on the noise or measurement errors.

An obvious example of the problem under consideration is recognition of raster images. Existing algorithms of recognition of raster images, such as, for example, those using neural networks [1], parametric algorithms, algorithms on the basis of the theory of morphological analysis [2] include a preliminary learning stage, during which the algorithm should be taught from images of the object obtained under various registration conditions. The purpose of the learning process is to fix some characteristics of the image, which can be used for subsequent recognition. The distinctive feature of the approach to the recognition we propose in our work from the traditional ones is the absence of the learning stage within the recognition algorithm.

The problem is formulated as follows. We are given a set of N rectangular $m \times n$-matrices $S = \{A^{(k)}\}_{k=1}^{N}$ whose elements $a_{ij}^{(k)}$ are real numbers. The matrix A is obtained from some matrix $A^{(k_0)} \in S$ in the course of noising. It is known that the values of the elements of the matrices can vary within the intervals $[a_{ij}^{(k)} - \delta_{ij}, a_{ij}^{(k)} + \delta_{ij}]$, $\delta_{ij} \in \mathbb{R}_+$ $(i = \overline{1,m},\ j = \overline{1,n})$. We need to identify k_0.

We associate the input matrices with the systems of interval linear equations of the form

$$\boldsymbol{A}^{(k)} x = e,$$

where $e = (1, \ldots, 1)^\top$ and $\boldsymbol{A}^{(k)}$ is an interval matrix built for a pair of matrices A and $A^{(k)}$ $(k = \overline{1, N})$. Let $\Xi^{(k)}$ denote the united solution set for the k-th linear system of equations. Lebesgue measure $\mu(\widetilde{\Xi}^{(k)})$ of enclosures $\widetilde{\Xi}^{(k)}$ of the sets $\Xi^{(k)}$ are used as recognition heuristics.

We consider specific procedures that construct the matrices $\boldsymbol{A}^{(k)}$ and justify the choice of the right-hand side vectors of the interval linear systems. The matrices can be built so that they are amenable to usual interval numerical methods. Specifically, the matrices $\boldsymbol{A}^{(k)}$ may be done H-matrices by construction, which makes it possible to use interval Gauss-Seidel method for enclosing their united solution set [3]. The total computational complexity of the proposed recognition algorithm is estimated as $O(d^2)$, where $d = \max\{m, n\}$.

We present the results of computational experiments and comparison with the other known approaches to the recognition problem. It is worthwhile to note that our numerical experiments include the recognition of grayscale and monochrome images.

References:

[1] H. Demuth, M. Beale, *Neural Network Toolbox for Use with MATLAB. Users Guide. Version 4.* Available at `http://cs.mipt.ru/docs/comp/eng/develop/software/matlab/nnet/main.pdf` [Accessed April 15, 2012].

[2] E.A. Kirnos, *A Comparative Analysis of Morphological Methods of Image Interpretation*, The dissertation of the candidate of phys.-math. sciences: 05.13.18, Moscow, 2004.

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

Maximizing stability degree of interval systems using coefficient method

Maxim I. Pushkarev, Sergey A. Gaivoronsky

Tomsk Polytechnic University
pushkarev@tpu.ru

Keywords: interval system, stability degree, coefficient method

The work is devoted to maximization of stability degree for linear systems having interval parameter uncertainty. We propose to solve this problem by so-called coefficient method, using a sufficient conditions for specified stability degree η.

If we are given characteristic polynomial of a linear system, $A(s) = a_n s^n + a_{n-1}s^{n-1} + \ldots + a_0$, $a_n > 0$, then the stability degree conditions, with regard to the controller tunings vector $\overline{k}$, can be written as follows [1]:

$$\begin{cases} \lambda_i\left(\overline{k},\eta\right) = \dfrac{a_{i-1}(\overline{k})a_{i+2}(\overline{k})}{\left(a_i(\overline{k}) - a_{i+1}(\overline{k})(n-i-1)\eta\right)\left(a_{i+1}(\overline{k}) - a_{i+2}(\overline{k})(n-i-2)\eta\right)}, \\ \qquad\qquad i = \overline{1, n-2}; \\ f_l\left(\overline{k},\eta\right) = a_l(\overline{k}) - a_{l+1}(\overline{k})(n-l-1)\eta, \qquad l = \overline{1,\ n-1}; \\ g\left(\overline{k},\eta\right) = a_0(\overline{k}) - a_1(\overline{k})\eta + \dfrac{2a_2(\overline{k})\eta^2}{3}. \end{cases} \tag{1}$$

Varying η in the above expressions allows one to find its maximum value, which will be considered as a lower estimate η^* of the maximum stability degree. In such case, the synthesis problem is to choose the controller parameters $\overline{k}^*$ that provide the lower estimate of the maximum stability degree $\eta^*_{\max}$, which may be called "quasimaximum stability degree" of the system.

Increasing η in each expression from (1) by the controller tunings change is possible up to the value when $\lambda_i\left(\overline{k},\eta\right) = 0.465$. Thereby, determination of $\eta^*_{\max}$ and $\overline{k}^*$ requires $(n-2)$ solutions of the following system of equations

$$\begin{cases} \lambda_i\left(\overline{k},\eta\right) = \lambda^*, \quad i = \overline{1,\ n-2}; \\ \lambda_j\left(\overline{k},\eta\right) < \lambda^*, \quad j = \overline{1,\ n-2}, \quad j \neq i; \\ f_l\left(\overline{k},\eta\right) \geq 0, \qquad l = \overline{1,\ n-1}; \\ g\left(\overline{k},\eta\right) \geq 0. \end{cases} \tag{2}$$

At each step, this results in the maximum value of η^*, and then we can choose the maximum estimate among them.

In case the system has interval uncertainty in its parameters, the characteristic polynomial turns to the form $A(s) = \boldsymbol{a}_n s^n + \boldsymbol{a}_{n-1} s^{n-1} + \ldots + \boldsymbol{a}_0$, with intervals $\boldsymbol{a}_n > 0$, and $\underline{a_i(\overline{k})} \le a_i(\overline{k}) \le \overline{a_i(\overline{k})}$, $i = \overline{0,\ n}$. We apply interval methods to (2), which lead to the result which is valid for any value of $a_i(\overline{k})$. That is why it is necessary to set such values of $a_i(\overline{k})$ in $\lambda_i(\overline{k}, \eta)$ from (2), when $\lambda_i(\overline{k}, \eta)$ possess maximal values. Note that it is necessary to substitute such values of interval coefficients, which provide minimum of expressions $f_l(\overline{k}, \eta)$ and $g_l(\overline{k}, \eta)$. This way, the conditions (2) takes the form

$$\begin{cases} \dfrac{\overline{a_{i-1}(\overline{k})}\ \overline{a_{i+2}(\overline{k})}}{\left(\underline{a_i(\overline{k})} - a_{i+1}(\overline{k})(n-i-1)\eta\right)\left(a_{i+1}(\overline{k}) - \overline{a_{i+2}(\overline{k})}(n-i-2)\eta\right)} = \lambda^*, \\ \qquad i = \overline{1,\ n-2}; \\ \dfrac{\overline{a_{j-1}(\overline{k})}\ \overline{a_{j+2}(\overline{k})}}{\left(\underline{a_j(\overline{k})} - a_{j+1}(\overline{k})(n-j-1)\eta\right)\left(a_{j+1}(\overline{k}) - \overline{a_{j+2}(\overline{k})}(n-j-2)\eta\right)} < \lambda^*, \\ \qquad j = \overline{1,\ n-2}, \quad j \ne i; \\ \underline{a_l(\overline{k})} - \overline{a_{l+1}(\overline{k})}(n-l-1)\eta \ge 0, \quad l = \overline{1,\ n-1}; \\ \underline{a_0(\overline{k})} - \overline{a_1(\overline{k})}\eta + \dfrac{2\underline{a_2(\overline{k})}\eta^2}{3} \ge 0. \end{cases} \tag{3}$$

The coefficients $a_{i+1}(\overline{k})$ and $a_{j+1}(\overline{k})$ can take both minimal and maximal values. Therefore, for the controller synthesis, it is necessary to consider four Kharitonov polynomials and three additional polynomials from (4):

$$\begin{aligned} D_1(s) &= \overline{a_0} + \underline{a_1} s + \overline{a_2} s^2 + \overline{a_3} s^3 + \underline{a_4} s^4 + \overline{a_5} s^5 + \overline{a_6} s^6 + \ldots, \\ D_2(s) &= \overline{a_0} + \overline{a_1} s + \underline{a_2} s^2 + \overline{a_3} s^3 + \overline{a_4} s^4 + \underline{a_5} s^5 + \overline{a_6} s^6 + \ldots, \\ D_3(s) &= \underline{a_0} + \overline{a_1} s + \overline{a_2} s^2 + \underline{a_3} s^3 + \overline{a_4} s^4 + \overline{a_5} s^5 + \underline{a_6} s^6 + \ldots \end{aligned}$$

For the verification of the condition $g\left(\overline{k}, \eta\right)$, it is necessary to consider the additional polynomial $D_4(s) = \underline{a_0} + \overline{a_1} s + \underline{a_2} s^2 + \overline{a_3} s^3 + \underline{a_4} s^4 + \overline{a_5} s^5 + \underline{a_6} s^6 + \ldots$

References:

[1] B.N. Petrov, N.I. Sokolov, A.V. Lipatov, *Control Systems of Plants with Variable Parameters: Engineering Methods of Analysis and Design*, Mashinostroenie, Moscow, 1986. (in Russian)

Exponential enclosure techniques for the computation of guaranteed state enclosures in ValEncIA-IVP

Andreas Rauh[1], Ekaterina Auer[2], Ramona Westphal[1], and Harald Aschemann[1]

[1]Chair of Mechatronics
University of Rostock
D-18059 Rostock, Germany
{Andreas.Rauh,Ramona.Westphal,Harald.Aschemann}@uni-rostock.de

[2]Faculty of Engineering, INKO
University of Duisburg-Essen
D-47048 Duisburg, Germany
Auer@inf.uni-due.de

Keywords: ordinary differential equations, initial value problems, complex interval arithmetic, ValEncIA-IVP

ValEncIA-IVP is a verified solver which computes guaranteed enclosures for the solution of initial value problems (IVPs) for systems of ordinary differential equations (ODEs) [1,3]. Originally, this solver has been implemented on the basis of a simple iteration scheme that allows us to determine guaranteed state enclosures for IVPs with continuously differentiable right hand sides. These state enclosures are given by a numerically computed approximate solution (for example by means of a classic explicit Euler or Runge-Kutta method) with additive guaranteed error bounds. In [3], this solution procedure was extended by an exponential enclosure approach, allowing us to compute tighter state enclosures for asymptotically stable processes.

To efficiently exploit the exponential enclosure approach, the state equations are first decoupled as far as possible. For that purpose, linear dynamic systems are transformed into their real Jordan normal form. After that, the IVP is solved for the equivalent problem. Finally, guaranteed state enclosures in the original coordinates are determined by a suitable verified backward transformation.

However, this decoupling procedure does not manage to eliminate the wrapping effect in cases in which the (locally) linearized system model has an oscillatory behavior. This results from the fact that the transformed system matrix

of the linearized model is no longer purely diagonal but has a block diagonal structure. Geometrically, each block corresponds to a rotation (and scaling) of state enclosures between two subsequent time steps.

To eliminate the wrapping effect that originates from this rotation, the above-given real-valued problem with a block diagonal system matrix can be replaced by a transformation into a complex-valued diagonal form if the linear system model does not have multiple eigenvalues. In this contribution, a solution procedure for the computation of state enclosures is presented which operates on complex-valued IVPs in the corresponding normal form. This allows us to determine contracting state enclosures for linear ODE systems with asymptotically stable, conjugate complex eigenvalues of multiplicity one by means of a complex-valued exponential enclosure approach with a suitable backward transformation onto the original problem.

The theory is demonstrated using selected real-life applications from the field of control engineering. Moreover, examples are presented to show the benefits of applying the corresponding transformation also to linear dynamic systems with multiple eigenvalues and uncertain parameters as well as to nonlinear processes which exhibit oscillatory dynamics. Finally, conclusions and an outlook on how to extend the corresponding techniques to solving IVPs for differential-algebraic equations in ValEncIA-IVP [4] are given.

References:

[1] E. Auer, A. Rauh, E.P. Hofer, and W. Luther, Validated modeling of mechanical systems with SmartMOBILE: improvement of performance by ValEncIA-IVP, *Lecture Notes in Computer Science*, Vol. 5045, Springer, 2008, pp. 1–27.

[2] A. Rauh and H. Aschemann, Structural analysis for the design of reliable controllers and state estimators for uncertain dynamical systems, In: M. Günther, A. Bartel, S. Schöps, M. Striebel, and M. Brunk (Eds.), *Progress in Industrial Mathematics at ECMI 2010*, Springer, Mathematics in Industry, 17 (2012), pp. 263–269.

[3] A. Rauh and E. Auer, Verified simulation of ODEs and DAEs in ValEncIA-IVP, *Reliable Computing*, 15 (2011), No. 4, pp. 370–381.

[4] A. Rauh, M. Brill, C. Günther, A novel interval arithmetic approach for solving differential-algebraic equations with ValEncIA-IVP, *International Journal of Applied Mathematics and Computer Science*, 19 (2009), No. 3, pp. 381–397.

Interval methods for model-predictive control and sensitivity-based state estimation of solid oxide fuel cell systems

Andreas Rauh, Luise Senkel, Thomas Dötschel, Julia Kersten, and Harald Aschemann

Chair of Mechatronics, University of Rostock
D-18059 Rostock, Germany
{Andreas.Rauh,Luise.Senkel,Thomas.Doetschel,Julia.Kersten,
Harald.Aschemann}@uni-rostock.de

Keywords: interval arithmetic, predictive control, verified sensitivity-based state estimation, real-time implementation, experimental validation

Control-oriented mathematical models for the thermal behavior of solid oxide fuel cells (SOFCs) [1] are characterized by the fact that internal parameters can be determined only within certain intervals. This is caused by simplifications which are necessary to make mathematical system models usable for the synthesis of control strategies such that they can be evaluated in real time. Furthermore, temperature uncertainty due to limited measurement facilities in the interior of a fuel cell stack module as well as limited knowledge about the spatial distribution of the electrochemical reaction processes can be expressed by interval parameters in a natural way. Finally, disturbances result from the variation of electrical load demands which are a-priori unknown to the controller. To determine control strategies which prevent the violation of constraints on the admissible maximum operating temperatures, it is reasonable to derive control laws directly accounting for the above-mentioned uncertainties.

The basic approaches considered for this purpose are model-predictive control as well as sensitivity-based state and parameter estimation. Both procedures are extended by using interval arithmetic to obtain a verified implementation which directly accounts for uncertain variables with a bounded range.

Model-predictive control approaches are well-known means to stabilize dynamic systems and to compute input signals online which allow for the tracking of desired state trajectories. These control procedures, which are partially implemented by means of algorithmic differentiation, are inherently robust and

can, therefore, be used to compensate unknown disturbances to some extent [2]. This holds even if the disturbances are neglected during the derivation of the predictive control strategy.

In this contribution, different verified extensions are described for the design of model-predictive control strategies. These controllers are implemented by applying interval arithmetic procedures in real time. The use of interval arithmetic allows one to design controllers which definitely prevent the violation of predefined tolerance intervals around the desired state trajectories under consideration of predefined limitations for the actuator operating range [3].

Like any other interval arithmetic procedure for the evaluation of dynamic system models, interval-based predictive control procedures suffer from overestimation due to multiple dependencies on identical interval variables as well as the wrapping effect. In the case of predictive control procedures, this overestimation may lead to control strategies which are more conservative than necessary. To detect overestimation in the interval evaluation of the predictive control procedure, physical conservation properties (derived on the basis of the first law of thermodynamics) can be exploited in an algebraic consistency test that can be evaluated in real time in parallel to the computation of the control law.

Finally, the implementation of the interval-based predictive control procedure is described for a SOFC test rig available at the Chair of Mechatronics at the University of Rostock. Here, non-measured state variables are reconstructed by a verified sensitivity-based observer [4]. This contribution is concluded by an outlook on future work focusing on algorithmic improvements for a reliable real-time capable control as well as state and parameter estimation.

References:

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

[2] Y. Cao and W.-H. Chen, Automatic differentiation based nonlinear model predictive control of satellites using magneto-torquers, *Proc. of IEEE Conf. on Industrial Electronics and Applications, ICIEA 2009*, Xi'an, China, 2009, pp. 913–918.

[3] A. Rauh, J. Kersten, E. Auer, and H. Aschemann, Sensitivity-based feedforward and feedback control for uncertain systems, *Computing*, 2012, No. 2–4, pp. 357–367.

[4] A. Rauh, L. Senkel, and H. Aschemann, Sensitivity-based state and parameter estimation for fuel cell systems, *Proc. of 7th IFAC Symposium on Robust Control Design*, Aalborg, Denmark, 2012.

On computer-aided proof of the correctness of non-polynomial oscillator realization of the generalized Verma module for non-linear superalgebras

Alexander Reshetnyak[1], Andrei Kuleshov[2]
and Vladimir Starichkov[3]

[1]Institute of Strength Physics and Materials Science SB RAS
2/4, Akademicheskii ave., 634021 Tomsk, Russia
[2] Elecard Company, 3 Razvitiya ave., 634021 Tomsk, Russia
[3]Institute of Cryptography, Communication and Computer Sciences
70, Michurinskii ave., 119602 Moscow, Russia
[1]reshet@ispms.tsc.ru, [2]ksv1986@sibmail.com, [3]Vstar@mail.ru

Keywords: symbolic computations, Verma module, non-linear commutator algebra, C#, C++ implementation

We consider the problem of computer verification of the correctness of the oscillator realization over Heisenberg superalgebra $A_{1,2}$ of a special nonlinear commutator superalgebra $\mathcal{A}(Y(1), AdS_d)$ with 3 odd (t_0, t_1, t_1^+) and 6 even $(l_0, g_0, l_1, l_1^+, l_2, l_2^+)$, generators within symbolic computational approach by means of new programm *NcNlSuperalgebra* on C# [1] (having the Russian Certificate of State Registration No.2010611602). The above superalgebra naturally arises within the procedure of construction of the Lagrangian formulation for the higher-spin spin-tensors living on the anti-de-Sitter (AdS) d-dimensional space-time, characterizing by non-vanishing inverse square AdS-radius r. The oscillator realization was based, firstly, on the generalized Verma module (on general concepts of Verma module see [2]) explicit construction for the superalgebra $\mathcal{A}(Y(1), AdS_d)$ with involution. The feature of such a procedure is that of the elements of Verma module $|n_1^0, n_1, n_2\rangle_V$, for $n_1^0 = 0, 1$; $n_1, n_2 \in \mathbb{N}_0$, are constructed with help of triangular-like decomposition of $\mathcal{A}(Y(1), AdS_d)$ and highest weight vector, $|0\rangle_V \equiv |0, 0, 0\rangle_V$, in opposite to Lie algebra case contain more number of elements in acting of Cartan-like and positive root vectors t_0, l_0, t_1, l_1 on $|n_1^0, n_1, n_2\rangle_V$ in corresponding linear combination when the values

of the components n_1, n_2 become large. Second, there exists one-to-one correspondence between the constructed generalized Verma module and special Fock space generated by the same number of Heisenberg superalgebra $A_{1,2}$ generating elements, f, f^+, b_i, b_i^+, $i = 1, 2$, as the number of Hermitian elements in $\mathcal{A}(Y(1), AdS_d)$. However, the realization of the elements t_0, l_0, t_1, l_1 in terms of f, f^+, b_i, b_i^+ are non-polynomial in comparison with the Lie algebra case (for $r = 0$).

In order to check the validity of the oscillator realization of $\mathcal{A}(Y(1), AdS_d)$ over $A_{1,2}$, i.e. that the found expressions for $(t_0, t_1, t_1^+, l_0, g_0, l_i, l_i^+)$ really satisfy to the given algebraic relations of the non-linear superalgebra, we have elaborated the program *NcNlSuperalgebra* permitting to solve this problem within the restricted induction principle in power of the parameter r. We have checked the correctness of the oscillator realization up to the sixth power in r. *NcNlSuperalgebra* has some advantages and deficiencies in comparison with *Plural* known as a non-commutative extension of the package *Singular* [3]. The computer program is planning both to translate on $C+$ to enhance its processing speed and to enlarge its possibilities for application to more complicated non-linear algebra considered in [4] and [5].

References:

[1] A. Kuleshov, A.A. Reshetnyak, *Programming Realization of Symbolic Computations for Non-linear Commutator Superalgebras over the Heisenberg–Weyl Superalgebra: Data Structures and Processing Methods*, arXiv:0905.2705 [hep-th] (2009).

[2] J. Dixmier, *Algebres Enveloppantes*, Gauthier-Villars, Paris, 1974.

[3] V. Levandovskyy, Plural, a non-commutative extension of singular: past, present and future, In *Proceedngs of the Int. Symposium on Mathematical Theory of Networks and System (MTNS'06)* (A. Iglesias, N. Takayama, eds), 2006.

[4] I.L. Buchbinder, A. Reshetnyak, General Lagrangian formulation for higher spin fields with arbitrary index symmetry. I. Bosonic fields, *Nuclear Physics B*, 862 (2012), pp. 270–326.

[5] C. Burdik, A. Reshetnyak, On representations of Higher Spin symmetry algebras for mixed-symmetry HS fields on AdS-spaces. Lagrangian formulation, *J. Phys.: Conf. Ser.*, 343 (2012), p. 012102.

Interval arithmetic over finitely many endpoints

Siegfried M. Rump

Institute for Reliable Computing,
Hamburg University of Technology,
Schwarzenbergstraße 95, 21071 Hamburg, Germany
and Visiting Professor at Waseda University,
Faculty of Science and Engineering,
3–4–1 Okubo, Shinjuku-ku, Tokyo 169–8555, Japan
rump@tu-harburg.de

Keywords: interval arithmetic, enclosure methods, verified bounds

To my knowledge all definitions of interval arithmetic start with real endpoints and prove properties. Then, for practical use, the definition is specialized to finitely many endpoints, where many of the mathematical properties are no longer valid. There seems no treatment how to choose this finite set of endpoints to preserve as many mathematical properties as possible.

Here we define interval endpoints directly using a finite set which, for example, may be based on the IEEE 754 floating-point standard. The corresponding interval operations emerge naturally from the corresponding power set operations. We present necessary and sufficient conditions on this finite set to ensure desirable mathematical properties, many of which are not satisfied by other definitions. For example, an interval product contains zero if and only if one of the factors does.

The key feature of the theoretical foundation is that "endpoints" of intervals are not points but non-overlapping closed, half-open or open intervals, each of which can be regarded as an atomic object. By using non-closed intervals among its "endpoints", intervals containing "arbitrarily large" and "arbitrarily close to but not equal to" a real number can be handled. The latter may be zero defining "tiny" numbers, but also any other quantity including transcendental numbers.

Our scheme can be implemented straightforwardly using the IEEE 754 floating-point standard.

The bijective coding in the constructive world of $\mathbb{R}^n_c$

Gennady G. Ryabov, Vladimir A. Serov

Research Computer Center of Moscow State University (MSU SRCC)
1, building 4, Leninskiye Gory, 119991 Moscow, Russia

Keywords: n-cube, quaternary coding, Hausdorff-Hemming metrics, simplicial partition, combinatorial filling, supercomputing

The development of bijective coding methods for the constructive world [1] of cubic structures in a standard lattice $\mathbb{R}^n_c$ (with given orthogonal-normal frame $B = \{0, \mathbf{e_1}, \ldots, \mathbf{e_n}\}$ in $\mathbb{R}^n$), consisted of n-cubes, adjoining to each other by $(n-1)$-hyperfaces [2] is considered. Such coding provides a one-to-one correspondence between the n-digital ternary word D ($d_i \in A = \{0, 1, 2\}$) and each k-face ($k = 0, \ldots, n$) in an n-cube. Since it is possible to represent each individual k-face as a Cartesian product (Π) of k unit intervals $\mathbf{I}(\mathbf{e_i})$ such, that $\mathbf{e_i} \in B_1 \subset B$, and translation (T) across the rest $(n-k)$ $\mathbf{e_j} \in B_2 \subset B (B_2 = B \setminus B_1)$, one may express a bijectivity property for the k-face f_{nk}: $f_{nk}(B_1, B_2) = \prod\limits_{k} \mathbf{I}(\mathbf{e_i}) + \mathop{\mathrm{T}}\limits_{n-k} (\mathbf{e_j}) \overset{[1:1]}{\longleftrightarrow} < d_1, \ldots, d_n >$, $d_i = 2$ for $\mathbf{e_i} \in B_1$, $d_j = 0, 1$ for $\mathbf{e_j} \in B_2$. The sets of all n-digital ternary words $A^*_n = \{< d_1, \ldots, d_n >\}$ are called *cubants* [3]. Let us supplement the alphabet A by the letter $\emptyset$ (empty set) and define a digit-wise operation "multiplication" for all words on $A'^*_n, A' = \{\emptyset, 0, 1, 2\}$: $0 \times 0 = 0$; $0 \times 1 = 1 \times 0 = \emptyset$; $0 \times 2 = 2 \times 0 = 0$; $1 \times 1 = 1$; $1 \times 2 = 2 \times 1 = 1$; $2 \times 2 = 2$; $\emptyset \times (0, 1, 2) = (0, 1, 2) \times \emptyset = \emptyset$. Many operations on cubants and their properties were defined in [3], including:

1. The number of letters $\emptyset$ in the product of cubants D_1 and D_2 is equal a minimal path length across edges between bijective faces:

$$\#(\emptyset)(D_1 \times D_2) = L_{min}(D_1; D_2). \tag{1}$$

2. Let D^*_1/D_2 be a cubant for the furthest part in face D_1 from face D_2. Then the algorithm for computing D^*_1/D_2 consists in analyzing all such pairs of digits that $d_{1i} \in D_1$, $d_{2i} \in D_2 \ldots$, and changing the letters in D_1 in accordance with the rules: for the case $(d_{1i} = 2; d_{2i} = 0)$ change $d^*_{1i} = 1$, and for the case $(d_{1i} = 2; d_{2i} = 1)$ change $d^*_{1i} = 0$; for the remaining cases there are no changes in D_1. Thus, on the basis of (1):

$$\max_{D_1 \to D_2} \{L_{min}(D^*_1/D_2; D_2)\} = \#(\emptyset)((D^*_1/D_2) \times D_2), \tag{2}$$

$$\max_{D_2 \to D_1} \{L_{min}(D_2^*/D_1; D_1)\} = \#(\emptyset)((D_2^*/D_1) \times D_1). \qquad (3)$$

With (2) and (3), we have a distance $\rho_{\mathrm{HH}}(D_1, D_2) = \max\{\#(\emptyset)((D_1^*/D_2) \times D_2); \#(\emptyset)((D_2^*/D_1) \times D_1)\}$. All the k-faces of an n-cube form a finite Hausdorff-Hemming metric space.

The simplicial partition of an n-cube is such that each simplex is based on successive circuit for $n+1$ vertices, beginning at $(00\ldots0)$ and completing at $(11\ldots1)$ under a Hemming distance 1 requirement for each successive pair of vertices. Each step in the circuit is parallel to $\mathbf{e_i}$. The general number of all such different circuits in an n-cube is equal $n!$. The vertex set V and the edge set E are calculated for circuit order $(\mathbf{e_{i1}}, \ldots \mathbf{e_{in}})$ as follows: $V = \{v_0 = (00\ldots0); v_i = v_{i-1} + \mathbf{e_{is}}; s = 1, \ldots, n\}$; $E = \{v_0v_1; v_0v_2; v_0v_3; v_1v_3; v_2v_3; \ldots v_{n-1}v_n\}$. V and E form a 1-skeleton of the simplex. The circuit order for a canonical partition of the individual k-face is given on set $B_1 = \{\mathbf{e_{is}} : d_{is} \in D; d_{is} = 2\} = \{\mathbf{e_{j1}}, \ldots \mathbf{e_{jk}}\}$ by substitution $P \in S_k$ (symmetric group): $P(\mathbf{e_{j1}}, \ldots \mathbf{e_{jk}}) = (\mathbf{e_{m1}}, \ldots \mathbf{e_{mk}})$. The following operations are realized analogously to the case of an n-cube. We denote the action of group S_k on D with respect to calculation of V and E as Θ, and the simplex with a 1-skeleton (V, E) as Δ. Then, $\Theta(D, P) = (V, E) \overset{[1:1]}{\longleftrightarrow} \Delta_0(D, P)$; $\Delta(D, P) = \Delta_0(D, P) + \mathrm{T}(\mathbf{e_{it}})$, $\mathbf{e_{it}} \in B_2, t = 1, \ldots, k$. The pair cubant-substitution $(< d_1, \ldots, d_n/m_1, \ldots, m_k >)$ can be entitled as *simpant*. The common alphabet consists of all the decimal figures and some tokens. Hence, each k-face consists of $k!$ simplices, bijectivial to $k!$ simpants, and their general number in an n-cube is $F_\Delta(\mathbf{I^n}) = \sum_{k=2}^{n} k! C_n^k 2^{n-k}$, $\lim_{n\to\infty} F_\Delta(\mathbf{I^n})/n! = e^2$. A notion of *combinatory filling* for cubic and simplex structures in $\mathbb{R}_c^n$ is proposed. Finally, we discuss possibility of using modern supercomputers for computing on sets with given combinatorial filling.

References:

[1] Y.I. Manin, Classical computing, quantum computing, and Shor's factoring algorithm, `http://arxiv.org/abs/quant-ph/9903008v1` (March 1999).

[2] N.P. Dolbilin, M.A. Shtan'ko, M.I. Shtogrin, Cubic manifolds in lattices, *Russian Acad. Sci. Izv. Math.*, 44 (1995), No. 2, pp. 301–313.

[3] G.G. Ryabov, V.A. Serov, On the metric-topological computing in the constructive world of cubic structures, *Numerical Methods and Programming*, 11 (2010), No. 2, pp. 326–335.

Estimation of model parameters

Ilshat R. Salakhov, Olga G. Kantor

Bashkir State University
32, Zaki Validi Street
450074 Ufa, Russia
salah-off@mail.ru

Keywords: differential equations, Runge-Kutta method, system dynamics models, estimation of model parameters

Mathematical modeling of economic processes and consecutive establishment of logical connections enable monitoring, control and management. It is the most effective tool for solving various problems: problems of optimization, decision-making, and many others.

One of complex problems with a nonlinear feedback studying methods is system dynamics, on the basis of which was construct a model (1). It was developed in the mid-twentieth century by professor of Massachusetts Institute of Technology, J. Forrester. The aim of our work is to solve the inverse problem of determining the control parameters of system dynamics.

This problem is resolved on the model, which describes changes in the population, taking into account the influence of many factors. Using complex numerical algorithms, the model was corrected to achieve the required accuracy of description [1, 2]:

$$\begin{aligned} \frac{dN}{dt} &= 8.139 \cdot 10^{-22} \cdot N^{0.05} \cdot S^2 - 64.1 \cdot N^{0.03} \cdot S^{0.3}, \\ \frac{dD}{dt} &= 560 \cdot D^{0.35} - 9900 \cdot I, \\ \frac{dI}{dt} &= 0.131 \cdot I^{-0.4} - 0.0072 \cdot S^{0.092}. \end{aligned} \tag{1}$$

Where the unknown parameters of the model are: N - the population of the Russian Federation, pers.; D - per capita income for the year, rub./person; I - the consumer price index; $S = \frac{N \cdot D}{I}$

In forecasting population change, put the next problem. What should be the system control parameters D and I to provide the necessary number in the coming year, while maintaining an adequate description of the source data. Next we formulated optimality principles:

$$|N(t) - N_{exp}(t)| \leq \delta_1; \quad |D(t) - D_{exp}(t)| \leq \delta_2; \quad |I(t) - I_{exp}(t)| \leq \delta_3,$$

All system parameters are place in a given corridor values relative to experimental data.

$$\overline{A_N} \leq 10\%; \overline{A_D} \leq 10\%; \overline{A_I} \leq 10\%;$$

For all three equations the average approximation error is less than 10%.

$$|N(t) - N_{exp}(t + \Delta)| \leq \varepsilon N(t),$$

Provided the necessary predictive value of the population N change is $\varepsilon = 0.001$ from the actual value in the last period of time.

To organize the computer simulation a software package of mathematical modeling methods and numerical algorithms was implemented including:

1. The direct problem solution of differential equations by numerical integration with the help of the Runge-Kutta method.

2. The initial approximations of model parameters chosen through the translation of the system of differential equations to integral equations.

3. Determination of ranges of coefficient variation in which the conditions are adequately described.

4. Search for the model parameters by analyzing the optimality criteria.

To optimize the planned experiment it is necessary to identify the ranges of coefficient variation in which the conditions are adequately described. We obtained that the coefficients of the first equation vary in the range [5; 9] and [58; 62.5], and the coefficients of the second equation vary in the ranges [325; 820] and [0; 19.200].

Analyzing the results of the experiment showed that to provide population growth from 0 to 0.1% it is necessary to increase per capita income from 1.4% to 27%, or increase the consumer price index from 5.4% to 7.3%.

References:

[1] S.I. Spivak, O.G. Kantor, I.R. Salakhov, Estimation of model parameters of system dynamics, *Journal Srednevolzhskaya Mathematical Society*, 13 (2011), No. 3, pp. 107–113.

[2] S.I. Spivak, O.G. Kantor, I.R. Salakhov, Modeling the Russian Federation population by the system dynamic method, in *Statistics. Modeling. Optimization*, Publishing Center of South Ural State University, Chelyabinsk, 2011, pp. 239–246.

Interval pseudo-inverse matrices: computation and applications

Pavel Saraev

Lipetsk State Technical University
30, Moskovskaya st., 398600 Lipetsk, Russia
psaraev@yandex.ru

Keywords: interval pseudo-inversion, interval matrices, optimization

For any square interval matrix $[A] \in \mathbb{IR}^{n\times n}$, an interval inverse matrix is the minimal interval matrix $[A]^{-1} \in \mathbb{IR}^{n\times n}$ such that $[A]^{-1} \supset \left\{A^{-1} : A \in [A]\right\}$ [4]. It can be computed using an algorithm based on an interval method for real inverse matrix computation [2]. Generalizing techniques elaborated for interval inverse matrices to singular square and rectangular matrices is of scientific and practical interest.

The pseudo-inverse matrix $A^+ \in \mathbb{R}^{n\times m}$ for $A \in \mathbb{R}^{m\times n}$, also known as Moore-Penrose generalized inverse, is the only matrix satisfying the following conditions [1]: $AA^+A = A$, $A^+AA^+ = A^+$, $(AA^+)^T = AA^+$, $(A^+A)^T = A^+A$. For any interval matrix $[A] \in \mathbb{IR}^{m\times n}$, we define the interval pseudo-inverse matrix $[A]^+ \in \mathbb{IR}^{n\times m}$ as the minimal interval matrix such that $[A]^+ \supset \{A^+ : A \in [A]\}$. So $[A]^+$ includes all real pseudo-inverse matrices A^+ for all $A \in [A]$. We need an enclosure for $[A]^+$ instead of exact interval pseudo-inverse matrix for most applications. This work presents the interval Greville algorithm for interval matrices pseudo-inverse enclosure. Let $[A] \in \mathbb{IR}^{m\times n}$, and $[a_k]$ be its k-th column, where $k = 1, \ldots, n$. Let $[A_k]$ be the submatrix of $[A]$ constructed from the first k columns of $[A]$: $[A_k] = \big[[a_1] \quad [a_2] \quad \ldots \quad [a_k]\big]$. If $k = 1$ then $[A_1] = [a_1]$. For $k = 2, \ldots, n$, it is clear that $[A_k] = \big[[A_{k-1}] \quad [a_k]\big]$.

Let $k = 1$. Assume $[d_1] = \|[a_1]\|^2 = \sum_{i=1}^{m} [a_{i1}]^2$.

$$[A_1]^+ = \begin{cases} 0, & \text{if } \overline{[d_1]} = 0, \\ [a_1]^T/[d_1], & \text{if } \underline{[d_1]} > 0, \\ 0 \cup [a_1]^T/[d_1], & \text{otherwise,} \end{cases}$$

where $0 \in \mathbb{IR}^m$ is the null interval vector, and $\cup$ is the interval hull of the union of interval vectors.

Let $k = 2, \ldots, n$.

$$[A_k]^+ = \begin{bmatrix} [A_{k-1}]^+([I] - [a_k][f_k]) \\ [f_k] \end{bmatrix},$$

where I is the identity matrix of the order m, and

$$[c_k] = (I - [A_{k-1}][A_{k-1}]^+)[a_k], \quad [d_k] = \|c_k\|^2,$$

$$[f_k] = \begin{cases} [c_k]^T/[d_k], & \text{if} \underline{[d_k]} > 0, \\ [a_k]^T([A_{k-1}]^+)^T[A_{k-1}]^+/1 + \|[A_{k-1}]^+[a_k]\|^2, & \text{if} \overline{[d_k]} = 0, \\ [c_k]^T/[d_k] \cup [a_k]^T([A_{k-1}]^+)^T[A_{k-1}]^+/1 + \|[A_{k-1}]^+[a_k]\|^2, & \text{otherwise.} \end{cases}$$

Hence, $[A_n]^+$ is the required enclosure for $[A]^+$. The result can have infinite bounds in some cases, and the probability of such situations increases for wide and large matrices. Accuracy criterion can use tracing the defect in Moore-Penrose conditions, which is defined as

$$\begin{aligned} t = &\|[A][A]^+[A] - [A]\| + \|[A]^+[A][A]^+ - [A]^+\| \\ &+ \|([A][A]^+)^T - [A][A]^+\| + \|([A]^+[A])^T - [A]^+[A]\|. \end{aligned}$$

An interval pseudo-inverse matrix can be applied in optimization problems for determination of decision set bounds, also it can be used for unstable real matrix pseudo-inversion detection. This can be done by computing an interval pseudo-inverse of an ε-inflation $[A_\varepsilon]$ of a given real matrix A. When $[A_\varepsilon]^+$ is wide or the defect is large, the pseudo-inversion is unstable.

Another interesting application is the guaranteed global parameter estimation in nonlinear least squares problems whose variables are separable. It is based on relation $u = \Psi(v)^+ y$ between linear and nonlinear vectors u and v respectively, with known response vector y, where $\Psi(v)$ is the matrix of basis functions built on the input-output data set [3]. For a subspace of nonlinear parameters $[v]$, an optimal subspace of linear parameters by $[u] = [\Psi([v])]^+ y$ can be estimated. The work is supported by RFBR, project N 11-07-97504-r_center_a.

References:

[1] A. Albert, *Regression and the Moore-Penrose Pseudoinverse*, Academic Press, New York and London, 1972.

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

[3] G.H. Golub, V. Pereyra, The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate, *SIAM J. Num. Anal.*, 10 (1973), pp. 413–432.

[4] J. Rohn, Inverse interval matrix, *SIAM J. Num. Anal.*, 3 (1993), No. 3, pp. 864–870.

Calculation of potential and attraction force of an ellipsoid

Alexander O. Savchenko

Institute of Computational Mathematics
and Mathematical Geophysics SB RAS
6, Lavrentiev ave.
630090 Novosibirsk, Russia
savch@ommfao1.sscc.ru

This work is devoted to numerical integration of the potential and attraction force of an ellipsoid. The problem is reduced to that of calculating the integral of a given density distribution with a singular kernel. An easy-to-implement semi-analytical method to calculate the integral is proposed.

The main idea of the method is to represent the sought-for function in the form of a triple integral in such a way that the inner integral of the kernel can be taken analytically. In doing this, the kernel is considered as a weight function. To approximate the inner integral, a quadrature formula for the product of functions, one of which has an integrable singularity, is proposed. This approach enables one to obtain an integrand with a weak logarithmic singularity. This singularity can be easily eliminated by a change of variables in the next outer integral. Thus, to calculate all the integrals, quadrature formulas without singularities are obtained. Additionally, the functions to be calculated do not have large values within the integration domain. To obtain higher accuracy of the numerical calculations, it is sufficient to simply increase the number of integration points along each of the coordinates. This approach is not always acceptable in many other integration methods because of the presence of a singularity in the integrands.

The method is illustrated by numerical experiments for which complicated test functions are constructed. These functions, which are the exact potential and exact attraction force of an ellipsoid of revolution with an elliptical distribution of density, have value of its own and can be used for other purposes.

References:

[1] A.O. Savchenko, Calculation of the volume potential for ellipsoidal bodies, *Sibirskii Zhurnal Industrial'noi Matematiki*, 15 (2012), No. 1, pp. 123–131.

A numerical verification method for solutions to systems of elliptic partial differential equations

Kouta Sekine[1], Akitoshi Takayasu[2] and Shin'ichi Oishi[2,3]

[1]Graduate School of Fundamental Science and Engineering, Waseda University
[2]Faculty of Science and Engineering, Waseda University
[3]CREST, JST

3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan

[1]s115100710@akane.waseda.jp

Keywords: Computer assisted proof, Finite element method, systems of elliptic partial differential equations

In this talk, a method of computer-assisted proof is proposed for systems of elliptic partial differential equations:

$$\begin{cases} -\varepsilon^2 \Delta u = f(u) - \delta v, & \text{in } \Omega, \\ -\Delta v = u - \gamma v, & \text{in } \Omega, \\ u = v = 0, & \text{on } \partial\Omega. \end{cases} \tag{1}$$

Here, Ω is a bounded polygonal domain in $\mathbb{R}^2$. $\varepsilon \neq 0, \gamma$ and δ are real parameters. A mapping $f : H_0^1(\Omega) \to L^2(\Omega)$ is assumed to be Fréchet differentiable. When u is a known function, the boundary value problem:

$$\begin{cases} -\Delta v = u - \gamma v, & \text{in } \Omega, \\ v = 0, & \text{on } \partial\Omega, \end{cases} \tag{2}$$

has a unique solution. Then, v is presented by $v = Bu$, where $B : L^2(\Omega) \to H_0^1(\Omega)$ is a solution operator of (2). Substituting this for (1), it follows

$$\begin{cases} -\Delta u = \dfrac{1}{\varepsilon^2} \left(f(u) - \delta B u \right), & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega. \end{cases} \tag{3}$$

Transforming (1) into (2) and (3) allows the verification of solutions. Y. Watanabe has studied this type of system (1) by Nakao's theory, which is based on

fixed-point theorems. Using the Newton-Kantorovich theorem with the operator norm $\|B\|_{L^2,H_0^1}$, a verification method for (2) and (3) is proposed. If γ is not an eigenvalue λ of the Laplace operator, there exists the solution operator B. The operator norm $\|B\|_{L^2,H_0^1}$ can be estimated as follows:

$$\|B\|_{L^2,H_0^1} \leq C_{e,2}K, \tag{4}$$

where $C_{e,2}$ is the Poincaré constant and

$$K := \max\left\{\left|\frac{\lambda}{\lambda+\gamma}\right| : \ \lambda \text{ is eigenvalue of the Laplace operator, } \gamma \in \mathbb{R}\setminus\{\lambda\}\right\}.$$

Hence, the upper bound of operator norm $\|B\|_{L^2,H_0^1}$ is obtained simply by the eigenvalue λ. Our verification method is based on the following two studies. A verified evaluation for eigenvalues of the Laplace operator has been shown by X. Liu and S. Oishi [2]. A. Takayasu, X. Liu and S. Oishi have proposed the verification method for solutions to nonlinear partial differential equations using the Newton-Kantorovich theorem [3]. In our procedure, approximate solutions $\hat{u}$ and $\hat{v}$ of the system (1) are calculated by the finite element method. The inequality (4) yields a rigorous upper bound of the norm $\|B\|_{L^2,H_0^1}$, which leads to the guaranteed error estimate $\|u-\hat{u}\|_{H_0^1}$ based on the Newton-Kantorovich theorem. Further, the upper bound of $\|v-\hat{v}\|_{H_0^1}$ is given by the operator norm $\|B\|_{L^2,H_0^1}$ and $\|u-\hat{u}\|_{H_0^1}$. Detailed proofs and numerical results will be presented.

References:

[1] Y. Watanabe, *A numerical verification method for two-coupled elliptic partial differential equation*, Japan Journal of Industrial and Applied Mathematics, 26 (2009), pp.233-247

[2] X. Liu and S. Oishi, *Verified eigenvalue evaluation for elliptic operator on arbitrary polygonal domain*, in preparation.

[3] A. Takayasu, X. Liu and S. Oishi, *Verified computations to semilinear elliptic boundary value problems on arbitrary polygonal domains*, submitted to publication.

Processing measurement uncertainty: from intervals and p-boxes to probabilistic nested intervals

Konstantin K. Semenov[1], Gennady N. Solopchenko[1], and Vladik Kreinovich[2]

[1]Measur. Info. Techn. Dept., Saint-Petersburg State Polytechnic Univ., Russia
[2]Computer Science Dept., University of Texas, El Paso, Texas 79968, USA
semenov.k.k@gmail.com, g.n.solopchenko@mail.ru, vladik@utep.edu

Keywords: measurement uncertainty, interval computations, p-boxes, probabilistic nested intervals

Due to measurement errors, the result $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ of processing measurement outcomes is, in general, different from the desired result $y = f(x_1, \ldots, x_n)$ of processing actual (unknown) values x_i. It is desirable to estimate the difference $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$ [4].

When we only know the bounds Δ_i on measurement errors $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$, the only information that we have about y is that $y \in \boldsymbol{y} \stackrel{\text{def}}{=} \{f(x_1, \ldots, x_n) : x_1 \in \boldsymbol{x}_1, \ldots, x_n \in \boldsymbol{x}_n\}$, where $\boldsymbol{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. Computing such a range $\boldsymbol{y}$ is one of the main problems solved by *interval computations* [2].

Often, in addition to the bounds Δ_i, we have partial information about the probability of different values Δx_i. A general probability distribution can be described by the cumulative distribution function (cdf) $F(x) \stackrel{\text{def}}{=} \text{Prob}(\eta \le x)$. Partial information means that instead of knowing the exact values $F(x)$, we only know *bounds* $\underline{F}(x) \le F(x) \le \overline{F}(x)$. The corresponding "interval-valued" cdf $[\underline{F}(x), \overline{F}(x)]$ is known as a *probability box*, or *p-box*, for short [1].

P-boxes are useful in decision making, where the objective is often to satisfy a given inequality-type constraint, and p-boxes provide the probability of satisfying this constraint. In many practical situations (e.g., in control applications), the objective is to find how far is the actual value y from our estimate $\widetilde{y}$. We know that the desired probability $p \stackrel{\text{def}}{=} \text{Prob}(-\Delta \le \eta \le \Delta)$ is equal to $F(\Delta) - F(-\Delta)$, so based on the known p-boxes, we can conclude that $p \le \widetilde{p} \stackrel{\text{def}}{=} \overline{F}(\Delta) - \underline{F}(-\Delta)$. However, often, this $\widetilde{p}$ is an overestimation: e.g., for $\Delta = 0$, we have $p = 0$, while for p-boxes of finite width w, we have $\widetilde{p} = 2w$.

To get better bounds for p, we use *probabilistic nested intervals*: 1-parametric families of confidence intervals $\boldsymbol{x}_i(\alpha)$ for which $\mathrm{Prob}(x_i \in \boldsymbol{x}_i(\alpha)) \geq 1-\alpha$ and $\boldsymbol{x}_i(\alpha) \subseteq \boldsymbol{x}_i(\alpha')$ when $\alpha' < \alpha$. E.g., when we have a systematic error component with known bounds $[-\Delta_{si}, \Delta_{si}]$ and a normally distributed random error component with a known σ_i, the confidence intervals are obtained by adding the usual Gaussian confidence interval to the interval $[\widetilde{x}_i - \Delta_{si}, \widetilde{x}_i + \Delta_{si}]$.

Probabilistic nested intervals are a particular case of nested intervals [3]. However, [3] focused on expert estimates, where it was reasonable to assume that when we know that $x_i \in \boldsymbol{x}_i(\alpha)$ with confidence $1-\alpha$, then $y = f(x_1, \ldots, x_n) \in f(\boldsymbol{x}_1(\alpha), \ldots, \boldsymbol{x}_n(\alpha))$ with the same confidence $1-\alpha$. This assumption led to explicit formulas for propagating expert-related nested intervals through computations.

In contrast, it is usually assumed that random errors of different measurements are independent [4]; in this case, when for each i, we have $x_i \in \boldsymbol{x}_i(\alpha)$ with probability $\geq 1-\alpha$, then we can only conclude that $(x_1, \ldots, x_n) \in \boldsymbol{x}_1(\alpha) \times \ldots \times \boldsymbol{x}_n(\alpha)$ (and thus, that $y = f(x_1, \ldots, x_n) \in f(\boldsymbol{x}_1(\alpha), \ldots, \boldsymbol{x}_n(\alpha))$) with probability $\leq (1-\alpha)^n \ll 1-\alpha$. So, we need *new formulas* for propagating probabilistic nested intervals. Such formulas will be described in the talk.

When measurement errors Δx_i are small, we can safely ignore terms quadratic (and of higher order) in Δx_i. For this linearized case, we can use *automatic differentiation* to design efficient algorithms. We can further speed up computations because in practice, inputs are usually known with 5-10% accuracy. In such situations, the result can only be computed with a similar 1-digit accuracy, so there is no need to perform iterations that improve the 2nd digit. A practical example of such a speed-up will be presented.

References:

[1] S. Ferson, *RAMAS Risk Calc 4.0: Risk Assessment with Uncertain Numbers*, CRC Press, Boca Raton, Florida, 2002.

[2] R.E. Moore, R.B. Kearfott, M.J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[3] H.T. Nguyen, V. Kreinovich, Nested intervals and sets: concepts, relations to fuzzy sets, and applications, in *Applications of Interval Computations* (R.B. Kearfott et al., eds.), Kluwer, Dordrecht, 1996, pp. 245–290.

[4] S. Rabinovich, *Measurement Errors and Uncertainties: Theory and Practice*, Springer, New York, 2005.

Deterministic global optimization using the Lipschitz condition

Yaroslav D. Sergeyev

University of Calabria, Rende, Italy
and N.I. Lobatchevsky University of Nizhni Novgorod, Russia
Via P. Bucci, Cubo 42-C, 87036 Rende (CS), Italy
yaro@si.deis.unical.it

Keywords: global optimization, Lipschitz condition, partitioning strategies

In this lecture, the global optimization problem of a multidimensional function satisfying the Lipschitz condition with an unknown Lipschitz constant over a multi-dimensional box is considered. It is supposed that the objective function can be "black box", multiextremal, and non-differentiable. It is also assumed that evaluation of the objective function at a point is a time-consuming operation. Many algorithms for solving this problem have been discussed in the literature (see [1–12] and references given therein). They can be distinguished, for example, by the way of obtaining an information about the Lipschitz constant and by the strategy used to explore the search domain.

Different exploration techniques based on various adaptive partition strategies are analyzed. The main attention is dedicated to diagonal algorithms, since they have a number of attractive theoretical properties and have proved to be efficient in solving applied problems. In these algorithms, the search box is adaptively partitioned into sub-boxes and the objective function is evaluated only at two vertices corresponding to the main diagonal of the generated sub-boxes.

It is demonstrated that the traditional diagonal partition strategies do not fulfill the requirements of computational efficiency because of executing many redundant evaluations of the objective function. A new adaptive diagonal partition strategy that allows one to avoid such computational redundancy is described. Some powerful multidimensional global optimization algorithms based on the new strategy are introduced. Results of extensive numerical experiments performed to test the methods proposed demonstrate their advantages with respect to diagonal algorithms in terms of both number of trials of the objective function and qualitative analysis of the search domain, which is characterized by the number of generated boxes. Finally, problems with Lipschitz first derivatives are considered and connections between the Lipschitz global optimization and interval analysis global optimization are discussed.

References:

[1] L.G. Casado, I. Garcia, Ya.D. Sergeyev, Interval algorithms for finding the minimal root in a set of multiextremal non-differentiable one-dimensional functions, *SIAM J. Scientific Computing*, 24 (2002), No. 2, pp. 359–376.

[2] Yu.G. Evtushenko, M.A. Posypkin, An application of the nonuniform covering method to global optimization of mixed integer nonlinear problems, *Comput. Math. Math. Phys.*, 51 (2011), No. 8, pp. 1286–1298.

[3] R. Horst and P.M. Pardalos, eds., *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht, 1995.

[4] D.E. Kvasov, Ya.D. Sergeyev, Univariate geometric Lipschitz global optimization algorithms, *NACO*, 2 (2012), No. 1, 69–90.

[5] D. Lera, Ya.D. Sergeyev, An information global minimization algorithm using the local improvement technique, *J. Global Optimization*, 48 (2010), No. 1, 99–112.

[6] J. Pintér, *Global Optimization in Action*, Kluwer, Dordrecht, 1996.

[7] Ya.D. Sergeyev, D.E. Kvasov, *Diagonal Global Optimization Methods*, FizMatLit, Moscow, 2008.

[8] Ya.D. Sergeyev, D.E. Kvasov, Lipschitz global optimization, *Wiley Encyclopaedia of Operations Research and Management Science*, 4 (2011), pp. 2812–2828.

[9] Ya.D. Sergeyev, D.E. Kvasov, Global search based on efficient diagonal partitions and a set of Lipschitz constants, *SIAM J. Optimization*, 16 (2006), No. 3, pp. 910–937.

[10] Ya.D. Sergeyev, P. Pugliese, D. Famularo, Index information algorithm with local tuning for solving multidimensional global optimization problems with multiextremal constraints, *Math. Programming*, 96 (2003), No. 3, pp. 489–512.

[11] R.G. Strongin, Ya.D. Sergeyev, *Global optimization with non-convex constraints: sequential and parallel algorithms*, Kluwer, Dordrecht, 2000.

[12] A.A. Zhigljavsky, A. Žilinskas, *Stochastic Global Optimization, Springer*, New York, 2008.

The Infinity Computer and numerical computations with infinite and infinitesimal numbers

Yaroslav D. Sergeyev

University of Calabria, Rende, Italy
and N.I. Lobatchevsky University of Nizhni Novgorod, Russia
Via P. Bucci, Cubo 42-C
87036 Rende (CS), Italy
yaro@si.deis.unical.it

Keywords: Infinities and infinitesimals, Infinity Computer, numerical computations

A new methodology (see [6,9]) allowing one to execute numerical computations with finite, infinite, and infinitesimal numbers on a new type of a computational device called *the Infinity Computer* (EU, USA, and Russian patents have been granted) is introduced. A calculator using the Infinity Computer technology is presented during the talk. The new approach (its relations with traditional approaches are discussed in [4-6,9]) applies the principle 'The part is less than the whole' to all numbers (finite, infinite, and infinitesimal) and to all sets and processes (finite and infinite). It is shown that it becomes possible to write down finite, infinite, and infinitesimal numbers by a finite number of symbols as particular cases of a unique framework (different from that of the non-standard Analysis). The new methodology (among other things) introduces infinite integers having both cardinal and ordinal properties.

The point of view on infinite and infinitesimal quantities presented in this talk uses strongly two methodological ideas borrowed from the modern Physics: relativity and interrelations holding between the object of an observation and the tool used for this observation. Thus, connections between different numeral systems used to describe mathematical objects and the objects themselves are emphasized. The new computational methodology gives the possibility both to execute numerical (not symbolic) computations of a new type and simplifies fields of Mathematics where the usage of the infinity and/or infinitesimals is necessary. Numerous examples and applications are given: differential equations, divergent series, fractals, linear and non-linear optimization, numerical differentiation, percolation, probability theory, Turing machines, etc. (see [1-10]).

A lot of additional information on the new methodology (papers, reviews, awards, etc.) can be downloaded from *http://www.theinfinitycomputer.com*

References:

[1] L. D'Alotto, Cellular automata using infinite computations, *Applied Mathematics and Computation*, 218 (2012), No. 16, pp. 8077–8082.

[2] S. De Cosmis, R. De Leone, The use of Grossone in Mathematical Programming and Operations Research, *Applied Mathematics and Computation*, 218 (2012), No. 16, pp. 8029–8038.

[3] D.I. Iudin, Ya.D. Sergeyev, M. Hayakawa, Interpretation of percolation in terms of infinity computations, *Applied Mathematics and Computation*, 218 (2012), No. 16, pp. 8099–8111.

[4] G. Lolli, Infinitesimals and infinites in the history of Mathematics: A brief survey, *Applied Mathematics and Computation*, 218 (2012), No. 16, pp. 7979–7988.

[5] M. Margenstern, Using Grossone to count the number of elements of infinite sets and the connection with bijections, *p-Adic Numbers, Ultrametric Analysis and Applications*, 3 (2011), No. 3, pp. 196–204.

[6] Ya.D. Sergeyev, A new applied approach for executing computations with infinite and infinitesimal quantities, *Informatica*, 19 (2008), No. 4, pp. 567–596.

[7] Ya.D. Sergeyev, Numerical point of view on Calculus for functions assuming finite, infinite, and infinitesimal values over finite, infinite, and infinitesimal domains, *Nonlinear Analysis Series A: Theory, Methods & Applications*, 71 (2009), No. 12, pp. e1688–e1707.

[8] Ya.D. Sergeyev, A. Garro, Observability of Turing Machines: a refinement of the theory of computation, *Informatica*, 21 (2010), No. 3, pp. 425–454.

[9] Ya.D. Sergeyev, *Lagrange Lecture*: Methodology of numerical computations with infinities and infinitesimals, *Rendiconti del Seminario Matematico dell'Universit e del Politecnico di Torino*, 68 (2010), No. 2, pp. 95–113.

[10] Ya.D. Sergeyev, Higher order numerical differentiation on the Infinity Computer, *Optimization Letters*, 5 (2011), pp. 575–585.

Towards a more realistic treatment of uncertainty in Earth and environmental sciences: beyond a simplified subdivision into interval and random components

Christian Servin[1,4], Craig Tweedie[2,4], and Aaron Velasco[3,4]

[1]Computational Sciences Program
[2]Environmental Science and Engineering Program
[3]Department of Geological Sciences
[4]Cyber-ShARE Center
University of Texas, El Paso, Texas 79968, USA
christains@utep.edu, ctweedie@utep.edu, velasco@geo.utep.edu

Keywords: interval computations, periodic error, time series, resolution

When processing data, it is often very important to take into account measurement uncertainty, i.e., the fact that the measurement results $\widetilde{x}$ are, in general, different from the actual (unknown) value x of the corresponding quantity. In measurement theory, traditionally, a measurement error $\Delta x \stackrel{\text{def}}{=} \widetilde{x} - x$ is subdivided into random and systematic components $\Delta x = \Delta_s x + \Delta_r x$ (see, e.g., [2]): the systematic error component $\Delta_s x$ is usually defined as the expected value $\Delta_s x = E[\Delta x]$, while the random error component is usually defined as the difference $\Delta_r x \stackrel{\text{def}}{=} \Delta x - \Delta_s x$. By definition, the systematic error component does not change from measurement to measurement, while the random errors $\Delta_r x$ corresponding to different measurements are usually assumed to be independent.

For the systematic error component, we only know the upper bound Δ_s for which $|\Delta_s x| \le \Delta_s$. Thus, the only information that we have about the value of this component is that it belongs to the interval $[-\Delta_s, \Delta_s]$. Because of this fact, interval computations are used for processing the systematic errors. The random error component is usually characterized by the corresponding probability distribution; often, it is assumed to be Gaussian, with a known standard deviation σ.

For many Earth and environmental science measurements, the differences $\Delta_r x = \Delta x - \Delta_s x$ corresponding to nearby moments of time are often strongly correlated. For example, meteorological sensors may have daytime or nighttime biases, or winter and summer biases. To capture this correlation, environmental science researchers proposed an empirically successful semi-heuristic three-component model of measurement error. In this model, the difference $\Delta x - \Delta_s x$ is represented as a combination of a "truly random" error $\Delta_t x$ (which is independent from one measurement to another), and a new "periodic" component $\Delta_p x$.

We provide a theoretical explanation for this heuristic three-component model, and we show how to extend the traditional interval and probabilistic error propagation techniques to this three-component model. Our preliminary results are described in [3].

In practice, instead of a *single* quantity x (temperature, density, etc.), we often have a *field* $x(s)$ in which the value of the quantity changes with a spatial location s (and, sometimes, with time t). For fields, the measurement error $\widetilde{x}(s) - x(s)$ is caused not only by the inaccuracy of the measuring instrument (MI), but also by the fact that the output $\widetilde{x}(s)$ of the MI is determined by the *average* $\int K(s - s') \cdot x(s')\, ds'$ over a neighborhood $s' \approx s$ (here, $K(s)$ describes the instrument's *spatial resolution*). In the talk, we describe how to take into account this additional uncertainty, and how to decrease it by merging ("fusing") two results $\widetilde{x}_1(s)$ and $\widetilde{x}_2(s)$ obtained from measuring the same field $x(s)$; our preliminary results appeared in [1]. As a case study, we consider the combination of density descriptions obtained from seismic measurements and from gravity measurements.

References:

[1] O. Ochoa, A. A. Velasco, C. Servin, and V. Kreinovich, Model Fusion under Probabilistic and Interval Uncertainty, with Application to Earth Sciences, *International Journal of Reliability and Safety*, 6 (2012), No. 1–3, pp. 167–187.

[2] S. Rabinovich, *Measurement Errors and Uncertainties: Theory and Practice*, American Institute of Physics, New York, 2005.

[3] C. Servin, M. Ceberio, A. Jaimes, C. Tweedie, and V. Kreinovich, *How to Describe and Propagate Uncertainty When Processing Time Series*, Technical Report UTEP-CS-12-01a, Univ. of Texas at El Paso, Dept. Computer Science, 2012, http://www.cs.utep.edu/vladik/2012/tr12-01a.pdf

Boundary intervals and visualization of AE-solution sets for interval system of linear equations

Irene A. Sharaya

Institute of Computational Technologies SD RAS
6, Lavrentiev ave., 630090 Novosibirsk, Russia
sharaya@ict.nsc.ru

Keywords: interval linear system, AE-solution set, boundary interval

Theory of AE-solution sets (AEss) for interval systems of linear equations was developed by Shary (see e.g. [1]). The united solution set (USS), the tolerable solution set and the controllable solution set are particular cases of the AE-solution sets.

It is known [1] that the intersection of an AE-solution set with a closed orthant is a convex polyhedron. A system of linear inequalities determining this polyhedron may be obtained from the initial interval system of equations.

Programs that allow 'to see' the AE-solution set are useful in analysis of it properties and in debugging the methods for estimation of this set. By now, there are several such programs:

author(s)	language	address	maximum size of system and solution type	process unbounded sets	process thin sets
Rump Z.	Matlab	[2]	3×3 USS	$-$	$+$
Krämer W., Paw G.	Java	[3]	3×3 USS	$\mp$	$\mp$
Krämer W., Braun S.	Maple	[3]	3×3 USS	$\mp$	$\mp$
Popova E.D.	Mathematica	[4]	3×3 USS	$\mp$	$-$
Popova E.D.	Mathematica	[5]	2×2 AEss	$\mp$	$-$
Sharaya I.A.	PostScript	[6]	2×2 AEss	$+$	$+$

These programs handle the systems with no more than 3 rows and have difficulties in processing unbounded and thin sets.

What will be presented in the talk are

- a new MATLAB package for visualization of AE-solution sets
- and boundary intervals method as a base of this package.

Boundary intervals method is a new visualization method for solution set of linear inequalities system. It may be used (and modified) for

— solution set to system of two-sided linear inequalities and

— AE-solution set to interval system of linear equations.

The key object of the method is a boundary interval.

Definition. Let us be given the system of linear inequalities $Ax \geq b$ with $A \in \mathbb{R}^{m\times 2}$, $b \in \mathbb{R}^m$. If the set $\{\, x \mid (A_{i:}x = b_i)\,\&\,(Ax \geq b)\,\}$ for $i \in \{1, \ldots, m\}$ is not empty, we call it *boundary interval*.

A boundary interval as a set of points on the plane may be a single point, a segment, a ray or a straight line. All edges of the set $\{x \mid Ax \geq b\}$ are boundary intervals. Some vertices of this set may be boundary intervals too.

Boundary intervals method allows 'to see' 2D and 3D AE-solution sets for interval linear systems with rectangular matrices and can process unbounded and thin sets.

The work is supported by the State Program for Support of Leading Scientific Schools of Russian Federation (grant No. NSh-6293.2012.9).

References:

[1] S.P. Shary, A new technique in systems analysis under interval uncertainty and ambiguity, *Reliable Computing*, 8 (2002), No. 5, pp. 321–419.

[2] `http://www.ti3.tu-harburg.de/rump/intlab/` (The Intlab is the Matlab toolbox for reliable computing. Intlab function `plotlinsol` plots united solution set of interval linear system in 2 or 3 unknowns.)

[3] W. Krämer, *Computing and Visualizing Solution Sets of Interval Linear Systems*, Preprint BUW-WRSWT 2006/8, `http://www2.math.uni-wuppertal.de/~xsc/preprints/prep_06_8.pdf`

[4] `http://cose.math.bas.bg/webMathematica/webComputing /IntervalSSet3D.jsp`

[5] `http://cose.math.bas.bg/webMathematica/webComputing /IntervalSSet-AE.jsp`

[6] `http://www.nsc.ru/interval/Programing/AE-solset.ps` (The input data – matrix, right-hand side vector of the system and, optionally, initial enclosing box, – can be entered into this file by a text editor. Then the program can be executed by any PostScript interpreter.)

Randomized interval methods for global optimization

Sergey P. Shary[1], Nikita V. Panov[2]

[1]Institute of computational technologies SD RAS
[1,2]Institute of design and technology for computing machinery SD RAS
Novosibirsk, Russia
[1]shary@ict.nsc.ru, [2]crtgl@mail.ru

Keywords: global optimization, randomized interval algorithms, interval simulated annealing, interval genetic algorithm

Our work is devoted to the problem of global optimization of a real-valued function $f : \mathbb{R}^n \supseteq \boldsymbol{X} \to \mathbb{R}$ over an axis-aligned interval box $\boldsymbol{X}$:

$$\text{find} \quad \min_{x \in \boldsymbol{X}} f(x). \tag{1}$$

During the last decades, various interval techniques [1,2,3] have been developed for the solution of the problem (1). They enable one to reliably compute two-sided bounds for both the global optimum of f and the argument at which it is attained. The common basis of these methods is adaptive, according to the "branch-and-bound" strategy, subdivision of the objective function domain $\boldsymbol{X}$ combined with interval evaluation of the ranges of f over the resulting subboxes of $\boldsymbol{X}$. When executing, such methods iteratively refine interval estimates of the objective function through splitting, step by step, the boxes on which the estimate is the best at the current step.

Extensive employing such interval global optimization algorithms has revealed a number of problems. If the dimension of the problem is large, and/or the objective function f has a lot of local extremums, the deterministic interval global optimization algorithms can have low performance and produce an answer with considerable overestimation.

Usual ways to improve efficiency of the interval global optimization methods include increasing accuracy of interval evaluation, incorporating, into the algorithm, procedures that exclude the subboxes without the optimum, etc. For complex objective functions, one of the main sources of inefficiency is a large amount of unnecessary splittings, and it makes sense to pay more attention to the selection of the box to be subdivied at each step of the algorithm.

In our work, we develop interval global optimization algorithms of a new type that are based on the traditional adaptive subdivision-estimation of the search area, but involve randomization, i. e. introduce a stochastic control into the usual deterministic scheme [4]. This combination provides improved computational efficiency in comparison with ordinary purely deterministic algorithms. Besides, implementation of the above general idea may result in either strictly verified algorithms or those providing only probabilistic guarantees of the answer.

The simplest randomized interval optimization algorithms are "random interval splitting" [4] and "random interval priority splitting" [5]. The latter is an improvement of the former one supplied with so-called prioritization of the subboxes according to their width and/or current estimate.

More involved algorithms we have constructed on this way are interval simulated annealing [4] and a few interval evolution algorithms that develop the general idea of genetic algorithms [5,6].

In the randomized interval methods, the use of stochastic control passes facilitates solving complex problems more efficiently than with the traditional deterministic interval methods. In particular, we feature "verified versions" of such algorithms that provide, in spite of their stochastic character, numerical verification of the answer and produce, on output, two-sided interval bounds for the global optimum.

References:

[1] H. Ratschek, J. Rokne, *New Computer Methods for Global Optimization*, Ellis Horwood, Halsted Press, Chichester, New York, 1988.

[2] E. Hansen, G.W. Walster, *Global Optimization Using Interval Analysis*, Marcel Dekker, New York, 2004.

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

[4] S.P. Shary, Randomized algorithms in interval global optimization, *Numerical Analysis and Applications*, 1 (2008), No. 4, pp. 376–389.

[5] N.V. Panov, A unification of stochastic and interval approaches for the solution of the problem of global optimization of functions, *Computational Technologies*, 14 (2009), No. 5, pp. 49–65 (in Russian).

[6] N.V. Panov, S.P. Shary, Interval evolutionary algorithm for global optimization, *Transactions of Altai State University*, No. 1 (69) (2011), vol. 2, pp. 108–113 (in Russian).

Verified templates for design of combinatorial algorithms

Nikolay V. Shilov

A.P. Ershov Institute of Informatics Systems, 6, Lavrentiev ave.
630090 Novosibirsk, Russia
shilov@iis.nsk.su

Keywords: dynamic programming, branch and bound, backtracking, formal specification, formal verification

There exists a split between *reliable computing* and *program verification* communities: sometimes it seems that computing people assume that program code that "implements" a reliable method can be justified by extensive testing, while verification people think that reliability of any specified computational program can be formally verified in automatic mode from scratch. We try to find a compromise both extreme viewpoints by suggesting, formalizing and verifying (manually but formally) *templates for design of algorithms for combinatorial optimization.*

In particular, we formalize three algorithmic design patterns that are core patterns in the combinatorial optimization, namely: Dynamic Programming (DyP), Backtracking (BTR) and Branch-and-Bound (B&B). They can be formalized as design templates, specified by correctness conditions, and formally verified in Floyd – Hoare methodology [1]. BTR and B&B templates have been considered in [2] in full details, DyP is sketched below. A methodological novelty consists in treatment (interpretation) of DyP as the set-theoretic *least fix-point* (lfp) *in a virtual domain* (according to Knaster–Tarski theorem).

Dynamic Programming [3] is a recursive method for solving optimization problems presented by appropriate Bellman equation. We can assume without loss of generality that the Bellman equation has the following *canonical form*

$$G(x) \;=\; if\ p(x)\ then\ f(x)\ else\ g(x,\ (G(t_i(x)),\ i \in [1..n])),$$

where $G : X \to Y$ is the objective function, $p \subseteq X$ is a known predicate, $f : X \to Y$ is a known function, $g : X^* \to X$ is a known function with a variable (but finite) number of arguments n, and all $t_i : X \to X$, $i \in [1..n]$ are known functions also.

Dynamic Programming template (specified in Hoare style [1]) follows.

\\Precondition:
{D is a non-empty set of argument values,
S and P are ‘‘trivial’’ and ‘‘target’’ subsets in D,
F:$2^D \to 2^D$ is a call-by-value total monotone function,
ρ:$2^D \times 2^D \to$Bool is a call-by-value total function
monotone on the second argument}
\\Template:
var Z:= S, Z1 : subsets of D;
repeat Z1:= Z ; Z:= F(Z) until (ρ(P,Z) or Z=Z1)
\\Postcondition:
{ρ(P,Z) $\Leftrightarrow$ ρ(P, $lfp\ \lambda Q.(S \cup$F$(Q))$)}

Proposition. (1) *Dynamic Programming template is partially correct, i.e. for any input data that meets the precondition, the algorithm instantiated from the template either loops or halts in such a way that the postcondition holds upon the termination. Assuming that for some input data the precondition of the Dynamic Programming template is valid, and the domain D is finite, then the algorithm instantiated from the template terminates after at most $|D|$ iterations of the loop* repeat-until.
(2) *Let us consider the above Bellman equation and let $SPP : X \to 2^D$ be the following support function: $SPP(x) = if\ p(x)\ then\ \{x\}\ else\ \{x\} \cup (\bigcup_{i \in [1..n]} SPP(t_i(x)))$. Let $v \in X$ be any value. If to adopt (in the DyP template) the graph of G on $SPP(v)$ as* D, *a set $\{(u, f(u)) | p(u)\ \&\ u \in SPP(v)\}$ as* S, *a singleton $\{(v, G(v))\}$ as* P, *a mapping $Q \mapsto \{(u, w) \in D \mid \exists w_1, \ldots w_n : (t_1(u), w_1), \ldots (t_n(u), w_n) \in Q\ \&\ w = g(u, w_1, \ldots w_n)\}$ as* F: $2^D \to 2^D$, *and $\exists w : (v, w) \in (R \cap Q)$ as $\rho(R, Q) : 2^D \times 2^D \to Bool$, then the instantiated algorithm computes $G(v)$ in the following sense: it terminates after iterating* repeat-until *loop $|SPP(v)|$ times at most, upon the termination $(v, G(v)) \in$*Z *and there is no any $w \in Y$ (other than $G(v)$) such that $(v, w) \in$*Z.

Some examples that illustrate the use of DyP template will be given in the conference talk and in a forthcoming full paper.

References:

[1] K.R. Apt, F.S. de Boer, E.-R. Olderog, *Verification of Sequential and Concurrent Programs*, Springer, 2009.

[2] N.V. Shilov, Verification of backtracking and branch and bound design templates, *Modeling and Analysis of Information Systems*, 18 (2011), No. 4, pp. 168–180 (in Russian).

[3] R. Bellman, The theory of dynamic programming, *Bulletin of the American Mathematical Society*, 60 (1954), pp. 503–516.

Informativity of experiments and uncertainty regions of model parameters

Semen I. Spivak

Bashkir State University,
Institute of petrochemistry and catalysis, Russian Academy of Science
Ufa, Russia
s.spivak@bashnet.ru

Keywords: inverse problems, informativity of experiment, uncertainty region

Our work considers problems of mathematical theory of measurements analysis. We assume that a model describes the measurements within the accuracy of the latter, provided that the following set of inequalities is satisfied:

$$|X_{exp} - X_{calc}| \leq \varepsilon \tag{1}$$

where ε is the vector of the maximum allowable inaccuracy of experimental measurements of X.

We define the uncertainty range for each calculated parameter k_i, $i = 1, \ldots, n$, as such an interval

$$d_i = [\min\ k_i, \max\ k_i] \tag{2}$$

that the system (1) is consistent for some values of the input data within that range.

The formulation of the problems of determining the ranges (2) provided that the set of constraints (1) is satisfied belongs to L.V.Kantorovich [1]. Nowdays, the terms *set-membership approach* or *error-bounded data* are usually used in connection to this approach.

The values of ε_j in the system of inequalities (1) are the characteristics of the maximum allowable experimental error. In such case, fulfillment of the conditions (2) means that the model describes the measurements within the limits conditioned by the maximum allowable measurement error, which is quite reasonable.

In our work, we developed Kantorovich's approach in application to the solution of inverse problems of chemical kinetics [2].

A principal feature of Kantorovich's approach is the fact that, based on mathematical programming ideas, it allows the measurement informativity to be analyzed using solutions of the conjugate problem (or the dual problem, in terms of linear programming). The solutions of the conjugate problem allow one to distinguish the points that define the minimum and maximum for each of the constants from a large set of experimental data. If the range d_j of a certain constant appears to be too large, analysis of the solution to the conjugate problem allows us to build the plan of measurements (conditions and accuracy of new experiments) in order to reduce the range for the value defined by some additional requirements.

Thus, the uncertainly ranges

$$d_i = [\min\ k_j, \max\ k_j\,], \qquad j = 1, \ldots, m,$$

for the parameters k_j, set by the equation (2), are ranges within which the inequality (1) is satisfied, i.e., within which the kinetic model does not contradict the measurements. The vector $d = (d_1 \ldots d_m)$ characterizes a degree of uncertainty for each of the target parameters caused by measurement errors. Using this vector, we can determine the measurement accuracy in certain points required to ensure that the degree of uncertainty in the parameters does not exceed a preset value.

The multidimensional uncertainty region will be understood as a set of points that correspond to parameter values in which the relation (1) is valid.

Thus, if the kinetic model of a reaction involves n parameters, the uncertainty region will be n-dimensional. Our goal is to find (in some sense) the uncertainty regions and their two-dimensional projections onto a plane defined by couples of parameters.

The major problem in the use of this method arises in the calculation of multidimensional uncertainty regions. The problems that arise are of both mathematical and physicochemical nature. In particular, the physicochemical interpretation of uncertainty regions becomes the main problem. These problems are the subject of our further studies in this direction.

References:

[1] L.V. Kantorovich, On some new approaches to numerical methods and reduction of observation, *Siberian mathematical journal*, 3 (1962), No. 5, pp. 701–709 (in Russian).

[2] S.I. Spivak, Informativity of kinetic measurements, *Khimicheskaya promyshlennost segodnya*, 2009, No. 9, pp. 52–56 (in Russian).

Analysis of non-uniqueness of the solution of inverse problems in the presence of measurements errors

Semen I. Spivak and Albina S. Ismagilova

Bashkir State University
Ufa, Russia
s.spivak@bashnet.ru

Keywords: informativity, non-uniqueness, kinetic constants

This study deals with inverse problems of identification of mechanisms of complex chemical reactions based on kinetic measurements.

The inverse problem consists in determining the rate constants of elementary steps involved in the mechanism of a complex chemical reaction from experimental data on the concentrations of compounds involved in the reaction.

The main difficulty is that, generally, only some of the compounds involve in a reaction can be measured. This insufficient informativity results in the non-uniqueness of the inverse problem solution.

The purpose of this article is a mathematical study of the informativity problem:

- a classification of non-uniqueness types of solutions of inverse problems of chemical kinetics depending on the type of the experiment is provided;

- a methodology is developed for analysis of informativity of kinetic measurements in the solution of inverse problems, which allows one to determine the number and form of independent combinations of reaction rate constants that can be evaluated unambiguously from various kinetic experiment types;

- it is proven that the measurable characteristics of a reaction mechanism are invariant with respect to certain transformations of kinetic parameters. It is proven that these transformations are group transformations (continuous or discrete, depending on the experiment type).

- a methodology is developed for reduction of systems of differential equations of chemical kinetics to systems with smaller dimensionality under the condition that they remain adequate to the actual sets of measurements.

The non-uniqueness of solutions of inverse problems of chemical kinetics results in the existence of uncertainty regions of kinetic constants. An uncertainty region will be understood as such a region [1] within which variation of kinetic

constants allows kinetic measurements to be described within their accuracy [2-3].

References:

[1] L.V. Kantorovich, About new approaches to the theory of treatment of observations, *Sibirskii Matematicheskii Zhurnal*, 3 (1962), pp. 701–708.

[2] S.I. Spivak, M.G. Slinko, V.I. Timoshenko, V.Yu. Mashkin, Interval estimation in the determination of parameters of a kinetic model, *Reaction Kinetics and Catalysis Letters*, 3 (1974), No. 1, pp. 105–113.

[3] S.I. Spivak, Informativity of kinetic measurements, *Khimicheskaya promyshlennost' segodnya*, 2009, No. 9, pp. 52–56.

Interval estimation of system dynamics model parameters

Semen I. Spivak, Olga G. Kantor

Institute of Social and Economic Research
71, October Avenue; 450054 Ufa, Russia
o_kantor@mail.ru.

Keywords: system dynamics, two-sided bounds for model parameters, L.V. Kantorovich approach.

The method of system dynamics in the case study of a model with two variables suggests dependency of the following form:

$$\begin{aligned} \frac{dx}{dt} &= a_1 x^{\alpha_1} y^{\beta_1} - a_2 x^{\alpha_2} y^{\beta_2}, \\ \frac{dy}{dt} &= a_3 x^{\alpha_3} y^{\beta_3} - a_4 x^{\alpha_4} y^{\beta_4}. \end{aligned} \tag{1}$$

The direct view of the system dynamics model (1) depends on the parameter values $\{ a_i, \alpha_i, \beta_i \}$, $i = \overline{1,4}$, which are defined based upon available statistical data. In the case where the researcher takes into consideration the variables, special methods are applied to obtain point estimates of parameters in response to the complex relationship between said variables. The aforementioned relationship does not allow even a first approximation to determine the parameters of system dynamics models. Moreover, it is necessary to know the permissible range of variation for the performance of the numerical experiment to "customize" the model.

The proposed method is based upon a linearization of system (1). Using Maclaurin expansion of the right-hand sides of the equations (1), based on the available observations, it is possible to identify point and interval estimates specifically for parameters $\{a_i\}$, $i = \overline{1,4}$. Therefore, the solution of the problem will be carried out in two steps. On the first step, based on Maclaurin series expansions of the equations (1), we define the point and interval estimates: $\{a_i^0\}$, $i = \overline{1,4}$ and $\{[a_i^-; a_i^+]\}$, $i = \overline{1,4}$. In the second step, we compute point and interval estimates for all the parameters of the model (1), using a linear expansion of the equations in Taylor series centered at $\{a_i^0, \alpha_i = 0, \beta_i = 0\}$, $i = \overline{1,4}$. Estimation of the intervals $\{[a_i^-; a_i^+]\}$, $i = \overline{1,4}$, allows variation of the expansion center at the second step.

The number of observations in practical problems is large, so the problem solved a priori is overdetermined. Moreover, they are characteristically flawed; approximate initial data implies the failure of requirements for well-posed problems. These circumstances significantly limit the number of methods for determining point and interval estimates of the parameters of the system dynamics models. In this regard, a particularly interesting approach of L.V. Kantorovich, who first floated the idea of obtaining accurate two-sided bounds for model parameters and the location of desired surfaces and observed values.

The problem of determining the parameters of the model (1) for each of the model equations separately, taking into account the above mentioned features, is reduced to solving an inconsistent system of m linear equations with n unknowns. Therefore, to verify that the calculated and experimental data agree in the deviation, for example, in the first equation of system (1), we have to consider

$$\eta_i = \left(\frac{dx}{dt}\right)^{calculated}\Bigg|_i - \left(\frac{dx}{dt}\right)^{experimental}\Bigg|_i, \qquad i = \overline{1,m}. \tag{2}$$

The standard way to solve the problem of determining the parameters of the model (1) is to minimize deviations $\{\eta_i,\ i = \overline{1,m}\}$ in terms of a certain introduced criterion. The basic selection criterion is the information on the distribution of measurement errors. In real systems, such information is missing as a rule, while we have an information on the maximum permissible error of measurement at our disposal. This fact is the main argument in favor of the approach of L.V. Kantorovich.

The condition that the model describes the observed values leads to a system of inequalities

$$|\eta_i| \le \varepsilon_i, \qquad i = \overline{1,m}, \tag{3}$$

where $\varepsilon_i - i^{th}$ is the i-th measurement error. Numerical solution of the abobe system involves the use as an initial approximation for at least one point, providing the validity of all the relations (3). This point can be found by enforcing the optimum condition for each criterion, which characterizes the agreement between the calculated and experimental data. For example, one such criterion can be the sum of squared deviations.

A significant advantage of this approach is its capability to take into account a priori constraints on the values of the parameters with the required dependencies, known from additional sources that can significantly reduce the uncertainty of problems.

Algorithm for sparse approximate inverse preconditioners refinement in conjugate gradient method

Irina Surodina[1] and Ilya Labutin[2]

[1]Institute of Computational Mathematics and Mathematical Geophysics
6, Akademika Lavrentjeva, Novosibirsk, 630090, Russia
sur@ommfao1.sscc.ru
[1]A.A. Trofimuk Institute of Petroleum Geology and Geophysics
3, Akademika Koptyuga, Novosibirsk, 630090, Russia
ilya.labutin@gmail.com

Keywords: sparse approximate inverse, conjugate gradient, GPU

The Conjugate Gradient (CG) algorithm is one of the best known iterative methods for solving linear systems with symmetric, positive definite matrix [1]. The performance of the CG can be dramatically increased with the suitable preconditioner. The concept of the preconditioning in iterative methods is to transform the original system into an equivalent system with the same solution, but a lower condition number. However, the computational overhead of applying the preconditioner must not cancel out the benefit of fewer iterations [2, 3].

Modern parallel implementations of the preconditioned conjugate gradient (PCG) on the graphical processing units (GPUs) uses sparse approximate inverse (AINV) preconditioners due to attractive features. First, the columns or rows of the approximate inverse matrix can be generated in parallel. Second, the preconditioner matrix is used in PCG through matrix-vector multiplications which are easy to parallelize [3]. Thereby the accuracy of the inverse approximation is important.

In this work, we present an algorithm for building a series of AINV preconditioners with arbitrary high approximation accuracy. Presented algorithm derives from the Hotelling-Schulz algorithm for inverse matrix elements correction [4, 5]. In this algorithm it is assumed that $\mathbf{D_0}$ is a certain initial approximation of the $\mathbf{A^{-1}}$. With the condition

$$\|\mathbf{R_0}\| \leq \mathbf{k} < \mathbf{1}$$

where

$$\mathbf{R_0} = \mathbf{I} - \mathbf{AD_0}$$

we can build the sequence:

$$\mathbf{D_1}=\mathbf{D_0}(\mathbf{I}+\mathbf{R_0}), \quad \mathbf{R_1}=\mathbf{I}-\mathbf{AD_1}$$

$$\mathbf{D_2}=\mathbf{D_1}(\mathbf{I}+\mathbf{R_1}), \quad \mathbf{R_2}=\mathbf{I}-\mathbf{AD_2}$$

$$\ldots\ldots\ldots\ldots$$

$$\mathbf{D_m}=\mathbf{D_{m-1}}(\mathbf{I}+\mathbf{R_{m-1}}), \quad \mathbf{R_m}=\mathbf{I}-\mathbf{AD_m}$$

It is shown that obtained sequence converges quickly to the $\mathbf{A^{-1}}$.

Due to presented approach we refined existed Jacobi and Symmetric Successive Over-Relaxation preconditioners [6] with reasonable approximation accuracy. All algorithms were implemented on NVIDIA™ GPU and numerical results obtained for real-life matrices.

References:

[1] R. Barrett, M. Berry, H. Van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, `www.netlib.org/templates/templates.pdf`

[2] J. Dongarra, I. Duff, D. Sorensen, H. Van der Vorst, *Numerical Linear Algebra for High-Performance Computers*, SIAM, Philadelphia, PA, 1998.

[3] R. Li, Y. Saad, *GPU-Accelerated Preconditioned Iterative Linear Solvers*, Technical Report umsi-2010-112, Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN, 2010.

[4] H. Hotelling, Analysis of a complex of statistical variables into principal components, *Journal of Educational Psychology*, 24 (1933), pp. 417–441 and 498–520.

[5] G. Schulz, Iterative Berechnung der reziproken Matrix, *Z. Angew. Math. Mech.*, 13 (1933), pp. 57–59..

[6] M. Ament, G. Knittel, D. Weiskopf, W. Strasser, A parallel preconditioned conjugate gradient solver for the Poisson problem on a multi-GPU platform, in *Proc. 18th Euromicro Conference on Parallel, Distributed and NetWork-Based Computing, Pisa, Italy, February 17-19, 2010*, pp. 583–592.

Computer-assisted error analysis for second-order elliptic equations in divergence form

Akitoshi Takayasu[1,2] and Shin'ichi Oishi[1,3]

[1]Faculty of Science and Engineering, Waseda university
[2]JSPS research fellow
[3]CREST/JST

3-4-1 Okubo, Shinjuku, Tokyo, 169-8555 Japan
takitoshi@aoni.waseda.jp

Keywords: computer-assisted analysis, finite element method, constructive a priori error estimates

In this talk, a method of computer-assisted error estimate is proposed for second-order divergence form problems on a bounded domain $\Omega \subset \mathbb{R}^N$ $(N = 1, 2, 3)$ with the Dirichlet boundary condition:

$$\begin{cases} -\mathrm{div}(a(x)\nabla u) = f(x), & \text{in } \Omega, \\ u|_{\partial\Omega} = 0. \end{cases}$$

Assuming that $f(x) \in L^2(\Omega)$ and $a(x) \in W^{1,\infty}(\Omega)$, the solvability of the elliptic problem with degenerate coercivity is shown. Here, degenerate coercivity means that there is a point $x \in \Omega$ satisfying $a(x) = 0$. Using a bilinear form, a weak formulation of the original problem is obtained.

$$\text{Find } u \in H_0^1(\Omega) \text{ satisfying } (a(x)\nabla u, \nabla v) = (f, v), \quad \forall v \in H_0^1(\Omega).$$

The solvability of the weak solution is related to the *inf-sup condition*, which is sometimes called as LBB-condition in FEM theory [1,2,3]. Using verified computations, the lower bound of a value with respect to the inf-sup condition is bounded. It is based on Fredholm's alternative theorem.

After that let $V_h \subset H_0^1(\Omega)$ be a certain finite element subspace. Constructive a priori error estimate is obtained for a certain orthogonal projection $\mathcal{R}_h : H_0^1(\Omega) \to V_h$ (Ritz-projection) defined by

$$(a(x)\nabla(u - \mathcal{R}_h u), \nabla v_h) = 0, \quad \forall v_h \in V_h.$$

If conditions continuity and coercivity are satisfied, Céa's lemma gives desired a priori error estimate: $\|u - \mathcal{R}_h u\|_{H_0^1} \leq C(h)\|f\|_{L^2}$. On the other hand, it is difficult to obtain the error bound with degenerate coercivity. Our main theorem gives the solvability and constructive a priori error estimate based on the inf-sup condition. Further, convergence rate of the error estimate is analyzed with computer-assistance. Computational results will be presented to show the solvability and error estimates for the weak solution of original problems.

References:

[1] O. Ladyženskaja, *Mathematicheskie Voprosy Dinamiki Vyazkoi Neszhimaemoi Zhidkosti*, Gosudarstv. Izdat. Fiz. Mat. Lit., Moscow, 1961.

[2] A.K. Aziz and I. Babuška, Survey lectures on the mathematical foundations of the finite element method, In *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations* (A.K. Aziz, ed.), Academic Press, New York, 1972.

[3] F. Brezzi, On the existence, uniqueness, and approximation of saddle-point problems arising from Lagrangian multipliers, *RAIRO Anal. Numer.*, 8 (1974), No. 2, pp. 129–151.

On affinity of physical processes of computing and measurements

Lev S. Terekhov[1] and Andrey A. Lavrukhin[2]

[1]Omsk Division of Sobolev Institute of Mathematics,
Siberian Branch of the RAS (OD IM SB RAS)
13, Pevtsova st., 644099 Omsk, Russia
lev.terekhov@gmail.com
[2]Omsk State Transport University (OSTU)
35, Marx ave., 644046 Omsk, Russia
lavruhinaa@gmail.com

Keywords: natural interval, dynamic uncertainty relation, natural derivative, numerical tests

Interval numbers and interval analysis [1–3] have been inspired primarily by computer calculation. In its turn, the computer calculation as a physical process borrows and inherits the interval structure of results of natural measurements.

Uncertainty relation (UR) of classical physics adequately describes the uncertainty intervals of natural measurements of a pair of mutually independent variables. To adequately determine the uncertainty interval of natural measurements of a pair of variables related functional dependence, the generalization of classical UR was postulated [4]. The uncertainty Δf of a measured dependent variable f proposed as equal to the sum of random and deterministic dynamic components. The generalization of classical UR which takes into account the dynamic of physical process was called dynamic uncertainty relation (DUR) [5, 6]. DUR generates an algorithm that provides the minimal uncertainty $\Delta f_{\min}(\Delta t^*)$ of the measured dependent variable f when it is measured within the interval with optimal width Δt^* of the independent variable t. The interval $\Delta f_{\min}$ is a potential accuracy of natural measurement and is not an artifact. The optimal interval Δt^*, hereinafter referred to as natural, is locally determined for each i-th sample unit: $\Delta t_i^* = \left(\sqrt{\mu_{i-1}} \cdot \left|\Delta f_{\min}(\Delta t_{i-1}^*)/\Delta t_{i-1}^*\right|\right)^{-0,5}$, $i = 2, 3, ...$ and is a measuring and computing element of the adaptive algorithm. In addition to the computation of natural interval Δt^* for each sample unit the interval $\Delta f_{\min}$ and interval one-dimensional derivative $\Delta f_{\min}/\Delta t^*$ are also calculated. The initial value of natural interval Δt_1^* is calculated by other algorithms. Natural interval is limited to the values $\Delta t^* > 0$ and its width Δt^* can not be arbitrarily reduced. Computing the derivative $\Delta f_{\min}/\Delta t^*$ on an interval is free from the problem of

inaccuracy. The proposed process of measurement is adaptation of the width Δt of measurement interval to the derivative $\Delta f_{\min}/\Delta t^*$ on this interval and to the signal-to-noise ratio μ of the measured variable f at each sample unit. In distinction to the known methods of natural measurements the proposed adaptive algorithm combines samples of both natural measurements and computer calculations, carried out simultaneously as part of a single measurement-and-calculation process. The purpose of this work is to show, on examples, that the algorithm originally found for natural measurements is identical, to within the accepted approximation, to the algorithm of computer calculations.

Numerical tests of the proposed method with known methods carried out by comparing the accumulated errors in the whole field of numerical integration. For the test calculations was chosen the integral $I = \int_{\exp(-p)}^{1}(1/x)\,dx = p$. Numerical integration of the method of trapezoids and average rectangles was done with optimal constant step, providing a minimum cumulative error over the entire area of integration. The integration of adaptation is also realized by schemes trapezoids and rectangles. A comparison shows a decrease in few times in the number of nodes by the method of adaptation compared to the classic, the accumulated error of the method of adaptation shows a decrease on 1–2 orders of magnitude.

As shown by numerical tests, the computer calculations and the natural measurements has affinity and structural identity to within the accepted approximation.

References:

[1] G. Schröder, Differentiation of interval functions, *Proceedings of the American Mathematical Society*, 36 (1972), No. 2, pp. 485–490.

[2] Y.I. Shokin, *Interval Analysis*, Nauka, Novosibirsk, 1981 (in Russian).

[3] S.P. Shary, *Finite-dimensional Interval Analysis*, `www.nsc.ru/interval/Library/InteBooks/SharyBook.pdf` (in Russian).

[4] L.S. Terekhov, On the complete error radio wave measurements of the inhomogeneous plasma layer, *Geomagnetism and Aeronomy*, 38 (1998), No. 6, pp. 142–148.

[5] L.S. Terekhov, On quantization of the uncertainty in measureable macroscopic quantity, *Russian Physics Journal*, 49 (2006), No. 9, pp. 981–986.

[6] L.S. Terekhov, Construction of an analogue of interval values of the derivative, `http://conf.nsc.ru/niknik-90/en/reportview/39121`

Automatic code transformation to optimize accuracy and speed in floating-point arithmetic

Laurent Thévenoux[1], Matthieu Martel[2] and Philippe Langlois[3]

[1] Univ. Perpignan Via Domitia, Digits, Architectures et Logiciels Informatiques, F-66860, Perpignan, France
[2] Univ. Montpellier II, Laboratoire d'Informatique Robotique et de Microélectronique de Montpellier, UMR 5506, F-34095, Montpellier, France
[3] CNRS, Laboratoire d'Informatique Robotique et de Microélectronique de Montpellier, UMR 5506, F-34095, Montpellier, France
firstname.lastname@univ-perp.fr

Keywords: floating-point arithmetic, accuracy, compensation, code transformation, instruction level parallelism, program optimization

Algorithms using IEEE-754 floating-point arithmetic [1] may suffer from inaccuracy generated by rounding since floating-point numbers are approximations of real numbers. This inaccuracy is a critical matter in scientific computing as well as for embedded systems. Several techniques have been introduced and applied to improve the accuracy of numerical algorithms, as for instance compensation or error-free transformations [2], ...

In practice these solutions are mainly known by experts and the corresponding program transformations must be implemented manually. Our objective is to allow the standard software developer to automatically transform his/her code. This transformation is actually an optimization since we aim to take into account two opposite criteria: accuracy and execution time. A first step towards this automatic optimization is presented in this work.

We propose to automatically introduce at the compile-time compensation steps in (parts of) the floating-point computations. We present a tool to parse C codes and to insert compensated floating-point operations. A new C code is generated by replacing in the original code, floating-point operations $(+, -, \times)$ by their respective compensated algorithms: TwoSum, TwoProd, etc. [2]. These compensated terms are computed and accumulated in parallel to the original operations. This provides a compensated computation that improves the accuracy of specific computing patterns.

We apply this approach to some test cases aiming to reproduce automatically what experts have done manually. In [3] for instance the authors propose a compensated polynomial evaluation. They evaluate the Horner form of the polynomial $p(x) = (0.75 - x)^5(1 - x)^{11}$ close to its multiple roots. They show that compensation improves the accuracy. The same results are generated by our automatic transformation as reported in Figure 1. This figure shows the value of $p(x)$ close to one of its roots, before and after the automatic transformation. As expected, original results are meaningless while the transformed code provides more accuracy and yields a smoother polynomial evaluation. Our tool allows non expert user to obtain automatically, quickly and easily such accuracy improvement.

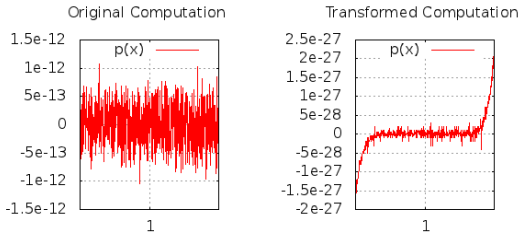

Figure 2: The leftmost graph shows $p(x)$ around his root 1 computed in double precision with the Horner algorithm. The rightmost graph shows the results of the automatically generated code.

The next step is to take into account the second optimization criteria: the execution time. Instruction level parallelism (ILP) or instructions like the FMA (Fused Multiply-Add) can be exploited by modern architectures to save execution time. Because we compute the compensated terms in parallel to the original arithmetic expressions, our transformation introduces ILP that favors a fast execution. This reduces the over-cost of these kind of transformation. We complete the transformation tool with the automatic analysis of this over-cost. So these two aspects will be integrated in a future work in order to optimize code. Time and accuracy criteria will be jointly optimized using trade-offs.

References:

[1] *IEEE Standard for Binary Floating-point Arithmetic*, Revision of Std 754-1985, 2008.

[2] T. Ogita, S.M. Rump, Sh. Oishi, Accurate Sum and Dot Product, *SIAM J. Sci. Comput.*, 26 (2005).

[3] S. Graillat, Ph. Langlois, N. Louvet, Algorithms for accurate, validated and fast polynomial evaluation, *Japan Journal of Industrial and Applied Mathematics*, 26 (2009), pp. 191–214.

Interval matrix multiplication on parallel architectures

Philippe Théveny and Nathalie Revol

LIP (UMR 5668 CNRS - ENS de Lyon - INRIA - Université Claude Bernard),
Université de Lyon
ENS de Lyon, 46 allée d'Italie
69007 Lyon, France
Philippe.Theveny@ens-lyon.fr, Nathalie.Revol@ens-lyon.fr

Keywords: interval arithmetic, matrix multiplication, parallel architectures

Getting efficiency when implementing interval arithmetic computations is a difficult task. The work presented here deals with the efficient implementation of interval matrix multiplication on parallel architectures.

A first issue is the choice of the formulas. The main principle we adopted consists in resorting, as often as possible, to optimized routines such as the BLAS3, as implemented in Intel's MKL for instance. To do so, the formulas chosen to implement interval arithmetic operations are based on the representation of intervals by their midpoint and radius. This approach has been advocated by S. Rump [3] and used, in particular, in his implementation IntLab. It is recalled that a panel of formulas for operations using the midpoint-radius representation exists: exact formulas can be found in A. Neumaier [1, pp. 22–23], S. Rump [3] gave approximate formulas with less operations, H.D. Nguyen [2] gave a choice of formulas reaching various tradeoffs in terms of operation count and accuracy. These formulas for the addition and multiplication of two intervals are used by [2,3] in the classical formulas for matrix multiplication and can be expressed as operations (addition and multiplication) of matrices of real numbers (either midpoints or radii), S. Rump recapitulates some such matrix expressions in [4]. In this presentation, the merits of each approach are discussed, in terms of number of elementary operations, use of BLAS3 routines for the matrix multiplication, and of accuracy. The comparison of the relative accuracies are based on the assumption that arithmetic operations are implemented using exact arithmetic. We also give a comparison of these accuracies, assuming that arithmetic operations are implemented using floating-point arithmetic.

A second issue concerns the adaptation to the architecture. Indeed, the architectures targeted in this study are parallel architectures such as multicores or GPU. When implemented on such architectures, some measures such as the arithmetic operations count are no more relevant: the measured execution times do not relate directly to the operations count. This is explained by considerations on memory usage, multithreaded computations... We will show some experiments that take these architectural parameters into account and reach good performances. We will give some tradeoffs between the memory consumption and memory traffic: it can for instance be beneficial to copy (parts of) the involved matrices in the right caches to avoid cache misses and heavy traffic.

References:

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

[2] H.D. Nguyen, N. Revol and P. Théveny, Tradeoffs between accuracy and efficiency for optimized and parallel interval matrix multiplication, *PARA 2012*.

[3] S.M. Rump, Fast and parallel interval arithmetic, *BIT Numerical Mathematics*, 39 (1999), No. 3, pp. 539–560.

[4] S.M. Rump, Fast interval matrix multiplication, *Numerical Algorithms*, 2011, 34 pages, to appear.

Fast infimum-supremum interval operations for double-double arithmetic in rounding-to-nearest

Naoya Yamanaka and Shin'ichi Oishi

Research Institute for Science and Engineering, Waseda University
3-4-1 Okubo Shinjuku, Tokyo, 169-8555 Japan
naoya_yamanaka@suou.waseda.jp

Keywords: double-double arithmetic, rounding-to-nearest, interval arithmetic

In a numerical calculation sometimes we need higher-than double-precision floating-point arithmetic to allow us to be confident of a result. One alternative is to rewrite the program to use a software package implementing arbitrary-precision extended floating-point arithmetic such as MPFR [1] or ARPREC [2], and try to choose a suitable precision. There are intermediate possibilities intermediate between the largest hardware floating-point format and the general arbitrary-precision software which combine a considerable amount of extra precision with a relatively speaking modest factor loss in speed. An alternative approach is to store numbers in a multiple-component format, where a number is expressed as an unevaluated sums of ordinary floating-point words, each with its own significand and exponent. The multiple-digit approach can represent a much larger range of numbers, whereas the multiple-component approach has the advantage in speed. Sometimes merely doubling the number of bits in a double-floating-point fraction is enough, in which case arithmetic on double-double operands would suffice.

A double-double number is an unevaluated sum of two double precision numbers, capable of representing at least 106 bits of significand. A natural idea is to manipulate such unevaluated sums. This is the underlying principle of double-double arithmetic. It consisted in representing a number x as the unevaluated sum of two basic precision floating-point numbers:

$$x = x_h + x_l$$

such that the significands of x_h and x_l do not overlap, which means here that

$$|x_l| \leq \mathbf{u} \, |x_h| \, ,$$

where $\mathbf{u}$ denotes the machine epsilon; in double precision $\mathbf{u} = 2^{-53}$.

Meanwhile, the interval arithmetic is a method for finding lower and upper bound on the value of a result by performing a computation in a manner that preserves these bounds. Thus it allows to develop numerical methods that yield reliable results. The infimum-supremum interval arithmetic is a method of finding lower and upper bounds for the possible values of a result by performing a computation on a manner which preserves these bounds, and thus developing numerical method that yield reliable results. Denote the set of intervals $\{[\underline{x},\ \overline{x}]\ :\ \underline{x},\ \overline{x} \in \mathbb{R}\}$ by $\mathbb{IR}$. then provided $0 \notin Y$ in the case of division, the result of the power set operation $X \circ Y$ for $X, Y \in \mathbb{IR}$ is again an interval, and we have

$$X \circ Y := [\min(\underline{x} \circ \underline{y}, \underline{x} \circ \overline{y}, \overline{x} \circ \underline{y}, \overline{x} \circ \overline{y}),\ \max(\underline{x} \circ \underline{y}, \underline{x} \circ \overline{y}, \overline{x} \circ \underline{y}, \overline{x} \circ \overline{y})].$$

In this talk we will describe fast algorithms to compute interval operations for double-double arithmetic. These algorithms are working in rounding to nearest, so that they don't need to take time for changing rounding mode. These algorithms evaluate the rounding error of the approximate value in rounding to nearest mode, and find an interval represented by double-double numbers including the true interval.

References:

[1] L. Fousse, G. Hanrot, V. Lefévre, P. Pélissier, P. Zimmermann, MPFR: A multiple-precision binary floating-point library with correct rounding, *ACM Transactions on Mathematical Software (TOMS)*, 33 (2007), No. 2, article 13, 15 pp.

[2] D.H. Bailey, Y. Hida, X.S. Li and B. Thompson, *ARPREC: an arbitrary precision computational package*, LBNL, Berkeley, 2002, 8 pp., `http://crd-legacy.lbl.gov/~dhbailey/dhbpapers/arprec.pdf`

[3] Y. Hida, X.S. Li and D.H. Bailey, *Quad-Double Arithmetic: Algorithms, Implementation, and Application*, Report LBL-46996, October 30, 2000, `http://crd-legacy.lbl.gov/~xiaoye/TR_qd.ps.gz`

[4] T. Ogita, S. M. Rump and S. Oishi, Accurate sum and dot product, *SIAM J. Sci. Comput.*, 26 (2005), No. 6, pp. 1955–1988.

Interval polynomial interpolation for bounded-error data

Ziyavidin Yuldashev, Alimzhan Ibragimov, Shukhrat Tadjibaev

National University of Uzbekistan
Vuzgorodok.
100174, Tashkent, Uzbekistan
{ziyaut, alim-ibragimov, tajibaevs}@mail.ru

Keywords: interval arithmetic, interval estimation, interval extension of functions

We consider a function $f(x)$, for which an interval extension $\boldsymbol{f}(\boldsymbol{x})$ is defined on $[a,b]$. Assume further that the intervals $\boldsymbol{y}_i = \boldsymbol{f}(\boldsymbol{x}_i)$ are defined for $\boldsymbol{x}_i = [\underline{x}_i, \overline{x}_i] \subseteq [a,b]$, $i = 1,2,\ldots,n$, such that

$$f(x_i) \in \boldsymbol{y}_i \ \text{ for any } x_i \in \boldsymbol{x}_i, \qquad i = 1,2,\ldots,n. \tag{1}$$

The *interpolation problem* for the interval-valued function $\boldsymbol{f}(\boldsymbol{x})$ requires construction of an interval-valued function $\boldsymbol{g}(\boldsymbol{x})$ that satisfies

$$\boldsymbol{g}(\boldsymbol{x}_i) = \boldsymbol{y}_i, \qquad i = 1,2,\ldots,n. \tag{2}$$

The problem of determining the function $\boldsymbol{g}(\boldsymbol{x})$, under conditions (1)–(2)), will be referred to as ***IIN1***. Similar to the real case, this problem has no unique solution.

Let the points $\boldsymbol{x}_i = [\underline{x}_i, \overline{x}_i] \subseteq [a,b]$, $i = 0,1,\ldots,n$, be such that

$$\underline{x}_0 = a, \qquad \boldsymbol{x}_i \cap \boldsymbol{x}_j = \varnothing \quad \text{for } \ i \neq j, \qquad \overline{x}_n = b, \tag{3}$$

and any real restriction of the function $\boldsymbol{g}(\boldsymbol{x})$ is a polynomial of the degree n:

$$\operatorname{Rs} \boldsymbol{g}(\boldsymbol{x}) \ \in \ \left\{ \textstyle\sum_{i=0}^{n} a_i x^i \mid a_i \in \mathbb{R} \right\}. \tag{4}$$

The problem (2)–(3)–(4) will be denoted as ***IIN2***.

Let the points $\boldsymbol{x}_i = [\underline{x}_i, \overline{x}_i] \subseteq [a,b]$, $i = 0,1,\ldots,n$, be such that

$$\operatorname{wid} \boldsymbol{x}_i = \operatorname{wid} \boldsymbol{x}_j \quad \text{for } \ i \neq j, \tag{5}$$

in particular,

$$\underline{x}_{i+1} - \overline{x}_i = \underline{x}_{i+2} - \overline{x}_{i+1} \quad \text{for } \ i = 0,1,\ldots,n-2, \tag{6}$$

The problem (2)–(6) is designated as ***IIN3***.

In our work, we have investigated the above problems and verified the results by numerical tests. In particular, for the solution of the problem ***IIN2***, we propose to use the function

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{L}_n(\boldsymbol{x}) = \sum_{k=0}^{n} \boldsymbol{y}_k \prod_{i \neq j}^{n} \frac{\boldsymbol{x} \ominus \boldsymbol{x}_j}{\boldsymbol{x}_k \ominus \boldsymbol{x}_j}, \tag{7}$$

where "$\ominus$" means non-standard Markov subtraction, and any real restriction of $\boldsymbol{g}(\boldsymbol{x})$ gives a Lagrange interpolation polynomial. We have proved

Theorem. *For the function $\boldsymbol{L}_n(\boldsymbol{x})$, defined by* (7)*, the conditions* (1)–(2) *are satisfied, and the following estimate is valid:*

$$\| \operatorname{wid} \boldsymbol{L}_n(\boldsymbol{x}) \| \leq \frac{M}{(n+1)!} \left\| \prod_{i=0}^{n} (\boldsymbol{x} - \boldsymbol{x}_i) \right\|, \tag{8}$$

where $\|[a,b]\| = \max\{|a|, |b|\}$, $M = \max_{x \in [a,b]} \left| f^{(n+1)}(x) \right|$.

Analogous results are also obtained for interval versions of the alternative interpolation formulae by Newton, Hermit and Chebyshev.

The interpolation interval polynomials constructed have been implemented and integrated into a scalable program system with an appropriate interface [1]. It enables one to compute the values of the interval interpolation formulae by simple overloading of the corresponding interval operations to those from a necessary interval arithmetic [2].

References:

[1] Z.Kh. Yuldashev, A.A. Ibragimov, P.Zh. Kalhanov, A package of interval algorithms for general public. *Registered in The state catalogue of the computer programs of Republic Uzbekistan, Certificate of official registration of the computer programs* No. DGU 02201, Tashkent, 5/19/2011.

[2] Z.Kh. Yuldashev, A.A. Ibragimov, P.Zh. Kalhanov, A program system for computing values of interval algebraically admissible expressions in various interval arithmetics. *Registered in The state catalogue of the computer programs of Republic Uzbekistan, Certificate of official registration of the computer programs* No. DGU 02202, Tashkent, 5/19/2011.

ANOVA, ANCOVA and time trends modeling: solving statistical problems using interval analysis

Sergei Zhilin

Altai State University
61, Lenin ave., 656049, Barnaul, Russia
sergei@asu.ru

Keywords: linear regression, interval error, ANOVA, ANCOVA, time trend

Interval approach to regression analysis meets a wide variety of real world applications and can be competitive to traditional statistical methods because its basic hypotheses are simpler, and interval representations of uncertainty are more natural for practitioners. Construction and analysis of linear regression

$$y = X\beta + \varepsilon \tag{1}$$

with unknown but bounded error ε is a well-studied area. A number of authors propose techniques for interval estimation of forecast and regression parameters, outlier detection, and experimental design for this model (e.g., [1] and references therein). In this work, we extend the interval approach to traditional statistical problems such as analysis of variance (ANOVA), analysis of covariance (ANCOVA), and time trend modeling in linear regression analysis.

ANOVA and ANCOVA refer to regression problems with qualitative predictors. The former assumes all the predictors are categorical, while the latter deals with a mixture of quantitative and qualitative predictors. Qualitative predictors can be incorporated in the regression model (1) by introducing "dummy" variables [2].

A k-level qualitative predictor requires $k-1$ dummy variables for its representation. One parameter is used to represent the overall mean effect or the mean of some reference level, and other levels are coded by values of $k-1$ variables. The coding scheme of levels is not unique, and its choice should be based on convenience of interpretation. The most popular scheme assumes dummy variables are binary (equal 1 for a corresponding level and 0 for others). In such a case, the coefficients of dummy variables act as supplemental interceptors and represent effects of switching to their levels. In statistics, coefficient estimation

is followed by performing statistical significance tests of the estimated parameters. Interpretations of diagnostic tests (F-test and t-test) rest heavily on the model assumptions, and sometimes the results of tests are more difficult to interpret if the model's assumptions are violated [3]. For example, if the error does not have a normal distribution, in small samples the estimated parameters do not follow normal distributions and complicate inference.

Boundedness of the error allows one to obtain certain (not confidential) interval estimates of parameters $\boldsymbol{\beta}_i$ which represent margins of effects and intervals of possible values of the regression output $\boldsymbol{y}^*$:

$$\boldsymbol{\beta}_i = \left[\min_{\beta\in B}\beta_i, \max_{\beta\in B}\beta_i\right], \quad \boldsymbol{y}^* = \left[\min_{\beta\in B}X^*\beta, \max_{\beta\in B}X^*\beta\right],$$

where $B = \bigcap_{i=1}^{N}\{\beta \mid |X_i\beta - y_i| \le \overline{\varepsilon}\}$, (X_i, y_i) is a row in a table of observations, and $\overline{\varepsilon}$ is upper bound of the error. Certain interval estimates do not need significance testing and may be interpreted explicitly by a researcher. In particular, testing null hypothesis of zero difference of coefficients can be replaced by checking whether an interval parameter estimate contains zero. It is easy to find the minimum value of the error bound $\overline{\varepsilon}^*$ under which the samples remain consistent with the model ($B \neq \varnothing$). The value of $\overline{\varepsilon}^*$ is very important additional information produced in the interval approach because it characterizes model precision and its relation to the dataset. We consider one of the simplest ANOVA-type problems (fixed effects, one-way classification), but the proposed technique also is applicable to more complex variants of ANOVA and ANCOVA.

Constructing a regression equation which takes into account time trends is yet another important problem where dummy variables also are helpful. There are many variants of this problem but the main idea and technique remains the same. Only the structure of the regression equation and the manner of dummy-coding of time moments may differ from one specific application to another. Using simple data sets from [2] we show how this technique can be used for the construction and analysis of a regression that takes into account two different time trends.

References:

[1] S.I. Zhilin, Simple method for outlier detection in fitting experimental data under interval error, *Chemometrics and Intelligent Laboratory Systems*, 88, No. 1 (15 August 2007), pp. 60–68.

[2] N.R. Draper, H. Smith, *Applied Regression Analysis*, Wiley, 1981.

[3] L.L. Harlow, S.A. Mulaik, J.H. Steiger, *What If There Were No Significance Tests?* Lawrence Erlbaum Associates, London, 1997.

Repeated filtration of numerical results for reliable error estimation

Vladimir Zhitnikov, Nataliya Sherykhalina and Sergey Porechny

Ufa State Aviation Technical University
12, K. Marx str.
450000 Ufa, Russia
zhitnik@ugatu.ac.ru

Keywords: reliable computing, numerical filtration, accuracy increase

Impressive successes in scientific calculations are achieved by methods of interval analysis. Nevertheless, there is a great number of developed numerical methods and their implementation as computer programs, which either have to be replaced almost completely by new methods to be able to benefit from interval analysis, or have to be modified by methods post-processing results.

A method is proposed, that post-processes numerical results in order to provide physical reliability of the obtained results along with errors estimations. Physical reliability can be achieved by the determination of an approximate value of the required parameter (and this value is called *standard*), its error estimation (the indeterminacy interval) and by a final verification consisting in intersecting intervals obtained by different ways.

Let us consider a problem discretized using meshes (or grids) and let us vary the number of grid knots for different discretizations of the same problem. There is a finite set of results, each corresponding to a different mesh. Each of the obtained values can be considered as a multicomponent model [1], i.e. as the sum of the required value and a few components of the error. The important feature of such a representation is the presence of an unknown addend which can contain the remainder term, a roundoff error and other constituents due to both the numerical method and the concrete program realization. In particular, the component due to roundoff errors does not tend to zero when the number of mesh nodes increases, but it increases in most cases.

In order to estimate the error term, it is proposed to divide the problem of the determination of the required value into two separate ones. The first subproblem is the mathematical model identification of numerical experiment results and the second subproblem is the test of obtained results with the help of some known particular solutions or some other methods.

The first subproblem does not consist in the determination of the theoretical value of the required parameter. Rather, it consists in the decomposition of the result in constituents (components) on some known beforehand or experimentally determined basis. In this latter case, the components have another meaning, because it is known that the main components of the error along with a constant are not included in the unknown addend. This first subproblem can be solved approximately by repeated numerical filtration. Filtration consists in eliminating of error component by means of linear or nonlinear combination of some results (as in Romberg, Aitken, Winn and other methods). Filtration formula is determined by the type of basis and the rule of grid knots choice. Filtration provides an approximate value of the required parameter and error estimation. In this work, we propose to separate the determination of the standard value and the estimation of the error. For this purpose, another filtration is conducted first. It proceeds by eliminating the standard value from the equations taken in pairs, somehow as in Gaussian elimination. Then, further filtrations yield estimations of the error independently of the standard value and choice of the minimal one from the set of error estimations, or a combination of the ones nearest to the minimum. Then the determination of the standard can be obtained by the filtration of the original system up to the number of the grid knots and filtration number corresponding to the minimum.

The second subproblem is testing. If some particular exact solution is known, this is the verification of whether it belongs to the obtained interval. It can also consists in comparing with an approximate solution, that is obtained independently by another numerical method: in this case, verification is obtained by intersecting the intervals centered in the approximate solution and of radius the error. This method based on additional information does not influence the formerly obtained estimations, as they were obtained independently by filtration. It only confirms them or refutes them. The theoretical estimation of reliability (the confidence probability) of joint result of these two problems decision is obtained [2].

References:

[1] V.P. Zhitnikov, N.M. Sherykhalina, *Modeling of Gravity Fluid Flows with Using of Multicomponent Analysis Methods*, Gilem, Ufa, 2009.

[2] V.P. Zhitnikov, N.M. Sherykhalina, Certainty estimation of numerical results at presence of several methods of solution of the problem, *Computation Technologies*, 4 (1999), No. 6, pp. 77–87.

Author index