Workshop on

Symbolic-Numeric Algebra for Polynomials
SNAP 96
15-17th July 1996
INRIA Sophia-Antipolis, France

This meeting is intended as a timely workshop to discuss the emerging understanding of problems involving polynomials with inexactly-known coefficients. Such polynomial problems arise, for example, when physical measurements or numerical computations are used to specify a polynomial system. In this context, the usual exact algorithms of computer algebra are not applicable and new approaches must be devised. Such approaches are the fruit of the cross-fertilization of symbolic and numeric computing, a domain sometimes referred to as *seminumerical computation*.

Points of interest include:

- GCD computation for polynomials with inexactly-known coefficients.
- Symbolic-numeric methods for simplifying and solving multivariate polynomial systems (e.g. Gröbner bases).
- Matrix methods oriented towards zero-dimensional ideals (e.g. multivariate resultants).
- Possible extension for the higher-dimensional ideal case.
- Sound methods for other approximate polynomial problems (e.g. factorization, Padé approximations).
- Software systems enabling efficient implementation of these algorithms, taking full advantage of existing numerical libraries such as LAPACK and NAG.

The organizing committee includes:

R. Corless	rmc@pineapple.apmaths.uwo.ca
I.Z. Emiris	emiris[at]sophia[dot]inria[dot]fr
A. Galligo	galligo@hera.unice.fr
B. Mourrain	mourrain@sophia.inria.fr
S. Watt	smwatt@sophia.inria.fr

We would like to thank France Limouzis for her priceless contribution in the organization of the workshop as well as G. dos Reis, T. Giordano, F. Livigni and F. Tournoi for helping in preparing these proceedings.

This meeting is sponsored by SAFIR (a joint project between the University of Nice – Sophia-Antipolis, INRIA, and CNRS), in cooperation with ACM SIGSAM, the FRISCO consortium and GDR AMI.

Address: Projet SAFIR, INRIA, 2004 Route des Lucioles, B.P. 93, 06902 Sophia-Antipolis Cedex FRANCE. Fax: +33 – 93 65 78 58.

Ioannis Z. Emiris and Bernard Mourrain
Sophia-Antipolis, 3 July 1996

Atelier sur les liens

Symboliques-Numériques pour l'Algèbre des Polynômes
SNAP 96
15-17 Juillet 1996
INRIA Sophia Antipolis, France

Cet atelier a pour objet de faire le point sur les problèmes faisant intervenir des polynômes dont les coefficients ne sont connus que de manière approchée. Ce type de problèmes apparait par exemple quand des mesures physiques ou des calculs numériques sont utilisés pour définir les polynômes. Dans ce contexte, les méthodes algébriques habituelles ne sont plus applicables et de nouvelles approches sont nécessaires. Ces approches sont le fruit d'interactions entre le calcul formel et le calcul numérique, un domain aussi connu comme *calcul semi-numérique.*

Les thèmes abordés comprennent :

- Les calculs de PGCD pour des polynômes dont les coefficients sont connus approximativement.
- Les méthodes numériques/symboliques pour simplifier et résoudre les systèmes polynomiaux à plusieurs variables (e.g. bases de Gröbner).
- Les méthodes matricielles pour la résolution de systèmes polynomiaux de dimension 0 (e.g. résultants en plusieurs variables).
- Les extensions aux variétés de dimension plus grande.
- D'autres méthodes pour des polynômes à coefficients approchés (e.g. factorisation, approximation de Padé).
- Implémentation, composants logiciels, utilisation de bibliothèques numériques comme LAPACK et NAG.

Les organisateurs sont :

R. Corless	rmc@pineapple.apmaths.uwo.ca
I.Z. Emiris	emiris[at]sophia[dot]inria[dot]fr
A. Galligo	galligo@hera.unice.fr
B. Mourrain	mourrain@sophia.inria.fr
S. Watt	smwatt@sophia.inria.fr

Nous tenons aussi à remercier France Limouzis pour son assistance formidable pendant l'organisation de cet atelier, ainsi que G. dos Reis, T. Giordano, F. Livigni and F. Tournoi pour leur contribution pour la préparation de ces actes.

SNAP 96 est organisé par le Projet SAFIR (projet commun à l'INRIA, le CNRS et l'Université de Nice–Sophia Antipolis), avec la coopération de ACM SIGSAM, le consortium FRISCO et GDR AMI.

Adresse : Projet SAFIR, INRIA, 2004 Route des Lucioles, B.P. 93, 06902 Sophia-Antipolis Cedex FRANCE. Fax : +33– 93 65 78 58.

Ioannis Z. Emiris et Bernard Mourrain
Sophia-Antipolis, 3 juillet 1996

Contents

When are two numerical polynomials relatively prime?

Bernhard Beckermann
Laboratoire d'Analyse Numerique et d'Optimisation,
Universite des Sciences et Technologies de Lille,
59655 Villeneuve d'Ascq Cedex, France
e–mail: bbecker@ano.univ-lille1.fr
and
George Labahn
Department of Computing Science
University of Waterloo, Waterloo, Ontario, Canada
e–mail: glabahn@daisy.uwaterloo.ca

Abstract

Let a and b be two polynomials having numerical coefficients. We consider the question: when are a and b relatively prime? Since the coefficients of a and b are approximant the question is the same as: when are two polynomials relatively prime, even after small changes of the coefficients?

In this paper we provide a fast algorithm for determining that two polynomials are prime, even under small perturbations of the coefficients. Our methods rely on an inversion formula for Sylvester matrices to establish an effective criterion for relative primeness. The inversion formula also allows for an estimate of the condition number of the linear problem. A modification of the well known Cabay-Meleshko numerical algorithm for Padé approximation is then used to compute the criterion for a given problem in a fast, numerically stable way.

Numerical Factorization of Multivariate Polynomial : symbolic computation of a numerical iteration function.

Anne-Mercedes BELLIDO [1]
L.A.C.O. Université de Limoges [2]
e-mail: bellido@cict.fr

This talk is devoted to the numerical approximation of d linear factors of a homogeneous multivariate polynomial.
The motivation of our study is the theory of the U-resultant of algebraic systems ([13, 10, 1]), but here, to begin with, we deal only with the special case in which the polynomial is exactly the product of d linear and distinct factors, in $\mathbb{C}[t_1, \ldots, t_n]$.
Our method differs in its principle from the usual numerical methods of search of linear factors ([1, 7]) and can at first be regarded as a generalization to the multivariate case of Grau's method for the simultaneous numerical factorization of a univariate polynomial ([4, 9]).
The idea is to construct a "resolvent" system of equations, a solution of which will give a factorization of the polynomial, and to apply Newton's method to this system.
We use here a slightly different point of view to construct the resolvent system, more similar to the one already used in [6, 2, 3].
But in this case, the system is underdetermined, and the originality of this work is the computation of an explicit form of the Moore-Penrose inverse of the Jacobian matrix ([11, 12]) of the resolvent system, which allows us to finally obtain an iteration function which is in some sense a analogue to the Weierstrass' iteration function ([14], also called Durand-Kerner's method [5, 8]) for the simultaneous computation of all the roots of a univariate polynomial.

References

[1] Auzinger W., Stetter J., *An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations*, International Series of Numerical Mathematics , Birkhauser Verlag Basel, Vol. 86, No., (1988) pp 11-30.

[2] Bellido A.-M., *Doctoral dissertation*, Université Paul Sabatier, Toulouse, France, (1992).

[3] Bellido A.-M., *Construction of iteration functions for the simultaneous computation of the solutions of equations and algebraic systems.*, Numerical Algorithms, Vol.6, (1994) pp 317-351.

[4] Carstensen, C., *" On Grau's method for simultaneous factorization of polynomials,"* SIAM J. Numer. Analysis , Vol. 29, No. 2, (1992) pp 601-613.

[5] Durand E., *Solutions Numériques des Équations Algébriques*, tome 1, Masson,Paris, (1960).

[6] Frommer A., *A Unified Approach to Methods for the simultaneous Computation of All Zeros of Generalized Polynomials*, Numer.Math., Vol.54, (1988) pp.105-116.

[7] Hohl J.-C., *Massively parallel search for linear factors in polynomials with many variables*, preprint, (1995).

[8] Kerner I.O., *Ein Gesamtschrittverfahren zur Berechnung von Nullstellen von Polynomen*, Numer. Math., Vol. 8, No. 2, (1966) pp.290-294.

[1] Work partially supported by European Project POSSO
[2] Unité de Recherche Associée au CNRS 1586

[9] Kirrinnis P., *Polynomial factorization and partial fraction decomposition*, submitted to J. of Complexity (1993).

[10] Lazard D., *Résolution des systèmes d'équations algébriques*, Theor. Comp. Sci. , Vol. 15, (1981) pp 77-110.

[11] Ortega J.M. and Rheinboldt W.C., *Iterative solutions of nonlinear equations in several variables*, Academic Press, New York, (1970).

[12] Shub M., Smale S., *Complexity of Bezout's theorem III : condition number and packing,* Journal of Complexity 9 (1993) pp 4-14.

[13] Van Der Waerden., *Modern Algebra*, Vol. II, Hungar, New York, (1950).

[14] Weierstrass K., *Neuer Beweis des Fundamentalsatzes der Algebra.*, Mathematische Werke , tome III, Mayer u. Müller Berlin, (1903) pp 251-269.

Remarks on the implementation of algorithms for the numerical computation of polynomial roots.

Dario Bini and Valeria Burchielli,
Dipartimento di Matematica, Università di Pisa

Given $\epsilon \geq 0$, a polynomial $a(x) = \sum_{i=0}^{n} a_i x^i$, and the norm $||b(x)|| = \max_i |b_i|/|a_i|$, where $b(x) = \sum_{i=0}^{n} b_i x^i$ (formally we assume that $0/0 = 0$ and $|b_i|/0 = +\infty$ for $b_i \neq 0$), define the *ϵ-neighborhood* of $a(x)$ the set

$$\mathcal{N}_\epsilon(a) = \{b(x) = \sum_{i=0}^{n} b_i x^i : \quad ||b(x) - a(x)|| \leq \epsilon\}.$$

$\mathcal{N}_\epsilon(a)$ is constituted by all the polynomials whose coefficients are obtained by perturbing the corresponding coefficients of $a(x)$ by a relative error at most ϵ, i.e., any polynomial $b(x)$ such that $|b_i - a_i| \leq \epsilon |a_i|$, $i = 0, \ldots, n$, belongs to $\mathcal{N}_\epsilon(a)$. Observe that, if $a_i = 0$ for some i, then $b_i = 0$ for any $b(x) \in \mathcal{N}_\epsilon(a)$.

By following Mosier [3], define the *ϵ-root neighborhood* of $a(x)$ the set

$$\mathcal{RN}(\epsilon) = \{z \in \mathbf{C} : \quad \exists b(x) \in \mathcal{N}_\epsilon(a),\ b(z) = 0\}.$$

Observe that the set $\mathcal{RN}(\epsilon)$ is made up by the roots of all the polynomials whose i-th coefficient is obtained by perturbing the corresponding coefficient a_i of $a(x)$ by a relative error at most ϵ.

We introduce a function $TEST$ which, applied in floating point arithmetic with precision $\mu \leq \epsilon/2$, outputs the following informations:

$$\begin{array}{lcl} TEST(x) = TRUE & \Rightarrow & x \in \mathcal{RN}(6n\epsilon) \\ TEST(x) = FALSE & \Rightarrow & x \notin \mathcal{RN}(\frac{1}{2}\epsilon) \end{array}$$

Based on the concept of root-neighborhood, on the above test function and on the technique of [2], we present and discuss the implementation of a polynomial rootfinder which performs the following computations.

Input: a relative input error bound $\epsilon_{in} \geq 0$; a relative output error bound $\epsilon_{out} > 0$; the degree n and approximations $\tilde{a}_0, \ldots, \tilde{a}_n$ of the coefficients $a_0, \ldots, a_n \in \mathbf{C}$, of a univariate polynomial $a(x) = \sum_{i=0}^{n} a_i x^i$, such that $|\tilde{a}_i - a_i| \leq \epsilon_{in}|a_i|$, $i = 0, \ldots, n$, where the real and the imaginary parts of $\tilde{a}_i$ can be represented in floating point, integer or rational form; a set $\mathcal{S}$ ($\mathbf{C}$, disk, circle, halfplane, straight line); a `goal` chosen among `count, isolate, approximate`.

Output: according to the chosen `goal`, count, Newton-isolate or ϵ_{out}-approximate the roots of $p(x)$ in $\mathcal{S}$. Where, "count" means to count the number of roots which are in the interior part of $\mathcal{S}$ and the number of roots which are in the boundary of $\mathcal{S}$, for $\epsilon_{in} = 0$; otherwise, for $\epsilon_{in} \neq 0$, "count" means to count the minimum number of the roots n_i, n_o, which *any* polynomial having coefficients $\tilde{a}_i$ such that $|\tilde{a}_i - a_i| \leq \epsilon_{in}|a_i|$, has in the interior parts of $\mathcal{S}$ and of $\bar{\mathcal{S}}$, respectively, where $\bar{\mathcal{S}}$ denotes the complement of $\mathcal{S}$.

In the case $\epsilon_{in} = 0$, "Newton-isolate" means to compute $x_1, \ldots, x_n \in \mathbf{C}$, $r_1, \ldots, r_n \in \mathbf{R}$, such that each isolated disk $C_i(x_i, r_i) = \{z \in \mathbf{C} : \quad |z - x_i| \leq r_i\}$ contains a root of $a(x)$ and is either Newton-isolated, i.e., $|x_i - x_j| - r_i - r_j > 5nr_i$, for $j \neq i$, or its relative radius is less than ϵ_{out}, i.e., $r_i < \epsilon_{out}|x_i|$. Moreover, the connected components of $\bigcup C(x_i, r_i)$ made up by k disks contain k roots. The multiplicity of the roots can be automatically detected by means of Mahler's inequality. If the disk $C(x_i, r_i)$ is Newton isolated than Newton's iteration applied to $a(x)$ and starting with x_i converges quadratically right from the start to the roots in $C(x_i, r_i)$ ([4]). In the case where $\epsilon_{in} \neq 0$, we may have disks $C_i(x_i, r_i)$ which do not satisfy the above conditions, but in this case we have $x_i \in \mathcal{RN}(6n\epsilon_{in})$.

The "ϵ_{out}-approximation" means that for each disk $C_i(x_i, r_i)$, either $r_i < \epsilon_{out}|x_i|$ or $x_i \in \mathcal{RN}(6n\epsilon_{in})$ and the disk contains a root of $a(x)$. Moreover, the connected components of $\bigcup C(x_i, r_i)$ made up by k disks contain k roots.

The program can be used for providing upper bounds of the approximate GCD of two polynomials, or for computing the exact GCD.

The algorithm has been implemented in Fortran90 by using the multiprecision package of [1].

References

[1] D.H. Bailey, A Portable High Performance Multiprecision Package, RNR Technical Report RNR-90-022, NASA Applied Research Branch, NASA Ames Research Center, Moffett Fields, CA, 1992.

[2] D. Bini, Numerical computation of polynomial zeros by means of Aberth's method, *Numerical Algorithms*, to appear.

[3] R.G. Mosier, Root Neighborhoods of a Polynomial, *Math. Comp.*, **47**, 265–273 (1986).

[4] P. Tilli, Convergence conditions of some methods for the simultaneous computation of polynomial zeros, *Calcolo* 1996, to appear.

Optimization Algorithms for the Floating-Point GCD

P. Chin and R. M. Corless
Department of Applied Mathematics,
University of Western Ontario
London, Ontario, Canada N6A 5B7

Abstract

We describe algorithms to compute the GCD of polynomials with inexact coefficients. We formulate the problem in two different ways, both as optimization problems. With the first approach, the algorithm has the advantage of being extremely simple to implement. The second strategy is more complicated, but may also be more efficient.

Introduction

In this paper, we consider the problem of computing the GCD of two or more polynomials with inexact floating-point coefficients. Examples of previous work in this area include [1], [2], and [3]. Our approach to this problem involves representing the polynomials as vectors of coefficients and formulating it as an optimization problem that can be solved with standard numerical techniques. For simplicity, we show only the two-polynomial case; extension to three or more polynomials is straightforward.

The first algorithm, cofactor iteration, involves successive minimization over subsets of the variables. In the second approach, we transform the problem to a minimization over a reduced set of variables and apply a well-known algorithm for unconstrained nonlinear optimization. A related paper that also uses an optimization formulation is [4].

Suppose we wish to find the GCD of two polynomials, $p(x)$ of degree n, and $q(x)$ of degree m. Without loss of generality we can assume they are monic. Because the coefficients of $p(x)$ and $q(x)$ are given inexactly, we actually wish to find $d(x) = d_0 + d_1x + \cdots + d_{k-1}x^{k-1} + x_k$ such that $d(x)$ divides $p(x) + \Delta p(x)$ and $q(x) + \Delta q(x)$ exactly, for some polynomials $\Delta p(x)$ of degree $n-1$ and $\Delta q(x)$ of degree $m-1$. In other words, we want

$$\begin{aligned} d(x)u(x) &= p(x) + \Delta p(x), & (1)\\ d(x)v(x) &= q(x) + \Delta q(x) & (2) \end{aligned}$$

for cofactor polynomials $u(x)$ of degree $n-k+1$ and $v(x)$ of degree $m-k+1$.

We assume that the degree k of the divisor is known. In the paper [2], a method is given using singular-value decomposition which ensures that k is a good choice and is essentially an upper bound for the degree of the GCD.

We would like to make the coefficients of $\Delta p(x)$ and $\Delta q(x)$ as small as possible. We work in the 2-norm because this allows us to take advantage of the algorithms described below. Let $\mathbf{d}$, $\mathbf{u}$, $\mathbf{v}$, $\Delta\mathbf{p}$ and $\Delta\mathbf{q}$ denote the vectors of coefficients of $d(x)$, $u(x)$, $v(x)$, $\Delta p(x)$ and $\Delta q(x)$ respectively (*e.g.* $\mathbf{d} = [d_0, d_1, \ldots, d_{k-1}]$). We can state the problem as the minimization problem

$$\min \quad f(\mathbf{d},\mathbf{u},\mathbf{v}) = \|\Delta\mathbf{p}\|^2 + \|\Delta\mathbf{q}\|^2. \qquad (3)$$

We further assume we have an initial guess available for $\mathbf{d}$. Finding such a guess is not a trivial task. One possibility is to use a stabilized Euclidean algorithm as discussed in, for example, [3].

Cofactor Iteration

We must first express $\|\Delta\mathbf{p}\|^2 + \|\Delta\mathbf{q}\|^2$ explicitly in terms of the vectors, $\mathbf{d}$, $\mathbf{u}$ and $\mathbf{v}$. By expanding the polynomials in (1) and (2) and matching coefficients, it can be shown that

$$\Delta\mathbf{p} = \mathbf{C}_1(\mathbf{d})\mathbf{u} - \mathbf{p} \qquad (4)$$

where $\mathbf{C}_1(\mathbf{d})$ is an $(n+1)\times(n-k+1)$ *Cauchy matrix*,

$$\begin{bmatrix} d_0 & & & \\ d_1 & d_0 & & \\ \vdots & d_1 & \ddots & \\ d_{k-1} & \vdots & \ddots & d_0 \\ 1 & d_{k-1} & & d_1 \\ & 1 & \ddots & \vdots \\ & & \ddots & d_{k-1} \\ & & & 1 \end{bmatrix}.$$

Further details regarding this procedure are given in [6]. Similarly, $\Delta\mathbf{q}$ can be expressed as

$$\Delta\mathbf{q} = \mathbf{C}_2(\mathbf{d})\mathbf{v} - \mathbf{q} \qquad (5)$$

where $\mathbf{C}_2(\mathbf{d})$ is an $(m+1)\times(m-k+1)$ Cauchy matrix.

Alternate expressions for $\Delta\mathbf{p}$ and $\Delta\mathbf{q}$ are

$$\begin{aligned} \Delta\mathbf{p} &= \mathbf{C}_u(\mathbf{u})\mathbf{d} - \mathbf{p}, & (6)\\ \Delta\mathbf{q} &= \mathbf{C}_v(\mathbf{v})\mathbf{d} - \mathbf{q} & (7) \end{aligned}$$

where $\mathbf{C}_u(\mathbf{u})$ is an $(n+1)\times k$ Cauchy matrix containing components of $\mathbf{u}$ instead of components of $\mathbf{d}$, and $\mathbf{C}_v(\mathbf{v})$ is an $(m+1)\times k$ Cauchy matrix containing components of $\mathbf{v}$. The fact that $\Delta\mathbf{p}$ and $\Delta\mathbf{q}$ can be expressed in these two different ways is what allows us to use the first algorithm, cofactor iteration, which was introduced in [6].

In cofactor iteration, we successively minimize first with respect to $\mathbf{u}$ and $\mathbf{v}$, and then with respect to $\mathbf{d}$. The algorithm is as follows: given an initial guess for $\mathbf{d}$, repeat the following two steps until convergence to a (local) minimum is achieved.

1. Fix $\mathbf{d}$ and form $\mathbf{C}_1$ and $\mathbf{C}_2$. Solve the following least squares problems for $\mathbf{u}$ and $\mathbf{v}$:

$$\begin{aligned} \mathbf{u} &= \text{argmin } \|\mathbf{C}_1\mathbf{u} - \mathbf{p}\|^2 & (8)\\ \mathbf{v} &= \text{argmin } \|\mathbf{C}_2\mathbf{v} - \mathbf{q}\|^2 & (9) \end{aligned}$$

2. Fix $\mathbf{u}$ and $\mathbf{v}$ and form $\mathbf{C}_u$ and $\mathbf{C}_v$. Solve the least squares problem

$$\begin{aligned} \mathbf{d} &= \text{argmin } \{\|\mathbf{C}_u\mathbf{d} - \mathbf{p}\|^2 + \|\mathbf{C}_v\mathbf{d} - \mathbf{q}\|^2\} \\ &= \text{argmin } \left\| \begin{bmatrix} \mathbf{C}_u \\ \mathbf{C}_v \end{bmatrix} \mathbf{d} - \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} \right\|^2 . \end{aligned} \tag{10}$$

Note that, in the first step, $\Delta\mathbf{p}$ and $\Delta\mathbf{q}$ are replaced by the expressions in (4) and (5), while in the second step, they are replaced by the ones in (6) and (7). The term "argmin" above refers to the value of the variable which achieves the function minimum.

The Reduced Problem

The cofactor iteration algorithm can be slow in converging, as it minimizes $\mathbf{d}$, $\mathbf{u}$ and $\mathbf{v}$ independently of one another. An alternate approach to cofactor iteration is to reduce the minimization problem,

$$\min \; f(\mathbf{d},\mathbf{u},\mathbf{v}) = \|\mathbf{C}_1(\mathbf{d})\mathbf{u} - \mathbf{p}\|^2 + \|\mathbf{C}_2(\mathbf{d})\mathbf{v} - \mathbf{q}\|^2, \tag{11}$$

to one involving only $\mathbf{d}$.

Suppose we let $\mathbf{u}$ and $\mathbf{v}$ be the least squares solutions of (8) and (9), without assuming that $\mathbf{d}$ is fixed. Then $\mathbf{u}$ and $\mathbf{v}$ are functions of $\mathbf{d}$:

$$\begin{aligned} \mathbf{u}(\mathbf{d}) &= (\mathbf{C}_1^t(\mathbf{d})\mathbf{C}_1(\mathbf{d}))^{-1}\mathbf{C}_1^t(\mathbf{d})\mathbf{p}, & (12)\\ \mathbf{v}(\mathbf{d}) &= (\mathbf{C}_2^t(\mathbf{d})\mathbf{C}_2(\mathbf{d}))^{-1}\mathbf{C}_2^t(\mathbf{d})\mathbf{q}. & (13) \end{aligned}$$

We can write (11) as a minimization problem in $\mathbf{d}$:

$$\min \; \hat{f}(\mathbf{d}) = \|\mathbf{C}_1(\mathbf{d})\mathbf{u}(\mathbf{d}) - \mathbf{p}\|^2 + \|\mathbf{C}_2(\mathbf{d})\mathbf{v}(\mathbf{d}) - \mathbf{q}\|^2, \tag{14}$$

where $\mathbf{u}(\mathbf{d})$ and $\mathbf{v}(\mathbf{d})$ are given by (12) and (13).

To solve this minimization problem, we use the BFGS algorithm [5] for unconstrained optimization. This algorithm is a quasi-Newton method and requires the gradient of $\hat{f}$. In our implementation, we provide an analytic gradient, computed from equations (12), (13) and (14).

Summary

Though only limited testing has been done so far, we have found that cofactor iteration generally requires many iterations but each one is inexpensive to compute. The BFGS algorithm for the reduced problem usually needs fewer iterations but each one requires several costly function and gradient evaluations. Both algorithms suffer from the problem that they go to local minima unless the initial point is close enough to the desired global minimum. In a future paper, more extensive analysis of the computational requirements and further experience with test cases will be reported. Alternative formulations of the optimization problem will also be explored.

References

[1] D. Bini & V. Pan, *Polynomial and Matrix Computations* , Birkhauser, 1994.

[2] R. M. Corless, P. M. Gianni, B. M. Trager, & S. M. Watt, "The Singular Value Decomposition for Polynomial Systems", *Proceedings of ISSAC '95* , Montreal, A. H. M. Levelt, ed., pp. 195–207.

[3] V. Hribernig & H. J. Stetter, "Detection and Validation of Clusters of Polynomial Zeros", *J. Symb. Comp.* 1995.

[4] N. Karmarkar & Lakshman Y. N. "Approximate Polynomial Greatest Common Divisors and Nearest Singular Polynomials", to appear in *Proceedings of ISSAC '96* .

[5] J. E. Dennis & R. E. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall Inc., 1983.

[6] R. M. Corless, "Editor's Corner: Cofactor Iteration," *SIGSAM Bulletin* , Vol. 30, No. 1, pp. 34-38, 1996.

CONDITION NUMBER ANALYSIS
FOR SPARSE POLYNOMIAL SYSTEMS

Jean-Pierre DEDIEU
Laboratoire Approximation et Optimisation
Université Paul Sabatier
31062 Toulouse Cedex France
e-mail : dedieu@cict.fr

Abstract. We define the condition number for sparse polynomial systems and we study the complexity of a continuation method for polynomial system solving in terms of this invariant.

NORT Package for Treating Multivariate Power Series and its Applications

Victor F. Edneral
Institute for Nuclear Physics of Moscow State University
edneral@theory.npi.msu.su

Abstract

In the report discussed applications of the NORT package for problems of creation multivariate power series which approximate periodic solutions of nonlinear autonomous ODEs (Edneral,1993), trajectories of periodic orbits of such systems, Lyapunov's values for an estimation of a number of small limit circles of a cubic planar system and for solving of systems of equation over formal power series (Edneral,1995), (Vasiliev,Edneral,1995).

Originally the NORT package was created for building Poincare – Dulac normal forms (Edneral, Khrustalev, 1985, 1992) till fixed order. It is written in Standard LISP and contains procedures for treating truncated multivariate power series of an arbitrary dimension. Besides of procedures for arithmetic operations with such series there are special procedures for the creation of normal forms and procedures for substitutions, calculations of roots, for differentiating, for printing and for inverting multivariate power series. The package can be used as a separate unit under a Standard Lisp environment or as a REDUCE package.

Complex numerical coefficients of the truncated power series may be evaluated in three different arithmetics: precision rational, floating point and approximate rational (Rodionov A.Ya, in progress), in according to the user's choice. Separately user may choose a representation of numbers (in a floating point or in a rational form) for output. This output has a REDUCE acceptable form, but REDUCE's internal procedures are not used by the NORT.

The program uses a recurrent polynomial representation separated in each order of the stored series. There is a compact enough use of memory. So for 8 - dimensional problem we managed to treat pieces of series with 150000 elements on computer with 64 Mbyte of RAM.

References

Edneral, V.F., Khrustalev, O.A. (1985) *The Normalizing Transformation for Nonlinear Systems of ODE. The Realization of the Algorithm*; Proceedings of the International Conference on Computer Algebra and its Application in Theoretical Physics (USSR, Dubna, September 1985). Dubna: JINR publ., 219–224. In Russian.

Edneral, V.F., Khrustalev, O.A. (1992) *Program for Recasting ODE Systems in the Normal Form*; "Programmirovanie" (Sov. J. of Programming), # 5, 73–80. In Russian.

Edneral, V.F. (1993) *Computer Generation of Normalizing Transformation for Systems of Nonlinear ODE*; Proceedings of the 1993 International Symposium on Symbolic and Algebraic Computation (Kiev, Ukraine, July 6-8, 1993). N.Y., ACM Press, edited by M.Bronstein, pp. 14– 19.

Vasiliev N.N., Edneral V.F., (1995) *About Constructing Canonical Basis for the Ring of Formal Power Series*; The Abstracts of the conference "Computer Methods of Celestial Mechanics (S.-Petersburg, Russia, October 17–20, 1995)", pp. 72–73. In Russian.

Edneral V.F., (1995) *To the Question of Multidimensional Power Series Solving*; The Abstracts of the PoSSo-95 "PoSSo Open Workshop on Applications of PoSSo and Real Solving, (Iraklio, Greece, June 7–10, 1995)", p. 34.

Certified Approximate Univariate GCDs

Ioannis Z. Emiris
Projet S.A.F.I.R., I.N.R.I.A., B.P. 93, Sophia-Antipolis 06902, France.
emiris(at(sophia(dot(inria(dot(fr

André Galligo
Laboratoire de Mathématiques, Université de Nice–Sophia Antipolis, Parc Valrose, Nice 06108, Cedex 2. and Projet S.A.F.I.R., I.N.R.I.A., B.P. 93, Sophia-Antipolis 06902, France.
galligo@math.unice.fr

Henri Lombardi
Laboratoire de Mathématiques, Université de Franche–Comté, Besançon 25030, France.
lombardi@vega.univ-fcomte.fr

Extended Abstract

We study the approximate GCD of two univariate polynomials given with limited accuracy or, equivalently, the exact GCD of the perturbed polynomials within some prescribed tolerance. A perturbed polynomial is regarded as a family of polynomials in a classification space.

The question of computing the approximate Greatest Common Divisor (GCD) of a polynomial pair is being studied with renewed interest, as illustrated by the variety of different approaches to the problem within the last couple of years. Euclid's algorithm has been applied in different ways, but yields only a lower bound on the degree of the approximate GCD, as illustrated in [EGL95].

[CGTW95] applies a Singular Value Decomposition (SVD) on the Sylvester matrix, which is a powerful and well-studied numerical operation. That paper used an a posteriori bound to estimate the perturbation of a polynomial pair or system but, still, the output does not necessarily *maximize* the approximate GCD degree. We provide a counterexample that illustrates the limitation of considering only the Sylvester matrix singular values, even when the extended euclidean algorithm is also used.

The problem is formalized as follows. Consider the standard euclidean L_2-norm for polynomials. Given $f, g \in \mathbb{C}[x]$ of degree, respectively, $n \geq m$ and tolerance $\epsilon \in (0, 1]$, the *degree of the ϵ-GCD* is defined to be the *maximum* integer r such that there exist $\widehat{f}, \widehat{g} \in \mathbb{C}[x]$ of degree bounded by n and m, respectively, with $|f - \widehat{f}|, |g - \widehat{g}| \leq \epsilon$ and $\deg(\gcd(\widehat{f}, \widehat{g})) = r$. The latter polynomial is an ϵ-GCD of f, g.

We consider the singular values of all subresultant mappings. The subresultant mapping $\mathrm{Sy}_r(f, g)$ is defined by

$$\mathrm{Sy}_r(f, g) : (u, v) \quad \mapsto \quad uf + vg, \qquad \deg u \leq m - r - 1, \deg v \leq n - r - 1.$$

$\mathrm{Sy}_r(f, g)$ has full rank ($= m + n - 2r$) if and only if $\deg(\gcd(f, g)) \leq r$. Based on such properties, SVD computations yield upper bounds on the degree of the approximate GCD. Let us denote by τ_r the last singular value of $\mathrm{Sy}_r(f, g)$. It is easy to prove that $\tau_k \geq \tau_{k-1}, \forall k > 0$.

The main contribution of [EGL96] is the following gap theorem. Suppose that $\tau_{r-1} \leq \epsilon < \tau_r$ and $1 \leq r \leq m \leq n$. If, moreover,

$$G = \tau_r - \tau_{r-1} 2^{2n+m-2r} > 0 \qquad \text{and} \qquad \left(1 + \frac{2 + \tau_r^2}{\tau_r - \tau_{r-1}}\right)^{n+1} \left(1 + \frac{2^{2n+m-2r}}{G}\right) \tau_{r-1} \leq \epsilon,$$

then the degree of the ϵ-GCD(f, g) is guaranteed to be equal to r. A corollary to the main theorem, under the additional hypothesis that $\tau_{r-1} \leq \epsilon < \tau_r \leq 23/25$, states that

$$\tau_{r-1} \leq 2^{-5n-3} \tau_r^{n+2} \epsilon \Rightarrow \deg \epsilon\text{-}\gcd(f, g) = r.$$

Our approach leads to the following algorithm of polynomial complexity in the degrees of the input and output polynomials.

1. If $\tau_0 > \epsilon$ then return 1. Initialize r to 1. While $\tau_r \leq \epsilon$, increment r. If the hypotheses of the main theorem for the candidate degree r are not satisfied then a variant of Euclid's algorithm or optimization techniques may be used.

2. Apply an SVD to write $\mathrm{Sy}_{r-1} = A\Sigma B$, where A, B are orthonormal and Σ diagonal. Read off pair (u, v) from the last row of B; this pair defines an *approximate syzygy* $T = uf - vg$. Obtain polynomial T, such that $|T| = \tau_{r-1}$, by multiplying the polynomial in the last column of A by τ_{r-1}.

3. Compute Q and R by polynomial division: $T = uQ + R$.

4. Polynomials u, v are approximately relatively prime. Compute s and t, such that $R = us - vt$, by solving for the vector whose image under the Sylvester matrix transformation $\mathrm{Sy}_0(u, v)$ is the vector expressing R.

5. Output perturbed polynomials $\widehat{f} = f - Q - s$ and $\widehat{g} = g - t$ which lie within the prescribed tolerance and possess an exact GCD of degree r. Return ϵ-$\gcd(f, g) = (g - t)/u$.

Lastly, we suggest the use of weighted norms in order to sharpen the gap in a more intrinsic context. The weighted L_2-norm, is defined as follows.

$$P = \sum_{i=0}^{d} p_i x^{d-i} \in \mathbb{C}[x] \Rightarrow \langle P \rangle = \left(|p_0|^2 / \binom{d}{0} + \cdots + |p_i|^2 / \binom{d}{i} + \cdots + |p_d|^2 / \binom{d}{d} \right)^{1/2}.$$

The first merit of this norm is that it remains invariant, for *bivariate homogeneous* polynomials, under unitary changes of variables. The second advantage is that the norm of the product is tightly bound above and below by the product of the norms.

In a *weighted monomial basis* the subresultant mapping matrix $\mathrm{Sw}_r(f, g)$ is obtained from $\mathrm{Sy}_r(f, g)$ by: first, multiplying column j corresponding to f, for $1 \leq j \leq m - r$, by $\sqrt{\binom{m-r-1}{j-1}}$, second, multiplying column $m-r+j$ corresponding to g, for $1 \leq j \leq n-r$, by $\sqrt{\binom{n-r-1}{j-1}}$, lastly dividing row i, $1 \leq i \leq m+n-r$, by $\sqrt{\binom{m+n-r-1}{i-1}}$. For the sequence of minimum singular values τ_i of $\mathrm{Sw}_i(f, g)$, $\mathrm{Sw}_r(f, g)$ has full rank $(= m + n - 2r)$ if and only if $\deg(\gcd(f, g)) \leq r$.

References

[CGTW95] R.M. Corless, P.M. Gianni, B.M. Trager, and S.M. Watt. The singular value decomposition for polynomial systems. In *Proc. ACM ISSAC*, pages 195–207, 1995.

[EGL95] I.Z. Emiris, A. Galligo, and H. Lombardi. Numerical univariate polynomial GCD. In J. Renegar, M. Shub, and S. Smale, editors, *Proc. AMS-SIAM Summer Seminar on Math. of Numerical Analysis*, Lectures in Applied Math., Park City, Utah, July 1995. To appear in 1996.

[EGL96] I.Z. Emiris, A. Galligo, and H. Lombardi. Certified approximate univariate GCDs. In *Proc. 4th MEGA*, June 1996.

Absolute Factorization of Bivariate Polynomials with Floating Point Coefficients

André Galligo and Stephen M. Watt
University of Nice-Sophia Antipolis

Given a polynomial $p(x, y)$ of degree d and complex floating point coefficients, we seek two polynomials f_1 and f_2 of degrees d_1 and d_2 such that $d_1 + d_2 = d$ and $f_1 \cdot f_2 = p + \Delta p$ for small Δp.

We view $p(x, y) = 0$ as defining $y(x)$, and develop y as a series in x about a generic point. Candidate factors are formed using truncations of $y(x)$ in bivariate polynomials of increasing degree.

The candidate factors are tested by approxiamate division with p, and the algorithm terminates when a pair f_1, f_2 is found which satisfy a given tolerance for Δp, or when the degree of the candidate factor exceeds $d/2$. The approximate division step requires the approximate solution of a linear system, for which we use the SVD to determine the numeric rank.

This a dense method, in the sense that Δp may introduce terms which are not in the support of p. To simplify treatment of leading terms we replace x, y with a randomized linear combination at the outset. We believe this first step is not essential, and could be replaced by a careful combinatoric reasoning.

This method can be applied to multivariate polynomials by considering generic bivariate linear restrictions of the given polynomial.

Computations with approximate ideals

P. Gianni
Dipartimento di Matematica, Università, via Buonarroti 2, I–56127 Pisa.
and
B.Trager
IBM Research, Yorktown Heights, NY

For zero dimensional ideals with no solutions at "infinity", the representation degree bound is linear in the degrees of the ideal generators. This allows us to use linear algebra to compute a basis for the vector space of all polynomials of any given degree in the ideal. This approach can be used even when the generators have coefficients which are "approximately known". Using this constuction we obtain algorithms for the computation of adjoints, multivariate gcd's, and multivariate factorization for approximately given polynomials.

Practical univariate complex root-finding by Schönhage's algorithm

Xavier Gourdon

In 1982, Schönhage [2] developped an asymptotically fast algorithm to compute the approximate factorization of univariate complex polynomials. In the context of implementations, his approach is interesting for two reasons. First, it relies on mathematical bounds used at every step of the algorithm, which gives the precision you need to perform the calculation. This feature makes it one of the first algorithms which really ensures robustness and guaranteed results. Secondly, this algorithm has been designed in order to be implemented, thus it is intended to be not only theorically, but also practically fast. Implementations have been written in PARI and MAGMA and give good results regarding timings and robustness [1]. The degree 1000 can be reached with totally guaranteed results at any desired precision on all polynomials, even very ill conditionned ones.

The algorithm recursively factors the polynomial into two polynomials and relies on the *splitting circle method*. It consists in two main steps. First, it finds a *splitting circle*, which is a circle containing a non trivial subset of the roots of the input polynomial P. Then it uses residue computations from this circle to compute an approximation of the factor whose roots are the roots of P inside the circle.

Finding the splitting circle

This is actually the key problem of the whole method. In practice, it requires about half of the total time. It essentially consists in computing the modulus of the roots. This is achieved thanks to the classical Graeffe process of *root squaring*. We use a variant especially suited to our context. Contrary to the common application of Graeffe's method, here the primary interest does not lie in circles on which the zeros are located, but in circles bounded away from all the zeroes, and for the latter purpose moderate precision is quite sufficient. Moreover, one should not try to compute all the root moduli simultaneously, which indeed may require prohibitive computations. Instead, one can shift the focus of precision by suitable scalings after each root squaring step such that the algorithm is directed to the computation of a particular modulus.

Approximate factorization from the splitting circle

We denote by $u_1, \ldots, u_n$ the roots of the input polynomial P numbered such that the roots of P inside the splitting circle $\mathcal{C}$ are $u_1, \ldots, u_k$ with $1 \leq k < n$. To find the factor $F = \prod_{i=1}^{k}(z - u_i)$ of P, we first compute a rough approximation F_0 of F by residue computation. The formula

$$s_p = u_1^p + \cdots + u_k^p = \frac{1}{2i\pi} \oint_{\mathcal{C}} \frac{P'(z)}{P(z)}\, z^p \, dz$$

enables us, by numerical integration, to approximate the Newton sums $s_1, \ldots, s_k$ of F and then to obtain an approximation F_0 of F. A better approximation of F is then obtained from F_0 by using the *Newton-Schönhage method*, which converges quadratically.

Approximation factorization *versus* approximate root-finding

It is a well known fact that clustered zeros are less stable under small perturbations of the coefficients than isolated zeros. Therefore, it seems to be more appropriate to understand any prescribed accuracy in the sense of backward analysis: a collection of approximate zeros should be considered acceptable if their elementary symmetric functions agree well enough with the coefficients of the given polynomial. In this way, the algorithm achieves the following task : given $\epsilon > 0$, compute approximate linear factors $L_1, \ldots, L_n$ of a polynomial P such that

$$|P - L_1 \cdots L_n| < \epsilon |P|,$$

where $| \cdot |$ stands for the ℓ^1-norm of the coefficient vectors.

From this approximate factorization, an algorithm relying on perturbation results is then used to return approximations of the roots of the input polynomial P, together with a good absolute error bound on each root.

References

[1] Xavier Gourdon. *Combinatoire, Algorithmique et Géométrie des Polynômes.* Thèse, École polytechnique, 1996.

[2] Arnold Schönhage. The fundamental theorem of algebra in terms of computational complexity. Technical report, Mathematisches Institut der Universität Tübingen, 1982. Preliminary report.

Symbolic-numeric methods for solving satellite equilibrium equations

Sergey A.Gutnik
Institute for Computer Aided Design
Russian Academy of Sciences
19/18 2-nd Brestskaya str. Moscow, 123056 Russia
sg@server.ru

An approach for the symbolic-numeric analysis of the satellites dynamical equations is presented. Stationary solutions of these equations are defined by the multivariate polynomial system.

The algebraic polynomial system has been investigated with the help of both the numerical and symbolic analysis. The symbolic investigation was made by means of Resultant and Grobner Basis methods [1].

On the base of this methods the problem of defining the equilibrium positions of a satellite in a circular orbit under the influence of gravitational, aerodynamical, gyrostatic and static torques was solved [2-4].

The stationary motions of a satellite subject to gravitational, aerodynamical, gyrostatic and static torques is governed by nine algebraic equations with nine parameters - projections of torques vectors onto the frame attached to the body of the satellite. The classes of stationary solutions of these algebraic equations have been found with the help of computer algebra system Maple [5] by applying the Resultant, the Groebner Basis and Factorization methods.

The equilibrium positions and their stability are analyzed numerically.

References

[1.] Buchberger B., Theoretical Basis for the Reduction of Polynomials to Canonical Forms. SIGSAM Bulletin (1976), pp. 19–29.

[2.] Sarychev V. A. and Gutnik S. A., Equilibrium positions of a satellite-gyrostat,*Proc. of XXXV-th International Astronautical Congress, IAF*, Lausanne, 1984, p. 356.

[3.] Gutnik S. A.,Application of Computer Algebra to Investigation of the Relative Equilibria of a Satellite,*Proceedings of International Symposium on Symbolic and Algebraic Computation* Kiev (1993), ISSAC'93, pp. 63–64. ACM Press.

[4.] Sarychev V. A. and Gutnik S. A., Equilibria of a Satellite Under the Influence of Gravitational and Static Torques. COSMIC RESEARCH (KOSMICHESKIE ISSLIDOVANIYA), Vol. 32, Nos. 4-5, pp. 386–391, (1995), Plenum Publishing Corporation.

[5.] Char B. W., Geddes K. O., Gonnet G. H., Monagan M. B., Watt S. M., Maple Reference Manual. Watcom Publications Limited, Waterloo, Canada, 1988.

The Kharitonov theorem and its applications in symbolic mathematical computation

Markus A. Hitz and Erich Kaltofen

Abstract

We are looking at families of univariate polynomials over the rational or complex numbers, whose coefficients are defined by a continuous function or by constraints. Given a simply connected domain $\mathcal{D} \subset \mathbb{C}$, are there efficient algorithms to decide whether all the roots of *all* members of the family are inside $\mathcal{D}$?

In 1978, V. L. Kharitonov showed that for interval polynomials with real coefficients, it is necessary and sufficient to test *four* special members of the family to decide whether all polynomials are Hurwitz (i.e., $\mathcal{D}$ is the left half-plane). He extended his result for complex coefficients in a follow-up paper; eight test polynomials are required in this case. Despite of being published in a Western journal, the result remained widely unknown until 1983, when it was introduced to the control theory community by Barmish and Bialas. Especially in the late eighties it sparked increased activity in the area of robust control. To our knowledge, the significance of Kharitonov's theorem to computer algebra has not been recognized so far.

The original proof is based on the Hermite-Biehler theorem. Following Minnicelli et al., we will show a proof using geometric arguments, which outlines a general procedure for generating test sets in other norms and for different domains. One example is the result by Tempo for a polynomial family given by constraints in a weighted 1-norm. The domain is the unit circle, the test set consists of four complex polynomials. Finally, we will give a generalization for circles around the origin with arbitrary radius and sectors from such circles.

There is an interesting interdependency between root domain, norm, and test set, which is not explored yet. Some general conditions for the existence of Kharitonov-like test sets for complex domains have been published. However, most of them require exponentially many members, and little is known on constructing *finite* sets. For the known domains with test sets of sub-exponential cardinality we immediately have a polynomial time decision procedure for root clustering of a whole family of polynomials by (exactly) computing Cauchy indices via Sturm sequences.

References

[1] V. L. Kharitonov, *Asymptotic stability of an equilibrium position of a family of systems of linear differential equations*, Differensial'nye Uravnenya, vol. 14, no. 11, pp. 2086–2088, 1978, and Differential Equations, vol. 14, pp. 1483–1485, 1979.

[2] B. R. Barmish, *Invariance of the strict Hurwitz property for polynomials with perturbed coefficients*, Proc. IEEE Conf. Dec. and Control, 353–355, 1983.

[3] S. Bialas, *A necessary and sufficient condition for the stability of interval matrices*, Int. J. Control, vol. 37, no. 4, pp. 717–722, 1983.

[4] R. J. Minnicelli, J. J. Anagnost, and C. A. Desoer, *An elementary proof of Kharitonov's stability theorem with extensions*, IEEE Trans. Auto. Control, vol. 34, pp. 995–998, 1989.

[5] R. Tempo, *A simple test for Schur stability of a diamond of complex polynomials*, Proc. IEEE Conf. Dec. and Control, pp. 1892–1895, 1989.

[6] A. C. Bartlett, C. V. Hollot, and Huang Lin, *Root locations of an entire polytope of polynomials: it suffices to check the edges*, Math. Control Signals Systems, vol. 1, pp. 61–71, 1988.

[7] Y. K. Foo and Y. C. Soh, *Root clustering of interval polynomials in the left-sector*, Systems & Control Letters, vol. 13, pp. 239–245, 1989.

[8] E. Kaltofen, *Computing the irreducible real factors and components of an algebraic curve*, AAECC, vol. 1, pp. 135–148, 1990.

[9] F. R. Gantmacher, *Matrix Theory*, vol. 2, Chelsea Publishing Company, New York, N.Y., 1959.

Approximate Polynomial Constraint Solving

Hoon Hong
Research Institute for Symbolic Computation
Johannes Kepler University
A-4040 Linz, Austria
e-mail: hhong@risc.uni-linz.ac.at

Approximate Polynomial Constraints are formulas made of

- multivariate polynomials with *interval* coefficients,
- relational operators such as $=$, $>$,
- logical connectives such as $\vee$, $\wedge$, $\Leftarrow$,
- quantifiers such as $\forall$, $\exists$.

where the variables range over real numbers. In this talk, we present the followings topics:

1. Clarification of the notion of "Solving".

   The notion of solving is tricky due to the "approximateness" of coefficients. We discuss several definitions and choose one out of them.

2. Some ideas/methods for solving approximate constraints.

   Utilize theories/methods from interval math, computer algebra, numerical analysis, parallel computation.

3. Our on-going software development project: **STURM**.

   STURM is a system for solving approximate polynomial constraints.

Représentation de Courbes et de Surfaces Implicites par des méthodes d'exclusion

Olivier Isson

Le But est de représenter la variété réelle définie par une équation du style $P(X) = 0$ avec P un polynôme et X $\in C^n$. Maple qui effectue ce genre de représentation produit des courbes ou des surfaces parfois fausses ou imcomplètes et rencontre des difficultés aux voisinages des points isolés ou singuliers.

Après avoir introduit la notion de fonction d'exclusion qui a pour but d'approcher la distance d'un point à une surface, la méthode de Graeffe à n variables sera rapidement exposée.

La notion d'épaisseur des variétés représentées sera introduite et permettra de calculer la complexité de l'algorithme d'exclusion en dimension n.

Enfin se pose le problème de la représentation des courbes et surfaces obtenues. En effet, l'algorithme d'exclusion donne lieu à une liste d'hypercubes qu'il faut représenter de telle façon à obtenir des courbes fines et correctement dessinées (en 2D) et à faire apparaitre tous les reliefs des surfaces (en 3D). Pour ceci, différentes méthodes seront employées et les résultats ainsi obtenus seront exposés.

On Approximate Polynomial Greatest Common Divisors

N. K. Karmarkar
Mathematical Sciences Research Center
Bell Laboratories, Murray Hill, NJ 07974, USA
Lakshman Y. N.
Dept. of Mathematics and Computer Science
Drexel University, Philadelphia, PA 19104

In this talk, we will describe the problem of finding approximate gcds of univariate polynomials and our solution to it.

Given two monic polynomials $f, g \in \mathcal{C}[x]$ with $\deg(f) = m$, $\deg(g) = n$, we wish to find polynomials $\widehat{f}, \widehat{g} \in \mathcal{C}[x]$ with $\deg(\widehat{f}) < m$, $\deg(\widehat{g}) < n$ such that $f + \widehat{f}$ and $g + \widehat{g}$ have a non-trivial gcd and $\|\widehat{f}\|^2 + \|\widehat{g}\|^2$ is minimized. We describe an efficient algorithm to determine $\widehat{f}, \widehat{g}$ and develop bounds the magnitude of the minimum perturbation. *The running time of the algorithm is polynomial in the degrees m, n and a bound on the bit sizes of the coefficients of f, g.* Our techniques are based on minimizing parametric quadratic forms. We then apply our techniques to several variants of this problem including the approximate gcd problem and the nearest singular polynomial problem.

If α is a common root of the perturbed polynomials, then, we show that the perturbation is smallest for that α which minimizes the function

$$\frac{f(\alpha)f^*(\alpha)}{\sum_{i=0}^{n-1}(\alpha\bar{\alpha})} + \frac{g(\alpha)g^*(\alpha)}{\sum_{i=0}^{m-1}(\alpha\bar{\alpha})}$$

where $f^*(\alpha), g^*(\alpha)$ denote the complex conjugates of $f(\alpha), g(\alpha)$ respectively. We also show a generalization of this to the situation when the perturbed polynomials have d distinct common roots $\alpha_1, \ldots, \alpha_d$.

We describe preliminary results of our experiments in trying to minimize the above functions.

A Problem from Quantum Field Theory: How the Roots of a Sequence of Polynomials Approach the Unit Circle

A. D. Kennedy
(adk@scri.fsu.edu)
SCRI, Florida State University, Tallahassee, Florida 32306-4052, U.S.A.

Introduction

One of the most active areas in theoretical high energy physics today is the study of the non-perturbative behaviour of Quantum Field Theory (QFT) by means of large-scale Monte Carlo computations. For technical reasons it is most convenient to consider a QFT in Euclidean space-time, where it is defined by an infinite-dimension functional integral $Z(J) \equiv \int d\phi \exp[-S(\phi) + J\phi]$ where the *field* ϕ and the *source* J are functions over space-time, and the *action* is a *local* function of the field $S(\phi) \equiv \int d^4x\, \mathcal{L}(\phi(x), \partial\phi(x)/\partial x)$. The *Lagrangian* $\mathcal{L}$ specifies the dynamics of the theory. The simplest quantity of physical interest is the *propagator*, or two-point *correlation function*

$$\Delta(x) \equiv \langle \phi(x)\phi(0) \rangle = \left.\frac{\partial^2 \ln Z}{\partial J(x) \partial J(0)}\right|_{J=0} = \frac{1}{Z(0)} \int d\phi\, e^{-S(\phi)} \phi(x)\phi(0).$$

How to define such a functional integral rigorously in general is not known. One method is to define the entire theory on a four-dimension hypercubic lattice (taking the place of the space-time continuum) and to attempt to construct the limit of such a lattice theory as the *lattice spacing* $a \to 0$ in such a way as to obtain a theory with the required symmetries and other properties. The simplest discrete version of the action is $S_L(\phi) \equiv a^4 \sum_{\frac{x}{a} \in \mathbf{Z}^4} \mathcal{L}(\phi(x), \nabla\phi(x))$, where ∇ is a finite difference operator such as $\nabla_\mu \phi(x) = [\phi(x + ae_\mu) - \phi(x)]/a$.

In the past year one of the topics under most active study has been the *improvement* of lattice actions. Just as for solutions of differential equations on a grid the discretization errors can be reduced by considering "improved" difference operators which more closely approximate derivatives in the underlying continuum.

One Dimensional Free Field Theory

For simplicity we shall consider just the simplest case of a one-dimensional free field theory. "Free" means that the action is only quadratic in the field, $S_L(\phi) = a \sum_{\frac{x}{a} \in \mathbf{Z}} \frac{1}{2} \phi(x)[m^2 - \nabla^2]\phi(x)$. The simplest discretization of the Laplacian ∇^2 is $\nabla^2\phi(x) = [\phi(x+a) - 2\phi(x) + \phi(x-a)]/a^2$.

Such a free field theory may be solved exactly: if we define $\tilde\phi(k) \equiv a \sum_{\frac{x}{a} \in \mathbf{Z}} \phi(x) e^{-ikx}$ then we obtain $S_L(\phi)$, which is to be compared with the corresponding continuum result $S(\phi)$, where

$$S_L(\phi) = \int_{-\frac{\pi}{a}}^{\frac{\pi}{a}} \frac{dk}{2\pi} \tfrac{1}{2} \tilde\phi(k)^* \left[m^2 + \left(\frac{2}{a} \sin \frac{ak}{2} \right)^2 \right] \tilde\phi(k); \qquad S(\phi) = \int_{-\infty}^{\infty} \frac{dk}{2\pi} \tfrac{1}{2} \tilde\phi(k)^* [m^2 + k^2] \tilde\phi(k).$$

There are two differences: on the lattice the Fourier transform of the Laplacian is a periodic approximation to k^2, and the domain of integration is compact.

Improved Laplacian

We may approximate $(ak)^2$ on the interval $(-\pi, \pi)$ with n cosines to within an error of $O(a^{2n+2})$,

$$p_n(ak) \equiv \sum_{r=0}^{n} a_{n,r} \cos(rka) = (ak)^2 + O(a^{2n+2}), \quad a_{n,0} = 2 \sum_{r=1}^{n} \frac{1}{r^2}, \quad a_{n,r} = \frac{(-1)^r 4n!^2}{r^2 (n-r)!(n+r)!} \quad \text{for } r > 0.$$

As $n \to \infty$ these coefficients approach those of the Fourier series $k^2 = \frac{1}{3}\pi^2 + \sum_{r=0}^{\infty} (-1)^r 4r^{-2} \cos rk$.

Computing the Propagator

In the continuum the propagator is $\Delta(x) = \int_{-\infty}^{\infty} \frac{dk}{2\pi} \frac{e^{ikx}}{m^2+k^2} = \frac{e^{-mx}}{2m}$, so $\frac{\Delta(x)}{\Delta(0)} = e^{-mx}$. Using the nth order improved lattice Laplacian

$$\Delta_L^{(n)}(x) = \int_{-\frac{\pi}{a}}^{\frac{\pi}{a}} \frac{dk}{2\pi} \frac{e^{ikx}}{m^2 + a^{-2}p_n(ka)} = \frac{a}{2\pi i} \oint_{|z|=1} dz \frac{z^{\frac{x}{a}+n-1}}{(ma)^2 z^n + q_n(z)},$$

where we have introduced the variable $z \equiv e^{ika}$ and the degree $2n$ polynomial $q_n(z) \equiv z^n p_n(ka) = \frac{1}{2}\sum_{r=0}^{n} a_{n,r}(z^{n+r} + z^{n-r})$. It is obvious that the roots of the denominator of the integrand are of the form $e^{\pm\mu_{n,r}(m)}$ for $r \in \mathbf{Z}_n$, $\mu_{n,r}(m) \in \mathbf{C}$, with the ordering $0 < \operatorname{Re}\mu_{n,1}(m) \leq \operatorname{Re}\mu_{n,2}(m) \leq \cdots \leq \operatorname{Re}\mu_{n,n}(m)$ for $m \neq 0$. We thus find $\Delta_L^{(n)}(x) = \sum_{r=1}^{n} \beta_{n,r}(m) e^{-\mu_{n,r}(m)x}$ where $\beta_{n,r}$ are known algebraic functions of m.

We have $q_n(e^\mu) = e^{\mu n} p_n(-i\mu) = e^{\mu n}\left[-\mu^2 + O(\mu^{2n+2})\right]$, so $\mu_{n,1} = ma[1 + O(a^{2n})]$, and $\frac{\Delta_L^{(n)}(x)}{\Delta_L^{(n)}(0)} = e^{-mx}[1 + O(a^{2n})] + \sum_{r=2}^{n} \beta_{n,r}(0) e^{-\mu_{n,r}(0)x/a}[1 + O(a)]$, where $\operatorname{Re}\mu_{n,r}(0) > 0$ for $r \geq 2$: for example, $\mu_{2,2} = -\ln(7 - 4\sqrt{3})$.

We thus see that as $a \to 0$ the lattice propagators approach the continuum one up to corrections of $O(a^{2n})$ and subleading terms which are not analytic at $a = 0$. The importance of these latter terms depends upon the size of $\operatorname{Re}\mu_{n,2}$, i.e., the location of the root of $q_n(z)/(z-1)^2$ which is nearest to the unit circle $|z| = 1$.

Perfect Improvement

It is interesting to compute the propagator for the limiting case $n \to \infty$, which just corresponds to the integral[3]

$$\Delta_L^{(\infty)}(x) = \int_{-\frac{\pi}{a}}^{\frac{\pi}{a}} \frac{dk}{2\pi} \frac{e^{ikx}}{m^2 + k^2} = \frac{1}{2\pi ma} \operatorname{Im}\left[-e^{-mx}\operatorname{Ei}\big((ma - i\pi)\frac{x}{a}\big) + e^{mx}\left\{i\pi + \operatorname{Ei}\big(-(ma + i\pi)\frac{x}{a}\big)\right\}\right].$$

The asymptotic expansion of the propagator is easily found to be $\frac{\Delta_L^{(\infty)}(x)}{\Delta_L^{(\infty)}(0)} \sim e^{-mx} - \frac{4ma^3}{\pi^4 x^2}\cos\frac{\pi x}{a} + O(a^5)$, which shows that all the polynomial corrections have been removed. It also shows that although the subleading terms are not analytic at $a = 0$, and are of the form $e^{-\mu x/a}$ with $\mu \neq 0$, they are not "exponentially small" because the exponent μ is purely imaginary. The subleading terms are thus only suppressed by a factor a^3 with a non-analytic oscillatory factor.

Conclusions

We have shown that "improving" the Laplacian for a lattice discretization of a Quantum Field Theory can improve the rate at which the lattice propagator approaches the continuum one as the lattice spacing $a \to 0$: the errors are $O(a^{2n})$. However, we have also found that as $n \to \infty$ the subleading terms are no longer exponentially suppressed, and that for any finite order of improvement the exponential suppression is $e^{-\operatorname{Re}\mu_{n,2}(0)x/a}$ where $z_n \equiv \exp\big(-\mu_{n,2}(0)\big)$ is the root of $q_n(z)/(z-1)^2$ which is nearest to the unit circle.

It would be interesting to know how z_n behaves as a function of n for large n either analytically or numerically, as this would tell us how large a lattice spacing a we can use for a given accuracy and order of improvement. The distance from $z = -1$ to the nearest root for n from 2 through 18 are 1.07, 1.06, 1.02, 0.99, 0.95, 0.93, 0.90, 0.87, 0.86, 0.83, 0.82, 0.80, 0.79, 0.77, 0.75, 0.74, and 0.73 (all with an error of ± 0.01), so it can been seen that $z_n \to -1$ only slowly, which requires isolating the roots of very high order polynomials.

Acknowledgements

This research was supported by by the U.S. Department of Energy through Contract Nos. DE-FG05-92ER40742 and DE-FC05-85ER250000.

[3]AXIOM and MACSYMA were unable to evaluate this integral, whereas Mathematica obtained the wrong answer.

Algebraic geometry and Mechanism theory

J-P. Merlet
INRIA, BP 93
06902 Sophia Antipolis Cedex
France

Abstract

Mechanism theory has always been a favorite field of study for mathematicians. But although many powerful results are available, still various open problems remain, especially when one deals with closed-loop mechanisms. This is basically due to the complexity of the equations and inequations which have to be solved, even for the most simple mechanism. Most of these equations are algebraic and are therefore suitable for being solved with the most recent techniques which have appeared recently in algebraic geometry.

The introduction of symbolic computation and of new mathematical tools is of great help for solving many mechanism theory problems. Still many problems remain open:

- what are the parameters for defining displacements of rigid bodies which are the more suitable for solving the motion equations ?
- what is the best available method for solving a given set of equations ?
- how to deal with inequations ?

We present a survey of problems arising in this field which are illustrated by examples, some of them being unsolved at this time:

- design and synthesis of mechanisms
- kinematics and workspace analysis of robots
- kinematics for automotive suspension mechanisms

Linearizing Polynomial Systems

B. Mourrain
INRIA, SAFIR
2004 route des Lucioles
06565 Valbonne
mourrain@sophia.inria.fr

In many applications such as robotics, computational vision, ..., polynomial systems can be given with approximate coefficients, using for instance floating point arithmetic. In these cases, classical approaches such as Groebner bases computations cannot be applied verbatim, because of instability effects. Therefore a natural question, on which we would like to focus here is the following: "How to adapt polynomial algebra to systems given with approximate coefficients ?" We will be concerned here with polynomial systems defining isolated points (over $\mathbb{C}$) or even with 0-dimensional complete intersection (square polynomial systems defining isolated points).

Our point of view is to transform the resolution of a polynomial problem into a linear one, for which we can use well-known and stable routines of our favorite library. Thus we recall briefly how solving a polynomial system I is equivalent to an eigenvalue problem for operators of multiplication in $B = R/I$ where R is the ring of polynomials and we insist on the role that both B and the dual $\hat{B}$ play in this analysis.

In a second part, we present some methods which can be used with approximate coefficients and which yield matrix formulations. They are related to determinantal formulas for resultants.

The first class of systems which fit directly to this approach, is the case of projective complete intersection. Here, there is no point (or a finite number of points) are infinity (in an appropriate sense) and the information needed to describe the quotient B, can be recovered from the multiples of a *small* given degree ν of the initial equations. This is the cornerstone of Macaulay's method for the construction of resultants, which we detail. In the same spirit, we also mention sparse resultants methods, which generalized the construction of Macaulay. Here the hyperplane at infinity is replaced by several parts at infinity, related to the toric variety in which we embed our problem, according to the support of the initial equations. Another method to derive multiplication matrices is based on the construction of Bezoutians. For any complete intersection, this object defines an explicit isomorphism between B and its dual $\hat{B}$, is used to define algebraic residues and often leads to smaller matrix formulations. We analyze the simplest case of projective complete intersection and illustrate these tools for polynomial systems with leading powers of variables. Finally we present some intermediate method between Macaulay's and Bezout's, which generalizes Dizon's method.

The case of affine complete intersection is more difficult to handle, for there combinations of hight degree (d^n if all the n polynomials are of degree d) can lead to polynomials of small degree ($n\,d - n$). We present here some tools which reduce the initial matrices, in order to get rid of the part at infinity.

Next, we assume that the multiplication matrices are given and we detail some methods which allow us to compute all the roots or only selected roots of the system, or to count the number of real roots in a domain. We end with some implementation considerations and some examples.

Hybrid Computation
Rational Function Approximation, Continued Fraction, and Error Estimates

Matu-Tarow Noda and Hiroshi Kai
Department of Computer Science, Ehime University
Matsuyama, 790–77 Japan

In this paper, we discuss how to estimate error of a hybrid rational function approximation algorithm. The algorithm is proposed by Noda, Miyahiro and Kai(1992) and is called as HRFA. In the algorithm, an approximate–GCD algorithm proposed by Sasaki and Noda(1989) is effectively used. The approximate–GCD of two polynomials with floating point coefficients can be obtained by slightly modified Euclidean GCD algorithm. There are another approaches to obtain the approximate–GCD have been intensively proposed by many authors. The approximate–GCD is applied to solve the problems which are hard to obtain results only by the symbolic or by the numeric computations. Among them the HRFA is one of the most important and interesting application of the approximate–GCD.

The HRFA gives an interpolating rational function as follows:

1. input is a set of data $(x_i, f_i), i = 0, \cdots, m+n$,
2. interpolate input data to a rational function $r_{m,n}$,
3. $r_{m,n}$ has unnecessary singularities by floating point computation,
4. to eliminate singularities in $r_{m,n}$, compute an approximate–GCD of the numerator and denominator polynomials ,
5. divide $r_{m,n}$ by the approximate–GCD and obtain a lower degree rational function which is called as *reduced rational function.*

The rational function $r_{m,n}$ passes through given data sets exactly. On the other hand, reduced rational function $\hat{r}_{m',n'}$, where $m' \leq m$ and $n' \leq n$, assures the continuity, but, in general, doesn't through given data set exactly. Then the error between both rational functions must be shown precisely.

To estimate the error contained in the HRFA algorithm, we consider following several approaches.

1. The approximate–GCD is expressed the continued fraction and rewritten in the Padé approximation form. The error is estimated by using some established facts in Padé approximation.
2. Coefficients of the rational function are replaced by interval numbers, where the approximate–GCD plays an important role. The *width* of the interval rational function gives the error bound.

Every one of aboves has both merits but demerits, that is, the first method can be applied only very narrow approximation area and the latter requires extra knowledge on unnecessary zeros of numerator and denominator polynomials.

We reconsider the continued fraction form and use some basic properties of continued fractions. We obtain useful theorems on error estimates of reduced rational functions. Finally, some practical examples show the error contained in the reduced rational function is consistent in our theorems.

Approximation of polynomial zeros and polynomial GCD

Victor Y. Pan
Mathematics and Computer Science Department
Lehman College, City University of New York
Bronx, NY 10468
Internet: VPAN@LCVAX.LEHMAN.CUNY.EDU

Abstract. We define approximate polynomial gcds (greatest common divisors) and extended gcds provided that approximations to the zeros of the input polynomials are available. We relate our novel definition to the older and weaker ones, based on perturbation of the coefficients of the input polynomials, we demonstrate some deficiency of the latter definitions (which our definition avoids), and we propose new effective sequential and parallel (RNC and NC) algorithms for computing approximate gcds and extended gcds. Our stronger results are obtained with no increase of the asymptotic bounds on the computational cost. This is partly due to application of our recent nearly optimal algorithms for approximating polynomial zeros. We also study and develop an alternative approach based on the techniques of Padé approximation and of computations with Hankel and Bezout matrices, and we show a new extension to the problem of computing numerical rank of a Hankel matrix, which is a bottleneck of Padé and Berlekamp-Massey computations, having important applications to signal processing, coding and transmission of information.

PoSSo – Realsolving : Description and Perspectives

F. Rouillier

We will see how to use the different possibilities of the toolbox by studying concrete examples.
In each available configuration, we propose (when possible) a solution for each following item :

- What to do first
- Multivariate problems
  - Hermite's algorithm
  - Rational Univariate Representation
  - Simultaneous inequalities
- Univariate problems (after having computed a univariate representation of a multivariate problem)
  - Sturm-Habicht-sequence
  - Isolation via Sturm-Habicht
  - Isolation via Uspensky
  - Simultaneous inequalities
  - Thom's coding

Principles of Numerical Polynomial Algebra

Hans J. Stetter

Numerical Algebra is more than replacing symbols by specified numbers; it requires principal changes in paradigms and, consequentially, in concepts, approaches, and algorithms. In the following, *numbers* are assumed to be from $\mathbb{C}$, with the metric in $\mathbb{C}$ defined by $|..|$.

Numerical Algebra must be able to deal with situations where some data of a problem (e.g. coefficients of polynomials) are known *with limited accuracy* only so that a specified polynomial actually represents a (small but dense) set of *indistinguishable* polynomials. Thus, zeros of polynomial systems turn into components of *pseudozero sets*. The test whether a point is a pseudozero is constructive and cheap.

The fact that there are no "exact" answers (in the sense of discrete algebra) opens a new *algorithmic approach* to the computation of polynomial zeros:

(i) Find the structure of the problem at *and around* the specified data; from this knowledge, find approximations for the zeros.

(ii) Test the quality of the approximate zeros. If satisfactory, finish; if not

(iii) Improve unsatisfactory zeros individually; goto (ii).

This approach permits and possibly necessitates small changes in the problem in (i); it permits the use of floating-point arithmetic in (i) and (iii), and – with special provisions – in (ii).

Representations of a polynomial ideal $\mathcal{I}$ (e.g. its Groebner basis) depend on a specified *monomial order* in $\mathbb{C}[x_1, \ldots, x_s]$, in particular, the *normal set* $\mathcal{N}$ of $\mathcal{I}$ which is the basis for most computations related to $\mathcal{I}$. However, $\mathcal{N}$ and all associated representations depend *discontinuously* on the data specifying $\mathcal{I}$: there are *representation singularities*. For data approaching a representation singularity, the standard representations "explode" (some coeffcients tend to infinity) which makes them unsuitable for subsequent numerical computations. Generally, there is no irregularity of any sort in the problem itself at and around a representation singularity; it is only due to the chosen representation.

Therefore, in step (i) of the algorithmic approach, we must recognize a potential nearby representation singularity; in this case we must find a *uniform residual basis* $\mathcal{N}_0$ for a full neighborhood of our problem and an associated representation which *varies smoothly* in this neighborhood. The appropriate set $\mathcal{N}_0$ of monomials is the normal set of the "most degenerate" ideal in the neighborhood. Representations based on $\mathcal{N}_0$ ("perturbed Groebner bases") are computationally manageable and permit a stable execution of step (i) near representation singularities.

About an Algorithm to Construct a Canonical Basis for the Ideals in the Ring of Formal Power Series

Nikolay N. Vassiliev
Institute of Theoretical Astronomy, RAS, 191187, 10, Nab. Kutuzova, St. Petersburg, Russia
e-mail: vasiliev@ita.spb.su

Victor F. Edneral
Institute for Nuclear Physics of Moscow University, Leninskie Gory, Moscow, 119899 Russia
e-mail: edneral@theory.npi.msu.su

It is well known fact that the problem to solve systems of polynomial algebraic equations can be reduced to the problem of construction of Groebner basis for the ideals in the ring of polynomials. In the case of the equations in the ring of formal power series this technique does not work because one of the important thing for Buchberger's algorithm is a definition of "leading term" of polynomial. It does not exist in the case of formal power series.

From other side the problem to solve systems of equations for the finite parts of the infinite formal series is very important for different applications. For example we could say about investigations of singular points of the systems of ODE. Another example is the problem of normalization of the ODE. [1,2].

Usually one can consider the solutions for the finite parts of the series only. It allow us to consider the system of the equations over the formal power series ring as usual polynomial system, but this approach does not lead to good result, because adding terms of highest order could change your solutions in lower terms.

We can show that some modification of Buchberger's algorithm allow us to construct a canonical basis whis is full analogue of the GB in the case of ideas in the ring of formal power series. The construction of such canonical basis and the proof of appropriate variant of the Buchberger's theorem about finiteness of the algorithm is very important not only in the connection with the problem to solve the equations in the ring of formal power series but for studying of the structure of the ring of the FPS and it's ideals. This algorithm get us a canonical representation for the ideals of this ring.

We use in our algorithm some special representation for basic elements of the ideal. This representation uses monomials which lie on the border of specially modified Newton polytopes for the series under consideration. For each series of the system of we have only a finite number of such monomials. If we have the system of finite number of "basic" monomials we can represent the series as a sum such monomials with coefficients which are also the series. It is very important thing for the algorithm that these coefficients are invertible elements of the ring of FPS. As a result we can evaluate by usual method S polynomials and make reductions of the leading terms only for Newton polytopes. It means that each modified Newton polytope of the series of the equations is irreducible relatively others modified Newton Polytopes. We suggest to use such sets of FPS as standard basis in the ring of FPS. Such sets of FPS have many qualities of usual Groebner basis in the case of polynomials. The proofs of the variants of the Buchberger's theorems are based on the fact that the ring of FPS over a Noetherian commutative ring is Noetherian.

References

[1] Bruno A. D. Local Method in Nonlinear Differential Equations. Part I—The Local Method of Nonlinear Analyses of Differential Equations, Part II—The Sets of Analyticity of a Normalizing Transformation. — Springer Series in Soviet Mathematics, 1988. — ISBN 3-540-18926-2, 370 p.

[2] Edneral V. F. Computer Generation of Normalizing Transformation for Systems of Nonlinear ODE. Proceedings of the 1993 International Symposium on Symbolic and Algebraic Computation (Kiev, Ukraine, July 6-8, 1993) / M. Bronstein. — N.Y.: ACM Press, 1993. P. 14–19.

[3] N.N.Vassiliev, V.F.Edneral, "About an Algorithm to Construct a Canonical Basis for the Ideals in the Ring of Formal Power Series", in the book of Abstracts of the Russian Conference "Computer Methods of Celestial Mechanics -95", St. Petersburg, 1995, Ed. by prof Sokolsky.

Condition Number for Multiple Roots.

Jean-Claude Yakoubsohn
Université Paul Sabatier, Toulouse.

We present a first tentative to give a description of a condition number for multiple roots of univariate polynomials. Two approachs are presented.

First by considering the Puiseux series. Let x a root of order p of a polynomial f. We know the roots of perturbated polynomial $f + tg$ have the following representation:

$$x(f + tg) = x + s_1 t^{1/p} + \ldots + s_k t^{k/p} + \ldots$$

These roots are not differentiable when $t \to 0$ but the sum of p roots has this property and its derivative is given by s_p. In the case of a double root:

$$s_2 = \frac{f^{(3)}(x)g(x)}{3f^{(2)}(x)^2} - \frac{g^{'}(x)}{f^{(2)(x)}}.$$

By this way we can define the condition number like the norm of the condition operator:

$$g \in \mathcal{P}_d \to \frac{f^{(3)}(x)g(x)}{3f^{(2)}(x)^2} - \frac{g^{'}(x)}{f^{(2)}} \in \mathbb{C}.$$

But for instance this approach is not a geometric description. Then we propose to consider the application:

$$F_p : (f, x) \in \mathcal{P}_d \times \mathbb{C} \to (f(x), f^{'}(x), \ldots, f^{(p-1)}(x)) \in \mathbb{C}^p$$

and to study the well-posed problem:

$$(f, x) \in \Sigma_{p-1} = \{(g, x) \ : \ F_p(g, x) = 0 \ \}$$

In this case it is possible to give a geometric deescription of the condition number for the multiple roots, i.e, the condition number is the inverse of the distance of polynomial f to the ill-posed problems in a sense which is precised in [1] or [2].

References.

[1] J-P Dedieu, Approximate Solutions of Numerical Problems, Condition Number Analysis and Condition Number Theorems, To appear in: Proceedings of the Summer Seminar on " Mathematics of Numerical Analysis: Real Number Algorithms", Park City, UT, July 17- August 11, Lecture in Applied Mathematics.

[2] M. Shub, S. Smale, Complexity of Bézout's Theorem 1: Geometric approach.Journal of the Amer. Math. Soc. 6 (1993) 459-501.

Résumés en français

Quand deux polynômes numériques sont–ils premiers entre eux?

Bernhard Beckermann
Laboratoire d'Analyse Numérique et d'Optimisation, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq CEDEX, France, e–mail: bbecker@ano.univ-lille1.fr
et
George Labahn
Department of Computing Science, University of Waterloo, Waterloo, Ontario, Canada, e–mail: glabahn@daisy.uwaterloo.ca

Etant donnés deux polynômes a et b à coefficients numériques, nous étudions quand ces polynômes sont premiers entre eux. Dû au fait que les coefficients sont connus seulement approximativement, un problème équivalent est de savoir si a et b restent premiers entre eux même après des petites perturbations des coefficients.

Dans ce travail, nous proposons une méthode rapide pour répondre à cette question. Notre approche est basée sur une formule explicite d'inversion des matrices de Sylvester pour établir un critère efficace. La formule d'inversion permet aussi d'estimer le conditionnement du problème linéaire associé. Une modification de l'algorithme de Cabay et Meleshko pour l'approximation de Padé est ensuite utilisée pour calculer les quantités impliquées dans notre critère, d'une manière efficace et numériquement stable.

Factorisation numérique d'un polynôme à plusieurs variables

Anne-Mercedes BELLIDO

Cet exposé traite de la factorisation d'un polynôme à plusieurs variables dans le cas particulier où il peut se factoriser en produit de facteurs linéaires (cf théorie du U-résultant d'un système d'équations algébriques). Il s'agit de la construction d'une méthode numérique itérative calculant simultanément tous les facteurs (cf méthode de Grau à une variable), basée sur la méthode de Newton généralisée pour un système sous-déterminé. L'originalité de la construction est le calcul symbolique de l'inverse de Moore-Penrose de la matrice Jacobienne utilisée, calcul qui permet d'obtenir une expression très simple de la fonction itérative.

Remarques sur límplémentation d'un algorithme pour le calcul numérique des racines d'un polynôme

D. Bini et V. Burchielli

Etant donné un nombre $\epsilon > 0$ et un polynôme a, nous définissons l'ϵ-voisinage $\mathcal{N}_\epsilon(a)$ dans l'espace des polynômes et l'ϵ-voisinage $\mathcal{RN}_\epsilon(a)$ des racines de a dans le plan complexe. Nous proposons ensuite un test de proximité qui nous permet de discuter de l'implémentation d'une «routine chercheuse» de zéros de polynômes pouvant compter, isoler au sens de Newton, ou calculer une ϵ_{sortie}-valeur approchée des

racines dans un ensemble donné. Cet algorithme, écrit en Fortran 90, peut être utile pour estimer une borne supérieure du pgcd approché de deux polynômes ou pour faire un calcul exact du pgcd.

Algorithmes d'optimisation pour le PGCD en coefficients flottants

P. Chin et R. M. Corless

Pour des polynômes à coefficients approchés, nous décrivons deux méthodes d'optimisation. La première est très simple à implémenter, tandis que la deuxième, plus compliquée, peut s'avérer plus efficace.

Le conditionnement d'un système polynomial creux

J.-P. Dedieu
Laboratoire Approximation et Optimisation, Université Paul Sabatier, 31062 Toulouse Cedex France.
e-mail : dedieu@cict.fr

Nous définissons le conditionnement d'un système polynomial creux et nous montrons que la complexité d'une méthode de continuation pour résoudre de tels systèmes dépend de cet invariant.

Le logiciel NORT pour manipuler des séries en plusieurs variables et ses applications

V. F. Edneral

Le logiciel NORT manipule des séries pour résoudre des équations différentielles. Il est écrit en Standard LISP, ses résultats sont compatibles avec REDUCE et il peut être utilisé comme une extension du dernier. Pour un problème en 8 dimensions, nous avons pu manipuler des séries avec 150000 éléments sur un ordinateur avec 64 MBytes de mémoire RAM.

Le PGCD approché certifié en une variable

I.Z. Emiris, H. Lombardi et A. Galligo

Nous définissons un saut dans les valeurs singulières des matrices de sous-résultant qui garantie le degré du PGCD approché. La preuve du théorème est constructive et donne un algorithme pour calculer le PGCD lui-même. Cet algorithme a une complexité polynomiale en les degrés des polynômes donnés et calculé et, de plus, il est numériquement stable puisqu'il utilise la décomposition SVD. Une norme pondérée est aussi présentée parce qu'elle aboutit, probablement, à des bornes plus précises.

Factorisation absolue de polynômes à deux variables ayant pour coefficients des nombres flottants

André Galligo et Stephen M. Watt
Université de Nice-Sophia Antipolis

Etant donné un polynôme $p(x, y)$ de degré d ayant pour coefficients des nombres flottants, on recherche deux polynômes f_1 et f_2 de degrés d_1 et d_2 tels que $d = d_1 + d_2$ et $f_1.f_2 = f = p + \Delta p$.

Si p est sans facteur carré, on considère en un point générique x_0 une solution implicite $y(x)$ de $p(x, y)$ que l'on développe en série. Des candidats facteurs sont obtenus en considérant des polynômes de degrés croissants qui admettent des développements limités de $y(x)$ comme solutions implicites tronquées.

Les candidats facteurs sont testés par des divisions approchées avec une certaine tolérance pour Δp. La division approchée nécessite la résolution d'un système linéaire, on utilise un calcul de valeurs singulières "SVD" pour déterminer un rang numérique.

Les polynômes considérés sont en représentation dense et on s'autorise des changements linéaires génériques de coordonnées.

Enfin, cette approche pourra s'étendre en n variables $n > 2$ en considérant des restrictions à des plans génériques de C^n.

Programmation avec des idéaux approchés

P. Gianni
Département de Mathématiques, Université, via Buonarroti 2, I–56127 Pise
et
B. Trager
IBM Research, Yorktown Heights, NY

Sous certaines hypothèses, nous pouvons utiliser l'algèbre linéaire pour calculer une base pour l'espace vectoriel de tous les polynômes de n'importe quel degré donné d'un idéal. Cette approche peut même être utilisée lorsque les générateurs ont des coefficients qui sont "connus approximativement".

En utilisant cette construction, nous obtenons des algorithmes pour le calcul des adjoints, du pgcd à plusieurs variables, et la factorisation à plusieurs variables pour des polynômes donnés approximativement.

Calcul pratique des racines complexes en une variable par l'algorithme de Schönage

X. Gourdon

Schönage a développé en 1982 un algotithme, asymptotiquement rapide, pour factoriser de façon approchée un polynôme en une indéterminée. Du point de vue pratique cet algorithme est robuste, donne des résultats sûrs, et a été conçu pour être implémenté : il n'est donc pas seulement un résultat théorique. Il fait maintenant partie intégrante des bibliothèques PARI et MAGMA et produit de bonnes performances. C'est un algorithme récursif basé sur la notion de cercle de séparation et le calcul de résidus.

Des méthodes formelles et numériques pour résoudre des équations d'équilibre de satellites

S. Gutnik

Les solutions stationnaires des équations dynamiques des satellites sont définies par des polynômes à plusieurs variables que nous étudions analytiquement et algébriquement avec les résultants et la méthode des bases de Gröbner. Avec ces techniques et le logiciel Maple, les classes de solutions stationnaires ont été mises à jour. Les positions d'équilibre et leur stabilité sont analysées numériquement.

Le théorème de Kharitonov et ses applications en calcul symbolique

M. Hitz et E. Kaltofen

Nous travaillons sur les familles de polynômes sur $\mathbb{Q}$ ou $\mathbb{C}$, dont les coefficients sont définis par des fonctions continues ou soumis à des contraintes. Pour tout domaine (simplement connexe) $\mathcal{D} \subset \mathbb{C}$, existe-t-il des algorithmes effectifs qui permettent de savoir si les racines de *tous* les éléments de la famille sont dans $\mathcal{D}$? En 1978, V. L. Kharitonov a montré que pour les polynômes dont les coefficients sont des intervalles réels, il est nécessaire et suffisant de tester des éléments particuliers de la famille pour décider si tous les polynômes sont Hurwitz. Il a étendu son résultat dans le cas des coefficients complexes dans un article qui suivent le premier ; huit tests sont nécessaires dans ce cas. Malgré le fait que ce résultat ait été publié dans une revue occidentale, il resta largement inconnu jusqu'en 1983, lorsque Barmish et Bialas l'introduisirent dans la communauté de la théorie du contrôle. A notre connaissance la profondeur du théorème de Kharithov en calcul formel est loin d'être reconnue. La preuve originale est basée sur un théorème de Hermite-Biehler. En suivant Minnicelli, nous exhibons une preuve géométrique qui donne les grandes lignes d'un algorithme général engendrant des test d'ensembles en d'autres normes et pour des domaines différents. Le résultat de Tempo pour une famille de polynômes soumis à des contraintes en norme-1 en est un exemple. Le domaine en question est le cercle unité, les tests d'ensembles consistent en quatre polynômes complexes. Enfin, nous donnons une généralisation pour les cercles accumulant l'origine et pour des secteurs arbitraires à partir de ces cercles. Il est existe un lien extraordinaire entre domaine des racines, normes, et tests d'ensembles encore inexploré. Des conditions générales d'existence de tests d'ensembles à la Kharitonov pour des domaines complexes ont été publiées. Cependant la plupart d'entre elles nécessitent un nombre exponentiel d'éléments (de la famille de polynômes), et très peu, utilisant un ensemble fini d'éléments, sont connues. Pour les domaines connus avec des tests d'ensembles de cardinal sous-exponentiel, nous disposons d'une procédure de décision en temps polynomial, en calculant les indices de Cauchy.

Résolution de Contraintes de Polynômes Approchés

H. Hong

Une clarification de la notion de "résolution" sera donnée dans un premier temps. Puis des idées et méthodes impliquant plusieurs domaines pour cette résolution seront présentées, pour finir par une présentation du logiciel en cours de développement : STURM.

Représentation de courbes et de surfaces implicites par des méthodes d'exclusion

Olivier Isson

Le but est de représenter la variété réelle définie par une équation du style $P(X) = 0$ avec P un polynôme et X $\in C^n$. En effet, l'algorithme d'exclusion donne lieu à une liste d'hypercubes qu'il faut représenter de telle façon à obtenir des courbes fines et correctement dessinées (en 2D) et à faire apparaitre tous les reliefs des surfaces (en 3D). Pour ceci, différentes méthodes seront employées et les résultats ainsi obtenus seront exposés.

Sur le plus grand commun diviseur polynomial approché

N. K. Karmarkar et Lakshman Y. N.

Nous décrivons un algorithme efficace pour ce problème, dont la complexité est polynomiale en les degrés des polynômes donnés et la taille de leur coefficients. Notre méthode est fondée sur la minimisation de formes quadratiques paramétrées.

Un problème d'électro-dynamique quantique : Comment les racines d'une suite de polynômes approchent le cercle unité

A.D. Kennedy

On étudie le comportement sans perturbations de l'électro-dynamique quantique en utilisant des calculs aléatoires de grande échelle et de type Monte-Carlo.

Géométrie algébrique et théorie des mécanismes

J-P. Merlet

La théorie des mécanismes a toujours été un domaine d'application favori des mathématiciens. Mais même si des résultats importants ont été établis, de nombreux problèmes non résolus subsistent, en particulier pour les chaînes fermées. Cela est du principalement à la complexité des équations et inégalités que l'on doit traiter, même pour le mécanisme le plus simple. Dans la plupart des cas, ces équations sont algébriques et sont donc susceptible de faire l'objet d'un traitement avec les outils les plus modernes de la géométrie algébrique.

L'utilisation du calcul formel est d'un grand apport dans cette branche mais se pose toutefois de nombreuses questions:

- quel est le paramétrage le plus approprié pour la résolution des équations (en particulier pour paramétrer les déplacements d'un solide)?
- quels sont les méthodes les plus efficaces pour résoudre le système d'équations obtenu?
- comment traiter les inégalités?

On présentera un état de l'art du domaine et on l'illustrera par des exemples pratiques dont certains sont encore non résolus à l'heure actuelle :

- la conception et la synthèse de mécanismes,
- les modèles géométriques et l'analyse des espaces de travail en particulier pour les robots et les suspensions automobiles.

Linéarisation de systèmes d'équations polynomiales

B. Mourrain
INRIA, SAFIR, 2004 route des Lucioles, 06565 Valbonne, mourrain@sophia.inria.fr

Nous nous intéressons à la résolution de systèmes d'équations polynomiales avec des coefficients approchés et aux applications de ces méthodes à la robotique et la vision. Dans ces cas là, les approches classiques telles que les bases de Gröbner sont inadaptées à cause des effets d'instabilité.

L'objectif est donc de pouvoir transformer le système d'équations polynomiales en un système linéaire pour lequel on sait appliquer des algorithmes bien connus et stables. Ces méthodes consistent à réduire le problème de résolution à un calcul de valeurs propres de matrices de multiplications. Elles peuvent être utilisées avec des coefficients approchés et s'appuient sur des formulations matricielles permettant de construire des Résultants (méthode de Macaulay, méthode de Bézout/Dixon, du résultant creux, ...).

Calcul Hybride: Approximation d'une Fonction Rationnelle, Fraction Continue et Estimations d'Erreur

M.-T. Noda et H. Kai

Plusieurs approches pour estimer l'erreur d'un algorithme d'approximation d'une fonction rationnelle hybride (HRFA) sont présentées. Dans le HRFA, un algorithme de PGCD approché est utilisé, le HRFA contituant une des principales applications du PGCD approché. Chacune de ces approches comporte des avantages et des inconvénients.

On reconsidère la représentation en fraction continue et on utilise quelques propriétés de base des fractions continues. On obtient des théorèmes utiles sur l'estimation d'erreur de fonctions rationnelles réduites, pour finir par quelques exemples pratiques.

L'approximation des zéros de polynômes et le PGCD polynomial

V. Y. Pan

Nous définissons le pgcd (plus grand commun diviseur) approché et généralisé de polynômes en fonction des valeurs approchées des racines des polynômes en entrée. Nous relions cette nouvelle définition aux plus anciennes et plus faibles en termes de perturbation des coefficients des polynômes en entrée ; nous montrons des lacunes des définitions déjà existantes (lacunes qu'évite notre méthode) ; puis nous proposons de nouveaux algorithmes effectifs séquentiels et parallèles (dans RNC et NC) pour le calcul du pgcd approché et du pgcd généralisé. Ces résultats plus forts sont obtenus sans augmenter pour autant les bornes asymptotiques de coût de ce type de calcul. Ceci est partiellement dû à une application de notre récent algorithme presque optimal pour l'approximation des racines d'un polynôme. Nous étudions aussi et développons une autre approche inspirée des techniques d'approximation de Padé et de calcul de matrices de Hankel et de Bézout. Nous montrons ensuite une nouvelle extension du problème du calcul numérique de rang de matrice de Hankel, goulet d'étranglement des calculs de Padé et de Berlekamp-Massey, ayant d'importantes applications en traitement du signal, théorie du codage et de la transmission d'informations.

Principes d'algèbre polynomiale numérique

H. Stetter

L'algèbre numérique nécessite des changements de concepts, d'approches et d'algorithmes, vu les erreurs sur les données (coefficients de polynômes,...). Pour des représentations d'idéaux polynomiaux basées sur une ordre de monômes, il y a parfois des problèmes avec des coefficients qui tendent vers l'infini, bien que géométriquement il n'y a pas de singularité. Il faut alors modifier un peu l'algorithme. Nous proposons une méthodologie pour chercher les zéros d'un tel système de polynômes : (a) Chercher la structure du problème autour des données et calculer les zéros approchés. (b) Vérifier la qualité des zéros. (c) Améliorer les zéros.

Un algorithme pour construire la base canonique d'un idéal dans l'anneau des séries formelles

N.N. Vassiliev et V.F. Edneral

Nous modifions l'algorithme de Buchberger pour construire une base canonique dans l'anneau des séries formelles en plusieurs variables. La représentation de la base utilise des monômes qui se trouvent sur la frontière d'un polyèdre de Newton spécial.

Le conditionnement des racines multiples

Jean-Claude Yakoubsohn
Université Paul Sabatier, Toulouse

Il s'agit d'étudier le conditionnement des racines multiples d'un polynôme d'une variable. Pour ce faire, nous présentons deux approches. Une est basée sur le développement en série de Puiseux de la racine multiple perturbée et l'autre, plus géométrique, reprend les travaux cités en référence. On se ramène par cette approche à un problème bien posé. On obtient ainsi un théorème de conditionnement géométrique qui dit que le conditionnement est l'inverse de la distance du polynôme à l'ensemble des problèmes mal posés en un sens que l'on précisera.

List of Participants

Jouanaidi Abdeljaoued ESSTT, 5, Ave. T. Hussein, 1008 Tunis, Tunisie
kamel.benrhouma@cck.rnrt.tn

René Aïd LMC-IMAG 44, Av. Felix Viallet 38 000 Grenoble
rene.aid@imag.fr

Anne-Mercedes Bellido Departement de Mathematiques Faculte des Sciences 123 Avenue Albert Thomas 87060 Limoges Cedex
bellido@cict.fr

Dario Bini Dipartimento di Matematica, Universita', via Buonarroti 2, Pisa
bini@morse.dm.unipi.it

Jean-Paul Cardinal 2, rue Niepce 75014 Paris
cardinal@micronet.fr

R. M. Corless Department of Applied Mathematics, University of Western Ontario, London, Ontario, Canada N6A 5B7
rmc@pineapple.apmaths.uwo.ca

Dedieu Jean Pierre LAO. Universite Paul Sabatier 31062 Toulouse Cedex
dedieu@cict.fr

Jérôme Dégot Universite de Limoges, Departement de Mathematiques, 123 Av Albert Thomas 87 060 Limoges Cedex
degot@cict.fr

Edneral Victor Institute for Nuclear Physics of Moscow State University Moscow, 119899 Russia
edneral@theory.npi.msu.su

Mohamed Elkadi Laboratoire de Mathématiques, Université de Nice–Sophia Antipolis, Parc Valrose, Nice 06108, Cedex 2.
elkadi@hera.unice.fr

Ioannis Z. Emiris Projet S.A.F.I.R., I.N.R.I.A. B.P. 93, 06902 Sophia-Antipolis, France
emiris,at,sophia,dot,inria,dot,fr

André Galligo Laboratoire de Mathématiques, Université de Nice–Sophia Antipolis, Parc Valrose, Nice 06108, Cedex 2.
galligo@math.unice.fr

Keith Geddes Department of Computer Science, University of Waterloo, Waterloo, Ontario, CANADA N2L 3G1
kogeddes@daisy.uwaterloo.ca

Patrizia Gianni IBM T.J.Watson Research Center P.O.Box 218 Yorktown Heights, NY 10598, USA
gianni@watson.ibm.com

Mark Giesbrecht Department of Computer Science University of Manitoba Winnipeg, Manitoba, R3T 3T6 Canada
mwg@cs.umanitoba.ca

Marc Giusti GAGE, Ecole polytechnique, 91128 Palaiseau Cedex.

giusti@medicis.polytechnique.fr

Xavier Gourdon Projet Algorithmes, INRIA Rocquencourt, 78153 Le Chesnay Cedex
Xavier.Gourdon@inria.fr

Sergey Gutnik Michurinsky Prosp. 31-2-271, Moscow, 117607, Russia
sg@server.ru

Markus Hitz Dept. of Math and Dept. of Comput. Sci. North Carolina State University
hitzm@cs.rpi.edu

Hoon Hong RISC-Linz, Johaness Kepler University, Linz, A-4040, Austria
hhong@risc.uni-linz.ac.at

Olivier Isson 34 bis rue Leon Soulie 31400 Toulouse
isson@cict.fr

D. Jeffrey University of Western Ontario, London, Ontario, Canada N6A 5B7

Erich Kaltofen Dept. of Math and Dept. of Comput. Sci. North Carolina State University
kaltofen@eos.ncsu.edu

N. Karmarkar Mathematical Sciences Research Center, Bell Laboratories, Murreay Hill, NJ 07974, USA
karmar@research.att.com

Anthony D. Kennedy SCRI Florida State University Tallahassee, FL 32306-4052 U.S.A.
adk@scri.fsu.edu

George Labahn Department of Computing Science, University of Waterloo, Waterloo, Ontario, Canada
glabahn@daisy.uwaterloo.ca

Lakshman Y. N. Dept. of Mathemetics and Computer Science Drexel University, Philadelphia, PA 19104, USA
lakshman@mcs.drexel.edu

Austin A. Lobo Computer & Information Sciences, 103 Smith Hall, University of Delaware, Newark, Delaware 19716-1501, USA
alobo@cis.udel.edu

Henri Lombardi Laboratoire de Mathématiques, Universite de Franche–Comté, Besançon 25030, France.
lombardi@vega.univ-fcomte.fr

Jean-Pierre Merlet Projet PRISME, INRIA, B.P. 93, 06902 Sophia-Antipolis, France
merlet@sophia.inria.fr

Bernard Mourrain Projet SAFIR, INRIA, B.P. 93, 06902 Sophia-Antipolis, France
mourrain@sophia.inria.fr

Matu-Tarow Noda Department of Computer Science, Ehime University Matsuyama 790-77 JAPAN
noda@cs.ehime-u.ac.jp

Victor Y. Pan Mathematics and Computer Science Department, Lehman College, City University of New York, Bronx, NY 10468.
VPAN@LCVAX.LEHMAN.CUNY.EDU

Mohamed Omar Rayes 16000 Bent Tree Forest Circle, 1512, Dallas, Texas 75248 USA
rayes@mcs.kent.edu

Fabrice Rouillier IRMAR, Campus de Beaulieu, Universite de Rennes 1 35042 Rennes cedex
rouillie@emmy.univ-rennes1.fr

David Rupprecht 45, rue d'Ulm 75005 PARIS
rupprech@ens.ens.fr

B. David Saunders Computer and Information Sciences Department Univ. of Delaware Newark, DE 19711 USA
saunders@udel.edu

Hans J. Stetter Technical University of Vienna (115.2) A 1040 Vienna
stetter@uranus.tuwien.ac.at

David R. Stoutemyer 3660 Waialae Avenue, suite 304 Honolulu, HI 96816, USA
swh@aloha.com

Karen L. Stoutemyer 3660 Waialae Avenue, suite 304 Honolulu, HI 96816, USA
swh@aloha.com

Monique Teillaud Projet PRISME, INRIA, B.P. 93, 06902 Sophia-Antipolis, France
teillaud@sophia.inria.fr

Barry Trager IBM T.J.Watson Research Center P.O.Box 218 Yorktown Heights, NY 10598, USA
bmt@watson.ibm.com

Nikolai Vassiliev 191187, Institute for Theoretical Astronomy of RAS 10 Nab.Kutuzova, St.Petersbourg, Russia
vasiliev@ita.spb.su

Thierry Vieville Projet Robotvis, INRIA, Sophia, BP93 06902 Sophia Cedex
vthierry@sophia.inria.fr

Villard Gilles LMC-IMAG 46, av F. Viallet 38031 Grenoble Cedex
Gilles.Villard@imag.fr

S. M. Watt Université de Nice Sophia-Antipolis Laboratoire d'Informatique, Parc Valrose, Nice 06108, Cedex 2.
smwatt@sophia.inria.fr

Jean-Claude Yakoubsohn Laboratoire Approximation et Optimisation Universite Paul Sabatier 31062 Toulouse
yak@cict.fr