Agenda

April 18

10:00 – 11:00	**Registration, coffee**
11:00 – 11:10	**Opening the conference**
11:10 – 12:00	**Vladimir V. Kornyak** Emergence of quantum structures in permutation dynamics
12:00 – 12:40	**N. V. Proskurin** Short history of the field of characteristic 1
12:40 – 13:30	**Gaiane Panina** Oriented area is a perfect Morse function
13:30 – 15:00	**Lunch**
15:00 – 15:30	**Alexander Tiskin** TBA
15:30 – 16:00	**Sergei D. Meshveliani** Provable programming of algebra: particular points, examples
16:00 – 16:30	**Coffee break**
17:15 – 17:45	**Maxim Karev** Universal finite-type invariant of plane curves
17:45 – 18:15	**Alexander Batkhin** Parametric Representation of the Resonance Set of Polynomial
18:15 – 18:40	**Vladimir Kassandrov** Collective algebraic dynamics on a rationally parameterized worldline
19:00	**WELCOME PARTY**

April 19

10:00 – 10:50	**Mark Spivakovsky** On local uniformization in positive characteristic.
10:50 – 11:20	**Coffee break**
11:20 – 12:10	**V.Gerdt, A.Khvedelidze, Yu. Palii** On the ring of local unitary invariants for mixed X-states of two qubits

12:10 – 12:50 **Fedor Petrov**
Divided Symmetrization of Polynomials

13:00 – 15:00 **Lunch**

15:00 – 15:30 **Alexandr V. Seliverstov**
On Cubic Hypersurfaces with Involutions

15:30 – 16:00 **Alexei Yu. Uteshev, Ivan Baravy**
Hankel polynomials in the interpolation problems

16:00 – 16:30 **Coffee break**

16:30 – 17:00 **A. Khvedelidze, A.Torosyan**
On the separability of 2-qubit $X-$states

17:00 – 17:30 **Andrei Malyutin**
Coloring triangulated surfaces

17:30 – 18:00 **Yu. A. Blinkov, V.P. Gerdt, K. B. Marinov**
Computer algebra aided generation of difference approximations to quasilinear evolution equations

18:00 – 18:30 **Sergey Baranov, Victor Nikiforov**
Software Simulation as a Means to Estimate Feasibility of Real-Time Software Applications

April 20

10:00 – 10:50 **D. Grigoriev**
Tropical Differential Equations

10:50 – 11:20 **Coffee break**

11:20 – 12:00 **Nikita Gogin, Mika Hirvensalo**
On the Generating Function of Discrete Chebyshev Polynomials

12:00 – 12:45 **Stesik O.L., Slavyanov S.Yu.**
Maple package generating Painleve equations

12:45 – 13:15 **Ilya Krepkiy**
Construction of automorphisms of 3-ary necklaces

13:00 – 15:00 **Lunch**

15:00 – 15:45 **Semjon Adlaj**
Dzhanibekov's flipping nut and Feynman's wobbling plate

15:45 – 16:15 **Mikhail D. Malykh**
On M.N. Lagutinski method for integration of ordinary differential equations

16:15 – 16:40 **Slavyanov S.Yu., Satco D.A.**
Generation and annihilation of apparent singularities

16:40 – 17:10 **Coffee break**

17:10 – 17:40 **Yu. A. Blinkov, V.P. Gerdt, A. V. Mesyanzhin**
Generation of finite-difference approximations to differential systems by using the package SymPy

17:40– 18:10 **P. V. Fokin, Yu. A. Blinkov**
SymPy-ready algorithm for solving SAT problems using algebraic approach and ZDDss

18:30 **Chamber music Concert. Euler Institute**

April 21

10:00 – 10:40 **Alexander L. Chistov**
Extension of the Newton–Puiseux algorithm to the case of a nonzero characteristic ground field

10:40 – 11:30 **Nikolay Vavilov**
TBA

11:30 – 11:50 **Coffee break**

11:50 – 12:30 **Dmitry Pavlov, Nikolay Vasiliev**
On the computational complexity of the three-body initial value problem

12:30 – 14:00 **Lunch**

14:00 **Bus Excursion around St.Petersburg**
The bus will start from the Euler Institute at 14:00.

19:00 **BANQUET**

April 22

10:00 – 10:40 **David Jeffrey**
Integration of rational polynomial expressions in Rubi

10:40 – 11:10	**Vasily Duzhin, Nikolay Vassilyev** Growth and oscillations of normalized dimensions of standard and strict Young diagrams
11:10 – 11:40	**Coffee break**
11:40 – 12:20	**Mikhail V. Babich** Birational coordinates on nilpotent coadjoint orbits of $\mathrm{SO}(N, \mathbb{C})$
12:20 – 12:50	**Ilia Posov** Mathematical tasks with verifiable parametrized answers
12:50 – 13:20	**Sergei Pozdniakov, Vasiliy Akimushkin** Tools for supporting of competitions and Olympiads
13:20 – 15:00	**Lunch**
15:00 – 15:30	**Victor F. Edneral** On the Application of Modular Arithmetic for Cumbersome Computational Tasks
15:30 – 17:00	**Round table Discussion**
	Closing of the Conference, coffee

Table of content

Dzhanibekov's flipping nut and Feynman's wobbling plate

Semjon Adlaj

Abstract. We demonstrate that explicit highly efficient formulas of motion of a freely rotating rigid body, as observed from an inertial frame, might be obtained (only) after exploring (all) the "symmetries" of its motion. We shall disclose the marvel of Galois' construction (which had evaded Poinsot) behind the final step towards a conclusive analytic solution!

Introduction

Vladimir Arnold discussed "the motion of a rigid body, in the absence of outside forces". He wrote, in [5, p. 146],[1] that

"The second revolution will be exactly like the first; if $\alpha = 2\pi p/q$, the motion is completely periodic; if the angle is not commensurable with 2π, the body will never return to its initial state."

Arnold is referring to Poinsot's construction, which (geometrically) describes the "trajectory" of the "tip" of the angular velocity of a freely moving rigid body. The said angle α is the angle of rotation of the body about the (fixed) angular momentum as the angular velocity (pseudo) vector returns to its initial state in body's (rotating) frame. Aside from implicitly presuming (unknown) integers, neither in the Russian nor in the English edition, did Arnold tell us what p and q were. He told us nothing more on calculating α. Lev Landau, on the other hand, did not undermine the calculation of that angle, nor did he divert our attention to "philosophical" statements concerning eternity, but honestly (twice) admitted the "complexity" in [10, p. 119],[2] and provided further reference. One might go on to trace this issue back to the tragedy when Poinsot joined those who failed to appreciate the exceptional significance of Galois' contributions (in uniting

[1] The next (translated) statement appears on page 130 of the third Russian edition (1989).
[2] The confession is (twice) made on page 155 of the fourth Russian edition (1988).

algebra with analysis), and (consequently!)[3] failed to arrive at an explicit formula for a (body) rotation matrix as a function of time. And, in particular, an explicit formula for calculating the rotation angle (which Arnold had denoted by α) remained out of reach!

Key formulas and calculations

Let A, B and C denote the principal moments of inertia, whereas let w and m denote the angular velocity and the (fixed) angular momentum, respectively. We shall exploit the same letters to denote the corresponding magnitudes, so that consistently with this notation we might express that square of the angular speed as $w^2 = w \cdot w$ and the square of the (constant) magnitude of the angular momentum as $m^2 = m \cdot m$, where the dot $(\cdot)$ denotes the scalar product. Let h denote the (constant) scalar product $m \cdot w$, that is twice the kinetic energy. We might initially assume that the moments of inertia A, B and C are pairwise distinct, and we might impose the additional assumption that $A_2 B_2 C_2 \neq 0$, where

$$A_2 = Ah - m^2,\ B_2 = Bh - m^2,\ C_2 = Ch - m^2.$$

There are two general cases here, namely, the case $B_2 < 0$ for which we impose the ordering $A < B < C$, and the case $B_2 > 0$ for which we impose the ordering $A > B > C$. The projection of the angular velocity w onto body's rotating frame is (doubly) periodic, with quarter period

$$T = \frac{\sqrt{ABC}\,\pi}{2\,M\left(\sqrt{(B-C)\,A_2}, \sqrt{(A-C)\,B_2}\right)},$$

where $M(x, y)$ is arithmetic-geometric mean of x and y. Evidently, the scalar function w^2 is, as well, (doubly) periodic (in any reference frame). Now and here, on this PCA 2016, April 18-23, annual conference in St. Petersburg, Russia, we shall present a simple and powerful formula (which we must attribute to Évariste Galois!)[4] for (highly efficiently) calculating the afore-discussed angle α as

$$\frac{\alpha}{4} = \frac{1}{m}\left(h\,T + \int_0^T \frac{A_2 B_2 C_2\,dt}{ABC\,(m^2 w^2 - h^2)}\right) =$$

$$= \frac{T}{m}\left(h + \frac{B_2\,C_2}{m^2(B-C)}\,N\left(\frac{(B-C)\,A_2}{(A-C)\,B_2}, 0, \frac{m^2(B-C)}{m^2(B-C) + C\,B_2}, \frac{m^2(B-C)}{C\,B_2}\right)\right),^{5}$$

[3] Poinsot, having deprived himself from realizing the crucial relevance of Galois' (deeply constructive) ideas for relating the angular velocity to a (corresponding) rotation matrix, was unable to carry out that last necessary step towards the final solution!

[4] Justice will be served if we attribute that said last step (which could not be accomplished by Poinsot) to Galois himself, who would have had undoubtedly carried it out had he been given a chance! Relevant details on Galois' amazing (yet far from fully appreciated) contribution to elliptic functions and modular equations are given in [3].

[5] Observe that the function α is homogeneous of degree 0 whether viewed as a function of the (principal) moments of inertia (for fixed energy and momentum), or as a function of h and m^2

where the function $N(x,\, a, b, c)$ is defined recursively via the relation

$$N(x,\, a, b, c) = N\Big(\sigma(x, 1),\, \sigma(x, a, c),\, \sigma(x, b, c),\, \sigma(x, c)\Big),$$

$$\sigma(x, y) := \sigma(x, y, y),\ \ \sigma(x, y, z) := \frac{(\sqrt{x} + y)\,(\sqrt{x} + z)}{2\,(y + z)\,\sqrt{x}}.$$

The value of this recursive function is the limit obtained from successively applying (linear) fractional transformations $L(\cdot, a, b, c)$ either to (successive) first arguments x, thereby generating the sequence $\{L(x, a, b, c) := (b-c)(x-a)/((b-a)(x-c))\}$, or to the (constant) value 1,[6] generating the sequence $\{L(1, a, b, c)\}$. Both sequences converge quadratically to their common point, as shown in [1].

Many limiting cases of the formula for calculating α might (and must) be considered but the first (and foremost) is the critical case with strictly vanishing B_2.[7] This case corresponds to a critical separating solution which is missed by Arnold and many others who (innocently) presumed that the case with $\alpha = \infty$ might be safely ignored. We are now being vividly reminded of this omission after a striking observation, made in 1985 (June 25^{th}) by the Soviet cosmonaut Vladimir Dzhanibekov, of a motion in proximity to a critical separating solution. A video demonstration from an orbiting space station is provided in [7]. That observation had gotten the attention of Terence Tao, who shared his interpretation of the phenomenon publicly on Google+ [9]. Another special case of motion was popularized by Richard Feynman in [8], and was subsequently referred to as the "Feynman's wobbling plate". The declared (by Feynman) spin to wobble ratio (2:1) was corrected by Benjamin Chao in 1989 (after Feynman's death) in [6]:

"A torque free plate wobbles twice as fast as it spins when the wobble angle is slight. The ratio of spin to wobble rates is 1:2 not 2:1!".[8]

Of course, another limiting case of our formula readily applies to an axially symmetric rigid body, rotating about its axis of symmetry, that is, $Ch = m^2 = C^2w^2$ and $C = \lambda\, B = \lambda\, A$, with (constant) $\lambda \in [0, 2]$. The spin to wobble ratio is then $1{:}\lambda$. It is, indeed, 1:2 for a "flat plate",[9] and 1:0 for a "rod" (with vanishing C), as was also rightfully noted by Chao. We might further formalize the definition of the spin to wobble ratio as $4hT/(m\alpha)$, in order to extend it to non-symmetric rigid bodies where we observe that the said ratio is

---

(for fixed moments of inertia). This property might be expressed via the relation

$$\alpha\left(\lambda A, \lambda B, \lambda C, \mu h, \mu m^2\right) = \alpha\left(A, B, C, h, m^2\right),$$

where λ and μ might (unnecessarily!) be restricted to be positive.

[6] The sequence of first arguments $\{x\}$ converges quadratically to 1.

[7] With B, as we must reemphasize, is the strictly(!) middle moment of inertia.

[8] Having investigated the so-called Chandler wobble phenomenon, Chao knew the correct ratio before he came across Feynman's error.

[9] The ratio is (exactly) 1:2 (only) when "the wobble angle" is strictly zero, that is, when the plate "does not wobble", not unlike the case where the "small-angle approximation of the period of the simple pendulum" turns out being exact (and unambiguously defined!) for calculating the period of the "resting" pendulum, as explained in [2].

strictly less than 1 if $B_2 < 0$. This ratio nears 1 for that flipping nut (which Dzhanibekov observed). The limit ratio at 1 is actually attained at a critical separating solution (as B_2 strictly vanishes)[10]. With this formally extended definition of the ratio, we are ready to assert that it (as ought be) is 1 for a "totally symmetric" rigid body (with $A = B = C$).

Conclusion

Two distinct classes of motion of a freely moving rigid body are separated by critical solutions,[11] which we shall explicitly demonstrate at this conference. The belief that uniqueness of solution (corresponding to a unique trajectory of motion) ought be determined by initial motion conditions will be scrutinized! Most significantly, the presented formula enables not merely calculating the angle α but might readily be adapted for (highly efficiently) constructing an orthogonal transition matrix from body's (rotating) frame to observer's (fixed) frame,[12] with the time domain (necessarily) compactified by adjoining the point at (complex) infinity, as we were incessantly reminded by Dmitry Abrarov (as clarified in [2]).

References

[1] Adlaj S. *An arithmetic-geometric mean of a third kind!* Available at `http://www.ccas.ru/depart/mechanics/TUMUS/Adlaj/GAGM.pdf`.

[2] Adlaj S. *An analytic unifying formula of oscillatory and rotary motion of a simple pendulum* (dedicated to 70$^{\text{th}}$ birthday of Jan Jerzy Slawianowski) // Proceedings of International Conference "Geometry, Integrability, Mechanics and Quantization", Varna, Bulgaria, 2014, June 6-11: 160-171.

[3] Adlaj S. *Multiplication and division on elliptic curves, torsion points and roots of modular equations*. Available at `http://www.ccas.ru/depart/mechanics/TUMUS/Adlaj/ECCD.pdf`.

[4] *Adlaj S.* Iterative algorithm for computing an elliptic integral // Issues on motion stability and stabilization, 2011: 104–110 (in Russian).

[5] Arnold V. *Mathematical Methods in Classical Mechanics*. Translated by Vogtmann K. & Weinstein A. Second Edition, Springer-Verlag 1989.

[6] Chao B. *Feynman's Dining Hall Dynamics* // Physics Today, **42**(2), 1989: 15.

[7] Dzhanibekov V. Video demonstration from a space station, available at `https://youtu.be/dL6Pt1O_gSE`.

[10] Although, then $\alpha = \infty$ as was already said.

[11] As was the case with the simple pendulum which oscillatory and rotary motion regimens are separated by Abrarov's critical solution, as discussed in [2].

[12] The quarter period T in the first line of the formula (for α) might be replaced with the "current" time t for determining "intermediate" rotation angle values. Highly efficient calculation of a corresponding (incomplete) integral relies on combining the expression in the second line (for calculating complete integral) with "dividing" elliptic arcs, as done in [4].

[8] Feynman R. *Surely, You're joking, Mr. Feynman!* A Bantam W.W. Norton edition, published in February 1985.

[9] Tao T. (on Google+), `https://plus.google.com/+TerenceTao27/posts/e3GLg4Ki4dj`.

[10] Landau L. Lifschitz E. *Course of Theoretical Physics* ***I****: Mechanics*. Second Edition, Pergamon Press 1969.

Semjon Adlaj
Division of Stability Theory and Mechanics of Controlled Systems
Computing Center of the Russian Academy of Sciences
Moscow 119333, Russia
e-mail: `SemjonAdlaj@gmail.com`

Birational coordinates on nilpotent coadjoint orbits of $\mathrm{SO}(N,\mathbb{C})$

Mikhail V. Babich

Symplectic spaces are very interesting, they have many applications in different fields of the modern mathematics. One of the sources of a symplectic structure is the Lie-algebraic construction called *a canonical Lie-Poisson-Kirillov-Kostant structure*. It is defined on any coadjoint orbit of any Lie group. The coordinatization of the orbits is an important problem. We consider a complicated case of the orbit of the orthogonal group, the orbit consisting of nilpotent matrices.

In contrast with the nilpotent orbits the construction of the orbits generated by the diagonalizable matrices is simple. The Darboux coordinates on them can be constructed using *projection and contraction* on the isotropic eigenspaces and the co-isotropic co-eigenspaces. The (co)eigenspaces of non-zero eigenvalues have necessary properties.

The kernel of the nilponent matrix is not isotropic but it has isotropic subspaces. The procedure of projections and contractions gives a set of functions connected by non-trivial equations in the nilpotent case. I will demonstrate how to organize the process of projections and contractions in such a way that the corresponding equations will be solved.

Let us define integer numbers δ_k. The number δ_k is the number of the Jordan blocks the sizes of which are greater than $k\times k$, and the sizes of these blocks have the parity opposite to the parity of k. We put $\delta_k=\delta_{-k}$, and consider the integer variables $n,k\in 1-M,2-M,\dots,M-1$, where $A^M=0\neq A^{M-1}$. Consider any matrix A from the orbit as a block-matrix, formed by the blocks $B_{nk},\ n,k\in 0,\pm1,\dots,\pm(M-1)$. The size B_{nk} is $\delta_n\times\delta_k$. Let the indexes $n,k\in 0,\pm i\dots$ numerate the blocks B_{nk}.

The main observation is the following. If each flight of the process has the smallest size, we get the strictly upper-triangular block-matrix with *the zero blocks along the diagonal* i.e. the blocks having common sides with the diagonal blocks are zero.

The iteration process of the factorization and the contraction produce a pair of the block-triangular matrices. One of them is lower-triangular, it is fixing the

flag related with the matrix on the orbit. The other one is upper-triangular block-matrix with the zero blocks along the diagonal. Non-trivial matrix elements of these matrices form the birational coordinates on the orbit.

Mikhail V. Babich

Software Simulation as a Means to Estimate Feasibility of Real-Time Software Applications

Sergey Baranov and Victor Nikiforov

Abstract. An approach to study the behavior of real-time multi-task applications is described based on a software simulation tool written in the Forth programming language.

Introduction

Software applications for real-time systems (RTS) are usually built as collections of tasks which are sequential programs closed w.r.t. to control flow. During application run its tasks share common system resources: the executive ones (processors and processor cores of multi-core processors) and informational ones – global data arrays, interface registers of peripheral devices, elements of human-machine interface, etc. Access to executive resources is governed by the scheduling mode in use, while for access to informational resources certain access protocols are used.

A key requirement to an RTS software application is guaranteed on-time execution of each of its tasks (called "application feasibility") in all correct situations for the application to run. Application feasibility may be checked either through an analytical estimation of the response time for each application task, or through simulation of the application run with an appropriate software tool. Verifying the correctness of a particular situation is another important task beyond the scope of this presentation.

For an RTS designed to run on a conventional single-core processor exact analytical estimations of their feasibility exist; however, applying these methods to RTS on multi-core processors provides substantially inexact results which are too pessimistic, and no exact analytical estimates are known by now.

In this presentation a software simulation tool is described which ensures a more exact estimation of application feasibility for a RTS application on a multi-core platform under various combinations of scheduling modes and protocols of access to shared resources than the known analytical methods.

1. The Problem

A characteristic feature of an RTS is the requirement for on-time execution, usually expressed as a requirement that for each task τ_i the longevity $r(\tau_i^j)$ of any of its jobs τ_i^j shall not exceed some pre-defined deadline value D_i: $\forall i, j(r(\tau_i^j) \leq D_i)$. With the notion of the task response time $R_i = max\{r(\tau_i^1), r(\tau_i^2), ...\}$ this may be reformulated as $\forall i(R_i \leq D_i)$ with any allowable scenario of system events and is often interpreted as the property of feasibility of the given multi-task application. To check application feasibility, various structural models of its tasks are built and analyzed to provide reliable estimates for the response times of the application tasks, taking into account all impacting factors.

Software simulation is an acknowledged method to check feasibility of real-time multi-task applications. This paper describes an experience of constructing such simulator in Forth with the VFX Forth for Windows [1] as a development platform. A freeware option for the platform is gForth [2]. Forth was selected as the implementations language due to the flexibility it provides for implementing programming solutions. The simulator employs a simple model of a multi-task application under study which may use several scheduling modes with various task priorities for allocation of the processor computational resource and several access protocols to access shared informational resources. The simulator helps to study multi-task application behavior and check whether a given combination of the scheduling mode and access protocol guarantees application feasibility under the given processor performance and system event scenarios. It may also identify the minimal processor performance which still ensures application feasibility under the given conditions.

By now, the nomenclature of scheduling modes and access protocols implemented in the simulator consists of two classical scheduling modes – RM (rate monotonic) and EDF (earliest deadline first) – and three access protocols – NI (no inheritance), BI (basic inheritance), and PI (priority inheritance). However, it may be further extended to simulate systems with other scheduling modes on a multi-processor and/or multi-core platform and other protocols of access to shared informational resources [3].

2. Solution

The computer program RTMT (Real-Time Multi-Tasking) [4] was designed to study various combinations of scheduling modes and protocols access to shared informational resources in real-time multi-task applications for a multi-core platform [4]. A particular interest is how these combinations impact application density [5]. Obtained characteristics determine an optimal selection for application parameter realization which ensure application feasibility under all possible scenarios of their correct interaction with the external environment.

Simulation is based on components of four kinds: *resources, tasks, jobs*, and *events*. Resources and tasks are entities of the application under study; jobs and

events are entities created and operated on by the simulator. Resources and tasks are also represented within the simulator with respective entities. The overall simulation process is governed by the structure *Eventlist* which contains events sorted w.r.t. their timestamps, and the structure *Joblist* which contains ready-to-run jobs sorted w.r.t. their current priorities. The overall simulation structure is represented in Figure 1.

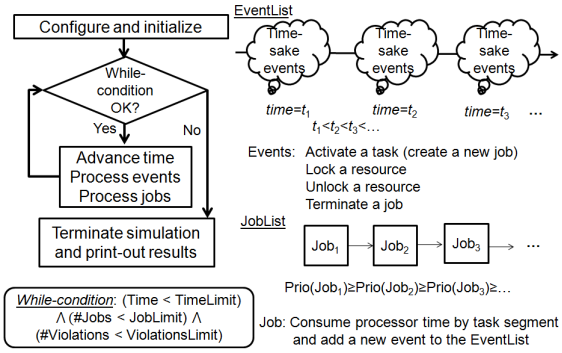


FIGURE 1. The overall structure of the RTMT simulator

The simulation internal loop considers considers time gaps between two successive groups of same-time events in *Eventlist* one after another, starting from *time=0*. The gap size is determined through the minimum of the upper bound of the time yet to be consumed and the time-stamp of the next same-time event group.

For all current jobs at the gap start their counts of consumed time are increased by the gap size, while the counts of yet-to-be-consumed time are respectively decreased. and if this count reaches zero, then a new event is added to *Eventlist* – to terminate this job at the respective time moment. Then all events from the same-time-event group are processed which may change the contents of *Joblist*. Upon completion of processing all these events, the updated contents of *Jpblist* is considered and RTMT transits to the next same-time event group. Thus successive simulation iterations advance the system time and terminate when the specified time limit is exhausted ($Time < TimeLimit$) or when the overall number of created jobs reaches the specified limit ($\#Jobs < JobLimi$) or the total number of registered response time violations is reached ($\#Violations < ViolationsLimit$), whatever occurs earlier. The described internal loop is embedded into the main loop controlled by the parameter of application hardness H for which its bounds

and stride are specified. It turned out that it's more convenient to specifiy the value $1/H$ from which H is recalculated.

Conclusion

The simulator was written in Forth with VFX Forth for Windows, version 4.70, provided to the author at the courtesy of MPE, and is just 985 lines of code under the respective coding standards. It uses only fixed-point arithmetic and works remarkably fast on a PC. To avoid memory overflow, the simulator uses its own simple subsystem for memory allocation and reuse for chained list elements, jobs and events. Further work will be focused on improving the user interface, extending the nomenclature of scheduling modes and access protocols of this simulator, and transition to simulation of multi-core and multiprocessor platforms, as well as running more experiments with models of real-time multi-task applications.

References

[1] VFX Forth for Windows. User manual. Manual revision 4.70, 19 August 2014. Micro-Processor Engineering Limited, Southampton, 2014.

[2] gForth. Free Software Foundation, Inc., (2014), available at `https://www.gnu.org/software/gforth/`.

[3] B. Andersson, S. Baruah, and J. Jonsson, *Static-Priority Scheduling on Multiprocessors.* Proc. of 22^{nd} IEEE Real-Time Systems Symposium. London (2001), pp.193-202.

[4] Baranov S.N. *The Program RTMT for Simulation of Multi-Task Application Run.* Certificate of official registration of a computer program No.201661395, 16 March 2016 (RU), (in Russian).

[5] Baranov, S., Nikiforov, V. *Density of Multi-Task Real-Time Applications.* Proc. Association FRUCT-17. Yaroslavl, (2015), pp.9-15.

Sergey Baranov
Lab.of information and computing systems and software engineering
SPIIRAS
St. Petersburg, Russia
e-mail: `SNBaranov@iias.spb.su`

Victor Nikiforov
Lab.of information and computing systems and software engineering
SPIIRAS
St. Petersburg, Russia
e-mail: `nik@iias.spb.su`

Parametric Representation of the Resonance Set of Polynomial

Alexander Batkhin

Abstract. We consider the resonance set of a real polynomial, i. e. the set of all the points of the coefficient space at which the polynomial has commensurable zeroes. The constructive algorithm of computation of polynomial representation of the resonance set is provided. The structure of the resonance set of a polynomial of degree n is described in terms of partitions of number n. The main algorithms, described in the preprint, are organized as a library of the computer algebra system `Maple`.

Introduction

Let $f_n(x)$ be a monic polynomial of degree n with real coefficients

$$f_n(x) \stackrel{\text{def}}{=} x^n + a_1 x^{n-1} + a_2 x^{n-2} + \cdots + a_n. \tag{1}$$

The n-dimensional space $\Pi \equiv \mathbb{R}^n$ of its coefficients $a_1, a_2, \ldots a_n$ is called the *coefficient space* of polynomial (1).

Definition 1. A pair of roots t_i, t_j, $i, j = 1, \ldots, n$, $i \neq j$, of the polynomial (1) is called $p : q$-*commensurable* if $t_i : t_j = p : q$.

Here and further we consider that $p \in \mathbb{Z}\backslash\{0\}$, $q \in \mathbb{N}$, i.e. we exclude the case when one of the commensurable root t_i or t_j is equal to zero due to the fact that zero root is commensurable with any other root.

Definition 2. Resonance set $\mathcal{R}_{p:q}(f_n)$ of the polynomial $f_n(x)$ is called the set of all points of the coefficient space Π at which $f_n(x)$ has at least a pair of $p : q$-commensurable roots, i.e.

$$\mathcal{R}_{p:q}(f_n) = \{P \in \Pi : \exists i, j = 1, \ldots, n, t_i : t_j = p : q\}. \tag{2}$$

The aim of this work is to present an algorithm of constructing polynomial representation of the resonance set $\mathcal{R}_{p:q}(f_n)$ of the real polynomial $f_n(x)$.

1. Condition on $p:q$-commensurability of polynomial roots

Let polynomial (1) has a pair of $p:q$-commensurable roots. It means that two polynomials $f_n(px)$ and $f_n(qx)$ has common root, or in terms of resultant one has that $\mathrm{Res}_x(f_n(px), f_n(qx)) = 0$. In the case when $p = q$ both polynomials $f_n(px)$ and $f_n(qx)$ have exactly n common roots. In case $a_n = 0$ one of the root is equal to zero, therefore resultant can be written in the form

$$\mathrm{Res}_x(f_n(px), f_n(qx)) = a_n(p-q)^n \,\mathrm{GD}_{p:q}(f_n), \tag{3}$$

where $\mathrm{GD}_{p:q}(f_n)$ is so called *generalized discriminant* of the polynomial (1) introduced in [1].

Polynomial (1) may have some pairs of $p:q$-commensurable roots.

Definition 3. The *chain* $\mathrm{Ch}^{(k)}_{p:q}(t_i)$ *of* $p:q$*-commensurable roots* of length k (shortly *chain of roots*) is called the finite part of geometric progression with common ratio p/q and scale factor t_i, each member of which is a root of polynomial (1). The value t_i is called the *generating root*.

The detail structure of the resonance set (2) can be described with the help of so called *i-th generalized subdiscriminants* $\mathrm{GD}^{(i)}_{p:q}(f_n)$, which are nontrivial factors of i-th subresultants of pair of polynomials $f_n(px)$ and $f_n(qx)$. Such subresultants can be computed as i-th inners of Sylvester matrix constructed from the coefficients of mentioned above polynomials. For more details see [2].

Theorem 1. *Polynomial $f_n(x)$ has exactly $n-d$ different chains of roots $\mathrm{Ch}^{(i)}_{p:q}(t_j)$, $j = 1, \ldots, n-d$ if and only if in the sequence $\left\{\mathrm{GD}^{(i)}_{p:q}(f_n), i = 0, \ldots, n-1\right\}$ of i-th generalized subdiscriminants of $f_n(x)$ the first nonzero subdiscriminant is d-th generalized subdiscriminant $\mathrm{GD}^{(d)}_{p:q}(f_n)$.*

2. Parametrization of the resonance set

Consider a partition $\lambda = [1^{n_1}2^{n_2}\ldots i^{n_i}\ldots]$ of $n \in \mathbb{N}$. Functions $p(n)$ and $p_l(n)$ return the number of all partitions and the number of all partitions of the length l of natural number n respectively. The value i in the partition λ defines the length of chain $\mathrm{Ch}^{(i)}_{p:q}(t_i)$ for a corresponding generating root t_i, the value n_i defines the number of different generating roots, which give the chains of root of the length i. Then $l = \sum_i n_i$ is the number of different generating roots of the polynomial $f_n(x)$ for the certain coefficient of commensurability $p:q$, and $\sum_i i n_i = n$.

Any partition λ of degree n of polynomial (1) defines a certain structure of $p:q$-commensurable roots of this polynomial and it corresponds to some algebraic variety $\mathcal{V}^i_l$, $i = 1, \ldots, p_l(n)$ of dimension l in the coefficient space Π. The number of such varieties of dimension l is equal to $p_l(n)$ and total number of all varieties consisting the resonance set $\mathcal{R}_{p:q}(f_n)$ is equal to $p(n) - 1$. It is so because the partition $[1^n]$ corresponds to the case when all the n roots of polynomial (1) are not commensurable.

Algorithm for parametric representation of any variety $\mathcal{V}_l$ from the resonance set $\mathcal{R}_{p:q}(f_n)$ is based on the following

Theorem 2. *Let $\mathcal{V}_l$, $\dim \mathcal{V}_l = l$, be a variety on which polynomial (1) has l different chains of $p:q$-commensurable roots and the chain generated by the root t_1 has length $m > 1$. Let denote by $\mathbf{r}_l(t_1, t_2, \ldots, t_l)$ parametrization of variety $\mathcal{V}_l$. Therefore the following formula*

$$\mathbf{r}_l(t_1,\ldots,t_l,v) = \mathbf{r}_l(t_1,\ldots,t_l) + \frac{p(q^m v - p^{m-1}t_1)}{t_1(p^m - q^m)}\left[\mathbf{r}_l(t_1,\ldots,t_l) - \mathbf{r}_l((q/p)t_1,\ldots,t_l)\right] \tag{4}$$

gives parametrization of the part of variety $\mathcal{V}_{l+1}$, on which there exists $\mathrm{Ch}_{p:q}^{(m-1)}(t_1)$, simple root v and other chains of roots are the same as on the initial variety $\mathcal{V}_l$.

From the geometrical point of view Theorem 2 means that part of variety $\mathcal{V}_{l+1}$ is formed as ruled hypersurface by the secant lines, which cross its directrix $\mathcal{V}_l$ at two points defined by such values of parameters t_1^1 and t_1^2 such that $t_1^1 : t_1^2 = q : p$. If polynomial $f_n(x)$ has on the variety $\mathcal{V}_{l+1}$ pairs of complex-conjugate roots it is necessary to make continuation of obtained parametrization (4).

Let start from partition $\left[n^1\right]$ which corresponds to variety $\mathcal{V}_1$ with the only chain $\mathrm{Ch}_{p:q}^{(n)}(t_1)$ of roots on it. One can apply transformation (4) of the Theorem 2 in succession and finally can obtain parametrization of variety $\mathcal{V}_{n-1}$ of the maximal dimension $\dim \mathcal{V}_{n-1} = n-1$. There exists only one chain of roots of the length 2 on it and other roots are simple.

Let define the following three operations, which make it possible to obtain parametrization of each variety $\mathcal{V}_l$ of dimensions from 2 to $n-1$.

ASCENT: allows to pass from variety $\mathcal{V}_i$ to the part of another variety $\mathcal{V}_{i+1}$ with dimension one greater.

CONTINUATION: allows to get the parametrization of the entier variety $\mathcal{V}_{i+1}$ obtained on the previous step.

DESCENT: allows to pass from variety $\mathcal{V}_j$, on which there exist two chains of roots with equal length, say k, to variety $\mathcal{V}_{j-1}$, on which there exists a chain of roots with length $2k$.

One can combine successively mentioned above operations to obtain parametric representation of each variety $\mathcal{V}_i$ from the resonance set (2).

Statement 1. *Resonance set $\mathcal{R}_{p:q}(f_n)$ of real polynomial $f_n(x)$ for a certain value of commensurability coefficient $p:q$ allows polynomial parametrization of each its variety $\mathcal{V}_l \subset \mathcal{R}_{p:q}(f_n)$.*

3. Resonance set of cubic polynomial

Consider real cubic polynomial

$$f_3 = x^3 + a_1 x^2 + a_2 x + a_3. \tag{5}$$

It has two generalized subdiscriminants

$$\mathrm{GD}^{(1)}_{p:q}(f_3) = pqa_1^2a_2 + (p^2+pq+q^2)a_1a_3 - (p+q)^2a_2^2,$$
$$\mathrm{GD}^{(0)}_{p:q}(f_3) = (pq(p+q))^2\, a_1^3a_3 - q^3p^3a_1^2a_2^2 - pq\left(p^2+pq+q^2\right)\times$$
$$\times\left(p^2+4\,pq+q^2\right)a_1a_2a_3 + (pq(p+q))^2\, a_2^3 + \left(p^2+pq+q^2\right)^3 a_3^2.$$

Resonance set $\mathcal{R}_{p:q}(f_3)$ shown in Figure 1 consists of two varieties

$$\mathcal{V}_1 : \left\{a_1 = -(p^2+pq+q^2)t_1,\ a_2 = pq(p^2+pq+q^2)t_1^2,\ t_3 = -(pqt_1)^3\right\},$$
$$\mathcal{V}_2 : \left\{a_1 = -(p+q)t_1 - t_2,\ a_2 = pqt_1^2 + (p+q)t_1t_2,\ a_3 = -pqt_1^2t_2\right\},$$

which corresponds to partitions $\left[3^1\right]$ and $\left[1^12^1\right]$ respectively.

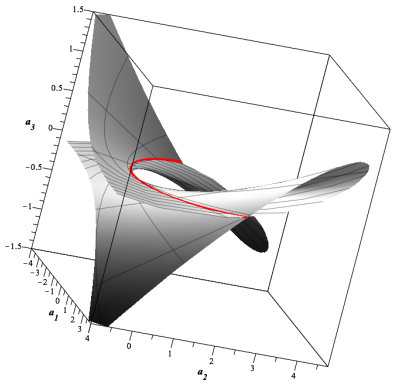


FIGURE 1. Resonance set $\mathcal{R}_{7:1}(f_3)$.

References

[1] A.B. Batkhin, *Segregation of stability domain of the Hamilton nonlinear systems* // Automation and Remote Control (2013), Vol. 74, No. 8, pp. 1269–1283.

[2] A.B. Batkhin, *Parametrization of the discriminant set of a real polynomial*. Preprints of KIAM. 2015. No. 76. 36 p. (in Russian) URL: `http://library.keldysh.ru/preprints.asp?id=2015-76`

Alexander Batkhin
Singular Problem Department
Keldysh Institute of Applied Mathematics of RAS
Moscow, Russia
e-mail: `batkhin@gmail.com`

Generation of finite-difference approximations to differential systems by using the package SymPy

Yu. A. Blinkov, V. P. Gerdt and A. V. Mesyanzhin

In papers [1, 2, 3, 4] we developed an algorithmic approach to generation of finite-difference approximations to polynomially-nonlinear differential equations and systems of such equations.

In the framework of the project GINV [5] we created the program module *PyGinv*, written in the language PYTHON3, for construction of Gröbner bases for difference ideals generated by a set of linear difference polynomials. We illustrate the use of *PyGinv* by the example taken from [6]

$$\begin{gathered} {\phi^{(i)}}_t + 6\phi^{(i)}\phi^{(i)}_\eta + {\phi^{(i)}}_{\eta\eta\eta} + F_i = 0, \quad i = 1,\ 2,\ 3; \\ F_1 = \phi^{(1)} - \phi^{(2)}, \quad F_2 = 2\phi^{(2)} - \phi^{(1)} - \phi^{(3)}, \quad F_3 = \phi^{(3)} - \phi^{(2)}. \end{gathered} \tag{1}$$

Since system (1) consists of three equations, and a Gröbner basis is determined by an admissible monomial ordering, the use of such basis violates the symmetry of the initial differential equation(s). We suggest here first to construct a Gröbner basis for some appropriate ordering, and then by using a certain sequential normal form computation for the constructed Gröbner basis to restore a difference approximation that possesses the symmetry of initial differential equations.

In the case of our example (1) this leads to following finite-difference approximations

$$\begin{gathered} \frac{{u^{(i)}}_j^{n+1} - {u^{(i)}}_j^n}{\tau} + 3\frac{({u^{(i)}}^2{}_{j+1}^{n+1} - {u^{(i)}}^2{}_{j-1}^{n+1}) + ({u^{(i)}}^2{}_{j+1}^{n} - {u^{(i)}}^2{}_{j-1}^{n})}{4h} + \\ + \frac{({u^{(i)}}_{j+2}^{n+1} - 2{u^{(i)}}_{j+1}^{n+1} + 2{u^{(i)}}_{j-1}^{n+1} - {u^{(i)}}_{j-2}^{n+1}) + ({u^{(i)}}_{j+2}^{n} - 2{u^{(i)}}_{j+1}^{n} + 2{u^{(i)}}_{j-1}^{n} - {u^{(i)}}_{j-2}^{n})}{4h^3} \\ + \frac{{F_i}_j^{n+1} + {F_i}_j^{n}}{2} = 0. \end{gathered}$$

The functionality of our program module *PyGInv* is similar to that of the package in [7], but unlike the last package the former one does not use MAPLE and

The work is supported in part by the Russian Foundation for Basic Research (grants No. 16-01-00080 and 16-01-00175).

is free. Moreover, *PyGInv* can be installed and run under (almost) any operating system including Android.

References

[1] V. P. Gerdt, Yu. A. Blinkov, and V. V. Mozzhilkin. Gröbner Bases and Generation of Difference Schemes for Partial Differential Equations. *Symmetry, Integrability and Geometry: Methods and Applications*, 2:26, 2006.

[2] Yu. A. Blinkov and V. V. Mozzhilkin. Generation of difference schemes for the Burgers equation by constructing Gröbner bases. *Programming and Computer Software*, 32(2):114–117, 2006.

[3] V. P. Gerdt and Yu. A. Blinkov. Involution and difference schemes for the Navier-Stokes equations. In *Computer Algebra in Scientific Computing*, volume 5743 of *Lecture Notes in Computer Science*, pages 94–105. Springer Berlin / Heidelberg, 2009.

[4] P. Amodio, Yu. A. Blinkov, V. P. Gerdt, and R. La Scala. On Consistency of Finite Difference Approximations to the Navier-Stokes Equations. In *Computer Algebra in Scientific Computing*, volume 8136 of *Lecture Notes in Computer Science*, pages 46–60. Springer Berlin / Heidelberg, 2013.

[5] Yu. A. Blinkov and V. P. Gerdt. Specialized Computer Algebra System GINV. *Programming and Computer Software*, 34(2):112–123, 2008.

[6] Yu. A. Blinkov, A. Yu. Blinkova, and L. I. Mogilevich. Non-linear Waves in Cylinder Shells with Viscous Incompressible Liquid inside, in Consideration of Energy Dissipation. *Vestnik Moskovskogo aviatsionnogo instituta*, 20(3):186–195, 2013.

[7] V. P. Gerdt and D. Robertz. A Maple Package for Computing Gröbner Bases for Linear Recurrence Relations. *Nuclear Instruments and Methods in Physics Research*, A559:215–219, 2006. arXiv:cs.SC/0509070.

Yu. A. Blinkov
Mechanics and Mathematics Department
Saratov State University
Saratov, Russia
e-mail: `blinkovua@info.sgu.ru`

V. P. Gerdt
Laboratory of Information Technologies
Joint Institute for Nuclear Research
Dubna, Russia
e-mail: `gerdt@jinr.ru`

A. V. Mesyanzhin
Industrial Automatics Design Bureau JSC
Saratov, Russia
e-mail: `a.v.mesyanzhin@gmail.com`

Computer algebra aided generation of difference approximations to quasilinear evolution equations

Yu. A. Blinkov, V. P. Gerdt and K. B. Marinov

• **Research object** • Let ∂_x be the derivation operator w.r.t. x and $\mathcal{R} := \mathbb{Q}(a_1, \ldots, a_i)\{u\}$ be the ordinary differential polynomial ring over the parametric field $\mathbb{Q}(a_1, \ldots, a_i)$ of constants. Here we consider quasilinear evolution equations of the form

$$u_t = au_m + F(u_{m-1}, \ldots, u_1, u)\,, \quad 0 \neq a \in \mathbb{Q}\,, \quad m \in \mathbb{N}_{>0}\,, \tag{1}$$

where $u_k := \partial_x^k u$ $(0 \leq k \leq m)$, $u_0 := u$ and $F \in R$ is a differential polynomial of the order $m-1$ in ∂_x (denotation: $\mathrm{ord}(F) = m-1$) such that there is a differential polynomial $P \in \mathcal{R}$ satisfying

$$F = \partial_x P = \sum_{k=0}^{m-2} u_{k+1} \frac{\partial P}{\partial u_k}\,. \tag{2}$$

Given F, one can algorithmically verify whether or not such P exists and construct it in the case of existence. The equality (2) means that (1) admits the conservation law form

$$u_t = \partial_x \left(au_{m-1} + P\right)\,, \quad P \in \mathcal{R}\,, \quad \mathrm{ord}(P) = m-2\,. \tag{3}$$

The set of evolution equations admitting the polynomial conservation law (3) contains most of classical evolution equations, e.g., the Korteveg-de Vries (KdV) equation and KdV hierarchy, the Burgers equation and Burgers hierarchy, the Kuramoto-Sivashinsky equation, the Burgers-Huxley equation, etc., and their various generalizations (cf. [1]). All these equations have exact solutions that are useful in analysis of numerical methods for their solving.

• **Discretization method** • To discretize equation (3), we follow the approach of paper [2] and convert the equation into the equivalent integral form

$$\oint_\Gamma (P + au_{m-1})\,dt + u\,dx = 0\,, \tag{4}$$

The work is partially supported by the grant No.16-01-00080 from the Russian Foundation for Basic Research.

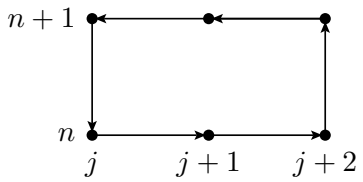


FIGURE 1. Basic integration contour

where Γ is an arbitrary singly connected integration contour. Using the standard notation $u_j^n = u(t_n, x_j)$ for the grid function and the Cartesian grid with $t_{n+1} - t_n = \tau$, $x_{j+1} - x_j = h$ we choose the rectangular integration contour as a "control volume" (cf. [2]) and add $m-2$ integral relations

$$\int_{x_j}^{x_{j+1}} u_k \, dx = u_{k+1}(t, x_{j+1}) - u_{k+1}(t, x_j)\,, \quad k = 1, \ldots, m-2\,. \tag{5}$$

Now, to discretize (4) we apply a numerical evaluation method to the contour integral (4) in order to express it in terms of the grid functions and also (possibly different) numerical evaluation methods to the integrals in the left-hand sides of (5). Thereby, we obtain a system of difference equations containing $u_j^n, u_{1j}^n, \ldots, u_{m-1j}^n$. The last step in generation of a finite difference approximation (FDA) to (1) is algebraic elimination of the grid functions $u_{1j}^n, \ldots, u_{m-1j}^n$, which correspond to the proper partial derivatives of u, from the discrete system obtained. Such elimination can be done by means of the MAPLE package *LDA* [3] which is freely available (`http://wwwb.math.rwth-aachen.de/Janet/`).

• **Example: FDA to the KdV equation•** We illustrate the above described approach by example of the KdV equation

$$u_t + u_{xxx} + 6uu_x = 0\,. \tag{6}$$

Its integral conservation law form for the contour $\mathcal{C}$ of FIGURE 1 reads

$$\oint_{\mathcal{C}} (u_{xx} + 3u^2)\, dt + u\, dx = 0\,. \tag{7}$$

To approximate numerically the contour integral, we apply the trapezoidal rule to the integration over t as well as to the integration over x. For numerical approximations of the integral relations we apply the trapezoidal rule for the integration of u_x and the midpoint rule for the integration of u_{xx}. This leads to the difference approximation to (6) which is outputted by the following MAPLE code shown in FIGURE 3 with $P := 3u^2$. The output is the left-hand side of the FDA to (6) written in the conventional form as

$$\frac{u_j^{n+1} - u_j^n}{\tau} + \frac{(P_{j+1}^{n+1} - P_{j-1}^{n+1}) + (P_{j+1}^n - P_{j-1}^n)}{4h} + \\ + \frac{(u_{j+2}^{n+1} - 2u_{j+1}^{n+1} + 2u_{j-1}^{n+1} - u_{j-2}^{n+1}) + (u_{j+2}^n - 2u_{j+1}^n + 2u_{j-1}^n - u_{j-2}^n)}{4h^3} = 0\,.$$

```
> with(LDA):
> L:=[(-(P(n,j)+P(n+1,j)-P(n,j+2)-P(n+1,j+2)) -
      (uxx(n,j)+uxx(n+1,j)-uxx(n,j+2)-uxx(n+1,j+2)))*tau/2+
      (u(n+1,j+1)-u(n,j+1))*2*h,
      (ux(n,j+1)+ux(n,j))*h/2-(u(n,j+1)-u(n,j)),
      2*uxx(n,j+1)*h-(ux(n,j+2)-ux(n,j))]:
> JanetBasis(L, [n,j], [uxx,ux,u,P],2):
> collect(%[1,1]/(4*tau*h**3),[tau,h]);
```

$$\frac{\frac{1}{4}P(n+1,j+3)+\frac{1}{4}P(n,j+3)-\frac{1}{4}P(n+1,j+1)-\frac{1}{4}P(n,j+1)}{h}$$
$$+\frac{1}{h^3}\left(\frac{1}{4}u(n+1,j+4)-\frac{1}{2}u(n+1,j+3)+\frac{1}{4}u(n,j+4)-\frac{1}{2}u(n,j\right.$$
$$\left.+3)+\frac{1}{2}u(n+1,j+1)-\frac{1}{4}u(n+1,j)+\frac{1}{2}u(n,j+1)-\frac{1}{4}u(n,j)\right)$$
$$+\frac{u(n+1,j+2)-u(n,j+2)}{\tau}$$

FIGURE 2. Construction of FDA to KdV with MAPLE

• **Numerical solution** • Since the obtained FDA to (6) has cubic nonlinearyty (due to $P = 3u^2$) in the grid function on the next time layer, in order to construct a numerical solution we use the following linearization

$$v_{k+1}^2 = v_{k+1}^2 - v_k^2 + v_k^2 = (v_{k+1} - v_k)(v_{k+1} + v_k) + v_k^2 \approx v_{k+1} \cdot 2v_k - v_k^2 \,.$$

By taking this linearization into account, we implemented a numerical procedure for construction of a solution to KdV in Python 2.7 freely downloadable from the Web page `https://www.python.org/download/releases/2.7/`. FIGURE 4 demonstrates the time evolution of numerical solution in the domain $x \in [0, 200]$ with $h = 0.4$ and $\tau = 0.2$ and for the initial value (Cauchy) problem with the initial data

$$u(t = 0) := f(x, 0, 0.4) + f(x - 20, 0, 0.2)$$

where

$$f(x, t, \kappa) := \frac{2\kappa^2}{\cosh^2\left[\kappa(x - 4\kappa^2 t)\right]}$$

is the exact one-soliton solution to (6). As FIGURE 4 shows, the constructed numerical solution reveals a behavior inherent to localized solutions of KdV.

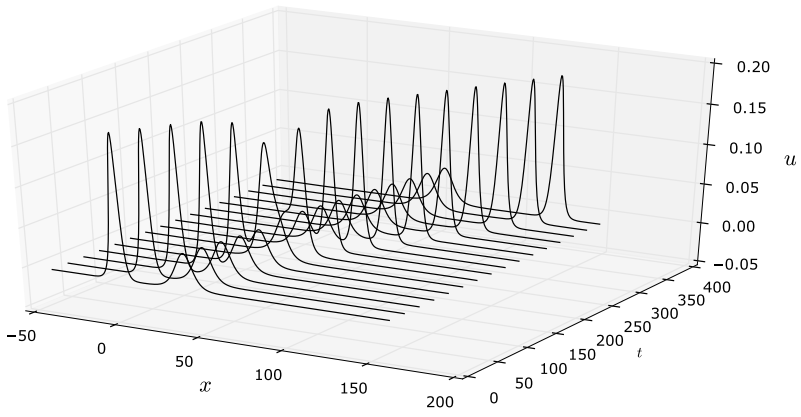


FIGURE 3. Dynamics of solution to KdV

References

[1] N.A.Kudryashov. Methods of Nonlinear Mathematical Physics. Publishing House "Intellect", 2010 (in Russian).

[2] V. P. Gerdt, Yu. A. Blinkov, and V. V. Mozzhilkin. Gröbner Bases and Generation of Difference Schemes for Partial Differential Equations. *Symmetry, Integrability and Geometry: Methods and Applications*, 2:26, 2006.

[3] V. P. Gerdt and D. Robertz. A Maple Package for Computing Gröbner Bases for Linear Recurrence Relations. *Nuclear Instruments and Methods in Physics Research*, A559:215–219, 2006. arXiv:cs.SC/0509070.

Yu. A. Blinkov
Mechanics and Mathematics Department
Saratov State University, Saratov, Russia
e-mail: `blinkovua@info.sgu.ru`

V. P. Gerdt
Laboratory of Information Technologies
Joint Institute for Nuclear Research, Dubna, Russia
e-mail: `gerdt@jinr.ru`

K. B. Marinov
International University "Dubna"
Dubna, Russia
e-mail: `marinov.kohctahtih@gmail.com`

Extension of the Newton–Puiseux algorithm to the case of a nonzero characteristic ground field

Alexander L. Chistov

Abstract. We suggest a generalization of the Newton–Puiseux algorithm for constructing roots of polynomials in the field of fractional power series to the case of nonzero characteristic of the ground field.

Let k be a ground field and $k((X))$ be the field of power series in X with coefficients in k. Let $f \in k((X))[Y]$ be a separable polynomial of the degree $\deg_Y f = d \geqslant 1$. We shall assume without loss of generality that $f \in k[[X]][Y]$ and the leading coefficient $\mathrm{lc}_Y f = 1$ (i.e., the coefficient from $k[[X]]$ of Y^d in the polynomial f). Denote by $\Delta = \mathrm{Res}\,(f, f'_Y)$ the discriminant of the polynomial f.

If the characteristic $\mathrm{char}(k) = 0$ the algebraic closure

$$\Omega = \overline{k((X))} = \bigcup_{\nu \geqslant 1} \overline{k}((X^{1/\nu})). \tag{1}$$

The classical Newton–Puiseux algorithm constructs the roots of the polynomial f in the field Ω using the method of Newton broken lines. Namely let $y_j = \sum_{i \geqslant 0} y_{j,i} X^{\alpha_{j,i}}$ be a root of f where all $y_{j,i} \in \overline{k}$, $\alpha_{j,0} < \alpha_{j,1} < \alpha_{j,2} < \ldots$, all $\alpha_{j,i} \in \frac{1}{e_j}\mathbb{Z}$ for some $1 \leqslant e_j \leqslant d$ (to fix e_j we assume that it is minimal possible). Then for every $r \geqslant 0$ the pair $(y_{j,r}, \alpha_{j,r})$ can be found considering the Newton broken line of the polynomial

$$f\left(Y - \sum_{0 \leqslant i < r} y_{j,i} X^{\alpha_{j,i}}\right).$$

This is an essence of the Newton–Puiseux algorithm.

Now the field $K_j = k((X))[y_j] = k_j((\pi_j))$ where k_j the field of residues of the field K_j and $\pi_j = X^{1/e_j}$ is a uniformizing element of the field K_j. The field k_j is a finite extension of the field k and generated over k by all the elements $y_{j,i}$ (actually by a finite number of them). The degree $[k_j : k] = f_j \leqslant d$. The degree of the minimal polynomial of the element y_j over $k((X))$ is equal to $f_j e_j$.

In what follows we suppose that $\mathrm{char}(k) = p > 0$. Then there are difficulties in comparison with the case $\mathrm{char}(k) = 0$. First of all one can not describe the field $\Omega = \overline{k((X))}$ in a simple way. Namely, (1) does not hold. More than that, let $y_j \in \Omega$ be a root of the polynomial f. Then in general one can not choose an element $\pi \in \Omega$ such that the root $y_j \in \overline{k}((\pi))$ (for this fixed j).

Still the field $K_j = k((X))[y_j]$ has a discrete valuation

$$\mathrm{ord} : K_j \to \frac{1}{e_j}\mathbb{Z} \cup \{+\infty\}$$

such that $\mathrm{ord}(X) = 1$ and $\mathrm{ord}(\pi_j) = 1/e_j$ for a uniformizing element π_j of the field K_j. The residue field k_j of the field K_j with respect to this valuation is a finite (not necessarily separable!) extension of the field k of degree f_j. Similarly to the case of zero–characteristic the degree of the minimal polynomial of the element y_j over $k((X))$ is equal to $f_j e_j$. There is a system of representatives Σ_j of the field k_j in K_j. We shall assume without loss of generality that $\Sigma_j \supset k$ and Σ_j is a linear space over k (in general one can not choose Σ_j to be an algebra over k). Denote by k_s the separable closure of the field k. Then the field $k_s \cap k_j \subset K_j$. So one can assume that $k_s \cap k_j \subset \Sigma_j$. Now the root y_j can be represented as a sum of the infinite series

$$y_j = \sum_{i_0 \leqslant i \in \mathbb{Z}} y_{j,i}\pi_j^i, \tag{2}$$

where all $y_{j,i} \in \Sigma_j$, $y_{j,i_0} \neq 0$. The field k_j is generated over k by all the residues of the elements $y_{j,i}$, $i \geqslant i_0$.

So the final aim of a generalization of the Newton–Puiseux algorithm for nonzero characteristic is to construct for every root y_j of the polynomial f a uniformizing element π_j, a system of representatives Σ_j and the expansion (2). More precisely, to obtain (2) it is sufficient to construct all the elements $y_{j,i} \in \Sigma_j$ for $i_0 \leqslant i \leqslant 1 + \mathrm{ord}(\Delta)$ (we assume that $\mathrm{ord}(\Delta)$ is known). After that subsequent elements $y_{j,i}$ can be found in a simple way using a variant of the Hensel lemma.

Unfortunately one can not obtain at once Σ_j and π_j. So we construct a finite number of elements $z_1, z_2, \ldots, \eta_1, \eta_2, \ldots$ (they depend on y_j; in what follows j is arbitrary but fixed) satisfying the following properties. For every m the orders $\mathrm{ord}(z_m) = a_m/(b_m p^{s_m})$, where $\mathrm{GCD}(a_m, p) = 1$, $\mathrm{GCD}(b_m, p) = 1$ and $s_m > s_{m-1}$ (we put $s_0 = 0$). Further, for every m denote by $\overline{\eta}_m$ the residue of the element η_m. The the field $k_s[\overline{\eta}_1, \ldots, \overline{\eta}_m]$ is purely inseparable over the field k_s and has the degree p^{r_m} over k_s where $1 \leqslant r_m \in \mathbb{Z}$ and $r_m > r_{m-1}$ (we put $r_0 = 0$).

Set $w(0) = v(0) = w(1) = v(1) = 0$, $\widetilde{y}_1 = y_j$. At the beginning of the q-th step of the algorithm the elements $z_1, z_2, \ldots, z_v$, $\eta_1, \eta_2, \ldots, \eta_w$ and $\widetilde{y}_q$ are known. Here the integer $q \geqslant 1$ and we shall write $w = w(q)$, $v = v(q)$. We have

$$v(q-1) \leqslant v(q) \leqslant v(q-1) + 1, \quad w(q-1) \leqslant w(q) \leqslant w(q-1) + 1,$$
$$(v(q-1), w(q-1)) \neq (v(q), w(q)) \quad \text{for} \quad q \geqslant 2.$$

Put $u = u(q) = s_{v(q)} - s_{v(q-1)} + r_{w(q)} - r_{w(q-1)}$.

Then using Newton broken lines we construct the expansion

$$\widetilde{y}_q^{p^u} = \sum_{(\alpha,i_1,\dots,i_v,j_1,\dots,j_w)\in A} y_{\alpha,i_1,\dots,i_v,j_1,\dots,j_w} X^\alpha z_1^{i_1}\cdot\ldots\cdot z_v^{i_v}\eta_1^{j_1}\cdot\ldots\cdot\eta_w^{j_w} + \widetilde{y}_{q+1}, \quad (3)$$

where

(i) A is a finite (or empty) subset of $\mathbb{Q}\times\mathbb{Z}^{v+w}$ (depending on q),
(ii) $0\leqslant j_m < p^{r_m - r_{m-1}}$ for all $1\leqslant m\leqslant w$,
(iii) there is an integer a_m such that $a_m\leqslant i_m < a_m + p^{s_m - s_{m-1}}$ for all $1\leqslant m\leqslant v$ (these integers a_m depend on q and y_j; in this extended abstract we don't explain the sense of introducing a_m),
(iv) $\alpha=\beta/\gamma\in\mathbb{Q}$, $\beta,\gamma\in\mathbb{Z}$, and $\mathrm{G\,C\,D}\,(\gamma,p)=1$,
(v) for every $(\alpha,i_1,\dots,i_v,j_1,\dots,j_w)\in A$ the element $0\neq y_{\alpha,i_1,\dots,i_v,j_1,\dots,j_w}\in k_s$.
(vi) for any pairwise distinct $(\alpha,i_1,\dots,i_v,j_1,\dots,j_w),(\alpha',i'_1,\dots,i'_v,j'_1,\dots,j'_w)\in A$ either $(j_1,\dots,j_w)\neq(j'_1,\dots,j'_w)$ or $\alpha+\sum_{1\leqslant m\leqslant v} i_m a_m/(b_m p^{s_m})\neq\alpha'+\sum_{1\leqslant m\leqslant v} i'_m a_m/(b_m p^{s_m})$.
(vii) For every $(\alpha,i_1,\dots,i_v,j_1,\dots,j_w)\in A$

$$\alpha+\sum_{1\leqslant m\leqslant v} i_m a_m/(b_m p^{s_m}) < \min\{\mathrm{ord}(\widetilde{y}_{q+1}),\mathrm{ord}(\Delta)+1\},$$

(viii) the number of elements $\#A$ is maximal possible, i.e., there is not a similar expansion with A' in place of A satisfying (i)–(vii) and such that $\#A'>\#A$.

If $\mathrm{ord}(\widetilde{y}_{q+1})<\mathrm{ord}(\Delta)+1$ then using the element $\widetilde{y}_{q+1}$ one can construct z_{v+1} or η_{w+1} (may be both of them), define $v(q+1)$, $w(q+1)$ and proceed to the next $(q+1)$-th step.

If $\mathrm{ord}(\widetilde{y}_{q+1})\geqslant\mathrm{ord}(\Delta)+1$ then the considered q-th step is final and after that one can construct Σ_j, π_j and expansion (2)

Actually this algorithm is *canonical*. More than that, it is natural to consider the family of expansions (3) for all q as a generalization for nonzero characteristic of one expansion (1) for zero characteristic. Of course we omit details here.

Assume that $f\in k[X,Y]$ and the field k is finitely generated over a primitive subfield. Then the interesting problem is to estimate the complexity of this algorithm and obtain the results in nonzero characteristic similar to [1], [2].

References

[1] **Chistov A. L.:** *"Polynomial complexity of the Newton–Puiseux algorithm"*, in: Lecture Notes in Computer Science, Vol. 233, Springer, 1986, p. 247–255.

[2] **Chistov A. L.:** *"Effective construction of an algebraic variety nonsingular in codimension one over a ground field of zero characteristic"*, Journal of Mathematical Sciences v.179 (2011), Issue 6, p. 729-740.

Alexander L. Chistov
St. Petersburg Department of Steklov Mathematical Institute
of the Academy of Sciences of Russia
Fontanka 27, St. Petersburg 191023, Russia,
e-mail: alch@pdmi.ras.ru

Growth and oscillations of normalized dimensions of standard and strict Young diagrams

Vasilii Duzhin and Nikolay Vasilyev

We present the results of a computer investigation of asymptotics for maximum dimensions of linear and projective representations of the symmetric group. This problem reduces to the investigation of standard and strict Young diagrams of maximum dimensions. We constructed some sequences for both standard and strict Young diagrams with extremely large dimensions. The conjecture that the limit of normalized dimensions exists was proposed by A. M. Vershik 30 years ago [1] and has not been proved yet.

We studied the growth and oscillations of the normalized dimension function in sequences of Young diagrams. Our approach is based on analyzing finite differences of their normalized dimensions. This analysis also allows us to give much more precise estimation of the limit constants.

We are particularly interested in studying the sequence of Young diagrams of maximum dimensions. Unfortunately, it is extremely hard to obtain such a sequence. However, it is possible to construct the so-called greedy sequence. It is constructed by choosing on each step the branch in the Young graph which leads to the diagram of maximum possible dimension. We are studying greedy sequences and expect that the asymptotics of their dimensions is same as the asymptotics of dimensions of maximum diagrams for both standard and strict diagram cases.

References

[1] A. M. Vershik and S. V. Kerov. "Asymptotic behavior of the maximum and generic dimensions of irreducible representations of the symmetric group." Funktsional. Anal. i Prilozhen., 19(1):25-36, 1985.

[2] Vasilyev N. N., Duzhin V. S. Building Irreducible Representations of a Symmetric Group S(n) with Large and Maximum Dimensions. Informatsionno-upravliaiushchie sistemy [Information and Control Systems], 2015, no. 3, pp. 1722 (In Russian). doi:10.15217/issn1684-8853.2015.1.17

[3] Vasilyev N. N., Duzhin V. S. A study of the growth of maximal and typical normalized dimensions of strict Young diagrams. To be published in Zap. Nauchn. Sem. POMI.

Vasilii Duzhin
Saint Petersburg Electrotechnical University
ul. Professora Popova 5, 197376 St. Petersburg, Russian Federation
e-mail: vduzhin.science@gmail.com

Nikolay Vasilyev
St.Petersburg department of Steklov Institute of mathematics RAS
nab.Fontanki 27,St.Petersburg, 191023, Russia
e-mail: vasiliev@pdmi.ras.ru

This work was supported by grant RSF 14-11-00581.

On the Application of Modular Arithmetic for Cumbersome Computational Tasks

Victor F. Edneral

We consider applications of the modular arithmetic to cumbersome computational tasks, i.e. to problems with a lot of operations with cumbersome numbers. Such problems often arise in computer algebra tasks. We mean evaluations of long polynomials with huge numerical coefficients. Traditionally a modular arithmetic is used for each separate arithmetic operation. But It is more effective to execute the programs from the beginning till the end modulo one prime. After several such calculations in modulo different primes we can finally restore the right values of all numbers of the result. With respect of the Chinese remainder theorem if you know remainders from division of a natural number by a number of noncomparable natural numbers you can restore the original number itself if it is not more than multiplication of all these divisors [1, 2]. We assume that all numbers in the problem are integer or rational. There is an generalization of the Chinese theorem for integer and rational numbers.

What advantages gives us modular approach on multiprocessor platforms? The main are:

- if we choose modulo numbers as primes a bit smaller than maximal integer for a used platform (2^{32} or 2^{64}), then in each process in modulo a prime we will have all integers of length shorter than this maximal integer. So if exclude subsidiary operations and restoring the final result integers from modular representation, we will have acceleration approximately as relation of operation of a long integers arithmetic in comparison with ordinary arithmetic. But how many processors do we need? An ordinary length of integer now is $\sim 2^{64} \sim 10^{20}$. It is possible to choose a number of primes a bit smaller than 10^{20} and to split our task on the corresponding number of processor. If we wish to treat integers about $2^{1024} \sim 10^{309}$ we need $log_{10}(2^{1024})/log_{10}(2^{64}) = 16$ processors and for 2^{2048} we need $2 * 16 = 32$ processor units. It is not too much;
- we suppose that each processor has its own RAM. But instead of integers of the original task in our case each processor will keep in RAM integers of an

ordinary length. So, we will have economization of RAM approximately 16 or 256 times in examples above. It can be critical important for large problems.

Takeing into account that all branches of such calculations do not demand any synchronization. It is case of so called *natural parallelization* [3, 4]. So, we can use a multicore computer or clouds or grid calculations.

Of course there are a lot of problems. Main problems are:

- if we need use rational numbers we should be sure that all denominators in the each calculation process are indivisible on the corresponding prime. But any chosen prime is a long integer and in some problems (for example at the normal form calculation [5]) denominators can be very long but consist of short prime factors only. Any case it is simpler to use this approach for integer problems;
- It can be problems with algorithms which have internal branching. But for some methods for example for a Gr$\acute{}$obner basis calculation there are conditions of applicability of the modular approach [6] . So, complicated algorithms should be investigated additionally.

Of course the written above should be discussed.

References

[1] https://habrahabr.ru/post/144886/

[2] Amos Omondi, Benjamin Premkumar, *Residue Number Systems: Theory and Implementation*, 2007.

[3] S.N. Andrianov, A.B. Degtyarev, *Parallel and distributed computing.* Sankt-Petersburg, Solo, pp. 59 (2007), ISBN: 978-5-98340-073-3. In Russian.

[4] Strzodka R. *The Natural Parallelism.* In collection *Facing the Multicore-Challenge*, LNCS 6310, chapter 3 (2010). http://dx.doi.org/10.1007/978-3-642-16233-6_3.

[5] V.F. Edneral, *On algorithm of the normal form building.* Proceed. of the 10th International Workshop on Computer Algebra in Scientific Computing (CASC 2007, September 16-20, 2007. Bonn, Germany), ed. by Ganzha et al., Springer-Verlag, LNCS 4770, (2007) 134142.

[6] Elizabeth A. Arnold, *Modular algorithms for computing Gr$\acute{}$obner bases.* Journal of Symbolic Computation **35** (2003) 403419

Victor F. Edneral
Skobeltsyn Institute of Nuclear Physics
Lomonosov Moscow State University
Leninskie Gory 1(2), Moscow. 119991, Russia
e-mail: edneral@theory.sinp.msu.ru

SymPy-ready algorithm for solving SAT problems using algebraic approach and ZDDs

P. V. Fokin and Yu. A. Blinkov

SymPy is a Python library for symbolic mathematics. The logic module for *SymPy* allows to form and manipulate logic expressions using symbolic and Boolean values. This module provides implementation of DPLL algorithm for solving SAT problems.

We suggest to use algebraic approach for solving SAT problems [6] with few modifications based on reductions of intermediate expressions as in the algorithms for construction Groebner basis. As internal representation of boolean expressions we propose to use ZDD diagrams [2, 3, 4, 5].

References

[1] The Satisfiability Library. `http://www.satlib.org`. Accessed: 2016-03-31.

[2] Shin-ichi Minato. Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems. In *Design Automation, 1993. 30th Conference on*, pages 272 – 277.

[3] Shin-ichi Minato. Zero-suppressed BDDs and their applications. *International Journal on Software Tools for Technology Transfer*, 3(2):156–170.

[4] Donald E. Knuth. Addison-Wesley Professional, 12th edition, 2009.

[5] P. V. Fokin and Yu. A. Blinkov. Presentation of Boolean Polynomials as ZDD-Diagrams. *Bulletin of Peoples' Friendship University of Russia. Series Mathematics. Information Sciences. Physics.*, (2):301–305, 2014.

[6] Samuel Lundqvist. Boolean ideals and their varieties. *Journal of Pure and Applied Algebra*, 219(10):4521–4540, 2015.

P. V. Fokin
Mechanics and Mathematics Department
Saratov State University
Saratov, Russia
e-mail: `fokinpv@gmail.com`

Yu. A. Blinkov
Mechanics and Mathematics Department
Saratov State University
Saratov, Russia
e-mail: `blinkovua@info.sgu.ru`

The work is supported the Russian Foundation for Basic Research (grant 16-01-00080).

On the Generating Function of Discrete Chebyshev Polynomials

Nikita Gogin and Mika Hirvensalo

Abstract. We give a closed form for the generating function of the discrete Chebyshev polynomials. The closed form consists of the MacWilliams transform of Jacobi polynomials together with a binomial multiplicative factor. It turns out that the desired closed form is a solution to a special case of Heun differential equation.

Introduction

The discrete Chebyshev polynomials belong to the rich family of *orthogonal polynomials* (see [7] for a general treatise on the orthogonal polynomials and [2] for a previous work of the authors).

The sum and the scalar product in $\mathcal{P}_N$ are defined pointwise, and the inner product is defined as

$$\langle p, q\rangle_w = \sum_{l=0}^{N} w_l p(l) q(l). \tag{0.1}$$

The *Krawtchouk polynomials* (see [5]) are orthogonal with respect to weight function $w_l = \binom{N}{l}$ and the discrete Chebyshev polynomials with respect to weight function $w_l = 1$ for each l.

As (orthogonal) polynomials with ascending degree, the discrete Chebyshev polynomials form a basis of $\mathcal{P}_N$, and hence any polynomial p of degree at most N can be uniquely represented as

$$p = d_0 D_0^{(N)} + d_1 D_1^{(N)} + \ldots + d_N D_N^{(N)}, \tag{0.2}$$

where $d_l \in \mathbb{C}$. Since the discrete Chebyshev polynomials are orthogonal with respect to constant weight function, they have the following property important in the approximation theory: With respect to norm $||p-q||^2 = \sum_{l=0}^{N}(p(l)-q(l))^2$, the best approximation of p in $\mathcal{P}_M$ can be found by simply taking $M+1$ first summands of (0.2) (see [4], for instance).

1. Preliminaries

1.1. The Discrete Chebyshev Polynomials

There are various ways to construct polynomials orthogonal with respect to inner product (0.1) with weight function $w_l = 1$ so that $\deg(D_k^{(N)}) = k$.

We choose a construction analogous to that of *Legendre polynomials* [7]. We first define the difference operator Δ by $\Delta f(x) = f(x+1) - f(x)$, the binomial coefficient by $\binom{x}{k} = \frac{1}{k!}x(x-1)\dots(x-k+1)$, and finally

$$D_k^{(N)}(x) = (-1)^k \Delta^k \left(\binom{x}{k} \binom{x-N-1}{k} \right). \tag{1.1}$$

The following recurrence relation is quite easy to verify:

$$k^2 D_k = (2k-1)D_1 D_{k-1} - (N+k)(N-k+2)D_{k-2}, \tag{1.2}$$

$D_0 = 1$, $D_1 = N - 2x$ (see [3]). The recurrence (1.2) also extends the definition of D_k to cases $k > N$.

The method of using *generating functions* is among the cornerstones of various areas of mathematics, and does not need any further introduction. On the other hand, the quest for the generating function of the discrete Chebyshev polynomials seems to be a more complicated task. In what follows, we give a closed form for the generating function

$$\sum_{k=0}^{\infty} D_k^{(N)}(x) t^k. \tag{1.3}$$

1.2. A Differential Equation for Jacobi Polynomials

For a nonnegative integer n, the *Jacobi polynomial* $P_n^{(\alpha,\beta)}(x)$ is, up to the constant factor, the unique entire rational solution to the *differential equation (for Jacobi polynomials)*

$$(1-x^2)y'' + \big(\beta - \alpha - (\alpha+\beta+2)x\big)y' + n(n+\alpha+\beta+1)y = 0 \tag{1.4}$$

(see [1]).

In this article, we are interested in Jacobi polynomials with parameters $\alpha = 0$, $\beta = -(N+1)$, where $N > 0$ is a fixed integer. We also substitute x for n and t for x in equation (1.4), and denote $J_x^{(N+1)}(t) = P_x^{(0,-N-1)}(t)$. We usually omit superscript $N+1$ and denote $J_x(t) = J_x^{(N+1)}(t)$. Then $J_x(t)$ satisfies differential equation

$$(1-t^2)J_x''(t) - (N+1-(N-1)t)J_x'(t) + x(x-N)J_x(t) = 0. \tag{1.5}$$

Recall that in this context, x is a fixed nonnegative integer. Polynomial $J_x(t)$ can be expressed as

$$J_x(t) = \frac{1}{2^x} \sum_{k=0}^{x} \binom{x}{k} \binom{x-N-1}{k} (t-1)^k (t+1)^{x-k} \tag{1.6}$$

(see [1]). Since equation (1.5) is clearly invariant under substitution $x \leftarrow N - x$, we have symmetry

$$J_{N-x}(t) = J_x(t) \tag{1.7}$$

1.3. MacWilliams Transform

The *MacWilliams transform* of order x for a polynomial P is defined as

$$\widehat{P}_x(t) = (1+t)^x P(\frac{1-t}{1+t}). \tag{1.8}$$

As definition (1.8) shows, MacWilliams transform is a special case of Möbius transformation together with factor $(1+t)^x$.

In what follows, $\widehat{J}_x(t)$ stands for the MacWilliams transform of J_x of order x. It is then straightforward to uncover a representation for $\widehat{J}_x(t)$:

$$\widehat{J}_x(t) = \widehat{(J_x)}_x(t) = \sum_{k=0}^{x} (-1)^k \binom{x}{k} \binom{x-N-1}{k} t^k. \tag{1.9}$$

The symmetry (1.7) implies straightforwardly

$$\begin{aligned} \widehat{J}_{N-x}(t) &= \widehat{(J_{N-x})}_{N-x}(t) = (1+t)^{N-x} J_{N-x}(\frac{1-t}{1+t}) \\ &= (1+t)^{N-2x}(1+t)^x J_x(\frac{1-t}{1+t}) = (1+t)^{N-2x}\widehat{J}_x(t). \end{aligned}$$

Equality

$$\widehat{J}_{N-x}(t) = (1+t)^{N-2x}\widehat{J}_x(t) \tag{1.10}$$

thus obtained will be important in understanding the alternative representation of the generating function introduced in Section 4.

2. Heun Equation

A differential equation for the MacWilliams transform of $J_x(t)$ can be found easily. For short, we denote $J(t) = J_x(t)$ and $\widehat{J}(t) = \widehat{J}_x(t)$ in the following lemmata.

Lemma 1. *$\widehat{J}(t)$ satisfies differential equation*

$$t(1+t)\widehat{J}''(t) + (Nt + 1 - 2t(x-1))\widehat{J}'(t) + x(x-N-1)\widehat{J}(t) = 0. \tag{2.1}$$

Lemma 2. *Let $T(t)$ be defined as $T(t) = (1+t)^{N-2x}\widehat{J}(-t^2)$. Then $T(t)$ satisfies differential equation*

$$\begin{aligned} (t^3 - t)T''(t) &+ \big(2t(N-2x) + 3t^2 - 1\big)T'(t) \\ &+ \big(N - 2x - tN(N+2)\big)T(t) = 0 \end{aligned} \tag{2.2}$$

3. The Generating Function

By equality (1.9) function $T(t) = (1+t)^{N-2x}\widehat{J}_x(-t^2)$ can be represented as

$$T(t) = (1+t)^{N-2x}\sum_{k=0}^{x}\binom{x}{k}\binom{x-N-1}{k}t^{2k}. \tag{3.1}$$

We are now ready to state the main result.

Theorem 1. *Function*

$$T_{N,x}(t) = (1+t)^{N-2x}\widehat{J}_x(-t^2) \tag{3.2}$$

is the generating function of discrete Chebyshev polynomials, i.e. $\tau_k(x) = D_k(x)$ *for each* $k \geq 0$.

4. Concluding Remarks

The main result of this article is the closed form 3.2 for Discrete Chebyshev polynomials. As the Discrete Chebyshev polynomials play a major role in approximation theory, this result is evidently interesting on its own. We present also some alternative forms for the generating function:

Theorem 2. *The generating function* $T_{N,x}(t)$ *can be also represented as*

$$T_{N,x}(t) = (1-t)^{2x-N}\widehat{J}_{N-x}(-t^2). \tag{4.1}$$

To combine Theorems 1 and 2 into a single presentation is straightforward:

Theorem 3 (The explicit polynomial form for $x \in \{0,1,\ldots,N\}$**).** *The generating function* $T_{N,x}(t)$ *can be presented as a polynomial in* t *of degree* N*:*

$$T_{N,x}(t) = (1+t\cdot\operatorname{sign}(N-2x))^{|N-2x|}\widehat{J}^{(N)}_{\min\{x,N-x\}}(-t^2).$$

Remark 1. Theorem 1 implies that

$$\sum_{0\leq l\leq k/2}\binom{N-2x}{k-2l}\binom{x}{l}\binom{x-N-1}{l} = \sum_{l=0}^{k}(-1)^l\binom{k}{l}\binom{N-x}{k-l}\binom{x}{l}. \tag{4.2}$$

A direct combinatorial proof of (4.2) appears challenging, for instance, the techniques of [6] appear powerless in this case. Theorem 2 implies an identity similar to (4.2).

References

[1] H. Bateman and A. Erdelyi: *Higher Transcendental Functions Vol 2.* McGraw-Hill (1953).

[2] N. Gogin and M. Hirvensalo: *Recurrent Construction of MacWilliams and Chebyshev Matrices.* Fundamenta Informaticae 116(1–4): 93–110 (2012).

[3] M. Hirvensalo: *Studies on Boolean Functions Related to Quantum Computing*. Ph.D thesis, University of Turku (2003).

[4] A. N. Kolmogorov and S.V. Fomin: *Introductory Real Analysis*. Dover (1975).

[5] F.J. MacWilliams and N.J.A. Sloane: *The Theory of Error-Correcting Codes*. North-Holland (1977).

[6] M. Petkovšek, H. S. Wilf, and D. Zeilberger: $A = B$. A. K. Peters Ltd, Wellesley (1996).

[7] G. Szegö: *Orthogonal Polynomials*. Providence, Rhode Island: American Mathematical Society, Colloquium Publications, Volume XXIII (1975).

Nikita Gogin
e-mail: ngiri@list.ru

Mika Hirvensalo
Department of Mathematics and Statistics
University of Turku
FI-20014, Turku, Finland
e-mail: mikhirve@utu.fi

Tropical Differential Equations

D. Grigoriev

Tropical differential equations are introduced. Similar to usual tropical algebraic equations which provide necessary conditions for solvability of systems of polynomial equations in Puiseux series, tropical differential equations express necessary conditions for solvability of systems of differential equations in power (or Hahn) series.

For a system of tropical linear differential equations we prove the existence of the minimal among its solutions. A polynomial complexity algorithm is designed for solving such systems. For tropical non-linear differential systems its NP-completeness is established.

Several open questions are supposed to be discussed and a recent partial answer to one of them by F. Aroca, C. Garay, Z. Toghani who have established a fundamental theorem of tropical differential algebra, namely, that any (tropical) solution of the tropicalization of a differential ideal can be lifted to a power series solution of the ideal.

On the separability of 2-qubit $X-$states

A. Khvedelidze and A.Torosyan

Abstract. It is shown, that there exists 4-parametric family of separable mixed $X-$states of 2-qubits with an arbitrary spectrum of the density matrix.

Introduction

Consider the Hilbert space $\mathcal{H}$ of binary quantum system, represented by the tensor product of 2-dimensional Hilbert spaces, $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$, and known under the name of 2-qubit system. The density matrix ϱ , describing mixed states of system $\mathcal{H}$, is *separable* if it allows the convex decomposition:

$$\varrho = \sum_k \omega_k \varrho_1^k \otimes \varrho_2^k \,, \qquad \sum_k \omega_k = 1, \quad \omega_k > 0 \,, \tag{1}$$

where ϱ_1^k and ϱ_2^k represent the density matrices, acting on the corresponding multiplier of $\mathcal{H}$. Otherwise it is *entangled* [1] .

Are there mixed states which are separable for an arbitrary spectrum of ϱ ? The present note aims to prove, that the answer to this question is affirmative for the wide class of density matrices, describing the so-called $X-$ states (see [2] and the modern review [3] for details). Furthermore, it will be shown, that among the elements of generic 7-dimensional space of $X-$ states with given spectrum, one can point a special 4-parametric family of separable density matrices.

1. Eigenvalue decomposition for $X-$ states

To formulate the statement precisely, consider the density matrices of the form:

$$\varrho_X := \begin{pmatrix} \varrho_{11} & 0 & 0 & \varrho_{14} \\ 0 & \varrho_{22} & \varrho_{23} & 0 \\ 0 & \varrho_{32} & \varrho_{33} & 0 \\ \varrho_{41} & 0 & 0 & \varrho_{44} \end{pmatrix} . \tag{2}$$

In (2) the diagonal entries are real numbers, while elements of the minor diagonal are pairwise complex conjugated, $\varrho_{14} = \overline{\varrho}_{14}$ and $\varrho_{23} = \overline{\varrho}_{32}$. Since non zero elements

in the density matrix (2) are distributed in a shape that reminds the Latin letter "X", the corresponding states are named as the $X-$states.

It can be proved, that the matrix (2) is unitary equivalent to the diagonal matrix

$$\varrho_X = KWP\,\mathrm{diag}\,(\lambda_1,\lambda_2,\lambda_3,\lambda_4)\;PW^{+}K^{+}\,, \tag{3}$$

where the spectrum $\{\lambda_1,\lambda_2,\lambda_3,\lambda_4\}$ forms the partially ordered simplex, $\underline{\Delta}_3$, i.e.,

$$\underline{\Delta}_3: \quad \sum_{i=1}^{4}\lambda_i = 1 \qquad 0\le\lambda_2\le\lambda_1\le 1, \quad 0\le\lambda_4\le\lambda_3\le 1, \tag{4}$$

and

$$W = \left(\begin{array}{c|c} e^{i\frac{\phi_1}{2}\sigma_2} & 0 \\ \hline 0 & e^{i\frac{\phi_2}{2}\sigma_2} \end{array}\right),\; P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \tag{5}$$

The matrix K in (3) is the element from the subgroup $SU(2)\times SU(2)\subset SU(4)$ of the form $K=\exp\,(i\frac{u}{2}\sigma_3)\times\exp\,(i\frac{v}{2}\sigma_3)$. Here σ_2 and σ_3 are the standard 2×2 Pauli matrices.

2. Applying the Peres-Horodecki separability criterion

Having representation (3), one can analyse the separability as a function of density matrices eigenvalues $\{\lambda\}$. According to the Peres-Horodecki criterion [4], which is a necessary and sufficient condition of separability for 2×2 and 2×3 dimensional systems, a state ϱ is separable iff its partial transposition, i.e., $\varrho^{T_2}=I\otimes T\varrho$, is the semi-positive as well.[1] Applying the Peres-Horodecki criterion to the density matrix (3), one can verify that the $X-$state density matrix is separable iff:

$$(\lambda_1-\lambda_2)^2\cos^2\phi_1+(\lambda_3-\lambda_4)^2\sin^2\phi_2\le(\lambda_1+\lambda_2)^2, \tag{6}$$

$$(\lambda_3-\lambda_4)^2\cos^2\phi_2+(\lambda_1-\lambda_2)^2\sin^2\phi_1\le(\lambda_3+\lambda_4)^2. \tag{7}$$

Introducing new variables

$$x=(\lambda_1-\lambda_2)^2\cos^2\phi_1, \quad y=(\lambda_3-\lambda_4)^2\cos^2\phi_2, \tag{8}$$

the inequalities (6) and (7) linearize

$$\begin{cases} x-y\le a, & 0\le x\le c,\\ y-x\le b, & 0\le y\le d, \end{cases} \tag{9}$$

with parameters (a,b,c,d), obeying the inequalities:

$$a+b\ge 0, \quad a+d\ge 0, \quad b+c\ge 0. \tag{10}$$

[1] Here we consider the partial transposition with respect to the ordinary transposition operation, T, in the second subsystem, similarly one can use the alternative action, $\varrho^{T_1}=T\otimes I\varrho$.

The parameters (a, b, c, d) are functions of the density matrix eigenvalues

$$a = (\lambda_1 + \lambda_2)^2 - (\lambda_3 - \lambda_4)^2, \qquad c = (\lambda_1 - \lambda_2)^2,$$
$$b = -(\lambda_1 - \lambda_2)^2 + (\lambda_3 + \lambda_4)^2, \qquad d = (\lambda_3 - \lambda_4)^2 .$$

Now it is an easy task to be convinced that (9) have solutions for all possible values of parameters from (10). In other words, for eigenvalues from the partially ordered simplex (4) the inequalities (6) and (7) determine non empty domain of definition for angles ϕ_1 and ϕ_2 in (3). Typical domains on the $(x, y)-$plane are depicted on the FIGURE 1.

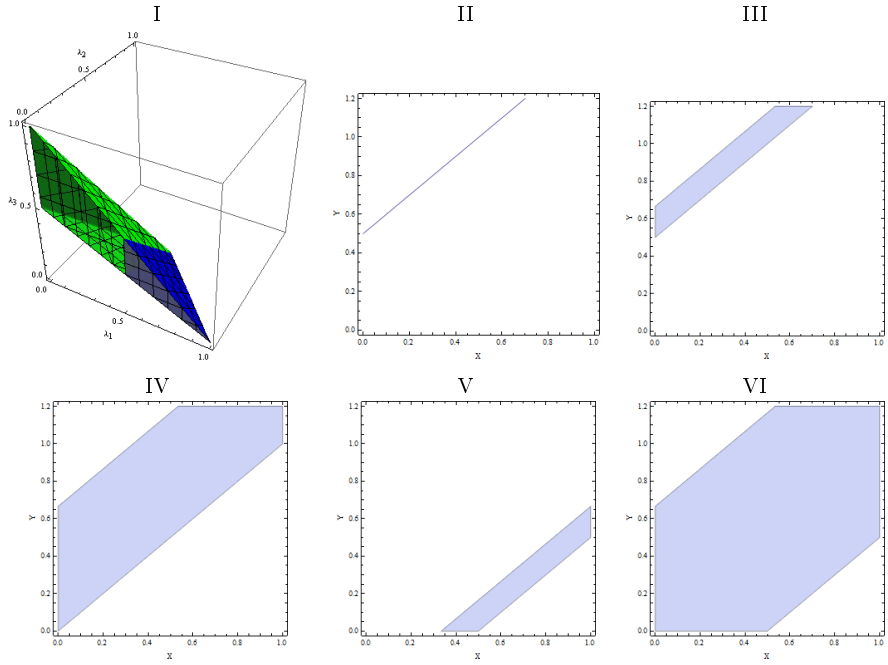


FIGURE 1. Plot (I) - the partially ordered simplex $\underline{\Delta}_3$. Plots (II-VI) - families of solutions to the Eqs. (9) corresponding to the following decomposition of $\underline{\Delta}_3$: Domain (II) : $a < 0$, $b = -a$, $c \geq 0$, $d \geq b$; Domain (III): $a < 0$, $b > -a$, $c \geq 0$, $d \geq -a$; Domain (IV): $a = 0$, $b \geq 0$, $c \geq 0$, $d \geq 0$; Domain (V): $a > 0$, $-a \leq b \leq 0$, $c \geq -b$, $d \geq 0$; Domain (VI): $a > 0$, $b > 0$, $c \geq 0$, $d \geq 0$.

Conclusion

The dependence of separability from the spectrum of 2-qubit density matrix for $X-$states has been discussed. Here it is in order to make few comments on possible generalization of obtained results to an arbitrary 2-qubit states. The Peres -Horodecki separability criterion can be written in the form of polynomial inequalities in the $SU(2)\times SU(2)-$invariant polynomials, (the determinants of correlation and Schlienz -Mahler matrices) [5], [6], treated as unknown variables, and $SU(4)$ Casimir invariants considered as free parameters. [2] Following the analysis, given in the present note, one can conjecture that these inequalities constraint only these two $SU(2) \times SU(2)-$invariant polynomial measures of entanglement.

References

[1] R.F. Werner, Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model, Phys. Rev. **A 40**, 4277-4281, (1989),

[2] Ting Yu and J.H. Eberly, Evolution from entanglement to decoherence of bipartite mixed "X" states. Quantum Information and Computation, **7**, 459-468, (2007).

[3] P. Mendonca, M. Marchiolli and D. Galetti, Entanglement universality of two-qubit $X-$states. Annals of Physics. **351**, 79-103, (2014).

[4] A.Peres, Separability criterion for density matrices, Phys. Rev. Lett. **77** 1413-1415 (1996).

[5] V. Gerdt, A. Khvedelidze and Yu. Palii, On the ring of local polynomial invariants for a pair of entangled qubits, Zapiski POMI **373**, 104-123, (2009).

[6] V. Gerdt, A. Khvedelidze and Yu. Palii, Constraints on SU(2)×SU(2) invariant polynomials for entangled qubit pair, Phys. of Atomic Nucleii, Vol. 74, 893-900, (2011).

A. Khvedelidze
A Razmadze Mathematical Institute,
Iv. Javakhishvili, Tbilisi State University,
Tbilisi, Georgia
Institute of Quantum Physics and Engineering Technologies,
Georgian Technical University,
Tbilisi, Georgia
National Research Nuclear University,
MEPhI (Moscow Engineering Physics Institute),
115409 Moscow, Russia
e-mail: akhved@jinr.ru

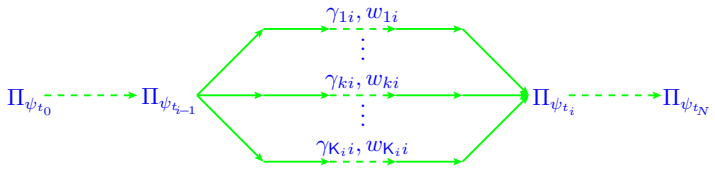
A.Torosyan
Laboratory of Information Technologies,
Joint Institute for Nuclear Research
141980 Dubna, Russia
e-mail: astghik92@mail.ru

[2] It is assumed, that the Hermicity and semi-positivity of the density matrix is taken into account.

Emergence of quantum structures in permutation dynamics

Vladimir V. Kornyak

Discrete model of quantum evolution. The trajectory of a quantum system is a sequence of unitary evolutions interspersed with observations. We consider a model of quantum dynamics represented by the following scheme



(1)

Here $t_0, \dots, t_N \in \mathbb{Z}$ are *times of observations* $\Pi_{\psi_{t_0}}, \dots, \Pi_{\psi_{t_N}}$ represented by *projectors*, namely, $\Pi_{\psi_{t_i}} = |\psi_{t_i}\rangle\langle\psi_{t_i}|$ is the projector that fixes $\psi_{t_i} \in \mathcal{H}$ (a Hilbert space) as the result of observation at the time t_i. The entities γ_{ki} are all possible products of $\Delta t_i = t_i - t_{i-1}$ elements of a finite *gauge group* $\mathsf{G} = \{\mathsf{g}_1, \dots, \mathsf{g}_\mathsf{M}\}$ with unitary representation U in $\mathcal{H}$. $w_{ki} \geq 0$ is the *weight* of γ_{ki}, $\sum_{k=1}^{K_i} w_{ki} = 1$. The parallel transports (gauge connections) γ_{ki} describe different ways of identification of the indistinguishable objects in the transition between the instants t_{i-1} and t_i.

Single unitary evolution as the most likely gauge connection. Standard quantum mechanics implies a single unitary evolution between measurements, i.e., for some $a \in \mathsf{G}$ the product $a^{\Delta t_i}$ has weight 1, and all other products have zero weights. The unitary evolution can be written as $U = \mathrm{U}\left(a^{\Delta t_i}\right)$, or, introducing the *Hamiltonian* $H = \mathrm{i} \ln \mathrm{U}(a)$, as $U = \mathrm{e}^{-\mathrm{i}H\Delta t_i}$. However, the fact that the gauge fields in physics are determined by the least action principle leads to the idea to study the case of general weights suggesting that a unique unitary evolution should occur as a dominant element in the set of all possible evolutions.

Selection of most likely trajectories and the principle of least action. The main problem in the study of the evolution is the search of the most probable trajectories. The *one-step transition probability* is given by the formula

$$\mathbf{P}_{\psi_{t_{i-1}} \to \psi_{t_i}} = \sum_{k=1}^{\mathsf{K}_i} w_{ki} \, \langle \varphi_{ki} | \, \Pi_{\psi_{t_i}} \, | \varphi_{ki} \rangle \,, \quad \text{where} \quad \varphi_{ki} = \mathrm{U}(\gamma_{ki}) \, \psi_{t_{i-1}} .$$

The probability of the whole trajectory can be calculated as the product

$$\mathbf{P}_{\psi_{t_0}\to\cdots\to\psi_{t_N}} = \prod_{i=1}^{N} \mathbf{P}_{\psi_{t_{i-1}}\to\psi_{t_i}}. \tag{2}$$

It is convenient to introduce the *one-step entropy* $\mathbf{S}_{\psi_{t_{i-1}}\to\psi_{t_i}} = \log \mathbf{P}_{\psi_{t_{i-1}}\to\psi_{t_i}}$ and replace the product of probabilities (2) by the *entropy of trajectory*

$$\mathbf{S}_{\psi_{t_0}\to\cdots\to\psi_{t_N}} = \sum_{i=1}^{N} \mathbf{S}_{\psi_{t_{i-1}}\to\psi_{t_i}}.$$

This is a discrete counterpart of the continuous *action* $\mathcal{S} = \int \mathcal{L} dt$.

Evolution of gauge connections. Let $\{w_m^t\}$, $1 \le m \le \mathsf{M}$ be a probabilistic distribution (i.e., $w_m^t \ge 0$ and $w_1^t + \ldots + w_{\mathsf{M}}^t = 1$) on the gauge group G at the time t. Consider a group algebra element of the form $A = w_1^1 \mathsf{g}_1 + \ldots + w_{\mathsf{M}}^1 \mathsf{g}_{\mathsf{M}}$. The time evolution of the statistics of parallel transports is described by the formula

$$w_1^t \mathsf{g}_1 + \ldots + w_{\mathsf{M}}^t \mathsf{g}_{\mathsf{M}} = A^t.$$

The uniform distribution, obviously, defines an idempotent in the group algebra: $B = \frac{1}{\mathsf{M}} (\mathsf{g}_1 + \ldots + \mathsf{g}_{\mathsf{M}}) \Longrightarrow B^t = B$. A^t tends at large time $(t \gg \operatorname{Exp} \mathsf{G})$ to the uniform distribution on the group generated by the elements with nonzero initial probabilities. For small time the evolution of statistics resembles the statistics evolution of a random walk. The behavior of the model — as is typical for quantum mechanics — depends on the choice of time intervals between measurements Δt_i.

Natural and standard representations of symmetric group. The most important and universal example of a finite group is the symmetric group S_{N}. We use here the N-dimensional *natural* and $(\mathsf{N}-1)$-dimensional irreducible *standard* representations of S_{N} to study the above model. For our purposes it is sufficient to consider the constructive version of natural representation that is obtained by replacing the complex space $\mathbb{C}^{\mathsf{N}}$ by the module $\mathbb{N}^{\mathsf{N}}$, where $\mathbb{N} = \{0, 1, \ldots\}$. Then, the constructive version of standard representation is obtained by the projection. As a test case, we consider the *quantum Zeno effect* for these representations within the framework of our approach.

Energy of permutation. Planck's formula $E = h\nu$ associates the energy with the periods of underlying microscopic processes. By analogy, we define the *"energy"* of a permutation p as its frequency: $\varepsilon_p = \dfrac{1}{\operatorname{ord} p}$. Of course, this definition is only approximate because the unitary operator associated with the permutation contains many frequencies: they are inverses of the lengths $\ell_1, \ldots, \ell_K$ of disjoint cycles that constitute the permutation and the period of the permutation is equal to $\operatorname{ord} p = \operatorname{lcm}(\ell_1, \ldots, \ell_K)$. However, the definition is useful for classifying the elements of permutation groups from a "physical point of view". Moreover, the permutations of the dominant conjugacy classes usually contain a small number of different frequencies and the definition becomes "almost exact".

Example. Computations with S_{50}. FIGURE 1 shows the "energy spectrum" for the dominant classes of the group S_{50} in the range of periods $[1..300]$. The numerical characteristics of the group are: $|\mathsf{S}_{50}| = 50! \approx 3.04 \times 10^{64}$; the number of conjugacy classes is equal to 204226; the maximum period is equal to 180180. The size of the largest class is approximately equal to 6.2×10^{62}, and its period is 49. Data for

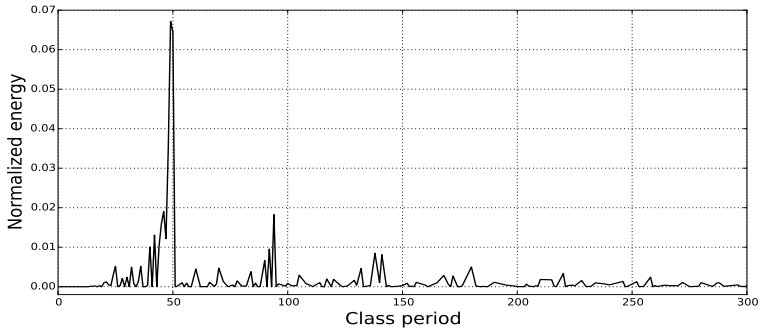


FIGURE 1. Energy spectrum of S_{50}. Exact computation.

the graph were obtained by exact calculations.

Monte Carlo approach. The study of the model discussed here by analytical methods seems to be problematic. A reasonable approach is to use Monte Carlo methods, which in their spirit are perfectly adequate to the *"irreducible quantum randomness"* (John von Neumann[1]). FIGURE 2 reproduces data obtained from the

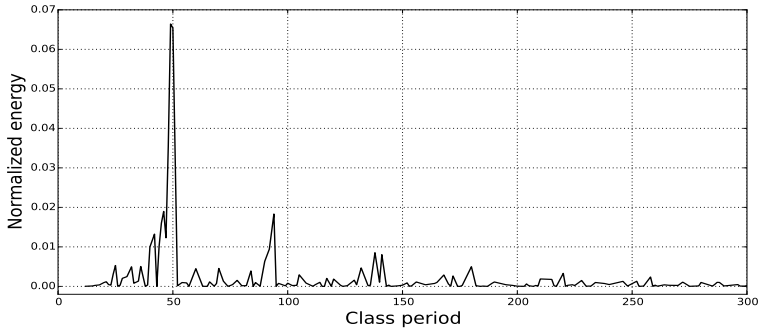


FIGURE 2. Energy spectrum of S_{50}. Randomly generated 10^6 permutations.

[1]Both the idea of the irreducibility of quantum randomness and Monte Carlo methods were actively promoted by John von Neumann.

Monte Carlo simulation. In this task — which took about 10 sec on 3.3 GHz Intel Core i3-2120 CPU — we have generated 10^6 elements of the group S_{50}. Comparing the figures, we see that the Monte Carlo simulation reproduces the general features of the distribution rather satisfactory taking into account that the ratio of the size of the generated sample to the size of the group is only about 10^{-58}.

References

[1] Kornyak V.V. *Combinatorial Approach to Modeling Quantum Systems.* EPJ Web of Conferences 108, 01007 (2016), DOI:10.1051/epjconf/201610801007; available at `http://arxiv.org/abs/1507.08896`

[2] Kornyak V.V. *Discrete dynamical models: combinatorics, statistics and continuum approximations.* Mathematical Modelling and Geometry, 2015, **3**, No 1, pp. 1–24; available at `http://arxiv.org/abs/1501.07356`

Vladimir V. Kornyak
Laboratory of Information Technologies
Joint Institute for Nuclear Research
Dubna, Russia, 141980
e-mail: `kornyak@jinr.ru`

Construction of automorphisms of 3-ary necklaces

I. A. Krepkiy

The concept of the sandpile group of graph will be described. Connection between necklaces and irreducible polynomials over the finite fields will be presented. The construction of group of automorphisms of necklaces based on this connection will be described. A direct approach for calculation of this group implies a calculation of minimal polynomials for elements of finite fields. An idea for modification of algorithm that will allow to avoid a calculation of minimal polynomials will be considered. Hypothesis about relation between automorphism groups of 3-ary necklaces and the sandpile groups of 3-ary de Bruijn graphs will be stated.

References

[1] Alexander E. Holroyd, Lionel Levine, Karola Meszaros, Yuval Peres, James Propp, David B. Wilson, *Chip-Firing and Rotor-Routing on Directed Graphs*, available at `http://arxiv.org/abs/0801.3306`.

[2] S.W. Golomb, *Irreducible polynomials, synchronization codes, primitive necklaces, and the cyclotomic algebra*, Univ. of North Carolina Monograph Series in Probability and Statistics (No. 4) (1967), pp. 358–370

[3] S. V. Duzhin, D. V. Pasechnik, *Groups acting on necklaces and sandpile groups*, Zap. Nauchn. Sem. POMI, 2014, 421, 81–93

I. A. Krepkiy
Mechanics and Mathematics Department
Saint Petersburg State University

e-mail: `feb418@gmail.com`

MathPartner one year later

Gennadi Malaschonok

Abstract. The report focuses on the changes that have occurred over the last year in the cloud mathematical project MathPartner. This includes Boolean functions, functions of set theory, additional special functions, new 3D graphics, new parallel interface, and significantly improved sidebar in the user interface.

One of the main fitures of Math Partner is the environment of working place. The definition of any mathematical object involves the definition of some environment, that is, the algebraic structure which contains this object. This structure is defined by numeric sets, algebraic operations and variable names. Examples: $R64[],\ R[x],\ Q[x,y]$ or $R_{max,+}[t]$.

However, Boolean algebra is available to the user in any environment. Operations of conjunction, disjunction, and negation are performed on the variables «true» and «false». But there are no special characters for «true» and «false». Zero is considered to be a «false» and any not zero element is the «true». All operations of comparison of mathematical objects return values 0 and 1 with a numerical type $Z64$.

These values can be used in the Boolean operations or in the branching statements ($for,\ if-else,\ while-do$) of Matnpar language. Boolean operations may be applied to matrices. In this case conjunction and disjunction operations do similar to the matrix * and + operations. So it is very easy to calculate the composition of relations as a conjunction of the relevant matrices.

The user is now available algebra of subsets of real numbers with operations: $\cup$ (union of sets), $\cap$ (intersection of sets), $\setminus$ (set minus), $\triangle$ (set symmetric minus), s' (complement to the set s).

They can use notations for such subsets: $\backslash(a,b\backslash)$ – open interval, $\backslash[a,b\backslash]$ – closed interval, $\backslash(a,b\backslash]$ – half-open interval, $\backslash[a,b\backslash)$ – half-closed interval, $\emptyset$ – empty set.

The list special functions, with which you can work, both in symbolic and numeric form additionally includes several new functions.

Bessel functions:

$$BesselJ(n,x) \text{ and } BesselY(n,x).$$

Legendre polynomials:

$$P(n,x) = 1/(2n)\sum_{k=0}^{n}\binom{n}{k}^2 (x-1)^{n-k}(x+1)^k.$$

Associated Legendre polynomials:

$$P(n,m,x) = P_n^m(x) = (-1)^m(1-x^2)^{m/2}(d/dx^m)P(n,x).$$

Spherical Harmonic:

$$Y(n,m,\theta,\phi) = [P(n,m,\cos(\theta))\cos(m\phi), P(n,m,\cos(\theta))\sin(m\phi)].$$

which is solution of Laplace's equation in spherical coordinates.
Spherical function:

$$Yr(n,m,r,\theta,\phi) = r^n Y(n,m,\theta,\phi)$$

Spherical Harmonic in Cartesian coordinates:

$$YCart(n,m,x,y,z) = Y(n,m,\theta,\phi)$$

for $r = \sqrt{(x^2+y^2+z^2)}$, $z = r\cos(\theta)$, $x = r\sin(\theta)\cos(\phi)$, $y = r\sin(\theta)\sin(\phi)$
Spherical function in Cartesian coordinates:

$$YrCart(n,m,x,y,z) = r^n YCart(n,m,x,y,z)$$

In the report, we discuss olso the new opportunities of constructing images of surfaces, a new parallel interface, and significantly improved sidebar in user interface.

New address of the mirror with Math Partner see at [4]. Here we have today 16G memory and 16 kernels.

References

[1] Malaschonok G.I. *Computer mathematics for computational network* // Tambov University Reports. Series Natural and Technical Sciences. Tambov, 2010. V. 15. Issue 1. P. 322-327. (in Russian).

[2] Malaschonok G.I. *Language Guide "Mathpar"* .Tambov: Publishing House of TSU, 2013, 133pp.

[3] Malaschonok G.I. *Project of Parallel Computer Algebra* // Tambov University Reports. Series Natural and Technical Sciences. Tambov, 2010. V. 15. Issue 1. P. 1724-1729.

[4] http://mathpar.cloud.unihub.ru

Gennadi Malaschonok
Laboratury for algebraic computations
Tambov State University
Tambov, Russia
e-mail: malaschonok@ya.ru

On M.N. Lagutinski method for integration of ordinary differential equations

Mikhail D. Malykh.

Researches of M. N. Lagutinski on the theory of integration of the differential equations were interrupted with his tragic death in 1915; here they are considered from viewpoint of modern computer algebra.

For a ring A with differentiation D and basis

$$B = \{\phi_1, \phi_2, \dots\}$$

the sequence of determinants is entered, they are called further as Lagutinski determinants. The notion of rational integral is entered without assumption about integrity of the ring. If one of Lagutinski determinants is equal to zero, the rational integral exists, moreover, we can always calculate this integral. The converse is proved at additional assumptions concerning the ring R.

The differentiation D will be called as contracting differentiation if there is a basis in which

$$D\phi_i = c_i\phi_i + o(\phi_i), \quad c_i \in c(R).$$

For the differentiation it is possible to write out simple necessary criterion of existence of integrals: among indexes c_i there are equal.

Possibilities of Lagutinski method are illustrated with simple examples in a ring $Q[x, y]$ and $Q[x, y, z]$, including the question about integrals of Brio and Bouquet differentiation. M. Singer's theorem opens a way to application of this method for integration of the differential equations in quadratures.

References

1. *Lagutinski M.N.* Prilozhenie poljarnyh operacij k integrirovaniju obyknovennyh differencial'nyh uravnenij v konechnom vide [The application of polar operations to integration of the ordinary differential equations

in finite terms]. // Soobshh. Har'kov. matem. obshh. 2 ser. 1911. T. 12. P. 111-243. (in Russian)

2. *Lagutinski M.N.* O nekotoryh polinomah i svjazi ih s algebraicheskim integrirovaniem obyknovennyh differencial'nyh algebraicheskih uravnenij [On some polynoms and their aplication for algebraic integration of ordinary differential algebraic equations]. // Soobshh. Har'kov. matem. obshh. 2 ser. 1912. T. 13. P. 200-224. (in Russian)

3. *Dobrovol'skij V.A., Strelcyn J.-M., Lokot' N.V.* Mihail Nikolaevich Lagutinskij (1871 – 1915). // Istoriko-matematicheskie issledovanija. 2001. T. 6 (41). P. 111-127. (in Russian)

Coloring triangulated surfaces

Andrei Malyutin

Abstract. A *coloring* of a singular triangulation is a partition of the set of its open simplices into classes (colors). A set of k colorings of one and the same triangulation is called a *k-layered coloring*. It turned out that $(k+m)$-dimensional manifolds and pseudomanifolds can be coded by k-layered colorings of triangulated m-dimensional manifolds. In particular, n-dimensional (pseudo)manifolds can be coded by $(n-1)$-layered colorings of sets of circles. In the first place, we investigate the coding of 3-pseudomanifolds by colorings of triangulated surfaces. Joint research with E. Fominykh and A. Vesnin.

Singular triangulations (simplicial CW-complexes). Let X be a simplicial complex and let Φ be a collection of affine homeomorphisms between some of the faces of X. Following [Mat03, p. 11], we refer to the pair (X, Φ) as to a *face identification scheme* (a *scheme*). A scheme (X, Φ) is called *inadmissible* if homeomorphisms in Φ induce a non-identity automorphism on a face in X. Otherwise, the scheme is called *admissible*. The *quotient space* $Q(X, \Phi)$ of the scheme (X, Φ) is defined as the space obtained from X by identification of faces via the homeomorphisms in Φ. The quotient space of an admissible scheme is naturally endowed with a structure of CW-complex. A CW-complex is said to be *simplicial* if it can be obtained as a quotient space of an admissible scheme.[1] The cell structure of a simplicial CW-complex is called a *singular triangulation*. A simplicial CW-complex X is said to be a *triangulated n-pseudomanifold* if removing of vertices (0-dimensional cell) from X yields an n-manifold.

Colorings. By a *coloring* of a CW-complex X we mean a map $\chi\colon X \to \mathcal{C}$ (where $\mathcal{C}$ is an arbitrary set, called the set of *colors*) which is constant on each open cell of X. We say that a coloring of a CW-complex is *trivial* if no two distinct open cells have the same color. We say that a coloring $\chi\colon X \to \mathcal{C}$ of a topological space X is *coherent* if for each pair of points x, y of the same color there exist arbitrarily small open neighbourhoods U_x and U_y of x and y with a color-preserving homeomorphism $U_x \to U_y$. The trivial coloring is coherent. We say that a coloring

Supported by RFBR grant 16-01-00609.

[1] Cf. *Δ-complexes* in [Hat02, p. 103], *side-pairing* in [Rat94, p. 435], *pseudocomplexes*, etc.

$\chi\colon S \to \mathcal{C}$ of a simplicial CW-complex S is *regular* if for each of its colors $C \in \mathcal{C}$ there exists an integer d such that $\chi^{-1}(C)$ consists of $d+2$ open cells of dimension d.

Derived complexes. Let S be a simplicial CW-complex. Let $V \subset S$ be the set of vertices of S, let $L_2(V)$ be the link of V in the second barycentric subdivision of S, let $\lambda(V)$ be the underlying topological space of $L_2(V)$ with CW-complex structure forgotten, and let $S^\dagger = \lambda^\dagger(V)$ be $\lambda(V)$ with the structure of CW-complex induced by that of S (that is, by the intersections with the cells of S). Clearly, the CW-complex $S^\dagger$ is simplicial. We say that $S^\dagger$ is the *derived* complex of S. If S is a triangulated n-pseudomanifold, then $S^\dagger$ is an $(n-1)$-manifold. If S is a triangulated n-manifold, then $S^\dagger$ is the union of $(n-1)$-spheres. Observe that a coloring of S induces a coloring of $S^\dagger$. A coherent coloring of S induces a coherent coloring of $S^\dagger$. A trivial coloring of S induces a coherent regular coloring of $S^\dagger$. This determines a map F_n from the set of (isomorphism classes of) triangulated n-pseudomanifolds without boundary to the set of (isomorphism classes of) coherent regular colorings of triangulated $(n-1)$-manifolds without boundary.

Theorem 1.

1. *The map F_3 is injective.*
2. *The map F_3 is not surjective.*

Partial colorings. By a *partial coloring* of a CW-complex X we mean a coloring $\chi\colon X \setminus V(X) \to \mathcal{C}$, where $V(X)$ is the set of vertices of X, such that χ is constant on open cells. Let π_2 be the natural forgetting map from the set of coherent regular colorings of singular triangulations on closed surfaces to the set of coherent regular *partial* colorings of singular triangulations on closed surfaces (the map π_2 'forgets' the colors of vertices).

Theorem 2.

1. *The map π_2 is injective.*
2. *The map π_2 is not surjective.*

References

[Hat02] A. Hatcher, *Algebraic topology*, Cambridge University Press, 2002.

[Mat03] S. Matveev, *Algorithmic topology and classification of 3-manifolds*, Algorithms and Computation in Mathematics, vol. 9, Springer-Verlag, Berlin, 2003.

[Rat94] John G. Ratcliffe, *Foundations of hyperbolic manifolds*, Graduate Texts in Mathematics. vol. 149. Springer-Verlag, New York, 1994.

Andrei Malyutin
St. Petersburg Department of
Steklov Institute of Mathematics
St. Petersburg, Russia
e-mail: malyutin@pdmi.ras.ru

Provable programming of algebra: particular points, examples.

Sergei D. Meshveliani

Abstract. It is discussed an experience in provable programming of a computer algebra library with using a purely functional language with dependent types (`Agda`). There are given several examples illustrating particular points of implementing the approach of constructive mathematics.

Keywords. constructive mathematics, computer algebra, dependent types.

1. Introduction

Applying the approach of constructive mathematics [2] [1] and technology of purely functional programming with *dependent types* [5] makes it a fully adequate approach to programming computation in algebra, in mathematics. In particular [3] [4] this **1)** solves completely the problem of representing an algebraic domain depending on a dynamic parameter, **2)** makes mathematical definitions and formal proofs an explicit part of a program, a part understood by the compiler and automatically checked before the running time.

So far, we use `Agda` as a language with dependent types.

This thesis describes in examples some of interesting features of the constructive-provable approach to programming algebra.

2. Example with termination proof

The *type checker* needs to verify termination for each function given in the program, in order to follow constructive mathematics. The main tool for this is finding an argument which is decreased in a certain syntactic ordering when the function is applied recursively. For example, apply the unary coding for natural numbers, with the constructor `suc` for the next number. Consider the program

This work is supported by the FANO project of the Russian Academy of Sciences, the project registration No AAAA-A16-116021760039-0.

```
0       + n = n
(suc m) + n = suc (m + n)
```

for summing natural numbers. The type checker decides that it is terminating, because in the second line the left argument term 'm' for _+_ on the right hand side is syntactically smaller than the argument (suc m) on the left hand side. For more complex functions, the programmer is often supposed to help the type checker by introducing the counter expression. The counter is syntactically decreased with each recursive call. And the program needs to specify the result when the counter turns zero. Consider the example for a provable program:

for any prime natural p *find the next prime.*

In reality, we tested the sieve method. But for this paper, let us consider the simplest algorithm of searching-through. The primality notion for $\mathbb{N}$ is defined in Agda as the function

```
IsPrime :  ℕ → Set
IsPrime p =  p ≢ 1  ×  (∀ {m n} → (m * n ≡ p) → m ≡ 1 ⊎ n ≡ 1)
```

that *returns a type*. This type has a value in it when the argument p is prime. It expresses the property "if m * n ≡ p, then m ≡ 1 or n ≡ 1".

Assume the following simplest constructs. The algorithm _|?_ for deciding divisibility is defined via division with remainder. The algorithm prime? for deciding primality of n is defined by searching through all 1 < m < n with applying m |? n. The algorithm firstFactor>1 finds for each n > 1 the first m > 1 that divides n. This is done by searching through. Also it is proved the statement
n > 1 → IsPrime (firstFactor>1 n). Now, the function

```
nextPrime :  ∀ p → IsPrime p →
                   (∃ \q → p < q × IsPrime q × IsFirstPrimeAfter q p)
```

needs to return q which is the first prime after p (and the corresponding witnesses). Program it as search-through: test (prime? n) for n = 1+p, 2+p, ..., until is finds a prime. To provide a termination proof, let us give an expression for *any* prime b = bound p such that p < b. Then, add the counter expression cnt = b - n to the search loop. The counter is decreased each time when n steps from n to suc n. If a prime is not found earlier, and it turns cnt = 0, then n ≡ b occurs the needed prime. With this, the type checker verifies termination.

And for b = bound p we need any expression such that the properties p < b and IsPrime b have an easy constructive proof. Choose this:

```
bound p = let  p₀, p₁, ..., p  =  all primes up to  p
                                  -- found by repeatedly applying 'prime?'
               a =  p₀ * p₁ * ... * p
          in
          firstFactor>1 (1 + a)
```

Then, a constructive proof for that (bound p) is a prime greater than p is not difficult to provide. With this, the program is verified. But it performs in a strange way at run-time: (nextPrime 31) hangs for a very long time!

The reason for this is that the condition b - n =? 0 is very expensive to check at run-time (see the algorithm for b). Its check is needed only to provide (statically) a termination proof, it is not needed at run-time. But there is no way to explain this to Agda, and this check gets into the executable program.

Improvement: apply the Bertrand – Chebyshev estimation —
"there exists a prime between `p` and `2*p`".

The counter of `2*p - n` is compared fast to zero. But a proof for this bound is large and complex! So: one bound is easy to prove but expensive to compute at run-time, another bound is computed fast but is complex to prove. What has one to put into the `Agda` program? The way out for `(nextPrime p)` is as follows.

Search-II:

first search before `bound2 = 2*p`. If the needed `q` is found, then stop.

Otherwise, search by new from `1+p` up to `(bound p)`.

This program 1) is verified, including termination, 2) has a fast comparison in the search loop. This is because the part of 'Otherwise' will never be performed at run-time.

But what is a way out for the case when there is proved an 'expensive' bound like above, and is not even known of any better bound?

The following solution is sufficient. In Search-II, put for `bound2` an unfeasible number — such one that will never be reached in practice in the above loop (I take this solution from the message by Ulf Norell).

In fact, this trick with unfeasible bound partly replaces the Markov's principle in constructive mathematics. This principle [2] allows a proof by contradiction for a termination proof, and it cannot be implemented in `Agda` without using the `'postulate'` construct.

3. Refuting the two prejudices

Prejudice 1:
"Proof by contradiction is not possible in constructive mathematics".

In fact: *it is possible* — when the relation has a decision algorithm.
Example: in most domains in computer algebra the equality relation has a decision algorithm `_=?_`. Respectively, a program of the kind

```
case x =? y of \ { (yes x≈y) → ...;  (no x≉y) → ... }
```

actually applies the *excluded third* law to this relation.

Prejudice 2: "Programs in the verified programming tools (like `Coq, Agda`) do not provide a proof itself, instead they provide an algorithm to build a proof witness for each concrete data".

I claim: *they also provide a proof in its ordinary meaning*
(this is so in `Agda`, and I expect, the same is with Coq).

Let us illustrate this with the example of proving the statement

$$\text{for all } n\ (n \le n).$$

for natural numbers. The relation `_≤_` is defined on $\mathbb{N}$ so that a witness for it can be built only with applying the two data constructors (axioms): `z≤n` — "`0 ≤ n` for all `n`", and `s≤s` — "if `m ≤ n`, then `suc m ≤ suc n`".
(Syntax: `z≤n, s≤s` are function *names*, as they are written without blanks).

For example: `(s≤s (s≤s z≤n))` is a proof for `2 ≤ 5`.

Consider the inductive proof for the goal statement. If `n = 0`, then `0 ≤ 0` is proved by the law `z≤n`. For a nonzero, it is needed to prove `suc n ≤ suc n`. By inductive supposition, there exists a proof `p` for `n ≤ n`. And the law `s≤s` applied to `p` yields a proof for `suc n ≤ suc n`.

Write the corresponding proof in `Agda`:

```
theorem :  ∀ n → n ≤ n
theorem 0       =  z≤n
theorem (suc n) =  s≤s (theorem n)
```

For each `n : ℕ` the function `theorem` returns a value in the type `n ≤ n`, that is the corresponding witness. The second pattern applies the function `theorem` recursively. This provides a *proof in the two meanings*.
(1) At the run-time, `(theorem n)` yields a proof for `n ≤ n` for each concrete `n`.
(2) The very algorithm expression for `theorem` is a symbolic expression that presents a general proof for the statement "for all `n (n ≤ n)`".

The algorithm (program) `theorem` is a symbolic expression (term), its parts depending on a variable `n`. This term is verified by the type checker statically — before run-time. And this is the same as checking an ordinary inductive proof. Reasoning by induction corresponds to setting a recursive call to the algorithm for forming a witness.

We see that (2) provides a real generic proof for the statement, while (1) provides a witness for each concrete n. Similar it is with all proofs.

References

[1] Per Martin-Löef, *Intuitionistic Type Theory*. Bibliopolis. ISBN 88-7088-105-9, 1984.

[2] A. A. Markov, *On constructive mathematics*, In Problems of the constructive direction in mathematics. Part 2. Constructive mathematical analysis, Collection of articles, Trudy Mat. Inst. Steklov., 67, Acad. Sci. USSR, Moscow–Leningrad, 1962, 8–14.

[3] S. D. Meshveliani, *On dependent types and intuitionism in programming mathematics*, (In Russian) In electronic journal Program systems: theory and applications, 2014, Vol. 5, No 3(21), pp. 27–50, available at
`http://psta.psiras.ru/read/psta2014_3_27-50.pdf`

[4] S. D. Meshveliani, *Programming basic computer algebra in a language with dependent types*, In electronic journal Program systems: theory and applications, 2015, 6:4(27), pp. 313–340. (In Russian), available at
`http://psta.psiras.ru/read/psta2015_4_313-340.pdf`

[5] U. Norell, J. Chapman, *Dependently Typed Programming in Agda*, available at
`http://www.cse.chalmers.se/~ulfn/papers/afp08/tutorial.pdf`

Sergei D. Meshveliani
Program Systems Institute of Russian Academy of sciences, Pereslavl-Zalessky, Russia
e-mail: `mechvel@botik.ru`

Maple package generating Painleve equations

Stesik O.L.,Slavyanov S.Yu.
Sankt-Petersburg State University
e-mail:o.stesik@spbu.ru
e-mail:slav@ss2034.spb.edu

March 14, 2016

Painleve equations play an important role in theory of integrable nonlinear equations. Several books are devoted to the theory of Painleve equations [1],[2].Painleve equations can be generated via linear Heun equations [3, 4, 5].

The corresponding calculations are simple in principle however boring in realization. Our goal was to simplify the computations by using ACS Maple. Ingoing data are Heun type equations and the outgoing data are various Painleve equations. In this manner our package can cover all Heun equations and Painleve equations presented in NIST project [6].We present the structure of our package and give results of particular computations.

References

[1] S.Yu.Slavyanov, W.Lay, "Special Functions: Unified Theory based on Singularities" Oxford University Press, 2000.

[2] "Painleve Equations and related Topics", De Gruyter, 2012.

[3] S.Yu. Slavyanov J, Phys A, 1996. Vol 29 p., 7329-7335

[4] S.Yu. Slavyanov, O.L. Stesik, TMPH, v.186, N 1, 2016.

[5] A.A.Bolibruch,"The inverse problems of monodromie in analytical theory of differential equations",MCMNS, Moscow, 2009.

[6] dlfm.nist.gov

Oriented area is a perfect Morse function

Gaiane Panina

Abstract. We show that an appropriate generalization of the oriented area function is a perfect Morse function on the space of three-dimensional configurations of an equilateral polygonal linkage with odd number of edges. Therefore cyclic equilateral polygons (which appear as critical Morse points) are interpreted as independent generators of the homology groups of the (decorated) configuration space.

Gaiane Panina
SPIIRAS,
St. Petersburg State University
Russia
e-mail: gaiane-panina@rambler.ru

On the computational complexity of the three-body initial value problem

Nikolay Vasiliev and Dmitry Pavlov

Abstract. A classical example of a chaotic dynamic system is considered as a computational problem, stated in terms of computable real numbers. A hypothetical Turing machine, using an infinite tape containing initial values as an input, and a number t as a time interval, can calculate the state of the system at time t as computable real numbers. Known algorithms of numerical integration require at least $O(t)$ time to complete, even disregarding the precision loss. This work deals with the question: can there exist such a machine that solves the initial value problem for three-body system in polynomial in $\log(t)$ time? Basing on the works of Sitnikov and Alexeyev, we can deduce that a particular case of the three-body problem is so combinatorially rich that no Turing machine can find the solution in a polynomial time.

Nikolay Vasiliev
Laboratory of Representation Theory and Dynamical Systems
St. Petersburg Department of V.A. Steklov Institute of Mathematics of the RAS
St. Petersburg, Russia
e-mail: vasiliev@pdmi.ras.ru

Dmitry Pavlov
Laboratory of Ephemeris Astronomy
Institute of Applied Astronomy RAS
St. Petersburg, Russia
e-mail: dpavlov@ipa.nw.ru

Mathematical tasks with verifiable parametrized answers

Ilia Posov

Abstract. Tasks with verifiable answers are the tasks, for which a student's answer can be automatically checked based only on its statement. This means that a teacher needs only to provide a statement for a task, and does not need to provide an answer, usually even does not need to solve a task, but the task still can be automatically checked. Parametrized answer means a type of answer that contains parameters from the statement. The report discusses the implementation of such tasks with and without the use of computer algebra systems.

Ilia Posov
Faculty of arts
St. Petersburg State University
Saint-Petersburg, Russia

Faculty of Computer Science and Technology
St. Petersburg Electrotechnical University "LETI"
Saint-Petersburg, Russia
e-mail: iposov@gmail.com

Tools for supporting of competitions and Olympiads

Sergei Pozdniakov and Vasiliy Akimushkin

Abstract. Comparative experience of using of various computer tools in the International competition "Bebras", "Construct-Test-Explore" contest and the Olympiad on discrete mathematics and theoretical informatics is presented in the report.

Sergei Pozdniakov
Faculty of Computer Science and Technology
St. Petersburg Electrotechnical University "LETI"
Saint-Petersburg, Russia
e-mail: pozdnkov@gmail.com

Vasiliy Akimushkin
Mathematics and Mechanics Faculty
St. Petersburg State University,
Saint-Petersburg, Russia
e-mail: vasiliy.akimushkin@gmail.com

Generation and annihilation of apparent singularities

Slavyanov S.Yu., Satco D.A.
Sankt-Petersburg State University
e-mail:slav@ss2034.spb.edu

March 24, 2016

The problem of reconstruction of coefficients for Fuchsian equations from monodromie data had been studied by Plemelj [1]. The main result was negative. In other words the parameters of equations can not be reconstructed from monodromie data. One of the reasons lays in existence of so-called apparent singularities which correspond to appropriate parameters but for which a monodromie matrix is trivial.

Here we give some examples of recipes how to generate or annihilate apparent singularities in linear differential equations with polynomial coefficients by turning to equations for derivatives and inverse derivatives of solutions. Previously this approach was proposed by A. Ishkanyan and Suominen [2] for numerical needs.

First we give the definition of apparent singularity valid for our case. Suppose we have an equation

$$\boldsymbol{L(D,z)y(z)=\sum_{k=0}^{n}P_k(z)D^k y(z)=0} \qquad (1)$$

where $L(D,z)$ is a polynomial in two variables: D – differentation operator and z – multiplication by variable z. The corresponding degrees are n and m respectively with $m \geq n+1$. Zeros z_j of the polynomial $P_n(z)$ and the point $z=\infty$ under additional conditions (which are not discussed here) are regular (fuchsian) singularities of equation (1). In this case equation (1) is a fuchsian equation. If two fuchsian singularities coalesce we arrive to irregular singularity and to confluences equation. The latter equations also can be considered as equations with polynomial coefficients.

Among fuchsian singularities apparent singularities can be distinguished in the vicinity of which general solution $y(z)$ is a holomorphic function. We show in examples that if we have equation without apparent singularities the equation for the derivative of a solution possess apparent singularities.

Example 1. Equations of Heun class [3] can be presented as

$$\sigma(z)w''(z) + \tau(z)w'(z) + \omega(z)w(z) - hw(z) = 0, \qquad (2)$$

where $\sigma(z)$ is a polynomial of third degree or less, $\tau(z)$ is a polynomial of second degree or less and $\omega(z)$ is a polynomial of first degree or less. These polynomials depend on parameter t and accessory parameter h.

We introduce beyond the function $w(z)$ its derivative $u(z)$

$$u(z) = w'(z) \qquad (3)$$

Differentiation of (2) leads to the second order equation for $u(z)$

$$\sigma(z)u''(z) + \left(\tau_1(z) - \sigma(z)\frac{\omega'(z)}{\omega(z)-h}\right)u'(z) + (\omega_1(z) - h)u(z) - \left(\tau(z)\frac{\omega'(z)}{\omega(z)-h}\right)u(z) = 0 \qquad (4)$$

with $\tau_1(z) = \tau(z) + \sigma'(z)$, $\omega_1(z) = \omega(z) + \tau'(z)$. Equation (2) beyond singularities characteristic for initial Heun equation has an additional apparent singularity $z = q$ being the single zero of $\omega(z) - h$. It had been shown in publication [?] that equation (4) generates nonlinear integrable Painlevé equations.

Example 2. Next in complexity in comparison with Heun equation is an equation similar to (2) but with $\sigma(z) = \prod_{j=1}^{4}(z - z_j)$ - fourth degree polynomial with increase by unity degrees of other polynomials in (2).Clearly the factor $P_0(z)$ in front of $y(z)$ would be a second degree polynomial depending on two accessory parameters. The equation for $u(z) = y'(z)$ will have two apparent singularities according to the degree of mentioned polynomial.

Example 3. Suppose $P_0(z) = A(z-q)^2$ that corresponds to zero with multiplicity 2. Then we arrive after differentiating twice to a single apparent singularity but slightly

different from the introduced above. Namely, characteristic indices at apparent singularity are 1 and 3.

What is the meaning of our presentation of apparent singularity. At ordinary point of a differential equation Cauchy data can be posed and afterwards all other derivatives of a solution can be computed via differential equation. However, in the case of apparent singularity the derivative of order equal or higher than the order of the equation can be arbitrary.

Example 4. A particular Fuchsian third order equation with singularities located at the points $z_1 = 0$, $z_2 = 1$, $z_3 = t$ can be considered

$$z^2(z-1)(z-t)y'''(z) + [(3-\alpha-\beta)z(z-1)(z-t) - \theta_3 z^2(z-1) - \theta_2 z^2(z-t)]y''(z) + (\alpha-1)(\beta-1)(z-1)(z-t)y'(z) + (\kappa z + h]y(z) = 0. \quad (5)$$

The Riemann scheme for this equation

$$\begin{pmatrix} z_1 & z_2 & z_3 & \infty & z \\ 0 & 0 & 0 & a & h \\ \alpha & 1 & 1 & b & \\ \beta & 2+\theta_3 & 2+\theta_2 & c & \end{pmatrix}$$

shows the Frobenius characteristic exponents.

$$a + b + c = -\alpha - \beta - (\theta_2 + \theta_3)$$

$$ab + bc + ac = \alpha\beta + (\theta_2 + \theta_3)$$

$$abc = \kappa$$

It means that at finite singularity $z = 0$ there is one holomorphic solution and two solution which are in general not holomorphic whenever at $z = 1$ and $z = t$ there are two linearly independent holomorphic solutions and one which is not holomorphic. The system for the coefficients a, b, c can be explicitly solved Let

$$\theta_2 + \theta_3 = \gamma$$

Then a can be chosen as any root of equation

$$a^3 + a^2(\alpha + \beta + \gamma) + a(\alpha\beta + \gamma) - \kappa = 0$$

and $b = x_1$ and $c = x_2$ are taken as roots of equation

$$ax^2 + (\alpha + \beta + \gamma + a)x + \kappa = 0$$

Differentiating (5) one obtains a third order equation with additional apparent singularity.

The presented examples allow to formulate the following conjectures.

Conjecture 1. *Any linear differential equation with polynomial coefficients containing apparent singularities can be obtained as an equation for a proper derivative of a solution of a linear differential equation without apparent singularities.*

If we turn to inverse derivatives we arrive to the opposite conjecture.

Conjecture 2. *Any linear differential equation with polynomial coefficients containing apparent singularities can be transformed to equation without apparent singularities by use of inverse derivatives of a solution*

To prove this conjectures is an open question.

References

[1] J. Plemelj, Monatsch. Math. Phys., 19:1, pp.211-246, 1908.

[2] A. Ishkanyan, K.A. Suominen, J. Phys. A, v.36, p.681, 2003.

[3] S.Yu.Slavyanov, W.Lay, "Special Functions: Unified Theory based on Singularities" Oxford University Press, 2000.

[4] S.Yu. Slavyanov, O.L. Stesik, TMPH, v.186, p.119, 2016.

On Cubic Hypersurfaces with Involutions

Alexandr V. Seliverstov

Abstract. Multidimensional generalizations of the Weierstrass normal form are considered, depending on the Waring decomposition. The straightforward generalization exists for Fermat-type cubic forms, but does not exist for the general cubic forms in four variables. On the other hand, if a cubic form has a sufficiently small rank, then the corresponding hypersurface is invariant under a nonidentity birational involution of the complex projective space. The involution can be calculated in terms of radicals.

Let us focus on cubic hypersurfaces that are invariant under a nonidentity birational involution of the complex projective space. Throughout the paper all coefficients are denoted by small Greek letters. A form means a homogeneous polynomial over the field of complex numbers. A hypersurface means a projective variety of codimension one. A hypersurface given by the form f is smooth if its gradient ∇f is nonzero outside of the origin; otherwise it is singular. Two forms f and g are equivalent to each other if there exists a nondegenerate linear transformation J such that $f(\mathbf{x}) = g(J\mathbf{x})$. A cubic form in three variables is equivalent to the Weierstrass normal form $y_0^2y_2 + y_1^3 + \alpha y_1y_2^2 + \beta y_2^3$. It is invariant under the linear involution $(y_0, y_1, y_2) \mapsto (-y_0, y_1, y_2)$. The rank of a form f of degree d is the minimal number of linear forms needed to represent f as a sum of d-powers. This sum is known as the Waring decomposition. For example, each ternary cubic form can be decomposed as the sum of five cubes (Sylvester Pentahedral Theorem). The next example shows the relationship between the Weierstrass normal form and the Waring decomposition. Let us consider the linear transformation given by two equations $x_0 = \frac{1}{6}y_1 + y_0$ and $x_1 = \frac{1}{6}y_1 - y_0$. Then $x_0^3 + x_1^3 = y_0^2y_1 + \frac{1}{108}y_1^3$.

Theorem 1. *The general cubic form in four variables is not equivalent to any form of the type* $y_0^2y_3 + g(y_1, y_2, y_3)$.

Proof. Let us suppose the general cubic form f in four variables is equivalent to a form of the type $y_0^2y_3 + g(y_1, y_2, y_3)$. One can assume that the surface given by the equation $f = 0$ is smooth. The requirement of smoothness does not reduce the dimension of the set of forms. The curve given by the equation $g = 0$ is smooth. Thus, the form g is equivalent to the second normal form $g = z_1^3+z_2^3+z_3^3-3\lambda z_1z_2z_3$

with replacement of three variables y_1, y_2, and y_3 by linear forms in three variables z_1, z_2, and z_3. Then $f = y_0^2(\rho_1^2 z_1 + \rho_2^2 z_2 + \rho_3^2 z_3) + z_1^3 + z_2^3 + z_3^3 - 3\lambda z_1 z_2 z_3$, where at least one of the coefficients ρ_1, ρ_2, or ρ_3 is nonzero. Otherwise, the form would not depend on the variable y_0; therefore the point with homogeneous coordinates $[1 : 0 : 0 : 0]$ would be a singular point of the surface. One can assume that $\rho_3 \neq 0$. Replacing the variable $y_0 = \rho_3 z_0$ yields an equivalent cubic form of the type $f = z_0^2(\mu_1 z_1 + \mu_2 z_2 + z_3) + z_1^3 + z_2^3 + z_3^3 - 3\lambda z_1 z_2 z_3$. So a cubic form in four variables with at most 20 monomials is defined by a matrix with 16 entries and three parameters λ, μ_1, and μ_2. Mapping of the pair consisting of the form $f(\mathbf{x})$ and the matrix J to another form $f(J\mathbf{x})$ obtained by the linear transformation of coordinates defines a regular surjection from the 19-dimensional affine complex space onto an open set of the 20-dimensional complex space. There is a small polydisc such that the map is bijective. This contradicts Brouwer's theorem. □

Theorem 2. *Given the cubic form $f = x_0^3 + \cdots + x_n^3 + (\alpha_0 x_0 + \cdots + \alpha_n x_n)^3$ in at least three variables $x_0, \ldots, x_n$. There exists a transformation of coordinates such that f is equal to the rational function $y_0^2 y_n + g(y_1, \ldots, y_n)$ in the complement of a hyperplane given by the linear equation $y_n = 0$ in at most three variables x_0, x_1, and x_n. The transformation is the identity map for all coordinates except three; moreover it can be calculated in terms of radicals.*

Proof. Let us consider the linear form $\ell = \alpha_0 x_0 + \cdots + \alpha_n x_n$ and the Hessian matrix H, whose entries are equal to $\frac{\partial^2 f}{\partial x_i \partial x_j}$. The matrix H is equal to the sum of the diagonal matrix $\operatorname{diag}(6x_0, \ldots, 6x_n)$ and the matrix with entries $6\alpha_i\alpha_j\ell$. Let us consider a point $\mathbf{u}$ with coordinates $u_i = 0$ for all $2 \leq i \leq n-1$ such that it is not the origin, and both $\ell(\mathbf{u})$ and $f(\mathbf{u})$ vanish. Its coordinates can be calculated in terms of radicals. The rank of the matrix $H(\mathbf{u})$ is at most three and does not increase under a linear transformation of the coordinates. Let us consider the quadratic form $h = u_0 x_0^2 + u_1 x_1^2 + u_n x_n^2$ with the matrix $H(\mathbf{u})$. It vanishes at the point $\mathbf{u}$ because $h(\mathbf{u}) = f(\mathbf{u}) = 0$; likewise both gradients $\nabla f(\mathbf{u})$ and $\nabla h(\mathbf{u})$ are collinear and nonzero. Both quadric $h = 0$ and cubic $f = 0$ have a common tangent hyperplane with defining linear form $z_n = u_0^2 x_0 + u_1^2 x_1 + u_n^2 x_n$ up to a nonzero factor. A linear subspace of codimension two lies on the quadric $h = 0$. It is defined by two linear equations $z_1 = z_n = 0$ for some linear form z_1 in three variables x_0, x_1, and x_n. Let us choose an independent linear form $z_0(x_0, x_1, x_n)$ such that $z_0(u_0, u_1, u_n) \neq 0$. Let us set at last $z_i = x_i$ for all indices $2 \leq i \leq n-1$. The linear transformation is nondegenerate. Thus, the set $\{z_i\}$ is a basis for the dual space. If both u_0 and u_n are nonzero, then one can choose the forms $z_0 = x_0$ and $z_1 = u_1 x_0 - u_0 x_1$.

The restriction of h to the subspace vanishes identically. Thus, the cubic form is equal to $f = \rho_0^2 z_0^2 z_n + 2\rho_0 z_0(\rho_1 z_1 + \cdots + \rho_n z_n) z_n + 2\tau\rho_0 z_0 z_1^2 + s(z_1, \ldots, z_n)$, where ρ_k and τ are complex numbers. As the cubic is not a cone, $\rho_0 \neq 0$.

In case $\tau = 0$, the cubic form can be transformed to $f = y_0^2 y_n + g(y_1, \ldots, y_n)$, where $y_0 = \rho_0 z_0 + \rho_1 z_1 + \cdots + \rho_n z_n$ and for all indices $i \neq 0$ we set $y_i = z_i$.

In case $\tau \neq 0$, if $z_n \neq 0$, then f is equal to

$$f = \rho_0^2 z_0^2 z_n + 2\rho_0 z_0 \left(\rho_1 z_1 + \cdots + \rho_n z_n + \tau \frac{z_1^2}{z_n}\right) z_n + s(z_1, \ldots, z_n).$$

Let us set $y_0 = \rho_0 z_0 + \rho_1 z_1 + \cdots + \rho_n z_n + \tau \frac{z_1^2}{z_n}$ and for all indices $i \neq 0$ we set $y_i = z_i$. Then $f = y_0^2 y_n + g(y_1, \ldots, y_n)$, but y_0 and g are rational functions. Their denominators are powers of a linear form $u_0^2 x_0 + u_1^2 x_1 + u_n^2 x_n$. □

Remark. There are $\frac{1}{6}(n^3 - n)$ choices of three coordinates x_i, x_j, and x_k instead of x_0, x_1, and x_n.

Theorem 3. *Given the cubic form $f = x_0^3 + \cdots + x_n^3 + (\alpha_0 x_0 + \cdots + \alpha_n x_n)^3$ in at least three variables $x_0, \ldots, x_n$. The corresponding cubic hypersurface is invariant under a nonidentity birational involution of the ambient projective space.*

Proof. According to Theorem 2, there is a birational map φ from the cubic hypersurface $f = 0$ to a hypersurface, which is invariant under the action of the linear involution $[y_0 : y_1 : \cdots : y_n] \mapsto [-y_0 : y_1 : \cdots : y_n]$. The composition of the map φ, the involution, and φ^{-1} yields a sought involution. □

Remark. All cubic surfaces are rational. Thus, a large set of birational involutions exists for any cubic surface. If there is a regular involution of an open set of the surface with a unique singular point, then the point is fixed under the involution. In this way, one can either localize the singular point, or verify smoothness of a cubic surface having at most one singular point. The requirement for uniqueness of the singular point is significant. Otherwise, two singular points can be mapped one into another under the involution.

The following theorem improves the result from [1] in case of cubic hypersurfaces. The homogeneous coordinates of $(-1, 1)$-points are equal to $[\pm 1 : \cdots : \pm 1 : 1]$ up to a common nonzero factor.

Theorem 4. *Given the cubic form $f = x_0^3 + \cdots + x_n^3 + (\alpha_0 x_0 + \cdots + \alpha_n x_n)^3$ in at least three variables $x_0, \ldots, x_n$, where all the coefficients α_k are nonzero. There exists a one-to-one correspondence between singular points of the cubic hypersurface $f = 0$ and $(-1, 1)$-points belonging to the hyperplane defined by the linear form $h = \beta_0 y_0 + \cdots + \beta_n y_n + y_{n+1}$ in $n + 2$ variables with the coefficients $\beta_k = \sqrt{-\alpha_k^3}$.*

Proof. Let us consider the cubic form $g = \beta_0 y_0^3 + \cdots + \beta_n y_n^3 + y_{n+1}^3$. Since all the coefficients β_k are nonzero, the hypersurface $g = 0$ is smooth. Its hyperplane section is projectively equivalent to the hypersurface $f = 0$. If both forms h and g vanish simultaneously at a $(-1, 1)$-point, then the hyperplane is tangent to the hypersurface $g = 0$ at this point. Thus, the section is singular.

At a singular point of the section, the hyperplane $h = 0$ coincides with the tangent hyperplane to the hypersurface $f = 0$. Since all the coefficients β_k are nonzero, both gradients ∇h and ∇g can be collinear only at the points whose coordinates satisfy the system of the equations $x_k^2 = x_j^2$ for all indices k and j. All the points are $(-1, 1)$-points. □

In accordance with the Alexander–Hirschowitz theorem [2], the rank of the general cubic form in four variables is equal to five. It is exactly one more than the number of variables. If the Waring decomposition is known, then Theorem 4 solves the system for cubic surface by means of an auxiliary combinatorial task that is equivalent to the set partition problem. Unfortunately, it is hard to find a $(-1, 1)$-point belonging to the hyperplane in high dimensions [3]. On the other hand, one can find $(-1, 1)$-points belonging to the hyperplane given by a linear form with integer coefficients near zero, using dynamic programming.

References

[1] I. V. Latkin and A. V. Seliverstov, Computational complexity of fragments of the theory of complex numbers, *Bulletin of University of Karaganda. Ser. Mathematics* **77** (2015) 47–55 [in Russian].

[2] M. C. Brambilla and G. Ottaviani, On the Alexander–Hirschowitz theorem, *J. Pure App. Algebra* **212** (2008) 1229–1251. DOI: 10.1016/j.jpaa.2007.09.014

[3] S. Margulies, S. Onn, and D. V. Pasechnik, On the complexity of Hilbert refutations for partition, *J. Symbolic Comput.* **66** (2015) 70–83. DOI: 10.1016/j.jsc.2013.06.005

Alexandr V. Seliverstov
Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute) Moscow, Russia
e-mail: slvstv@iitp.ru

Hankel polynomials in the interpolation problems

Alexei Yu. Uteshev and Ivan Baravy

Abstract. We treat the univariate interpolation problem $\{f(x_j) = y_j\}_{j=1}^N$ for polynomial and rational functions. Developing the approach by C.Jacobi, we represent the interpolants by virtue of the Hankel polynomials generated by the sequences $\{\sum_{j=1}^N x_j^k y_j / W'(x_j)\}_{k\in\mathbb{N}}$ and $\{\sum_{j=1}^N x_j^k / (y_j W'(x_j))\}_{k\in\mathbb{N}}$; here $W(x) = \prod_{j=1}^N (x - x_j)$. The obtained results are applied for the error correction problem, i.e. the problem of reconstructing the polynomial from a redundant set of its values some of which are probably erroneous. The problem of evaluation of the resultant of polynomials $p(x)$ and $q(x)$ from the set of their values is also tackled within the framework of this approach.

Introduction

Compared with the polynomial interpolation problem which is known for numerous practical applications since its very beginning in the 17th century, rational interpolation problem has received its real life application points in the late 20th century. In case of infinite fields, these are Control Theory (recovering of the transfer function from the frequency responses) while for the case of finite fields these are Error Correcting Codes (Berlekamp-Welch agorithm).

We solve the polynomial and the rational interpolation problems $\{f(x_j) = y_j\}_{j=1}^N$ with the aid of the Hankel polynomials [1], i.e. polynomials in x with the following representation in the determinantal form

$$\mathcal{H}_k(x;\{c\}) = \begin{vmatrix} c_0 & c_1 & c_2 & \dots & c_k \\ c_1 & c_2 & c_3 & \dots & c_{k+1} \\ \vdots & & & \ddots & \vdots \\ c_{k-1} & c_k & c_{k+1} & \dots & c_{2k-1} \\ 1 & x & x^2 & \dots & x^k \end{vmatrix} = H_k(\{c\})x^k + h_{k1}x^{k-1} + \dots \quad (1)$$

Here the generators $c_0, c_1, \dots c_{2k-1}$ of the polynomial are the elements of some (finite or infinite) field and $H_k(\{c\}) = \det[c_{i+j-2}]_{i,j=1}^k$ is the Hankel determinant.

1. Rational interpolation

We are looking for the rational interpolant in the form $f(x) = p(x)/q(x)$ where $p(x)$ and $q(x)$ are polynomials, $\deg p \le n, \deg g \le m$ and $N = n + m + 1$. The interpolation problem is not always solvable.

Theorem 1. *Let* $\{y_j \neq 0\}_{j=1}^N$, *and* $W(x) = \prod_{j=1}^N (x - x_j)$. *Compute the values*

$$\tau_k = \sum_{j=1}^N y_j \frac{x_j^k}{W'(x_j)} \quad \text{for } k \in \{0, \dots, 2m\} \tag{2}$$

and

$$\widetilde{\tau}_k = \sum_{j=1}^N \frac{1}{y_j} \frac{x_j^k}{W'(x_j)} \quad \text{for } k \in \{0, \dots, 2n-2\}, \tag{3}$$

and generate the corresponding Hankel polynomials $\mathcal{H}_m(x; \{\tau\})$ *and* $\mathcal{H}_n(x; \{\widetilde{\tau}\})$. *If* $H_n(\{\widetilde{\tau}\}) \neq 0$ *and* $\{\mathcal{H}_m(x_j; \{\tau\}) \neq 0\}_{j=1}^N$ *then there exists a unique solution to rational interpolation problem with* $f(x) \equiv p(x)/q(x)$ *where* $\deg p(x) = n, \deg q(x) \le m = N - n - 1$. *It can be expressed as:*

$$p(x) = H_{m+1}(\{\tau\})\mathcal{H}_n(x; \{\widetilde{\tau}\}) \ , \ q(x) = H_n(\{\widetilde{\tau}\})\mathcal{H}_m(x; \{\tau\}) \, . \tag{4}$$

2. Polynomial interpolation

For $m = 0$, Theorem 1 yields a solution to the polynomial interpolation problem. We apply this result to the error correction in the data set, i.e. for the case where the data set $\{(x_j, y_j)\}_{j=1}^N$ is redundant for the interpolant computation ($n < N-1$) but probably contains erroneous values.

Theorem 2. *Let* $E \in \{2, 3, \dots, \lfloor N/2 \rfloor - 1\}$. *Let polynomial* $p(x)$ *be of a degree* $n < N - 2E$. *Let the data set* $\{(x_j, y_j)\}_{j=1}^N$ *satisfy the conditions*

(a) $y_j \neq 0$ *for* $j \in \{1, \dots, N\}$,
(b) $y_j = p(x_j)$ *for* $j \in \{1, \dots, N\} \setminus \{e_1, \dots, e_E\}$,
(c) $\widehat{y}_{e_s} = p(x_{e_s}) \neq y_{e_s}$ *and* $\widehat{y}_{e_s} \neq 0$ *for* $s \in \{1, \dots, E\}$.

Then

$$\mathcal{H}_E(x; \{\tau\}) \equiv \frac{\prod_{s=1}^E (y_{e_s} - \widehat{y}_{e_s}) \prod_{1 \le s < t \le E} (x_{e_t} - x_{e_s})^2}{\prod_{s=1}^E W'(x_{e_s})} \prod_{s=1}^E (x - x_{e_s}) \, . \tag{5}$$

This result means that the sequence of the Hankel polynomials $\{H_k(x; \{\tau\})\}_{k=1}^{N-1}$ contains at least one polynomial with the zero set coinciding with the set of all the nodes x_j corresponding to corrupted values y_j in the data set initially generated by the polynomial $p(x)$. In terms of Coding Theory, polynomial $\mathcal{H}_E(x; \{\tau\})$ can be referred to as the *error locator polynomial*.

3. Recurrence relation for the Hankel determinants

The problem of computation of the parameter dependent polynomials is not a trivial one. Fortunately, the following result concerning the Hankel polynomials was retrieved by the present authors in the 19th century papers:

Theorem 3 (Jacobi, Joachimsthal). *For any generating sequence $\{c\}$, any three consecutive Hankel polynomials $\mathcal{H}_{k-2}(x), \mathcal{H}_{k-1}(x), \mathcal{H}_k(x)$ are linked by the following identity*

$$H_k^2\mathcal{H}_{k-2}(x)+(H_k h_{k-1,1} - H_{k-1}h_{k1} - H_k H_{k-1}x)\,\mathcal{H}_{k-1}(x)+H_{k-1}^2\mathcal{H}_k(x) \equiv 0\,. \quad (6)$$

If $H_{k-1} \neq 0$, formula (6) permits one to organize the computation of $\mathcal{H}_k(x)$ recursively in k. We also suggest a result to deal with an exceptional case when $H_{k-1} = 0$. These results can be applied for an effective computation of the whole family of rational interpolants for all the possible combinations for the orders of numerator and denominator satisfying the restriction $\deg p(x) + \deg q(x) = N - 1$.

4. Resultant interpolation

It turns out that the numerical factors in formulas (4) are related to the resultant $\mathcal{R}(p(x), q(x))$ of the polynomials from the numerator and the denominator of the rational function.

Theorem 4. *Let $\{p(x_j) \neq 0, q(x_j) \neq 0\}_{j=1}^N$ and $\deg p(x) = n$. Let the values (2) and (3) be calculated for $\{y_j = p(x_j)/q(x_j)\}_{j=1}^N$. The following equalities are valid*

$$H_n(\{\widetilde{\tau}\}) = \frac{(-1)^{mn+n(n+1)/2}p_0}{\prod_{j=1}^N p(x_j)}\mathcal{R}(p(x), q(x))\,,$$

$$H_{m+1}(\{\tau\}) = \frac{(-1)^{m(m+1)/2}p_0}{\prod_{j=1}^N q(x_j)}\mathcal{R}(p(x), q(x))\,;$$

here p_0 stands for the leading coefficient of $p(x)$.

This result might be useful for the problems where the canonical representations for polynomials $p(x)$ and $q(x)$ are a priori unknown, like, for instance, the problem of establishing the existence of a common eigenvalue for the given pair of matrices.

Conclusion

The problem of interpolation of the given data set $\{y_j\}_{j=1}^N$ by the set $\{p(x_j)\}_{j=1}^N$ of values of a polynomial of a degree $n \ll N$, if treated in $\mathbb{R}$, is frequently tackled by the least squares method. However the LSM is sensitive to the occurrence of *outliers*, i.e. incidental systematic errors. On the basis of the result of Theorem 2, we hope to formulate criteria for distinguishing systematic errors from the non-systematic ones in the "noisy" data set.

The extension of the results of Sections 1–4 to the multivariate case is also a subject of future investigation.

References

[1] P. Henrici, *Applied and Computational Complex Analysis*. V. 1. Wiley, NY, 1974.

[2] A.Yu. Uteshev, and I. Baravy, *Solution of Interpolation Problems via the Hankel Polynomial Construction*, preprint (2016), available at `http://arxiv.org/abs/1603.08752`.

Alexei Yu. Uteshev
e-mail: `alexeiuteshev@gmail.com`

Ivan Baravy
e-mail: `ivan.borovoj@gmail.com`
St.Petersburg State University
Faculty Applied Mathematics
St.Petersburg, Russia

On the ring of local unitary invariants for mixed $X-$states of two qubits

V.Gerdt, A.Khvedelidze and Yu. Palii

• **Two-qubit quantum states** • Generally, the state of a two-qubit quantum system is described by the density matrix which has the following general structure

$$\varrho = \frac{1}{4}\left[I_2 \otimes I_2 + \sum_{i=1}^{3} a_i \sigma_i \otimes I_2 + \sum_{i=1}^{3} b_i I_2 \otimes \sigma_i + \sum_{i,j=1}^{3} c_{ij} \sigma_i \otimes \sigma_j \right] . \qquad (1)$$

where $\sigma_1, sigma_2, sigma_3$ are the Pauli matrices, and I_2 is the unit 2×2 matrix. The 15 real parameters a_i, b_i and c_{ij} , $i, j = 1, 2, 3$, define the space

$$W := \{ (a_i, b_j, c_{kl}) \in \mathbb{R}^{15} \mid i, j, k, l = 1, 2, 3 \} , \qquad (2)$$

and the corresponding $SU(2) \times SU(2)-$invariant polynomials accumulate all relevant information on the quantum two-qubit entanglement.

The 15-dimensional space (2) is subject to the physical constraints coming from the semipositivity condition imposed on the density matrix:

$$\varrho \geq 0 . \qquad (3)$$

Explicitly, the semipositivity condition (3) reads as a set of polynomial inequalities in the fifteen variables a_i, b_i and c_{ij} , and thereby determines a semialgebraic variety of (2) (see, e.g., [11] and references therein).

• **The research object** • We study the special 7-dimensional subspace of (2), the space of so-called $X-$states [1]. These states got such name due to the visual similarity of the density matrix, whose non-zero entries lie only on the main and minor (secondary) diagonals, with the Latin letter "X":

$$\varrho_X := \begin{pmatrix} \varrho_{11} & 0 & 0 & \varrho_{14} \\ 0 & \varrho_{22} & \varrho_{23} & 0 \\ 0 & \varrho_{32} & \varrho_{33} & 0 \\ \varrho_{41} & 0 & 0 & \varrho_{44} \end{pmatrix} . \qquad (4)$$

In (4) the diagonal entries are real numbers, while elements of the minor diagonal are pairwise complex conjugated, $\varrho_{14} = \overline{\varrho}_{14}$ and $\varrho_{23} = \overline{\varrho}_{32}$. Our interest to this subspace of (2) is due to fact that many well-known states, e.g. the Bell

states [2], Werner states [3], isotropic states [4] and maximally entangled mixed states [5, 6] are particular subsets of the $X-$states. Since their introduction in [1], many interesting properties of $X-$states have been established. Particularly, it was shown that for a fixed set of eigenvalues the states of maximal concurrence, negativity or relative entropy of entanglement are the $X-$states. [1]

• **Main results** • Here we pose the question about the algebraic structure of the local unitary polynomial invariants algebra corresponding to the $X-$states. More precisely, the fate of generic $SU(2) \times SU(2)$-invariant polynomial ring of 2-qubits [8]–[11] under the restriction of the total 2-qubit state space to its subspace

$$W_X := \{\, w \in W \mid c_{13} = c_{23} = c_{31} = c_{32} = 0\,, a_i = b_i = 0\,,\ i = 1, 2\,\}$$

will be discussed. Our research is based on the classical invariant theory [13] and its computational aspects [14, 15] based on computer algebra. The quotient structure of the ring obtained as a result of restriction will be determined. Furthermore, we establish an injective homomorphism between this ring and the and the ring $\mathbb{R}[W_X]^{SO(2)\times SO(2)}$ of local unitary invariant polynomials for the 2-qubit $X-$states. In doing so, we show that the latter ring is *freely* generated by five homogeneous invariants of degrees 1,1,1,2,2.

Список литературы

[1] Ting Yu and J.H. Eberly. Evolution from entanglement to decoherence of bipartite mixed "X"states. Quantum Information and Computation, **7**, 459-468 (2007).

[2] M.A. Nielsen and I.L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press (2011).

[3] R.F. Werner. Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model. Phys. Rev. **A 40(8)** 4277–4281 (1989).

[4] M. Horodecki and P. Horodecki. Reduction criterion of separability and limits for a class of distillation protocols. Phys. Rev. **A 59** 4206–4216 (1999).

[5] S. Ishizaka and T. Hiroshima. Maximally entangled mixed states under nonlocal unitary operations in two qubits. Phys. Rev. **A 62** 22310 (4pp) (2000).

[6] F. Verstraete, K. Audenaert, T.D. Bie and B.D. Moor. Maximally entangled mixed states of two qubits. Phys. Rev. **A 64** 012316 (6pp) (2001).

[7] P. Mendonca, M. Marchiolli and D. Galetti, Entanglement universality of two-qubit $X-$states. Annals of Physics. **351**, 79—103 (2014).

[8] C. Quesne. $SU(2)\times SU(2)$ scalars in the enveloping algebra of SU(4), J. Math. Phys. **17**, 1452–1467 (1976).

[9] M. Grassl, M. Rotteler and T. Beth. Computing local invariants of qubit quantum systems. Phys. Rev. A**58**, 1833–1839 (1998).

[10] R. C. King, T. A. Welsh and P D Jarvis. The mixed two-qubit system and the structure of its ring of local invariants, J. Phys. A: Math. Theor. **40**, 10083–10108 (2007).

[1] For detailed review of the $X-$states and their applications we refer to the recent article [7].

[11] V. Gerdt, A. Khvedelidze and Yu. Palii. On the ring of local polynomial invariants for a pair of entangled qubits, Zapiski POMI **373**, 104-123 (2009).

[12] U. Fano. Description of States in Quantum Mechanics by Density Matrix and Operator Techniques. Reviews of Modern Physics **29**, 74—93 (1957).

[13] V.L. Popov and E.B. Vinberg. Invariant Theory, In:N.N. Parshin and I.R. Shafarevich, eds., Algebraic Geometry IV, Encyclopedia of Mathematical Sciences **55**, Springer-Verlag, Berlin, Heidelberg (1994).

[14] H. Derksen and G. Kemper. Computational Invariant Theory. Springer-Verlag, Berlin, Heidelberg (2002).

[15] B. Sturmfels. Algorithms in Invariant Theory. Texts and Monographs in Symbolic Computation. Second Edition. SpringerWienNewYork (2008).

[16] H. Derksen and G. Kemper. Computing invariants of algebraic groups in arbitrary characteristic. Adv. Math. **217**, 2089–2129 (2008).

V.Gerdt
Laboratory of Information Technologies,
Joint Institute for Nuclear Research
141980 Dubna, Russia
University "Dubna"
141982 Dubna, Russia
e-mail: gerdt@jinr.ru

A.Khvedelidze
Institute of Quantum Physics and Engineering Technologies,
Georgian Technical University,
Tbilisi, Georgia
A Razmadze Mathematical Institute,
Iv. Javakhishvili, Tbilisi State University,
Tbilisi, Georgia
National Research Nuclear University,
MEPhI (Moscow Engineering Physics Institute),
115409 Moscow, Russia
e-mail: akhved@jinr.ru

Yu. Palii
Institute of Applied Physics,
Chisinau, Republic of Moldova
e-mail: palii@jinr.ru