Федеральный исследовательский центр
Информатика и Управление
Российской академии наук

РУДН

ордена Ленина
им. М.В. Келдыша РАН

МГУ
МЦ
ИПМ
Московский центр фундаментальной и прикладной математики

КОМПЬЮТЕРНАЯ АЛГЕБРА

Материалы 5-й международной конференции

Москва, 26–28 июня 2023 года

COMPUTER
ALGEBRA
INTERNATIONAL
CONFERENCE
MOSCOW
2023

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
«ФЕДЕРАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР
«ИНФОРМАТИКА И УПРАВЛЕНИЕ»
РОССИЙСКОЙ АКАДЕМИИ НАУК»

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
ИМЕНИ ПАТРИСА ЛУМУМБЫ»

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
«ФЕДЕРАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ ИМ. М. В. КЕЛДЫША
РОССИЙСКОЙ АКАДЕМИИ НАУК»

КОМПЬЮТЕРНАЯ АЛГЕБРА

Материалы 5-й Международной Конференции

Москва, 26–28 июня 2023 года

COMPUTER ALGEBRA

5th International Conference Materials

Moscow, June 26–28, 2023

Москва
ИПМ им. М.В. Келдыша РАН
2023

УДК 519.6(063)
ББК 22.19;31
К637

Ответственные редакторы:
д-р физ.-мат. наук С.А. АБРАМОВ,
д-р физ.-мат. наук А.Б. БАТХИН,
д-р физ.-мат. наук Л.А. СЕВАСТЬЯНОВ

Рецензенты: канд. техн. наук Ю.О. Трусова,
канд. физ.-мат. наук К.П. Ловецкий

Компьютерная алгебра: материалы 5-й международной конференции. Москва, 26–28 июня 2023 г./ отв. ред. С.А. Абрамов, А.Б. Батхин, Л.А. Севастьянов. – Москва: ИПМ им. М.В. Келдыша, 2023.

ISBN 978-5-98354-067-5
https://doi.org/10.20948/ca-2023

Международная конференция проводится совместно ФИЦ «Информатика и управление» РАН, Российским университетом дружбы народов им. Патриса Лумумбы и ФИЦ Институтом прикладной математики им. М.В. Келдыша РАН. В представленных на конференции докладах обсуждаются актуальные вопросы компьютерной алгебры — научной дисциплины, алгоритмы которой ориентированы на точное решение математических и прикладных задач с помощью компьютера.

UDC 519.6(063)
BBC 22.19;431

Responsible editors:
Doctor of Physical and Mathematical Sciences S.A. Abramov,
Doctor of Physical and Mathematical Sciences A.B. Batkhin,
Doctor of Physical and Mathematical Sciences L.A. Sevastianov

Reviewers: PhD Yu.O. Trusova, PhD K.P. Lovetskiy

Computer algebra: 5th International Conference Materials. Moscow, 26–28 June, 2023/ ed. S.A. Abramov, A.B. Batkhin, L.A. Sevastyanov. Moscow: KIAM, 2023.

ISBN 978-5-98354-067-5
https://doi.org/10.20948/ca-2023

The international conference is organized jointly by Federal Research Center “Computer Science and Control” of Russian Academy of Sciences, Peoples’ Friendship University of Russia named after Patrice Lumumba and Keldysh Institute of Applied Mathematics of Russian Academy of Sciences. The talks presented at the conference discuss actual problems of computer algebra — the discipline whose algorithms are focused on the exact solution of mathematical and applied problems using a computer.

Conference Chair

I.A. Sokolov Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia

Program Committee General Co-Chairs

Yu.G. Evtushenko Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia
A.I. Aptekarev Keldysh Institute of Applied Mathematics of Russian Academy of Sciences, Russia
K.E. Samuylov Applied Mathematics and Communications Technology Institute, Peoples' Friendship University of Russia, Russia

Program Committee Vice Chairs

L.A. Sevastianov Peoples' Friendship University of Russia, and Joint Institute for Nuclear Research, Russia
A.B.Batkhin Keldysh Institute of Applied Mathematics of Russian Academy of Sciences, Russia
S.A. Abramov Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia

Program Committee

Yu.A. Flerov Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia
M. Barkatou Universite de Limoges, France
O.V. Druzhinina Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia
A.V. Bernstein Skolkovo Institute of Science and Technology, Russia
Yu.A. Blinkov Saratov State University, Russia
M. Wu East China Normal University, Shanghai, P.R.China
E.V. Zima Wilfrid Laurier University, Waterloo, Canada
V.V. Kornyak Joint Institute for Nuclear Research, Russia
D.S. Kulyabov Peoples' Friendship University of Russia, and Joint Institute for Nuclear Research, Russia
M.D. Malykh Peoples' Friendship University of Russia, Russia
A.A. Mikhalev Moscow State University, Russia
M. Petkovšek University of Ljubljana, Slovenia
A.N. Prokopenya Warsaw University of Life Sciences, Poland
T.M. Sadykov Plekhanov Russian University of Economics, Russia
A.A. Bogolubskaya Joint Institute for Nuclear Research, Russia
N.N. Vasilyev St.Petersburg Department of V.A.Steklov Mathematical Institute, St.Petersburg, Russia
R.R. Gontsov Institute for Information Transmission Problems of Russian Academy of Sciences, Russia
A.V. Korolkova Peoples' Friendship University of Russia, Russia
A.P. Kryukov Moscow State University, Russia
D.A. Pavlov St.Petersburg State University, Saint Petersburg Electrotechnical University "LETI", St.Petersburg, Russia

Organising Committee Chair

A.B. Batkhin Keldysh Institute of Applied Mathematics of Russian Academy of Sciences, Russia

Organising Committee Vice-Chairs

D.V. Divakov Peoples' Friendship University of Russia, Russia
A.A. Ryabenko Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia
A.A. Tiutiunnik Peoples' Friendship University of Russia, Russia

Organising Committee

Y.A. Zonn Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia
T.R. Velieva Peoples' Friendship University of Russia, Russia
A.V. Demidova Peoples' Friendship University of Russia, Russia
V.V. Dorodnicyna Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia
Yu.O. Trusova Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia
G.M. Mikhailov Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia
D.E. Khmelnov Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia
S.V. Vladimirova Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia
K.B. Teimurazov Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia

Foreword

The fifth International Conference "Computer algebra"

http://www.ccas.ru/ca/conference

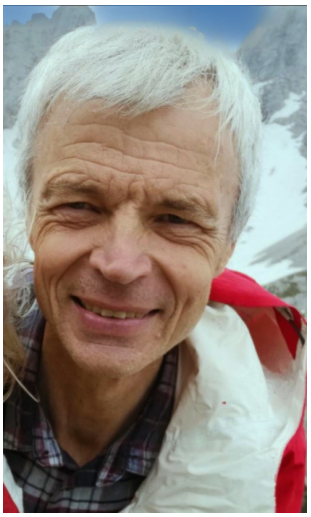
is organized in Moscow from 26 to 28 June 2023 jointly by the Dorodnicyn Computing Centre (Federal Research Center "Computer Science and Control") of Russian Academy of Science, the Russian University of Peoples' Friendship named after Patrice Lumumba and Keldysh Institute of Applied Mathematics of Russian Academy of Sciences.

The first, second, third and fourth conferences were held in Moscow in 2016, 2017, 2019 and 2021:

http://www.ccas.ru/ca/conference2016,
http://www.ccas.ru/ca/conference2017,
http://www.ccas.ru/ca/conference2019,
http://www.ccas.ru/ca/conference2021.

Computer algebra algorithms are focused on the exact solutions of mathematical and applied problems using a computer. The participants of this conference present new results obtained in this field.

During the Conference a special session in memory of Marko Petkovšek is held.



Marko Petkovšek
9.4.1955 – 24.3.2023

Program and Organizing Committees of the conference

Contents

Invited talks

Contributed talks

Invited talks

Asymptotic Nonlinear Analysis as a Calculus and Applications

A.D. Bruno
Keldysh Institute of Applied Mathematics of RAS, Russia
e-mail: abruno@keldysh.ru

Abstract

In the last 60 years, there was formed an universal asymptotic nonlinear analysis, whose unified methods allow to find asymptotic forms and asymptotic expansions to solutions of nonlinear equations and systems of different types: algebraic, ordinary differential (ODE), partial differential (PDE) and systems of mixed-type equations. This are in two methods: (1) Reducing equations to the normal form and (2) Separating truncated equations. Two kinds of transformations of coordinate can be used to simplify the obtained equations: (A) Power and (B) Logarithmic.

In this lecture, the basic ideas of this calculus are explained for the simplest cases: a single algebraic equation in Section 1, Section 2 considers the autonomous ODE system. A single partial differential equation is considered in Section 3. An overview of applications are given in Section 4. Enlarged review was published in [1].

Keywords: nonlinear analysis, power geometry, asymptotic form, asymptotic expansion

1. Calculus

There are two universal methods for local study of nonlinear equations and systems of different kinds (algebraic, ordinary and partial differential): (a) normal form and (b) truncated equations.

(a) Equations with a linear part can be reduced to its normal form by a local change of coordinates. For algebraic equation, it is Implicit Function Theorem. For systems of ordinary differential equations (ODE), I completed the theory of normal forms, began by Poincaré (1879) and Dulac (1912) for general systems [2, 3] and began by Birkhoff (1929) for Hamiltonian systems [4].

(b) Equations without linear part: I proposed to study properties of solutions to equations (algebraic, ordinary differential and partial differential) by studying sets of vector power exponents of terms of these equations. Namely to select more simple ("truncated") equations [5, 6, 7] by means of generalization to polyhedrons the Newton (1678) and the Hadamard (1893) polygons. By means of power transformations [5, 6, 8] the truncated equations can be strongly simplified and often solved. Solutions of the truncated equations are asymptotically the first approximations of the solutions to the full equations. Continuing that process, we can obtain approximations of any precision to solutions of initial equations. Basing on the developed Asymptotic Nonlinear Analysis, I proposed algorithms for solutions of a wide set of singular problems. In particular, for computation of six different types of asymptotic expansions of solutions to ODE [9, 10, 11], including expansions into trans-series [12].

2. Applications in complicated problems of (c) Mathematics, (d) Mechanics, (e) Celestial Mechanics and (f) Hydromechanics

(c) In Mathematics: together with my students I found all asymptotic expansions of five types of solutions to the Painlevé equations (1906) [10, 13] and also gave very effective method of determination of integrability of ODE system [14, 15].

(d) In Mechanics: I computed with high precision influence of small mutation oscillations on velocity of precession of a gyroscope [6] and also studied values of parameters of a centrifuge, ensuring stability of its rotation [16].

(e) In Celestial Mechanics: together with my students I studied periodic solutions of the Beletsky equation (1956) [17, 18], describing motion of satellite around its mass center, moving along an elliptic orbit. I found new families of periodic solutions, which are important for passive orientation of the satellite [6], including cases with big values of the eccentricity of the orbit, inducing a singularity. Besides, simultaneously with Hénon (1997), I found all regular and singular generating families of periodic solutions of the restricted three-body problem and studied bifurcations of generated families. It allowed to explain some singularities of motions of small bodies of the Solar System [19]. In particular, I found orbits of periodic flies round planets with close approach to the Earth [20].

(f) In Hydromechanics: I studied small surface waves on a water [7], a boundary layer on a needle [21], where equations of a flow have a singularity, and a model of a turbulent flow [22, 23].

References

[1] Bruno A. D. Nonlinear Analysis as a Calculus // *London Journal of Research in Science: Natural and Formal*. 2023. Vol. 23, no. 5. P. 1–31.

[2] Bruno A. D. Analytical form of differential equations (I) // *Trans. Moscow Math. Soc.* 1971. Vol. 25. P. 131–288.

[3] Bruno A. D. Analytical form of differential equations (II) // *Trans. Moscow Math. Soc.* 1972. Vol. 26. P. 199–239.

[4] Bruno A. D. The Restricted 3–body Problem: Plane Periodic Orbits. Berlin : Walter de Gruyter, 1994. = Nauka, Moscow, 1990. 296 p. (in Russian).

[5] Bruno A. D. The asymptotic behavior of solutions of nonlinear systems of differential equations // *Soviet Math. Dokl.* 1962. Vol. 3. P. 464–467.

[6] Bruno A. D. Local Methods in Nonlinear Differential Equations. Berlin – Heidelberg – New York – London – Paris – Tokyo : Springer–Verlag, 1989.

[7] Bruno A. D. Power Geometry in Algebraic and Differential Equations. Amsterdam : Elsevier Science, 2000.

[8] Bruno A. D. On the generalized normal form of ODE systems // *Qual. Theory Dyn. Syst.* 2022. Vol. 21, no. 1. doi: 10.1007/s12346-021-00531-4.

[9] Bruno A. D. Asymptotics and expansions of solutions to an ordinary differential equation // *Russian Mathem. Surveys*. 2004. Vol. 59, no. 3. P. 429–480.

[10] Bruno A. D., Goruchkina I. V. Asymptotic expansions of solutions of the sixth Painlevé equation // *Transactions of Moscow Math. Soc.* 2010. Vol. 71. P. 1–104.

[11] Bruno A. D. Complicated and exotic expansions of solutions to the Painlevé equations / ed. by Filipuk Galina, Lastra Alberto, Michalik Sławomir. Springer Proceedings in Mathematics & Statistics. Springer Cham, 2018. P. 103–145.

[12] Bruno A. D. Power-exponential transseries as solutions to ODE // *Journal of Mathematical Sciences: Advances and Applications*. 2019. Vol. 59. P. 33–60.

[13] Bruno A. D. Power geometry and expansions of solutions to the Painlevé equations // *Transnational Journal of Pure and Applied Mathematics*. 2018. Vol. 1, no. 1. P. 43–61.

[14] Bruno A. D., Enderal V. F. Algorithmic analysis of local integrability // *Doklady Mathematics*. 2009. Vol. 79, no. 1. P. 48–52.

[15] Bruno A. D., Enderal V. F., Romanovski V. G. Computer Algebra in Scientific Computing / ed. by Gerdt V. P., et al. Berlin Heidelberg : Springer, 2017. Vol. 10490 of *Lecture Notes in Computer Science*. doi: 10.1007/978-3-642-32973-9.

[16] Batkhin A. B., Bruno A. D., Varin V. P. Stability sets of multiparameter Hamiltonian systems // *Journal of Applied Mathematics and Mechanics*. 2012. Vol. 76, no. 1. P. 56–92. doi: 10.1016/j.jappmathmech.2012.03.006.

[17] Bruno A. D. Families of periodic solutions to the Beletsky equation // *Cosmic Research*. 2002. Vol. 40, no. 3. P. 274–295.

[18] Bruno A. D., Varin V. P. Classes of families of generalized periodic solutions to the Beletsky equation // *Celestial Mechanics and Dynamical Astronomy*. 2004. Vol. 88, no. 4. P. 325–341.

[19] Bruno A. D., Varin V. P. Periodic solutions of the restricted three-body problem for small mass ratio // *J. Appl. Math. Mech.* 2007. Vol. 71, no. 6. P. 933–960.

[20] Bruno A. D. On periodic flybys of the Moon // *Celestial Mechanics*. 1981. Vol. 24, no. 3. P. 255–268.

[21] Bruno A. D., Shadrina T. V. Axisymmetric boundary layer on a needle // *Transactions of Moscow Math. Soc.* 2007. Vol. 68. P. 201–259.

[22] Bruno A. D., Batkhin A. B. Computation of asymptotic forms of solutions to system of nonlinear partial differential equations // *Preprints of KIAM*. 2022. no. 48. P. 36. doi: 10.20948/prepr-2022-48 (in Russian).

[23] Bruno A. D., Batkhin A. B. Asymptotic forms of solutions to system of nonlinear partial differential equations // *Universe*. 2023. Vol. 9, no. 1. P. 35. doi: 10.3390/universe9010035.

Hermite Reduction for D-finite Functions[1]

Shaoshi Chen[1,2], Lixin Du[3], Manuel Kauers[3]

[1]*KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China*

[2]*School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China*

[3]*Institute for Algebra, Johannes Kepler University, Linz, A4040, Austria*

e-mail: schen@amss.ac.cn, lixin.du@jku.at, manuel.kauers@jku.at

Abstract

Trager's Hermite reduction solves the integration problem for algebraic functions via integral bases. A generalization of this algorithm to D-finite functions has so far been limited to the Fuchsian case. In the present paper, we remove this restriction and propose a reduction algorithm based on integral bases that is applicable to arbitrary D-finite functions.

Keywords: additive decomposition, creative telescoping, symbolic integration

Let R be a certain class of functions in one variable x with the derivation D_x. For example, R can be the field of rational functions or algebraic functions. In the context of symbolic integration, the *integrability problem* consists in deciding whether a given element $f \in R$ is of the form $f = D_x(g)$ for some $g \in R$. If such a g exists, we say that f is *integrable* in R. A relaxed form of the integrability problem is the *decomposition problem,* which consists in constructing for a given $f \in R$ elements $g, r \in R$ such that $f = D_x(g) + r$ and r is minimal in a certain sense. Ideally the "certain sense" should be such that $r = 0$ whenever f is integrable. If $f \in R$ depends on a second variable t, one can also consider the *creative telescoping* problem: given an element $f \in R$, the task is to construct $c_0, \ldots, c_r \in R$, not all zero, such that c_i is free of x for all $i \in \{0, \ldots, r\}$ and

$$c_r D_t^r(f) + \cdots + c_0 f = D_x(g) \quad \text{for some } g \in R.$$

The operator $L = c_r D_t^r + \cdots + c_0$, if it exists, is called a *telescoper* for f, and g is called a *certificate* for L.

Zeilberger first showed the existence of telescopers for D-finite functions [1]. Almkvist and Zeilberger [2] solved the integrability problem and the creative telscoping problem for hyperexponential functions. Using the adjoint Ore algebra, Abramov and van Hoeij [3] solved the accurate integration problem for D-finite functions. Chyzak [4] extended the method of creative telescoping from hyperexponential functions to general D-finite functions. During the past ten years, a reduction-based telescoping approach has become popular, which can find a telescoper without computing the corresponding certificate. This approach was first formulated for rational functions [5] and later extended to hyperexponential functions [6], algebraic functions [7], Fuchsian D-finite functions [8] and D-finite functions [9, 10]. The reduction-based telescoping algorithms for algebraic functions and for Fuchsian D-finite functions employ the notion of integral bases, while the known reduction-based telescoping algorithms applicable to arbitrary D-finite functions work differently.

The notion of integrality proposed by Kauers and Koutschan [11] for Fuchsian D-finite functions has recently been generalized by Aldossari [12] to arbitrary D-finite functions, so that the question arises whether there is also a reduction-based telescoping algorithm for

[1]This extended abstract is based on our recent paper submitted to the conference ISSAC'23.

arbitrary D-finite functions based on integral bases. The purpose of this work is to answer this question affirmatively, which is based on the results of Chapter 6 of the second author's Ph.D. thesis [13].

Let C be a field of characteristic zero and $\bar{C}$ be the algebraic closure of C. Let $C(x)[D]$ be an Ore algebra, where D is the differentiation with respect to x and satisfies the commutation rule $Dx = xD + 1$. For an operator $L = \ell_0 + \ell_1 D + \cdots + \ell_n D^n \in C(x)[D]$ with $\ell_n \neq 0$, we consider the left $C(x)[D]$-module $A = C(x)[D]/\langle L\rangle$, where $\langle L\rangle = C(x)[D]L$. We call the elements of A "functions", even though they are not functions in the usual sense. This is fair because A is isomorphic to a $C(x)[D]$-module containing actual functions. When there is no ambiguity, an equivalence class $f + \langle L\rangle$ in A is also denoted by f. Every element of A can be uniquely represented by $f = f_0 + f_1 D + \cdots + f_{n-1}D^{n-1}$ with $f_i \in C(x)$. Let $W = (\omega_1, \ldots, \omega_n) \in A^n$ be an integral basis of A that is normal at infinity (for the precise definition of integral bases for arbitrary linear differential operators, see [11, 14, 15, 12]). There exists $T = \operatorname{diag}\big(x^{\tau_1}, \ldots, x^{\tau_n}\big) \in C(x)^{n\times n}$ with $\tau_i \in \mathbb{Z}$ such that $V := TW$ is a local integral basis at infinity. (Theoretically, we can also start with W being a local integral basis at $\bar{C} \setminus \{\alpha\} \cup \{\infty\}$ that is normal at α.) Let $e, a \in C[x]$ and $M, B \in C[x]^{n\times n}$ be such that $eW' = MW$ and $aV' = BV$. Since the derivative of V is $V' = (TW)' = (T' + \frac{1}{e}TM)T^{-1}V$, we may assume that $a = x^\lambda e$ for some $\lambda \in \mathbb{N}$. For $\mu, \delta \in \mathbb{Z}$ with $\mu \leq \delta$, we define a subspace of Laurent polynomials in $C[x, x^{-1}]$ as $C[x]_{\mu,\delta} := \{\sum_{i=\mu}^{\delta} a_i x^i \,|\, a_i \in C\}$. The main result of this work is the following additive decomposition for a general D-finite function.

Theorem 1. *Let $W, V \in A^n$ be as described above. Then any element $f \in A$ can be decomposed into*

$$f = g' + \frac{1}{d}RW + \frac{1}{x^\lambda e}QV, \tag{1}$$

where $g \in A$, $d \in C[x]$ is squarefree and $\gcd(d, e) = 1$, $R \in C[x]^n$, $Q \in C[x]^n_{\mu,\delta}$ with $\deg_x(R) < \deg_x(d)$, $\mu = \min\{-\tau_1, \ldots, -\tau_n, 0\}$ and $\delta = \max\{\lambda + \deg_x(e), \deg_x(B)\} - 1$. Moreover, f is integrable in A if and only if $R = 0$ and

$$\frac{1}{x^\lambda e}QV \in U' \quad \text{with} \quad U = \left\{ \frac{1}{u}cV \;\middle|\; c \in C[x]^n_{\mu',\delta'} \right\},$$

where $u = \gcd(e, e')$, $\mu' = \min\{-\tau_1, \ldots, -\tau_n, \nu_0(u)\}$ and

$$\delta' = \max\{\deg_x(u), \deg_x(B) - \lambda - \deg_x(e) + \deg_x(u)\}.$$

References

[1] Zeilberger Doron. A holonomic systems approach to special functions identities // *Journal of Computational and Applied Mathematics*. 1990. Vol. 32. P. 321–368.

[2] Almkvist Gert, Zeilberger Doron. The method of differentiating under the integral sign // *Journal of Symbolic Computation*. 1990. Vol. 10. P. 571–591.

[3] Abramov Sergei A, Hoeij Mark Van. Integration of solutions of linear functional equations // *Integral Transforms and Special Functions*. 1999. Vol. 8, no. 1-2. P. 3–12.

[4] Chyzak Frédéric. An extension of Zeilberger's fast algorithm to general holonomic functions // *Discrete Mathematics*. 2000. Vol. 217. P. 115–134.

[5] Complexity of creative telescoping for bivariate rational functions / Bostan Alin, Chen Shaoshi, Chyzak Frédéric, and Li Ziming // Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation. New York, NY, USA : ACM. 2010. P. 203–210.

[6] Hermite reduction and creative telescoping for hyperexponential functions / Bostan Alin, Chen Shaoshi, Chyzak Frédéric, Li Ziming, and Xin Guoce // Proceedings of the 2013 International Symposium on Symbolic and Algebraic Computation. New York, NY, USA : ACM. 2013. P. 77–84.

[7] Chen Shaoshi, Kauers Manuel, Koutschan C. Reduction-based creative telescoping for algebraic functions // Proceedings of the 2016 International Symposium on Symbolic and Algebraic Computation. New York, NY, USA : ACM. 2016. P. 175–182.

[8] Reduction-based creative telescoping for Fuchsian D-finite functions / Chen Shaoshi, van Hoeij Mark, Kauers Manuel, and Koutschan Christoph // *Journal of Symbolic Computation*. 2018. Vol. 85. P. 108–127.

[9] van der Hoeven Joris. Constructing reductions for creative telescoping: the general differentially finite case // *Applicable Algebra in Engineering, Communication and Computing*. 2021. nov. Vol. 32, no. 5. P. 575–602.

[10] Generalized Hermite reduction, creative telescoping and definite integration of D-finite functions / Bostan Alin, Chyzak Frédéric, Lairez Pierre, and Salvy Bruno // Proceedings of the 2018 International Symposium on Symbolic and Algebraic Computation. New York, NY, USA : ACM. 2018. P. 95–102.

[11] Kauers Manuel, Koutschan Christoph. Integral D-finite functions // Proceedings of the 2015 International Symposium on Symbolic and Algebraic Computation. New York, NY, USA : ACM. 2015. P. 251–258.

[12] Aldossari Shayea. Algorithms for Simplifying Differential Equations. Florida State University, 2020. PhD thesis.

[13] Du Lixin. Generalized Integral Bases and Applications in Creative Telescoping. Johannes Kepler University, 2022. PhD thesis.

[14] Imamoglu Erdal. Algorithms for Solving Linear Differential Equations with Rational Function Coefficients. Florida State University, 2017. PhD thesis.

[15] Imamoglu Erdal, van Hoeij Mark. Computing hypergeometric solutions of second order linear differential equations using quotients of formal solutions and integral bases // *Journal of Symbolic Computation*. 2017. Vol. 83. P. 254–271.

A Saga on a Generating Function of the Squares of Legendre Polynomials

M. van Hoeij

Florida State University, Department of Mathematics, USA

e-mail: hoeij@math.fsu.edu

Abstract

We decompose the generating function $\sum_{n=0}^{\infty} \binom{2n}{n} P_n(y)^2 z^n$ of the squares of Legendre polynomials as a product of periods of hyperelliptic curves. These periods satisfy *second* order differential equations which is highly unusual since *four* is the expected order for genus 2. These second order equations are arithmetic and yet their monodromy group is dense in $\mathrm{SL}_2(\mathbb{R})$. This implies that they cannot be solved in terms of hypergeometric functions, which is novel for an arithmetic equation that occurred naturally. This is joint work with Duco van Straten and Wadim Zudilin.

Two Methods for Efficient Generic Inversion

Stephen M. Watt
Cheriton School of Computer Science, University of Waterloo, Canada
e-mail: smwatt@uwaterloo.ca

Abstract

Two generic methods to compute multiplicative inverses are presented. These methods apply to integers, polynomials and matrices and are asymptotically faster than classical algorithms. The first method is to use a modified Newton iteration to compute quotients via shifted inverses in a not-necessarily commutative Euclidean domain. The second method is to use the Moore-Penrose inverse to avoid pivots and whole-row operations in block matrix inversion.

1. Introduction

In computer algebra it is desirable to find algorithms that can be expressed over abstract algebraic domains and to implement these generically. For example, programs to multiply polynomials in $R[x]$ or perform Gaussian elimination on matrices in $F^{n\times n}$ can be expressed as programs that use the ring and field operations from R and F. This avoids multiple implementations of the same algorithms and allows flexible composition of domains. Systems such as Axiom are built around this principle, and others can support it through mechanisms such as the Maple Domains package. Most modern programming languages, such as C++, C#, Go, Java, Rust and Typescript, support generic programming in one form or another.

In mathematical computing it is required to find quotients or inverses of various types of objects, including integers, polynomials and matrices. In this paper we summarize some earlier results showing how to compute these quotients and inverses efficiently and generically. Section 2 shows a generic algorithm to compute integer and polynomial quotients. This algorithm is both useful in practice and can be asymptotically fast (depending on the multiplication method) and performs all operations without leaving the original domain. Section 3 shows how matrix inverses may be computed on a block representation without breaking the block abstraction.

2. Modified Newton Iteration for Euclidean Domains

On a Euclidean domain D with valuation $N : D \to \mathbb{R}_{\geq 0}$, we define the quotient and remainder of u by v as the unique values q and r such that $u = q \times v + r, \quad N(r) < N(v)$ and write $q = u \operatorname{quo} v$ and $r = u \operatorname{rem} v$. For both integers and polynomials it is well known how to compute quotients efficiently using a Newton iteration. For $u, v \in \mathbb{Z}$, the quotient of u by v may be found by first computing v^{-1} in $\mathbb{R}$ to sufficient precision with a Newton iteration solving $f(x) = 1/x - v = 0$. For $u, v \in F[x]$, F a field, the quotient may be computed in $F[x]/\langle x^{m+1}\rangle$ using Newton iteration to find the inverse of the reverse polynomial $\operatorname{rev}_k v = x^k v(1/x)$, where k and $k+m$ are the degrees of v and u respectively. In both cases, the computation leaves the original domain, which can complicate library structure. In earlier work [8], we have shown how to compute these quotients using only ring operations and shifts with values remaining in the original domain. We summarize those results here.

We define the operations "prec", "shift" and "shinv" on base-B integers and polynomials in x as follows:

Number of coefficients:	$\text{prec}_B(w) = \lfloor \log_B \lvert w \rvert \rfloor + 1$	$\text{prec}_x(p) = \text{degree}_x p + 1$
Whole shift:	$\text{shift}_{n,B}(w) = \lfloor wB^n \rfloor$	$\text{shift}_{n,x}(p) = \sum_{i+n \geq 0} p_i x^{i+n}$
Whole shifted inverse:	$\text{shinv}_n(w) = \lfloor B^n / w \rfloor$	$\text{shinv}_{n,x}(p) = x^n \operatorname{quo} p$

where $p = \sum_{i=0}^{h} p_i x^i$, $n \in \mathbb{Z}$ and $B \in \mathbb{Z}_{\geq 2}$. When the base B or variable x are clear from context, they may be omitted and we simply write shift_n and shinv_n. Depending on the implementation, the shift operation can be performed in time $O(1)$ or $O(n)$. With these definitions, efficient quotients may be computed generically by the following theorems.

Theorem 1. *Let D be $\mathbb{Z}$ or $F[x]$, F a field. Given $u, v \in D$, and* $\text{prec}\, u \leq h + 1$,

$$u \operatorname{quo} v = \text{shift}_{-h}(u \cdot \text{shinv}_h\, v) + \delta, \tag{1}$$

where $\delta = 0$ when $D = F[x]$ and $\delta \in \{0, 1\}$ when $D = \mathbb{Z}$.

Theorem 2. *Let D be $\mathbb{Z}$ or $F[x]$, F a field. Given $v \in D$,* $\text{prec}\, v = k + 1 < h + 1$ *and suitable starting value $w_{(0)}$, the sequence of iterates*

$$w_{(i+1)} = w_{(i)} + \text{shift}_{-h}\left(w_{(i)}(\text{shift}_h\, 1 - v w_{(i)})\right)$$

converges to $\text{shinv}_h\, v + \delta$ *in* $\lceil \log_2(h - k) \rceil$ *steps.*

The $\delta = 1$ integer case causes no problems, as it is easy to first check whether $u < v + v$. After dispensing with special cases, the shifted inverse of the first two places for integer v may be computed as

$$V := v_k B^2 + v_{k-1} B + v_{k-2} \qquad\qquad w_0 := (B^4 - V) \operatorname{quo} V + 1,$$

assuming $B \geq 16$ and prec $v - 1 = k \geq 2$. The quotient to produce w_0 is obtained by dividing a 4-place quantity by a 2-place quantity. If the base B is less than 16, then digits may be grouped to give a sufficiently large base. For polynomials, the shifted inverse of the first two places of v will be

$$w_0 := x/v_k - v_{k-1}/v_k^2.$$

In both cases, the h-shifted inverse of v may then be computed generically as $\textsc{Shinv}(v, h, k, w_0, 2)$, as shown in Algorithm 1. Note that for polynomials a 1-place initial value would be sufficient to start the iteration, but the generic algorithm takes a simpler form when two places are given. The function HasCarries indicates whether addition can cause carries from one coefficient place to another in the arithmetic of the domain. It gives "true" for integers and "false" for polynomials. The details of this algorithm are justified by the following theorems in the integer case. They relate to iterates of the function $S_{\mathbb{Z}}$, defined as

$$S_{\mathbb{Z}}(h, v, w) := w + \left\lfloor w(B^h - vw)B^{-h} \right\rfloor, \tag{2}$$

which has fixed points at $0, 1, \lfloor B^h/v \rfloor - 1$ and $\lfloor B^h/v \rfloor$. Together they show how to compute intermediate iterates with shorter quantities using two guard digits. For polynomials the situation is simpler and these shorter intermediate results may be used without guard digits.

Algorithm 1 Generic iteration to compute $\text{shinv}_h(v)$ in D given initial approximation w_0

function SHINV (v, h, k, w_0, ℓ)
  ▷ *w is the current approximation. ℓ is the number of leading correct places of w.*
  ▷ *g is the number of guard places. d is the precision doubling shortfall.*
  if HASCARRIES(D) **then** $g \leftarrow 2;\ d \leftarrow 1$ **else** $g \leftarrow 0;\ d \leftarrow 0$
  $w \leftarrow \text{shift}(w,\ g)$
  while $h - k + 1 - d > \ell$ **do**
    $m \leftarrow \min(h - k + 1 - \ell,\ \ell);\quad s \leftarrow \max(0,\ k - 2\ell + 1 - g)$
    $w \leftarrow \text{shift}_{-d}\left(\text{STEP}\big(k + \ell + m - s - 1 + d + g,\ \text{shift}_{-s} v,\ w,\ m,\ \ell - g\big)\right)$
    $\ell \leftarrow \ell + m - d$
  return $\text{shift}_{-g}(w)$

function STEP $(h, v, w, m, \ell) = \text{shift}_m\, w + \text{shift}_{2m-h}\left(w \times \text{POWDIFF}(v, w, h - m, \ell)\right)$

function POWDIFF (v, w, h, ℓ) ▷ *Compute $\text{shift}_h\, 1 - v \times w$ efficiently.*
  $c \leftarrow$ **if** HASCARRIES(D) **then** 1 **else** 0
  $L \leftarrow \text{prec}\ v + \text{prec}\ w\ - \ell + c$ ▷ *c for coeff to peek*
  if $v = 0 \vee w = 0 \vee L \geq h$ **then return** $\text{shift}_h\, 1 - v \times w$
  else
    $P \leftarrow \text{multmod}(v, w, L)$ ▷ *Lower L places of product vw.*
    if HASCARRIES$(D) \wedge \text{coeff}(P, L - 1) \neq 0$ **then return** $\text{shift}_L\, 1 - P$
    else return $-P$

Theorem 3 (Shift Extension)**.** *Let $w = \text{shinv}_h\, v$, $B^k \leq v < B^{k+1} \leq B^h$ and let $w_{[n]} = \text{shift}_{n\ell-h+k}(w)$ be the leading $n\ell$ digits of w, with $n\ell \leq h - k$. Then*

$$0 \leq w_{[2]} - S_{\mathbb{Z}}(k + 2\ell,\ v,\ \text{shift}_\ell\, w_{[1]}) \leq B.$$

Theorem 4 (Divisor Sensitivity)**.** *Let $w_{[n]}$ be as in Theorem 3 and let Δ be the change obtained by perturbing the divisor v by δ in $S_{\mathbb{Z}}(k + 2\ell,\ v,\ \text{shift}_\ell\, w_{[1]})$, i.e.*

$$\Delta = S_{\mathbb{Z}}(k + 2\ell,\ v - \delta,\ \text{shift}_\ell\, w_{[1]}) - S_{\mathbb{Z}}(k + 2\ell,\ v,\ \text{shift}_\ell\, w_{[1]}).$$

Then

$$B^{2\ell-k-2}\delta - 1 < \Delta < B^{2\ell-k}\delta + 1.$$

In particular, if $\delta \leq B^{k-2\ell+1}$, then $0 \leq \Delta \leq B$.

Theorem 5 (Close Differences)**.** *When $\left| B^h - vw \right| \leq B^e,\quad e < h$, only the lower e digits of the product vw need be computed since the upper $h - e$ digits will be determined. The quantity e satisfies*

$$e \leq k + t - \ell + g,$$

where $\text{prec}\ v = k + 1$ *and* $\text{prec}\ w = t + 1$*, ℓ is the number of known correct places in w and g is the required number of guard digits.*

Theorems 3 and 4 together show two guard digits are required when the domain is $\mathbb{Z}$. None are required for polynomials since there cannot be carries. Theorem 5 shows how to compute the difference $\text{shift}_h\, 1 - vw$, using only a suffix of vw. This will give a savings for some multiplication algorithms, but not for the asymptotically fastest ones.

These results can be extended to non-commutative polynomials with left and right quotients [9]. Define "lquo" and "rquo" by $u = v \times (u \operatorname{lquo} v) + r_L = (u \operatorname{rquo} v) \times v + r_R$, where each of r_L and r_R either zero or with degree less than v. For $R[x]$, where R is not necessarily commutative, shinv is remains well-defined, that is it can be shown $\operatorname{shinv}_{n,x}(v) = x^n \operatorname{lquo} v = x^n \operatorname{rquo} v$. It remains the case that shinv may be computed in a logarithmic number of steps and quotients may be computed according to the following:

Theorem 6 (Left and right quotients from the whole shifted inverse in $R[x]$)**.** *Let $u, v \in R[x]$, R a ring, with* $\operatorname{degree} v = k$ *and v_k invertible in R. Then for $h \geq \operatorname{degree} u$,*

$$u \operatorname{lquo} v = \operatorname{shift}_{-h}(\operatorname{shinv}_h(v) \times u) \qquad u \operatorname{rquo} v = \operatorname{shift}_{-h}(u \times \operatorname{shinv}_h(v)). \tag{3}$$

To have a well-defined notion of degree for polynomials where the variable does not commute with the coefficients, we are led to skew polynomials as Ore extensions [2, 4]. In this case, we may define the left (right) whole shift by multiplying on the left (right) by x^n and left (right) whole shifted inverse as the left (right) quotient of x^n and we have the following theorem, though the computation is no longer asymptotically fast.

Theorem 7 (Right quotient from the whole shifted inverse in $R[x; \sigma, \delta]$)**.** *Let $u, v \in R[x, \delta]$, R a ring, $k = \operatorname{degree} v$, and v_k invertible in R. Then, for $h \geq \operatorname{degree} u$,*

$$u \operatorname{rquo} v = \operatorname{rshift}_{-h}(u \times \operatorname{lshinv}_h v). \tag{4}$$

3. Inversion without Pivots for Block Matrices over Division Rings

As has been noted by Abdali and Wise [1], a useful computational representation of matrices over a ring R is with a recursive 2×2 block structure. This representation allows efficient implementation of sub-n^3 multiplication and related operations [6]. It supports row-based and column-based traversal equally well, it is reasonably efficient in representing dense, sparse and structured matrices, and it can provide good locality of reference for block algorithms. Most algebraic operations can be expressed naturally in terms of a recursive abstract data type represented as quadtrees with leaves in R and, if desired, these may be stored densely without using pointers [5].

Most ring operations on block matrices may be performed in a straightforward manner using only block operations. That is, for a block matrix $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, only ring operations involving A, B, C and D are required. Computing the matrix inverse is less obvious, however. If all of the blocks of M are invertible, the inverse of M may be computed as

$$M^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & (C - DB^{-1}A)^{-1} \\ (B - AC^{-1}D)^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}.$$

In practice, the usual approach is to compute only two inverses—that of A and that of its Schur complement, $S_A = D - CA^{-1}B$,

$$M^{-1} = \begin{bmatrix} I & -A^{-1}B \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & S_A^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} = \begin{bmatrix} A^{-1} + A^{-1}BS_A^{-1}CA^{-1} & -A^{-1}BS_A^{-1} \\ -S_A^{-1}CA^{-1} & S_A^{-1} \end{bmatrix}. \tag{5}$$

If A is not invertible, then a similar formula involving the inverse of another block and its Schur complement may be used, perhaps after a permutation of rows or columns. The problem with this approach is that M may be invertible even when all of A, B, C and D are singular. In this situation, permuting the blocks is of no help. One approach is to break the block abstraction and use operations on whole rows of M viewed as a flat $2^k \times 2^k$ matrix [3].

In earlier work [7], we have shown how to compute inverses using only block operations, without pivots and without breaking the block abstraction. The technique is to use the Moore-Penrose inverse so that the principal minors are guaranteed to be invertible and equation (5) may be used. We summarize those results here. We use the notation $R^{(2\times 2)^k}$ to mean the ring of $2^k \times 2^k$ matrices with elements in R, structured in recursive 2×2 blocks. Any $n \times n$ matrix may be easily be embedded in such a ring.

Theorem 8. *If R is a formally real division ring and $M \in R^{n\times n}$ is invertible, then it is possible to compute M^{-1} as $(M^T M)^{-1} M^T$ using only block operations. By block operations, we mean ring operations in $R^{(2\times 2)^k}$.*

Examples of formally real rings are $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{Q}[\sqrt{2}]$ and $R[x, \partial]$ for formally real R.

Theorem 9. *Let C be a division ring with a formally real sub-ring R and involution "$*$", such that for all $c \in C$, $c^* \times c$ is a sum of squares in R. If $M \in C^{n\times n}$ is invertible, then it is possible to compute M^{-1} as $(M^* M)^{-1} M^*$ using only block operations. Here, block operations are ring operations in $C^{(2\times 2)^k}$.*

Examples of such rings are the complexification of a formally real ring R as $R[i]/\langle i^2 + 1\rangle$ or quaternions over R with the involution $(a + bi + cj + dk)^* = a - bi - cj - dk$.

Theorem 10. *Let K be a field. If $M \in K^{n\times n}$ is invertible, then it is possible to compute M^{-1} as $(M^\circ M)^{-1} M^\circ$ using only block operations, that is ring operations in $K(t)^{(2\times 2)^k}$.*

Here $M^\circ = Q_n^{-1} M^T Q_n$ is a group conjugate of M^T, with $Q_n = \mathrm{diag}(1, t, \ldots, t^{n-1})$.

4. Conclusion

We have shown two generic techniques to compute multiplicative inverses efficiently in algebraic domains of importance to computer algebra. These methods are supported by theorems stated here and proven in earlier work.

References

[1] S. Kamal Abdali and David S. Wise. Experiments with quadtree representation of matrices. In *Symbolic and Algebraic Computation*, pages 96–108. Springer, 1989.

[2] Sergei A. Abramov, H. Q. Le, and Ziming Li. Univariate Ore polynomial rings in computer algebra. *Journal of Mathematical Sciences*, 131(5):5885–5903, 2005.

[3] Alfred V. Aho, John E Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, Mass., 1974.

[4] Manuel Bronstein and Marko Petkovšek. An introduction to pseudo-linear algebra. *Theoretical Computer Science*, 157(1):3–33, 1996.

[5] Irene Gargantini. An effective way to represent quadtrees. *Communications of the ACM*, 25(12):905–910, December 1982.

[6] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969.

[7] Stephen M. Watt. Pivot-free block matrix inversion. In *Proc. 8th Int'l Symp. on Symbolic and Numeric Algorithms for Scientific Computing*, pages 151–155. IEEE Press, 2006.

[8] Stephen M. Watt. Efficient generic quotients using exact arithmetic. In *Proc. International Symposium on Symbolic and Algebraic Computation*, New York, 2023. ACM.

[9] Stephen M. Watt. Efficient quotients of non-commutative polynomials, arxiv:2305.17877, 2023.

Modular Arithmetic With Special Choice of Moduli

E.V. Zima

Wilfrid Laurier University, Waterloo, Canada

e-mail: ezima@wlu.ca

Abstract

Several methods of selection of moduli in modular arithmetic are discussed. With the proposed choice of moduli both modular reduction of an integer and reconstruction from modular images are accelerated. Special attention is paid to the moduli of the forms $2^n \pm 1$ and $2^n \pm 2^k \pm 1$. Different schemes of choice of these types of moduli and accelerated conversion of arbitrary precision integers into the modular representation and back are considered. Results of experimental implementation of different modular schemes confirm practicality of proposed methods.

Keywords: residue number system, modular reduction, modular reconstruction

1. Intorduction

Conversion to a modular representation is a popular technique to accelerate arithmetic of computer algebra systems. Modular algorithms are successfully used for calculation of the greatest common divisors of polynomials, polynomial factorization, in problems of symbolic linear algebra, for symbolic integration and summation. Modular approach to symbolic computations has several advantageous features. One of them is the possibility to control the size of intermediate results, which is often impossible in the case of computations with arbitrary precision integer or rational numbers. This implies faster and more efficient implementations of symbolic algorithms.

The idea of modular arithmetic is to select positive integers $m_0, m_1, \ldots, m_k$, referred to as moduli; replace the initial integer data by residues modulo $m_i, i = 0, \ldots, k$; and perform a series of identical calculations modulo m_i (for every $i = 0, \ldots, k$) instead of required calculations with long integer numbers. On the final stage, the result needs to be reconstructed from the set of residues modulo m_i. The possibility of reconstruction is based on the Chinese remainder theorem [6, 8] if, for example, all the moduli are pairwise co-prime and the final result does not exceed $m_0 m_1 \ldots m_k$.

The choice of specific values of m_i can influence significantly the time complexity of both calculations modulo m_i and reconstruction of the result. For example, after multiplication of numbers less then m_i, in the general case, division by m_i may be needed to obtain the residue value. When the divisor (the modulus in this case) has a special form, more efficient algorithms for modular reduction may be available. One popular approach is to choose many moduli that fit machine word and use hardware arithmetic for the simultaneous reduction/reconstruction [4]. The simultaneous reduction and reconstruction without requirement of moduli to be small was also explored in [3] and [2].

Another approach is related to the choice of moduli with special shape (bit pattern) that accelerates reduction modulo m_i. One of the oldest examples of such approach is described in [10, 8]: several relatively prime moduli of the form $2^n - 1$ are selected. This replaces division with remainder in the residue computation by shift and addition operations that are much simpler (using that $2^n \equiv 1 \bmod (2^n - 1)$, the remainder from division of x by $2^n - 1$ can be obtained by splitting x into several numbers of bit-length n from right to left and adding them modulo $2^n - 1$). However, this choice does not offer significant improvement to the bit-complexity of the reconstruction phase of the computations. In the following section we will discuss alternative choices of moduli with certain bit-pattern.

2. Choosing moduli with a bit-pattern

Consider pairwise co-prime natural numbers $m_0, m_1, \ldots, m_k$ and assume we work with non-negative representatives in each class of residues modulo m_i. Modular reconstruction problem is: given non-negative $x_0, x_1, \ldots, x_k$ ($x_i < m_i$), find non-negative $X < m_0 m_1 \ldots m_k$ such that $X = x_i \bmod m_i$ for $i = 0, 1, \ldots, k$. Standard modular reconstruction algorithms [5, 6] (pre-)compute products of moduli – $M_i = \prod_{j=0}^{i-1} m_j$ and inverses $M_i^{-1} \bmod m_i, i = 0, 1, \ldots, k-1$. The process of reconstruction involves several multiplications by these quantities which provide significant contribution to the total complexity of the reconstruction. When deciding a particular approach to the choice of moduli m_i one can try to satisfy the following natural requirements:

1. $\gcd(m_i, m_j) = 1$ for $i \neq j$;
2. reduction modulo m_i is "simpler" than division with remainder;
3. products of moduli and their inverses mentioned above have bit-pattern (preferably scalable) that allows to accelerate multiplication by those quantities;
4. bit-length of moduli m_i is balanced.

■ It is easy to select moduli of the form $2^n - 1$ that satisfy requirement 1 (as $\gcd(2^n - 1, 2^m - 1) = 1$ if and only if $\gcd(n, m) = 1$), requirement 2 (as discussed in the Introduction), and requirement 4 (by choosing relatively prime exponents of close values).
■ Different strategies of selecting the moduli of the form $2^n + 1$ were considered in [12]. Relative primality of such moduli guaranteed by the proper choice of exponents driven by the following fact:

$$\gcd(2^m + 1, 2^n + 1) = 1 \iff v_2(m) \neq v_2(n),$$

where $v_2(x)$ is the binary valuation of x (the number of trailing zeros in the binary representation of x). This choice of moduli also satisfies condition 2 (using $2^n \equiv -1 \bmod (2^n + 1)$, the remainder from division of x by $2^n + 1$ can be obtained by splitting x into several numbers of bit-length n from right to left and subtracting/adding them modulo $2^n + 1$ (see [12] for details)). It is also easy to satisfy condition 3. Consider moduli of the form $m_i = 2^{a2^i} + 1, \quad i = 0, 2, \ldots, k$, where a is an arbitrary positive integer, and products $M_i = \prod_{j=0}^{i-1} m_j = \prod_{j=0}^{i-1} \left(2^{a2^j} + 1\right), \quad i = 0, 1, \ldots, k-1$. Then

$$M_i^{-1} \bmod m_i = 2^{a2^i - 1} - 2^{a-1} + 1, \quad i = 1, 2, ..., k. \tag{1}$$

With this choice of moduli there is no need to (pre-)compute and to store inverses. Inverse is defined by the value of a (which is the same for all moduli) and the index i. This allows reconstruction to become essentially multiplication-free: multiplication by the sparse inverse is just 2 shifts, 1 addition, and 1 subtraction. When $a = 1$ (i.e., the moduli are consecutive Fermat numbers), $M_i^{-1} \bmod m_i = 2^{2^i - 1}, \quad i = 1, 2, ...$ and multiplication by the inverse requires shift only. However, such choice of moduli does not satisfy requirement 4. In fact, the bit length of m_i is larger than the bit-length of product $m_0 m_1 \ldots m_{i-1}$.
■ An attempt to "repair" the imbalance of moduli size while preserving sparsity and scalability of inverses as in (1), naturally leads to the sets of moduli of the form $2^n \pm 2^k \pm 1$ with fixed value of n. Note, that such a "three-term" moduli were already considered and used in applications [11, 9, 7], but not in the context of modular reconstruction (these applications are mainly concerned with a single modulus of this form being prime number, and use the bit-pattern of modulus for fast operations in the corresponding finite filed).

Consider moduli $m_1 = 2^n - 2^\ell + 1, m_2 = 2^n - 2^k + 1, n > k > \ell$. To satisfy requirement 1 one can use very simple sufficient condition of co-primality of m_1, m_2: if $n \bmod (k -$

$\ell) = k \bmod (k-\ell)$ or $k \bmod (k-\ell) = 0$ then $\gcd(m_1, m_2) = 1$. This follows from the inspection of remainder sequence for m_1, m_2 while applying combined steps of binary and regular Euclidean algorithm: $2^n - 2^\ell + 1, 2^n - 2^k + 1, 2^\ell(2^{k-\ell} - 1), \ldots$ and the equality

$$(2^n - 2^k + 1) \bmod (2^{k-\ell} - 1) = 2^{n \bmod (k-\ell)} - 2^{k \bmod (k-\ell)} + 1.$$

Similar conditions hold for different choice of $+/-$ signs between terms of moduli.

Requirement 2 is easy to satisfy. Consider $m = 2^n - 2^k + 1$ and a $2n$-bit number x. Using $2^n \equiv 2^k - 1 \bmod m$ one can compute (in division/multiplication-free manner (see also [9])) $r = \mathrm{rem}(x, 2^n)$, $q = \mathrm{quo}(x, 2^n)$, $y = r + q \cdot (2^k - 1)$, obtaining number y of length about $n + k$ bits with $x \bmod m = y \bmod m$. This process can be continued until we get residue of x. If $k \leq cn$ for fixed constant c: $0 < c < 1$, then the number of iterations in this process is bounded by $\lceil \frac{1}{1-c} \rceil$, i.e., effectively bounded by constant and does not depend on n.

Now, to satisfy requirement 3 one needs to search for moduli in advance using careful inspection of application of binary and regular extended Euclidean algorithm to m_1, m_2 with fixed n and variable k, ℓ. This search is to be performed only once, and produces moduli and inverses that can be re-scaled and reused for different sizes of input. The scalability follows from simple properties of remainder sequences: if $\gcd(2^n - 2^\ell + 1, 2^n - 2^k + 1) = 1$ then for any natural a also $\gcd(2^{an} - 2^{a\ell} + 1, 2^{an} - 2^{ak} + 1) = 1$ (the remainder sequence for scaled moduli will be the same as original with all exponents scaled by the factor a). Also, if for $k > \ell + 1$ the inverse $m_2^{-1} \bmod m_1$ has sparse bit pattern, then scaling moduli the by same factor a preserves the bit pattern (again, remainder sequence in binary and regular extended Euclidean algorithm remains the same with all exponents scaled). For example, $(2^{100} - 2^{60} + 1)^{-1} \bmod (2^{100} - 2^{50} + 1) = 2^{40} + 2^{30} + 2^{20} + 2^{10} + 1$ and scaling by arbitrary natural a gives $(2^{100a} - 2^{60a} + 1)^{-1} \bmod (2^{100a} - 2^{50a} + 1) = 2^{40a} + 2^{30a} + 2^{20a} + 2^{10a} + 1$.

As for requirement 4, it is obviously satisfied, as all moduli of the form $2^n - 2^k + 1, n > k$ have the same bit-length.

Note, that after three-terms moduli satisfying requirements 1–4 are selected, one can add 2^n and $2^n + 1$ to the set of moduli, as these new moduli are relatively prime to previously selected, and also $(2^n - 2^k + 1)^{-1} \bmod 2^n = 2^k + 1$, $(2^n - 2^k + 1)^{-1} \bmod (2^n + 1) = 2^{n-k}$ and $(2^n + 1)^{-1} \bmod 2^n = 1$, i.e., inverses are sparse and scalable.

3. Two-layer modular arithmetic

In [4] an algorithm for simultaneous conversions between a given set of integers and their modular representations based on linear algebra is described. Authors provide a highly optimized implementation of the algorithm that exploits the computational features of modern processors. This implementation performance on the standard benchmark of matrix multiplication starts to deteriorate when the size of entries of randomly selected integer matrices becomes very large (2^{18} or more bits). To improve this two layer experimental modular approach was implemented by Yu Li and Benjamin Chen (University of Waterloo). The idea is to select large moduli discussed in previous section on the first layer, and reduce the problem to several problems with entries bit-size amenable for FFLAS-FFPACK. On the second layer simultaneous conversion [4] is used. Result from multiple calls to FFLAS-FFPACK are used to reconstruct the final answer using accelerated reconstruction with specially selected moduli. This approach has shown improvement of the running time of the standard benchmark by the factor between 2 and 3 for the matrices with entries having bit-size greater than 2^{18}.

4. Conclusion

Careful selection of moduli with fixed bit-pattern provides practical improvement to the standard modular algorithms. This selection (satisfying requirements 1–4) uses search with back-tracking and is based on inspection of remainder sequences in combined binary and regular extended Euclidean algorithm. There is a similarity between few-terms moduli discussed here and polynomials with few terms (such as trinomials or pentanomials) over the integers. For example, given natural $n > k > \ell$, if $n \bmod (k - \ell) = k \bmod (k - \ell)$ or $k \bmod (k - \ell) = 0$ then polynomials $x^n - x^\ell + 1$ and $x^n - x^k + 1$ are relatively prime. It is anticipated that the inspection of the structure of polynomial remainder sequences for fewnomials with unit coefficients over integers can help in the search of balanced moduli with three, five, or generally "few" terms, satisfying requirements 1–4. Note, that fewnomials over finite fields were studied extensively (see, for example [1]). However, it seems that the structure of polynomial remainder sequences of fewnomials over the ring of integers deserves additional study.

References

[1] Banegas G., Custódio R., and Panario D. A new class of irreducible pentanomials for polynomial-based multipliers in binary fields. *J Cryptogr. Eng.*, vol. 9, 2019, pp. 359–373.

[2] Bernstein D. Scaled remainder trees. *Available from* https://cr.yp.to/arith/scaledmod-20040820.pdf, 2004.

[3] Borodin A., Moenck R. Fast modular transforms. *Journal of Computer and System Sciences*, vol. 8, 1974, pp. 366–386.

[4] Doliskani J., Giorgi P., Lebreton R., Schost E. Simultaneous Conversions with the Residue Number System Using Linear Algebra. *ACM Transactions on Mathematical Software*, Volume 44, Issue 3, Article No.: 27, pp. 1–21, 2018.

[5] Garner H. The Residue Number System. *IRE Transactions, EC-8.* pp.140–147, 1959.

[6] Geddes K.O., Czapor S.R., and Labahn G. *Algorithms for Computer Algebra* (6 printing). Boston: Kluwer Academic, 1992.

[7] Granger R., Moss A. Generalized Mersenne numbers revisited. *Mathematics of Computation*, vol. 82, No. 284, 2013, pp. 2389-2420.

[8] Knuth D.E. *The Art of Computer Programming*, Vol 2.

[9] Sakai, Y., Sakurai, K. Simple Power Analysis on Fast Modular Reduction with NIST Recommended Elliptic Curves. *In: Qing, S., Mao, W., López, J., Wang, G. (eds) Information and Communications Security*, 2005, Springer LNCS, vol. 3783, pp.169–180.

[10] Schönhage A. Multiplikation großer Zahlen. *Computing* vol. 1, 1966, pp. 182–196.

[11] Solinas J. A. Generalized Mersenne Numbers. Waterloo: Faculty of Mathematics, University of Waterloo, 1999.

[12] Stewart A.M., Zima E.V. Base-2 Cunningham numbers in modular arithmetic. *Program. Comput. Software*, 2007, vol. 33, pp.80 -86.

Contributed talks

On Incomplete Rank Matrices

S.A. Abramov[1], M. Petkovšek[2,*], A.A. Ryabenko[1]

[1] *Federal Research Center "Computer Science and Control" of RAS, Russia*

[2] *University of Ljubljana, Faculty of Mathematics and Physics, Slovenia*

e-mail: sergeyabramov@mail.ru, anna.ryabenko@gmail.com

Abstract

Consider a matrix A of size $m \times n$ over a field K with $r = \operatorname{rank} A$ and $d = \min\{m, n\} - r > 0$, which implies that the rank of A is not full. We demonstrate that in such cases, it is possible to choose d elements from A such that, upon replacement of their values with other values from K, yield a matrix $\tilde{A}$ of full rank (when $m = n$, $\tilde{A}$ is nonsingular). We discuss as well the implications of this result for matrices with truncated formal series as their elements.

Keywords: matrices over fields, full and incomplete rank matrices, formal power series, formal Laurent series, truncated series

1. Introduction

Matrices are used in all areas of mathematics. The rank serves as an essential characteristic of a matrix. If K is a field and $m \times n$-matrix A over K (i.e. $A \in K^{m\times n}$), $r = \operatorname{rank} A$ then the situation of incomplete rank is possible, i.e. the situation in which $d = \min\{m, n\} - r > 0$. This is an obstacle to carrying out some transformations of the matrix A and performing calculations related to A. The case of matrices with elements in the form of truncated series is considered especially. The series themselves and matrices, whose elements are series, can be given in a truncated form, when instead of each infinite series one of its initial segments is specified. This can be viewed as an approximation of the data, or more generally, as incomplete information about the original data.

2. Rank regulation

To prove Lemma 1 below, the notion of a basic minor of a matrix A will be important. In [1] this notion is defined as follows: "The determinant of a submatrix C of order k is a basic minor if and only if it is nonzero and all submatrices of order $k + 1$ which contain C have zero determinant. The system of rows (columns) of a basic minor form a maximal linearly independent subsystem of the system of all rows (columns) of the matrix." (See also [2].)

Lemma 1. *Let K be a field, let m, n be positive integers, and let $A \in K^{m\times n}$ be a matrix with* $\operatorname{rank} A < \min\{m, n\}$. *Then*

(i) replacing any one element of the matrix A by some other element belonging to K cannot increase $\operatorname{rank} A$ *by more than* 1*;*

(ii) the matrix A contains at least one element such that its replacement by any element belonging to the field K that is not equal to it increases $\operatorname{rank} A$ *by 1.*

Proof. (i) Assertion (i) is almost trivial. Nevertheless, we give for completeness its proof:

Let $A = [a_{ij}]$ and assume that replacement of some a_{ij} by $\tilde{a}_{ij} \in K$ increases the rank of A by $\rho > 1$. Denote by $\tilde{A}$ the matrix resulting from this substitution, and choose some

*Our friend and co-author Marko Petkovšek passed away on March 24, 2023 (*S.Abramov, A.Ryabenko).*

basic minor of $\tilde{A}$ includes the element $\tilde{a}_{ij}$. The Laplace expansion of this minor along the row containing $\tilde{a}_{ij}$, is a sum of products, one of the factors in each of which is equal to zero (up to the sign, that factor is equal to the determinant of a submatrix of A of order $\operatorname{rank} A + \rho - 1 > \operatorname{rank} A$). Hence for $\rho > 1$, the minor under consideration cannot be a basic one; thus $\rho \leq 1$.

(ii) Let B be a basic minor of the matrix A. Since $\operatorname{rank} A < n$, the matrix A contains a row and a column that are not related to the minor B; let them be the i-th row and the j-th column of A. In A, replace the element a_{ij} with some $\tilde{a}_{ij} \neq a_{ij}$, and append the i-th row and the j-th column of the modified matrix $\tilde{A}$ to the set of rows and columns related to the minor B. The modified minor $\tilde{B}$ is non-zero: up to a sign, its determinant equals

$$(\tilde{a}_{ij} - a_{ij}) \det B,$$

which follows from the Laplace expansion of $\det \tilde{B}$ along the row containing $\tilde{a}_{ij}$. Thus this minor, whose order equals $\operatorname{rank} A + 1$, is nonzero. By virtue of (i), this minor is basic for $\tilde{A}$. □

A matrix element, whose replacement increases the rank of the matrix, will be called a *rank-regulating element*.

The following example shows that not every element is rank-regulating.

Example 1.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 0 & 1 & 1 \end{bmatrix}.$$

It can be seen that, for example, the element a_{33} does not affect the rank, unlike, say, a_{13}.

Proposition 1. *Let K be a field, $n > 1$ and $A \in K^{n\times n}$. Let $\operatorname{rank} A = r < n$. Then in the matrix A one can choose $n - r$ such elements that their replacement by any elements of the field K unequal to them will give a full rank matrix.*

Proof. Consider some basic minor B of the matrix A and write out the numbers $i_1, \ldots, i_{n-r}$ of the rows and the numbers $j_1, \ldots, j_{n-r}$ of the columns not related to the minor B. By using Lemma 1(ii) repeatedly, we see that the elements in the list

$$(a_{i_1,j_1}, \ldots, a_{i_{n-r},j_{n-r}}) \tag{1}$$

have the desired property. □

Of course, for $n - r > 1$ such a list will not be unique. For example, for any mapping φ of the set $\{1, \ldots, n - r\}$ onto itself, the elements of $(a_{i_1,j_{\varphi(1)}}, \ldots, a_{i_{n-r},j_{\varphi(n-r)}})$ have the desired property, too.

Example 2. Consider the following 5×5-matrix over the field of rational numbers:

$$A = \begin{bmatrix} 1 & -1 & 2 & 3 & 4 \\ 2 & 1 & -1 & 2 & 0 \\ -1 & 2 & 1 & 1 & 3 \\ 1 & 5 & -8 & -5 & -12 \\ 3 & -7 & 8 & 9 & 13 \end{bmatrix}. \tag{2}$$

Its rank is 3, and a basic minor can be obtained, for example, by selecting rows and columns with numbers 1, 3, 5. It is not hard to see that a_{22} and a_{44} are rank-regulating elements of

A. Replacing them by zeros, we obtain a matrix $\tilde{A}$ with $\det \tilde{A} = 55$; if instead we add 1 to each of the initial a_{22}, a_{44}, then for the resulting matrix $\tilde{\tilde{A}}$ we have $\det \tilde{\tilde{A}} = -11$ (obviously, $\operatorname{rank} \tilde{A} = \operatorname{rank} \tilde{\tilde{A}} = 5$).

Example 3. Consider a non-square matrix. Let A be the following 6×4-matrix over the field of rational numbers:

$$A = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 \\ 1 & 1 & 4 & 1 \\ 4 & 5 & 6 & 3 \\ 1 & -2 & 0 & 0 \\ 1 & 1 & 4 & 1 \end{bmatrix}. \tag{3}$$

Its rank is 3, and a basic minor can be obtained, for example, by selecting rows and columns with numbers 1, 2, 3.

Thus any of $a_{4,4}, a_{5,4}, a_{6,4}$ is a rank-regulating element of A. For example, replacing $a_{5,4}$ by 1 we obtain $\tilde{A}$ of full rank: $\operatorname{rank} \tilde{A} = 4 = \min\{6, 4\}$. A basic minor of $\tilde{A}$ can be obtained, for example, by selecting rows 1, 2, 3, 5 and all the matrix columns.

3. Matrices over truncated formal series

In this section, we consider the field K as the formal Laurent series field $F((x))$ over a field F. The field $F((x))$ is the quotient field of the formal power series ring $F[[x]]$. Let the elements of the matrix A be polynomials, which are considered as truncated power series. If $\det A = 0$ then A has obviously a prolongation which is a singular matrix belonging to $F[[x]]^{n \times n}$: such a prolongation can be obtained by adding to each element of A an infinite sequence of zero terms. On the other hand, using the recipe from Lemma 1 and Proposition 1, we can construct a prolongation which gives a nonsingular matrix $\tilde{A}$. To do this, we can, for example, add to each of the rank-regulating elements some terms that have degrees higher (say, by 1) than the degrees of the elements of the matrix A.

Thus, the following proposition is valid:

Proposition 2. *For an incomplete rank polynomial matrix $A = [a_{ij}] \in F[x]^{m \times n}$, there exists and can be constructed a polynomial full rank matrix $\tilde{A} = [\tilde{a}_{ij}] \in F[x]^{m \times n}$ which is a prolongation of A; wherein, if $a_{ij} = 0$ then $\tilde{a}_{ij} = 0$ or $\deg \tilde{a}_{ij} = 0$, otherwise $\deg \tilde{a}_{ij} \leq \deg a_{ij} + 1$, $i = 1, \ldots, m$, $j = 1, \ldots, n$.*

Example 4. A simple example is given by the following polynomial matrix over the field of rational numbers:

$$A = \begin{bmatrix} 1 & x \\ x & x^2 \end{bmatrix}.$$

Its rank is 1, and all its first-order minors are basic. Thus, the prolongation of any one of its elements by a non-zero term of degree 1 for a_{11}, of degree 2 for a_{12} or a_{21}, and of degree 3 for a_{22} results in a matrix of rank 2. Take, for example, the element a_{12} and add $-x^2$ to it. This gives

$$\tilde{A} = \begin{bmatrix} 1 & x - x^2 \\ x & x^2 \end{bmatrix}$$

with $\det \tilde{A} = x^3$.

Example 5. Consider the matrix (2) as truncated. Adding x to a_{22} and $-2x$ to a_{44} we get

$$\tilde{A} = \begin{bmatrix} 1 & -1 & 2 & 3 & 4 \\ 2 & 1+x & -1 & 2 & 0 \\ -1 & 2 & 1 & 1 & 3 \\ 1 & 5 & -8 & -5-2x & -12 \\ 3 & -7 & 8 & 9 & 13 \end{bmatrix}$$

with $\det \tilde{A} = 22x^2$, $\operatorname{rank} \tilde{A} = 5$.

Acknowledgements

The authors are grateful to A. Aparicio Monforte for useful comments.

References

[1] Encyclopedia of Mathematics. Retrieved 10.04.2023 from ULR: https://encyclopediaofmath.org/wiki/Minor

[2] *Shilov G.E.* Linear Algebra, revised English edition. Translated from the Russian and edited by Richard A. Silverman. Dover Publications, Inc., New York. (1977).

Power Algebra for Power Geometry

A.B. Aranson

Scientific Research Institute of Long-Range Radio Communication, Russia

e-mail: aboar@yandex.ru

Abstract

We suggest effective computation procedures for calculations by power geometry algorithms connected with Newton polyhedrons. These polyhedrons are suitable for visual explanations and graphical computations by hand in small dimension cases. But computer implementation of Newton polyhedron algorithms directly by A.D. Bruno definitions is too complicated and convoluted. Instead of geometrical definitions we use power substitution, linear inequalities and method similar to Cayley trick. We describe our methods in detail on example of calculating Puiseus expansions of solutions for Lotka-Volterra equations and then apply our methods for Euler-Poisson equations.

Keywords: power substitution, linear inequalities, Puiseux expansion, expandability

1. Introduction

We suggest effective computation procedures for calculations by power geometry algorithms connected with using of Newton polyhedrons [1] for calculating power expansions of solutions for systems of ODEs. These polyhedrons are suitable for visual explanations and graphical computations by hand in small dimension cases. But computer implementation of Newton polyhedron algorithms directly by A.D. Bruno definitions is too complicated and convoluted. Instead of geometrical definitions we use power substitution, linear inequalities and method similar to Cayley trick [2]. To calculate a vector exponent of a term directly by definitions we have to analyze every multiplier in the term that is not supported by builtin functions of CAS. Power substitution allows easy to calculate vector exponent by builtin functions of CAS and allows substitute power exponent of substituted variable in form of expression in several variables. Computer calculations of polyhedral objects (faces, normal cones, their intersections and etc) are solving of linear inequality systems but calculation of minimal and maximal power exponents after power substitution allows immediately to write linear inequality system and to avoid introducing redundant geometrical objects. We describe our methods in detail on example of calculating Puiseus expansions of solutions for Lotka-Volterra equations. Then we apply our methods for Euler-Poisson equations .

2. Lotka-Volterra system

We consider ODEs of Lotka-Volterra system [3]

$$dx/dt = kx - axy, \quad dy/dt = -ly + bxy, \tag{1}$$

where t - independent variable, x, y — dependent variables, $k, a, l, b > 0$ — parameters.

We find solutions of the system (1) in form of Puiseux series with finite nonzero principal part

$$x(t) = t^{\alpha_1}(x_0 + \sum_{j=1}^{\infty} x_j t^{j\Delta}), \quad y(t) = t^{\alpha_2}(y_0 + \sum_{j=1}^{\infty} y_j t^{j\Delta}), \tag{2}$$

where coefficients $x_0, y_0 \neq 0$, power exponents $\alpha_1, \alpha_2 < 0$ — rational, the step of the arithmetical progression of power exponents $\Delta > 0$ — rational. We move derivations in system (1) to right side and substitute into one the expansion (2) to obtain the expansion of the system (1)

$$t^{\beta_1}(c_{1,0} + \sum_{j=0}^{\infty} c_{1,j}t^{j\Delta}) = 0, \quad t^{\beta_2}(c_{2,0} + \sum_{j=0}^{\infty} c_{2,j}t^{j\Delta}) = 0, \tag{3}$$

where power exponents β_1, β_2 — rational, coefficients x_0, x_j, y_0, y_j are solutions of equations $c_{1,0}(x_0, y_0) = 0$, $c_{2,0}(x_0, y_0) = 0$, $c_{1,j}(x_j, y_j) = 0$, $c_{2,j}(x_j, y_j) = 0$.

To calculate power exponents β_1, β_2 we substitute leading terms of (2) $x_0t^{\alpha_1}$, $y_0t^{\alpha_2}$ to the system (1) and source differential equations transform to one-dimensional polynomials in variable t

$$-\alpha_1 x_0 t^{\alpha_1 - 1} + kx_0t^{\alpha_1} - ax_0y_0t^{\alpha_1+\alpha_2}, \quad -\alpha_2 y_0 t^{\alpha_2 - 1} - ly_0t^{\alpha_2} + bx_0y_0t^{\alpha_1+\alpha_2}, \tag{4}$$

and power exponents $\beta_1 = \min(\alpha_1 - 1, \alpha_1, \alpha_1 + \alpha_2)$, $\beta_2 = \min(\alpha_2 - 1, \alpha_2, \alpha_1 + \alpha_2)$. Conditions for variables α_i, β_i, $i = 1, 2$ we can write in form of weak and strict inequalities

$$\begin{array}{llll} 1) & -1 + \alpha_1 \geq \beta_1 & 4) & -1 + \alpha_2 \geq \beta_2 \\ 2) & \alpha_1 \geq \beta_1 & 5) & \alpha_2 \geq \beta_2 \\ 3) & \alpha_1 + \alpha_2 \geq \beta_1 & 6) & \alpha_1 + \alpha_2 \geq \beta_2 \end{array} \quad \begin{array}{ll} 7) & \alpha_1 < 0 \\ 8) & \alpha_1 < 0 \end{array} \tag{5}$$

and assign a unique number for each inequality in (5). To find solutions for these inequalities we introduce a vector of projective coordinates $V = (\hat{\alpha}_0, \hat{\alpha}_1, \hat{\alpha}_2, \hat{\beta}_1, \hat{\beta}_2)$, where $\hat{\alpha}_1 = \alpha_1\hat{\alpha}_0$, $\hat{\alpha}_2 = \alpha_2\hat{\alpha}_0$, $\hat{\beta}_1 = \beta_1\hat{\alpha}_0$, $\hat{\beta}_2 = \beta_2\hat{\alpha}_0$ and vector $\mathbf{0} = (0, 0, 0, 0, 0)$. Then we introduce matrix Q of coefficients of homogeneous weak inequalities and write inequalities (5) in matrix form

$$VQ \leq \mathbf{0}, \quad \text{where } Q = \begin{bmatrix} -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & -1 & 0 \\ -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \end{bmatrix}. \tag{6}$$

Columns of matrix Q coincide with inequalities (5). Appended column is condition $\hat{\alpha}_0 < 0$ for reverse inequality sign. First row of Q is constants in left side of inequalities (5). Next two rows are coefficients of α_1, α_2 in (5). Next two rows are coefficients of β_1, β_2 in (5) with opposite signes because we carry β to left side of inequalities. Turning inequality sign for expressions 7), 8) in in (5) because $\hat{\alpha}_0 < 0$ is compensated by opposite signs in 7-th and 8-th columns of matrix Q.

Method of supplementary dimension for each system of inequalities in tuple of systems is similar to Cayley trick [2]. I modified this method by appending columns to matrix Q with conditions for variables.

System (6) is solved by author computer program [4] by Motzkin-Burger algorithm. This program dismisses solution if one is strict only inequality for all inequalities with some β_i.

System‘(6) has 9 solutions. One of them is vector $V = (2\mu-2, 2-\mu+\eta_1, 2-\mu+\eta_2, 4-2\mu+\eta_1+\eta_2, 4-2\mu+\eta_1+\eta_2)$, where $0 < \mu < 1$, $\eta_1, \eta_2 > 0$. Then $\alpha_1 = (2-\mu+\eta_1)/(2\mu-2) = -1-\hat{\eta}_1$, where $\hat{\eta}_1 > 0$, because $\alpha_1 \to -\infty$ when $\mu \to 1$ and $\alpha_1 \to -1 - \eta_1/2$ when $\mu \to 0$. Similarly $\alpha_2 = -1 - \hat{\eta}_2$, where $\hat{\eta}_2 > 0$. Accordingly $\beta_1 = \beta_2 = -2 - \hat{\eta}_1 - \hat{\eta}_2$. For these values α_i, β_i inequalities 3), 6) of system (5) are equalities. Accordingly with (4) $ax_0y_0t^{-2-\hat{\eta}_1-\hat{\eta}_2} = 0$, $bx_0y_0t^{-2-\hat{\eta}_1-\hat{\eta}_2} = 0$. But its contradict to conditions $a, b, x_0, y_0 \neq 0$. Verification of such conditions was automated by author scripts for CAS Maxima [5]. Contradiction to these

conditions was exposed for other 7 solutions $V=(-1,1-\mu_1,1-\mu_2,2-\mu_1,2-\mu_2)$, $V=(-1,1+\eta,1-\mu,2+\eta,2+\eta-\mu)$, $V=(-1,1-\mu,1+\eta,2+\eta-\mu,2+\eta)$, $V=(-1,1-\mu,1,2-\mu,2)$, $V=(-1,1,1-\mu,2,2-\mu)$, $V=(-1,1+\eta,1,2+\eta,2+\eta)$, $V=(-1,1,1+\eta,2+\eta,2+\eta)$, where $0<\mu,\mu_1,\mu_2<1$, $\eta>0$.

If solution $V=(-1,1,1,2,2)$, then $\alpha_1=\alpha_2=1/-1=-1$ and $\beta_1=\beta_2=2/-1=-2$. For these values α_i,β_i inequalities 1), 3), 4), 6) of system (5) are equalities. Accordingly with (4)

$$\begin{aligned} -(-1)x_0t^{-1-1}-ax_0y_0t^{-1+(-1)}=x_0(1-ay_0)t^{-2}=0, \quad y_0=1/a,\\ -(-1)y_0t^{-1-1}+bx_0y_0t^{-1+(-1)}=y_0(1+bx_0)t^{-2}=0, \quad x_0=-1/b \end{aligned} \tag{7}$$

Then we substitute to system (1) first two terms of the expansion (2) with already calculated powers exponents and coefficients $x=-t^{-1}/b+x_1t^{-1+\Delta}$, $y=t^{-1}/a+y_1t^{-1+\Delta}$ and reduce similar terms. In result

$$\begin{aligned} (ay_1/b-x_1\Delta)t^{-2+\Delta}-kt^{-1}/b+kx_1t^{-1+\Delta}-ax_1y_1t^{-2+2\Delta},\\ (bx_1/a-y_1\Delta)t^{-2+\Delta}-lt^{-1}/a-ly_1t^{-1+\Delta}+bx_1y_1t^{-2+2\Delta}. \end{aligned} \tag{8}$$

Power exponents $-2+\Delta<-2+2\Delta$ and $-1<-1+\Delta$, so we consider terms with power exponents $-2+\Delta$ and -1 only. If step $0<\Delta<1$ then coefficients x_1,y_1 are solutions of the homogenious linear algebraic equations system $-\Delta x_1+(a/b)y_1=0$, $(b/a)x_1-\Delta y_1=0$, but this system doesn't have solutions if $\Delta\neq\pm1$. If step $\Delta=1$ then coefficients x_1,y_1 are solutions of the linear algebraic equations system $-x_1+(a/b)y_1=k/b$, $(b/a)x_1-y_1=l/a$. This system has solution $y_1=(bx_1-l)/a$, where x_1 is arbitrary coefficient, if $k=-l$ only that contradict to condition $k,l>0$. Condition $k=-l$ we call *expandability condition* into Puiseux series.

Solution of Lotka-Volterra system is expandable to Puiseux series with not allowed conditions for parameters.

3. Euler-Poisson equations

Motion of a rigid body with a fixed point is described by Euler-Poisson equations [6]

$$\begin{aligned} Adp/dt=(B-C)qr-Mg(z_0\gamma_2-y_0\gamma_3), \quad d\gamma_1/dt=r\gamma_2-q\gamma_3,\\ Bdq/dt=(C-A)rp-Mg(x_0\gamma_3-z_0\gamma_1), \quad d\gamma_2/dt=p\gamma_3-r\gamma_1,\\ Cdr/dt=(A-B)pq-Mg(y_0\gamma_1-x_0\gamma_2), \quad d\gamma_3/dt=q\gamma_1-p\gamma_2, \end{aligned} \tag{9}$$

where t - time, A,B,C - principal moments of inertia, which satisfy triangle inequalities $A>0,B>0,C>0,A+B\geq C,A+C\geq B,B+C\geq A$, Mg - the body weight, x_0, y_0, z_0 - coordinates of the center of gravity of the rigid body in the body frame, p,q,r - projections of the angular velocity vector onto the body frame axes, $\gamma_1,\gamma_2,\gamma_3$ - direction cosines of the vertical in the body frame. System (7) has three general first integrals

$$\begin{aligned} Ap^2+Bq^2+Cr^2-2Mg(x_0\gamma_1+y_0\gamma_2+z_0\gamma_3)=h=\text{const},\\ Ap\gamma_1+Bq\gamma_2+Cr\gamma_3=l=\text{const}, \quad \gamma_1^2+\gamma_2^2+\gamma_3^2=1. \end{aligned} \tag{10}$$

These are energy, momentum and geometry integrals.

We find solutions of systems (9)(10) in form of Puiseux series with finite nonzero principal

part

$$p(t) = t^{\alpha_1}(p_0 + \sum_{j=1}^{\infty} p_j t^{j\Delta}), \qquad q(t) = t^{\alpha_2}(q_0 + \sum_{j=1}^{\infty} q_j t^{j\Delta}), \qquad r(t) = t^{\alpha_3}(r_0 + \sum_{j=1}^{\infty} r_j t^{j\Delta}),$$
$$\gamma_1(t) = t^{\alpha_4}(\gamma_{1,0} + \sum_{j=1}^{\infty} \gamma_{1,j} t^{j\Delta}), \quad \gamma_2(t) = t^{\alpha_5}(\gamma_{2,0} + \sum_{j=1}^{\infty} \gamma_{2,j} t^{j\Delta}), \quad \gamma_3(t) = t^{\alpha_6}(\gamma_{3,0} + \sum_{j=1}^{\infty} \gamma_{3,j} t^{j\Delta}), \tag{11}$$

where coefficients $p_0, q_0, r_0, \gamma_{1,0}, \gamma_{2,0}, \gamma_{3,0} \neq 0$, power exponents $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6 < 0$ — rational, the step of the arithmetical progression of power exponents $\Delta > 0$ — rational.

We apply algorithms and programs described and demonstrated above for Lotka-Volterra system. Matrix Q of coefficients of inequalities has dimension 16×43 (7 variables + 6 equations + 3 integrals) $\times$ (21 terms of equations (9) + 15 terms of integrals (10) + 6 conditions $\hat{\alpha}_1, \ldots, \hat{\alpha}_6 > 0$ + condition $\hat{\alpha}_0 < 0$). Solvability conditions of equations for expansion coefficients are conditions for parameters of system (9) (expandability conditions). Our calculations show that all known today integrability conditions for Euler-Poisson system [7] are expandability conditions. Grioly solution is entire functions and is not expanded to considered Puiseux series and our calculations are not correct in this case, but Grioly condition appeares. Also, we calculated new cases of expandability for Euler-Poisson system.

4. Conclusion

We calculated all known integrability conditions for Euler-Poisson system that confirm correctness described here algorithms and computer programs. We believe calculations of expansions for Euler-Poisson system may be good task for testing and benchmarking of computer implementations of power geometry algorithms.

References

[1] *Bruno A.D.* Power Geometry in Algebraic and Differential Equations), Amsterdam: Elsevier, 2000

[2] *Huber B., Rambau J., Santos F.* The Cayley Trick, lifting subdivisions and the Bohne-Dress theorem on zonotopal tilings. Journal of the European Mathematical Society. 2. 179-198. (2000).

[3] *Arnold V.I.* Ordinary Differential Equations. Springer-Verlag. (1992).

[4] *Aranson A.B.*T Computation of Collections of Correlate Faces for Several Polyhedrons, in Computer Algebra in Scientific Computing, Munchen: Techn, Univ. Munchen, pp. 13-17. 2003

[5] *Aranson A.B.* Calculation of power expansion solutions of N. Kowalewski modified ODE system by power geometry algorithm, System Programming and Computer Software 37, 87-98. (2011).

[6] *Golubev V.V.* Lectures on Integration of Equations Motion of a Heavy Rigid Body near Fixed Point, Moscow: GITTL, 1953. (In Russian)

[7] *Gashenenko I.N., Gorr G.V., Kovalev A.M.* Classical Problems of the Dynamics of a Rigid Body, Kiev, Naukova Dumka, 2012. (In Russian)

On Computation of Power Transformations

A.A. Azimov[1], A.D. Bruno[2]

[1]*Department of Algebra and Geometry, Samarkand State University after Sh. Rashidov, Uzbekistan*

[2]*Keldysh Institute of Applied Mathematics of RAS, Russia*

e-mail: azimov_alijon_akhmadovich@mail.ru, abruno@keldysh.ru

Abstract

An algorithm for solving the following problem is described. Let $m < n$ integer vectors in the n-dimensional real space be given. Their linear span forms a linear subspace L in $\mathbb{R}^n$. It is required to find a unimodular matrix such that the linear transformation defined by it takes the subspace L into a coordinate subspace. Computer programs implementing the proposed algorithms and the power transforms for which they are designed are described.

Keywords: continued fraction, unimodular matrix, Euler algorithm, power transformation

1. Introduction

Recall that a square matrix is said to be unimodular if all its elements are integers and its determinant equals ± 1. Its inverse is also unimodular.

We will write vectors as row vectors $A = (a_1, \dots, a_n)$, and $[a]$ is the integer part of the real number a.

Problem 1. Let m, $(m < n)$ integer vectors $A_1, \dots, A_m$ be given in the n-dimensional real space $\mathbb{R}^n$. Their linear span

$$L = \left\{ X = \sum_{j=1}^{m} \lambda_j A_j,\ \lambda_j \in \mathbb{R}, \quad j = 1, \dots, m \right\} \tag{1}$$

forms a linear subspace in $\mathbb{R}^n$. It is required to find a unimodular matrix α such that the transformation $X\alpha = Y$ takes L to the coordinate subspace

$$M = \{Y : y_{n-l+1} = \cdots = y_n = 0\},$$

where $l = \dim L$.

In this talk, we give an algorithm for solving this problem and provide its implementations in computer algebra systems [1]. If $n = 2$ and $m = 1$, then Problem 1 is solved by Eucledean algorithm or by continued fraction [2]. In Section 2, we describe the Euler algorithm [3], which generalizes the *Euclidean algorithm* (i.e., the *continued fraction algorithm*) to the n-dimensional integer vector. In Section 3 we describe a solution of Problem 1. In Section 4 we consider power transformations, for the calculation of the unimodular matrices of which, all these algorithms are developed.

2. Euler's algorithm and a generalization of continued fraction

Problem 2. Let an n-dimensional integer vector $A = (a_1, a_2, \dots, a_n)$ be given. Find an n-dimensional unimodular matrix α such that the vector $A\alpha = C = (c_1, \dots, c_n)$ contains only one nonzero component c_n.

Euler proposed the following algorithm for solving this problem [3]. Suppose for the time being that all components of vector A are nonzero. Using the permutation $A\alpha_0 = (\widetilde{a}_1, \widetilde{a}_2, \ldots, \widetilde{a}_n)$ arrange its components in nondecreasing order $\widetilde{a}_j \le \widetilde{a}_{j+1}$, $j = 1, \ldots, n-1$. Hereα_0 is the unimodular matrix of the permutation. Let $\widetilde{a}_k$ be the least number among $\widetilde{a}_j$ that is distinct from zero.

Let $b_j = [\widetilde{a}_j/\widetilde{a}_k]$, $j = 1, \ldots, n$. Here $b_1 = \cdots = b_{k-1} = 0$, $b_k = 1$. Make the transformation

$$d_j = \widetilde{a}_j - b_j\widetilde{a}_k,\ 1 \le j \le n,\ j \ne k, d_k = \widetilde{\alpha}_k. \tag{2}$$

It is associated with the unimodular matrix α_1 the diagonal of which consists of ones, and the k-th row is

$$0, 0, \ldots, 0, 1, -b_{k+1}, \ldots, -b_n, \text{ i.e. } \tilde{A}\alpha_1 = D = (d_1, \ldots, d_n).$$

Now arrange the components of the vector D in non-decreasing order using the unimodular permutation matrix β_0 so that $D\beta_0 = \tilde{D} = \left(0, \ldots, 0, \widetilde{d}_k, \ldots, \widetilde{d}_n\right)$,where $\widetilde{d}_j \le \widetilde{d}_{j+1}$.

Let $\widetilde{d}_l$ be the least of $\widetilde{d}_j$, distinct from zero, and let $e_j = \left[\widetilde{d}_j/\widetilde{d}_l\right]$, $j = 1, \ldots, l$. Make the transformation

$$f_j = \widetilde{d}_j - e_j\widetilde{d}_l, \quad 1 \le j \le n, \quad j \ne l, \quad f_l = \widetilde{d}_l,$$

and soon. At each step, the maximum of the components of the vector decreases and it is the n-th component. Therefore, in a finite number of steps we obtain a vector with the only (last) nonzero component. This component equals the GCD of all original components $a_1, \ldots, a_n$. Each step involves a permutation matrix and a triangular matrix with the unit diagonal:

$$A\alpha_0\alpha_1\beta_0\beta_1\gamma_0\gamma_1 \ldots \omega_0\omega_1 = A\alpha = C = (0, \ldots, 0, c_n).$$

The matrix

$$\alpha = \alpha_0\alpha_1\beta_0\beta_1\gamma_0\gamma_1 \cdots \omega_0\omega_1 \tag{3}$$

is a solution of Problem 2.

If not all components a_j of the original vector A have the same sign, then we first arrange them in non-decreasing order of their moduli $|\widetilde{a}_j| \le |\widetilde{a}_{j+1}|$ and set $b_j = [|\widetilde{a}_j| \,/\, |\widetilde{a}_k|] \operatorname{sign} \widetilde{a}_j \operatorname{sign} \widetilde{a}_k$.

Let the given vector A be perpendicular to a linear variety. Then, after the transformation using the matrix α, we obtain the vector in which all first $n-1$ components are zero. Therefore, the last component of all vectors of the original variety will be zero after this transformation.

Euler's algorithm generalizes the continued fraction algorithm only for integer vectors. Such a generalization for arbitrary real vectors was sought by all major mathematicians of the 19th century, but without success. Such a generalization of the continued fraction algorithm for the n-dimensional vector was proposed in [4]. It gives a sequence of best approximations, and it is periodic if all the components of the original vector are roots of a polynomial of degree n with integer coefficients.

3. Solution to Problem 1

Let integer vectors

$$\begin{aligned} A_1 &= (a_{11}, a_{12}, \ldots, a_{1n}), \\ A_2 &= (a_{21}, a_{22}, \ldots, a_{2n}), \\ &\ldots \\ A_m &= (a_{m1}, a_{m2}, \ldots, a_{mn}) \end{aligned} \tag{4}$$

$(m < n)$ and a linear space (1) be given.

First, we check if there are identical vectors among them. If there are any, we discard duplicates and leave only one of them. Now, we are sure that all vectors (4) are different. Apply Euler's algorithm to the vector A, i.e., calculate the matrix α such that $A_1\alpha_0 = C_1 = c_nE_n$, where c_n is an integer and E_k is the k-th unit vector.

Let $A_j\alpha_0 = C_j = (c_{j1}, \ldots, c_{jn})$, $j = 2, \ldots, m$. Set $A_j^1 = (c_{j1}, \ldots, c_{jn-1})$, $j = 2, \ldots, m$. Apply Euler's algorithm to the $(n-1)$-dimensional vector A_2^1 to obtain $A_2^1\alpha_1 = C_2^1 = (0, 0, \ldots, c_{n-1}^1)$, where α_1 is an $(n-1)$-dimensional square matrix. Let

$$A_j^1\alpha_1 = C_j^1 = (c_{j1}^1, \ldots, c_{jn-1}^1), \quad j = 3, \ldots, m.$$

Apply Euler's algorithm to the $(n-2)$-dimensional vector C_3^1, and so on. Finally, we obtain the sequence of matrices $\alpha_0, \alpha_1, \ldots, \alpha_{m-1}$ of decreasing size $n, n-1, \ldots, n-m+1$. Form the block matrices

$$\beta_j = \begin{pmatrix} \alpha_j & 0 \\ 0 & I_{j+1} \end{pmatrix}, \quad j = 0, \ldots, n-m,$$

of size n, where I_{j+1} are the identity matrices of size $j+1$. Set $\gamma = \beta_0\beta_1 \cdots \beta_{m-1}$. Then

$$A_j\gamma = (0, 0, \ldots, 0, w_{j,n-j+1}, \ldots, w_{j,n}) = W_j, \quad j = 1, \ldots, m.$$

The matrix γ is a solution to Problem 1.

4. Power transformations

Let the polynomial

$$f(X) = \sum f_Q X^Q, Q \in \mathbf{S}, \tag{5}$$

where $X = (x_1, \ldots, x_n) \in \mathbb{R}^n$ or $\mathbb{C}^n$, $Q = (q_1, \ldots, q_n) \in \mathbb{Z}^n, Q \geq 0, f_Q$ are constant coefficients from $\mathbb{R}$ or $\mathbb{C}$, $\mathbf{S} = \mathbf{S}(f)$ is the support of f, be given. Let $\mathcal{F}$ be the algebraic variety $f(X) = 0$ and the point $X = X^0 \in \mathcal{F}$.

If X^0 is a simple point, i.e., if at least one derivative $\partial f/\partial x_j$ is nonzero at X^0 then the implicit function theorem implies that the variety $\mathcal{F}$ in the neighborhood of X^0 is described by the equation

$$\Delta x_j = \varphi(\Delta x_1, \ldots, \Delta x_{j-1}, \Delta x_{j+1}, \ldots, \Delta x_n), \tag{6}$$

where $\Delta x_k = x_k - x_k^0$ and φ is a convergent series of its arguments.

If X^0 is not a simple point, then, according to [5, 6] we can seek the branches of the variety $\mathcal{F}$, passing through X^0 in the form of parametric expansions

$$\Delta x_j = \varphi_j(\xi_1, \ldots, \xi_{n-1}), i = 1, \ldots, n, \tag{7}$$

where ξ_k are small parameters and φ_j — are converging power series. To this end the convex hull Γ of the support $\mathbf{S}$ in the space is constructed. Then, Γ is the polyhedron the boundary $\partial\Gamma$ of which consists of (generalized) faces $\Gamma_j^{(d)}$ of dimension d, $0 \leq d < n$. Here j is the face index. Since all vertices $\Gamma_j^{(0)}$ of Γ are integer, each face $\Gamma_j^{(d)}$ has $n-d$ integer linearly independent normals $N_{j1}^{(d)}, \ldots, N_{jn-d}^{(d)} \in \mathbb{R}_*^n$ i.e., normals belonging to the space $\mathbb{R}_*^n$, which is dual of the space $\mathbb{R}^n$.

In addition, each face $\Gamma_j^{(d)}$ is associated with the boundary set

$$D_j^{(d)} = \left\{ Q \in \mathbf{S} \cap \Gamma_j^{(d)} \right\},$$

and the truncated sum is

$$\hat{f}_j^{(d)}(X) = \sum f_Q X^Q \text{ over } Q \in D_j^{(d)}. \tag{8}$$

Theorem 1 ([5, Corollary in Chapter II, § 3], [6, Theorem 3.1]). *For the face* $\Gamma_j^{(d)}$ *there exists a power transformation*

$$\ln Y = \ln X \cdot \alpha,$$

where $\ln Y = (\ln y_1, \ldots, \ln y_n)$ *and* $\ln X = (\ln x_1, \ldots, \ln x_n)$ *with a unimodular matrix* α*, that takes the truncated sum* (8) *to a polynomial* g *of* d *variables, i.e.,*

$$\hat{f}_j^{(d)}(X) = Y^T g(y_1, \ldots, y_d), \tag{9}$$

where $T = (t_1, \ldots, t_n) \in \mathbb{Z}^n$.

However in [5, 6], it was not pointed out how the unimodular matrix α can be calculated. This is done in the current paper. In [7, Part I, Ch. I, Section 1.9] it was made for $n = 2$. In [1, 8] we describe software of these algorithms. It will be considered in our talk.

References

[1] *Bruno, A.D., Azimov, A.A.* Computing unimodular matrices of power transformations // Programming and Computer Software, 2023, Vol. 49, No. 1, pp. 32–41. DOI: 10.1134/S0361768823010036

[2] *Khinchin, A. Ya.* Continued Fractions, Moscow: Fizmatgiz, 1961 (in Russian); Mineola, NY: Dover, 1997.

[3] *Euler L.* De relatione inter ternas pluresve quantitates instituenda // 1785, All Works 591.

[4] *Bruno, A.D.* Computation of the fundamental units of number rings using a generalized continued fraction, Program. Comput. Software, 2019, vol. 45, no. 2, pp. 37–50. DOI: 10.1134/S036176881902004X.

[5] *Bruno, A.D.* Power Geometry in Algebraic and Differential Equations, Moscow: Nauka, 1998 (in Russian); Amsterdam: Elsevier, 2000.

[6] *Bruno, A.D., Batkhin, A.B.* Resolution of an algebraic singularity by power geometry algorithms, Program. Comput. Software, 2012, vol. 38, no. 2, pp. 57–72.

[7] *Bruno, A.D.* Local Methods in Nonlinear Differential Equations, Berlin: Springer, 1989.

[8] *Bruno, A.D., Azimov, A.A.* Computation of Unimodular Matrices // Preprints of KIAM, Moscow, No 46, 2022 (in Russian). DOI: `https://doi.org/10.20948/prepr-2022-46`

Structure of Resonant Variety in Hamiltonian Systems with Three Degrees of Freedom*

A.B. Batkhin[1,2], Z.Kh. Khaydarov[3]

[1]*Keldysh Institute of Applied Mathematics of RAS, Russia*

[2]*Department of Theoretical Mechanics, Moscow Institute of Physics and Technology, Russia*

[3]*Department of Algebra Geometry, Samarkand State University named after Sh. Rashidov, Uzbekistan*

e-mail: batkhin@gmail.com, zafarxx@gmail.com

Abstract

For elementary singular point of a multiparameter Hamiltonian system we discuss a method of computing the condition of existence of a resonance of arbitrary order and multiplicity. For a certain resonant vector this condition defines a resonant variety as a variety in the space of coefficients of the characteristic polynomial of the linear part of the Hamiltonian system. By means of computer algebra and power geometry techniques polynomial parametrization of the resonant variety is proven. The obtained results can be used to investigate the formal stability regions of the equilibrium of a Hamiltonian multiparameter system as well as for the asymptotic integration of its normal form.

Keywords: Hamiltonian system, resonance, polynomial parametrization, formal stability

1. Problem setting

In the generic case, an analytic time-independent Hamiltonian function $H(\mathbf{z})$ in the vicinity of the *stationary point* (SP), coinciding with the origin, is expanded into a convergent series of homogeneous polynomials H_k of degree k of its phase variables $\mathbf{z} = (\mathbf{x}, \mathbf{y})$

$$H(\mathbf{z}; \mathbf{P}) = \sum_{k=2}^{\infty} H_k(\mathbf{z}; \mathbf{P}), \tag{1}$$

where $\mathbf{P}$ is a vector of parameters.

In common case the series (1) starts with the quadratic Hamiltonian $H_2(\mathbf{z}; \mathbf{P})$ defining the local dynamics near the SP. The behavior of the phase flow in the first approximation is described by a linear Hamiltonian system

$$\dot{\mathbf{z}}(t) = B(\mathbf{P})\mathbf{z}, \quad B(\mathbf{P}) = \frac{1}{2} J \frac{\partial^2 H_2(\mathbf{P})}{\partial \mathbf{z} \partial \mathbf{z}}.$$

All eigenvalues λ_j, $j = 1, \ldots, 2n$, of the matrix B can be reordered in such a way that $\lambda_{j+n} = -\lambda_j$, $j = 1, \ldots, n$. Denote by vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$ the set of *basic eigenvalues*. The characteristic polynomial $\check{f}(\lambda)$ of the matrix B contains only even powers of λ, so it is a polynomial in $\mu = \lambda^2$. The following polynomial is called *semi-characteristic*

$$f(\mu) = \sum_{k=0}^{n} a_{n-k}(\mathbf{P}) \mu^k, \quad a_0 \equiv 1. \tag{2}$$

*This extended abstract is based on our papers [1, 2].

The *normal form* (NF) of a system of ordinary differential equations (ODE) computed in the vicinity of an invariant set (stationary point, periodic solution, or invariant torus) is a powerful tool for analyzing the local dynamics of phase flow in the neighborhood of such an invariant structure. Even though the NF is a formal object, it can be used for finding first integrals of the system, families of periodic solutions, for investigating integrability, stability, and bifurcations (for details see [3, 4]).

Theorem 1 ([5, § 12])**.** *There exists a canonical formal transformation in the form of power series, which reduces the initial system* (1) *into its normal form*

$$\dot{\mathbf{u}} = \partial h/\partial \mathbf{v}, \quad \dot{\mathbf{v}} = -\partial h/\partial \mathbf{u}$$

defined by the normalized Hamiltonian $h(\mathbf{u},\mathbf{v}) = \sum_{j=1}^{n} \lambda_j u_j v_j + \sum h_{\mathbf{pq}} \mathbf{u}^{\mathbf{p}} \mathbf{v}^{\mathbf{q}}$ *containing only resonant terms* $h_{\mathbf{pq}} \mathbf{u}^{\mathbf{p}} \mathbf{v}^{\mathbf{q}}$ *with*

$$\langle \mathbf{p} - \mathbf{q}, \boldsymbol{\lambda} \rangle = 0. \tag{3}$$

Here $0 \leqslant \mathbf{p}, \mathbf{q} \in \mathbb{Z}^n$, $|\mathbf{p}| + |\mathbf{q}| \geqslant 2$ *and* $h_{\mathbf{pq}}$ *are constant coefficients.*

The terms $h_{\mathbf{pq}} \mathbf{u}^{\mathbf{p}} \mathbf{v}^{\mathbf{q}}$ that have $\mathbf{p} = \mathbf{q}$ are called *secular*, all others are called *strong resonant.*

Definition 1. *Resonance multiplicity* $\mathfrak{k}$ is the number of linearly independent solutions $\mathbf{p} \in \mathbb{Z}^n$ of the resonance equation $\langle \mathbf{p}, \boldsymbol{\lambda} \rangle = 0$. *Resonance order* is $\mathfrak{q} = \min |\mathbf{p}|$ by $\mathbf{p} \in \mathbb{Z}^n$, $\mathbf{p} \neq 0, \langle \mathbf{p}, \boldsymbol{\lambda} \rangle = 0$. If the solution of the resonance equation contains only two eigenvalues, then such resonance is called *two-frequency resonance*, if more than two, then it is called *multifrequency resonance.* Resonances with orders 2, 3 or 4 are called *strong resonances*

Definition 2. A variety $\mathcal{R}_n^{\mathbf{p}}$ in the space $\mathbf{K}$ of the coefficients $a_1, \ldots, a_n$ of the semi-characteristic polynomial $f_n(\mu)$ of degree n is called *resonant variety*, where the vector of basic eigenvalues $\boldsymbol{\lambda}$ of the corresponding characteristic polynomial $\check{f}(\lambda)$ is a nontrivial solution of the resonance equation $\langle \mathbf{L}, \boldsymbol{\lambda} \rangle = 0$ for a fixed integer vector $\mathbf{p}$. The analytic representation of the variety $\mathcal{R}_n^{\mathbf{p}}$ in implicit or parametric forms is denoted below by $R_n^{\mathbf{p}}$.

For a multiparameter Hamiltonian system with 3 degrees of freedom, give a description of regions in the system parameter space, in which there are no strong resonances:

order 2**:** of order $\mathfrak{q} = 2$: $\mathbf{p} = (1, 1, 0)$ is the case of multiple roots, which is described by the discriminant set $R_3^{(1,1,0)} \equiv D(f) = 0$;

order 3**:** of order $\mathfrak{q} = 3$: for the 2-frequency case $\mathbf{p} = (2, 1, 0)$, described by the q-discriminant $R_3^{(2,1,0)} \equiv D_4(f) = 0$; for 3-frequency case: described by the condition $R_3^{(1,1,1)} = 0$;

order 4**:** of order $\mathfrak{q} = 4$: for the 2-frequency case $\mathbf{p} = (3, 1, 0)$, described by the q-discriminant $R_3^{(3,1,0)} \equiv D_9(f) = 0$; for 3-frequency case: described by the condition $R_3^{(2,1,1)} = 0$.

To solve this problem, we should obtain a description of boundaries of the regions, which are free of strong resonances. These boundaries consist of parts of algebraic varieties on which the resonance equation (3) has a nontrivial solution.

Let us decompose the main problem into several auxiliary problems.

1. Obtain an analytic representation in the coefficient space $\mathbf{K} = (a_1, a_2, a_3)$ of the cubic polynomial of resonant varieties $\mathcal{R}_3^{\mathbf{p}}$ for all vectors $\mathbf{p}$ orders 2, 3 and 4.

2. Find the mutual location of all resonant varieties found above.

2. Condition on resonance existence

We considered two ways of computing condition of resonance existence for a given resonant vector $\mathbf{p}^*$.

- The first method allows us to obtain an *implicit representation* of the variety $\mathcal{R}_3^{\mathbf{p}}$.
- The second method allows to obtain a *parametric representation* of the variety $\mathcal{R}_3^{\mathbf{p}}$.

In each of these methods one should first find the resonant relation between the roots μ_j of the polynomial $f(\mu)$ for a given vector $\mathbf{p}^*$.

A general description of the procedure for obtaining condition on the existence of two and multi-frequency resonances is as follows:

1. for a certain vector $\mathbf{p}^* = (r, q, 1)$, where $r, q \in \mathbb{Q}$, $r, q \neq 0$, satisfying the resonance equation $\langle \mathbf{p}, \boldsymbol{\lambda} \rangle = 0$, a polynomial ideal is composed $\mathcal{J} = \left\{ \langle \mathbf{p}^*, \boldsymbol{\lambda} \rangle, \lambda_j^2 - \mu_j, j = 1, \ldots, n \right\}$;
2. Gröbner basis $\mathcal{G}$ of this ideal with the elimination monomial order of variables $\lambda_j, \mu_j, j = 1, \ldots, n$ is computed. The first polynomial $\mathfrak{g}_1$ of $\mathcal{G}$ is a quasi-homogeneous polynomial in the variables $\mu_j, j = 1, \ldots, n$. Its zeroes determine the condition of existence of resonance for a given vector $\mathbf{p}^*$.

This condition takes the form

$$R_3^{(r,q,1)}(\mu_j) \equiv q^4\mu_2^2 - 2q^2r^2\mu_1\mu_2 + r^4\mu_1^2 - 2q^2\mu_2\mu_3 - 2r^2\mu_1\mu_3 + \mu_3^2 = 0. \tag{4}$$

To obtain the corresponding resonant condition in the implicit form as zeroes of a polynomial with coefficients $a_j, j = 1, \ldots, 3$ of the polynomial $f(\mu)$, a new Gröbner basis $\mathcal{F}$ of the ideal is constructed. This method turns out to be very time-consuming for resonances of the general form, it leads to very cumbersome expressions. Its generalization for cases with degrees of freedom greater than 3 is not possible.

For condition (4) a power transformation $\mu_1 = s_2s_3$, $\mu_2 = s_1s_3$, $\mu_3 = s_3$ is done. It reduces $R_3^{(r,q,1)}(\mu_j)$ into a polynomial of two variables

$$\tilde{R}_3^{(r,q,1)} \equiv q^4s_1^2 - 2q^2r^2s_1s_2 + r^4s_2^2 - 2q^2s_1 - 2r^2s_2 + 1 = 0,$$

which has the parametric representation of the roots

$$\mu_1 = \left(r^2u(q+1) + q - 1\right)^2 v,\ \mu_2 = \left(r^2u - 1\right)^2 vr^2,\ \mu_3 = \left(r^2u + 2q - 1\right)^2 vr^2.$$

This parametric representation using elementary symmetric polynomials gives a polynomial parametrization of the coefficients

$$\begin{aligned}
a_1 &= -v\left[r^4(2 + (q+1)^2)u^2 + 2(q-1)r^2(2r^2 + q + 1)u + 2r^2(2q^2 - 2q + 1) + (q+1)^2\right],\\
a_2 &= v^2r^2[r^8\left(r^2 + (q+1)^2\right)u^4 + 4(q-1)r^6\left(r^2 + q^2 + 3q + 2\right)u^3 +\\
&\quad + r^4\{\left(4q^2 - 12q + 6\right)r^2 + 4q^4 + 12q^3 - 8q^2 - 12q + 12\}u^2 +\\
&\quad - 4(q-1)r^2\left((2q-1)r^2 - 2q^3 - q^2 + 3q - 2\right)u + (2q-1)^2r^2 + 2(2q^2 - 2q + 1)(q-1)^2],\\
a_3 &= -r^4\left(r^2u(q+1) + q - 1\right)^2 v^3\left(r^2u - 1\right)^2\left(r^2u + 2q - 1\right)^2.
\end{aligned}$$

Excluding the parameters u, v we can obtain an implicit representation of $R_3^{\mathbf{p}^*}$ of the condition for the existence of resonance via the coefficients a_j of the polynomial. This expression is not given here due to its cumbersomeness: it is a quasi-homogeneous polynomial consisting of 19 monomials.

For each strong resonance of orders 2, 3 and 4 parametric representation of the corresponding variety was obtained. Their mutual location is shown in Fig. 1.

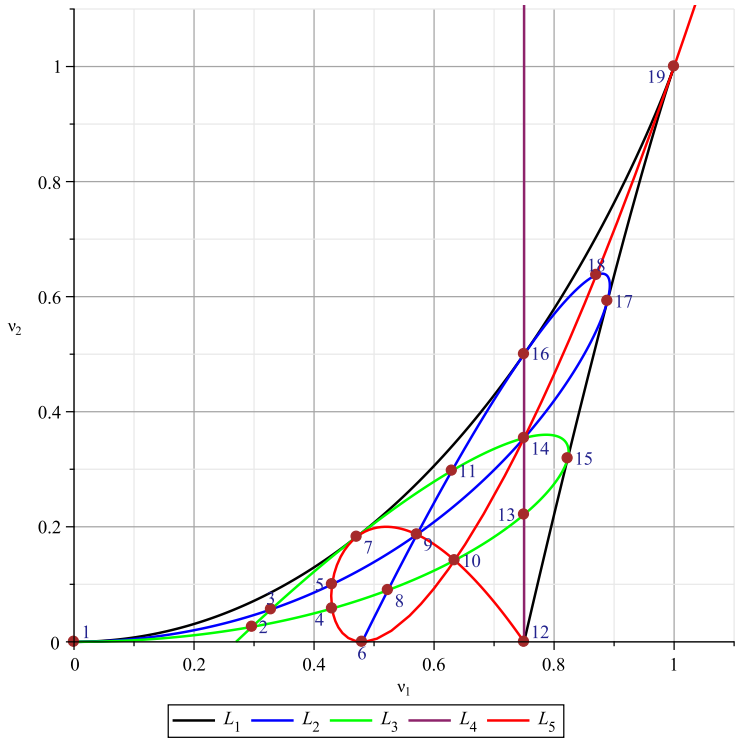


Figure 1: Resonant varieties in parametric variables.

References

[1] Batkhin A. B., Khaydarov Z.K. Strong resonances in nonlinear Hamiltonian system // *KIAM Preprints*. 2022. Vol. 59. doi: 10.20948/prepr-2022-59 (in Russian).

[2] Batkhin A. B., Khaydarov Z. Kh. Calculation of a Strong Resonance Condition in a Hamiltonian System // *Computational Mathematics and Mathematical Physics*. 2023. Vol. 63, no. 5. P. 697–714. doi: 10.31857/S0044466923050071 (in Russian).

[3] Bruno A. D. The Restricted 3-body Problem: Plane Periodic Orbits. Berlin : Walter de Gruyter, 1994. 362 p. = Nauka, Moscow, 1990. 296 p. (in Russian).

[4] Zhuravlev V. F., Petrov A. G., Shunderyuk M. M. Selected Problems of Hamiltonian Mechanics. Moscow : LENAND, 2015. P. 304. (in Russian).

[5] Bruno A. D. Analytical form of differential equations (II) // *Trans. Moscow Math. Soc.* 1972. Vol. 26. P. 199–239. = Trudy Moskov. Mat. Obsc. 25 (1971) 119–262 (in Russian).

The First Differential Approximation on the Example of the Van der Pol Oscillator

Y. A. Blinkov
Saratov State University, Russia
e-mail: blinkovua@info.sgu.ru

Abstract

Systems of ordinary differential equations depending on parameters is considered, using the Van der Pol oscillator as an example. Advantages of the first differential approximation method and its implementation in the computer algebra systems are discussed. It is shown that, the presented method allows to estimate the stiffness of the Van der Pol oscillator and error of numerical methods and to propose simple criteria for choosing a step in calculations. The presented implementation of the method use a standard tools of computer algebra and can be applied systems with a polynomial right-had side.

Keywords: computer algebra, first differential approximation, stiff equation

1. Introduction

In the 60s of the last century, N.N. Yanenko [1] formulated the differential approximations method to investigate difference schemes. The main idea of this method is to replace the investigation of the properties of a difference scheme by the investigation of some problem with differential equations occupying an intermediate position between the original differential problem and the difference scheme approximating it.

First Differential Approximation (FDA) for PDEs of evolutionary type and in particular the Korteweg-de Vries equation using computer algebra systems is discussed in [2].

In [3] FDA is reviewed for difference schemes describing ordinary differential equations. The connection between the singular perturbation of the original system and the concept of FDA is discussed. For this simple case, a relationship is shown between the method for estimating the approximation error of the solution based on the FDA analysis and the Richardson-Kalitkin method. It should be noted, that a consistent system of PDEs can be approximate by inconsistent difference systems of equations. Examples are given in [3]. As a way to check the consistency of a system of difference equations, it is proposed to check the consistency of the FDA for a difference system. The issues of FDA calculation in computer algebra systems, Sage and SymPy are considered.

This paper considers systems of ordinary differential equations (ODE) depending on parameters, using the Van der Pol oscillator as an example.

There are many numerical methods for solving ODE. The usage of the FDA allows to get information about the quality of the selected numerical method for a particular system using only symbolic calculations. In this paper, Runge-Kutta methods and some multi-step methods will be considered.

2. FDA for Van der Pol oscillator

Let's write Van der Pol oscillator[4]

$$u_{tt} - \mu(1 - u^2)u_t + u = 0 \tag{1}$$

as a system of two equations

$$\begin{cases} u_t - u_1 = 0, \\ u_{1t} - \mu(1-u^2)u_1 + u = 0. \end{cases} \tag{2}$$

Applying the fourth-order Runge-Kutta method [5] to (2) we get

$$\begin{cases} u_1 + h\left(-\dfrac{\mu u_1 (u-1)(u+1)}{2} - \dfrac{u}{2}\right) + \ldots = 0, \\ -\mu u_1 (u-1)(u+1) - u + h\left(\dfrac{\mu^2 u_1 (u-1)^2 (u+1)^2}{2} + \right. \\ \qquad \left. + \dfrac{\mu u (u^2 - 2u_1^2 - 1)}{2} - \dfrac{u_1}{2}\right) + \ldots = 0. \end{cases} \tag{3}$$

Taylor expansion for $t = 0$ for (3) gives

$$\begin{cases} u_t - u_1 + h\left(\dfrac{\mu u_1 (u-1)(u+1)}{2} + \dfrac{u + u_{tt}}{2}\right) + \mathcal{O}(h^2) = 0, \\ u_{1t} - \mu u_1 (1-u^2) + u + h\left(-\dfrac{\mu^2 u_1 (u-1)^2 (u+1)^2}{2} - \right. \\ \qquad \left. - \dfrac{\mu u (u^2 - 2u_1^2 - 1)}{2} + \dfrac{u_1 + u1_{tt}}{2}\right) + \mathcal{O}(h^2) = 0. \end{cases} \tag{4}$$

Using algorithms for constructing Gröbner bases for series in a computer algebra system SymPy, we build FDA for (4)

$$\begin{cases} u_t - u_1 + h^4\left(\dfrac{\mu^4 u_1 (u-1)^4 (u+1)^4}{120} + \ldots + \dfrac{u_1}{120}\right) + \mathcal{O}(h^5) = 0, \\ u_{1t} - \mu u_1 (1-u^2) + u + h^4\left(-\dfrac{\mu^5 u_1 (u-1)^5 (u+1)^5}{120} - \right. \\ \qquad - \dfrac{\mu^4 u (u-1)^3 (u+1)^3 (u^2 - 2u_1^2 - 1)}{120} + \\ \qquad + \dfrac{\mu^3 u_1 (u-1)(u+1)(15u^2 u_1^2 - 8u^2 - 7u_1^2 + 8)}{240} - \\ \qquad - \dfrac{\mu^2 u (2u^4 - 25u^2 u_1^2 + 4u^2 + 10u_1^4 + 5u_1^2 - 6)}{240} + \\ \qquad \left. + \dfrac{\mu u_1 (7u^2 - 6u_1^2 + 6)}{240} - \dfrac{u}{120}\right) + \mathcal{O}(h^5) = 0. \end{cases} \tag{5}$$

The construction of the FDA, which is independent of the Taylor expansion point, made it possible to correctly determine the order of the numerical method and its remainder term. We built the FDA for various explicit and implicit Runge-Kutta methods, Adams–Bashforth and Adams–Moulton multi-step methods. All results are shown that the remainder terms of the second equation of the system (2) have a form $h^p(C\mu^{p+1}(u^2-1)^{p+1}) + \ldots)$, where p is the order of the method, C is some constant. The remainder term shows the stiffness of the [6] system with respect to the μ parameter. As a result, it is necessary to choose the step h in such a way as to ensure the smallness of the remainder term.

3. Numerical experiments

Calculations were made for the Runge-Kutta method (3) with initial conditions $u = 0, u_t = 1$ for $t = 0$. The calculations were interrupted when $abs(u) >= 5$. The designation max shows the maximum value $|u|$ and is chosen for accuracy control.

As shown on Fig. 1 calculations at constant step $h = 0.05$. As a result, it can be seen that when the value of $h^4\mu^5$ is large, it is necessary to decrease the step h.

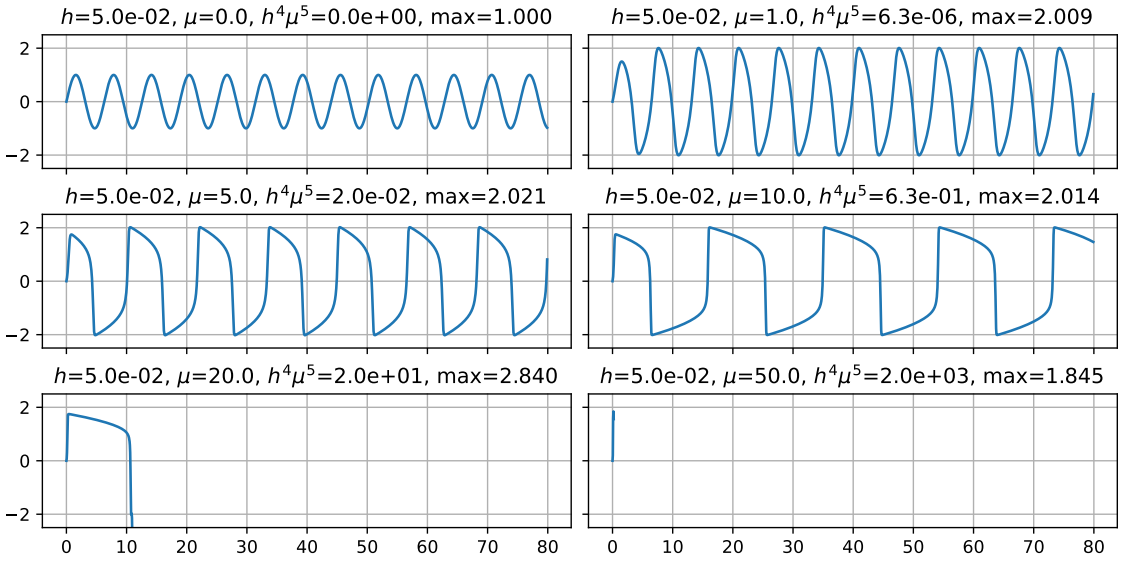


Figure 1: Calculations with a constant step

As shown on Fig. 2 calculations with variable step h using the control of the remainder term in the second equation of (5).

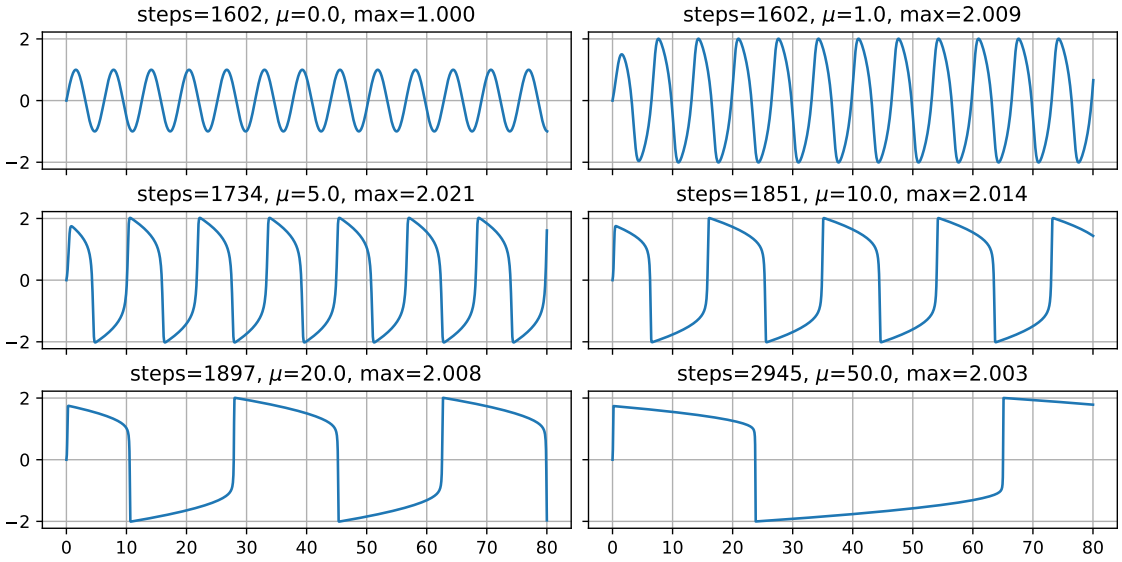


Figure 2: Calculations with variable step

On Fig. 3 the value of the FDA remainder term is shown when calculating with a variable step. It can be seen that the error remains within the specified value 0.001.

The performed calculations show that usage of the FDA allows to estimate the discrepancy of the numerical methods with respect to the parameters of the problem, to detect and

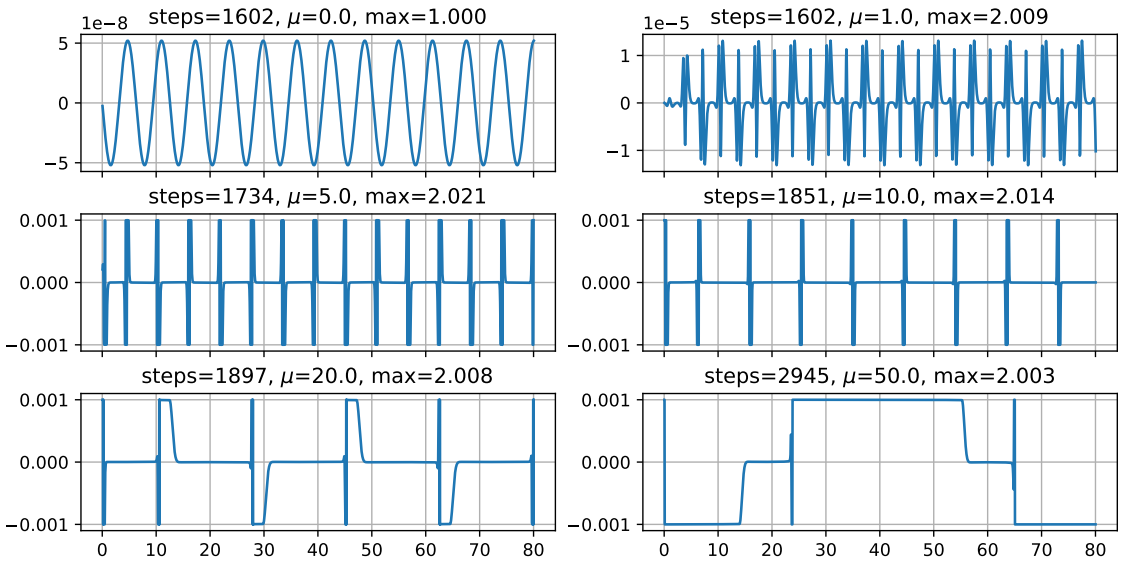


Figure 3: Accuracy in calculations with a variable step

to evaluate the stiffness of the ODE system. Furthermore, we can use residual FDA member for selection of variable step. The source texts of the programs and the presented calculations are given at `github.com/blinkovua/sharing-blinkov/tree/master/FDA_ODE`. The presented method use a standard tools of computer algebra and can be applied systems with a polynomial right-had side.

References

[1] *Yanenko N.N., Shokin Yu.I.* The first differential approximation of difference schemes for hyperbolic systems of equations. Siberian Mathematical Journal. 1969. Vol. 10, N. 5. P. 868–880. DOI: 10.1007/BF00971662

[2] *Blinkov Y.A., Gerdt V.P., Marinov K.B.* Discretization of quasilinear evolution equations by computer algebra methods. Programming and Computer Software. 2017. Vol. 43, N. 2. P. 84–89. DOI: 10.1134/S0361768817020049

[3] *Blinkov Y.A., Malykh M.D., Sevastianov L.A.* On differential approximations of difference schemes. Izvestiya of Saratov University. Mathematics. Mechanics. Informatics. 2021. Vol. 21, N. 4. P. 472–488. DOI: 10.18500/1816-9791-2021-21-4-472-488

[4] *Van der Pol B.* On "Relaxation-Oscillations". The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. 1926. N. 2. P. 978–89992. DOI: 10.1080/14786442608564127

[5] *Kutta M.* Beitrag zur näherungsweisen Integration totaler Differentialgleichungen. Zeitschrift für Mathematik und Physik. 1901. Vol. 46. P. 435–453.

[6] *Curtiss C.F., Hirschfelder J.O.* Integration of stiff equations. Proceedings of the National Academy of Sciences of the USA. 1952. Vol. 38. N. 3. P. 235–243. DOI: 10.1073/pnas.38.3.235

Hermite Interpolation Polynomials on Parallelepipeds and FEM Applications

G. Chuluunbaatar[1,2], A.A. Gusev[1], O. Chuluunbaatar[1], S.I. Vinitsky[1,2]

[1] *Joint Institute for Nuclear Research, Dubna, Russia*

[2] *Peoples' Friendship University of Russia, Russia*

e-mail: gooseff@jinr.ru, galmandakh@mail.ru, chuka@jinr.ru, vinitsky@theor.jinr.ru

Abstract

An algorithm for the analytical construction of multidimensional Hermite interpolation polynomials in a multidimensional hypercube is presented. In the case of a d-dimensional cube, the basis functions are determined by the products of d Hermite interpolation polynomials depending on each of the d variables given explicitly in analytic form. The efficiency of finite element schemes, algorithms and programs implemented in the MAPLE system is demonstrated by reference calculations of the Harmonic oscillator problem.

Keywords: Hermite interpolation polynomials, multidimensional boundary-value problem, finite element method

1. Introduction

The definition and properties of Hermitian interpolation polynomials (HIPs) or Birkhoff interpolants and their application in the finite element method (FEM) are discussed in a number of papers, for example [1, 2]. Piecewise polynomial FEM functions constructed by matching HIPs have continuous derivatives up to a given order at the finite element boundaries, in contrast to Lagrange interpolation polynomials (LIPs). Therefore, FEM with HIPs is used in problems where continuity is required not only for the approximate solution, but also for its derivatives [3]. A constructive approach to the determination of multidimensional HIPs inside a d-dimensional hypercube in the form of a polynomial of d variables of degree p' with a set of $(p'+1)^d$ unknown coefficients, which are calculated in integer arithmetic by solving a system of $(p'+1)^d$ inhomogeneous algebraic equations, was implemented as a program for $d = 3$ and $p' = 3$ in [3]. With an increase in d and p' and the dimension of the system, its solution in integer arithmetic becomes too difficult, therefore, in the general case, it is necessary to develop new algorithms free of this drawback.

In this work, we implement in Maple and Mathematica an algorithm for constructing multidimensional HIPs inside a d-dimensional hypercube as a product of d pieces of one-dimensional HIPs of degree p' in each variable, in which there is no need to solve the above-mentioned system of equations [4]. One-dimensional HIPs are calculated analytically using the authors' recurrent relations [5]. As a result, multidimensional HIPs are also calculated in an analytical form and satisfy all the conditions for their definition and properties. In the particular case $d = 3$, $p' = 3$, as shown in [6], they coincide with the 3D HIPs in [3].

The efficiency of our finite element schemes, algorithms and program GCMFEM implemented in Maple and Mathematica is demonstrated by reference calculations of the BVP for multidimensional harmonic and anharmonic oscillator used in the Geometric Collective Model (GCM) of atomic nuclei [8].

2. Algorithm

The HIPs $\varphi_{rq}^{\kappa p'}(x)$ depending on d variables in an element of a d-dimensional parallelepiped

$$x = (x_1, \ldots, x_d) \in [x_{1;\min}, x_{1;\max}] \otimes \cdots \otimes [x_{d;\min}, x_{d;\max}] = \Delta_q \subset \mathcal{R}^d \tag{1}$$

at nodes $x_r = (x_{1r_1}, \ldots, x_{dr_d})$, $x_{ir_i} = ((p - r_i)x_{i;\min} + r_i x_{i;\max})/p$; $r_i = 0, \ldots, p$, $i = 1, \ldots, d$ are determined by the relations [1, 2]

$$\varphi_{rq}^{\kappa p'}(x_{r'}) = \delta_{r_1 r_1'} \cdots \delta_{r_d r_d'} \delta_{\kappa_1 0} \cdots \delta_{\kappa_d 0}, \quad \kappa = \kappa_1 \cdots \kappa_d, \quad r = r_1 \cdots r_d, \tag{2}$$
$$\frac{\partial^{|\kappa'|}}{\partial x^{\kappa'}} \varphi_{rq}^{\kappa p'}(x)\bigg|_{x=x_{r'}} = \delta_{r_1 r_1'} \cdots \delta_{r_d r_d'} \delta_{\kappa_1 \kappa_1'} \cdots \delta_{\kappa_d \kappa_d'}, \quad \frac{\partial^{|\kappa'|}}{\partial x^{\kappa'}} = \frac{\partial^{\kappa_1}}{\partial x_1^{\kappa_1}} \cdots \frac{\partial^{\kappa_d}}{\partial x_d^{\kappa_d}}.$$

These HIPs are calculated as a product of 1D HIPs $\varphi_{r_s q}^{\kappa_s p'}(x_s)$,

$$\varphi_{rq}^{\kappa p'}(x) = \prod_{s=1}^{d} \varphi_{r_s q}^{\kappa_s p'}(x_s). \tag{3}$$

The values of the functions $\varphi_{rq}^{\kappa p'}(x)$ with their derivatives up to the order $(\kappa_r^{\max} - 1)$, i.e. $\kappa = 0, \ldots, \kappa_r^{\max} - 1$, where $\kappa = \kappa_s$, $r = r_s$ and $x = x_s$ and $\kappa_r^{\max}$ is referred to as the multiplicity of the node x_r, are determined by the expressions [1]

$$\varphi_{rq}^{\kappa p'}(x_{r'}) = \delta_{rr'} \delta_{\kappa 0}, \quad \frac{d^{\kappa'} \varphi_{rq}^{\kappa p'}(x)}{dx^{\kappa'}}\bigg|_{x=x_{r'}} = \delta_{rr'} \delta_{\kappa \kappa'}. \tag{4}$$

In particular case $\kappa_r^{\max} = 1$, the shape functions are determined only their values called by Lagrange interpolation polynomials (LIPs). To calculate the 1D HIPs the auxiliary weight function

$$w_{rq}(x) = \prod_{r'=0, r' \neq r}^{p} \left(\frac{x - x_{r'q}}{x_{rq} - x_{r'q}} \right)^{\kappa_{r'}^{\max}}, \qquad w_{rq}(x_{rq}) = 1 \tag{5}$$

is used. The weight function derivatives can be presented as a product

$$\frac{d^{\kappa} w_{rq}(x)}{dx^{\kappa}} = w_{rq}(x) g_{rq}^{\kappa}(x), \tag{6}$$

where the factor $g_{rq}^{\kappa}(x)$ is calculated by means of the recurrence relations

$$g_{rq}^{\kappa}(x) = \frac{d g_{rq}^{\kappa-1}(x)}{dx} + g_{rq}^{1}(x) g_{rq}^{\kappa-1}(x), \quad g_{rq}^{0}(x) = 1, \quad g_{rq}^{1}(x) = \sum_{r'=0, r' \neq r}^{p} \frac{\kappa_{r'q}^{\max}}{x - x_{r'q}}. \tag{7}$$

We will seek for the HIPs $\varphi_{rq}^{\kappa}(x)$ in the following form:

$$\varphi_{rq}^{\kappa p'}(x) = w_{rq}(x) \sum_{\kappa'=0}^{\kappa_r^{\max}-1} a_{rq}^{\kappa, \kappa'} (x - x_{rq})^{\kappa'}. \tag{8}$$

Differentiating the function (8) by x at the point of x_{rq} and using Eqs. (5) ,and (4) we arrive at the recurrence relations for the coefficients $a_r^{\kappa, \kappa'}$

$$a_{rq}^{\kappa, \kappa'} = \left\{ 0, \kappa' < \kappa; \frac{1}{\kappa'!}, \kappa' = \kappa; - \sum_{\kappa''=\kappa}^{\kappa'-1} \frac{1}{(\kappa' - \kappa'')!} g_{rq}^{\kappa' - \kappa''}(x_{rq}) a_{rq}^{\kappa, \kappa''}, \kappa' > \kappa. \right. \tag{9}$$

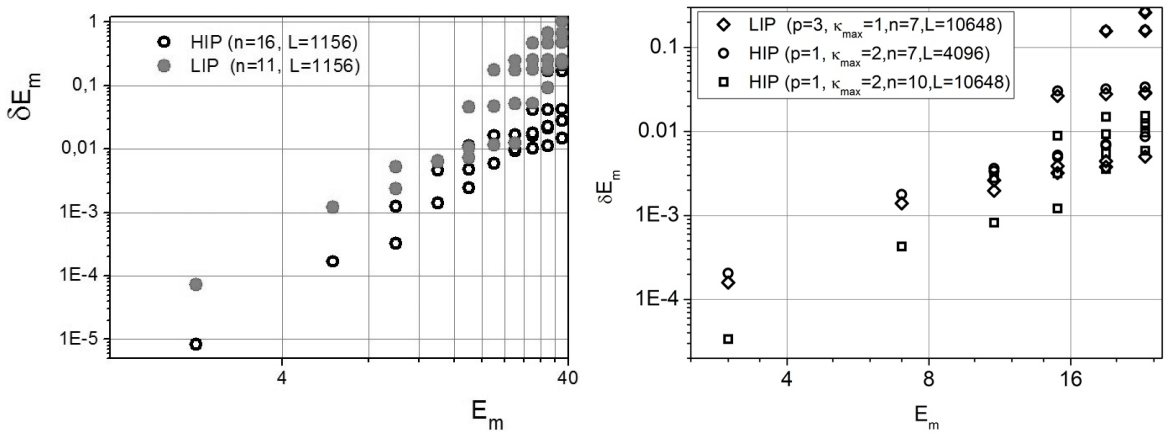


Figure 1: The discrepancy $\delta E_m{=}E_m^h{-}E_m$, $m{=}0,1,...$ of the computed eigenvalue E_m^h oscillator problem from their exact values of E_m (at R_+): $E_m{=}2_1, 6_2, 10_3, ...$ for $d{=}2$ (left panel) and $E_m{=}3_1, 7_3, 11_6, ...$ for $d{=}3$ (right panel), where the degeneracy multiplicity is indicated by subscript. The results of the FEM for cubic elements are noted — a product of one-dimensional LIPs ($p{=}3$, $\kappa_{\max}{=}1$) and HIPs ($p{=}1$, $\kappa_{\max}{=}2$) of the third order, while the square or cube was divided into n^d equal squares or cubes. The dimension of the matrix of the algebraic problem is $L{\times}L$.

3. Examples

As an example of the application of the algorithms described above, we present the results of solving the BVP in $x = (x_1, ..., x_d) \in R^d$

$$(H{-}E_m)\,\Phi_m(x){\equiv}\left(-\frac{1}{g_0(x)}\sum_{i,j=1}^{d}\frac{\partial}{\partial x_i}g_{ij}(x)\frac{\partial}{\partial x_j}{+}V(x){-}E_m\right)\Phi_m(x){=}0. \qquad (10)$$

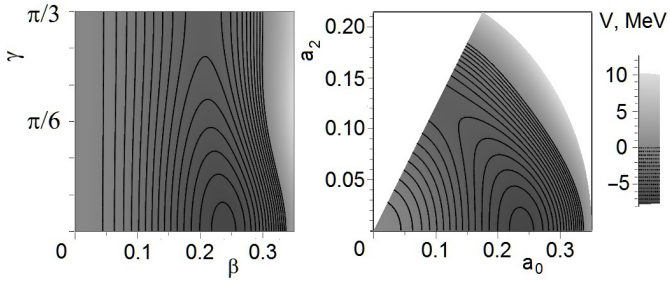
Assumed that $g_0(x) > 0$, $g_{ji}(x) = g_{ij}(x)$ and $V(x)$ are real-valued functions, continuous together with their generalized derivatives up to a given order in the domain $x \in \bar{\Omega} = \Omega \cup \partial\Omega \in \mathcal{R}^d$ with the piecewise continuous boundary $S = \partial\Omega$, which provides the existence of nontrivial solutions $\Phi(x)$ obeying the Neumann or Dirichlet boundary conditions [4].

This problem with oscillator potential $V(x){=}x_1^2{+}...{+}x_d^2$ has the known degenerate pure discrete spectrum $E_m \equiv E_{k_1...k_d}$ and corresponding set of eigenfunctions $\Phi_m(x){\equiv}\Phi_{k_1...k_d}(x)$. The 2D ($d{=}2$) and 3D ($d{=}3$) oscillator problems were solved in a square or cube $[0,7]^d$ with Neumann boundary conditions. The discrepancies $\delta E_m{=}E_m^h - E_m$, $m{=}0,1,\ldots$ between the numerical eigenvalues E_m^h of this problem and the exact values E_m are shown in Fig. 1. As is seen from the Fig. 1, the accuracy of the approximate FEM solution of the algebraic eigenvalue problems with the same dimension $L \times L$, calculated using the HIPs is higher than the accuracy of the approximate FEM solution calculated using the LIPs.

In paper [8] the GCM FORTRAN program for solving the BVP for 5D nonlinear oscillator of GCM Model of atomic nuclei with a pure discrete degenerated spectrum of eigenvalues of energy $E_n^L{=}E_1^L{<}E_2^L$, $<E_3^L,<...$ has been created. It was done in irreducible representations of the rotational group O(3) parameterized by the Euler angles x_3, x_4, x_5 in the intrinsic frame (IF). They are specify by a set of quantum numbers of the integer angular momentum $L{=}0,2,3,4,...$, and its projections $-L{\le}M{\le}L$ and $0 \le K \le L$ on the third axes of laboratory and intrinsic frames and basis functions $\Phi_K^L(x_1, x_2)$, $x_1{=}a_0{=}\beta/\sqrt{2}\cos\gamma$, $x_2{=}a_2{=}\beta\sin\gamma$ in IF. This problem at any fixed L and M is reduced to a set of $L/2{+}1$ for even L, or $(L{-}1)/2$ for odd L, of 2D BVP coupling by a three-diagonal matrix[9]. In Table 1 we compare our FEM

Table 1: The eigenenergies E_n^L (B) and E_n^L (F) (in MeV) of nucleus ^{186}Os at the parameter P_3=0 (see [8]) calculated by expansion over a basis and by FEM with HIPs, p=1, $\kappa_{\max}$=2, p'=3 in grid $[\beta=0.08, 0.107, ..., 0.35]\otimes[\gamma=0., \pi/30, ..., \pi/3]$ and potential energy surface $V(x_1, x_2)$ vs variables a_0 and a_2 (in fm), and β (in fm) and angle γ.

L	n	E_n^L (B)	E_n^L (F)
0	1	−5.491	−5.493
2	1	−5.378	−5.381
2	2	−4.411	−4.414
3	1	−4.221	−4.222
4	1	−5.139	−5.157
4	2	−4.092	−4.109
4	3	−3.439	−3.453
5	1	−3.837	−3.857



results for a low part of the spectrum of energy E_n^L of nucleus ^{186}Os calculated by FEM using HIPs with one calculated by expansion of desired solution over the basis functions implemented in the GCM FORTRAN code [8]. There is a good agrement results obtained by GCMFEM with HIPs and the expansion over the basis functions. Moreover, GCMFEM is applicable to a more wide class of BVP (10).

References

[1] *Berezin I.S.,Zhidkov, N.P.* Computing Methods. Oxford. Pergamon Press. (1965).

[2] *Lorentz R.A.* Multivariate Birkhoff interpolation. Berlin. Springer-Verlag. (1992).

[3] *Lekien F., Marsden J.* Tricubic interpolation in three dimensions. Int. J. Num. Meth. Eng. 2005. Vol. 63. P. 455–471.

[4] *Chuluunbaatar G. et al.* Construction of Multivariate Interpolation Hermite Polynomials for Finite Element Method. EPJ Web of Conferences. 2020. Vol. 226. P. 02007.

[5] *Gusev A.A. et al.* Symbolic-numerical solution of boundary-value problems with self-adjoint second-order differential equation using the finite element method with interpolation Hermite polynomials. Lecture Notes Computer Sci. 2014. Vol. 8660. P. 138–154.

[6] *Gusev A.A. et al.* Algorithm for calculating interpolation Hermite polynomials in d-dimensional hypercube in the analytical form. in "Computer algebra" Conference Materials, Moscow, June 17–21, 2019 / ed. S.A. Abramov, L.A. Sevastianov. – Peoples' Friendship University of Russia, P. 119–128

[7] *Abramowitz M., Stegun I.A.* Handbook of Mathematical Functions. New York. Dover. (1972).

[8] *Troltenier D. et al.* Numerical Application of the Geometric Collective Model. Langanke K., Maruhn J.A., Konin S.E. (eds.) Computational Nuclear Physics, Vol. 1, P. 116–139. Berlin. Springer–Verlag. (1991).

[9] *Troltenier D. et al.* A general numerical solution of collective quadrupole surface motion applied to microscopically calculated potential energy surfaces. Z. Phys. A. Hadrons and Nuclei. 1992. Vol. 343, P. 25-34.

Asymptotic Approximations and Symbolic Representation of Parametric Families of Feedback Controls in Nonlinear Systems

Yu.E. Danik, M.G. Dmitriev

Federal Research Center "Computer Science and Control" of RAS, Russia

e-mail: yuliadanik@gmail.com, mdmitriev@mail.ru

Abstract

In this report the possibility of obtaining a symbolic description of a parametric family of synthesizing controls in nonlinear systems in the classical formulation, without constraints on the values of control functions, is considered based on the use of the Padé approximation technique and an approach to the synthesis of controls for nonlinear systems using the formal application of the Kalman-Letov algorithm.

Keywords: Padé approximation, parametric family of synthesizing controls, small parameter, asymptotic methods

1. Introduction

In typical applied problems the structure of mathematical models is often fixed and the specifics of a control object are defined by the values of specific parameters that are determined in the process of tuning and trial operations. During the development of control laws for dynamic systems, situations may arise when similar motions are actually the functions of specific parameters of the model. Due to the emergence of numerous applied autonomous controlled objects, it is relevant to find control laws in real time, and for this the symbolic description of the possible solutions can be helpful.

We have proposed an approach based on the introduction of one or several parameters in the system equations of motion. These parameters change over a certain interval and generate a family of typical motions. In the vicinity of the ends of the introduced interval small parameters are introduced and, on their basis, asymptotic approximations are constructed for the solutions of the corresponding matrix Riccati equations to determine the gains coefficients matrix in the closed-loop controls. Then, the technique of Padé approximations (*PA*) [1] is used for an approximate symbolic description of the family of regulators based on the asymptotic approximations. *PA*s often have good interpolation and extrapolation properties and can serve as good initial approximations for various optimization algorithms for nonlinear problems. Here we considers the possibility of obtaining a symbolic description of a parametric family of synthesizing control laws in nonlinear control systems, without constraints on control.

For constructing the control laws of the feedback type for nonlinear systems, we use the SDRE (State Dependent Riccati Equation) approach [2] for the continuous systems and the D-SDRE approach [3] for discrete systems. In these approaches the Kalman-Letov algorithms for solving linear-quadratic optimal control problems are formally applied for nonlinear control problems, where the equations of motion are preliminary transformed to the form linear in state and control but with matrices that can be functions of state variables. The criterion is presented in a quadratic form. In this report we continue our previous works for continuous [4, 5, 6] and discrete [7, 8] control systems on the *PA* technique for constructing families of feedback controls in nonlinear systems. In these papers we have introduced a matrix Padé approximation of the Riccati equation solution for continuous and discrete problems with a small parameter that varied on the semi-axis. Here we develop

this approach for a new class of discrete control systems, where the possibility of the Padé approximation existence on a finite parameter variation interval is emphasized.

2. Approximate symbolic representation of feedback controls

Let the system be

$$\begin{array}{l} x(t+1) = \varepsilon A(x)x(t) + B(x)u(t), \\ x(0) = x_0,\ x(t) \in X \subset R^n,\ u(t) \in R^r,\ t = 0, 1, 2... \ , \alpha_1 \le \varepsilon \le \alpha_2, \end{array} \tag{1}$$

where ε is a positive parameter, $A(x) \in R^{n\times n}$, $B(x) \in R^{n\times r}$, $X \subset R^n$ is a given bounded closed state space subset, moreover, the trajectories of the closed-loop system exist and are unique in X for all admissible $t = 0, 1, 2, \ldots$

Here $\varepsilon A(x), B(x)$ is a controllable pair $\forall x \in X$, $\alpha_1 \le \varepsilon \le \alpha_2$.

The stabilizing control $u(x, \varepsilon)$ is found using the auxiliary optimal control problem with the quality criterion

$$I(u) = \frac{1}{2}\sum_{t=0}^{\infty}(x^T Q(x,\varepsilon)x + u^T R_0 u) \to \min, \tag{2}$$

where $Q(x, \varepsilon) \in R^{n\times n}$, $R_0 \in R^{r\times r} > 0$, $Q(x, \varepsilon) > 0$. Criterion matrices (2) are selected during the regulator construction process to ensure the stabilization of (1). Here $\varepsilon A(x)$,$Q^{1/2}(x, \varepsilon)$ is an observable pair $\forall x \in X$, $\alpha_1 \le \varepsilon \le \alpha_2$.

The control function is founded in the form of nonlinear state feedback as in the D-SDRE approach [3]

$$u(x,\varepsilon) = -\varepsilon\left[R_0 + B(x(t))^T P(x(t))B(x(t))\right]^{-1} B(x(t))^T P(x(t))A(x(t))x(t), \tag{3}$$

where $P(x(t))$ matrix is the solution of the corresponding discrete algebraic state-dependent Riccati equation (D-SDRE).

The proposed algorithm for nonlinear stabilizing regulator construction for (1), (2) is based on the approximate solution of the D-SDRE

$$\varepsilon^2 A^T(x)PA(x) - P - \varepsilon^2 A^T(x,\mu)PB(x,\mu)\tilde{R}^{-1}(x,\varepsilon)B^T(x)PA(x) + Q(x,\varepsilon) = 0,$$

where $\tilde{R}(x, \varepsilon) = (R_0 + B^T(x)PB(x))$ is invertible $\forall x \in X$, $\alpha_1 \le \varepsilon \le \alpha_2$.

Here the asymptotics in the vicinity of two boundary points α_1, α_2 of the parameter variation interval are used. In the neighborhood of α_1, the asymptotic series expansion by small parameter $\eta = \varepsilon - \alpha_1$ is constructed, which approximates matrix $P(x, \varepsilon)$ for ε in the right neighborhood of point α_1, and a positive parameter $\mu = \alpha_2 - \varepsilon > 0$ is introduced to construct the expansion of matrix $P(x, \varepsilon) = P(x, \alpha_2 - \mu)$ in the left neighborhood of the point $\varepsilon = \alpha_2$.

As in [6] under the conditions of the existence of Riccati equation solution and the terms of the asymptotic approximations $P_2^R(x, \eta)$,$P_2^L(x, \mu)$ and the controllability and observability of matrix pairs $(\alpha_2 A(x), B(x)), (\alpha_2 A(x), Q_0 + \alpha_2 Q_1(x) + \alpha_2^2 Q_2(x))$ and $(\alpha_1 A(x), B(x)), (\alpha_1 A(x), Q_0 + \alpha_1 Q_1(x) + \alpha_1^2 Q_2(x))$ the asymptotic estimates for the remainder of the second-order asymptotics can be identified for sufficiently small value of $\eta_0 > 0$ and $\mu_0 > 0$.

These two approximations are combined into one symbolic construction using a two-point Pade approximation [1] of order [2/2]. For the Pade approximation construction we

introduce the matrix Pade approximation of [2/2] order of the solution of the equation (3) in the form

$$PA_{[2/2]}(x,\varepsilon) = \left(\mathrm{M}_0(x) + \varepsilon\,\mathrm{M}_1(x) + \varepsilon^2\mathrm{M}_2(x)\right) \times \left(E + \varepsilon\,\mathrm{N}_1(x) + \varepsilon^2\mathrm{N}_2(x)\right)^{-1}, \qquad (4)$$

where $E_{n\times n}-$ identity matrix. The Riccati equation (3) solution is found as in the form $K_{[2/2]}(x,\varepsilon) = \left(PA^T_{[2/2]}(x,\varepsilon) + PA_{[2/2]}(x,\varepsilon)\right)/2$.

Algorithm

1. The asymptotic approximation $P_2^R(x,\eta) = P_0^R(x) + \eta P_1^R(x) + \eta^2 P_2^R(x)$ for ε in the right neighborhood of point α_1 is constructed, $\varepsilon = (\alpha_1 + \eta),\ \eta \to 0$.

2. The asymptotic approximation $P_2^L(x,\mu) = P_0^L(x) + \mu P_1^L(x) + \mu^2 P_2^L(x)$ for ε in the left neighborhood of point α_2 is constructed, $\varepsilon = \alpha_2 - \mu, \mu \to 0$.

3. Two-point Pade approximation $PA_{[2/2]}(x,\varepsilon)$ of order [2/2] is constructed.

The coefficients of the two-point Pade approximation are found by simultaneously equating the representation (4) with $P_2^R(x,\eta)$ and $P_2^L(x,\mu)$, that is $PA_{[2/2]}(x,\varepsilon) = P_0^R(x) + (\varepsilon - \alpha_1)P_1^R(x) + (\varepsilon - \alpha_1)^2 P_2^R(x)$; $PA_{[2/2]}(x,\varepsilon) = P_0^L(x) + (\alpha_2 - \varepsilon)P_1^L(x) + (\alpha_2 - \varepsilon)^2 P_2^L(x)$, from which we get the following solution

$$\begin{pmatrix} \mathrm{M}_0 \\ \mathrm{M}_1 \\ \mathrm{M}_2 \\ \mathrm{N}_1 \\ \mathrm{N}_2 \end{pmatrix} = \begin{pmatrix} E & 0 & 0 & 0 & 0 \\ 0 & E & 0 & -\beta_1 & 0 \\ 0 & E & 0 & -\beta_2 & 0 \\ 0 & 0 & E & -(-P_1^L - 2\alpha_2 P_2^L) & -\beta_1 \\ 0 & 0 & E & -(P_1^R - 2\alpha_1 P_2^R) & -\beta_2 \end{pmatrix}^{-1} \times \begin{pmatrix} \beta_2 \\ (-P_1^L - 2\alpha_2 P_2^L) \\ (P_1^R - 2\alpha_1 P_2^R) \\ P_2^L \\ P_2^R \end{pmatrix},$$

where $\beta_1 = P_0^L + \alpha_2 P_1^L + \alpha_2^2 P_2^L$, $\beta_2 = P_0^R - \alpha_1 P_1^R + \alpha_1^2 P_2^R$. As in [8] several additional correction parameters can be introduced in the Pade matrix as scalar multipliers that are then selected using the quality criterion (2) optimization.

3. Numerical example

System and criterion matrices are $A(x) = \varepsilon\begin{pmatrix} 1 & 0.1 \\ 1 & 0.5 + x_1 \end{pmatrix}$, $B(x) = \begin{pmatrix} 0 \\ 0.5 \end{pmatrix}$, $Q_0 = \begin{pmatrix} 10 & 1 \\ 1 & 10 \end{pmatrix}$, $Q_1(x) = Q_2(x) = \begin{pmatrix} 11 + 0.01x_1^2 & 0 \\ 0 & 11 + 0.01x_2^2 \end{pmatrix}$, $x_0 = [1.8; 0.1]$, $\alpha_1 = 1$, $\alpha_2 = 3$. Fig. 1 shows the criterion values (2) along the trajectories of a closed-loop system with the $PA_{[2/2]}(x,\varepsilon)$ for different parameter values.

4. Conclusion

A set of asymptotic expansions with respect to the introduced small parameter can not only serve as a basis for approximating various functions on parameter variation intervals based on the *PA*, but also serve as a kind of conditional basis for constructing efficient numerical algorithms.

Asymptotic approximations of a certain order and the corrresponding *PA*, reflect a qualitative picture of the exact solution in some areas of parameter variation and can often serve as initial approximations in multi-extremal nonlinear programming problems that appear during the solution of complex nonlinear problems. The use of asymptotic approximations

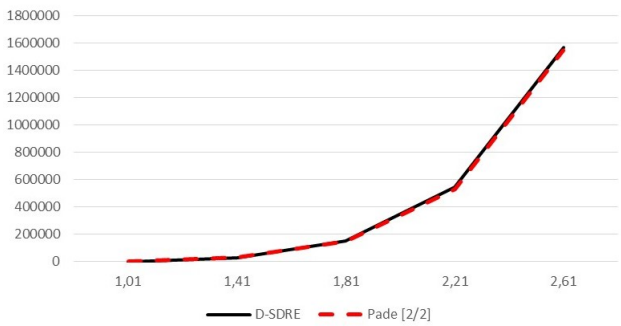


Figure 1: Criterion values (2) along the trajectories of a closed-loop system for different parameter values

for feedback controls construction helps to reduce the number of calculations by several orders of magnitude while achieving the same accuracy. This is also due to the fact that the *PA* structure suggests directions for its modification with the help of additional optimization procedures to improve its quality.

Funding: Research is supported by the Russian Science Foundation (Project No. 21-11-00202), https://rscf.ru/project/21-11-00202/. **Acknowledgments**: M.G.Dmitriev expresses his gratitude to the VIASM (Hanoi, Vietnam), where he was on a business trip connected with this scientific study.

References

[1] *Baker G., Graves-Morris P.R.* Pade approximations. Addison-Wesley Publishing. (1981).

[2] *Cimen T.* Systematic and effective design of nonlinear feedback controllers via the State-Dependent Riccati Equation (SDRE) method. Annual Reviews in control. 2010. Vol. 34. P. 32–51.

[3] *Dutka A.S., Ordys A.W., Grimble M.J.* Optimized discrete-time state dependent Riccati equation regulator. Proceedings of the American Control Conference. 2005. P. 2293–2298.

[4] *Danik Yu.E., Dmitriev M.G.* Construction of Parametric Regulators for Nonlinear Control Systems Based on the Pade Approximations of the Matrix Riccati Equation Solution. IFAC-PapersOnLine. 2018. Vol. 51. P. 815–820.

[5] *Danik Yu., Dmitriev M., Makarov D., Zarodnyuk T.* Numerical-Analytical Algorithms for Nonlinear Optimal Control Problems on a Large Time Interval. Springer Proceedings in Mathematics & Statistics. 2018. Vol. 248. P. 113–124.

[6] *Danik Yu.E., Dmitriev M.G.* The construction of stabilizing regulators sets for nonlinear control systems with the help of Pade approximations. Nonlinear Dynamics of Discrete and Continuous Systems. Springer International. 2021. P. 45–62.

[7] *Danik Yu.E., Dmitriev M.G.* One D-SDRE regulator for weakly nonlinear discrete state dependent coefficients control systems. The 7th International Conference on Control, Decision and Information Technologies (CODIT 2020). 2020. P. 616–621.

[8] *Danik Yu.E., Dmitriev M.G.* Symbolic Regulator Sets for a Weakly Nonlinear Discrete Control System with a Small Step. Mathematics. 2022. Vol. 10. P. 1–14.

Modeling of One-Step Processes Using Computer Algebra Tools

A.V. Demidova[1], O.V. Druzhinina[2], O.N. Masina[3], A.A. Petrov[3]

[1]*Peoples' Friendship University of Russia, Russia*

[2]*Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russia*

[3]*Bunin Yelets State University, Russia*

e-mail: demidova-av@rudn.ru, ovdruzh@mail.ru, olga121@inbox.ru, xeal91@yandex.ru

Abstract

The issues of using computer algebra tools for modeling of dynamic systems whose behavior can be described by one-step processes are considered. An approach based on the representation of interactions between the elements of the system under study in the form of a graph is developed. This approach makes it possible, as a result of transformations, to obtain a symbolic representation of the differential equations of the model in both the stochastic and deterministic cases. The results can be used in solving problems of constructing and researching models of natural science.

Keywords: one-step processes, scheme of interactions,computer algebra, dynamic systems, Python

1. Introduction

The development and application of computer algebra tools for symbolic calculations in solving modeling problems of nonlinear dynamical systems are relevant scientific areas [1, 2]. Due to the variety of systems with one-step processes, it is of particular interest to develop a tool for obtaining a formalized representation of dynamic models based on a general description of the principles of interaction between the elements of these systems.

A method for constructing stochastic self-consistent models was developed in [3, 4, 5]. The method is based on the combinatorial methodology [6, 7]. This method allows to make the transition to a stochastic model, an important stage of the study of which is the assessment of the introduction of stochastics on the qualitative properties of the model.

The results of the development of a software package for modeling dynamic systems whose behavior can be described by one-stage processes are presented in [8]. Examples of modeling population dynamics systems are considered. The software package allows you to obtain the corresponding stochastic model in symbolic form, as well as to conduct a detailed analysis of the model. An important aspect of software development is the application of computer algebra methods in problems of model analysis and control synthesis. This article is a continuation of [8]. Here we analyze the possibilities of a preliminary description of the system taking into account the modernization and further unification of the software package.

2. Models construction and symbolic calculations

A large class of systems can be described using multidimensional one-step processes based on graph representation (Fig. 1). Transitions from one state to another correspond to pairwise interactions between the elements of the system with the corresponding components. By $\lambda_i, \mu_i, \delta_i, \gamma_i$ we denote transition intensities. For a system that can be described in a similar way, we apply the method of constructing self-consistent stochastic models [3, 4, 5].

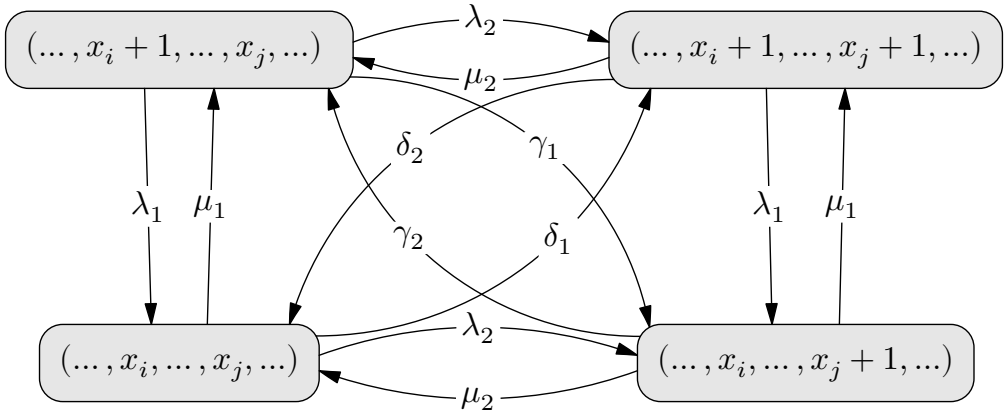


Figure 1: Transition graph of one-step process

This paper presents a software implementation of the algorithm for applying this method in the n-dimensional case, where n is the dimension of the system. The `PopModel` class is used to the model construction. Calling the `PopModel(n)` constructor creates such a class object which is a formal model of a system with pairwise interactions.

The main methods of the `PopModel` class are `adder()` and `display_infos()`. The `adder(self,type_int, id_pop_1, id_pop_2, coef=0)` method is used to add equations of the interaction scheme. Here `type_int` is interaction type, `coef` is interaction coefficients, `id_pop_1` and `id_pop_2` corresponds to the indices of elements in the system state vector. The `display_infos(self,model,X,coef)` method is intended for symbolic display of the interaction scheme.

The universal module for obtaining the coefficients of the Fokker–Planck equation from the interaction scheme is described in detail in [9]. The specified module is implemented using the computer algebra tools of the SymPy library. The transformation is performed as a sequence of operations on vector and matrix data.

The following are accepted as input data:

a) matrices `M` and `N` states of the system before and after interaction;

b) vector `X` of the system state;

c) vectors `K_plus` and `K_minus` containing the coefficients of interaction in the system.

As a result, we obtain a symbolic form of the coefficients for the Fokker–Planck equation. The main functions of the module are `drift_vector()` and `diffusion_matrix()`. The first function is designed to obtain a vector of demolitions. The demolition vector allows to get a deterministic description of the system. The second function is designed to obtain a diffusion matrix for assess the effect of stochastics in system behavior.

3. Example of the algorithm implementation for a model construction

To illustrate the capabilities of the software package, we consider an example with constructing a classical three-dimensional epidemic model (SIR model). In this system, there are three states of individuals: S – susceptible, I – infected, R – recovered (or immune). The following ODE system describes this model:

$$\frac{dS}{dt} = -\beta SI, \quad \frac{dI}{dt} = \beta SI - \gamma I, \quad \frac{dR}{dt} = \gamma I. \tag{1}$$

where β is infection rate, γ is recovery rate.

At the first stage, the vector of the system states is set in symbolic form:`X = sp.Matrix(['s', 'i', 'r'])` and an object of the class is defined: `model_sir = PopModel(3)`. Then interactions are added to the model:

```
model_sir.adder(4,1,2,"a")
model_sir.adder(7,2,3,"b")
```

In the above description, the first line describes the infection process, and the second line describes the recovery process. Using the `display_infos()` method in the Jupiter interactive shell allows to display the interaction scheme (Fig. 2).

```
a=model_sir.display_infos(model_sir,XX)
```

$$i + s = [\alpha] \Rightarrow 2i$$

$$i = [\beta] \Rightarrow r$$

Figure 2: The output of the `display_infos()` method

Figure 3 shows the derivation of the functions for obtaining the coefficients for the Fokker–Planck equation for the SIR model.

```
de.drift_vector(XX, k_plus, model_sir.matr_N(), model_sir.matr_M())
```

$$\begin{bmatrix} -is\alpha \\ is\alpha - i\beta \\ i\beta \end{bmatrix}$$

```
de.diffusion_matrix(XX, k_plus, model_sir.matr_N(), model_sir.matr_M())
```

$$\begin{bmatrix} is\alpha & -is\alpha & 0 \\ -is\alpha & is\alpha + i\beta & -i\beta \\ 0 & -i\beta & i\beta \end{bmatrix}$$

Figure 3: The output of the `drift_vector()` and `diffusion_matrix()` functions

The result is the construction of an epidemiological SIR model in both stochastic and deterministic cases. After constructing the model, the remaining modules of the software package can be used to study the system, for example, the module for adding and investigating control, the module for symbolic and numerical obtaining of the system stationary states, the module for obtaining numerical solutions.

4. Conclusion

Thus, the approach to constructing models based on the description of interactions in the form of a graph makes it possible to take into account additional factors during modeling, effectively correct the model and perform comparative analysis based on visualization of solutions of differential equations.

It should be noted that the developed software package allows for further expansion and improvement in the direction of constructing of non-stationary models of one-step processes. This direction is connected with the need to study time-dependent parameters when solving problems of population dynamics and other problems of natural science. The considered approach allows such an extension based on transitions in interaction schemes from constant parameters to time functions. The software implementation is carried out by means of the SymPy library and has a number of features in comparison with the modeling of stationary one-step processes.

A promising direction for the development of the work is the use of the symbolic regression method to build models of paired interactions taking into account control. This method allows the construction of a control function in an analytical form. To implement the symbolic regression method, it is supposed to use the PySR library of the Python programming language.

References

[1] *Kulyabov D.S., Kokotchikova M.G.* Analytical review of symbolic computing systems. RUDN J. Math. Inf. Sci. Phys. 2007. N. 1-2. pp. 38–45. (In Russian)

[2] *Banshchikov A.V., Burlakova L.A., Irtegov V.D., Titorenko T.N.* Symbolic computation in modeling and qualitative analysis of dynamical systems. Computational Technologies. 2014. N. 6. pp. 3–18. (In Russian)

[3] *Demidova A.V.* Equations of population dynamics in the form of stochastic differential equations. RUDN J. Math. Inf. Sci. Phys. 2013. N. 1. pp. 67–76. (In Russian)

[4] *Gevorkyan M.N., Velieva T.R., Korolkova A.V., Kulyabov D.S., Sevastyanov L.A.* Stochastic Runge–Kutta Software Package for Stochastic Differential Equations. Advances in Intelligent Systems and Computing. 2016. Vol. 470. pp. 169–179.

[5] *Korolkova A.V., Kulyabov D.S.* One-step stochastization method for open systems. EPJ Web of Conferences. 2020. Vol. 226. P. 02014.

[6] *Gardiner C.* Handbook of Stochastic Methods: For Physics, Chemistry and the Natural Sciences. Germany. Heidelberg. Springer. (1985).

[7] *Van Kampen N.* Stochastic Processes in Physics and Chemistry. Amsterdam. Netherlands. Elsevier (1992).

[8] *Demidova A. V., Druzhinina O. V., Masina O. N., Petrov A. A.* Development of Algorithms and Software for Modeling Controlled Dynamic Systems Using Symbolic Computations and Stochastic Methods. Programming and Computer Software. 2023. Vol. 49, N. 2. pp. 150–163.

[9] *Gevorkyan M.N., Demidova A.V., Velieva T.R., Korol'kova A.V., Kulyabov D.S., Sevast'yanov L.A.* Implementing a method for stochastization of one-step processes in a computer algebra system. Programming and Computer Software. 2018. Vol. 44, N. 2. pp. 86–93.

Symbolic-Numerical Investigation of Asymptotic Method for Studying Waveguide Propagation Problems

D.V. Divakov[1,2], A.A. Tiutiunnik[1,2],

[1] *Peoples' Friendship University of Russia (RUDN University), 6 Miklukho-Maklaya St, Moscow, 117198, Russian Federation*

[2] *Joint Institute for Nuclear Research, 6 Joliot-Curie St, Dubna, Moscow Region, 141980, Russian Federation*

e-mail: divakov-dv@rudn.ru, tyutyunnik-aa@rudn.ru

Abstract

A symbolic-numerical algorithm for solving the problem of waveguiding propagation of polarized light in irregular waveguides is considered. Within the framework of the adiabatic waveguide modes (AWM) model, the system of Maxwell's equations is reduced to a system of four ordinary differential equations and two algebraic equations for six components of the electromagnetic field in the zeroth approximation and the same number of equations in the first approximation. The paper describes a procedure for the symbolic reduction of Maxwell's equations to systems in the zeroth and first approximations of the AWM model. The steps of the symbolic-numerical method for solving the waveguide problem are described.

Keywords: smoothly irregular integrated-optical multilayer waveguides, eigenvalue and eigenvector problems, single-mode propagation of adiabatic waveguide modes

1. Introduction

We consider the guided propagation of monochromatic electromagnetic radiation in the optical range through thin–film integrated optical structures. Such structures are complex waveguides formed by applying additional guiding layers having various (smoothly irregular) geometric configurations onto a flat substrate. By a thin–film waveguide we mean a waveguide whose guiding layer thickness is comparable to the wavelength of the propagating radiation.

Integral optical structures are called smoothly irregular if the geometry of their additional guiding layer satisfies the inequalities $\left|\frac{\partial h}{\partial y}\right|, \left|\frac{\partial h}{\partial z}\right| \ll 1$.

The propagation of monochromatic polarized electromagnetic radiation through integrated optical waveguides is described by Maxwell's equations.

In the absence of foreign charges and currents, Maxwell's scalar equations follow from the vector equations, and the boundary conditions for normal components follow from the boundary conditions for the tangential components of the electromagnetic field [1]. In Cartesian coordinates related to the geometry of the substrate (or the three–layer planar dielectric waveguide), Maxwell's equations have the form [1, 2, 3, 4]

$$\begin{aligned}
&\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} = \frac{\varepsilon}{c}\frac{\partial E_x}{\partial t}, \frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\frac{\mu}{c}\frac{\partial H_x}{\partial t},\\
&\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = \frac{\varepsilon}{c}\frac{\partial E_y}{\partial t}, \frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -\frac{\mu}{c}\frac{\partial H_y}{\partial t},\\
&\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = \frac{\varepsilon}{c}\frac{\partial E_z}{\partial t}, \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\frac{\mu}{c}\frac{\partial H_z}{\partial t}.
\end{aligned} \tag{1}$$

2. Method

To construct a model of adiabatic waveguide modes (AWMs), we represent the solutions of Eqs. (1) in the form of locally normal guided modes of a locally planar reference waveguide (see [5, 6, 7]), which in the asymptotic expansion method take the form:

$$\vec{E}\left(x;y,z,t\right)=\sum_{s=0}^{\infty}\frac{\vec{E}_s\left(x;y,z\right)}{\left(-i\omega\right)^{\gamma+s}}\exp\left\{i\omega t-ik_0\varphi\left(y,z\right)\right\}, \tag{2}$$

$$\vec{H}\left(x;y,z,t\right)=\sum_{s=0}^{\infty}\frac{\vec{H}_s\left(x;y,z\right)}{\left(-i\omega\right)^{\gamma+s}}\exp\left\{i\omega t-ik_0\varphi\left(y,z\right)\right\}. \tag{3}$$

Using the asymptotic expansion (1), (2) in terms of dimensional small parameter ω^{-1}, we obtain in computer algebra system `Maple` (CAS `Maple`) a system of homogeneous equations in the zeroth approximation [8, 9] and a system of first–order equations:

$$-ik_0\frac{\partial\varphi}{\partial y}\frac{H_1^z}{(-i\omega)}+\frac{\partial H_0^z}{\partial y}+ik_0\frac{\partial\varphi}{\partial z}\frac{H_1^y}{(-i\omega)}-\frac{\partial H_0^y}{\partial z}=ik_0\varepsilon\frac{E_1^x}{(-i\omega)}, \tag{4}$$

$$-ik_0\frac{\partial\varphi}{\partial z}\frac{H_1^x}{(-i\omega)}+\frac{\partial H_0^x}{\partial z}-\frac{\partial H_1^z}{\partial x}\frac{1}{(-i\omega)}=ik_0\varepsilon\frac{E_1^y}{(-i\omega)}, \tag{5}$$

$$\frac{\partial H_1^y}{\partial x}\frac{1}{(-i\omega)}+ik_0\frac{\partial\varphi}{\partial y}\frac{H_1^x}{(-i\omega)}-\frac{\partial H_0^x}{\partial y}=ik_0\varepsilon\frac{E_1^z}{(-i\omega)}, \tag{6}$$

$$-ik_0\frac{\partial\varphi}{\partial y}\frac{E_1^z}{(-i\omega)}+\frac{\partial E_0^z}{\partial y}+ik_0\frac{\partial\varphi}{\partial z}\frac{E_1^y}{(-i\omega)}-\frac{\partial E_0^y}{\partial z}=-ik_0\mu\frac{H_1^x}{(-i\omega)}, \tag{7}$$

$$-ik_0\frac{\partial\varphi}{\partial z}\frac{E_1^x}{(-i\omega)}+\frac{\partial E_0^x}{\partial z}-\frac{\partial E_1^z}{\partial x}\frac{1}{(-i\omega)}=-ik_0\mu\frac{H_1^y}{(-i\omega)}, \tag{8}$$

$$\frac{\partial E_1^y}{\partial x}\frac{1}{(-i\omega)}+ik_0\frac{\partial\varphi}{\partial y}\frac{E_1^x}{(-i\omega)}-\frac{\partial E_0^x}{\partial y}=-ik_0\mu\frac{H_1^z}{(-i\omega)}. \tag{9}$$

To construct the system of first–order ODEs, it is first necessary to solve the homogeneous system of zero–order ODEs. Then for each zero–order solution we write down an inhomogeneous system of first–order ODEs.

The system of homogeneous ODEs for the zero–order contributions to the adiabatic guided modes in a three–layer thin–film waveguide reduces symbolically to a homogeneous system of linear algebraic equations (SLAE) of the following form [8, 9]

$$\hat{M}\left(\beta_0\left(z\right)\right)\vec{A}_0\left(\beta_0\left(z\right)\right)=\vec{0}. \tag{10}$$

The solvability condition for SLAE (10) is

$$det\hat{M}\left(\beta_0\left(z\right)\right)=0 \tag{11}$$

at any z. The nonlinear equation (11) is formulated in symbolic form in CAS `Maple` and solved numerically by means of numerical methods, implemented in `Maple`.

The system of inhomogeneous ODEs for the first–order contributions to AWMs, obtained similar to the approach [8, 9], contains in its right–hand side the analytical expressions depending on the derivatives $\frac{\partial}{\partial z}$ of $\vec{A}_0\left(\beta_0\left(z\right)\right)$ and $\beta_0\left(z\right)$. Therefore, to write down the concrete dependence of the right–hand side on the solutions of Eqs. (10) and (11), these

solutions should be obtained in the class of continuously differentiable functions. A method to find such zero–order solutions was proposed in Ref. [9].

After using symbolic calculations to get explicit expressions for $\vec{A}_0(\beta_0(z))$ depending on the numerical solution $\beta_0(z)$, we can reduce the system of inhomogeneous ODEs for the first–order contributions to a system of inhomogeneous SLAE

$$\hat{M}(\beta_1(z))\,\vec{A}_1(\beta_1(z)) = \vec{F}\left(\frac{\partial\beta_0}{\partial z}, \frac{\partial\vec{A}_0}{\partial z}\right) \tag{12}$$

with the same symbolically defined matrix as for zero–order contributions [8], but depending on the other parameter $\beta_1(z)$. As before, in this case it is necessary to require the solvability of the inhomogeneous SLAE (12).

Having obtained the solutions of the zero–order and first–order equations for the AWM model in the closed form, we finally can use them to express the electromagnetic fields as

$$\vec{E}(x;y,z) = \vec{E}_0(x;y,z) + \frac{i}{\omega}\vec{E}_1(x;y,z), \tag{13}$$

$$\vec{H}(x;y,z) = \vec{H}_0(x;y,z) + \frac{i}{\omega}\vec{H}_1(x;y,z). \tag{14}$$

References

[1] Adams M.J. An Introduction to Optical Waveguides. Wiley, New York (1981).

[2] Stevenson A.F. General Theory of Electromagnetic Horns // J. Appl. Phys. 1951. V.22. № 12. P.1447.

[3] Schelkunoff S.A. Conversion of Maxwell's equations into generalized telegraphist's equations // Bell Syst. Tech. J. 1955. V.34. P.995–1043.

[4] Katsenelenbaum B.Z., Mercader del Rio L., Pereyaslavets M., Sorolla Ayza M., Thumm M. Theory of Nonuniform Waveguides: the cross-section method. The Institution of Engineering and Technology, London (1998).

[5] Sevastianov L.A., Egorov A.A. Theoretical analysis of the waveguide propagation of electromagnetic waves in dielectric smoothlyirregular integrated structures // Optics and Spectroscopy. 2008. V.105. № 4. P.576–584.

[6] Egorov A.A., Sevastianov L.A. Structure of modes of a smoothly irregular integrated optical four-layer three-dimensional waveguide // Quantum Electronics. 2009. V.39. № 6. P.566–574.

[7] Egorov A.A., Lovetskiy K.P., Sevastianov A.L., Sevastianov L.A. Simulation of guided modes (eigenmodes) and synthesis of a thin-film generalised waveguide Luneburg lens in the zero-order vector approximation // Quantum Electronics. 2010. V.40. № 9. P.830–836.

[8] Divakov D.V., Sevastianov A.L. The Implementation of the Symbolic-Numerical Method for Finding the Adiabatic Waveguide Modes of Integrated Optical Waveguides in CAS Maple. // Lecture Notes in Computer Science. 2019. V.11661. P.107–121.

[9] Divakov D.V., Lovetskiy K.P., Sevastianov L.A., Tiutiunnik A.A. A single-mode model of cross-sectional method in a smoothly irregular transition between planar thin-film dielectric waveguides // Proceedings of SPIE. 2021. V.11846. P.118460T.

Integrable Cases of the Resonant Bautin System

V.F. Edneral
Skobeltsyn Institute of Nuclear Physics of
Lomonosov Moscow State University, Russia
e-mail: edneral@theory.sinp.msu.ru

Abstract

Using the example of a polynomial resonance case of the Bautin system with parameters, we have written out the conditions for local integrability near stationary points and found restrictions on the parameters under which these conditions are satisfied. The resulting constraint is written as a system of algebraic equations for the ODE parameters. It is shown that for parameter values that are solutions of such an algebraic system, the ODE turns out to be integrable.

In this way we have found several cases of integrability. We propose a heuristic method that allows one to a priory determine the cases of integrability of an autonomous ODE with a polynomial right-hand side. The paper has an experimental character.

Keywords: resonance normal form, integrability, Bautin system, computer algebra

1. Introduction

We use an approach based on local analysis. It uses the resonant normal form computed near stationary points [1, 2]. In the paper [3] we proposed a method for searching for integrable cases based on determining the parameter values for which the dynamical system is locally integrable at all stationary points simultaneously. Because at regular points, local integrability always holds, so such a requirement is equivalent to the requirement of local integrability at every point of the domain under consideration.

Note that the integrability of an autonomous planar system implies the solvability of the system in quadratures.

Problem

We will check our method on the example of a resonance case of the Bautin system [4]

$$\begin{aligned} \dot{x} &= \alpha x + \beta y + a_1 x^2 + a_2 x\, y + a_3 y^2, \\ \dot{y} &= \gamma x - \delta y + b_1 x^2 + b_2 x\, y + b_3 y^2, \end{aligned} \tag{1}$$

here x and y are functions in time and $\alpha, \beta, \gamma, \delta, a_1, a_2, a_3, b_1, b_2, b_3$ are parameters. We consider here the resonance case with a center point at the origin, that is $\beta = 1, \alpha = \delta = 0, \gamma = -1$, where M is non-negative integer. Each value of M corresponds to the resonance $M : 1$.

The problem is to find integrable cases of system (1).

2. Condition of Local Integrability

We calculated the normal form for resonances [1:1] by the MATHEMATICA program [5] till the terms of eight order and got the necessary condition of local integrability at the

origin as three algebraic equations on the parameters. The first couple of these equations are

$$\begin{array}{l} a_1(a_2-2b_1)+a_2a_3+2a_3b_3-b_1b_2-b_2b_3=0,\\ 4a_1^3(5a_2-10b_1-9b_3)+a_1^2(a_2(40a_3-b_2)-4a_3(b_1+5b_3)+2b_2(9b_1+5b_3))+\\ a_1(5a_2^3-a_2^2(9b_1+13b_3)+a_2(40a_3^2+18a_3b_2-18b_1^2-8b_1b_3-b_2^2-10b_3^2)+\\ 20a_3^2(b_1+2b_3)+8a_3b_2(b_1+b_3)-40b_1^3-40b_1^2b_3+9b_1b_2^2+20b_1b_3^2+13b_2^2b_3+\\ 36b_3^3)+5a_2^3a_3+a_2^2(b_2(b_1+b_3)-a_3(5b_1+9b_3))+a_2(20a_3^3+19a_3^2b_2-\\ a_3(10b_1^2+8b_1b_3+b_2^2+18b_3^2)+b_2(-19b_1^2-18b_1b_3+b_3^2))+40a_3^3b_3+\\ 10a_3^2b_1b_2+18a_3^2b_2b_3-20a_3b_1^2b_3+5a_3b_1b_2^2+4a_3b_1b_3^2+9a_3b_2^2b_3+40a_3b_3^3-\\ 20b_1^3b_2-40b_1^2b_2b_3-5b_1b_2^3-40b_1b_2b_3^2-5b_2^3b_3-20b_2b_3^3=0. \end{array} \tag{2}$$

There are 7 solutions of that system. These solutions are good candidates for integrability cases. We found the corresponding first integrals by the MATHMATICA-11 procedure DSolve. At all solutions of (2) system (1) is integrable.

Results

We have got the first integrals for these cases. That is these cases are integrable. The are:

1) $\dot{x} = y + a_2xy,$
$\dot{y} = -x + b_1x^2 + b_3y^2,$
$I(x,y) = (a_2x+1)^{-\frac{2b_3}{a_2}}(b_3x(2(a_2+b_1-b_3)-b_1(a_2-2b_3)x)+$
$b_3(a_2-b_3)(a_2-2b_3)y^2+a_2+b_1-b_3);$

2) $\dot{x} = y + a_1x^2 + a_2xy - a_1y^2,$
$\dot{y} = -x + b_1x^2 - 2a_1xy - \frac{1}{2}a_2y^2,$
$I(x,y) = -6a_1x^2y + 2a_1y^3 - 3y^2(a_2x+1) + x^2(2b_1x-3);$

3) $\dot{x} = y + a_1x^2 + a_2xy - a_1y^2,$
$\dot{y} = -x + \frac{1}{2}a_2x^2 - 2a_1xy - \frac{1}{2}a_2y^2,$
$I(x,y) = x^2(2a_1y+1) - \frac{2}{3}a_1y^3 + a_2xy^2 - \frac{1}{3}a_2x^3 + y^2;$

4) $\dot{x} = y + a_3y^2,$
$\dot{y} = -x + 2a_3xy,$
$I(x,y) = 2a_3y(a_3y+3) + 3\log(1-2a_3y) - 4a_3^2x^2;$

5) $\dot{x} = y + a_2xy + \frac{(b_2^2-a_2^2)y^2}{2b_2},$
$\dot{y} = -x + b_2xy + \frac{(b_2^2-a_2^2)y^2}{2a_2},$
$I(x,y) = a_2(\log(-2a_2b_2y(a_2+b_2^2x) + 2a_2(a_2+b_2^2x)+$
$b_2^2(a_2-b_2)(a_2+b_2)y^2) + b_2y) - b_2^2x;$

6) $\dot{x} = \quad y + a_2 xy - b_2 y^2,$
$\dot{y} = -x - a_2 x^2 + b_2 xy,$
$I(x,y) = x^2 + y^2;$

7) $\dot{x} = \quad y + a_1 x^2 + a_2 xy - a_1 y^2,$
$\dot{y} = -x + b_1 x^2 - 2a_1 xy - b_1 y^2,$

$I(x,y) = \int \frac{x - b_1 x^2 + 2a_1 xy + b_1 y^2}{R(x,y)}\, dx, \quad$ where

$R(x,y) = -1 - x(-3a_1^2 x + a_2(-1 + b_1 x)^2 + b_1(-2 + x(b_1 + 2a_1^2 x))) +$
$a_1 x(-a_2 + 2b_1 + 6a_1^2 x + 3a_2 b_1 x)y + (b_1(a_2 + b_1)(1 + a_2 x) +$
$a_1^2(3 + 2(a_2 + b_1)x))y^2 - a_1(2a_1^2 + a_2 b_1)y^3.$

Conclusions

Based on the hypothesis about the connection between global and local integrability, 7 integrable cases of the resonant case of the Bautin system are algorithmically obtained.

We have shown that for a plane autonomous system of ODEs with polynomial right-hand sides, one can write down a system of algebraic equations with respect to the parameters of the system, the solutions of which will correspond to the integrable cases of this system of ODEs.

References

[1] A.D. Bruno, *Analytical form of differential equations (I, II)*. Trudy Moskov. Mat. Obsc. **25**, 119–262 (1971), **26**, 199–239 (1972) (in Russian) = Trans. Moscow Math. Soc. **25**, 131–288 (1971), **26**, 199–239 (1972) (in English).

[2] A.D. Bruno, *Local Methods in Nonlinear Differential Equations*. Nauka, Moscow 1979 (in Russian) = A.D. Bruno, *Local Methods in Nonlinear Differential Equations*. Translated by W. Hovingh and S. Coleman. Springer-Verlag, Berlin Heidelberg NewYork London Paris Tokyo (1989). 348 p.

[3] A.D. Bruno, V.F. Edneral, V.G. Romanovski, *On new integrals of the Algaba-Gamero-Garcia system*. In Gerdt, V.P., Koepf, W., Seiler, W.M., Vorozhtsov, E.V. (eds.) CASC 2017. Springer-Verlag series: LNCS **10490** (2017) 40–50.

[4] N.N. Bautin, *On the number of limit cycles appearing with variation of the coefficients from an equilibrium state of the type of a focus or a center*. Mat. Sb. (N.S.) **72**, # 1 (1952), 181–196 (In Russian).

[5] V.F. Edneral, R. Khanin, *Application of the resonant normal form to high-order non-linear odes using Mathematica*, Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **502**, # 2-3 (2003) 643–645.

An Optimized Procedure for Deciding Affinity of Finite Quasigroups

A.V. Galatenko, A.E. Pankratiev, R.A. Zhigliaev
Lomonosov Moscow State University, Russia
e-mail: agalat@msu.ru, apankrat@intsys.msu.ru, rzhiglyaev@mail.ru

Abstract

Finite quasigroups are becoming a popular platform for the design of cryptographic primitives. Quasigroup affinity is one of the crucial properties in the framework of cryptography. In our paper we propose an optimized procedure that decides affinity with the complexity $O\left(n^2 \log n\right)$, where n is the quasigroup order, and present the results of experiments with the software implementation.

Keywords: finite quasigroup, affinity

1. Introduction

Finite quasigroups are becoming a popular platform for the design of cryptographic primitives (see, e.g., the reviews [1, 2, 3]). The framework of cryptographic applications imposes a number of restrictions on suitable quasigroups. Non-affinity is one of such restrictions. Deciding solvability of systems of equations over an affine quasigroup is polynomial, whereas transition to non-affine quasigroups makes the problem NP-complete [4], thus algebraic attacks against systems based on non-affine quasigroups are much less feasible. In [5] the authors proposed an algorithm that decides affinity of quasigroups specified by Cayley tables with time complexity which is cubic in the quasigroup order; efficiency of a practical implementation of this algorithm was discussed in [6]. We propose an optimization of the algorithm that allows reducing complexity to $O\left(n^2 \log n\right)$, where n is the quasigroup order, and present the results of experiments with the software implementation of the optimized algorithm.

The rest of the paper is organized as follows. In Section 2 we give basic definitions. Section 3 is devoted to the description of our algorithm. In Section 4 we list the results of numerical experiments. Section 5 is the conclusion.

2. Basic definitions

Definition 1. A finite quasigroup is a pair (Q, f), where Q is a finite set, $f\colon Q \times Q \to Q$ is such that for any $a, b \in Q$ the equations $f(x, a) = b$ and $f(a, y) = b$ are uniquely solvable.

For the sake of brevity the word "finite" will be omitted.

Without loss of generality we assume that $Q = \{0, \ldots, n-1\}$ for some $n \in \mathbb{N}$. A quasigroup can be naturally represented by its Cayley table, i.e., an $n \times n$ matrix with rows and columns enumerated starting from 0 and such that the element in the ith row and the jth column equals $f(i, j)$. If the table is stored in memory, then evaluation of the quasigroup operation consists in a single memory lookup.

Definition 2. A quasigroup (Q, f) is affine if there exists an Abelian group $(Q, +)$, automorphisms α, β of this group and a constant $c \in Q$ such that $f(x, y) \equiv \alpha(x) + \beta(y) + c$.

Assume that (Q, g) is a finite set endowed with a binary operation (i.e., a magma), $Q' \subseteq Q$. Then $g(Q')$ denotes the set of all elements of Q that can be obtained by iterative application of the operation g to the elements of Q'. We say that Q' is a generator with respect to g if $g(Q') = Q$.

In complexity analysis we assume that elementary operations are reading and writing data from/to a cell in memory (in particular, evaluation of a quasigroup operation is elementary) and arithmetic operations in $\mathbb{Z}_n$. The base of logarithms equals 2.

3. Algorithm description

In [5] the authors proposed the following algorithm for deciding affinity of a quasigroup (Q, f) specified by the Cayley table L.

```
1. reorder the rows and columns of L so that the first row
                and the first column specify the identical permutation
2. check whether the resulting table L' is symmetric
                (i.e., the operation is commutative)
    if not, return ‘‘non-affine’’
3. check whether the operation specified by L' is associative
    if not, return ‘‘non-affine’’
4. set A equal to the column of L starting with 0
5. set B equal to the row of L table starting with 0
6. if A or B are not automorphisms with respect to the operation
                specified by L'
    return ''non-affine''
7. set C equal to the upper left element of L
8. check that the identity f(x,y) = A(x) + B(y) + C holds
    if it does then return ‘‘affine’’
    else return ‘‘non-affine’’
```

All steps of this algorithm except associativity test can be straightforwardly implemented with complexity that is at most quadratic in the quasigroup order. Straightforward associativity test used in [5] and slightly optimized in [6] has cubic complexity. In [7] Tarjan noticed that associativity check for a magma (Q, g) can be reduced to verifying the equality $g(g(x, a), y) = g(x, g(a, y))$ for all $x, y \in Q$ and all $a \in Q'$, where Q' is a generator with respect to g. In can be easily shown that if (Q, g) is a quasigroup, then there exists a generator of the size at most $\log |Q| + 1$ (see, e.g., [8, Lemma 1]) that can be found with quadratic complexity. Thus replacing step 3 with the block

```
3.1. find a generator G of the quasigroup
3.2. for all x,y from Q and a from G check that
                f (f (x, a), y) = f (x, f (a, y))
    if not, return ‘‘non-affine’’
```

yields the following fact.

Theorem 1. *The modified algorithm decides whether a quasigroup is affine with the complexity* $O\left(|Q|^2 \log |Q|\right)$.

Consider a group $\mathbb{Z}_2^k$ for some $k \in \mathbb{N}$. If we take a block of the Cayley table formed by rows number $2s, 2s+1$ and columns number $2t, 2t+1$ for some s, t, it will obviously have the form

$$\begin{array}{cc} 2s \oplus 2t & 2s \oplus 2t \oplus 1 \\ 2s \oplus 2t \oplus 1 & 2s \oplus 2t \end{array}$$

Swapping the rows of this block inside the Cayley table yields the Cayley table of a non-associative quasigroup operation. The blocks for different values of s, t obviously do not intersect, so deciding non-affinity requires analysis of all such blocks, thus we obtain a quadratic lower bound on algorithm complexity.

4. Experimental results

We implemented the original and the optimized algorithm in C++ programming language and applied these implementations to two series of quasigroups: random (uniformly distributed) quasigroups generated using the method proposed by Jacobson and Matthews in [9] and affine quasigroups obtained from $(\mathbb{Z}_n, +)$ by imposing automorphisms on x and y and adding a constant.

All experiments were performed on a workstation with CPU Intel(R) Core(TM) i5-11400H @2.70GHz and 8Gb of RAM.

Interestingly in the case of random quasigroups the optimization did not give any speedup (see Fig. 1). It can be explained by the fact that almost all quasigroups are known to be non-affine (and non-isotopic to affine quasigroups; see, e.g. [10]), and the decision is made at an early stage: either commutativity test fails or a non-associative triple is found quickly.

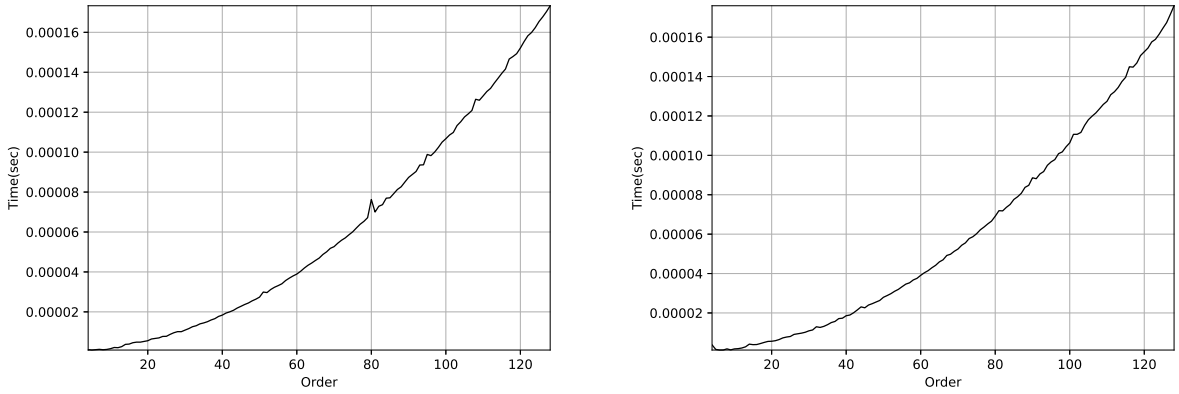


Figure 1: Running time of the original version (left) and the optimized version (right) for random (uniformly distributed) quasigroups.

In the case of affine quasigroups algorithm optimization lead to an essential speedup (see Fig. 2; note the difference in the scale of y axes).

5. Conclusion

Non-affinity is a property of finite quasigroups which plays a crucial role in cryptographic applications. We proposed an optimization that allowed us to reduce affinity decision complexity from $O(n^3)$ to $O(n^2 \log n)$, where n is quasigroup order. Computational experiments showed that the optimization does not essentially affect running time for random quasigroups while providing an essential speedup for affine quasigroups.

The authors thank A.V. Vasilev and I.N. Ponomarenko for the idea of the optimization.

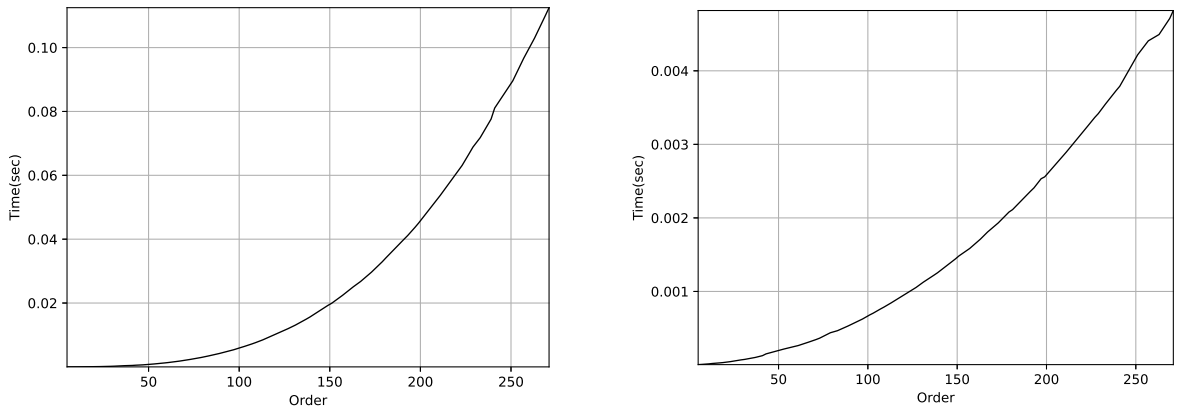


Figure 2: Running time of the original version (left) and the optimized version (right) for affine quasigroups.

References

[1] *Glukhov M.M.* Some applications of quasigroups in cryptography. Prikl. Diskr. Mat. 2008. N. 2 (2), P. 28–32. (in Russian)

[2] *Shcherbacov V.A.* Quasigroups in cryptology. Comput. Sci. J. Mold. 2009. Vol. 17, N. 2 (50). P. 193–228.

[3] *Chauhan D., Gupta I., Verma R.* Quasigroups and their applications in cryptography. Cryptologia. 2021. Vol. 45, N. 3. P. 227–265.

[4] *Larose B., Zádori L.* Taylor terms, constraint satisfaction and the complexity of polynomial equations over finite algebras. Int. J. Algebra Comput. 2006. Vol. 16. P. 563–581.

[5] *Galatenko A.V., Pankratiev A.E.* The complexity of checking the polynomial completeness of finite quasigroups. Discrete Math. Appl. 2020. Vol. 30, N. 3. P. 169–175.

[6] *Galatenko A.V., Pankratiev A.E., Staroverov V.M.* Efficient verification of polynomial completeness of quasigroups. Lobachevskii J. Math. 2020. Vol. 41, N. 8. P. 1444-1453.

[7] *Tarjan R.E.* Determining whether a groupoid is a group. Inf Process Lett. 1972. Vol. 1, N. 3. P. 120–124.

[8] *Miller G.L.* On the $n^{\log n}$ isomorphism technique: a preliminary report. Proceedings of Tenth Annual ACM Symposium on Theory of Computing. 1978. P. 51–58.

[9] *Jacobson M.T., Matthews P.* Generating uniformly distributed random Latin squares. J. Comb. Des. 1996. Vol. 4, N. 6. P. 405–437.

[10] *Galatenko A.V., Galatenko V.V., Pankrat'ev A.E.* Strong Polynomial Completeness of Almost All Quasigroups. Math Notes. 2022. Vol. 111. P. 7–12.

Analytical Geometry of the Projective Space $\mathbb{R}\mathrm{P}^3$ in Terms of Plücker Coordinates and Geometric Algebra

M.N. Gevorkyan[1], A.V. Korolkova[1], D.S. Kulyabov[1,2]

[1]*Peoples' Friendship University of Russia, Russia*

[2]*Laboratory of Information Technologies Joint Institute for Nuclear Research 6 Joliot-Curie, Dubna, Moscow region, 141980, Russia*

e-mail: gevorkyan-mn@rudn.ru, korolkova-av@rudn.ru, kulyabov-ds@rudn.ru

Abstract

Computer graphics uses a model of projective space to display three-dimensional scenes. The paper presents the basics of the analytic projective geometry of the space $\mathbb{R}\mathrm{P}^3$ in terms of Plücker coordinates. The interpretation of projective space and Plücker coordinates in terms of geometric algebra is also presented.

Keywords: projective geometry, Plücker coordinates, computer graphics, geometric algebra

1. Introduction

In this paper, we consider the description of the projective space $\mathbb{R}\mathrm{P}^3$ in terms of geometric algebra. The choice of this particular model of projective space is due to its important practical significance, since it is used in computer graphics, robotics and machine vision.

The main task of computer graphics can be formulated as a plausible simulation of the three-dimensional world surrounding a person. Therefore, naturally, all the developments in this field from fine art, architecture and engineering migrated to computer graphics. In art, the concept of perspective has emerged since ancient times as a means of conveying the volume of the surrounding world in a flat drawing. The mathematical description of perspective was formulated within the framework of projective geometry, the beginning of which was laid by J. Desargues, M. Chasles, K. von Staudt, Y. Plücker, etc [1].

The application of the ideas of projective geometry makes it possible to avoid exceptional cases in calculations with geometric primitives (points, lines and planes). So, for example, in a projective space, all planes intersect along some straight line, which can be proper (an ordinary finite straight line) or improper (an ideal straight line located at infinity). With the correct organization of data structures in the program code, all exceptional cases can be efficiently handled.

This paper summarizes the classical approach to analytic projective geometry based on homogeneous coordinates. Homogeneous coordinates for points, Plücker coordinates for a straight line and for a plane are introduced. The main emphasis is on the Plücker coordinates and different approaches to their description. The following is an interpretation of projective geometry in terms of geometric algebra. Interestingly, there are two approaches to this description. In conclusion, a brief overview of a number of open libraries for symbolic and numerical calculations for Python and Julia languages is made.

The report also uses illustrations created programmatically using the Asymptote language. These results are valuable because they make possible to visually check the correctness of formulas.

2. Analytic projective geometry of space $\mathbb{R}\mathrm{P}^3$

In the course of studying and presenting projective geometry, a synthetic approach is widely used. This is justified, since there are no metric concepts in projective geometry and the purity of the presentation of the theory dictates the requirements not to involve them in theoretical research. However, there is no such restriction in applied problems and the $\mathbb{R}\mathrm{P}^3$ model of a three-dimensional projective space is used, for modeling which a four-dimensional Cartesian space is used [2].

The use of homogeneous coordinates and homogeneous equations for planes is well known. Homogeneous coordinates are described in one way or another in many textbooks on the mathematical foundations of machine graphics, and the homogeneous equation of the plane is the general equation of the plane, which is studied in all standard courses of analytical geometry. However, the homogeneous description of the straight line is much worse known. To define a straight line in a homogeneous form, the Plücker coordinates are used, within which the straight line is described by six parameters. In our presentation, this is the guiding vector of the line $\mathbf{v} = (v_x, v_y, v_z)^T$ and the moment of the line $\mathbf{m} = (m_x, m_y, m_z)^T$. The moment of the line is defined as $\mathbf{m} = \mathbf{p} \times \mathbf{v}$, where $\mathbf{p}$ is an arbitrary point of the line, and $\times$ is a vector product. Interestingly, the moment of a straight line does not depend on the choice of a point of a straight line and is the same for any point lying on a straight line. From the definition of the vector product, the relation $(\mathbf{v}, \mathbf{m}) = 0$ naturally arises, which is called the Plücker relation and imposes on the six parameters $(v_x, v_y, v_z, m_x, m_y, m_z)$ a second-order ratio, limiting the number of degrees of freedom when choosing parameters from six to four.

It should be noted that such a method of introducing Plücker coordinates was developed in the theory of screw calculus, and a more classical method consists in introducing parameters p_{ij}, defined through determinants and written as a skew-symmetric matrix 4×4. The approach we use lends itself better to geometric interpretation and makes it easier to draw analogies with geometric algebra.

Using Plücker coordinates, homogeneous coordinates for points and a homogeneous equation for a plane, allows you to reduce all problems with points, lines and planes to an analytical form. If you enter notation for points $(\mathbf{p} \mid w) = (x : y : z : w)$, for straight lines $\{\mathbf{v} \mid \mathbf{m}\}$ and for planes $[\mathbf{n} \mid d]$, then the tasks of finding the mutual position are reduced to a set of capacious formulas covering all possible cases [3].

3. Projective geometry in terms of geometric algebra

The use of concepts from the Grassmann algebra (outer product, p-vectors) and Clifford algebra (geometric product, multivectors) allows us to significantly generalize the formulas of analytic projective geometry and give them a more understandable interpretation. The principle of duality gets a clearer interpretation, and with the introduction of additional dual operations, such as the anti-external product $\vee$, it becomes possible to record geometric constructions, such as the intersection of a straight line and a plane or drawing through a straight line and a point of the plane, in the form of algebraic operations.

To do this, consider a vector space with a basis of the following form: $\langle \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4 \rangle$, where the scalar product is $(\mathbf{e}_i, \mathbf{e}_j) = \delta_{ij}$ for $i, j = 1, 2, 3$ and $(\mathbf{e}_4, \mathbf{e}_4) = 0$. Within this framework, it is possible to construct an external algebra of p-vectors, where $p = 1, 2, 3, 4$. Vectors (1-vectors) are naturally interpreted as points and have 4 components, 2-vectors (bivectors) are interpreted as planes, have 6 components corresponding to Plücker coordinates, and 3-vectors correspond to planes and their four components exactly correspond to homogeneous

coordinates $n_x : n_y : n_y : d$.

Consider the basis of bivectors, which consists of six elements of the following form:

$$\mathbf{e}_{23} = \mathbf{e}_2 \wedge \mathbf{e}_3,\ \mathbf{e}_{31} = \mathbf{e}_3 \wedge \mathbf{e}_1,\ \mathbf{e}_{12} = \mathbf{e}_1 \wedge \mathbf{e}_2,$$
$$\mathbf{e}_{43} = \mathbf{e}_4 \wedge \mathbf{e}_3,\ \mathbf{e}_{42} = \mathbf{e}_4 \wedge \mathbf{e}_2,\ \mathbf{e}_{41} = \mathbf{e}_4 \wedge \mathbf{e}_1.$$

Any bivector in this basis takes the following form: $\mathbf{L} = v_x\mathbf{e}_{41} + v_y\mathbf{e}_{42} + v_z\mathbf{e}_{43} + m_x\mathbf{e}_{23} + m_y\mathbf{e}_{31} + m_z\mathbf{e}_{12}$, where the components are (v_x, v_y, v_z) correspond to the guiding vector of the straight line, and (m_x, m_y, m_z)— to the moment of the straight line. In order for this correspondence to be carried out up to the sign, the basis vectors are deliberately taken in the specified order, and not in ascending order of indices, as is usually customary.

This correspondence is easy to check if we take two points in the form of $\mathbf{p} = p_x\mathbf{e}_1 + p_y\mathbf{e}_2 + p_z\mathbf{e}_3 + p_w\mathbf{e}_4$ and $\mathbf{q} = q_x\mathbf{e}_1 + q_y\mathbf{e}_2 + q_z\mathbf{e}_3 + q_w\mathbf{e}_4$ and then find their outer product: $\mathbf{p} \wedge \mathbf{q} = (q_xp_w - p_xq_w)\mathbf{e}_{41} + (q_yp_w - p_yq_w)\mathbf{e}_{42} + (q_zp_w - p_zq_w)\mathbf{e}_{43} + (p_yq_z - p_zq_y)\mathbf{e}_{23} + (p_zq_x - p_xq_z)\mathbf{e}_{31} + (p_xq_y - p_yq_x)\mathbf{e}_{12}$.

Similarly, we can check that the outer product of a point and a straight line $\mathbf{L} \wedge \mathbf{p}$ will give a plane (trivector). In turn, the inner product of two planes (trivectors) will give a bivector (a straight line) and is interpreted as a straight line along which these two planes intersect. Moreover, in the case of parallelism, the components of the bivector with the basic 2-vectors $\mathbf{e}_{41}, \mathbf{e}_{42}, \mathbf{e}_{43}$ will be zero, which means that the line is not straight.

Having found the inner product of a straight line and a plane, we get an object of rank 1, that is, a vector or a point of intersection of a plane and a straight line.

A full-fledged description of projective geometry within the framework of geometric algebra requires the introduction of additional auxiliary operations, almost all of which can eventually be expressed through external or geometric multiplication.

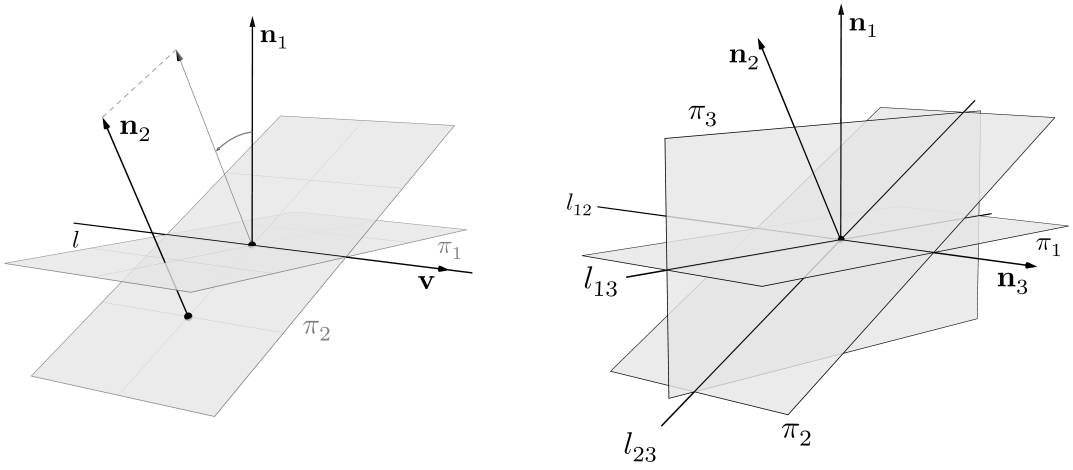
It is worth noting an alternative approach in which work is carried out with dual representations of objects. Within its framework, a bivector also corresponds to a straight line, but points and planes represent trivectors and vectors, respectively. This approach allows us to obtain correct results algebraically, but it seems less intuitive than the direct approach.

4. Software

The authors used a number of packages to work with geometric algebra in its non-projective version. These are primarily the `galgebra` package for SymPy, the `clifford` module (Python language) for numerical problems and `Grassmann.jl` for the Julia language.

It should be noted that in `galgebra` it is impossible to define a metric tensor containing zeros on the diagonal, which makes it unable to work in projective space. The clifford package has such a possibility, but it does not implement some necessary operations, such as right and left additions, which can be circumvented by using geometric multiplication by $\mathbf{e}_{1234}$ with the correct sign.

To test the formulas in practice, the authors used the Asymptote vector graphics language. This language is intended for creating vector illustrations. The language has a C-like syntax and allows you to add your own data structures. We used this opportunity to define data structures for a point, a straight line and a plane in a projective form, as well as basic operations for finding tangent and normal vectors, as well as points and lines of intersection of lines and planes. The resulting library prototype makes it easier to build three-dimensional illustrations? for example fig. 1.



(a) (b)

Figure 1: These images are constructed using the Asymptote language. All elements of the image are calculated in homogeneous coordinates: normal vectors of planes, guiding vectors of straight lines, points and straight intersections.

5. Conclusion

Its presentation in the language of geometric algebra can, on the one hand, simplify the study of the mathematical apparatus by technical specialists, and on the other hand, generalize it to large dimensions. It is important that not only primitives are generalized: points, lines and planes, but also their transformations: linear and affine. It becomes possible to combine complex numbers, quaternions and parabolic biquaternions within one formalism. All these objects are used in computer graphics, but are introduced for the most part as some ad-hoc constructions. Within the framework of geometric algebra, they are all generalized by the concept of a multivector and a geometric product. Currently, the mathematical theory is sufficiently developed, but there is no accessible presentation of it in textbooks. There are also no reliable libraries that implement all geometric algebra operations, especially in symbolic form.

References

[1] *Coxeter, H. S. M.* Projective Geometry, 2nd ed. New York: Springer-Verlag, 1987.

[2] *Hartley R., Zisserman A.* Multiple View Geometry in Computer Vision. — 2d ed. — Cambridge : Cambridge University Press, 2004. — 655 p.

[3] *Lengyel E.* Foundations of Game Engine Development. In 4 v. V. 1. Mathematics. — Lincoln, California : Terathon Software LLC, 2016. — 195 p.

Generalized Power Series Solutions of q-Difference Equations and the Small Divisors Phenomenon*

R.R. Gontsov[1], I.V. Goryuchkina[2]

[1]*Institute for Information Transmission Problems of RAS, Russia*

[2]*Keldysh Institute of Applied Mathematics of RAS, Russia*

e-mail: gontsovrr@gmail.com, igoryuchkina@gmail.com

Abstract

The problem of the convergence of generalized formal power series (with complex power exponents) solutions of q-difference equations is studied in the situation where the small divisors phenomenon arises; a sufficient condition of convergence generalizing corresponding conditions for classical power series solutions is obtained; an illustrating example is given.

Keywords: q-difference equation, generalized power series solution, convergence, small divisors

Our talk is based on the joint work with A. Lastra, see [4]. We consider a q-difference equation

$$F(z, y, \sigma y, \sigma^2 y, \dots, \sigma^n y) = 0, \qquad z \in \mathbb{C}, \tag{1}$$

where $F = F(z, y_0, y_1, \dots, y_n)$ is a polynomial and σ stands for the dilatation operator

$$\sigma : \ y(z) \mapsto y(qz),$$

$q \neq 0, 1$ being a fixed complex number. We study the question of the convergence of its *generalized* formal power series solutions $y = \varphi$ of the form

$$\varphi = \sum_{j=0}^{\infty} c_j z^{\lambda_j}, \qquad c_j, \lambda_j \in \mathbb{C}, \tag{2}$$

where $c_0 \neq 0$ and the sequence of the exponents λ_j possesses the following two properties:

(i) $\operatorname{Re}\lambda_j \leqslant \operatorname{Re}\lambda_{j+1}$ for all $j \geqslant 0$,

(ii) $\lim_{j\to\infty} \operatorname{Re}\lambda_j = +\infty$.

We note that the conditions (i), (ii) make the set of all generalized formal power series an algebra over $\mathbb{C}$. The definition of the dilatation operator extends naturally to this algebra after fixing the value of $\ln q$ by the condition $0 \leqslant \arg q < 2\pi$:

$$\sigma\Big(\sum_{j=0}^{\infty} c_j z^{\lambda_j}\Big) = \sum_{j=0}^{\infty} c_j q^{\lambda_j} z^{\lambda_j}.$$

Thus the notion of a generalized formal power series solution of (1) is correctly defined in view of the above remarks: such a series φ is said to be a *formal solution* of (1) if the substitution of $y_i = \sigma^i\varphi$, $i = 0, 1, \dots, n$, into the polynomial F leads to a generalized power series with zero coefficients.

*The research is carried out at IITP RAS under support of Russian Science Foundation, grant no. 22-21-00717, https://rscf.ru/en/project/22-21-00717/.

Formal solutions (2) generalize classical power series solutions of the form $\sum_{j=0}^{\infty} c_j z^j$. The convergence of the latter was widely studied within the last decades: there are two principally different cases, that of $|q| \neq 1$ (see [8], [5]) and that of $|q| = 1$, q not being a root of unity, where the small divisors phenomenon may arise (see [1], [2]). Namely, the coefficients c_j of a formal power series solution $\sum_{j=0}^{\infty} c_j z^j$ of (1) are determined recurrently via relations $Q(q^j)\, c_j = P_j(c_0, c_1, \ldots, c_{j-1})$, with some polynomials $Q, \{P_j\}$. Therefore the sequence q^j tends neither to infinity nor to zero if $|q| = 1$ and may come arbitrarily close to a root of Q, which may cause a high growth of the coefficients c_j obstructing the convergence of the series.

For the generalized formal power series solution (2) of (1), assume that each $F'_{y_i}(z, \varphi, \sigma\varphi, \ldots, \sigma^n\varphi)$ is of the form

$$\frac{\partial F}{\partial y_i}(z, \varphi, \sigma\varphi, \ldots, \sigma^n\varphi) = A_i z^{\gamma} + B_i z^{\gamma_i} + \ldots, \qquad \operatorname{Re}\gamma_i > \operatorname{Re}\gamma \geqslant 0,$$

$\gamma \in \mathbb{C}$ being the same for all $i = 0, 1, \ldots, n$, and at least one of the A_i's being non-zero. Then under a generic assumption on the power exponents λ_j of (2) that, starting with some $j_0 \in \mathbb{Z}_+$, the q^{λ_j}'s are not the roots of a non-zero polynomial

$$L(\xi) = A_n \xi^n + \ldots + A_1 \xi + A_0$$

of degree $\leqslant n$, one can assert that all $\lambda_j - \lambda_{j_0}$, $j > j_0$, belong to a finitely generated additive semi-group $\Gamma \subset \mathbb{C}$ whose generators $\alpha_1, \ldots, \alpha_s$ all have a positive real part (see Lemmas 1, 2 in [3]). Thus we may initially consider the formal solution (2) in the form

$$\varphi = \sum_{j=0}^{\infty} c_j z^{\lambda_j} = \sum_{j=0}^{j_0} c_j z^{\lambda_j} + \sum_{(m_1,\ldots,m_s)\in\mathbb{Z}_+^s\setminus\{0\}} c_{m_1,\ldots,m_s} z^{\lambda_{j_0}+m_1\alpha_1+\ldots+m_s\alpha_s}. \tag{3}$$

For such a formal series solution the small divisors phenomenon does not arise if all the α_k's lie *strictly above* or *strictly under* the line $\mathcal{L}$ passing through $0 \in \mathbb{C}$ and having the slope $\ln|q|/\arg q$ (or, equivalently, all the q^{α_k}'s lie *strictly inside* or *strictly outside* the unit circle). This condition defines an analogue (and generalization) of the case of $|q| \neq 1$ in the classical situation. The convergence of (3) under such a condition was studied in our previous work [3]. Contrariwise, the placement of α_k's on both sides of (or on) $\mathcal{L}$ may cause the small divisors phenomenon. The study of this situation is the main subject of our present talk and we propose the following theorem on the convergence of φ.

Theorem 1. *Let the generalized formal power series* (3) *satisfy* (1). *If* $\deg L = n$, $L(0) \neq 0$, *and for each root* $\xi = a$ *of the polynomial* $(\xi - q^{\lambda_{j_0}})L(\xi)$ *the following diophantine condition is fulfilled:*

$$|(\lambda_{j_0} + m_1\alpha_1 + \ldots + m_s\alpha_s)\ln q - \ln a - 2\pi m\mathrm{i}| > c\,(m_1 + \ldots + m_s)^{-\nu} \quad \text{for all } m_i \in \mathbb{Z}_+,\ m \in \mathbb{Z} \tag{4}$$

(*with the exception of* $m_1 = \ldots = m_s = 0$), *where* c *and* ν *are some positive constants, then* (3) *has a non-zero radius of convergence* (*that is, it converges uniformly in any sector* $S \subset \mathbb{C}$ *of sufficiently small radius with the vertex at the origin and of the opening less than* 2π *defining there a germ of a holomorphic function*).

Remarks 1. The diophantine condition of Theorem 1 is generically fulfilled. As for concrete examples, one can apply in particular Schmidt's result [6] from which it follows that (4) holds for $a = q^{\lambda_{j_0}}$, if

1. the real parts of all $\frac{1}{2\pi\mathrm{i}}\alpha_1 \ln q, \dots, \frac{1}{2\pi\mathrm{i}}\alpha_s \ln q$ are algebraic and together with 1 linearly independent over $\mathbb{Z}$ or

2. the imaginary parts of all $\frac{1}{2\pi\mathrm{i}}\alpha_1 \ln q, \dots, \frac{1}{2\pi\mathrm{i}}\alpha_s \ln q$ are algebraic and linearly independent over $\mathbb{Z}$.

(If L has roots $\xi = a$ other than $q^{\lambda_{j_0}}$ then the number $\frac{1}{2\pi\mathrm{i}} \ln(q^{\lambda_{j_0}}/a)$ should be added to the set of numbers in the above conditions 1, 2 for each such $a \neq q^{\lambda_{j_0}}$.)

The proof of the theorem is based on Siegel's ideas [7] of studying a first order equation $\sigma y = f(y)$ describing the linearization of a diffeomorphism f of $(\mathbb{C}, 0)$. This uses the majorant method adapted to our "multi-index case" for the construction of a convergent series majorizing (3).

Some particular placements of the α_k's with respect to the line $\mathcal{L}$ allow one to weak assumptions of Theorem 1. Therefore we formulate a separate statement which follows from Theorem 1 and distinguishes all these particular cases of the placement of the α_k's on the plane.

Theorem 2. *The statement of Theorem 1 holds in the following particular cases:*

a) $L(0) \neq 0$ *and all the* α_k*'s lie* strictly above *the line* $\mathcal{L}$*;*

b) $\deg L = n$ *and all the* α_k*'s lie* strictly under *the line* $\mathcal{L}$*;*

c) *all the* α_k*'s lie* on *the line* $\mathcal{L}$ *and the condition* (4) *is fulfilled for those roots* $\xi = a$ *of the polynomial* $(\xi - q^{\lambda_{j_0}})L(\xi)$ *that lie on the circle* $\{|\xi| = |q^{\lambda_{j_0}}|\}$*;*

d) $L(0) \neq 0$*, all the* α_k*'s lie* above *or* on *the line* $\mathcal{L}$*, and the condition* (4) *is fulfilled for those roots* $\xi = a$ *of the polynomial* $(\xi - q^{\lambda_{j_0}})L(\xi)$ *that lie inside the closed disk* $\{|\xi| \leqslant |q^{\lambda_{j_0}}|\}$*;*

e) $\deg L = n$*, all the* α_k*'s lie* under *or* on *the line* $\mathcal{L}$*, and the condition* (4) *is fulfilled for those roots* $\xi = a$ *of the polynomial* $(\xi - q^{\lambda_{j_0}})L(\xi)$ *that lie outside the open disk* $\{|\xi| < |q^{\lambda_{j_0}}|\}$.

Note that the small divisors phenomenon for classical power series solutions of (1) arising in the case of $q = e^{2\pi\mathrm{i}\omega}$, $\omega \in \mathbb{R} \setminus \mathbb{Q}$, and studied in [1], [2], is contained in the case c) of Theorem 2: the line $\mathcal{L}$ coincides with the Ox axis, $\lambda_{j_0} = 0$, the set of power exponents is generated by the unique $\alpha_1 = 1 \in \mathcal{L}$ and the condition (4) is reduced to

$$|j\omega - (1/2\pi\mathrm{i}) \ln a - m| > cj^{-\nu} \qquad \text{for all } j \in \mathbb{N},\ m \in \mathbb{Z},$$

in this case (see Th. 6.1 in [1] and Th. 8 in [2]).

Example 1. Consider a kind of a q-difference analogue of the Painlevé III equation with $a = b = 0$, $c = d = 1$:

$$y\,\sigma^2 y - (\sigma y)^2 - z^2 y^4 - z^2 = 0,$$

where $q = e^{2\mathrm{i}\pi\omega}$, $\omega \in \mathbb{R} \setminus \mathbb{Q}$. This equation possesses a two-parameter family of formal solutions:

$$\varphi = \sum_{m_1, m_2 \in \mathbb{Z}_+} c_{m_1,m_2}\, z^{r+m_1(2-2r)+m_2(2+2r)},$$

where the complex coefficient $c_{0,0} \neq 0$ is arbitrary, $-1 < \operatorname{Re} r < 1$, the other complex coefficients c_{m_1,m_2} are uniquely determined by $c_{0,0}$ and r. The numbers $q^{2\pm 2r}$ lie on the opposite sides of the unit circle (if $\operatorname{Im} r \neq 0$) or on the unit circle (if $\operatorname{Im} r = 0$), whereas the second degree polynomial $L(\xi) = c_{0,0}(\xi - q^r)^2$ does not vanish at 0. Therefore taking r, ω in such a way that the condition of Theorem 1 holds,

$$|(m_1(2-2r)\omega + m_2(2+2r)\omega - m| > c\,(m_1 + m_2)^{-\nu}$$

for some positive c and ν, we obtain the convergent φ. For example, it is sufficient for ω to be algebraic and for r simply to have a non-zero imaginary part. Indeed, then for any $m_1 \neq m_2$ one has

$$|(m_1(2-2r)\omega + m_2(2+2r)\omega - m| \geqslant 2|\omega \operatorname{Im} r| \cdot |m_2 - m_1| > c\,(m_1+m_2)^{-\nu},$$

whereas for $m_1 = m_2$ it follows that

$$|(m_1(2-2r)\omega + m_2(2+2r)\omega - m| = |4\omega\, m_1 - m| > c\, m_1^{-\nu}.$$

References

[1] *Bézivin J.-P.* Sur les équations fonctionnelles aux q-différences. Aequat. Math. 1992. Vol. 43. P. 159–176.

[2] *Di Vizio L.* An ultrametric version of the Maillet–Malgrange theorem for nonlinear q-difference equations. Proc. Amer. Math. Soc. 2008. Vol. 136, N. 8. P. 2803–2814.

[3] *Gontsov R., Goryuchkina I., Lastra A.* On the convergence of generalized power series solutions of q-difference equations. Aequat. Math. 2022. Vol. 96, N. 3. P. 579–597.

[4] *Gontsov R., Goryuchkina I., Lastra A.* Small divisors in the problem of the convergence of generalized power series solutions of q-difference equations. 2022. arXiv: 2209.09365, 16pp.

[5] *Li X., Zhang C.* Existence of analytic solutions to analytic nonlinear q-difference equations. J. Math. Anal. Appl. 2011. Vol. 375. P. 412–417.

[6] *Schmidt W.M.* Simultaneous approximation to algebraic numbers by rationals. Acta Math. 1970. Vol. 125. P. 189–201.

[7] *Siegel C.L.* Iteration of analytic functions. Ann. of Math. 1942. Vol. 43, N. 4. P. 607–612.

[8] *Zhang C.* Sur un théorème du type de Maillet–Malgrange pour les équations q-différences-différentielles. Asymptot. Anal. 1998. Vol. 17, N. 4. P. 309–314.

Automatic Differentiation. Practical Aspects

A.Yu. Gorchakov, V.I. Zubov
Federal Research Center "Computer Science and Control" of RAS, Russia
e-mail: agorchakov@frccsc.ru, vladimir.zubov@mail.ru

Abstract

The paper discusses the practical aspects of calculating derivatives in solving optimization and optimal control problems. Direct and inverse methods for calculating the gradient of scalar and vector functions are described. Methods of obtaining the gradient calculation code and packages implementing automatic differentiation methods are given.

Keywords: fast automatic differentiation, multi-stage process, standard software packages

1. Introduction

Currently, numerical methods for solving optimization and optimal control problems are increasingly being used in various fields of science, technology and production. They occupy a special place in modern methods of solving real problems. Due to the intensification of production in recent decades, there has been a growing interest in optimization problems and optimal control problems of complex dynamic systems.

In practice, when solving optimization and optimal control problems, gradient methods are used using the first derivatives or gradient (conjugate gradient method, quasi-Newtonian methods), the first derivatives of the vector function or Jacobian (Levenberg-Marquardt method), the second derivatives (Newton methods).

The simplest way to calculate the gradient of a relatively complex function is to determine components of the gradient with the help of the finite difference method. However, this approach does not allow you to get the exact value of the gradient. The difficulties encountered when using the finite difference method to calculate the gradient of the cost function in problems of optimal control of complex systems were described in [1].

Another approach is to use the technique of automatic differentiation (AD). The AD method is much more effective than analytical differentiation (more precisely, formulas obtained analytically) and finite-difference calculation of derivatives. A general approach to the differentiation of composite functions was proposed by Yevtushenko in [2, 3, 4]. In particular, it was shown that the AD methodology allows us to consider various problems in a unified way. For example, using the general differentiation formulas given in [2, 3, 4], it is easy to derive AD formulas for determining the gradient of a function of several variables. The definition of a function is presented as a multi-stage process with the introduction of new state variables. They are functions of independent variables with respect to which derivatives of this function are calculated.

The direct application of this methodology gives excellent results both in accuracy and in the speed of gradient calculation, but in some cases it may be too time-consuming, requiring a lot of analytical work to derive the necessary formulas. In order to reduce labor costs, it is possible to use existing rapid automatic differentiation packages, for example, Adept [5] or CodiPack [6] packages.

2. Methodology of automatic differentiation

Let $z \in \mathbb{R}^n$ and $u \in \mathbb{R}^r$ be vectors. Differentiable functions $W(z,u)$ and $\Phi(z,u)$ define maps $W : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^1, \Phi : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^n$. The Vectors z and u satisfy the following system of n nonlinear scalar equations $\Phi(z,u) = 0$. If the matrix $\Phi_z^T(z,u)$ is non-degenerate, then the complex function $\Omega(u) = W(z(u),u)$ is differentiable and its gradient with respect to variables is calculated using formulas (1)-(2) for the forward differentiation method.

$$\frac{d\Omega}{du} = W_u(z(u),u) + N(u)W_z^T(z(u),u), \tag{1}$$

where the rectangular $r \times n$ matrix $N(u)$ is from the solution of a linear algebraic system:

$$W_z^T(z,u) + N(u)\Phi_z^T(z,u) = 0 \tag{2}$$

and (3)-(4) for reverse differentiation method:

$$\frac{d\Omega}{du} = W_z(z(u),u) + \Phi_z^T(z(u),u) \cdot p(u), \tag{3}$$

where the vector $p \in \mathbb{R}^n$ is from the solution of a linear algebraic system:

$$W_z(z,u) + \Phi_z^T(z,u) \cdot p. \tag{4}$$

Here and further, the index T denotes transposition, the subscripts z, u denotes partial derivatives of functions with respect to vectors z and u:

$$W_u = \frac{\partial W}{\partial u}, W_z = \frac{\partial W}{\partial z}, \Phi_u^T = \frac{\partial \Phi^T}{\partial u}, \Phi_z^T = \frac{\partial \Phi^T}{\partial z}. \tag{5}$$

We will also denote i-th and j-th components of vectors z and u as z_i, z_j, u_i, u_j.

The forward method allows you to calculate the gradient of the function $T(grad(f))$ in a time not exceeding:

$$T(grad(f)) \leq C_{forward} \cdot r \cdot T(f), \tag{6}$$

and the reverse method in time:

$$T(grad(f)) \leq C_{reverse} \cdot T(f), \tag{7}$$

where $T(f)$ is the calculation time of the function and $C_{forward}$ and $C_{reverse}$ — some constants. Theoretical estimates of these constants (excluding access time to RAM and the possibility of parallel computing) are given in [2]:

$$C_{forward} = C_{reverse} = 3. \tag{8}$$

Practically achievable estimates for $C_{reverse}$ are given in [5]: 2-4 for the manual method of encoding derivatives and 2.7-4 for the package Adept. For the forward differentiation method, the practically achievable estimates $C_{forward}$, as a rule, do not exceed the theoretical ones.

In the case of a vector function $f(u) : \mathbb{R}^r \to \mathbb{R}^m$, the matrix of the first derivatives of the function (the Jacobi matrix) is calculated in a time not exceeding:

$$T(grad(f)) \leq C_{forward} \cdot r \cdot T(f), \tag{9}$$

and the reverse method in time:

$$T(grad(f)) \leq C_{reverse} \cdot m \cdot T(f), \tag{10}$$

As can be seen from (7) and (8), the inverse differentiation method is effective for $r >> m$.

The simplest and most common way to implement methods of automatic differentiation is to use the operator overloading mechanism, which is available in many modern programming languages. The forward differentiation method is implemented quite straightforwardly — a new data type is introduced, which contains not only the value of a variable, but also all the values of its derivatives for one or more input variables [8]. The reverse method is more difficult to implement, since when calculating a function, it is necessary to write the results of calculating each basic elementary function into memory, creating an "information graph", and then go through this graph in reverse order, calculating derivatives [4].

3. Automatic differentiation packages

The electronic resource of the https://www.autodiff.org is dedicated to automatic differentiation packages. On this resource you can find packages that support the calculation of derivatives in more than 10 high-level algorithmic languages. In particular, C/C++, Fortran and Python are supported. For the C/C++ language, we recommend using the Adept [5] and CoDiPack [6] packages.

The Adept package supports 2-pass forward and reverse methods for obtaining gradients of scalar and vector functions using an information graph. The package is easy to use and modify. Therefore, if you need to calculate gradients using the reverse method, then this will be the best choice. The program is modified using formal transformations – the *double* data type is replaced with a special *adouble* type; then a few lines of initialization and gradient calculation are added. For example:

Initial program

- double W(const double x[2]) {
- double y = 4.0;
- double s = 2.0*x[0] + 3.0*x[1]*x[1];
- y *= sin(s);
- return y;
- }

Modified program

- adouble W(const adouble x[2]) {
- adouble y = 4.0;
- adouble s = 2.0*x[0] + 3.0*x[1]*x[1];
- y *= sin(s);
- return y;}
- using namespace adept;
- Stack stack;
- adouble x[2] = x_val[0], x_val[1];

- stack.new_recording();
- adouble Y = W(x);
- Y.set_gradient(*Y_ad);
- stack.reverse();
- x_ad[0] = x[0].get_gradient();
- x_ad[1] = x[1].get_gradient();
- *Y_ad = Y.get_gradient();

The CoDiPack [6] package has great functionality (but also great complexity), allows you to efficiently calculate the first derivatives (and derivatives of higher orders) both forward and reverse methods. The advantages of the package are that the direct method is parallelized using AVX2, AVX512 technologies and does not require the allocation of RAM for the information graph.

References

[1] *A. F. Albu, V. I. Zubov* Determination of functional gradient in an optimal control problem related to metal solidification. Comput. Math. Math. Phys. 49 (1), 47–70 (2009).

[2] *Aida-zade K. R., Evtushenko Y. G.* Fast automatic differentiation on computers //Matematicheskoe modelirovanie. – 1989. – Vol. 1. – N. 1. – P. 120-131.

[3] *Evtushenko Y.* Computation of exact gradients in distributed dynamic systems //Optimization methods and software. – 1998. – Vol. 9. – N. 1-3. – P. 45-75.

[4] *Evtushenko Y. G.* Optimization and fast automatic differentiation. Dorodnicyn Computing Center of Russian Academy of Sciences, 2013. [in Russian].

[5] *R. J. Hogan* Fast reverse-mode automatic differentiation using expression templates in C++. ACM Transactions on Mathematical Software (TOMS). 2014. – Vol. 40. – N. 4. – P. 26

[6] *T. Albring et al.* An aerodynamic design framework based on algorithmic differentiation //ERCOFTAC Bull. – 2015. – Vol. 102. – P. 10-16

[7] *Kolmogorov A.N.* On the concept of algorithm. Uspekhi Mat. Nauk. 1953. Vol. 8, N. 4 (56). P. 175–176. (In Russian)

[8] *Griewank and A. Walther* Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation 2nd Ed. SIAM. 2008

Symbolic Investigation of the Plane Equilibria of the System of Two Connected Bodies on a Circular Orbit

S.A. Gutnik
MGIMO University, Russia
Moscow Institute of Physics and Technology, Russia
e-mail: s.gutnik@inno.mgimo.ru

Abstract

Computer algebra methods are used to study the plane equilibrium orientations of a system of two bodies connected by a spherical hinge that moves on a circular orbit. The main attention is given to the study of the equilibria of the two-body system in the plane of the circular orbit, in the plane perpendicular to the circular orbital plane and in the plane tangent to the circular orbital plane. A new method is proposed for transforming the system of trigonometric equations determining the equilibria into a system of polynomial equations, which in turn are reduced by calculating the resultant to a single algebraic equation. The domains with an identical number of equilibria are classified using algebraic methods for constructing a discriminant hypersurface.

Keywords: connected rigid bodies, spherical hinge, circular orbit, Lagrange equations, algebraic equations, resultant, equilibrium orientation, computer algebra

1. Introduction

In the present work, we use symbolic computations to investigate the equilibrium orientations of a system of two bodies (satellite and stabilizer) connected by a spherical hinge that moves along a circular orbit. Determining the equilibria for the system of bodies on a circular orbit is of practical interest for designing composite gravitational orientation systems of satellites that can stay on the orbit for a long time without energy consumption. The dynamics of various composite schemes for satellite-stabilizer gravitational orientation systems was discussed in detail in [1] and [2]. In [3], [4], [5] equilibrium orientations for the two-body system in the orbital plane were found in the case where the spherical hinge is positioned at the intersection of the principal central axes of inertia of the satellite and stabilizer, as well as in the case where the hinge is positioned on the line of intersection between two planes formed by the principal central axes of inertia of the satellite and stabilizer. In this work, we study the equilibrium orientations of the two-body system in the orbital plane, in the plane perpendicular to the circular orbital plane and in the plane tangent to the circular orbital plane in the case when the hinge is positioned on the line of intersection between two planes formed by the principal central axes of inertia of the satellite and stabilizer.

2. Equilibrium Orientations of a System of Two Bodies

We consider a system of two bodies connected by a spherical hinge that moves along a circular orbit [1]. To write the corresponding equations of motion, we introduce the following right-handed rectangular coordinate systems. The orbital coordinate system is $OXYZ$. The $OZ-$ axis is directed along the radius vector that connects the Earth's center of mass with the center of mass of the two-body system O, the $OX-$ axis is directed along the linear velocity vector of the center of mass O, while the $OY-$ axis is directed along the normal to the orbital plane. The coordinate system of the ith body ($i=$ 1, 2) is $O_i x_i y_i z_i$, where the axis of these coordinate systems are the principal central axes of inertia of the ith body.

The orientation of coordinate system $O_ix_iy_iz_i$ with respect to the orbital coordinate system is determined using aircraft angles α_i (pitch), β_i (yaw), and γ_i (roll) [1].

Let (a_i, b_i, c_i) are the coordinates of spherical hinge in the coordinate system $O_ix_iy_iz_i$; A_i, B_i, C_i are the principal central moments of inertia of the each bodies; $M = M_1M_2/(M_1 + M_2)$; M_i is the mass of the ith body.

1. We consider the first case where the hinge is located at the line of intersection of two planes formed by the principal central axes of inertia of the satellite and the stabilizer when $b_1 = b_2 = 0$ and the equilibrium orientations of the two-body system are in the orbital plane ($\alpha_1 \neq 0, \alpha_2 \neq 0, \beta_1 = \beta_1 = 0, \gamma_1 = \gamma_2 = 0$) [5]. Then, in the coordinate systems connected to body 1 and body 2, the spherical hinge has coordinates $(a_i, 0, c_i)$.

Using the expression of the kinetic energy T of the system [5]

$$\begin{aligned} T &= 1/2\big(B_1 + M(a_1^2 + c_1^2)\big)(\dot{\alpha_1} + \omega_0)^2 + 1/2\big(B_2 + M(a_2^2 + c_2^2)\big)(\dot{\alpha_2} + \omega_0)^2 \\ &- M\big((a_1a_2 + c_1c_2)\cos(\alpha_1 - \alpha_2) \\ &- (a_1c_2 - a_2c_1)\sin(\alpha_1 - \alpha_2)\big)(\dot{\alpha_1} + \omega_0)(\dot{\alpha_2} + \omega_0), \end{aligned} \tag{1}$$

and the force function

$$\begin{aligned} U &= -3/2\omega_0^2\big((A_1 - C_1)\sin^2\alpha_1 + (A_2 - C_2)\sin^2\alpha_2\big) \\ &+ 3/2M\omega_0^2\big((a_1\sin\alpha_1 - c_1\cos\alpha_1) - (a_2\sin\alpha_2 - c_2\cos\alpha_2)\big)^2 \\ &+ M\omega_0^2\big((a_1a_2 + c_1c_2)\cos(\alpha_1 - \alpha_2) - (a_1c_2 - a_2c_1)\sin(\alpha_1 - \alpha_2)\big), \end{aligned} \tag{2}$$

the equations of motion for this system can be written as the Lagrange equations of the second kind

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\alpha_i}} - \frac{\partial T}{\partial \alpha_i} - \frac{\partial U}{\partial \alpha_i} = 0, \quad i = \overline{1,2}. \tag{3}$$

By applying symbolic differentiation in (3) using Wolfram Mathematica system [6], equations (3) can be presented in the form of a system of second-order ordinary differential equations in variables α_1 and α_2.

Then from Lagrange equations we can obtain the stationary trigonometric system which allows us to determine equilibrium orientations for the system of two bodies connected by the spherical hinge in the orbital coordinate system. Seting in Lagrange equations $\alpha_1 = \alpha_{01} = \text{const}$, $\alpha_2 = \alpha_{02} = \text{const}$ we obtain the stationary equations

$$\begin{aligned} ((A_1 - C_1)/M)\sin\alpha_{01}\cos\alpha_{01} - (a_1\cos\alpha_{01} + c_1\sin\alpha_{01})(a_1\sin\alpha_{01} - c_1\cos\alpha_{01}) \\ -(a_1\cos\alpha_{01} + c_1\sin\alpha_{01})(c_2\cos\alpha_{02} - a_2\sin\alpha_{02}) = 0, \\ ((A_2 - C_2)/M)\sin\alpha_{02}\cos\alpha_{02} - (a_2\cos\alpha_{02} + c_2\sin\alpha_{02})(a_2\sin\alpha_{02} - c_2\cos\alpha_{02}) \\ -(a_2\cos\alpha_{02} + c_2\sin\alpha_{02})(c_1\cos\alpha_{01} - a_{01}\sin\alpha_{01}) = 0. \end{aligned} \tag{4}$$

The trigonometric system (4) cannot be solved analytically for two unknown aircraft angles. To solve system (4), we use the universal approach whereby the sines and cosines of angles α_{0i} are replaced by their tangents $t_i = \tan(\alpha_{0i})$.

As a result, we obtain from (4) the algebraic system of two equations in two unknowns t_1, t_2

$$\begin{aligned} \bar{a}_0t_1^3 + \bar{a}_1t_1^2 + \bar{a}_2t_1 + \bar{a}_3 = 0, \\ \bar{b}_0t_1^2 + \bar{b}_1t_1 + \bar{b}_2 = 0, \end{aligned} \tag{5}$$

where $\bar{a}_i, \bar{b}_i$ are the polynomials depending on six system parameters $a_1, a_2, c_1, c_2, d_1 = (A_1 - C_1)/M, d_2 = (A_2 - C_2)/M$ and tangent t_2.

By using the resultant approach to eliminate t_1 from system (5) and symbolic computations in Wolfram Mathematica 12.1 [6] to find the determinant of the resultant matrix, we obtain a twelfth-order algebraic equation in one unknown t_2, which upon factorization, turns into a product of three polynomials: $P(t_2) = P_1(t_2)P_2(t_2)P_3(t_2) = 0$. Here $P_1(t_2)$, $P_2(t_2)$ are second-order polynomials and $P_3(t_2)$ is an eighth-order polynomial, the coefficients of which are polynomials in six system parameters.

By the definition of the resultant, each root of equation $P(t_2) = 0$ corresponds to one common root of system (5). The algebraic equation obtained has the even number of real roots, which does not exceed 12. By substituting real root of algebraic equation $P(t_2) = 0$ into the equations of system (5), we find the common root of these equations. It can be shown that two equilibrium solutions of the original system correspond to each real root of equations (2).

Since the total number of real roots of $P(t_2) = 0$ does not exceed 12, the satellite–stabilizer system in the plane of the circular orbit can have no more than 24 equilibrium orientations in the orbital coordinate system. Using obtained equations for each set of system parameters, we can determine all equilibrium orientations of the satellite–stabilizer system in the orbital coordinate system. Each of the polynomial $P_1(t_2)$, $P_2(t_2)$ and $P_3(t_2)$ describes the separate class of equilibrium orientations.

To investigate the number of equilibrium solutions for the satellite-stabilizer system, we define domains with equal numbers of real roots of $P_3(t_2) = 0$ in the space of the six parameters. For this purpose, we construct a discriminant hypersurface of this polynomial, which defines the boundary of the domains with equal numbers of real roots.

2. Using the above approach we can obtain the results of investigation the equilibrium orientations of the two-body system in the plane perpendicular to the circular orbital plane and in the plane tangent to the circular orbital plane. In our works [7] and [8], it was shown that the satellite-stabilizer system can have up to 24 equilibrium orientations in the orbital coordinate system in the plane orthogonal to the orbital plane and in the plane tangent the orbital plane.

In the each of the above cases the equilibrium orientations of the two-body system in the plane perpendicular to the circular orbital plane and in the plane tangent to the circular orbital plane are defined by the real roots of the twelfth-order algebraic equation in one unknown $\tan\gamma_{02}$ or $\tan\beta_{02}$.

The use of computer algebra methods allowed us to solve the classical problem of space flight mechanics in a fairly simple form.

References

[1] *Sarychev, V.A.* Problems of orientation of satellites, Itogi Nauki i Tekhniki. Ser. Space Research, Vol. 11. VINITI, Moscow (1978) (in Russian)

[2] *Rauschenbakh, B.V., Ovchinnikov, M.Yu., McKenna–Lawlor, S.* Essential Spaceflight Dynamics and Magnetospherics. Kluwer Academic Publishers (2003)

[3] *Sarychev V.A.* Relative equilibrium orientations of two bodies connected by a spherical hinge on a circular orbit. Cosmic Research. 1967. Vol. 5, P. 360–364.

[4] *Gutnik S.A., Sarychev V.A.* Symbolic investigation of the dynamics of a system of two connected bodies moving along a circular orbit. In: England, M., Koepf, W., Sadykov, T.M., Seiler, W.M., Vorozhtsov, E.V. (eds.) CASC 2019. LNCS, Vol. 11661, P. 164–178. Springer, Cham (2019)

[5] *Gutnik S.A., Sarychev V.A.* Research into the Dynamics of a System of Two Connected Bodies Moving in the Plane of a Circular Orbit by Applying Computer Algebra Methods. Computational Mathematics and Mathematical Physics. 2023. Vol. 63, N. 1. P. 106–114.

[6] *Wolfram S* The Mathematica Book, 5th edn. Wolfram media, Inc. Champaign (2003)

[7] *Gutnik S.A., Sarychev V.A.* Symbolic methods for studying the equilibrium orientations of a system of two connected bodies in a circular orbit. Programming and Computer Software. 2022. Vol. 48, N. 2. P. 73–79.

[8] *Gutnik S.A., Sarychev V.A.* Computer algebra methods for searching the stationary motions of the connected bodies system moving in gravitational field. Math. Comput. Sci. 2022. Vol. 16, P. 1–15.

Regularity Criterion for a Linear Differential System with Meromorphic Coefficients

D.O. Ilyukhin[2], A.V. Parusnikova[1]

[1] *HSE University, Russia*

[2] *Volgograd MOU secondary school No. 18, Russia*

e-mail: dennic.96@mail.ru, aparusnikova@hse.ru

Abstract

Regularity criterion for the singular point of a linear meromorphic system is obtained. Its proof is based on trasform of the system to a linear equation with meromorphic coefficients. We can check, whether the singular point of the system is regular using computer algebra system.

Keywords: computer algebra, regular point

We prove the regularity criterion for the singular point $t = 0$ of a second order system $\dot{x} = A(t)x$, where $A(t)$ is the coefficient matrix $a_{ij}(t)$, $i, j = 1, 2$ with meromorphic elements. The proof of this criterion is based on several theoretical concepts and properties, including these definitions: singular point, Fuchs type point, meromorphic function [1, 2]. Based on the main lemmas, a linear system was obtained using a linear gauge transformation, from which a transition to a second-order linear differential equation is possible.

A regularity criterion for a n-th order system is obtained. Computations can be carried out using a computer algebra system.

The preliminary version of this work can be found in [3].

Acknowledgements

This part of work was supported by the Russian Science Foundation, grant no. 19-71-10003, https://rscf.ru/en/project/19-71-10003/.

We are grateful to R.R. Gontsov.

References

[1] *Gontsov R.R.* Refined Fuchs inequalities for systems of linear differential equations / Izv. Math. 2004. Vol. 68. N. 2. P. 259–272.

[2] *Zoladek H.* The monodromy group / Instytut matematyczny PAN. - Basel: Birkhauser Verlag. (2006).

[3] *Ilyukhin D.O., Parusnikova A.V.* Regularity criterion for a linear differential small order system with meromorphic coefficients / Trudy "Priokskoi Nauchnoi Konferentsii GSGU "Differential Equations and Related Issues". 2019. P. 65–73. (In Russian)

Symbolic Calculus for Optimal Control in Multi-Agent Economic Model

B.B. Iusup-Akhunov[1], I.G. Kamenev[2], A.A. Zhukova[2], N.P. Pilnik[2,3]

[1] *The Phystech School of Applied Mathematics and Informatics, Moscow Institute of Physics and Technology, Russia*

[2] *Federal Research Center "Computer Science and Control" of RAS, Russia*

[3] *Department of Applied Economics, Faculty of Economic Sciences, HSE University, Russia*

e-mail: batyrhan@phystech.edu, igekam@gmail.com, sasha.mymail@gmail.com, u4d@yandex.ru

Abstract

This paper presents the development of the support system for modeling socio-economic processes based on an open source platform. This system is based on the approach of the ECOMOD system developed for complex economic models where agents plan decisions based on optimal control problems. The system enables analysis of a model containing a set of several agents, each either solving an optimal control planning problem or following a scenario. The combined descriptions form a complex system of nonlinear relations, which is difficult to write down without errors in mathematical expressions on paper or using computer. The system includes elements for verifying the correctness of the model record: balances, dimensions etc.

Keywords: optimal control, symbolic calculation, economic modeling

1. Symbolic calculus to support for economic modeling

The representative agent principle is one of the key foundations of applied macroeconomic modeling. In a broad sense, it means that the system under study is described as the interaction of rational Decision-Makers who act in accordance with their goals, constraints and available information. In this interpretation, even most of social and many ecological models are also agent-based. In general, the logic of constructing an agent's problem starts from description of agents' optimization problems separately, then interaction models (markets, balances, etc.) are added and then comes to numerical and scenario analysis of the solution. Such models require not only an adequate description of agent decision-making and agent interactions, but also technological means of working with the complex mathematical formulation of the model. Typically it is a system of nonlinear relations, which is difficult to write down without errors in mathematical expressions on paper. This requires software systems capable of: support modeling at all stages, starting with the formal description of the model and ending with the presentation of computation results; simplification of the system of model relations, preferably preserving the attributes of equations and variables; automatic verification the correctness of the dimensions of variables in the ratios, which avoids errors and erroneous calculations due to incorrect formulation of the problem; verifies the balances of money and product transfers between agents.

The existing systems are built on commercial software tools, strongly limited by rigid assumptions about the nature of the model. The most common is DYNARE [1] based on a commercial Matlab product, as well as a less common one, named EcoMod, but developed by another group and with a different approach (static general equilibrium of the economy), also based on Matlab [2, 3]. Software for automated model processing based on open source platforms is beginning to develop (both general, see Wolfram Alfa, and specialized, see

Dolo [4, 5]). However, the vast majority of such systems are aimed at automated research of the model "as is", and not at automating the construction of models.

The authors of the paper propose a support system based on the previous realization of the support system ECOMOD, developed by the group led by I. G. Pospelov at the Computing Center of RAS [7, 6], and applied in a wide range of projects on modeling national economies. Complex systems, especially in economics, are capable of self-development exhibit similar challenges when modeling. At the same time, due to the preferences of the authors or customers of the model computations, these structures use different sets of concepts and, being simplified descriptions, neglect individual deviations from the patterns described in them [8, 9]. Sometimes, one team of researchers uses different constructions when modeling economic subsystems - bank, international trader, partial market, with different details or, conversely, aggregation of indicators [10]. The description of micro-systems in an applied model can use a large number of indicators that are linked by several ratios [11], which makes the task of analyzing the model difficult from the point of view of calculations. Despite all its advantages (the Ecomod system supports basic checks and has a device for integrating models of different agents), it is based on the non-free Maple platform and does not have a special interface designed for third-party users.

2. System architecture

The system must be able to solve the following tasks:

- automation of data input and output in the generally accepted mathematical format (LaTeX);
- cross-checking the system of variables, informing the user about the errors committed in it and suggestions of typical fixes for identified problems
- information of model balances, identification of problems of unclosed and unrelated interactions between agents;
- check the dimensionality of equations;
- automated output of optimality conditions.

The ECOMOD [12] system is an end-to-end processor of agent models and the output of optimality conditions for them, implemented in Python. The SymPy library is used as the main tool to convert symbolic expressions and check their correctness. External interfaces to work with the system, such as LaTeX, are also integrated.
As subsystems, ECOMOD contains : **Support engine**, **Core** and **Export & Import** modules.

The input and output points of the system are the following representations of the agent models:

1. Native Python
   One can describe models using the internal Python (SymPy) interface and use the system as a python-library in exploring your models and developing them further. This representation is used to transfer information within the system. All other representations are translated with appropriate processing modules to this data model. All expressions are fully compatible with SymPy syntax, so you can operate with those using any SymPy functions.

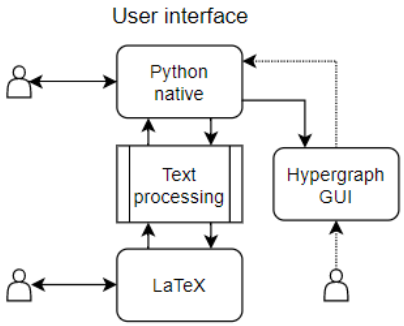


Figure 1: Use cases of the system. The dotted lines indicate links that have not yet been implemented

2. LaTeX
   One can describe models by importing agent models written as a LaTeX structure file. The internal structure of the file must be represented as YAML (KV-storage). Such a representation is convenient, when operating the system as a black box, which will derive the optimality conditions for the economic model. For unambiguous processing of LaTeX formulas in the system, rules for writing LaTeX formulas have been suggested. Examples can be found in the attached repository. To form the final LaTeX file, the template provided in the jinja2 system is used.

3. Hypergraph GUI
   One can also visualize the model as a hypergraph where the vertices correspond to agents and the edges correspond to the flows between the agents. The hypernetx library is responsible for the visualization In the future, we plan to create a visual interface that allows interactive work with the hypergraph, ie create and modify agent models.

The kernel and the support system are paired to process and validate economic models. The support system checks the completeness of the system of equations by comparing the set of variables used and the declared variables. It also checks the dimensionality of equations and inequalities in the model by substituting the dimensions of variables with random coefficients. The kernel, on the other hand, is a set of transformation functions written for Sympy expressions. The set of these functions computes auxiliary objects (e.g. Lagrangian) as well as optimality conditions for the control problem (Euler equations, transversality conditions, etc.) for further import of these objects for analysis.

3. Conclusions

Automation of work on agency models allows you to create complex models of complex systems that are based not on statistical patterns, but on the logic of optimal choice. Such models make it possible to find optimal solutions even in an unstable external environment, when the social or technical system as a whole switches between qualitatively different modes. The main limitation in the development of system agent models remains precisely the high probability of errors in the description of agents, which increases rapidly with an increase in the number of agents.

The presented architecture of the automatic processing of agent models already has all the main advantages of the Ecomod system, while at the same time having the main

advantages of competing systems: implementation on publicly available programming tools (Python SymPy), the possibility of portable deployment on a local device without purchasing commercial licenses. The feedback tools embedded in the system are significantly superior to existing analogues.

This allows us to count on the great potential of practical application of the system, primarily in the construction of models of general equilibrium of large systems (both in economics and in other complex systems that can be formalized by the same types of mathematical models).

References

[1] *K. Yano*, Dynare and Dynamic Stochastic General Equilibrium Models: Application to New Keynesian Models. Economic Analysis, vol. 181, 2009, pp. 155–190.

[2] *R. Kirkby*, Quantitative Macroeconomics: Lessons Learned from Fourteen Replications, Computational Economics, 2022, pp. 1–22.

[3] *A. Bayar, B. Bratta, S. Carta, P. Di Caro, M. Manzo, C. Orecchia*, Assessing the effects of VAT policies with an integrated CGE-microsimulation approach: evidence on Italy, Working paper No. wp2021–14., 2001.

[4] P. Winant, *Dolo* [Electronic repository]. URL: https://github.com/econforge/dolo.

[5] *L. Maliar, S. Maliar, P. Winant*, Deep learning for solving dynamic economic models. Journal of Monetary Economics, vol.122, 2021, pp. 76–101.

[6] *Pilnik N., Radionov S.* The model of the production side of the Russian economy. Advances in Systems Science and Applications, 2021, V. 21 N.3. pp. 63–74.

[7] *Vasilyev, S. B., Pilnik, N. P., Radionov, S. A.* The relaxation of complementary slackness conditions in dynamic general equilibrium models. Mathematical Models and Computer Simulations, 2019. Vol. 11, 611–621.

[8] *I. G. Pospelov, M. A. Khokhlov,* Dimensionality control method for economy dynamics models. Matematicheskoe modelirovanie, vol.18, №. 10., 2006, pp. 113–122.

[9] *M. A. Khokhlov, I. G. Pospelov, L. Y. Pospelova,* Technology of development and implementation of realistic (country–specific) models of intertemporal equilibrium. International Journal of Computational Economics and Econometrics, vol. 4, №. 1-2, 2014, pp. 234–253.

[10] *M. Yu. Andreev, V. P. Vrzheshch, N. P. Pilnik, I. G. Pospelov, M. A. Khokhlov, A. A. Zhukova, S. A. Radionov,* Model of intertemporal equilibrium of the Russian economy based on disaggregation of the macroeconomic balance. Proceedings of the I.G. Petrovsky Seminar, vol. 29, 2013, pp. 43–145 (in Russian).

[11] *F. Smets, R. Wouters*, An estimated dynamic stochastic general equilibrium model of the euro area. Journal of the European economic association, vol. 1, №. 5, 2003, pp. 1123–1175.

[12] B. B. Iusup-Akhunov, *EcoMod* [Electronic repository]. URL: https://github.com/ChainedGenius/EcoMod.

Algorithm EG as a Tool for Finding Laurent Solutions of Linear Differential Systems with Truncated Series Coefficients

D.E. Khmelnov, A.A. Ryabenko

Federal Research Center "Computer Science and Control" of RAS, Russia

e-mail: dennis_khmelnov@mail.ru, anna.ryabenko@gmail.com

Abstract

Laurent solutions of systems of linear ordinary differential equations with the truncated power series coefficients are considered. The Laurent series in the solutions are also truncated. We use induced recurrent systems for constructing the solutions and have previously proposed an algorithm for the case when the induced system has a non-singular leading matrix. The algorithm finds the maximum possible number of terms of the series in the solutions that are invariant with respect to any prolongation of the original system. Below we present advances in extending our algorithm to the case when the leading matrix is singular using algorithm EG as an auxiliary tool.

Keywords: differential systems, truncated series coefficients, truncated Laurent series solutions

1. Starting Point

We consider systems of the form

$$A_r(x)\theta^r y(x) + A_{r-1}(x)\theta^{r-1} y(x) + \cdots + A_0(x)y(x) = 0, \tag{1}$$

where $y(x) = (y_1(x), y_2(x), \ldots, y_m(x))^T$ is the vector of unknowns, $A_r(x), \ldots, A_0(x)$ are $m \times m$-matrices of coefficients with entries in the form of power series in x over the field of algebraic numbers, $\theta = x\frac{d}{dx}$.

A solution $y(x) = (y_1(x), \ldots, y_m(x))^T$ of the differential system (1) the components of which are formal Laurent series is referred to as a Laurent solution:

$$y(x) = \sum_{n=v}^{\infty} u(n)x^n, \tag{2}$$

where $v \in \mathbb{Z}$ is a *valuation* of the series, $u(n) = (u_1(n), \ldots, u_m(n))^T$ are vectors of coefficients of Laurent series for $n \in \mathbb{Z}$.

For a full-rank system the coefficients of which are series specified algorithmically (i.e. an algorithm is given which computes the coefficient of any term x^s of any series), the algorithm from [2] finds all its truncated Laurent solutions with any given truncation degree. The algorithm is based on the construction of an *induced recurrent system* $R(u) = 0$ which is satisfied by the sequence of vectors $u(n)$ from (2). The induced recurrent system is constructed with the transformation

$$x \to E^{-1}, \;\; \theta \to n, \tag{3}$$

applied to the original differential system (1). E^{-1} denotes the shift operator: $E^{-1}u(n) = u(n-1)$. Thus $R = B_0(n) + B_{-1}(n)E^{-1} + B_{-2}(n)E^{-2} + \ldots$ and the induced system is written as

$$B_0(n)u(n) + B_{-1}(n)u(n-1) + \ldots = 0, \tag{4}$$

where $u(n) = (u_1(n), \ldots, u_m(n))^T$ is the column vector of unknown sequences such that $u_i(n) = 0$ for all negative n with large enough value of $|n|$, $i = 1, \ldots, m$; $B_0(n), B_{-1}(n), \ldots$ are matrices of polynomials in n; $B_0(n)$ is the *leading matrix* of system (4). If $B_0(n)$ is non-singular, then we can consider the equation $\det B_0(n) = 0$ as an indicial equation of the original differential system: the set of integer roots of this algebraic equation includes the set of all possible valuations of the Laurent solutions of system (1). This makes it possible, in particular, to find the lower bound for the valuations of all Laurent solutions of the system. If $\det B_0(n) = 0$ has no integer roots, then the system has no Laurent solutions. If $B_0(n)$ is singular, then algorithm $\mathrm{EG}_\sigma^\infty$ (it is a version of original EG algorithm from [1] introduced in [2] for infinite recurrent systems) initially applied to transform the induced recurrent system to the embracing recurrent system of the same form with a non-singular leading matrix, and it constructs any given number of its initial terms. The embracing recurrent system supplemented with a set of linear constraints, which are also constructed by $\mathrm{EG}_\sigma^\infty$ algorithm, has the same set of solutions as the system (4).

In our current research we are focused on the case when the series in the system coefficients are represented in a truncated form, with the truncation degree being different for different coefficients. We refer to such systems as *truncated systems*. Each truncated series is represented as

$$a(x) + O(x^{t+1}), \tag{5}$$

where $a(x)$ is a polynomial, the integer $t \geq \deg a(x)$ is a *truncation degree*.

The *prolongation* of a truncated series is a series (possibly, also truncated) the initial terms of which coincide with known initial terms of the original truncated series. In turn, the prolongation of a truncated equation is an equation the coefficients of which are prolongations of coefficients of the original equation, and the prolongation of a system of equations is a system the equations of which are prolongations of the equations of the original system.

We are interested in finding the maximum possible number of terms of truncated Laurent solutions of a given truncated system that are invariant with respect to any prolongations of the truncated coefficients of the given system. Solutions with arbitrary truncation degree cannot be calculated for a truncated system. This statement was proved in [3] for a particular case, namely, for a scalar equation ($m = 1$). In [3] we also proposed an algorithm which finds such truncated Laurent solutions for scalar equations. We utilized induced recurrent systems and *literals* as a foundation for finding the solutions. Literals are symbols used to represent unspecified coefficients of the truncated series involved in the system. For a series of form (5), we say that the coefficients of terms x^s,$s > t_{kij}$ are *unspecified*. When constructing solutions of the truncated system, these coefficients are represented by symbols, i.e., by literals. Our algorithm for truncated systems is a modification of the algorithms for the systems with the algorithmically specified coefficients. The key idea is to represent the truncated series (5) algorithmically: the algorithm returns the known coefficient of the series if $s \leq t$, and it returns literals if $s > t$.

In [6] we applied the approach from [3] to the systems with $m > 1$ and proposed an algorithm for constructing Laurent solutions of the system for the case when the determinant of the leading matrix of the induced system is not zero (i.e. the leading matrix is non-singular) and does not contain literals.

2. Advances and Further Plans

Our advances in the research of finding Laurent solutions of the systems in hand are related to the use of the algorithm $\mathrm{EG}_\sigma^\infty$ to extend the applicability of the algorithm from [6]

to the systems whose induced recurrent systems have singular leading matrix. We continue the adaptation of the algorithm for the systems with the algorithmically specified coefficients by representing the truncated series algorithmically with the help of literals.

The $\mathrm{EG}_\sigma^\infty$ algorithm consists in the successive repetition of reduction and shift steps, which continues until the rows of the leading matrix remain linear dependent. On the reduction step, coefficients of the dependence are found; then, the equation corresponding to one of the dependent rows is replaced with the linear combination of the other equations, hence the row of the leading matrix is set zero. On the shift step, the shift operator E is applied to the new equation. The termination of the algorithm is guaranteed with using a simple rule when selecting equations to be replaced. The reduction steps also lead to a finite set of linear constraints, each of which involves a finite number of elements of a sequential solution and is a linear combination of these elements with constant coefficients. The linear constraints correspond to the integer roots of a polynomial which is the coefficient of the row of the replaced equation in the linear combination (we further refer to the polynomials as the *constraint polynomials*).

The main obstacle is that literals may appear in intermediate calculations. If there are no literals both in the determinant of the leading matrix and in the constraint polynomials after $\mathrm{EG}_\sigma^\infty$ execution then the further calculations with the resulting induced recurrence and the linear constraints in the way as in algorithm from [6] gain desired truncated Laurent solutions which form the extended version of our algorithm in the case. It is preliminary implemented in Maple ([7]) as an updated version of procedure LaurentSolution of package TruncatedSeries [4, 5] (for more information about the package, please visit http://www.ccas.ru/ca/TruncatedSeries).

Let's consider the following system

$$\begin{pmatrix} 3x+O(x^2) & 7x^2+O(x^4) \\ O(x^2) & 17x^2+O(x^4) \end{pmatrix}\theta^2 y(x) + \begin{pmatrix} -1+2x+O(x^2) & x+5x^2+O(x^4) \\ O(x^2) & 11x^2+O(x^4) \end{pmatrix}\theta y(x) + \\ + \begin{pmatrix} O(1) & x-3x^2+O(x^4) \\ 1+O(x^2) & -6x^2+O(x^4) \end{pmatrix} y(x) = 0. \quad (6)$$

The leading matrix of its induced recurrent system is singular:

$$\begin{pmatrix} U_{[1,1],[0,0]}-n & 0 \\ 1 & 0 \end{pmatrix}.$$

$U_{[i,j][k,s]}$ is the representation of the literal denoting an unspecified coefficient of x^s in (i,j)-th element of matrix coefficient $A_k(x)$ of the system (1). After execution of $\mathrm{EG}_\sigma^\infty$ the transformed leading matrix becomes non-singular:

$$\begin{pmatrix} 3n^2+2n+U_{[1,1],[0,1]} & n+1 \\ 1 & 0 \end{pmatrix},$$

and its determinant $(-n-1)$ contains no literals. No linear constraints constructed by $\mathrm{EG}_\sigma^\infty$, so there is no constraint polynomials with literals as well. It is the case when our new algorithm is applicable and it computes the truncated Laurent solution (c_1 is an arbitrary constant):

$$\begin{pmatrix} 6x^2c_1+O(x^3) \\ \frac{c_1}{x}+c_1+O(x) \end{pmatrix}.$$

Let's consider another system:

$$\begin{pmatrix} O(x^5) & -1+O(x^5) \\ 1+O(x^5) & O(x^5) \end{pmatrix}\theta y(x) + \begin{pmatrix} O(x^5) & O(1) \\ 2+O(x^5) & O(x^5) \end{pmatrix} y(x). \quad (7)$$

The leading matrix of its induced recurrent system is already non-singular:

$$\begin{pmatrix} 0 & U_{[1,2],[0,0]} - n \\ 2+n & 0 \end{pmatrix}.$$

However there is a literal in its determinant $(n - U_{[1,2],[0,0]})(2 + n)$. It is seen that the determinant has the roots -1 and $U_{[1,2],[0,0]}$. It means that the set of integer roots of the determinant may be different for different integer values of the literal $U_{[1,2],[0,0]}$. It is easy to check that there is no desired Laurent solutions of the system. It may be done by constructing various prolongations of the original system with substituting various integer values of the literal $U_{[1,2],[0,0]}$ and finding Laurent solutions of the prolongations with our algorithm from [6] (the algorithm is applicable to the prolongations since for each of them the leading matrix of the induced recurrence is still non-singular and there are no literals in its determinant already). For example, for $U_{[1,2],[0,0]} = 5$ the solution of the prolongation is

$$\begin{pmatrix} O(x^{10}) \\ c_1 x^5 + O(x^6) \end{pmatrix}$$

and for $U_{[1,2],[0,0]} = 6$ the solution of the prolongation is

$$\begin{pmatrix} O(x^{11}) \\ c_1 x^6 + O(x^7) \end{pmatrix}.$$

Since the solutions of the prolongations has no coinciding initial terms of the series, there is no desired Laurent solution of the original system.

Let $p(n)$ be the determinant of the leading matrix (either of the induced recurrent system if its leading matrix is non-singular, or of the embraced recurrent system after the application of $\mathrm{EG}^{\infty}_{\sigma}$ otherwise). If $p(n)$ contains literals then it may be represented in the general case as $p(n) = a(u_1, \ldots, u_s)(n - r_1) \ldots (n - r_k)(b_q(u_1, \ldots, u_s)n^q + \cdots + b_1(u_1, \ldots, u_s)n + b_0(u_1, \ldots, u_s))$, where $u_1, \ldots, u_s$ are literals involved in $p(n)$, $a(u_1, \ldots, u_s)$, $b_0(u_1, \ldots, u_s)$, $b_1(u_1, \ldots, u_s), \cdots, b_q(u_1, \ldots, u_s)$ are polynomials in the literals, $r_1 \ldots r_k$ are integer roots of $p(n)$ independent from the literals. If $a(u_1, \ldots, u_s)$ does contain literals (i.e. it is not a number) then there are such values of the literals $u_1, \ldots, u_s$ that $a(u_1, \ldots, u_s) = 0$ and, hence $p(n) = 0$, i.e. the leading matrix is singular for the values of the literals. If $a(u_1, \ldots, u_s)$ is a number, then the solution of the algebraic equation $(b_q(u_1, \ldots, u_s)r_0^q + \cdots + b_1(u_1, \ldots, u_s)r_0 + b_0(u_1, \ldots, u_s)) = 0$ in respect to $u_1, \ldots, u_s$ allows getting such values of the literals that $p(n)$ has any desired root r_0 in addition to the roots $r_1 \ldots r_k$. It means that in all cases the set of integer roots of $p(n)$ is not invariant in respect to the prolongations of the differential system in hand. The same reasoning is true for the constraint polynomials as well.

$\mathrm{EG}^{\infty}_{\sigma}$ may have variability in its execution. In spite of the fact that the special rule should be used for choosing the equation to be replaced to guarantee the termination of the computation, it is still possible to have more than one option to choose. It leads to the fact that for the same recurrent system $\mathrm{EG}^{\infty}_{\sigma}$ may result in different embraced systems. For example, for the system (6) $\mathrm{EG}^{\infty}_{\sigma}$ may be executed in another way that leads to the embraced recurrent system with another leading matrix:

$$\begin{pmatrix} U_{[1,1],[0,0]} - n & 0 \\ 3n^2 + 2n + U_{[1,1],[0,1]} & n+1 \end{pmatrix}$$

with the determinant $\left(U_{[1,1],[0,0]} - n\right)(n + 1)$ which contains the literal $U_{[1,1],[0,0]}$. It can be seen that the determinant is very similar to the determinant of the leading matrix of the

induced recurrence for the system (7) that has no desired truncated Laurent solution. In addition, the second variant of $\mathrm{EG}_\sigma^\infty$ execution gives the constraint polynomial $-U[[1,1],[0,0]]+n$ which also contains the literal. Still, we know from the first variant of $\mathrm{EG}_\sigma^\infty$ execution, that the system has desired truncated Laurent solution. It gives us the counterexample to the conjecture that there is no desired Laurent solution as soon as the determinant of the leading matrix and/or the constraint polynomials contain literals.

We experiment with the modification of our algorithm for the case of determinants and constraint polynomials containing literals, which takes into account only invariant integer roots of the determinant and constraint polynomial (i.e. those roots which are independent of literals). The experiments show that the modification of the algorithm gives correct answers for the system (6) for both variants of $\mathrm{EG}_\sigma^\infty$ execution and for the system (7), as well as for more other systems. Our further plans is either to prove that the approach is always correct or to identify the limitations of its applicability.

References

[1] *Abramov S.A.* EG-eliminations. J. Difference Equations Appl. 1999. Vol. 5. P. 393–433.

[2] *Abramov S.A., Barkatou M.A., Khmelnov D.E.* On full rank differential systems with power series coefficients. J. Symbolic Comput. 2015. Vol. 68. P. 120–137.

[3] *Abramov S.A., Ryabenko A.A., Khmelnov D.E.* Linear ordinary differential equations and truncated series. Comput. Math. and Math. Phys. 2019. Vol. 59, N. 10. P. 1649–1659.

[4] *Abramov S.A., Ryabenko A.A., Khmelnov D.E.* Procedures for constructing truncated solutions of linear differential equations with infinite and truncated power series in the role of coefficients. Program. Comput. Software. 2021. Vol. 47. P. 144–152.

[5] *Abramov S.A., Khmelnov D.E., Ryabenko A.A.,* The TruncatedSeries package for solving linear ordinary differential equations having truncated series coefficients. Maple in Mathematics Education and Research, Springer Nature Switzerland. 2021. P. 19–33.

[6] *Abramov S.A., Ryabenko A.A., Khmelnov D.E.* Searching for Laurent solutions of systems of linear differential equations with truncated power series in the role of coefficients. Program. Comput. Software. 2023 (to be published).

[7] *Maplesoft* Maple online help. http://www.maplesoft.com/support/help/

On the States of N-Level Quantum System With Positive Wigner Function

A. Khvedelidze[1,2,3], A. Torosyan[3]

[1]*Institute of Quantum Physics and Engineering Technologies, Georgian Technical University, Tbilisi, Georgia*

[3]*Andrea Razmadze Mathematical Institute of I. Javakhishvili Tbilisi State University, Tbilisi, Georgia*

[3]*Meshcheryakov Laboratory of Information Technologies, Joint Institute for Nuclear Research, Dubna, Russia*

e-mail: akhved@jinr.ru, astghik@jinr.ru

Abstract

In the present report, within the phase-space formulation of quantum theory of N-level systems, we discuss the existence of "classical states" which are defined as those states whose Wigner function is positive semi-definite. An explicit description of a set of classical states is given using the associated convex bodies inside the simplex of density matrices' eigenvalues. It is demonstrated how these results allow one to calculate three measures of classicality constructed out of the quasiprobability distributions: the nonclassicality distance, Kenfack-Życzkowski indicator, and the global indicator.

Keywords: quantum information theory, quantum systems, Wigner quasiprobability distribution, classicality indicator.

1. Motivation

An N-level quantum system is considered as an analogue of a classical statistical system with the probability distributions between N mutually exclusive events represented by the probability simplex. But the analogue is not absolute, since in quantum cases, following strictly this paradigm and attempting to introduce the concept of probability distribution on a phase space, we face various incompatibilities with the principles of canonical probability theory (see e.g., discussions in [1] and more recent review [2]).

The present report is based on our recent studies [3]–[9] and is focused on considering such a peculiarity of the quantum statistical description. Namely, we emphasize the significance of the existence of states that are characterized by Wigner distributions whose lower bound is negative. After a detailed algebraic description of those states, following a commonly accepted opinion that the negativity of a probability distribution is an essential attribute of the "quantumness" of states, we outline some technical issues of computing the corresponding "classicality-quantumness" characteristics.

2. Wigner function of states and their classicality

One of the most studied analogues of classical probability distribution for $N-$level quantum systems is the Wigner function $W_\varrho^{(\boldsymbol{\nu})}(\Omega_N)$ constructed via the dual pairing [3],

$$W_\varrho^{(\boldsymbol{\nu})}(\Omega_N) = \mathrm{tr}\left[\varrho\,\Delta(\Omega_N\,|\,\boldsymbol{\nu})\right]\,,$$

of a state ϱ from the state space $\mathfrak{P}_N$,

$$\mathfrak{P}_N = \{\varrho \in M_N(\mathbb{C})\,|\,\varrho = \varrho^\dagger\,,\ \varrho \geq 0\,,\ \mathrm{Tr}\,\varrho = 1\}\,,$$

and matrix $\Delta(\Omega_N \,|\, \boldsymbol{\nu})$, termed as the Stratonovich-Weyl (SW) kernel. There is a whole family of SW kernels forming the dual space $\mathfrak{P}_N^*$ according to the following equations:

$$\mathfrak{P}_N^* = \{X \in M_N(\mathbb{C}) \mid X = X^\dagger\,, \quad \mathrm{tr}\,(X) = 1\,, \quad \mathrm{tr}\left(X^2\right) = N\}\,. \tag{1}$$

The SW kernel provides a mapping between phase space Ω_N and state space $\mathfrak{P}_N$ and, according to the master equations (1), can be categorised by the set of moduli parameters, $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_s)\,,\ s \leq N-2\,$. A separate analysis of the moduli space of the Wigner function is given in [4].

In order to effectively compute the "classicality-quantumness" measures, an explicit description of states attributed to "classical" ones is necessary. The "classical states" form the subset $\mathfrak{P}_{\mathrm{Cl}}^N \subset \mathfrak{P}_N$ of states whose Wigner function is non-negative everywhere over the phase space:

$$\mathfrak{P}_N^{\mathrm{Cl}} = \{\, \varrho \in \mathfrak{P}_N \;\mid W_\varrho^{(\boldsymbol{\nu})}(z) \geq 0\,, \quad \forall z \in \Omega_N \,\}\,.$$

Furthermore, one can consider a more refined classification of classical states based on the decomposition of the state space $\mathfrak{P}_N$ on strata which are characterised by the same isotropy group, $H_\alpha \subset SU(N)$, i.e., belong to a class with the same "orbit type" $[H_\alpha]$:

$$\mathfrak{P}_N = \bigcup_{\text{orbit types}} \mathfrak{P}_{[H_\alpha]}\,, \tag{2}$$

with each component of (2) consisting of density matrices with a fixed algebraic degeneracy,

$$\mathfrak{P}_{[H_\alpha]} = \bigcup_{\omega \in S_n} \mathfrak{P}_{k_{\omega(1)}, k_{\omega(2)}, \ldots, k_{\omega(n}}\,. \tag{3}$$

In (3) S_n is a symmetric group acting on a given partition of N into n natural numbers $k_1, k_2, \ldots, k_n$. Algebraically, $\mathfrak{P}_{k_1,k_2,\ldots,k_n}$, being a set of states with a fixed degeneracy and maximal rank, is defined via the characteristic polynomial of a density matrix:

$$\mathfrak{P}_{k_1,k_2,\ldots,k_n} = \{\, \varrho \in \mathfrak{P}_N\,, k_i \in \mathbb{Z}_+ \mid \det(\varrho - \lambda) = \prod_{i=1}^{n} (r_i - \lambda)^{k_i}\,, \quad \sum_{i=1}^{n} k_i = N \,\}\,.$$

Geometrically, the set $\mathfrak{P}_{k_1,k_2,\ldots,k_n}$ with $k_1 = k_2 = \cdots = k_N = 1$ represents the interior of an $(N-1)-$dimensional simplex C_{N-1} of eigenvalues:

$$C_{N-1} := \{\, \boldsymbol{r} \in \mathbb{R}^N \,\Big|\, \sum_{i=1}^{N} r_i = 1\,, \quad 1 \geq r_1 \geq r_2 \geq \cdots \geq r_{N-1} \geq r_N \geq 0 \,\}\,,$$

while for all other admissible tuples $\boldsymbol{k} = (k_1, k_2, \ldots, k_n)$ each $\mathfrak{P}_{k_1,k_2,\ldots,k_n}$ represents the union of the faces and edges of the $(N-1)-$simplex parameterized by the corresponding degenerate barycentric coordinates.

Based on the observation above, one can similarly define the "classical states on a fixed stratum" $\mathfrak{P}_{H_\alpha}$:

$$\mathfrak{P}_{H_\alpha}^{\mathrm{Cl}} = \mathfrak{P}_N^{\mathrm{Cl}} \cap \mathfrak{P}_{H_\alpha}\,.$$

In the report the set of classical states $\mathfrak{P}_{\mathrm{Cl}}$ as well as the classical states on each stratum $\mathfrak{P}_{H_\alpha}^{\mathrm{Cl}}$, will be described using the associated convex bodies inside the simplex of density matrices eigenvalues C_{N-1}. Particularly, it will be demonstrated that the image of $\mathfrak{P}_{[H_\alpha]}^{\mathrm{Cl}}$ under the canonical quotient map p onto the orbit space,

$$\mathcal{C}_{N-1}^*(H_\alpha) = \{\, p(x) \;\mid x \in \mathfrak{P}_{[H_\alpha]}^{\mathrm{Cl}} \,\}\,,$$

admits identification with the dual cone:

$$\mathcal{C}_{N-1}^*(H_\alpha) = \left\{\, \boldsymbol{\pi} \in \mathbf{spec}\,(\Delta(\Omega_N \,|\, \boldsymbol{\nu})) \;\mid\; (\boldsymbol{r}^{\downarrow}, \boldsymbol{\pi}^{\uparrow}) \geq 0, \quad \forall\, \boldsymbol{r} \in \mathcal{C}_{N-1}(H_\alpha) \,\right\}\,,$$

where $(\boldsymbol{r}^{\downarrow}, \boldsymbol{\pi}^{\uparrow}) = r_1\pi_N + r_2\pi_{N-1} + \cdots + r_N\pi_1\,.$

3. Computing classicality-quantumness measures

Bearing in mind the above description of classical states, we study the following measures of classicality:

(I) the *nonclassicality distance* – a measure of nonclassicality defined as an infinum of the distance D of a state ϱ from the reference set of "classical states" $\mathfrak{P}_{\mathrm{Cl}}$:

$$d_N(\varrho) = \inf_{x \in \mathfrak{P}_N^{\mathrm{Cl}}} D(\varrho, x)\,; \tag{4}$$

(II) *Kenfack-Życzkowski indicator* – a measure of nonclassicality of quantum states based on the volume of a phase space Ω_N region where the Wigner function is negative [5]:

$$\delta_N(\varrho) = \int_{\Omega_N} \mathrm{d}\Omega_N \left| W_\varrho^{(\nu)}(\Omega_N) \right| - 1\,;$$

(III) the *global indicator of classicality* – geometric probability of classicality defined as the relative volume of the classical subspace $\mathfrak{P}_N^{\mathrm{Cl}}$ with respect to the total volume of the state space:

$$\mathcal{Q}_N = \frac{\mathrm{Volume}\left(\mathfrak{P}_N^{\mathrm{Cl}}\right)}{\mathrm{Volume}\left(\mathfrak{P}_N\right)}\,. \tag{5}$$

Note that similarly to measures (I)-(III), defined in (4)-(5), corresponding counterparts for the states located on the unitary strata of given orbit type $[H_\alpha]$ can be introduced as well. For details we refer to our recent publications [6, 7, 8, 9].

References

[1] *Hillery M., O'Connell R.F., Scully M.O. and Wigner E.P.*, Distribution functions in physics: Fundamentals, Physics Reports, 106 (3), 121–167 (1984).

[2] *Russell P. Rundle and Everitt M.J.*, Overview of the phase space formulation of quantum mechanics with application to quantum technologies, Adv. Quantum Technol., 2100016 (2021).

[3] *Abgaryan V. and Khvedelidze A.*, On families of Wigner functions for N-level quantum systems, Symmetry, 13 (6), 1013 (2021).

[4] *Abgaryan V., Khvedelidze A. and Torosyan A.*, On the moduli space of Wigner quasiprobability distributions for N-dimensional quantum systems, J. Math. Sci. 240, 617–633 (2019).

[5] *Abgaryan V., Khvedelidze A. and Torosyan A.*, Kenfack-Życzkowski indicator of nonclassicality for two non-equivalent representations of Wigner function of qutrit, Phys. Let. A, 412, 127591 (2021).

[6] *Abgaryan V., Khvedelidze A. and Torosyan A.*, The global indicator of classicality of an arbitrary N-Level quantum system, J. Math. Sci. 251, 301–314 (2020).

[7] *Abbasli N., Abgaryan V., Bures M., Khvedelidze A., Rogojin I. and Torosyan A.*, On measures of classicality/quantumness in quasiprobability representations of finite-dimensional quantum systems, Physics of Particles and Nuclei, 51, 443–447 (2020).

[8] *Abgaryan V., Khvedelidze A. and Rogojin I.*, On overall measure of non-classicality of N-level quantum system and its universality in the large N limit, Lecture Notes in Computer Science, 12563, 244–255 (2021).

[9] *Khvedelidze A. and Torosyan A.*, Comparing classicality of qutrits from Hilbert–Schmidt, Bures and Bogoliubov–Kubo–Mori ensembles, Zap. Nauchn. Sem. POMI, 517, 250–267 (2022).

A Constructive Approach to Problems of Quantum Mechanics

V.V. Kornyak

Laboratory of Information Technologies
Joint Institute for Nuclear Research, Dubna, Russia

e-mail: vkornyak@gmail.com

Abstract

We consider a constructive modification of quantum mechanics based on permutation representations of finite groups in Hilbert spaces over cyclotomic fields, and its connection with the Weyl–Schwinger "finite quantum mechanics". Constructive quantum mechanics requires mathematical tools that differ significantly from those used in traditional continuous theory: number theory, finite fields, complex Hadamard matrices, finite geometries, etc. A natural approach to the various problems that arise in the field are computer calculations based on the methods of computer algebra and computational group theory.

Keywords: permutation quantum mechanics, Pontryagin duality, mutually unbiased bases, quantum informatics

The standard formulation of quantum mechanics is essentially non-constructive, since it is based on continuous unitary groups and number fields $\mathbb{R}$ and $\mathbb{C}$. This descriptive flaw does not allow one to study some fine details of the structure of quantum systems and sometimes leads to artifacts.

In [1, 2, 3], we considered a modification of quantum mechanics based on permutation representations of finite groups in Hilbert spaces over cyclotomic fields. This *permutation quantum mechanics* (PQM) "can accurately reproduce all of the results of conventional quantum mechanics" [4] in the permutation invariant *standard subspace* of the Hilbert space. Unitary evolution in PQM is generated by a permutation of *ontic* elements, which form a basis of the Hilbert space. By decomposing the permutation into a product of disjoint cycles, we can split the Hilbert space into a direct sum of subspaces, in each of which the evolution generated by a cyclic permutation occurs independently. Thus, in an N-dimensional Hilbert space, it suffices to consider the evolutions generated by cycles of length N. Such a cycle generates the group $\mathbb{Z}_N$. Since any projective representation of a cyclic group is trivial, to describe quantum mechanical phenomena it is necessary to consider the product $\mathbb{Z}_N \times \tilde{\mathbb{Z}}_N$, where $\tilde{\mathbb{Z}}_N \, (\simeq \mathbb{Z}_N)$ is the Pontryagin dual group to $\mathbb{Z}_N$. Note that we have only changed the description slightly, without introducing any additional external information: if X and Z are matrices representing generators of $\mathbb{Z}_N$ and $\tilde{\mathbb{Z}}_N$, respectively, then Z is simply the diagonal form of X, obtained by the Fourier transform.

In fact, we have come to the Weyl–Schwinger version of quantum mechanics, which is sometimes called *finite quantum mechanics* (FQM). FQM arose as a result of Weyl's correction of Heisenberg's canonical commutation relation, which cannot be realized in finite-dimensional Hilbert spaces. Weyl's canonical commutation relation has the form

$$XZ = \omega ZX, \ \omega = \mathrm{e}^{2\pi \mathbf{i}/N},$$

where X and Z are the matrices mentioned above. Weyl proved that the X and Z are generators of a projective representation of $\mathbb{Z}_N \times \mathbb{Z}_N$ in the N-dimensional Hilbert space. The orthonormal bases associated with the matrices X and Z are *mutually unbiased bases*, a concept introduced by Schwinger.

FQM, constructive by its nature, requires mathematical tools that differ significantly from those used in traditional continuous theory: number theory, Galois field theory, complex Hadamard matrices, finite geometries, etc.

At the same time, in FQM, it is possible to pose and solve problems that are important for fundamental quantum theory and quantum informatics, but which are difficult or even impossible to formulate within the framework of standard quantum mechanics. Let us give examples of problems in which the structure of the decomposition of the dimension of the Hilbert space into prime numbers is essential, which does not make sense in continuous quantum mechanics:

- decomposition of a quantum system into smaller subsystems;
- calculation of sets of mutually unbiased bases (sets of orthonormal bases in Hilbert space, measurements in which give maximum information about the quantum state);
- construction of symmetric information-complete positive operator-valued measures (SIC-POVM, a symmetric set of vectors in a Hilbert space, important for quantum measurement theory and related to Hilbert's 12th problem).

Modern problems of quantum physics and quantum informatics require a detailed analysis of the "fine structure" of quantum systems, which cannot be carried out using traditional approximate methods of quantum mechanics. However, exact methods are complex and often involve open (unsolved) mathematical problems. In these circumstances, a natural approach is to use computer calculations based on the methods of computer algebra and computational group theory.

References

[1] *Kornyak V.V.* Quantum models based on finite groups. IOP Conf. Series: Journal of Physics: Conf. Series **965**, 012023, 2018. arXiv:1803.00408 [physics.gen-ph]

[2] *Kornyak V.V.* Modeling Quantum Behavior in the Framework of Permutation Groups. EPJ Web of Conferences **173**, 01007, 2018. arXiv:1709.01831 [quant-ph]

[3] *Kornyak V.V.* Mathematical Modeling of Finite Quantum Systems. In: Adam G. *et al* (eds) MMCP 2011. LNCS, **7125**. Springer, 2012. arXiv:1107.5675 [quant-ph]

[4] *Banks T.* Finite Deformations of Quantum Mechanics. arXiv:2001.07662 [hep-th], 20 p., 2020.

Nonlinear Effects of Motion Near the Equilibrium Manifold of Nonholonomic Systems

A.S. Kuleshov, N.M. Vidov

Department of Mechanics and Mathematics, Lomonosov Moscow State University, Russia

e-mail: kuleshov@mech.math.msu.su, nikitavidov98@gmail.com

Abstract

A general analysis of nonlinear oscillations of conservative nonholonomic systems, possessing the equilibrium manifold is presented. The procedure of normalization of equations of motion near the equilibrium manifold is discussed. As an example of the general theory the problem of motion of a heavy rigid thin rod on a perfectly rough right circular cylinder is considered.

Keywords: nonholonomic system, equilibrium manifold, transgression effect

1. Normalization of the system of differential equations in a neighborhood of the equilibrium manifold

Let on the phase space Φ there is a vector field $\boldsymbol{Z}$ that vanishes on the manifold E. Suppose that in suitable coordinates X_k, ξ_k the manifold E is locally specified by the equations $E = \{\xi_k = 0\}$. Thus, the coordinates X_k changes along E, and ξ_k changes transversally. For any function f depending on X_k, ξ_k we have

$$f(\boldsymbol{X}, \boldsymbol{\xi}) = \sum_{M\geq 0} f^{(M)}(\boldsymbol{X}, \boldsymbol{\xi}) = \sum_{M\geq 0} \sum_{|j|=M} f^{(j)}(\boldsymbol{X})\, \boldsymbol{\xi}^j,$$

$$\frac{df}{dt} = \sum_{M\geq 0} f^{[M]}(\boldsymbol{X}, \boldsymbol{\xi}) = \sum_{M\geq 0} \sum_{|j|=M} f^{[j]}(\boldsymbol{X})\, \boldsymbol{\xi}^j,$$

where $\boldsymbol{\xi}^j = \xi_1^{j_1} \cdot \xi_2^{j_2} \cdot \ldots$, $|j| = j_1 + j_2 + \ldots$

Here $\frac{df}{dt} = \boldsymbol{Z}(f)$. In particular, in variables X_k, ξ_k the vector field $\boldsymbol{Z}$ is represented as follows:

$$\dot{\xi}_k = \sum_{|j|\geq 1} \xi_k^{[j]}(\boldsymbol{X})\, \xi^j, \quad \dot{X}_k = \sum_{|j|\geq 1} X_k^{[j]}(\boldsymbol{X})\, \xi^j.$$

Because of $\boldsymbol{Z} = 0$ on E, therefore $\xi_k^{[0]} = 0$, $X_k^{[0]} = 0$.

We consider a small (with respect to $\boldsymbol{\xi}$) ε-neighbourhood of the manifold E. In order to establish a proper analogy with perturbation theory, we introduce a small parameter ε, putting

$$\boldsymbol{\xi} = \varepsilon \boldsymbol{\zeta}.$$

Then

$$\dot{\zeta}_k = \sum_{M\geq 1} \varepsilon^{M-1} \sum_{|j|=M} \xi_k^{[j]}(\boldsymbol{X})\, \zeta^j, \quad \dot{X}_k = \sum_{M\geq 1} \varepsilon^M \sum_{|j|=M} X_k^{[j]}(\boldsymbol{X})\, \zeta^j.$$

When $\varepsilon = 0$ we obtain the first order approximation

$$\dot{X}_k = 0, \quad \dot{\zeta}_k = \sum_{\nu} \xi_k^{\nu}(\boldsymbol{X})\, \zeta_\nu.$$

We shall assume that the first order approximation system has diagonal form

$$\dot{X}_k = 0, \quad \dot{\xi}_k = \lambda_k(\boldsymbol{X})\, \xi_k.$$

To obtain the approximation of the N-the order, $N \geq 2$ we must retain terms with ε up to the power $N-1$ inclusive.

It is easy to understand the appearance of the N-th order approximation in the variables X_k, ξ_k. We assume that the considered system has a form:

$$\dot{X}_k = \sum_{|j|\geq 1} X_k^{[j]}(\boldsymbol{X})\,\xi^j, \quad \dot{\xi}_k = \lambda_k(\boldsymbol{X})\,\xi_k + \sum_{|j|\geq 2} \xi_k^{[j]}(\boldsymbol{X})\,\xi^j \tag{1}$$

and neglect in the right-hand sides all monomials starting with degree N in X_k and degree $N+1$ in ξ_k. Then we can use a reduction to the normal form procedure [1, 2, 3, 4], i.e. we can construct a change of variables after which the successive approximations take their simplest form.

The dependence of the coefficients in system (1) on X_k dives definite difficulties (which will be pointed out) compared with the normalization of ordinary quasilinear systems. We will start with the change of variables

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{Y}^{(N-1)}(\boldsymbol{X},\,\boldsymbol{\xi}), \quad \boldsymbol{\eta} = \boldsymbol{\xi} + \boldsymbol{\eta}^{(N)}(\boldsymbol{X},\,\boldsymbol{\xi}), \quad N \geq 2. \tag{2}$$

We directly quote the inverse expressions

$$\boldsymbol{X} = \boldsymbol{Y} - \boldsymbol{Y}^{(N-1)}(\boldsymbol{Y},\,\boldsymbol{\eta}) + \ldots, \quad \boldsymbol{\xi} = \boldsymbol{\eta} - \boldsymbol{\eta}^{(N)}(\boldsymbol{Y},\,\boldsymbol{\eta}) + \ldots. \tag{3}$$

Here new symbols have been substituted into the expressions $\boldsymbol{Y}^{(N-1)}$, $\boldsymbol{\eta}^{(N)}$ and the dots indicate high order terms (only these terms changes due to dependence of the polynomials (2) on X_k).

Now we differentiate expressions (2). According to (1) in the derivative of the monomial $F(\boldsymbol{X})\boldsymbol{\xi}^j$ of degree $M \geq 1$ there will be monomial $(\boldsymbol{\lambda}(\boldsymbol{X})\cdot\boldsymbol{j})\,F(\boldsymbol{X})\,\boldsymbol{\xi}^j$ of the same degree plus terms of higher degrees including terms from the differentiation of F with respect to $\boldsymbol{X}$. Hence in the variables $\boldsymbol{X}$ and $\boldsymbol{\xi}$ we have

$$\begin{aligned}
\dot{Y}_k &= X_k^{[1]} + \ldots + X_k^{[N-2]} + \sum_{|j|=N-1} \left(X_k^{[j]} + (\boldsymbol{\lambda}\cdot\boldsymbol{j})\,Y_k^{(j)}\right)\xi^j + \ldots \\
\dot{\eta}_k &= \lambda_k\xi_k + \xi_k^{[2]} + \ldots + \xi_k^{[N-1]} + \sum_{|j|=N} \left(\xi_k^{[j]} + (\boldsymbol{\lambda}\cdot\boldsymbol{j})\,\eta_k^{(j)}\right)\xi^j + \ldots.
\end{aligned} \tag{4}$$

On the right-hand sides of (4) we change to variables $\boldsymbol{Y}$ and $\boldsymbol{\eta}$ having substituted expressions (3). After this transformation every monomial $F(\boldsymbol{X})\,\boldsymbol{\xi}^j$ of degree M is represented as

$$F(\boldsymbol{X})\,\boldsymbol{\xi}^j = F\left(\boldsymbol{Y} - \boldsymbol{Y}^{(N-1)}(\boldsymbol{Y},\,\boldsymbol{\eta}) + \ldots\right)\left(\boldsymbol{\eta} - \boldsymbol{\eta}^{(N)}(\boldsymbol{Y},\,\boldsymbol{\eta}) + \ldots\right)^j. \tag{5}$$

The expansion (5) gives a term $F(\boldsymbol{Y})\,\boldsymbol{\eta}^j$ of degree M, and then terms of degree $M+N-1$ and, finally, terms of degree greater than N. The degree $N = M+N-1$ only for $M=1$, so that the coefficients of degree N for η_k changes only as a result of the transformation of the polynomial

$$\lambda_k(\boldsymbol{X})\,\xi_k = \lambda_k(\boldsymbol{Y})\,\eta_k - \sum_j \frac{\partial\lambda_k}{\partial Y_j} Y_j^{(N-1)}\eta_j - \lambda_k(\boldsymbol{Y})\eta^{(N)} + \ldots.$$

Consequently, in variables $\boldsymbol{Y}$ and $\boldsymbol{\eta}$ we obtain:

$$Y_k^{[j]} = X_k^{[j]} + (\boldsymbol{\lambda}\cdot\boldsymbol{j})\,Y_k^{(j)}, \quad |j| = N-1, \tag{6}$$

$$\eta_k^{[j]} = \xi_k^{[j]} - \sum_i \frac{\partial \lambda_k}{\partial Y_i} Y_i^{(j-e_k)} + (\boldsymbol{\lambda}\cdot\boldsymbol{j} - \boldsymbol{e}_k)\eta_k^{(j)}, \tag{7}$$

$$|j| = N, \quad j_k \neq 0, \quad \boldsymbol{e}_k = (0, \ldots, \underset{(k)}{1}, \ldots, 0),$$

$$\eta_k^{[j]} = \xi_k^{[j]} + (-\lambda_k + (\boldsymbol{\lambda}\cdot\boldsymbol{j}))\eta_k^{(j)}, \quad |j| = N, \quad j_k = 0.$$

If $(\boldsymbol{\lambda}\cdot\boldsymbol{j}) \neq 0$, $|j| = N-1$, then by a suitable choice of $Y_k^{(j)}$ we can eliminate $Y_k^{[j]}$, after which a choice of $\eta_k^{(j)}$, $|j| = N$, $j_k \neq 0$ eliminates the corresponding $\eta_k^{[j]}$. We can separately eliminate $\eta_k^{[j]}$, $|j| = N$, $j_k = 0$, if

$$\lambda_k \neq j_1\lambda_1 + \ldots + j_{k-1}\lambda_{k-1} + j_{k+1}\lambda_{k+1} + \ldots$$

When λ_k does not depend on $\boldsymbol{X}$, solution (6) does not affect (7), and the normalization procedure proceeds according to the usual scheme. However, in this case the dependence on $\boldsymbol{X}$ of other significantly complicates the calculation of the successive approximations. If elimination is impossible, the corresponding coefficients from (2) will be taken to be zero.

To implement the normalization procedure of the system of differential equations near the equilibrium manifold the special program was developed using the known complex of symbolic computations MAPLE. Below we discuss results of application of this program to the problem of motion of a heavy thin rod on a surface of a right circular inclined cylinder.

2. The problem of motion of a rod on an inclined cylinder

Let the rigid thin rod of the mass M moves without sliding on a rigid circular cylinder of radius R. We suppose that the rod touches the cylinder by the point P. We will define the position of the point P on the cylinder by the cylindrical coordinates φ and z. According to [5], let us introduce the moving coordinate system $Px_1x_2x_3$ with the unit vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ and $\boldsymbol{e}$, so that the radius-vector $\boldsymbol{PG}$ of the center of mass G of the rod equals $\boldsymbol{PG} = s\boldsymbol{e}_1$. The vector $\boldsymbol{e}$ is the normal vector to the cylinder at P. We denote by θ the angle of rotation of the rod about $\boldsymbol{e}$. We suppose that the rod moves on the cylinder under the action of gravity and the generatrix of cylinder has a constant angle $\frac{\pi}{2} - \alpha$ with the direction of gravity. Then equations of motion of the rod on a cylinder is written as follows:

$$\dot{s} = u, \quad \dot{\theta} = \omega, \quad \dot{z} = -u\sin\theta,$$

$$\dot{\varphi} = -\frac{u\cos\theta}{R}, \quad \dot{u} = -\frac{Msu^2}{J+Ms^2} + 3\omega u\tan\theta - \frac{MgRs\cos\varphi\cos\alpha}{(J+Ms^2)\cos^2\theta}, \tag{8}$$

$$\dot{\omega} = -\frac{Msu\omega}{J+Ms^2} - \frac{u^2\sin\theta\cos^3\theta}{R^2} - \frac{Mgs\sin\theta\sin\varphi\cos\alpha}{J+Ms^2} - \frac{Mgs\cos\theta\sin\alpha}{J+Ms^2}.$$

Here J is the moment of inertia of the rod with respect to the axis perpendicular to the rod and passing through its center of mass. Let us introduce the following coordinates:

$$\Phi = \varphi + \frac{s\cos\theta}{R}, \quad Q = z + s\sin\theta.$$

The equilibrium manifold is

$$E = \{s = 0,\, u = 0,\, \omega = 0\}.$$

Analysis of the normal form of equations (8) gives the following result. The process of the motion of the rod can be qualitatively represented as oscillations in s, φ, z with amplitude of order ε about an equilibrium position with coordinates Φ and Q combined with the slow rotation of the rod (in the first approximation the latter effect is not present). In a time of order $\frac{1}{\varepsilon}$ the angle θ changes by a finite amount, and the equilibrium position about which the oscillations occurs is displaced by an amount of order ε^2 (this is the so called transgression effect). The displacement occurs along the curve

$$Q = Q_0 - 2R\cot\alpha\,(\cos\Phi - \cos\Phi_0)\,.$$

Thus, using the normal form method we can investigate the behavior of the thin rigid rod, moving on an inclined cylinder.

References

[1] *Bruno A.D.* Local Methods in Nonlinear Differential Equations. Berlin, Heidelberg: Springer. (1989).

[2] *Bruno A.D.* Analytical form of differential equations I // Trans. Mosc. Math. Soc. 1971. V. 25. P. 119–262. (in Russian)

[3] *Bruno A.D.* Analytical form of differential equations II // Trans. Mosc. Math. Soc. 1972. V. 26. P. 199–239. (in Russian)

[4] *Edneral V.F.* Looking for Periodic Solutions of ODE Systems by the Normal Form Method. In: Wang D., Zheng Z. (eds) Differential Equations with Symbolic Computation. Trends in Mathematics. Birkhauser Basel. 2005. P. 173–200.

[5] *Kuleshov A.S., Ifraimov S.V.* Motion of the Rod on a Convex Surface // Vestnik St. Petersb. Univ. Ser. 1. Maths. Mechs. Astr. 2013. No. 2. P. 105–110. (in Russian)

Primitive Elements of Free Non-Associative Algebras Over Finite Fields

M.V. Maisuradze, A.A. Mikhalev

Faculty of Mechanics and Mathematics, Lomonosov Moscow State University, Russia

e-mail: maisuradzemv@my.msu.ru, aamikhalev@mail.ru

Abstract

The representation of elements of free non-associative algebras in the form of a set of multidimensional tables of coefficients is determined. The operation of finding partial derivatives of elements of free non-associative algebras in the same form is considered. Using this representation, a criterion for the primitiveness of elements of length two in terms of matrix ranks is obtained, as well as a primitivity test of elements of arbitrary length. With this test, the number of primitive elements with two generators was estimated.

Keywords: Schreier varieties of algebras, free non-associative algebras, primitive elements in free non-associative algebras, free differential calculus in free algebras.

1. Introduction

In 1947 A.G. Kurosh [2] proved that subalgebras of free non-associative algebras are free. Similar result for free Lie algebras was proved by A.I. Shirshov in 1953 [9].

Primitive element of a free algebra is an element of some set of free generators of this free algebra.

In 1994 A.A. Mikhalev and A.A. Zolotykh [8] using free differential calculus constructed algorithms to recognize primitive elements of free Lie algebras and superalgebras and to construct complements of primitive elements with respect to free generating sets. For free non-associative algebras these algorithms were constructed by A.A. Mikhalev, U.U. Umirbaev, and J.-T. Yu [6]. Modified versions of these algorithms with computer realization was suggested by A.A. Chepovskii. In the monographs [7, 5] many properties of primitive elements were considered. In 2021 M.V. Maisuradze [3] created software implementation of algorithms for free non-associative algebras and free differential calculus in the SageMath computer algebra system.

In this talk we consider primitive elements of free non-associative algebras over finite fields. In particular, we propose a new approach to the study of elements of free non-associative algebras and ways of counting primitive elements in these algebras over finite fields. In our study, we will use the technique of free differential calculus and the criterion of primitivity [6]:

The system $a_1, a_2, \ldots, a_r$ of elements of a free non-associative algebra A is primitive if and only if the matrix $(\partial(a_1), \ldots, \partial(a_r))$ is left invertible over $U(A)$. In particular, an element $a \in A$ is primitive if and only if there are elements $m_1, \ldots, m_n \in U(A)$ such that $\sum_{i=1}^{n} m_i \frac{\partial a}{\partial x_i} = 1$.

Here, $U(A)$ – *universal multiplicative enveloping algebra*, which is a free associative algebra with a set of free generators $S = \{r_w, l_w \mid w \in W\} \cup \{1\}$ – left and right multiplication operators on words from the free groupoid W on X.

The technique of free differential calculus uses a linear differentiation operator $\mathcal{D}$ satisfying the Leibniz rule: $\mathcal{D}(uv) = \mathcal{D}(u)v + u\mathcal{D}(v)$, $u, v \in A$ and commuting with field elements: $\mathcal{D}(au) = a\mathcal{D}(u)$, $a \in K, u \in A$. *Partial derivatives* $\frac{\partial u}{\partial x_i}$ of an element $u \in A$ are elements of $U(A)$ such that

$$\mathcal{D}(u) = \frac{\partial u}{\partial x_1}\mathcal{D}(x_1) + \ldots + \frac{\partial u}{\partial x_n}\mathcal{D}(x_n)$$

2. Linear subspaces

The elements of a free algebra over a field K can be represented as elements of an arithmetic vector space – tuples composed of coefficients of terms. Due to the Schreier property, one can consider a free algebra as their direct sum of its subalgebras. At the same time, various decompositions into direct terms can be used, depending on the current task.

To begin with, let's consider the decomposition by the length of the word and the placement of brackets.

Let A be a free algebra with n generators. We divide the basis of the vector space into groups according to the length of the word. In the non-associative case, an additional division into groups by the arrangement of brackets will also be required. In each group of words of length k there will be n^k elements that can be written as a k-dimensional table. It is also convenient to write the coefficients for these monomials in the form of a k-dimensional table.

It is also convenient to represent the unit and words of a free groupoid in the form of multidimensional tables. Denote: $\bar{x} \times \ldots \times \bar{x}$ is a k-dimensional table in which in place $(i_1, \ldots, i_k)$ there is an element $x_{i_1} \ldots x_{i_k}$. In the non-associative case, the arrangement of brackets in the product $\bar{x} \times \ldots \times \bar{x}$ coincides with the arrangement of brackets in the element $x_{i_1} \ldots x_{i_k}$. Considering multiplication between tables of coefficients and words, as well as addition between elements of one of the subspaces described above, element-wise, we get a more convenient representation of the elements of free algebras.

Another useful way to decompose a free algebra into a direct sum is obtained if one of the symbols for each of its possible variants is fixed in a word. Let's call them *layers*.

When differentiating monomials of length 1, we obtain the same vector of free terms of partial derivatives. Differentiating all monomials of length 2 with coefficients written as a matrix will give us the same matrix of coefficients for right derivatives and transposed for left ones, in which each layer (in the sense described above) corresponds to the coefficients of partial derivatives. In the general case, when differentiating a monomial of length k, one obtains k of various basic monomials of a universal multiplicative enveloping algebra. Thus, a set of partial derivatives of an element of a free non-associative algebra can be written in the form of multidimensional tables that coincide up to transpositions with such a representation of the element.

This allows us to formulate a criterion for the primitiveness of elements of length 2 in terms of linear algebra.

3. Criterion of primitiveness of an element of length 2 and a sign of primitiveness

Statement 1. *The element $h = a \cdot \bar{x} + B \cdot \bar{x} \otimes \bar{x}$ of free non-associative algebra is primitive if and only if the rank of the matrix $\left(a \mid B \mid B^T\right)$ is greater than the rank of the matrix $\left(B \mid B^T\right)$.*

Proof. We use the technique of free differential calculus and the criterion of primitiveness.

An element is primitive if and only if there are $m_1, m_2, \ldots, m_n \in U(A)$ such that $m_1\frac{\partial}{\partial x_1} + m_2\frac{\partial}{\partial x_2} + \ldots + m_n\frac{\partial}{\partial x_n} = 1$.

The matrix $\left(a \mid B \mid B^T\right)$ contains n rows, where n is the number of free generators. Each of the rows of the matrix corresponds to the representation of a partial derivative of one of the variables in the form of multidimensional tables.

The problem of determining primitiveness is solved by a reduction algorithm, the step of which is to eliminate the higher monomials from derivatives at the expense of other derivatives. Since all the higher monomials l_{x_i} and r_{x_i} of derived elements of length 2 (in general, we will write op_{x_i}) can be obtained either as $a \cdot op_{x_i} = b \cdot (c \cdot op_{x_i})$, or as $a \cdot op_{x_i} = (b \cdot op_{x_i}) \cdot c$, then it is enough to consider two cases.

Case 1. For reduction, multiplication of derivatives by elements $U(A)$ different from constants is used. In this case, it from partial derivatives must initially be a constant other than zero. By the criterion of primitiveness, the element is primitive.

Obviously, $\text{rk}\left(a \mid B \mid B^T\right) > \text{rk}\left(B \mid B^T\right)$, because in the matrix $\left(B \mid B^T\right)$ there is a zero row, and in the vector a there is a non-zero constant in this row.

Case 2. For reduction, only multiplication by elements of F is used.

In this case, the reduction algorithm reduces to solving linear system over the field F.

Using the Kronecker-Capelli criterion, we obtain a proof of the statement. □

The criterion obtained for elements of length 2 can be generalized to a test of the primitiveness of elements of arbitrary length. The transposition of the matrices is replaced by a permutation of the sides of the coefficient tables. And instead of the rank of the matrix, it is necessary to consider the rank of a system of vectors composed of elements of layers of multidimensional coefficient tables.

Only the case when only a linear reduction of a system of partial derivatives is performed falls under the action of this test. I.e., a linear system is solved:

$$\alpha_1 \frac{\partial}{\partial x_1} + \ldots + \alpha_n \frac{\partial}{\partial x_n} = 1, \quad \alpha_1, \ldots, \alpha_n \in F.$$

4. Estimation of the number of primitive elements with two generators of arbitrary length

Assuming that the coefficient for the word w is denoted as a_w, and using the primitiveness test, we found that the coefficients for words with the same arrangement of brackets are proportional. For example, for monomials of length 3 of the form $x(xx)$:

$$\begin{aligned} a_{x_2(x_2x_2)} &= t a_{x_2(x_2x_1)} = t^2 a_{x_1(x_2x_1)} \\ &= t a_{x_2(x_1x_2)} = t^2 a_{x_2(x_1x_1)} \\ &= t a_{x_1(x_2x_2)} = t^2 a_{x_1(x_1x_2)} = t^3 a_{x_1(x_1x_1)}. \end{aligned}$$

where t is the general coefficient of proportionality between nonunit partial derivative monomes. When expressing $a_{x_2(x_2x_2)}$ through other coefficients, its degree is equal to the number of x_1 in the monomial corresponding to the coefficient and vice versa, if we express $a_{x_1(x_1x_1)}$.

This rule is valid for monomials of any length. If the coefficient matrix for any group of monomials of length k is written by the k-dimensional table $(a_{i_1 i_2 \ldots i_k})$, $i_j = 1..2$, then either $a_{11\ldots1} = \ldots = t^k a_{22\ldots2}$, or $a_{22\ldots2} = \ldots = s^k a_{11\ldots1}$.

The Catalan number C_{k-1} of equations relates coefficients for monomials of length k.

For monomials of shorter length, the number of equations is $\sum_{i=2}^{k-1} C_{i-1}$.

Total $\sum_{i=2}^{k} C_{i-1}$ of equations relates coefficients for monomials.

Denoting $S_2^k(q)$ – the number of primitive elements of length k of a free non-associative algebra with two generators over the field F_q, we obtain the following expression:

$$S_2^k(q) \geqslant q(q-1)(q+1)q^{\sum_{i=2}^{k-1} C_{i-1}} \left(q^{C_{k-1}} - 1\right).$$

In particular,

$S_2^2(q) = q(q-1)(q+1)q^0(q^1-1) = q(q-1)^2(q+1)$ (equality is due to the fact that the primitiveness criterion for elements of length 2, which we used, has been proved).

$S_2^3(q) \geqslant q(q1)(q+1)q^1(q^2-1) = q^2(q-1)^2(q+1)^2$

$S_2^4(q) \geqslant q(q-1)(q+1)q^3(q^5-1) = q^4(q-1)(q+1)(q^5-1)$

$S_2^5(q) \geqslant q(q-1)(q+1)q^8(q^{14}-1) = q^9(q-1)(q+1)(q^{14}-1)$

The right part of the obtained estimate in the case of lengths 2 and 3 coincides with the formulas for counting such elements obtained earlier by A.A. Chepovskii in the dissertation [1].

References

[1] *A.A. Chepovskii.* Primitive elements of the algebras of Schreier varieties. Dissertation for the degree of Candidate of Physical and Mathematical Sciences. Moscow, 2011.

[2] *A.G. Kurosh.* Non-associative free algebras and free products of algebras. Matem. Sb. 20 (1977), no. 2, 239-262.

[3] *M.V. Maisuradze.* Software implementation of algorithms for working with primitive elements in free non-associative algebras. J. Intelligent Systems. Theory and Applications, Volume 25, Issue 4, Pages 170-175, 2021.

[4] *A.A. Mikhalev, A.V. Mikhalev, A.A. Chepovskii, K. Champagnier.* Primitive systems of free non-associative algebras. J. Math. Sci. 156 (2009), no. 2, 320-335.

[5] *A.A. Mikhalev, V. Shpilrain, J.-T. Yu.* Combinatorial Methods: Free Groups, Polynomials, and Free Algebras. Springer, New York, 2004.

[6] *A.A. Mikhalev, U.U. Umirbaev, J.-T. Yu.* Authomorphic orbits in free non-associative algebras. J. Algebra 243 (2001), 198-223.

[7] *A.A. Mikhalev, A.A. Zolotykh.* Combinatorial Aspects of Lie superalgebras. CRC Press, Boca Raton, 1995.

[8] *A.A. Mikhalev, A.A. Zolotykh.* Rank and primitivity of elements of free color Lie (p-) superalgenras. Internat. Journal Algebra Comput. 4 (1994), 617-656.

[9] *A.I. Shirshov.* Subalgebras of free Lie algebras. Matem. Sb. 33 (1953), no. 2, 441-452.

Computing of Tropical Sequences Associated with Somos Sequences in Gfan Package

F. Mikhailov

Saint Petersburg Electrotechnical University "LETI", Russia

e-mail: mifa_98@mail.ru

Abstract

The main objective of this work is to study tropical recurrent sequences associated with Somos sequences. For a set of tropical recurrent sequences, D. Grigoriev put forward a hypothesis of stabilization of the maximum dimensions of solutions to systems of tropical equations given by polynomials, which depend on the length of the sequence under consideration. The validity of such a hypothesis would make it possible to calculate the dimensions of these solutions for systems of arbitrary length. The main purpose of this work is to compute tropical sequences associated with Somos sequences using the Gfan package and to test the Grigoriev hypothesis.

Keywords: tropical recurrent sequence, tropical prevariety, Gfan package, Somos sequence

1. Introduction

Tropical mathematics is a young area of modern mathematics related to the study of semirings with idempotent addition. Despite its novelty, it has already found its application in algebra, geometry, mathematical physics, biology [1], economics, neural network theory [3], dynamic programming, and other areas.

This work is a continuation of the work [3], which was devoted to tropical linear recurrent sequences. As part of this work, tropical sequences associated with Somos sequences are computed in the Gfan package. The purpose of this work, as well as the previous one, is to test Grigoriev's hypothesis about the stabilization of the maximum dimensions of solutions to systems of tropical equations given by polynomials that depend on the length of the sequence under consideration. The validity of such a hypothesis would make it possible to calculate the dimensions of these solutions for systems of arbitrary length.

Gfan is a software package for computing universal Gröbner bases, some related geometric objects (Gröbner fans) and tropical varieties, developed in 2005 by A. Jensen, based on the algorithms described and developed in his dissertation [4]. The Gfan package can compute universal Gröbner bases, Gröbner fans, tropical prevarieties, varieties given by a system of tropical polynomials, and other objects of tropical geometry and the theory of Gröbner bases. It is currently the most powerful software tool for such computations. Gfan is distributed as a standard Linux package and is part of the Debian distribution.

2. Formulation of the problem

One of the main objects of tropical mathematics is the tropical semiring $(\mathbb{R} \cup \{-\infty\}, \oplus, \otimes)$. This set consists of real numbers with an additional element – minus infinity. In the tropical semiring, the classical operations of addition and multiplication over real numbers are replaced by the operations of taking the maximum and classical addition respectively: $x \oplus y := \max\{x, y\}$, $x \otimes y := x + y$. Tropical mathematics has its analogues of polynomial algebra, linear algebra and other areas of mathematics [5]. Taking the minimum

can be considered as tropical addition, then the additional element to the set of real numbers will be plus infinity.

Let $k \geq 2$ be a natural number and

$$\alpha = \{\alpha_i | 1 \leq i \leq [k/2]\}, \quad x = \{x_j | -k/2 < j \leq [k/2]\}$$

- two sets of independent formal variables in the amount of $[k/2]$ in the first case and k in the second. The sequence of rational functions Somos-k of variables from α and x, $S_k(n) = Sk(n; \alpha; x)(n \in \mathbb{Z})$, is defined by the recursive relation

$$S_k\left(n + \left[\frac{k+1}{2}\right]\right) S_k\left(n - \left[\frac{k}{2}\right]\right) = \sum_{1 \leq i \leq k/2} \alpha_i S_k\left(n + \left[\frac{k+1}{2}\right] - i\right) S_k\left(n - \left[\frac{k}{2}\right] + i\right).$$

This sequence for $k = 6$, $\alpha_1 = \alpha_2 = \alpha_3 = 1$, $x_{-2} = x_{-1} = x_0 = x_1 = x_2 = x_3 = 1$ was first considered by Michael Somos in connection with the study of the properties of elliptic theta functions [6].

In this work, we study the tropical sequences $p_k(n)$ associated with $S_k(n)$ that satisfy the recurrent relation

$$p_k\left(n + \left[\frac{k+1}{2}\right]\right) + p_k\left(n - \left[\frac{k}{2}\right]\right) = \min_{1 \leq i \leq k/2} \left\{ p_k\left(n + \left[\frac{k+1}{2}\right] - i\right) + p_k\left(n - \left[\frac{k}{2}\right] + i\right) \right\}.$$

An interesting fact is that the tropical analogue of such sequences is related to the classical Somos sequences by some relation. It was proved in [7] that $S_k(n)$ is a Laurent polynomial in the initial variables x_j and an ordinary polynomial in α_i. Therefore, it can be written as

$$S_k(n) = \left(\prod_{-k/2 < j \leq [k/2]} x_j^{p_k^{(j)}(n)} \right) P_k(n),$$

where $P_k(n) = P_k(n; \alpha; x$ are polynomials with integer coefficients and $p_k^{(j)}(n)$ are integer sequences.

In this work, we will consider all solutions of the finite sequences $p_k(n)$ with $0 \leq n \leq s$ for $k = 4$ and $k = 5$. To do this, we transform the tropical recurrent sequence into a system of tropical polynomials. The solution for the system of tropical polynomials will be found using the Gfan package, computing the tropical prevarieties for the system. Detailed computations for tropical linear recurrent sequences with all definitions are presented in [3].

3. Computations of Somos-4 sequences in the Gfan package

In this work, we consider sequences Somos-4 and Somos-5. Let us consider detailed computations for $k = 4$. To compute the sequences $p_4(n)$, we consider the sequences

$$q_4(n) = \Delta^2 p_4(n) = \Delta p_4(n+1) - \Delta p_4(n) = p_4(n+2) - 2p_4(n+1) + p_4(n).$$

Then the tropical relations will look like

$$q_4(n-1) + q_4(n) + q_4(n+1) + \max\{0, q_4(n)\} = 0$$

For computation in the Gfan package, we reduce this relation to a tropical polynomial. Let $y_n = q_4(n)$. Then we get

$$\max\{y_{n-1} + y_n + y_{n+1}, y_{n-1} + 2y_n + y_{n+1}\} = y_{n-1} \otimes y_n \otimes y_{n+1} \oplus y_{n-1} \otimes y_n^{\otimes 2} \otimes y_{n+1}$$

To find solutions to this relation, we find tropical prevarieties. Since tropical prevarieties are the set of nonsmoothness of a tropical polynomial, the difficulty for this is that this polynomial is equal to zero. To solve this problem, add 0 as a term to the tropical polynomial

$$y_{n-1} \otimes y_n \otimes y_{n+1} \oplus y_{n-1} \otimes y_n^{\otimes 2} \otimes y_{n+1} \oplus 0.$$

We can notice that the system of tropical polynomials $\max\{y_{n-1}+y_n+y_{n+1}, y_{n-1}+2y_n+y_{n+1}\}$ for $1 \leq n \leq s-1$ reaches a maximum greater than zero only in two cases: $y_0 > 0$ and $y_s > 0$. Because of this, the addition of the term 0 does not affect the dimension of the tropical prevariety. Therefore, to compute the dimensions of the solution space, these cases were excluded. This idea was verified experimentally in the Gfan package for computed finite sequences.

To compute tropical prevarieties, we compose a system of tropical polynomials for all relations for $1 \leq n \leq s-1$. Tropical prevarieties can be computed using the function **gfan_tropicalintersection** of the Gfan package [8]. Denote the dimension of the solution space by d_s. The obtained dimensions of the solution space are presented in Table. 1.

Table 1: Dimensions of the Somos-4 solution space

s	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
d_s	2	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5	6	6	6

The obtained solutions correspond to the calculations carried out in [9].

4. Computations of Somos-5 sequences in the Gfan package

The tropical relations in this case look like

$$q_5(n-2) + q_5(n-1) + q_5(n) + q_4(n+1) + \max\{0, q_5(n-1) + q_5(n)\} = 0.$$

Let $y_n = q_4(n)$. Then we get

$$\max\{y_{n-2} + y_{n-1} + y_n + y_{n+1}, y_{n-2} + 2y_{n-1} + 2y_n + y_{n+1}\} = 0.$$

Then we consider tropical prevarieties for the following polynomial

$$y_{n-2} \otimes y_{n-1} \otimes y_n \otimes y_{n+1} \oplus y_{n-2} \otimes y_{n-1}^{\otimes 2} \otimes y_n^{\otimes 2} \otimes y_{n+1} \oplus 0.$$

We can notice that the system of tropical polynomials $\max\{y_{n-2}+y_{n-1}+y_n+y_{n+1}, y_{n-2}+2y_{n-1}+2y_n+y_{n+1}\}$ for $2 \leq n \leq s-1$ reaches a maximum greater than zero only in three linear cases: $y_0 > 0$, $y_s > 0$ and $y_n = (-1)^n$. Because of this, the addition of the term 0 does not affect the dimension of the tropical prevariety. Therefore, to compute the dimensions of the solution space, these cases were excluded.

The obtained dimensions of the solution space are presented in Table. 2.

5. Conclusion

Based on the computed tropical prevarieties, we can make the assumption that for Somos-4 sequences $d_s = \left[\frac{s-2}{4}\right] + 2$. Then for such sequences the tropical entropy [3] takes the value

Table 2: Dimensions of the Somos-5 solution space

s	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
d_s	3	3	3	4	4	4	4	4	5	5	6	6	6	6	6	7	7	8	8	8

$H = 1/4$. For systems of tropical polynomials $y_{n-1} \otimes y_n \otimes y_{n+1} \oplus y_{n-1} \otimes y_n^{\otimes 2} \otimes y_{n+1}$ for $1 \leq n \leq s-1$ without addition 0, it is obtained that $d_s = 2$ for any s. Then for such sequences the tropical entropy takes the value $H = 0$.

Based on the computed tropical prevarieties, we can make the assumption that for Somos-5 the tropical entropy takes the value $H = 2/7$. For systems of tropical polynomials $y_{n-2} \otimes y_{n-1} \otimes y_n \otimes y_{n+1} \oplus y_{n-2} \otimes y_{n-1}^{\otimes 2} \otimes y_n^{\otimes 2} \otimes y_{n+1}$ for $2 \leq n \leq s-1$ without addition 0, it is obtained that $d_s = 3$ for any s. Then for such sequences the tropical entropy takes the value $H = 0$.

For the Somos-6 and Somos-7 cases, it is more difficult to find the dimension of the solution space using the computation of tropical prevarieties. The problem is that before adding zero as a tropical monomial to tropical polynomials, the solution space of finite sequences increases.

The results obtained are consistent with Grigoriev's hypotises on the stabilization of the maximum dimensions of solutions to systems of tropical sequences.

References

[1] *Sturmfels B.* Algebraic statistics for Computational Biology. Cambridge University Press, 2005.

[2] *Zhang L., Naitzat G., Lim L.* Tropical Geometry of Deep Neural Networks Proceedings of the 35th International Conference on Machine Learning, 2018.

[3] *F. Mikhailov* "Computing of the Dimensions of the Components of Tropical Prevarieties Described by Linear Tropical Recurrent Relations, Computer tools in education, no.1, pp.40–54, 2023. (In Russian)

[4] *Jensen A.N.* Algorithmic Aspects of Gröbner Fans and Tropical Varieties. Department of Mathematical Sciences, Aarhus, 2007.

[5] *Maclagan D., Sturmfels B.* Introduction to Tropical Geometry. Providence: American Mathematical Society, 2015.

[6] *Propp J.* The Somos Sequence Site. http://faculty.uml.edu/jpropp/somos.html.

[7] *Fomin S., Zelevinsky A.* The Laurent Phenomenon, Adv. Appl. Math., vol. 28, pp. 119-144, 2002.

[8] *Jensen A.N.* Gfan version 0.6: A User's Manual. Department of Mathematical Science. University of Aarhus, 2017.

[9] *Bykovskii V. A., Romanov M. A., Ustinov A. V.* Tropical sequences associated with Somos sequences, Chebyshevskii sbornik, vol. 22, no. 1, pp. 118–132, 2021. (In Russian)

Bounding the Support in the Differential Elimination Problem

Y.S. Mukhina

Faculty of Mechanics and Mathematics, Lomonosov Moscow State University, Russia

e-mail: js.mukhina@mail.ru

Abstract

We will discuss bounding the support in the differential elimination problem. The main result is the characterisation of the support of the result of the differential elimination for a planar system with generic polynomials of fixed degree in the right-hand side.

Keywords: differential elimination, Newton polytope, dynamical systems

Differential elimination is a differential analogue of elimination for polynomial systems and Gaussian elimination from linear algebra. Its study has been initiated by Ritt [1], the founder of differential algebra, in the 1930s. He developed the foundations of the characteristic set approach, which has been made fully constructive by Seidenberg [2]. The algorithmic aspect of this research culminated in the Rosenfeld-Gröbner algorithm [3, 4] implemented in the BLAD library [5] (available through Maple). In theory, differential elimination problem can be stated (and solved) in full generality but here we will focus on an important special case.

Let R be a differential ring. Consider a ring of polynomials in infinitely many variables

$$R[x^{(\infty)}] := R[x, x', x'', x^{(3)}, ...]$$

and extend the derivation from R to this ring by $(x^{(j)})' := x^{(j+1)}$. The resulting differential ring is called the ring of differential polynomials in x over R. The ring of differential polynomials in several variables is defined by iterating this construction.

Let $S := R[x_1^{\infty}, ..., x_n^{\infty}]$ be a ring of differential polynomials over a differential ring R. An ideal $I \subset S$ is called a differential ideal if $a' \in I$ for every $a \in I$. Denote by $\langle f_1, ..., f_s \rangle^{(\infty)}$ the differential ideal

$$\langle f_1^{(\infty)}, ..., f_s^{(\infty)} \rangle$$

for every $f_1, ..., f_s \in S$.

Consider a system of differential equations of the form

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}),$$

where $\mathbf{x} = (x_1, ..., x_n)$ is a tuple of differential indeterminates and $\mathbf{f} = (f_1, ..., f_n)$ is a tuple of polynomials from $\mathbb{C}[\mathbf{x}]$. Systems of these form describe dynamical systems with polynomial dynamics and appear often in the literature. One natural elimination task is to eliminate all the variables except one, say x_1, that is, describe a differential ideal

$$\langle x_1' - f_1(\mathbf{x}), ..., x_n' - f_n(\mathbf{x}) \rangle^{(\infty)} \cap \mathbb{C}[x_1^{(\infty)}].$$

In this report we will consider the case of system

$$x_1' - g_1(x_1, x_2) = x_2' - g_2(x_1, x_2) = 0,$$

where g_1 and g_2 are generic polynomials of degrees d_1 and d_2, respectively, and prove the characterization of Newton polytopes of the minimal polynomial of the corresponding elimination ideal for the degree pairs (d_1, d_2).

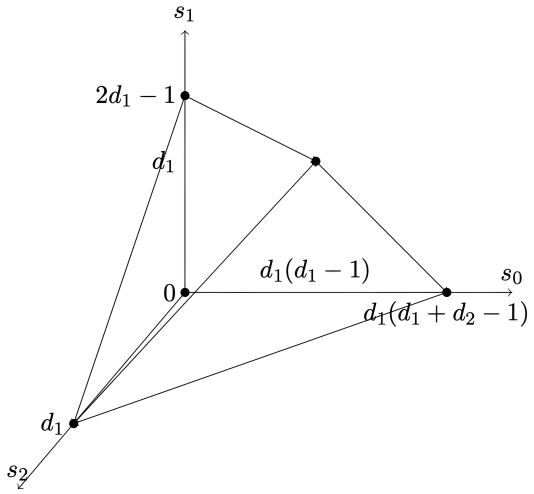


Figure 1: Newton polytope of the minimal polynomial for (d_1, d_2), $d_1 > d_2$ case.

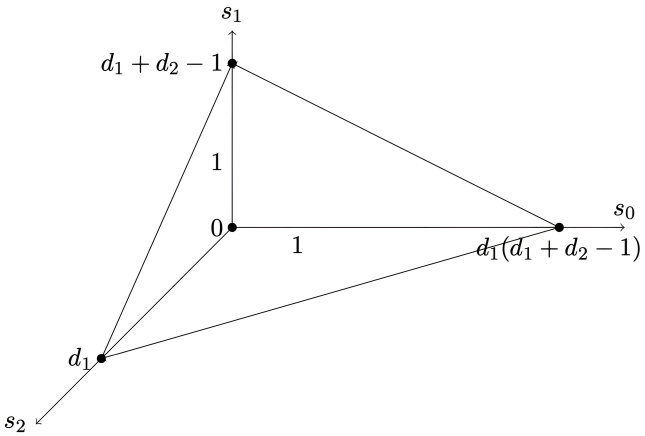


Figure 2: Newton polytope of the minimal polynomial for (d_1, d_2), $d_1 \geq d_2$ case.

Theorem 1. *Consider the system*

$$\begin{cases} x_1' = g_1(x_1, x_2), \\ x_2' = g_2(x_1, x_2), \end{cases} \tag{1}$$

where g_1 and g_2 are generic polynomials of degrees d_1 and d_2, respectively. Then the Newton polytope of the minimal polynomial of (1) *in (s_0, s_1, s_2)-coordinates $(x_1^{s_0}(x_1')^{s_1}(x_1'')^{s_2})$ is*

1. *a pyramid with vertices*

$$(0,0,0),\ (d_1(d_1+d_2-1),0,0),\ (d_1(d_1-1),d_1,0),\ (0,2d_1-1,0),\ (0,0,d_1)$$

   if $d_1 > d_2$ (see figure 1).

2. *a tetrahedron with vertices*

$$(0,0,0),\ (d_1(d_1+d_2-1),0,0),\ (0,d_1+d_2-1,0),\ (0,0,d_1)$$

   if $d_1 \leqslant d_2$ (see figure 2).

References

[1] *Ritt J.F.* Differential Equations from the Algebraic Standpoint. Colloquium Publications, American Mathematical Society. 1932.

[2] *Seidenberg A.* An elimination theory for differential algebra. University of California publications in Mathematics. 1956. Vol. III. P. 31–66.

[3] *Boulier F., Lazard D., Ollivier F., Petitot M.* Computing representations for radicals of finitely generated differential ideals. Applicable Algebra in Engineering, Communication and Computing. 2009. Vol. 20. P. 73–121.

[4] *Hubert E.* Notes on triangular sets and triangulation-decomposition algorithms II: Differential systems. Lecture Notes in Computer Science. Springer Berlin Heidelberg. 2003. P. 40–87.

[5] *Boulier F.* BLAD. Bibliothèques Lilloises d'Algèbre Différentielle.

A Note on Application of Program Specialization to Computer Algebra

A.P. Nemytykh
Program Systems Institute of RAS, Russia
e-mail: nemytykh@math.botik.ru

Abstract

Given an input program P, program specialization aims at run-time optimization of P w.r.t. its sintactic structures. The simplest example is to generate a definition of a (partial) subfunction of the (partial) function defined by P. One may wonder in some of syntactic properties of the program q resulted by specialization rather than run-time of q. We show a number of corollaries of known mathematical constructions, which derived by a general purpose tool, a specializer, and shortly introduce those properties of the tool, that allow it to achieve such interesting results. The idea of using such a tool for generating mathematical formulae was originated by Alexandr Korlukov.

Keywords: program specialization, finite extension of field, divisibility property, well quasi-order, free monoid

Memory of Alexandr V. Korlyukov

We use the following presentation programming language R. The alphabet $\mathcal{A}$ is a union of a set of `symbols` and the natural numbers. The data set is defined by the grammar: `d ::= (`$\mathtt{d}_1$`) | `$\mathtt{d}_1$ $\mathtt{d}_2$ ` | symbol | empty`

A program in R is a term rewriting system. The semantics of the language is based on pattern matching and call-by-value evaluation. As usually, the rewriting rules are ordered for matching from the top to the bottom. The terms are generated using two constructors. The first is concatenation and is denoted by the blank. The data set is a free monoid w.r.t. the concatenation[1]. The second constructor is unary. It is denoted with its parenthesis only (that is without a name) and is used for constructing tree structures. Every function is unary. Since the parenthesis is used as the constructor, the fucntion call is denoted by angle brackets closing both the function name and its arguments. Empty sequence is a special basic datum denoted with nothing. It is the neutral element of concatenation. Below we use sometimes the meta-symbol `[]` for the empty expression. There exist two types of variables: $name_e$ and $name_s$. An e-variable can take any datum as its value, an s-variable ranges over $\mathcal{A}$. For every rewriting rule its set of variables from the left side includes its set of variables from the right side.

1. Specialization for Deriving Formulae

The main goal of specialization of programs is their running time optimization. Nevertheless a specialized program represents a residual code that may be interesting by itself and even is not intended to be evaluated with some input.

[1]Associativity of concatenation may cause pattern matching to be ambiguous on some patterns. For example, the following equation $\mathtt{x}_e$ `` `y`$_e$ = A B `` has three solutions: 1) $\mathtt{x}_e$ `= [], `$\mathtt{y}_e$ `= A B`; 2) $\mathtt{x}_e$ `= A, `$\mathtt{y}_e$ `= B`; 3) $\mathtt{x}_e$ `= A B, `$\mathtt{y}_e$ `= []`; . In such cases the R pattern matching chooses the solution with the minimal length of the datum assigned to the first e-variable (from the left to the right) and so on by induction. See the examples given below and [10] for details. In our case the first solution $\mathtt{x}_e$ `= [], `$\mathtt{y}_e$ `= A B` will be chosen.

Let `theorem` be a program with two parameters $\texttt{condition}_1$ and $\texttt{condition}_2$. Then deriving a corollary $\texttt{theorem}_{condition_1}$ from `theorem` and $\texttt{condition}_1$ is a good example demonstrating Korlukov's idea:

$$theorem_{condition_1}(condition_2) = theorem(condition_1, condition_2).$$

The corollary $\texttt{theorem}_{condition_1}$ can be more useful then the general `theorem` when we are in the scope of $\texttt{condition}_1$.

Divisibility criteria are ways of telling whether one natural number divides another without actually carrying the division through. Divisibility criteria are constructed in terms of the digits that compose a given number. The criteria have to be simpler than the direct division of the second number by the first one. Let N be the number whose divisibility by another number d we are going to investigate. In the decimal system: $N = 10^n d_n + 10^{n-1} d_{n-1} + \cdots + 10 d_1 + d_0$, where $d_n \neq 0,\ 0 \leq d_i \leq 9$.

Let us consider a positive integer $\texttt{q}_s$ the following program represents an algorithm being a divisibility criterion by the number $\texttt{q}_s$. I.e. checking a given input (of the program) on divisibility is reduced to checking the output.

```
$ENTRY divide { q_s (ds_e) = <div 1 0 q_s (ds_e)>; }
div {
 m_s res_s q_s ( ) = res_s ;
 m_s res_s q_s (ds_e d_s) = <div <Mod <* m_s 10> q_s>> <+ res_s <* d_s m_s>> q_s (ds_e)>; }
```

Where `Mod` is a primitive function returning the remainder of dividing the first argument by the second, the primitive functions `+` and `*` stand for the corresponding mathematical functions.

Function `divide` takes as arguments a given number $\texttt{q}_s$ and a sequence of the decimal digits $\texttt{ds}_e$ $(d_n, d_{n-1}, \ldots d_0)$ of the number whose divisibility (by $\texttt{q}_s$) is investigated. We consequently calculate the remainders of the divisions of 10^i by $\texttt{q}_s$ and add them. We have united all rewriting rules of `div` and enclosed them with curly brackets following the function name. Figures 1 and 2 below shows a number of examples of specializing this program w.r.t $\texttt{q}_s^0$. Additional comments to the examples above will be given after a short introduction to the specialization method used.

Arithmetic in Finite Field Extensions. Given an algebraic closed field K of characteristic 0 and its subfield F there is a classical constructive method consructing finite extensions of F inside of K [8, 3]. The method is uniform both on K and its subfield set. Given a non-constant polynomial $p(x) \in K[x]$ which is irreducible over F, the field of fractions of $F[x]/(p)$ is an extension of F, which is isomorphic to the extension of F by a root of p (from K). For instance, $\mathbb{Q}(i) \cong \mathbb{Q}(x)/(x^2+1)$ is the field of the rational complex numbers. Thus the procedure of extension takes two arguments: F and $p(x)$.

Let `Extension` be a uniform definition of the arithmetic operations in the field of fractions. Consider specialization of `<Extension` $F[x], p_0$`>` w.r.t. a given irreducible polynomial p_0. The result of the specialization is definitions of the arithmetic operations in terms of $F[x]$. Thus, for example, if $p_0(x) = x^2 + 1$ (irreducible over F) the residual program represents formulae for calculating the complex numbers over the field F. Let $p_0(x) = x^2 - 2$ be irreducible over an F, then we obtain formulae for calculating in $F(\sqrt{2})$.

Like in the function `divide` above, we declare the operations in the original filed F undefined during specialization. That is to say, we call for functions defining the operations, but declare the functions as external w.r.t. the given program module (in the case of `divide` similar functions were primitive and the specializer had an information of properties of the functions). If the external functions are defined (in another module) as the operations over

Specialization of the program `divide` and its entry configuration <divide 3 (ds_e)> gives the following result:

```
$ENTRY Go { (ds_e d0_s) = <F19 (ds_e) d0_s>;}/* d0 + ...  + dn is divided by 3 */
F19 { ( )       x2_s = x2_s;
      (x1_e x3_s) x2_s = <F19 (x1_e) <+ x2_s x3_s>>; }
```

If the entry configuration is <divide 8 (e.ds)> the residual program looks as:

```
$ENTRY Go { (d0_s) = d0_s ; /* d0 + 2*d1 + 4*d2 is divided by 8 */
            (d1_s d0_s) = <+ d0_s <* d1_s 2>> ;
            (x1_e d2_s d1_s d0_s) = <+ <+ d0_s <* d1_s 2>> <* d2_s 4>>; }
```

For <divide 10 (ds_e)> the output of the specializer is:

```
$ENTRY Go { (ds_e d0_s) = d0_s ; } /* d0 is divided by 10 */
```

If we are interested in a divisibility criterion by 18, the specializer produces:

```
/* d0 + 10*(d1 + ...  + dn) is devided by 18 */
$ENTRY Go { (ds_e d0_s) = <F19 (ds_e) d0_s>; }
F19 { () x2_s = x2_s ;
      (x1_e x3_s) x2_s = <F19 (x1_e) <+ x2_s <* x3_s 10>>>; }
```

Figure 1: The results of specialization of the program `divide`.

the rational numbers, then in the two examples given above we have formulae for calculation in $\mathbb{Q}(i)$ and $\mathbb{Q}(\sqrt{2})$, respectively.

2. Some of Properties of the Specializer Used

The residual programs mentioned above were produced by the supercompiler SCP4 [6], a specializer based on specialization method known as Turchin's supercompilation [9]. We note that the language R specified above is a functional programming language. For the sake of simplicity in this section we consider only R-programs defining partial predicates rather than arbitrary partial functions.

Any R-program P can be seen as a evaluation tree being infinite, as a rule, i.e. any recursion is unfolded. The edges are labeled with P-patterns. Parameterized states of P label the tree nodes. They are predicates. The tree root is labeled with the initial parameterized state. Given a program and its initial parameterized state SCP4 explores the corresponding evaluation tree, starting from the tree root. It considers the predicates labeling some of the nodes as hypotheses to be proven. Given such a node it tries to prove the corresponding hypothesis by induction on the R-machine steps along every path originating in this node. There may be a number of hypotheses labeling different nodes to be simultaneously proven. The main problems are as follows. How does SCP4 determine the hypotheses-nodes to be proven? How does it decide that a current hypothesis is too strong to be automatically proven? And if it takes such a decision, then taking into account the main specialization aim, how should the statement be weakened? Both the problems arise from undecidability of the program optimization task per se. Thus SCP4 has to approximate the corresponding solutions. It is based on variants of Higman–Kruskal relation [1, 4], being well quasi-orders on the parameterized program states along the evaluation tree paths.

It is also possible to manually create annotations in the input program to be specialized, which may provide some support for SCP4. The structure of the proofs of all hypotheses

By 12 : $d_0 + 10d_1 + 4(d_2 + \cdots + d_n)$
That is equivalent to: $d_0 - 2d_1 + 4(d_2 + \cdots + d_n)$.
By 37 : $(d_0 + 10d_1 + 26d_2) + (d_3 + 10d_4 + 26d_5) + \ldots$
or $(d_0 + 10d_1 - 11d_2) + (d_3 + 10d_4 - 11d_5) + \ldots$
By 101 : $(d_0 + 10d_1 + 100d_2 + 91d_3 + d_4) + (d_5 + 10d_6 + 100d_7 + 91d_8 + d_9) + \ldots$
or $(d_0 + 10d_1 - d_2 - 10d_3 + d_4) + (d_5 + 10d_6 - d_7 - 10d_8 + d_9) + \ldots$

For example 2023 is not divided by 37 because $3 + 20 + 2 = 25$ is not divided by 37.

Figure 2: Additional divisibility criteria derived by the specializer.

encountered and maybe weakened specifies the residual program. We refer the reader to [9, 5] for details.

References

[1] *Higman G.* Ordering by Divisibility in Abstract Algebras. In Bulletin of London Math. Soc. 1952. Vol. 3, N. 2. P. 326–336.

[2] *Korlyukov A.V.* A number of examples of the program transformations by the supercompiler SCP4. 2001. (In Russian) `http://www.refal.net/~korlukov/pearls/`.

[3] *Kostrikin A.I.* Introduction to Algebra. New York, NY: Springer-Verlag. (1982).

[4] *Kruskal J.B.* Well-quasi Ordering, the Tree Theorem, and Vazsonyi's Conjecture. In Transactions of the American Mathematical Society. 1960. Vol. 95. P. 210–225.

[5] *Lisitsa A.P., and Nemytykh A.P.* Verification as a Parameterized Testing (Experiments with the SCP4 Supercompiler), J. Programming and Computer Software. 2007. Vol. 33, No. 1, P. 14–23, (The paper is a translation from the Russian version of the journal.)

[6] *Nemytykh A.P., and Turchin V.F.* The Supercompiler SCP4: sources, on-line demonstration. `http://www.botik.ru/pub/local/scp/refal5/`. (2000–2023)

[7] *Nemytykh A.P.* A Note on Elimination of Simplest Recursions. In Proc. of the ASIAN symposium on PEPM. 2002. P. 138–146.

[8] *Postnikov M.M.* Galois Theory. Moscow, Fiz.-Mat. Lit. (1963). (In Russian)

[9] *Turchin V.F.* The Concept of a Supercompiler. ACM Transactions on Programming Languages and Systems. 1986. Vol. 8, P. 292–325.

[10] *Turchin V.F.* Refal-5, Programming Guide & Reference Manual. Holyoke, Massachusetts: New England Publishing Co. (1989) (*electronic version:* `http://www.botik.ru/pub/local/scp/refal5/`, 2000).

Learning Port-Hamiltonian Systems

V.N. Salnikov
CNRS / La Rochelle University, France
e-mail: vladimir.salnikov@univ-lr.fr

Abstract

The port-Hamiltonian approach provides a natural formalism for studying mechanical and physical systems with interaction or dissipation. We address the problem of recovering an optimal port-Hamiltonian structure for systems of ordinary differential equations. The procedure includes a machine learning based phase – isolating the nodes in the connectivity graph of the system and a "deterministic" phase – spelling-out the internal geometric structure of each node.

Keywords: geometric mechanics, port-Hamiltonian systems, symplectic and Poisson structures, learning connectivity.

Introduction / motivation

This contribution is a part of a series of works related to geometric numerical methods or more generally geometric mechanics – that is a study of methods and tools coming from differential geometry which are useful for describing physical and mechanical systems, for their qualitative analysis as well as reliable and efficient modelling and computer simulation. We have made an overview of active research directions in the context ([1]) and realized that some of them give rise to problems suitable for computer algebra algorithms ([2]).

We have in particular mentioned the so-called port-Hamiltonian systems (PHS). The idea of those is rather natural: consider several mechanical systems governed by the classical *Hamiltonian dynamics* (see e.g. [3] for the terminology from analytical mechanics); make them interact by introducing some forces which will be encoded in *ports*. The construction, to the best of our knowledge, first appeared in [4] with a rather straightforward engineering approach, and was revisited in [5], where a kind of catalog of "components" appeared.

While conceptually the representation of such interacting systems in the port-Hamiltonian form is merely a "game of notations", it has important consequences for efficient simulation of those. One popular approach is to use the structure of PHS for a smart strategy of distributed computations. An example (far from being unique) of such a strategy is provided by modelling in synthetic music and acoustics ([6]).

One notices immediately that in all what is mentioned so far, there is one natural direction of constructing a PHS (and the evolution equations) from the knowledge of physics of the studied system and the interaction of its submodules. A more complicated direction would be the other way around: from an arbitrary system of differential equations reconstruct the structure of a PHS that represents it. The motivation for that construction will be the same as before: produce a smart way of parallelizing / distributing the simulation of the studied system. In what follows we will present an algorithm of this reconstruction.

PHS in a nutshell

Let us briefly recall that port-Hamiltonian systems are traditionally ([5]) decomposed to a collection of interconnected subsystems (nodes), each of them can be written in the form:

$$\dot{\mathbf{x}} = (J(\mathbf{x}) - R(\mathbf{x}))\frac{\partial H}{\partial \mathbf{x}} + w(\mathbf{x})\mathbf{u}, \tag{1}$$

Here $\mathbf{x}$ is a vector-valued variable characterizing the state of the node, $J(\mathbf{x})$ is a skew-symmetric matrix (subject to some differential conditions), together with the Hamiltonian function it corresponds to the conservative (internal) part of the node. $R(\mathbf{x}), w(\mathbf{x})$ and $\mathbf{u}$ are the new terms corresponding to *ports*. Morally, R encodes internal forces, while w and $\mathbf{u}$ are responsible for interaction with the "external world" (for the given node). One of the messages of [5] is to classify the physically motivated origins of these terms: storage, dissipation, control, etc, but for the current study we will not need those details.

The only important thing left to mention is that the whole PHS is constructed out of systems like (1) (obviously with different $\mathbf{x}$, J, R, ...) like building blocks – they are interconnected via ports. The J, R and H variables stay internal for each node, while w and $\mathbf{u}$ from different nodes start literally "talking to each other", those interactions are called input–output or flux–effort. Figure 1 from [7] is a schematic representation of a PHS in the form of a decorated graph.

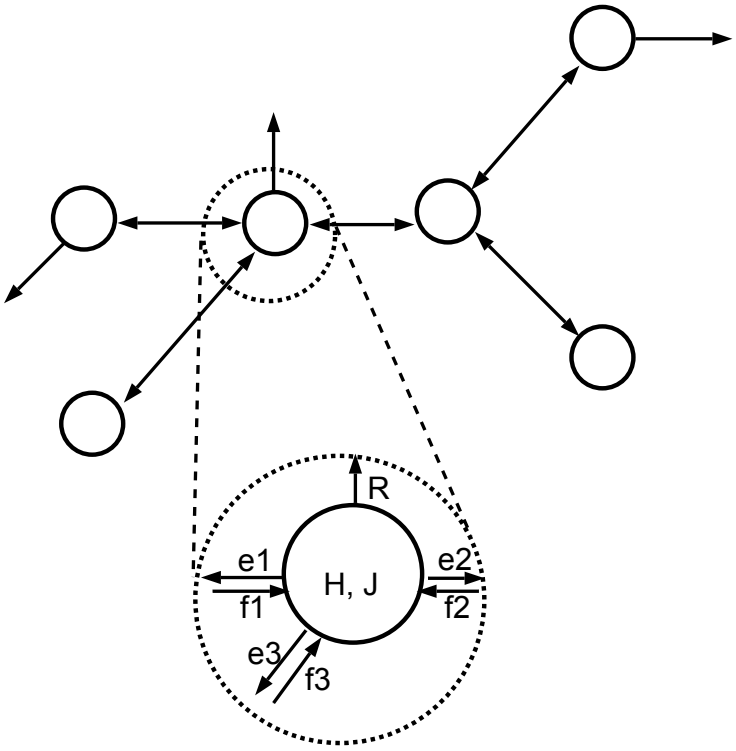


Figure 1: Representation of a port-Hamiltonian system. Image courtesy – [7].

The question we are addressing now is how to recover this graph structure together with the data of Equation (1) for each node, given an arbitrary system of differential equations $\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X})$. Very roughly speaking the difficulty lies in severe non-uniqueness of the solution, it is due to the fact that several systems of the form (1) connected together produce again a (larger) system of the form (1). So one extreme case would be a graph with just one node and only internal structure. The other extreme case is the opposite: forgetting completely the conservative part of the subsystems, declare all the components of $\mathbf{X}$ to be nodes and all the right-hand-sides to be ports. Both such solutions clearly do not make sense from the perspective of distributed computations, so the real question we ask is to recover the PHS structure which is somehow optimal. The solution is provided by the algorithm below.

Algorithm

Input data: a system of differential equations in the form $\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X})$

I. Recovering the connectivity graph structure:

1. Declare all the components of $\mathbf{X}$ to be vertexes of the graph. From the right-hand-sides recover the connectivity information for it.
2. Regroup strongly connected subsets of vertexes into clusters – those will be the nodes of the final graph.
3. For each node separate the terms in the right-hand-sides that correspond to internal interactions (i.e. depending only on variables of their own node) and external ones (all the others).

II. Decorating the graph part 1 – for each node recovering the Hamiltonian structure:

1. Construct (or select from a catalog constructed in advance) all the matrices J of suitable size.
2. From the internal variables select the maximal combination of terms, preserving one of the structures from the previous step. Check compatibility conditions.
3. Construct the function H satisfying (1), corresponding to the selected combination of terms. If this step results in unreasonably complicated symbolic computations, come back to steps 1.-2., dropping highly non-linear terms from the selection.

III. Decorating the graph part 2 – for each node recovering the ports:

1. The internal terms not selected in step II.2. are declared internal ports, associate "virtual" vertexes to them.
2. To external terms for each group (responsible for interactions) assign an edge in the resulting graph.

Output: the decorated graph, like on figure (1) is constructed.

Discussion

Several remarks are in place. First of all, the algorithm presented above is more a strategy of action, we have provided details of various steps of it in [7] and [8]. Let us briefly sketch them here for self-consistency of the paper.

The step I.2. is actually the crucial part of the whole construction. It turns out that purely symbolic (deterministic) algorithms are very difficult to produce – this is explained by the non-uniqueness of solutions mentioned above and a rather vague notion of optimality. The way out was suggested by the use of machine learning algorithms being extremely popular nowadays, hence the word "learning" in the title. Indeed, the problem turned out to be very suitable for simple neural networks algorithms: we are able to generate sufficiently many port-Hamiltonian systems with a known connectivity structure (for example using the results of [6] and the library [9]) – those are used to train the network. For the precise choice of the algorithms, we have tried several approaches ranging from purely matrix networks

to ones specially tailored for graph analysis. The most efficient method from the scalability perspective turned out to be based on a rather standard graph pooling – details and benchmarking will be available in [8].

Second, the step II.2. is also an intricate question, having its origins in symplectic or Poisson geometry. The mentioned compatibility conditions are related to some cohomological constructions. We have presented the details in [7], and the necessary vocabulary is reviewed in [2]. It is important to note that in the purely Hamiltonian formulation the problem does not necessarily admit a solution, but the possibility to shift some of the terms to ports actually explains the existence.

Last but not least, is a rather conceptual remark about the perspectives of this study. The approach presented above can be viewed regardless of the potential application for modelling and simulation of mechanical systems. In the essence it is a way to "order" a system of ordinary differential equation. In contrast to standard approaches often based on ordering polynomials or classification of elementary functions, the suggested algorithm rather takes into account the internal structure of the system as a whole. Hence it may be viewed as a new approach to definition of normal forms of systems of differential equations.

References

[1] V. Salnikov, A. Hamdouni, D. Loziienko, Generalized and graded geometry for mechanics: a comprehensive introduction, Mathematics and Mechanics of Complex Systems, Vol. 9, No. 1, 2021.

[2] V. Salnikov, A. Hamdouni, Differential Geometry and Mechanics - a source of problems for computer algebra, Programming and Computer Software, Vol. 46, Issue 2, 2020.

[3] V. I. Arnold, Mathematical Methods of Classical Mechanics, Graduate texts in Mathematics, 60, Springer, 1989.

[4] H. M. Paynter, Analysis and Design of Engineering Systems, MIT Press, Cambridge, Massachusetts, 1961.

[5] A. van der Schaft, Port-Hamiltonian systems: an introductory survey, Proceedings of the International Congress of Mathematicians, Madrid, 2006.

[6] A. Falaize, Modélisation, simulation, génération de code et correction de systèmes multiphysiques audios: Approche par réseau de composants et formulation hamiltonienne à ports, PhD thesis, Télécommunication et Électronique de Paris, Université Pierre et Marie Curie, 2016.

[7] V. Salnikov, A. Falaize, D. Loziienko, Learning port-Hamiltonian systems – algorithms, Computational Mathematics and Mathematical Physics, 63, 2023.

[8] V. Salnikov, A. Falaize, D. Loziienko, Learning port-Hamiltonian systems – applications, in preparation.

[9] Modeling, simulation and code-generation of multiphysical Port-Hamiltonian Systems in Python: https://github.com/pyphs/pyphs

On a Simple Lower Bound for the Matrix Rank

A.V. Seliverstov

Institute for Information Transmission Problems of Russian Academy of Sciences (Kharkevich Institute), Russia

e-mail: slvstv@iitp.ru

Abstract

Over a field of characteristic not equal to two, we proved a lower bound for the rank of a square matrix, where every entry outside the leading diagonal is equal to either zero or one, but every diagonal entry is neither zero nor one. This lower bound equals half of the order of the matrix. It is tight.

Keywords: matrix rank, affine subspace, computational complexity

The rank of an $n \times n$ matrix over a field can be calculated using a polynomial number of processors and performing only $O(\log_2^2 n)$ algebraic operations per processor [1, 2]. On the other hand, the computational complexity of both matrix rank [3] and the characteristic polynomial [4, 5] is equivalent in complexity to matrix multiplication. In practice, calculating the matrix rank requires a lot of time or a large number of processors. Simple lower bounds are important for planning calculations because a sufficiently large rank ensures the applicability of some algorithms for solving pseudo-Boolean programming problems [6, 7]. The distribution of the matrix rank over a finite field is used in cryptography [8].

Let us denote by K an arbitrary field of characteristic not equal to two. Let us consider an $n \times n$ matrix over the field K, where every entry outside the leading diagonal belongs to the set $\{0, 1\}$, but every diagonal entry is neither 0 nor 1. How small can its rank be?

This problem has a simple geometric interpretation. We consider an affine space over a field K with a fixed system of Cartesian coordinates. A point is identified with a column, where entries are coordinates of the point in this coordinate system. A column of zeros and ones corresponds to a $(0, 1)$-point, i.e., to a vertex of the unit cube. In matrices under consideration, each column corresponds to a point in a straight line passing through two adjacent $(0, 1)$-points, but this point does not coincide with any of $(0, 1)$-points. Moreover, different columns of the matrix correspond to non-parallel straight lines.

The rank of a matrix A is related to the dimensionality of the affine hull L of all points corresponding to columns of the matrix. If L passes through the origin, then $\operatorname{rank}(A) = \dim(L)$, else $\operatorname{rank}(A) = \dim(L) + 1$.

Theorem 1. *Given an $n \times n$ matrix A over the field K, where every entry outside the leading diagonal belongs to the set $\{0, 1\}$, but every diagonal entry is neither 0 nor 1. The rank of the matrix A is at least $n/2$.*

Proof. The theorem is obvious when the matrix A has at most two columns because $\operatorname{rank}(A) \geq 1$.

Let the theorem be proved for some $n \geq 3$ and for all $m \times m$ matrices with $m < n$. Let us consider an $n \times n$ matrix A.

A column of the matrix A corresponds to a point in a straight line passing through two adjacent $(0, 1)$-points, but this point itself is different from any $(0, 1)$-point. Changes of coordinates $x_k \to 1 - x_k$ (for different indices k) commute with each other and map each $(0, 1)$-point to some $(0, 1)$-point. Such coordinate transformations preserve the dimensionality of the affine hull of given points, as well as the number of $(0, 1)$-points belonging to this affine hull. Therefore, if no $(0, 1)$-point belongs to this affine hull, then such transformations do not affect the rank of the matrix.

By applying these transformations to the matrix A, one can obtain a matrix M of the same type so that in the last column of the matrix M all entries vanish except for the entry belonging to the leading diagonal. Removing both last column and last row from the matrix M, we get the $(n-1)\times(n-1)$ matrix B of lower rank. By the inductive hypothesis, $\operatorname{rank}(B)\geq(n-1)/2$. Thus, $\operatorname{rank}(M)\geq n/2$.

Let us denote by L the affine hull of all points corresponding to columns of M. Two cases are possible. If the origin belongs to L, then $\operatorname{rank}(M)=\dim(L)$. Therefore, the rank $\operatorname{rank}(A)\geq\dim(L)=\operatorname{rank}(M)\geq n/2$.

Else if the origin does not belong to L, then $\operatorname{rank}(A)\geq\operatorname{rank}(M)-1=\operatorname{rank}(B)$. By applying some transformations to the matrix B, one can obtain a matrix N of the same type so that in the last column of the matrix N all entries vanish except for the entry belonging to the leading diagonal. Moreover, $\operatorname{rank}(B)\geq\operatorname{rank}(N)$. Removing both last column and last row from the matrix N, we get the $(n-2)\times(n-2)$ matrix C of lower rank. By the inductive hypothesis, $\operatorname{rank}(C)\geq(n-2)/2=(n/2)-1$. Thus, $\operatorname{rank}(N)\geq n/2$. Therefore, $\operatorname{rank}(A)\geq\operatorname{rank}(B)\geq\operatorname{rank}(N)\geq n/2$. □

The lower bound is tight. Let $\lceil\cdot\rceil$ denote rounding up.

Theorem 2. *For every odd n, there is an $n\times n$ matrix A over the field K such that every entry outside the leading diagonal belongs to the set $\{0,1\}$, every diagonal entry is neither 0 nor 1, no $(0,1)$-point belongs to the affine hull of all points corresponding to columns of the matrix A, and the equality $\operatorname{rank}(A)=\lceil n/2\rceil$ holds.*

Proof. Let us consider the $n\times n$ matrix

$$A=\begin{pmatrix} 1/2 & 0 & 1 & 0 & 1 & \cdots & 0 & 1\\ 0 & -1 & 1 & 0 & 0 & \cdots & 0 & 0\\ 0 & 1 & -1 & 0 & 0 & \cdots & 0 & 0\\ 0 & 0 & 0 & -1 & 1 & \cdots & 0 & 0\\ 0 & 0 & 0 & 1 & -1 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & 0 & 0 & \cdots & -1 & 1\\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & -1 \end{pmatrix}.$$

Let us denote by B an $(n-1)\times(n-1)$ matrix obtained by removing both first column and first row from the matrix A. Obviously, $\operatorname{rank}(A)=\operatorname{rank}(B)+1$. The matrix B is block-diagonal with 2×2 blocks. All blocks are degenerate. Thus, $\operatorname{rank}(B)=(n-1)/2$. Next, $\operatorname{rank}(A)=\operatorname{rank}(B)+1=(n+1)/2=\lceil n/2\rceil$.

Every column of the matrix A is a solution to the inhomogeneous system of equations

$$\begin{cases} 2x_1-x_2-\cdots-x_{2k}-\cdots-x_{n-1} &= 1\\ x_{2k}+x_{2k+1} &= 0,\quad 1\leq k\leq(n-1)/2 \end{cases}$$

This system defines the affine hull, which does not pass through any $(0,1)$-point. □

Example 1. For the 3×3 matrix

$$\begin{pmatrix} 1/2 & 0 & 1\\ 0 & -1 & 1\\ 0 & 1 & -1 \end{pmatrix},$$

the rank equals two. Three columns correspond to three points belonging to a straight line L. The straight line L is given by the system of two equations $1-2x_1+x_2=0$ and $x_2+x_3=0$. But the straight line L does not pass through any of the $(0,1)$-points.

Example 2. For 2×2 matrices under consideration, the rank equals one for matrices

$$\begin{pmatrix} 1/\alpha & 1 \\ 1 & \alpha \end{pmatrix},$$

where $\alpha \notin \{0, 1\}$. Two points corresponding to columns of this matrix belong to a straight line that passes through the origin, i.e., through a $(0, 1)$-point. This straight line is given by the equation $x_2 = \alpha x_1$. Therefore, if no $(0, 1)$-point belongs to the affine hull of all points corresponding to columns of the matrix A, then $\mathrm{rank}(A) = 2$.

Theorem 3. *Given an even n and an $n \times n$ matrix A over the field K, where every entry outside the leading diagonal belongs to the set $\{0, 1\}$, but every diagonal entry is neither 0 nor 1. If no $(0, 1)$-point belongs to the affine hull of all points corresponding to columns of the matrix A, then the rank of the matrix A is at least $(n/2) + 1$.*

References

[1] *Chistov A.L.* Fast parallel calculation of the rank of matrices over a field of arbitrary characteristic. In: L. Budach (eds) Fundamentals of Computation Theory. FCT 1985. Lecture Notes in Computer Science, vol. 199. Springer, Heidelberg, 1985, pp. 63–69. https://doi.org/10.1007/BFb0028792

[2] *Mulmuley K.* A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. Combinatorica. 1987. Vol. 7, N. 1, pp. 101–104. https://doi.org/10.1007/BF02579205

[3] *Cheung H.Y., Kwok T.C., Lau L.C.* Fast matrix rank algorithms and applications. Journal of the ACM. 2013. Vol. 60, N. 5, Article 31, pp 1–25. https://doi.org/10.1145/2528404

[4] *Pereslavtseva O.N.* Calculation of the characteristic polynomial of a matrix. Discrete Mathematics and Applications. 2011. Vol. 21, N. 1, pp. 109–128. https://doi.org/10.1515/DMA.2011.008

[5] *Neiger V., Pernet C.* Deterministic computation of the characteristic polynomial in the time of matrix multiplication. Journal of Complexity. 2021. Vol. 67, N. 101572, pp. 1–35. https://doi.org/10.1016/j.jco.2021.101572

[6] *Seliverstov A.V.* Binary solutions to large systems of linear equations. Prikladnaya Diskretnaya Matematika. 2021. N. 52, pp. 5–15. (In Russian) https://doi.org/10.17223/20710410/52/1

[7] *Seliverstov A.V.* Generalization of the subset sum problem and cubic forms. Computational Mathematics and Mathematical Physics. 2023. Vol. 63, N. 1, pp. 48–56. https://doi.org/10.1134/S0965542523010116

[8] *Kruglov V.I., Mikhailov V.G.* On the rank of random matrix over prime field consisting of independent rows with given numbers of nonzero elements. Matematicheskie Voprosy Kriptografii. 2020. Vol. 11, N. 3, pp. 41–52. (In Russian) https://doi.org/10.4213/mvk331

Calculations of Quantum Corrections in Supersymmetric Theories Using Computer Algebra Methods

I.E. Shirokov

Department of Theoretical Physics, Faculty of Physics, Lomonosov Moscow State University, Russia

e-mail: shi95@yandex.ru

Abstract

We propose a symbolic algorithm and a C++ program for generating and calculating supersymmetric Feynman diagrams for $\mathcal{N} = 1$ supersymmetric electrodynamics. The program generates all diagrams that are necessary to calculate a specific contribution to the two-point Green function of matter superfields in the needed order, and then reduces the answer to the sum of Euclidean momentum integrals.

Keywords: computer algebra, quantum corrections, supersymmetry

1. Introduction

Computer algebra methods have been used in high energy physics for a long time. [1].

Various programs for calculations in high-energy physics can be divided into several groups. The first group includes programs that can generate Feynman diagrams in the given theories. The most famous ones are QGRAPH [2] and FeynArts [3, 4]. The first one uses mathematical graph theory, the second one works directly with the Vick theorem.

The second group can work with momentum integrals produced by diagrams. The problem in this case is that such integrals are usually divergent so some regularization must be implemented. Mostly such programs work with dimensional regularization. In the case of integrals in d dimension a lot of methods were found to take them. Usually integrals are reduced using special methods [5] to limited number of so called master integrals (e.g. LiteRed [6]). Then using standard methods these integrals can be taken (e.g. AMBRE [7]). The third group appeared recently and try to combine generation of graphs and calculation of integrals. These are FeynMaster [8], HepLib [9] and `tapir` [10]. Mostly they use QGRAPH program to generate graphs and different software to calculate integrals.

Some of the mentioned programs were designed to work with such theories (e.g. FeynArts [11]). But they work in component field terms. It is more comfortable to make calculations in terms of superfields where manifest supersymmetry exists. There are also such programs that can work in terms of superfields (SUSYCAL [12] and Susymath [13]). Unfortunately they are rather limited, i.e. they cannot generate supegraphs, moreover now they are unavailable to download.

Thus, we can note that despite the significant progress in this area, there is a noticeable lack of software for working within supersymmetric theories in terms of superfields. So recently new computer-algebraic approaches for working with superfields in the superspace were proposed [14]. Using them, a program was created, that is capable to generate Feynman diagrams in terms of superspace, as well as to perform various operations with them, after which the result is output in the form of standard Feynman integrals.

2. $\mathcal{N} = 1$ superspace formalism

$\mathcal{N} = 1$ superspace is a space with the coordinates (ct, x, y, z, θ), where θ is a Majorana spinor[1]. The spinor indices are raised and lowered using charge conjugation matrices:

$$\theta^a \equiv \theta_b C^{ba}; \qquad \theta_a = \theta^b C_{ab}.$$

The supersymmetric covariant derivative is usually introduced as follows:

$$\bar{D}_{\dot{a}} = \frac{\partial}{\partial \bar{\theta}^{\dot{a}}} - i(\gamma^\mu)_{\dot{a}}{}^b \theta_b \, \partial_\mu.$$

The usual fields in this approach are components of superfields. So, the gauge field is a component of the real superfield $V(x^\mu, \theta)$, spinor, and scalar superfields are components of chiral or antichiral fields ($\phi(x^\mu, \theta)$ and $\phi^*(x^\mu, \theta)$, respectively), which, by definition, satisfy the conditions:

$$\bar{D}_{\dot{a}}\phi = 0, \quad D_a\phi^* = 0.$$

Integration is introduced with respect to θ variables. In our notation , it can be defined as follows:

$$\int d^2\bar{\theta} = \frac{1}{2}\bar{D}^2 = \frac{1}{2}\bar{D}^{\dot{a}}\bar{D}_{\dot{a}}, \quad \int d^2\theta = -\frac{1}{2}D^2 = \frac{1}{2}D^a D_a$$

$$\int d^4\theta = \int d^2\bar{\theta} d^2\theta.$$

3. Perturbation theory in $\mathcal{N} = 1$ superspace formalism

We now consider how the standard perturbation theory works in the superspace[2]. It is most convenient to carry out quantization by the functional integral method. The main element of this approach, from which various quantities can be obtained, is the generating functional Z, which is constructed as follows:

$$Z = e^{iS_{int}(\frac{1}{i}\frac{\delta}{\delta j}, \frac{1}{i}\frac{\delta}{\delta J})} \int \mathcal{D}(\text{superfields})\, e^{i(S^{(2)} + S_{\text{sources}})} = e^{iS_{int}(\frac{1}{i}\frac{\delta}{\delta j}, \frac{1}{i}\frac{\delta}{\delta J})} Z_0 \tag{1}$$

where $S^{(2)}$ is a contribution to the action quadratic in fields, and S_{int} is a sum of contributions of degree higher than 2, $S_{\text{sources}} = j\phi + JV$ and $\mathcal{D}(\text{superfields})$ is a measure of the functional integration. In the end all sources must be set to zero. The Gaussian integral Z_0 is taken using standard methods. The interaction term series expansion is interpreted graphically using Feynman diagrams. However we will consider the effective action. It is obtained by Legendre transformation:

$$\Gamma = -i \ln Z[\text{Sources}] - S_{\text{sources}}|_{\text{sources}\longrightarrow\text{superfields}}, \tag{2}$$

In fact, Γ removes all disconnected diagrams and one-particle reducible diagrams from Z. These are diagrams that can be divided by cutting a single internal line.

[1] You can read more about spinors, for example, in [15]

[2] You can read about the usual perturbation theory in QFT, for example, in [16]

4. Main steps of calculation and result

First of all exponent in the (1) should be expanded. We work directly with this formula and add external fields to the exponent in order to obtain effective action part according to Legendre transformation (2). For example one of such terms takes form:

```
-1/8*i*e^4*F#_1^1{-1}*J_1^1{-7}*j_1_1{-11}*j#_2^1{7}*J_2^1{-7}*F_2_1{1}
*j#_3^1{1001}*J_3^1{-7}*J_3^1{-11}*j_3_1{-13}
```

$$\Gamma \sim -i\frac{e_0}{2}\int d^8x_1\,\phi_1^*\frac{1}{i}\frac{\delta}{\delta J_1}\frac{1}{i}\frac{\delta}{\delta j_1}\frac{e_0^2}{2}\int d^8x_2\,\frac{1}{i}\frac{\delta}{\delta j_2^*}\frac{1}{i}\frac{\delta}{\delta J_2}\frac{1}{i}\frac{\delta}{\delta J_2}\frac{1}{i}\frac{\delta}{\delta j_2}\times\frac{e_0}{2}\int d^8x_3\,\frac{1}{i}\frac{\delta}{\delta j_3^*}\frac{1}{i}\frac{\delta}{\delta J_3}\phi_3 Z_0\Big|_{J,j=0}$$

here `J_2^1{-7}` and `j_3_1{-13}` means derivatives by sources.

At the second step we need to collect all the derivatives in pairs, because Z_0 has only squares of sources.

```
-1/4*i*e^4*F#_1^1{-1}*F_2_1{1}*J_1^1{-7}*J_3^1{-7}*j_1_1{-11}*j#_3^1{1001}
*j#_2^1{7}*j_3_1{-13}*J_2^1{-7}*J_3^1{-11}
```

$$\Gamma \sim \frac{-ie_0^4}{8}\int d^8x_1\,d^8x_2\,d^8x_3\,\phi_1^*\phi_3\overbrace{\frac{\delta}{\delta J_1}\frac{\delta}{\delta j_1}\frac{\delta}{\delta j_2^*}\frac{\delta}{\delta J_2}}\overbrace{\frac{\delta}{\delta J_2}\frac{\delta}{\delta j_2}\frac{\delta}{\delta j_3^*}\frac{\delta}{\delta J_3}}Z_0\Big|_{J,j=0}$$

From the programming point of view this means that we simply move pairs together to the end of list. At this point it is necessary to check if the diagram is one-particle reducible or not and delete reducible ones. Next we need to use Z_0. For example in supersymmetric quantum electrodynamics (SQED) it takes form:

$$Z_0(J,j)=\exp\{\frac{i}{2}\int d^8xJ\left[-\frac{2}{R\partial^4}\right]J+i\int d^8x\left(j\frac{1}{\partial^2}j^*+\tilde{j}\frac{1}{\partial^2}\tilde{j^*}\right)\}.$$

Each pair produces so-called propagators that also consists of superspacial delta function, that acts just like simple delta-function but in superspace:

$$\frac{\delta}{\delta J_1}\frac{\delta}{\delta J_2}Z_0(J,j)=-\frac{2}{R\partial^2}\delta_{12}^8 \rightarrow \texttt{-2*e^0*I\{1^-2\}*K4\{1\}*d\_\{1\}\{12\}}$$

$$\frac{\delta}{\delta j_1}\frac{\delta}{\delta j_2^*}Z_0(J,j)=\frac{\bar{D}_1^2D_2^2}{4\partial^2}\delta_{12}^8 \rightarrow \texttt{1/4*e^0*I\{1^-2\}*D\#\_1(D\_2(d\_\{1\}\{12\}))}$$

Here we need to move to momentum representation. Each momentum is given by a prime number, and the sum of the momenta corresponds to their product, so each number uniquely sets the sum. For example: we have momenta k, l and q. Let us assign k-number 2, l-number 3, q-number 5. Then, for example, $k+l$ will be 6, and $k+l+q$ will be 30. This makes comparison and other operations easier. The results takes form:

```
1/16*e^4*F#_1^1{-1}*F_2_2{1}*d_{-3}{13}*d_{2}{23}*D_3(D#_1(d_{3}{13}))
*D_2(D#_3(d_{-2}{23}))*K4{3}*I{3^-2}*K4{2}*I{2^-2}*I{3^-2}*I{2^-2}
```

Where `I{3^-2}` is momentum in some power and `K4{3}` is some function of the momentum. Then according to [14] we use rules of supesymmetric covariant derivatives algebra and obtain final result.

```
-1*e^4*F#_1^1{-1}*F_1_2{1}*K4{3}*I{3^-2}*K4{2}*I{2^-4}*I{6^-2}
```

In the analytic form:

$$-e_0^4 \int \frac{d^4K}{(2\pi)^4}\frac{d^4L}{(2\pi)^4} d^4\theta\, \phi^*_\alpha(0,\theta)\phi_\alpha(0,\theta)\frac{1}{R_K K^4 R_L L^2 (K+L)^2}. \tag{3}$$

The program neglects some parts of the result as integrations over momenta and superspace. Expressions like (3) are parts of the final result that program produces. At the end it try to collect terms using some types of integral transformations. In the lowest orders it is usually enough but in higher ones sometimes collecting of terms is not full, and a lot of work to be done by hand. Making collecting of terms more efficient is an open problem.

The program was used to check results of the paper [17]. Also new result i.e. three-loop anomalous dimension of SQED regularized by higher derivatives was also obtained by it [18].

References

[1] J. A. Campbell and A. C. Hearn, J. Comput. Phys. **5** (1970), 280–327.

[2] P. Nogueira, J. Comput. Phys. **105** (1993), 279–289.

[3] J. Kublbeck, M. Bohm and A. Denner, Comput. Phys. Commun. **60** (1990), 165–180.

[4] T. Hahn, Comput. Phys. Commun. **140** (2001), 418–431.

[5] K. G. Chetyrkin and F. V. Tkachov, Nucl. Phys. B **192** (1981), 159–204.

[6] R. N. Lee, J. Phys. Conf. Ser. **523** (2014), 012059.

[7] I. Dubovyk, J. Gluza, T. Riemann and J. Usovitsch, PoS **LL2016** (2016), 034.

[8] D. Fontes and J. C. Romão, Comput. Phys. Commun. **256** (2020), 107311.

[9] F. Feng, Y. F. Xie, Q. C. Zhou and S. R. Tang, Comput. Phys. Commun. **265** (2021), 107982.

[10] M. Gerlach, F. Herren and M. Lang, Comput. Phys. Commun. **282** (2023), 108544.

[11] T. Hahn and C. Schappacher, Comput. Phys. Commun. **143** (2002), 54–68.

[12] T. Kreuzberger, W. Kummer and M. Schweda, Comput. Phys. Commun. **58** (1990), 89–104.

[13] A. F. Ferrari, Comput. Phys. Commun. **176** (2007), 334–346.

[14] I. E. Shirokov, Program. Comput. Software **49** (2023), 122–130.

[15] K. V. Stepanyantz, "Classical field theory," PHYSMATHLIT. (2009), 1–540, In Russian.

[16] N. N. Bogolyubov and D. V. Shirkov, "INTRODUCTION TO THE THEORY OF QUANTIZED FIELDS," Intersci. Monogr. Phys. Astron. **3** (1959), 1–720 1 [Moscow: Nauka, (1973) 416 p, In Russian].

[17] S. S. Aleshin, I. S. Durandina, D. S. Kolupaev, D. S. Korneev, M. D. Kuzmichev, N. P. Meshcheriakov, S. V. Novgorodtsev, I. A. Petrov, V. V. Shatalova and I. E. Shirokov, *et al.* Nucl. Phys. B **956** (2020), 115020.

[18] I. Shirokov and K. Stepanyantz, JHEP **04** (2022), 108.

On Tight and Efficient Bound Propagation For Neural Networks Based on Bernstein Polynomial Approximations

Min Wu

School of Computer Science and Software Engineering, East China Normal University, China

e-mail: mwu@sei.ecnu.edu.cn

Abstract

To Marko Petkovšek, in memoriam

Bound propagation is a critical step in wide range of Neural Network model checkers and reachability analysis tools. So far, linear and convex optimizations have been used to perform bound propagation, however, these methods suffer from introducing large errors due to the high non-convexness. In this work, we study several techniques on how to produce both tight and efficient bound propagation for Neural Networks based on Bernstein polynomial approximations.

Keywords: neural networks, bound propagation, Bernstein polynomials, interval arithmetic

Optimisation of Computer Algebra Techniques Application for Rician Data Analysis

T.V. Yakovleva

Federal Research Center "Computer Science and Control" of RAS, Russia

e-mail: tan-ya@bk.ru

Abstract

The paper presents a mathematical research directed on the optimization of the computer-algebra methods' application for solving the task of stochastic data analysis. Within the conducted theoretical investigation a few mathematical techniques of the statistical data analysis have been elaborated which allow essential simplifying of solving the task by computer algebra methods. The developed two-parameter approach to data analysis is efficiently applicable to a wide spectrum of scientific and applied tasks, in which the signal to be analyzed is described by the Rice statistical model.

Keywords: Rice distribution, data processing, two-parameter analysis, information technologies

1. Introduction

At the random signals processing, in particular, at handling the problem of noise suppression, recently an approach is being widely developed based on the methods of mathematical statistics such as the method of moments, the maximum likelihood method, etc. Obviously, at applying such an approach the peculiarities of the statistical distribution of the data being analyzed have a substantial significance for the possibility of the task solution. The present paper deals with the problem of simplifying the application of the computer algebra methods and decreasing the calculative resources for solving the task of two-parameter Rician signals processing. The so-called two-parameter methods provide the joint calculation of both the required signal value and the noise dispersion value.

The Rice statistical distribution is known to describe a wide range of information processing problems when the output signal is composed as a sum of the sought-for initial signal and a random noise generated by many independent normally-distributed summands of zero mean value. The variable to be measured and analyzed is an amplitude, or an envelope of the sum signal, while this value is known to obey the Rice statistical distribution, [1].

The so-called two-parameter approach to the Rician signals' analysis consists in solving the task of joint determination of both parameters of the Rice distribution. In contrast to the traditional one-parameter approximation this approach is free of limitations that are inherent to the one-parametric approximation based upon the supposition that one of the task statistical parameters — the noise dispersion — is known a priori, [2]–[4]. That's why the technique of the two-parametric task solution ensures much more correct estimation of the required values.

2. Theoretical aspects and numerical testing results

The task of joint computing of the Rice distribution's parameters allows efficient reconstruction of the informative component of the signal against the noise background. In [5]–[7] there has been developed an accurate theory of Rician signals statistical processing: new

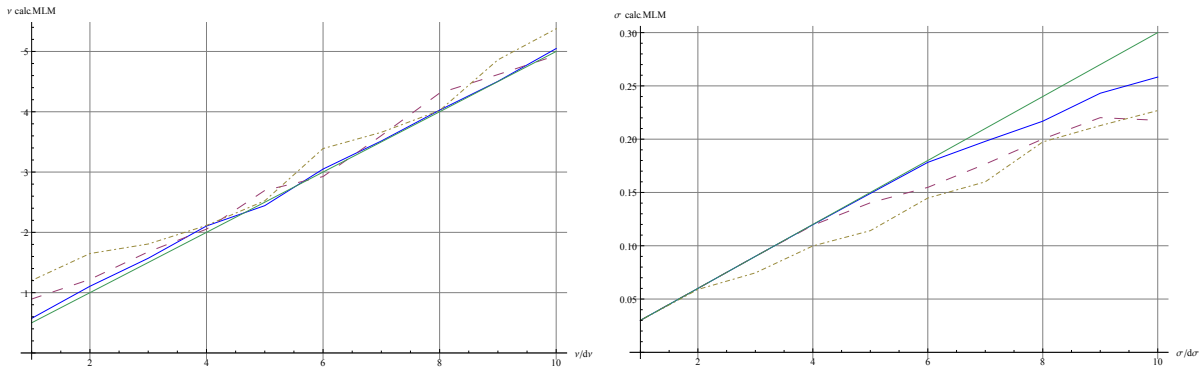
mathematical methods have been elaborated and strictly substantiated for solving the task of data analysis by means of joint signal and noise evaluation.

However this task is connected with finding solution of a system of two essentially nonlinear equations what is conjugated with considerable difficulties of both the theoretical and the computational character. On the other hand the Rician data analysis is especially needed in such areas which imply the necessity of the signal processing in systems with the priority of operation in a real time mode. In this connection it is important to simplify the algorithms of the Rician parameters joint computing and just these aspects of the problem have become the subject of the present paper.

As it has been shown (see, for example, [5]), in virtue of the specific peculiarities and nonlinear properties of the Rice statistical distribution, the Rician data analysis demands a development of the particular methods and the corresponding mathematical apparatus.

The particular theoretical methods having been developed within the two-parameter analysis of the Rician signal in [5]–[7] which differ in underlying statistical principles they are based upon. In the present paper the problem of simplifying the algorithms of joint signal and noise parameters computing is demonstrated by the example of using a method that presents a combination of the maximum likelihood technique and the method of moments. By means of application of this method for Rician data analysis the task of numerical solution of the problem connected with the necessity to solve a system of two essentially nonlinear equation for two variables has been reduced to solution of just one equation for one variabe. This obviously means an essential simplifying the applicability of the computer algebra means to solving the task as well as a significant decreasing of the required calculative resources.

Fig. 1 demonstrates the plots for the sought-for Rician parameters obtained by application of such a simplified algorithm at Wolfram Mathematica system. In Fig. 1(a) the solid, dashed and dot-dashed curves correspond to various values of standard deviation paramer σ, namely: $\sigma = 0.5$ (solid curve); $\sigma = 1.0$ (dashed line); $\sigma = 1.5$ (dot-dashed line). In Fig. 1(b) the solid, dashed and dot-dashed curves correspond to various values of the initial signal parameter ν, namely: $\nu = 3.0$, $\nu = 2.0$ and $\nu = 1.0$, correspondingly. The straight lines in both grapfs correspond to the initially determined value of the corresponding parameter. The sample lenght n in the presented variants of computing was $n = 4$ with the averaging over 25 samples (in real systems of digital signal processing the number of averaged samples is normally within $10^3 \div 10^4$) what ensures still more high accuracy at computing the Rician parameters, i.e. the informtive and the noise components of the signal.



(a) (b)

Figure 1: The results of computing the signal (a) and the noise (b) parameters in dependence of the signal-to-noise ratio SNR

The presented graphical illustration of the computing efficiency demonstrates the follow-

ing expected result: the precision of the calculated sought-for parameters noticeably decrease with the increase of the signal-to-noise ratio.

3. Conclusion

The joint computing of the Rice distribution's parameters ensures an efficient reconstruction of the informative component of the signal against the noise background. The presented mathematical research allows an essential decreasing of the needed calculating resources and the simplification of the numerical algorithms for Rician data analysis as solving the system of two essentially nonlinear equations has been mathematically reduced to solving just one equation for one unknown variable. Such an optimization of the task solution by computer algebra methods and the needed calculative resources decreasing mean enhancing the informative capacity and the precision of stochastic data processing, what in its turn make it possible to apply the elaborated techniques in the information technologies and systems with the priority of operation in a real-time mode.

The numerical results confirm the possibility of solving the problem of the Rician signals analysis by the developed methods ensuring a high precision in a wide range of the signal-to-noise ratio's values.

References

[1] *Rice S.O.* Mathematical analysis of random noise. Bell Syst. Technological J. 1944. Vol. 23. P. 282–332.

[2] *Benedict T.R., Soong T.T.* The joint estimation of signal and noise from the sum envelope. IEEE Trans. Inf. Theory. 1967. Vol. IT-13, N. 3. P. 447–454.

[3] *Talukdar K.K., Lawing W.D.* Estimation of the parameters of Rice distribution. J. Acoust. Soc. Amer. 1991. Vol. 89, N. 3. P. 1193–1197.

[4] *Sijbers J., den Dekker A.J., Scheunders P., Van Dyck D.* Maximum-Likelihood estimation of Rician distribution parameters. IEEE Transactions on Medical Imaging. 1998. Vol. 17, N. 3. P. 357–361.

[5] *Yakovleva T.V.* A theory of signal processing at the Rice distribution. Dorodnicyn Computing Centre, RAS, Moscow (2015).

[6] *Yakovleva T.* Peculiarities of the Rice statistical distribution: mathematical substantiation. Applied and Computational Mathematics. 2018. Vol. 7(4). P. 188–196. Science Publishing Group. DOI: 10.11648/j.acm.20180704.12.

[7] *Yakovleva T.* Study of accuracy of the signal reconstruction against the noise background by maximum likelihood technique at two-parameter analysis of Rician data. IEEE Proceedings of the 2022 International Conference on Information Technology and Nanotechnology (ITNT). 2022. P. 1–5. INSPEC Accession Number: 21992010, DOI: 10.1109/ITNT55410.2022.9848594.

Author index

Научное издание

КОМПЬЮТЕРНАЯ АЛГЕБРА
Материалы 5-й Международной конференции
Москва, 26–28 июня 2023 г.

Напечатано с готового оригинал-макета
Подписано в печать 01.06.2023.
Формат 60 × 84 1/16. Усл. печ. л. 6,7.
Тираж 150 экз. Заказ А-5.

ИПМ им. М.В. Келдыша
125047, Москва, Миусская пл., 4, http://www.keldysh.ru