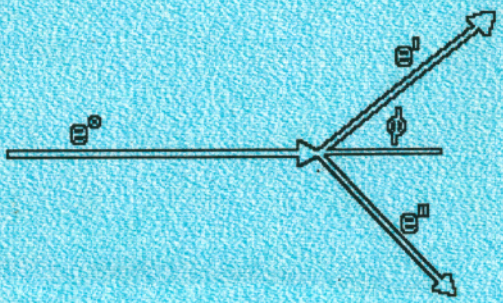
Proceedings of the International Workshop

Computer Algebra and its Application to Physics

CAAP-2001

Edited by V.P. Gerdt



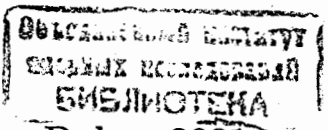
Joint Institute for Nuclear Research
Laboratory of Information Technologies

COMPUTER ALGEBRA AND ITS APPLICATION TO PHYSICS

CAAP-2001

Proceedings of the International Workshop
Dubna, Russia, June 28-30, 2001

Edited by V.P. Gerdt



Dubna 2001

УДК [004.8/9 + 519.6] : 539.12.01(063)
ББК [22.186 + 22.19 + 22.382.1] я431
С73

The contributions are reproduced directly from the originals presented by the Organizing Committee.

Computer Algebra and its Application to Physics: Proceedings
С73 of the International workshop (Dubna, June 28–30, 2001) / Editor V. P. Gerdt. — Dubna: JINR, 2002. — 359 p.: photo.

ISBN 5-85165-689-1

This volume contains papers and abstracts of talks presented at the workshop on computer algebra algorithms, systems and software packages and their application to physical and mathematical research and education. The collection consists of 34 full papers and 15 abstracts written by mathematicians, physicsts and computer scientists who develop and use computer algebra methods and tools.

The material of the book appeals to readers from postgraduate students to researchers and university teachers of computational mathematics, physics and computer science.

Компьютерная алгебра и ее приложения в физике: Труды международного рабочего совещания (Дубна, 28–30 июня 2001 г.) / Под ред. В. П. Гердта. — Дубна: ОИЯИ, 2002. — 359 с.: фото.

ISBN 5-85165-689-1

В книге собраны научные статьи и аннотации представленных на совещании докладов по алгоритмам, системам и пакетам программ компьютерной алгебры, их использованию в физических и математических исследованиях, а также в образовании. Сборник содержит 34 статьи и 15 аннотаций докладов физиков, математиков и программистов, разрабатывающих и применяющих методы и программы компьютерной алгебры.

Книга предназначена для широкого круга читателей — исследователей в области физики и компьютерной математики, а также преподавателей университетов и аспирантов.

УДК [004.8/9 + 519.6] : 539.12.01(063)
ББК [22.186 + 22.19 + 22.382.1] я431

ISBN 5-85165-689-1

Preface

The International workshop "Computer Algebra and its Application to Physics"/ CAAP-2001 took place at the Laboratory of Information Technologies of the Joint Institute for Nuclear Research (JINR) in Dubna, Russia, in June 28-30, 2001. This meeting was supported by the Russian Foundation for Basic Research and the Scientific Center for Applied Research in JINR and brought together more than 70 scientists and researchers from Byelorussia, Bulgaria, Georgia, Germany, Canada, Poland, Russia, South Africa, Slovenia and Ukraine. Forty nine reports were presented at the workshop and their written forms as full papers or abstracts are contained in this volume.

This meeting was the fifth in a series of workshops on computer algebra and its application to physics, held in Dubna in 1979, 1982, 1985 and 1990. The workshop provided a forum for researchers on computer algebra methods, algorithms and software and for those who use this tool in theoretical, mathematical and experimental physics, applied mathematics, engineering and education.

The CAAP-2001 workshop grew out of the recognition of the need for such a meeting based on the fact that, although there is a number of regular meetings of computer algebraists and users of computer algebra systems, those meetings do not pay so much attention to specific needs of physical sciences as they did in 70th and 80th. We believe that our workshop has helped to establish new links between algorithmic and software aspects of computer algebra and those from research problems in natural sciences, especially in physics.

It is my pleasure to acknowledge all contributors to these proceedings. Unfortunately, we did not receive full papers for a number of interesting talks, and some full papers were not accepted for the proceedings. For all these cases we decided to include abstracts of the talks.

Vladimir P. Gerdt

Contents

Minimal Multiplicative and Additive Decompositions of Hypergeometric Terms in One Variable

S. A. Abramov[a1], M. Petkovšek[b2]

[a] Computer Center of the Russian Academy of Science,
Vavilova 40, Moscow 117967 Russia;
e-mail: abramov@ccas.ru
[b]Department of Mathematics and Mechanics,
Faculty of Mathematics and Physics, University of Ljubljana,
Jadranska 19, 1000 Ljubljana, Slovenia;
e-mail: marko.petkovsek@fmf.uni-lj.si

In this talk we sum up our investigations [1, 2]. We describe a multiplicative normal form for rational functions which exhibits the shift structure of the factors, and investigate its properties. On the basis of this form we propose an algorithm which, given a rational function R, extracts a rational part U from the indefinite product of R: $\prod_{k=0}^{n} R(k) = U(n)\prod_{k=0}^{n} V(k)$, where the numerator and denominator of the rational function V have the lowest possible degrees. This gives a minimal representation (or a minimal multiplicative decomposition) of the hypergeometric term $\prod_{k=0}^{n} R(k)$. For example,

$$\prod_{k=0}^{n-1}\frac{(k+3)(2k+5)(3k+1)(4k+1)}{(k+1)(k+4)(2k+1)(3k+4)} = 4^n\frac{(n+1)(n+2)(2n+1)(2n+3)}{6(3n+1)}\prod_{k=0}^{n-1}\frac{k+\frac{1}{4}}{k+4}.$$

We also present an algorithm which, given a hypergeometric term $T(n)$, constructs hypergeometric terms $T_1(n)$ and $T_2(n)$ such that $T(n) = \Delta T_1(n) + T_2(n)$ $(\Delta T_1(n) = T_1(n+1) - T_1(n))$ and $T_2(n)$ is minimal in some sense (see example below). This solves the decomposition problem for indefinite sums of hypergeometric terms: $T_1(n+1) - T_1(n)$ is the "summable part" and $T_2(n)$ the "non-summable part" of $T(n)$. In other words, we get a minimal additive decomposition of the hypergeometric term $T(n)$. For example,

$$\left(-\frac{2n^2+3n+2}{n^2+n}\right)\prod_{k=0}^{n-1}\frac{1}{k+2} = \Delta\left(\frac{1}{n}\prod_{k=0}^{n-1}\frac{1}{k+1}\right) - \frac{1}{n}\prod_{k=0}^{n-1}\frac{1}{k+1},$$

where the minimal representation $\frac{1}{n}\prod_{k=0}^{n-1}\frac{1}{k+1}$ of "non-summable" part has the rational factor $\frac{1}{n}$ with the denominator of the lowest possible degree.

[1]Partially supported by the French-Russian Lyapunov Institute under grant 98-03.
[2]Partially supported by MZT RS under grant J2-8549.

References

[1] S. A. Abramov and M. Petkovšek, Canonical representations of hypergeometric terms, In *Proceedings FPSAC'01*, 2001. To appear.

[2] S. A. Abramov and M. Petkovšek, Minimal decomposition of indefinite hypergeometric sums, In *Proceedings ISSAC'01*, 2001. To appear.

On Some Algebraic Problems Arising in Quantum Mechanical Description of Biological Systems

M.V.Altaisky

Laboratory of Information Technologies,
Joint Institute for Nuclear Research, Dubna, 141980 Russia;
altaisky@mx.iki.rssi.ru

The biological hierarchy and the differences between living and non-living matter are considered from the standpoint of quantum mechanics. Starting from the Schrödinger question "How the life can be understood from the standpoint of quantum mechanics?", we analyze what algebraic constraints may be caused by a nontrivial "the part – the whole" relation, when the state of a part is constrained by the state of the whole.

References

[1] M.V.Altaisky, quant-ph/0007023.

Project "CalcPHEP: Calculus for Precision High Energy Physics"

D.Bardin, G.Passarino*, L.Kalinovskaya, P.Christova, A.Andonov, S.Bondarenko** and G.Nanava

Laboratory of Nuclear Problems, JINR
Joliot-Curie, 6, Dubna, Moscow region, Russia;
e-mail: bardin@nusun.jinr.ru
* Dipartimento di Fisica Teorica, Università di Torino, Italy
INFN, Sezione di Torino, Italy;
** Bogoluobov Laboratory of Theoretical Physics, JINR

Work supported by INTAS N°00-00313 and by the European Union under contract HPRN-CT-2000-00149.

1 Introduction

The CalcPHEP collaboration joins the efforts of several groups of theorists known very well in the field of theoretical support of various experiments in HEP, particularly at SLAC and LEP, (see, for instance [1], [2] and [3]). The first phase of the CalcPHEP system was realized in the site *http://brg.jinr.ru/* in 2000–2001. It is written mostly in FORM3, [4]. In this talk, we will describe the present status and our plans for the realization of next phases of the CalcPHEP project aimed at the theoretical support of experiments at modern and future accelerators: TEVATRON, LHC, electron Linear Colliders (LC's) i.e. TESLA, NLC, CLIC, and muon factories. Within this project, we are creating a four-level computer system which eventually must automatically calculate pseudo- and realistic observables for more and more complicated processes of elementary particle interactions, using the principle of knowledge storing. Upon completion of the second phase of the project, started January 2002 with duration of about three years, we plan to have a complete set of computer codes, accessible via an Internet-based environment and realizing the complete chain of calculations "from the Lagrangian to the realistic distributions" at the one-loop level precision including all $1 \to 2$ decays, $2 \to 2$ processes and certain classes of $2 \to 3$ processes.

1.1 CalcPHEP group

The CalcPHEP group was formed in 2001 in sector N°1 NEOVP LJAP.

During the first phase of the project in 2000–2001, the CalcPHEP group created the site brg.jinr.ru, where the development in two strategic directions is foreseen:

1. Creation of a softwear product, capable to compute HEP observables with one-loop precision for complicated processes of elementary particle interactions, using the principle of knowledge storing. Application: LHC.
2. Works towards two-loop precision level control of simple processes: $1 \to 2$, $1 \to 3$ and $2 \to 2$. Application: GigaZ option of electron LC's.

1.2 A little bit of history

There are two historical **sources** of `CalcPHEP` project:
1. From one side it roots back to many codes written by Dubna group aimed at a theoretical support of HEP experiments in the past:
1975 – 1986: support of CERN DIS experiments (BCDMS, EMC, NMC), creation of program `TERAD`; support of CERN neutrino experiments (CHARM-I, CDHSW and CHARM-II), creation of programs `NUDIS`, `INVMUD`, `NUFITTER`.
1983 – 1989: Foundation of the DZRCG — "Dubna–Zeuthen Radiative Correction Group", creation of EW library `DIZET`; creation of the program `ZBIZON` — the fore-runner of `ZFITTER` [5].
1989 – 1997: support of the DIS experiments at HERA, creation of the program `HECTOR`; participation in SMC experiment at CERN with the program *μela*.
1989 – 2001: Theoretical support of experiments at LEP, SLC (DELPHI, L3, ALEPH, OPAL and SLD).
2. From the other side, a monograph "The Standard Model in the Making" was written [6]. While working on the book, the authors wrote hundreds of "book-supporting" form-codes, which comprised the proto-type of future `CalcPHEP` system.

Like well known codes of LEP era: `TOPAZ0` [7], `ZFITTER` [5], *KK*MC [8], `CalcPHEP` is supposed to be a tool for precision calculations of pseudo- and realistic observables. Let's remind these definitions that arose in depth of LEP community:

Definition 1 *Realistic Observables are the (differential) cross-sections (more general event distributions) for a reaction, e.g.*

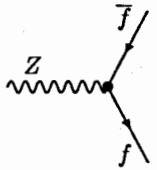
$$e^+e^- \to (\gamma, Z) \to f\bar{f}(n\gamma)$$

calculated with all available in the literature higher order corrections (QCD, EW), including real and virtual QED photonic corrections, possibly accounting for kinematical cuts.

Definition 2 *Pseudo-Observables are related to measured quantities by some de-convolution or unfolding procedure (e.g. undressing of QED corrections). The concept itself of pseudo-observability is rather difficult to define. One way say that the experiments measure some primordial distributions which are then reduced to secondary quantities under some set of specific assumptions (definitions).*

Z decay partial width represents typical example of pseudo-observables, i.e. it has to be *defined*. At the tree level, we define it as a quantity described by the square of one

diagram:



2 LEP, Precision High Energy Physics and its Future

One may say that during recent years a new physical discipline was born. We call it PHEP, Precision High Energy Physics. Experimentally, it finally shaped in the result of glorious 12 year LEP era: measurements at Z resonance in **1989 − 1995**, and reaching an unprecedented experimental accuracy $\leq 10^{-3}$, and measurements above Z resonance in **1995 − 2000**, at higher energies, where high enough experimental accuracy was also reached $\leq 1\%$. By 2/11/2000 LEP2 possibly saw hints of "God blessed" particle — Higgs boson, but was stopped, unfortunately, mainly due to lack of financing.

For the first time huge HEP facility challenged for theoreticians to perform calculations with uncertainty better than experimental errors of $\mathcal{O}(10^{-3})$ and, eventually, efforts of many groups of theoreticians allowed the achievement of the theoretical precision of the order $2.5 \cdot 10^{-4}$ at the Z resonance and $2 - 3 \cdot 10^{-4}$ at LEP2 energies.

This, in turn, greatly contributed to the success of precision tests of the SM, the main result of LEP era, which laid the foundation of the Precision High Energy Physics. This is why our project got this suffix PHEP.

2.1 Future of PHEP

PHEP has good perspectives and after the end of LEP. Several Input parameters of the Standard Model (SM) are expected to be improved in near future.

Recent discrepancy in the muon amm:

$$\begin{aligned} a_\mu^{SM} &= 116591661(114) \times 10^{-11} \\ a_\mu^{EXP}(Average) &= 116592023(151) \\ a_\mu^{EXP} - a_\mu^{SM} &= 362(189) \quad 2\sigma \text{ difference} \\ & \text{exp. error (151) should be improved soon up to} \sim (50), \end{aligned} \tag{1}$$

necessitates an improvement of the knowledge of the hadronic contribution of $\Delta\alpha_h^{(5)}\left(M_z^2\right)$ to the running e.m. coupling. An experimental input for $\sigma(e^+e^- \to \text{hadrons})$ at cms energies (1-4 GeV) is expected from BES-II, BEPC (Beijing), VEPP2000 (Novossibirsk) and DAFNE at cms energies around ϕ-meson.

Very important should be projected improvements of mass measurements: M_W, m_t.

LEP1 finished with *indirect* result for the top mass: $m_t = 169^{+10}_{-8}$ GeV; while LEP1 $\oplus$ TEVATRON constraint yielded: $m_t = 174.5^{+4.4}_{-4.2}$ GeV.

LEP2 reached for W mass M_W: $M_W = 80.450 \pm 0.039$, in the *direct* measurements and $M_W = 80.373 \pm 0.023$ as *indirect* result.

TEVATRON in RUN-I reached: $M_W = 80.454 \pm 0.060$ GeV, $m_t = 174.3 \pm 5.1$ GeV.

Much better precision tags are expected to be reached at TEVATRON, RUN-II (recently started): $\Delta M_W \sim 20$MeV, $\Delta m_t \sim 2$ GeV; and later at LHC (not so sooner than in 2006, however): $\Delta M_W \sim 15$ MeV, $\Delta m_t \sim 1$ GeV.

Where, when and with which mass Higgs boson might be discovered?
– TEVATRON has a serious chance to see Higgs up to mass 180 GeV; however it will require very high integrated luminosity: $\int \mathcal{L} \geq 5 fb^{-1}$;
– LHC, will cover all allowed mass range up to 500 GeV (not so soon, after 2007);
– LC's and muon factories (after 2010–2012).

New horizon of PHEP will be opened with experiments at electron LC's: TESLA (DESY) particularly with GigaZ option, i.e. coming back to Z resonance with statistics 10^9; CLIC (CERN); JLC (KEK), NLC(SLAC, LNBL, LLNL, FNAL) and Muon Storage Rings (Higgs Factory) — all that more than in ten years from now.

One expects fantastic precision tags there in:
– $\Delta \sin^2 \theta_{eff} \sim 0.00002$;
– $\Delta M_W \sim 6$ MeV, $\Delta m_t \sim 100 - 200$MeV;
– $\Delta M_H \sim 100$ MeV (from $e^+e^- \to ZH$);
– and detail study of Higgs boson properties.

Given our LEP1 experience one should definitely state that ***2-loop precision level control will be absolutely necessary for the analysis of these data!***

One may conclude that PHEP has a bright future: all future colliders — TEVATRON, LHC, electron LC's (TESLA, NLC, CLIC) and muon factories will be, actually, PHEP facilities! For data analysis, they will surely require qualitatively new level of both theoretical predictions and principally new computer codes.

3 Necessary notion

In order to understand the language of `CalcPHEP` one has to introduce many notions and notations.

3.1 Input Parameter Set, IPS

The Minimal Standard Model (MSM), contains large number of Input Parameters:
$25 = 2$ interaction constants α and α_s
$\oplus$ 8 mixing angles (CKM and possible lepton analogs)
$\oplus$ 15 masses (12 fundamental fermions and 3 fundamental bosons Z, W, H).

However, the number 25 is *minimal.* MSM is **unable** to compute its IPS from first principles; MSM is **able** to compute *any observable* $O_i^{\rm exp}$ in terms of its IPS:

$$O_i^{\rm exp}\,(\text{measured}) \quad \leftrightarrow \quad O_i^{\rm theor}\,(\text{calculated, as a function of IPS})\,. \tag{2}$$

This is the way how precision measurements set *constraints* on IPS.

3.1.1 Number of free parameters in fits of Z resonance observables

At Z resonance, not all 25 parameters matter. Actually only 5 parameters:

$$\Delta\alpha_h^{(5)}\left(M_Z^2\right), \qquad \alpha_s\left(M_Z^2\right), \qquad m_t\,, \qquad M_Z\,, \qquad M_H\,, \tag{3}$$

which we call the Standard LEP1 IPS, matter.

Using M_Z, measured at Z peak itself with the precision $\sim 2 \times 10^{-5}$, and also reach information from the other measurements for:

$$\alpha_s\left(M_z^2\right), \qquad m_t, \qquad M_W, \tag{4}$$

we *approach* one-parameter fit, with Higgs boson mass M_H being the only fitted parameter. The result of such a fit was shown in the *Blue band* figure, the most celebrated LEP era figure, derived with the aid of TOPAZO [7] and ZFITTER [6] codes.

3.2 Quantum Fields of the SM

Here we sketch all fundamental quantum fields of the SM in one of the most general gauges — R_ξ, with three arbitrary gauge parameters ξ_A, ξ_Z, ξ.

Three generation of fermions or matter fields:

$$\longrightarrow \quad f \;=\; \left\{ \begin{array}{lllll} \begin{pmatrix} \nu \\ l \end{pmatrix} & = & \begin{pmatrix} \nu_e \\ e^- \end{pmatrix} & \begin{pmatrix} \nu_\mu \\ \mu \end{pmatrix} & \begin{pmatrix} \nu_\tau \\ \tau \end{pmatrix} \\ \begin{pmatrix} U \\ D \end{pmatrix} & = & \begin{pmatrix} u \\ d \end{pmatrix} & \begin{pmatrix} c \\ s \end{pmatrix} & \begin{pmatrix} t \\ b \end{pmatrix} \end{array} \right.$$

possess masses, m_f, charges, Q_f, and third projections of weak isospin, $I_f^{(3)}$:

$$m_f, \quad Q_f = \begin{pmatrix} \nu & l & U & D \\ 0 & -1 & +\frac{2}{3} & -\frac{1}{3} \end{pmatrix}, \qquad I_f^{(3)} = \begin{pmatrix} \nu & l & U & D \\ +\frac{1}{2} & -\frac{1}{2} & +\frac{1}{2} & -\frac{1}{2} \end{pmatrix}.$$

Gauge fields:

Vector bosons	Unphysical scalars	Faddeev–Popov ghosts
A		Y^A
$Z\,(M_z)$	ϕ^0	Y^z
$W^\pm(M_w)$	$\phi^\pm$	$X^\pm$
Gluon possesses strong interaction		
g		Y^a

possess physical charges and physical masses

possess physical charges and unphysical masses and unphysical charges.

Higgs field:

-------- $H\,(M_H)$ is a scalar, neutral, massive field.

3.2.1 The Lagrangian in R_ξ gauge, Feynman Rules

At the ground level of CalcPHEP system one has this Lagrangian

$$\mathcal{L} = \mathcal{L}(\text{IPS of 25 parameters, 17 fields, 3 gauge parameters}), \tag{5}$$

from which one derives *primary Feynman rules for vertices.*

3.2.2 Propagators in R_ξ gauge

Here we list propagators in R_ξ gauge, the other important bricks of CalcPHEP system.
Propagator of a fermion, f:

$$\frac{-i\not{p} + m_f}{p^2 + m_f^2}$$

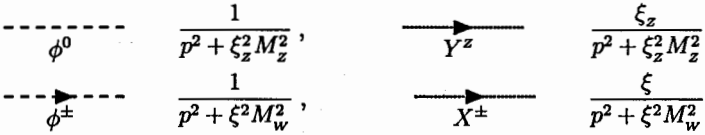
Vector boson propagators:

$$A \qquad \frac{1}{p^2}\left\{\delta_{\mu\nu} + \left(\xi_A^2 - 1\right)\frac{p_\mu p_\nu}{p^2}\right\}$$

$$Z \qquad \frac{1}{p^2 + M_z^2}\left\{\delta_{\mu\nu} + \left(\xi_z^2 - 1\right)\frac{p_\mu p_\nu}{p^2 + \xi_z^2 M_z^2}\right\}$$

$$W^\pm \qquad \frac{1}{p^2 + M_w^2}\left\{\delta_{\mu\nu} + \left(\xi^2 - 1\right)\frac{p_\mu p_\nu}{p^2 + \xi^2 M_w^2}\right\}$$

Propagators of unphysical fields:



$$\phi^0 \qquad \frac{1}{p^2 + \xi_z^2 M_z^2}, \qquad Y^A \qquad \frac{\xi_A}{p^2}$$

$$\phi^\pm \qquad \frac{1}{p^2 + \xi^2 M_w^2}, \qquad Y^z \qquad \frac{\xi_z}{p^2 + \xi_z^2 M_z^2}$$

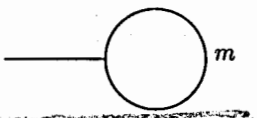
$$X^\pm \qquad \frac{\xi}{p^2 + \xi^2 M_w^2}$$

Propagator of the physical scalar field, H-boson

$$H \qquad \frac{1}{p^2 + M_H^2}$$

3.3 Scalar A_0, B_0, etc functions

For calculation of one-loop integrals CalcPHEP uses the standard scalar A_0, B_0, C_0 and D_0 functions [6].
One-point integrals or A_0 functions, are met in tadpoles diagrams:



We give its defining expression:

$$i\pi^2 A_0(m) = \mu^{4-n} \int d^n q \frac{1}{q^2 + m^2 - i\epsilon}, \tag{6}$$

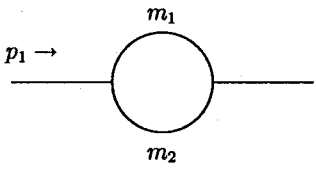
and the answer in the *dimensional regularization*:

$$A_0(m) = m^2 \left(-\frac{1}{\bar{\varepsilon}} - 1 + \ln \frac{m^2}{\mu^2} \right) + \mathcal{O}(\varepsilon), \tag{7}$$

where the ultraviolet pole is:

$$\frac{1}{\bar{\varepsilon}} = \frac{2}{\varepsilon} - \gamma - \ln \pi, \qquad n = 4 - \varepsilon. \tag{8}$$

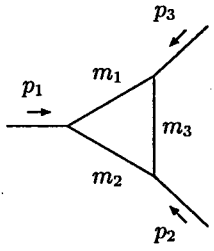
Two-point integrals or B_0-functions are met in self-energy diagrams:



We limit ourselves by giving its defining expression:

$$\begin{aligned} &i\pi^2 B_0\left(p_1^2; m_1, m_2\right) = \mu^{4-n} \int d^n q \frac{1}{d_0 d_1}, \\ &d_0 = q^2 + m_1^2 - i\epsilon, \qquad d_1 = (q + p_1)^2 + m_2^2 - i\epsilon. \end{aligned} \tag{9}$$

Three-point integrals, C functions, are met in vertices:



Its defining expression reads:

$$i\pi^2 C_0\left(p_1^2, p_2^2, Q^2; m_1, m_2, m_3\right) = \mu^{4-n} \int d^n q \frac{1}{d_0 d_1 d_2}, \tag{10}$$

$$d_0 = q^2 + m_1^2 - i\epsilon, \quad d_1 = (q + p_1)^2 + m_2^2 - i\epsilon, \quad d_2 = (q + p_1 + p_2)^2 + m_3^2 - i\epsilon, \tag{11}$$

where $Q^2 = (p_1 + p_2)^2$ is one of the Mandelstamm variables: s, t or u.
Four-point integrals, D-functions, are met in boxes.

Presently, `CalcPHEP` knows ALL about reduction of up to four-point functions up to third rank tensors and of the so-called *special* functions, which are due to peculiar form of the photonic propagator in the R_ξ gauge, see [6].

3.4 Processes in the SM

One should be aware of a hierarchical classification of processes accepted in CalcPHEP and of a relevant notion of *independent structures*, or *independent amplitude form factors*, which number is deeply related to the number of *independent helicity amplitudes* by which a process may be described (below we present these numbers for unpolarized cases).

3.4.1 Decays $1 \to 2$

There are $B \to f\bar{f}$ and $3B$ decays:

- $H \to f\bar{f}$ (one structure)
- $Z \to f\bar{f}$, $(\gamma \to f\bar{f})$ (three structures)
- $W \to f\bar{f}'$, $(t \to W^+b)$ (four structures)
- $H \to ZZ$, W^+W^-
- $Z \to W^+W^-$

3.4.2 Processes $2 \to 2$

There are $2f \to 2f$ processes, which in turn are subdivided into Neutral Current (NC) and Charged Current (CC) ones:

- NC: $f\bar{f} \to (\gamma, Z, H) \to f'\bar{f}'$ (4,6) 10 structures depending on whether initial and final state fermion masses are ignored
- CC: $f_1\bar{f}_2 \to (W) \to f_3\bar{f}_4$

Next, there are many processes of a kind $Vf \to f'V'$, in particular

- compton-effect: $\gamma e \to \gamma e$, $Z \to f\bar{f}\gamma$
- $e^+e^- \to W^+W^-$, ZZ, $Z\gamma, \gamma\gamma$

Decays $1 \to 3$ are cross-channels of the previous processes and their one-loop description in terms of independent objects, mentioned above, one gets for free. Present level of CalcPHEP has a lot of preparations for all above processes, but far not all is put into the working areas of the site brg.jinr.ru.

3.4.3 Processes $2 \to 3$

They comprise a very reach family, for instance:

- $e^+e^- \to (\gamma, Z, H) \to f\bar{f}\gamma$.

Their implementation is one of main goals of the second phase of CalcPHEP project. Corresponding decays $1 \to 4$ are again cross-channels of the previous processes and need not be studied separately.

3.4.4 Processes $2 \to 4$

To this family belongs 4 fermion processes of LEP2. Their study is not foreseen at the second phase of CalcPHEP project, but might be a subject of its third phase.

4 Building Blocks and knowledge storing

4.1 Simplest decay: $Z \to f\overline{f}$

4.1.1 Amplitude of $Z \to f\overline{f}$ decay at tree level

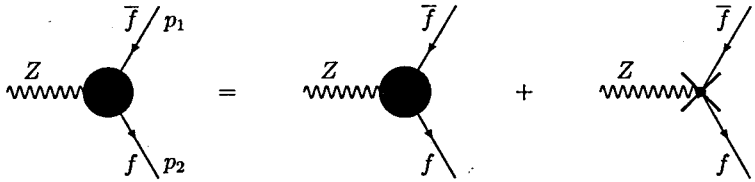
Its tree level diagram was already presented at the end of Section 1.2; the corresponding amplitude reads:

$$V_\mu^{Zf\overline{f}} = (2\pi)^4 i \frac{ig}{2c_w} \gamma_\mu \left[I_f^{(3)}(1+\gamma_5) - 2Q_f s_w^2 \right], \tag{12}$$

with vector and axial coupling constants: $v_f = I_f^{(3)} - 2Q_f s_w^2$, $a_f = I_f^{(3)}$. Note appearance of the two structures in Eq. (12), which might be termed as L and Q structures, correspondingly. Note also, that Eq. (12) as well as all below, are written in Pauli metrics that is used by CalcPHEP.

4.1.2 Amplitude of $Z \to f\overline{f}$ decay with loop corrections

It might be schematically depicted as a sum of one-loop vertices and counter terms:



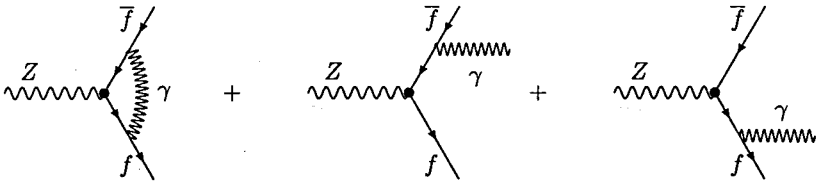
In the most general case (but for unpolarized study) the one-loop amplitude may be parameterized by the three *scalar form factors*:

$$V_\mu^{Zf\overline{f}} = (2\pi)^4 i \frac{g^3}{16\pi^2 2c_w} \gamma_\mu \left[iI_f^{(3)} F_L \gamma_+ - 2iQ_f s_w^2 F_Q + m_f (p_1 - p_2) F_D \right]. \tag{13}$$

Given similarity of Eqs. (12) and (13), the latter is called sometimes Improved Born Approximation (IBA) amplitude.

4.1.3 QED diagrams and corrections

The QED diagrams comprise gauge invariant subsets, this is why they are considered sometimes separately:



Their contribution to the partial Z widths, in the case when no photon cuts are imposed, reads:

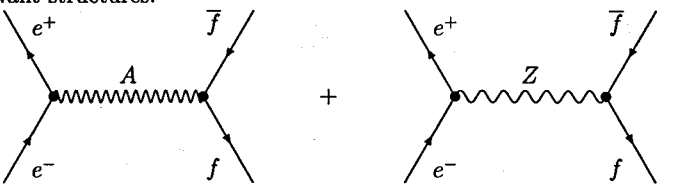
$$\Gamma_f^{\rm QED} = \Gamma_f \left(1 + \frac{3\alpha}{4\pi} Q_f^2\right). \tag{14}$$

4.2 Process $e^+e^- \to f\overline{f}$

Coming to a more complicated case of a $2f \to 2f$ process, we will illustrate how building blocks, derived for a study of a lower level process, might be use at a higher level.

4.2.1 Tree-level diagrams and amplitudes of $e^+e^- \to f\overline{f}$

Consider first the two tree-level diagrams with γ and Z exchanges in order to introduce basis of relevant structures.



$$\begin{aligned}
\mathcal{A}_\gamma^{\rm Born} &= \frac{e^2 Q_e Q_f}{s} \gamma_\mu \otimes \gamma_\mu\,, \\
\mathcal{A}_z^{\rm Born} &= \frac{e^2}{4 s_w^2 c_w^2} \chi_z(s)\, \gamma_\mu (v_e + a_e\gamma_5) \otimes \gamma_\mu (v_f + a_f \gamma_5) \\
&= \frac{e^2}{4 s_w^2 c_w^2} \chi_z(s)\, \gamma_\mu \left[I_e^{(3)}\gamma_+ - 2Q_e s_w^2\right] \otimes \gamma_\mu \left[I_f^{(3)}\gamma_+ - 2Q_f s_w^2\right],
\end{aligned} \tag{15}$$

where $\gamma_\pm = 1 \pm \gamma_5$ and symbol $\otimes$ stands for a short-hand notation

$$\gamma_\mu (v_1 + a_1\gamma_5) \otimes \gamma_\nu (v_2 + a_2\gamma_5) \;=\; \bar v(p_+)\, \gamma_\mu (v_1 + a_1\gamma_5)\, u(p_-)\, \bar u(q_-)\, \gamma_\nu (v_2 + a_2\gamma_5)\, v(q_+)$$

and

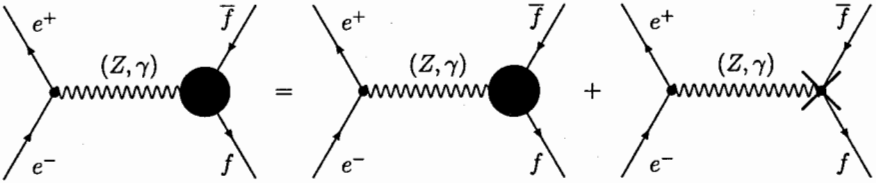
$$\chi_z(s) = \frac{1}{s - M_z^2 + i s \Gamma_z / M_z}\,. \tag{16}$$

This amplitude is characterized by four structures:

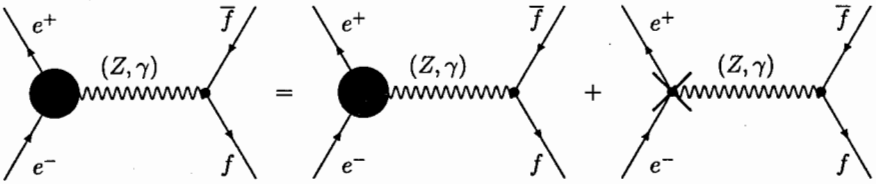
$$LL = \gamma_\mu\gamma_+ \otimes \gamma_\mu\gamma_+\,, \quad LQ = \gamma_\mu\gamma_+ \otimes \gamma_\mu\,, \quad QL = \gamma_\mu \otimes \gamma_\mu\gamma_+\,, \quad QQ = \gamma_\mu \otimes \gamma_\mu\,. \tag{17}$$

4.2.2 One-loop amplitude for $e^+e^- \to f\overline{f}$

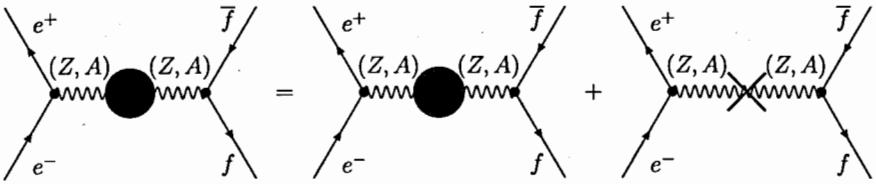
"Dressed" with one-loop vertices and counterterms, the γ and Z exchanges may be symbolically depicted as:



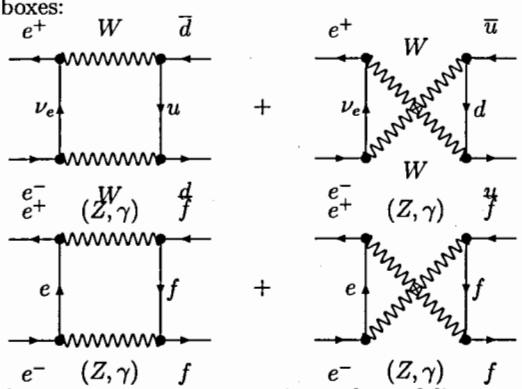
And similarly for the initial state vertex:



Where one can easily recognize building blocks already known from the calculation of one-loop radiative corrections for Z decay, however, now we need to dress $\gamma \to f\overline{f}$ vertex too and add into consideration "dressing" of propagators:



To complete calculations of one-loop EWRC for the process $e^+e^- \to f\overline{f}$ one should add WW and ZZ boxes:



ZZ, $\gamma\gamma$ and $Z\gamma$ boxes comprise gauge invariant subset of diagrams. Moreover, $\gamma\gamma$ and $Z\gamma$ boxes QED, vertices and QED bremsstrahlung for NC $2f \to 2f$ processes often are separated into a gauge invariant QED subset of diagrams.

Virtual QED one-loop diagrams together with four QED bremsstrahlung diagrams form an Infra-Red Divergence (IRD) free subset.

This example clearly shows how the principle of knowledge storing is implemented within CalcPHEP project: one starts from the simplest decays and collects all relevant building blocks, BB's (off-shell with respect to boson mass). Then one moves to next level of complexity where all BB's computed at the previous level are requested, but on top one needs more complicated objects (here boxes).

This strategy was realized in our recent calculations of the EWRC to the $e^+e^- \to f\bar{f}$ process, which are completely done with the aid of CalcPHEP system [9]. There is another study accomplished with CalcPHEP [10].

5 Status of the project

Before discussing what is already available at the site brg.jinr.ru, we present some general information about CalcPHEP system.

5.1 Basic information about CalcPHEP, keywords

CalcPHEP is **four-level computer system** for automatic calculation of pseudo- and realistic observables (decay rates, event distributions) for more and more complicated processes of elementary particle interactions, using the principle of knowledge storing.

At each of the four levels there are:

1. Codes (written in FORM3), realizing full chain of analytic calculations from the SM Lagrangian $\mathcal{L}_{SM}$ to the Ultra Violet Free Amplitudes, UVFA, parameterized by a minimal set of scalar form factors;
2. Codes (written in FORM3), realizing analytic calculations of a minimal subset of Helicity Amplitudes, HA's, followed by an automatic procedure of generation of codes for numerical calculations of HA's (presently FORTRAN codes, and in a near perspective C^{++} codes).
3. Codes, realizing the so-called "infrared rearrangement" of HA's. This is needed if the multiple photon emission is being exponentiated at the amplitude level. Currently, bremsstrahlung photons are added in the lowest order and the third level is skipped.
4. Codes, that use HA's derived at the second (or third) level together with tree-level HA's for one-photon (or multiple-photon) emission, within a Monte Carlo event generator, which is supposed to compute realistic distributions (presently FORTRAN codes, and in a near perspective C^{++} codes.)

It is an **Internet based** and **Database based** system. The latter means that there is a storage of source codes written in different languages, which talk to each other. They are placed into a homogeneous environment written in JAVA.

It follows **Intermediate access principle** i.e. full chain "from the Lagrangian to realistic distribution" should work out completely in real time, if someone requests this, however, it is supposed to have several "entries", say after each level, or just providing the user with its final product — a Monte Carlo event generator.

5.2 Some technical data about CalcPHEP

1. Address *http://brg.jinr.ru/*
2. For realization of the site one used:
   - Apache web server under Linux,
   - form3 compiler,
   - mySQL server for relational databases.
3. In the current version, user-interface is realized with the use of PHP.
4. Nowadays, everything is being rewritten in JAVA in order to reach better "interactivity" and to use reach possibilities of already written in this language libraries. **Main goal of this rewriting is to create a homogeneous environment both for accessing our codes from the database and for offering a possibility for simultaneous work of several members of the group and external users.**

5.3 Present and nearest versions of CalcPHEP system

In 2001, we released two test-versions of CalcPHEP:

1. **v0.01 from March'01** realizes analytic calculations of one-loop UVFA for decays $1 \to 2$ (level-1).
2. **v0.02 from September'01** returns numbers for one-loop decay widths (levels-1,2) via temporary bypass of level 4. It realizes also levels-1,2 for $2f \to 2f$ NC process.
3. One has very many almost finished "preparations" for the other processes $2 \to 2$ and decays $1 \to 3$ (level-1). All this should comprise **v0.03** of Summer 2002.
4. An active work is being realized on implementation of level-4 for decays $1 \to 2$, this should complete full chain "from the SM Lagrangian to pseudo-observables" for the simplest decays.
5. There are many problems to be solved at the second or later phases of the project. Among them one should mention:
   - automatic generation of Feynman Rules from a Lagrangian,
   - automatic generation of topologies of Feynman diagrams,
   - graphical representation of the results.

6 Conclusion

At a Symposium in honor of Professor Alberto Sirlin's 70th Birthday was said: *A new frontier is as the horizon: most likely it is goodbye to the one man show. Running a new Radiative Correction project will be a little like running an experiment [11].*

Indeed, projects of such a kind as CalcPHEP are definitely long term projects. Remember, that ZFITTER took about 12 years, about the same time exists already FeynArts [12].

Our nearest goal is the realization of the second phase of the project upon completion of which we plan to have a complete software product, accessible via an Internet-based environment, and realizing the chain of calculations "from the Lagrangian to the realistic

distributions" at the one-loop level precision including some processes $2 \to 3$ and decays $1 \to 4$. Plans also assume to perform an *R&D* for the third phase of the project (see also [13]–[16]) which should begin in 2004.

Second phase is basically oriented on a common work of theoreticians of the Dubna group and the Knoxville–Krakow collaboration [8].

United group proposes to realize in 2002-2004 an important phase of CalcPHEP project: oriented toward a merger of analytic results to be produced by Dubna team with MC event generators to be developed by Knoxville–Krakow collaboration[1].

Among most important milestones of first year, one should mention: realization of the levels 2-4 for the simplest $Z(H,W) \to f\bar{f}$ decays; completion of level 1 for the radiative Z decay, $Z \to f\bar{f}\gamma$, work on which is already under way; completion of levels 2-4 for the radiative Z decay.

References

[1] D. Bardin, G. Passarino, and W. Hollik (eds.), "Reports of the working group on precision calculations for the Z resonance", CERN Report, 95–03 (1995).

[2] D. Bardin, M. Gruenewald and G. Passarino, "Precision calculation project report", February 1999, hep-ph/9902452.

[3] M.Kobel, Z. Was, C.Ainsley, A.Arbuzov, S.Arcelli, D.Bardin, I.Boyko, D.Bourilkov, P.Christova, J.Fujimoto, M.Grunewald, T.Ishikawa, M.Jack, S.Jadach, L.Kalinovskaya, Y.Kurihara, A.Leike, R.Mcpherson, M.-N.Minard, G.Montagna, M.Moretti, T.Munehisa, O.Nicrosini, A.Olchevski, F.Piccinini, B.Pietrzyk, W.Placzek, S.Riemann, T.Riemann, G.Taylor, Y.Shimizu, M.Skrzypek, S.Spagnolo, and B.Ward, "Two fermion production in electron positron collisions", in Proc. of LEP2 Monte-Carlo Workshop, Geneva, Switzerland, 1999-2000, report hep-ph/0007180 (G.Passarino, R.Pittau, and S.Jadach, eds.), vol.1, pp.1–113, 2000.

[4] J. Vermaseren, "New features of form", math-ph/0010025.

[5] D. Bardin, M. Bilenky, P. Christova, M. Jack, L. Kalinovskaya, A. Olchevski, S. Riemann, and T. Riemann, *Comput. Phys. Commun.* **133** (2001) 229.

[6] D. Bardin and G. Passarino, "The standard model in the making: Precision study of the electroweak interactions", Oxford, UK: Clarendon, 1999, 685p.

[7] G. Montagna, O. Nicrosini, F. Piccinini and G. Passarino, *Comput. Phys. Commun.* **117** (1999) 278.

[8] S. Jadach, Z. Was and B.F.L. Ward, *Comput. Phys. Commun.* **124** (2000) 233; *Comput. Phys. Commun.* **130** (2000) 260.

[1] In this connection it is necessary to emphasize that any future code aimed at a comparison of experimental data with theory predictions should be a MC generator, since the processes at very high energies will have multi-particle final states that make impossible a semi-analytic approach used at LEP within ZFITTER project.

[9] D. Bardin and L. Kalinovskaya and G. Nanava, "An electroweak library for the calculation of EWRC to $e^+e^- \to f\bar{f}$ within the `CalcPHEP` project", `hep-ph/0012080`; revised version CERN-TH/308-2000, November 2001.

[10] D. Bardin, P. Christova, L. Kalinovskaya and G. Passarino, "Atomic Parity Violation and Precision Physics", `hep-ph/0102233`, *Eur.Phys.J.* **C22** (2001) 99.

[11] Giampiero Passarino, "Precision Physics Near LEP Shutdown and Evolutionary Developments" Presented at 50 Years of Electroweak Physics A symposium in honor of Professor Alberto Sirlin's 70th Birthday October 27-28, 2000, `hep-ph/0101299`.

[12] T. Hahn, *Comput. Phys. Commun.* **140** (2001) 418.

[13] Dmitri Bardin, "12 years of precision calculations for LEP. What's next?", Presented at 50 Years of Electroweak Physics A symposium in honor of Professor Alberto Sirlin's 70th Birthday October 27-28, 2000, `hep-ph/0101295`.

[14] D. Yu. Bardin and L. V. Kalinovskaya and F. V. Tkachov, "New algebraic-numeric methods for loop integrals: Some 1-loop experience", QFTHEP-2000, Tver, September 2000. Eds.: V.I.Savrin and B.B.Levtchenko, SINP MSU, Moscow, 2001, `hep-ph/0012209`.

[15] G. Passarino, S. Uccirati, "Algebraic-numerical evaluation of Feynman diagrams: Two-loop self-energies", `hep-ph/0112004`.

[16] G. Passarino, *Nucl.Phys.* **B619** (2001) 257, `hep-ph/0108252`.

The Approximation of Some Physical Processes by Exponential Functions

T. Bochorishvili, E. A. Grebenikov

University of Podlasie, Siedlce, Poland;
e-mail: tengiz@ap.siedlce.pl

The problem of the separation of radioactive substances from radioactive mixture is connected with data processing obtained from the experimental measurements. The rules of decomposition of the radioactive chemical elements are described by exponential functions. It is natural that the problem of the best approximation of a finite set of measurements by the exponential functions is adequate, which fundamental parameters are half-life of unknown components of the radioactive mixture.

In 60-th XX century the known American mathematician C. Lanczos pointed in [1] two problems of great practical importance:

Problem 1. Find hidden periodicities in the polyharmonic processes, given in big sets of measurements.

Problem 2. It is necessary to determine the hidden exponents in processes of the radioactive disintegration, represented by massive measurements.

Both of these problems are related to so called "ill posed problems" [2] and hence if solutions exist, there are several solutions.

The functional, that one has to minimize for determining the unknown parameters, is transcendental. More ever, the number of parameters also is unknown, which complicate one more the problem. C. Lanczos has solved both of these problems for the simplest models: the measurements are realized on equidistant time grid and the number of parameters is known [1].

The first problem for non-uniform division of time and with unknown number of parameters has been solved by E.A. Grebenikov and S.V. Mironov[3]. They constructed the so called method of two dimensional iteration, that has been successfully used in problems of the cosmic dynamics.

In our paper is proposed a new algorithm of the pick-out of the exponential functions, based on the approximation discreet experimental measurements by exponent polynomials. It is possible to realize this algorithm on the non-equidistant time grid.

Let be done N points

$$(t_k, y_k) \in R^2, \quad 0 \leq t_k \leq T, \quad (k = 0, 1, 2, ..., N).$$

We are looking for a function $f(t)$, $t \in [0, T]$, the graph of which contains these points. More ever, this function must be of the form:

$$f(t) = \sum_{k=1}^{m} \alpha_k e^{\lambda_k t}, \quad t \in [0, T], \tag{1}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_m)$, $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_m)$ the natural number m are to be determined, usually m is much less then N, i.e. $m << N$. More ever, in order to solve this problem it is necessary to determine the lower and upper bound of m $(m_1 \leq m \leq m_2)$.

To find the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$, first of all we fix the natural number $m = m_0$. In order to use the method of least squares, we must construct the functional

$$\Psi(\boldsymbol{\alpha}, \boldsymbol{\lambda}, f) = \sum_{k=0}^{N} \left(y_k - \sum_{i=1}^{m_0} \alpha_i e^{\lambda_i t_k} \right)^2 \tag{2}$$

and we have to find the minimum of that

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\lambda}} \Psi(\boldsymbol{\alpha}, \boldsymbol{\lambda}, f) \longmapsto 0. \tag{3}$$

Like this found solutions are supposed roots and require laborious analyses.

The problem (3) is referred to as the problem of absolute (unconditional) minimization. In such kind main difficulty consists in finding the initial point. In what follows we put in evidence an algorithm to solve this problem.

For this we generate a set of pseudo-random vectors:

$$\{\boldsymbol{\alpha}, \boldsymbol{\lambda}\} = (\alpha_1^s, \alpha_2^s, ..., \alpha_m^s, \lambda_1^s, \lambda_2^s, ..., \lambda_m^s), \quad s = 1, ..., \bar{N} \tag{4}$$

where $\bar{N}$ is big number about $10^6 - 10^7$. We calculate the values of our functional in these points, and so, we obtain a finite set of values of Ψ in the points (4). Let denote them by Ψ_s, $s = 1, ..., \bar{N}$. From this set we choose the minimal value $\Psi_{\min}$, and then we filter the set (4) in such way, we drop out the points, which does not satisfy the following inequality:

$$\Psi_s \leq \Psi_{\min} + \varepsilon \tag{5}$$

where ε is sufficiently small.

On this way we select a subset of the set (4), on which the values of the functional is small enough, and which accumulate in clusters around the points of minima. If the functional has at least one point of minimum, then there is at least one isolated cluster. We divide this cluster on other accumulations groups, enumerate them and denote by $c_1, c_2, ..., c_p$, where p represents the number of these small clusters. Let $l_1, l_2, ..., l_p$ denote the number of points in each cluster.

After this we find for each groups c_i, $(i = 1, 2, ..., p)$ the centroid (center of gravity) of the derivatives from the points of minimum and denote them by X_{c_i}. This point can be determined by formula:

$$X_{c_i} = \frac{\sum_{k=1}^{l_i} x_k \left(\Psi_{\max}^i - \Psi_k\right)^{31}}{\sum_{k=1}^{l_i} \left(\Psi_{\max}^i - \Psi_k\right)^{31}} \tag{6}$$

where $x_k, (k = 1, 2, ..., l_i)$ are points of the cluster c_i, $\Psi^i_{\max}$ is the maximum value of the functional on these points. The values of X_{c_i}, found in this way will, represent the first approximation (initial point). The more exact solution will be determined by the method of steepest descent. Each solution is probably one of the problem (3).

This construction has been purposed actually for that case, when one has a very big number of measurements, and to find a good first approximation (initial point) for one of the iteration algorithm for finding the points of minima of the quadratic functional. To be certain, that the obtain values of parameters $\{\alpha, \lambda\}$ give us the solution we have to do further analyses.

We proposed to divide the interval $[0, T]$ in to m_0 parts and for each subinterval to repeat the above algorithm all over again. If the new values of the parameters are close enough to previous one, then we can conclude, that the find solutions is the corrects one. If not, then we increase the number m_0 (i.e. increases the number of exponential functions in the term (1)) and we repeat the process for the new value of m_0 from the beginning.

This process one has to repeat until:

a) Either after some steps of iteration we obtain desired result;

b) Either we continue the calculations until we achieve the maximal number m_0.

A numerical experiment was realized to estimate an efficiency of the suggested algorithm. It was calculated the values of functional (2) at the $6 \cdot 10^6$ points and was obtained good approximations of a vector activity α and the coefficients of decomposition λ. Based on made calculations we can conclude, that it is possible to use above algorithm for not only numerical experiments.

References

[1] C. Lanczos, Applied Analysis, M, Goc. izdatelstvo fiz.-mat. literatury, 1961, p. 524.

[2] Tihonov A.N., Arsenin B.I., Metody reshenia nekorrektnyx zadach, M, Nauka,1986.

[3] Grebenikov E.A., Kiosa M.N., Mironov C.V., Chislenno-analiticheskie metody issledovania regularno vozmushennyx mnogochastotnyx system, M, Izd-vo MGU, 1986, p. 184.

Equivalence Transformations for Abel Equations - a Polynomial Method

G. Czichowski

University of Greifswald;
e-mail: czicho@mail.uni-greifswald.de

We present a polynomial method for deciding the equivalence of two given Abel equations and to compute then the corresponding equivalence transformation.

We consider the class of Abel differential equations

$$y' = A_0(x) + A_1(x)y + A_2(x)y^2 + A_3(x)y^3,$$

which is invariant with respect to coordinate transformations $(x, y) \to (u, v)$ of the form $u = F(x), v = G(x)y + H(x)$, which form the so called structure group G of this class. Structure groups of such classes of ODE's may be computed effectively in terms of the corresponding infinitesimal generators $\partial = \xi(x, y)\partial_x + \eta(x, y)\partial_y$. In the above case the structure group is given by generators ∂ with $\xi_y = 0, \eta_{yy} = 0$.

The question investigated here is to decide whether two Abel equations DE_1 and DE_2 are equivalent under the action of the structure group G. For polynomial computations we restrict this problem at first to the case of rational function coefficients $A_k(x)$, later the procedure may be generalized to the case of algebraic functions as coefficients.

The following method is based on ideas in of M.Berth and uses two differential invariants ABS_1 and ABS_2 for Abel equations with respect to the structure group G. For abbreviation we give the corresponding values only for an Abel equation $y' = A_0(x) + A_1(x)y + A_3(x)y^3$, that means $A_2(x) = 0$. This form may be realized easily by a "Tschirnhaus"-transformation belonging to the structure group

$$ABS_1 = \frac{(3A_0A_1A_3 + A_0A_3' - A_0'A_3)^3}{A_0^5A_3^4},$$

$$ABS_2 = \frac{(3A_0^2A_3 - A_0'A_3y + A_0A_3'y + 3A_0A_1A_3y + 3A_0A_3^2y^3)}{A_3^3y^6}.$$

The polynomial method presented now works as follows: Write the Abel equations DE_1 and DE_2 in variables $x, y, z = y'$ and $u, v, w = v'$ respectively. By evaluation of ABS_1, ABS_2 with respect to both ODE's we get equations

$$G_1(x, u) = 0, \quad G_2(x, y, u, v) = 0.$$

Elimination, Factorization and cancelling exponents of prime factors as well as nonessential factors leads to several candidates for the equivalence transformation. These candidates are then checked to realize an equivalence transformation from DE_1 to DE_2 or not.

By corresponding calculations with minimal polynomials this method may be extended to the case of algebraic coefficient functions.

Furthermore we consider examples from a special class of "Lie equations" of the form

$$y'' = A_0(x,y) + A_1(x,a)y' + A_2(x,y)y'^2$$

with corresponding invariants and analogous computations. Here the structure group is given by $\xi_y = 0,\ \eta_{yyy} = 0$ (fibre preserving transformations which are Moebius transformations with respect to y).

Numerical Solution of Boundary Value Problems for the Heat and Related Equations

Ivan Dimovski, Margarita Spiridonova

Institute of Mathematics and Informatics,
Bulgarian Academy of Sciences
G. Bonchev Str. Bl. 8, 1113 Sofia, Bulgaria;
e-mail: mspirid@math.bas.bg

1 Introduction

It is well known that the mathematical models of many problems in science and technology are described by boundary value problems(BVP) for partial differential equations (PDEs) and most frequentely they can be solved only numerically. There are a number of numerical methods for solution of PDEs and the most common of them are the Finite-Difference Methods (FDM). Some disadvantages of these methods are well known too.

The solution of boundary-values problems is usually needed in numerical form. The authors propose an analytic approach for numerical solution of linear (local and nonlocal) BVP for the heat and related equations. It is based on an extension of the Duhamel principle from the time variable to space variables.

In order to remind the Duhamel principle, let us consider the simplest case of its application. If we are looking for the solution of the BVP

$$\begin{aligned} &u_t = u_{xx} \\ &u(0,t) = 0,\ u(1,t) = \varphi(t) \\ &u(x,0) = 0 \end{aligned} \tag{1}$$

in the strip $0 \leq x \leq 1,\ t \geq 0$, then we can reduce it to the same problem but for the special choice $\varphi(t) \equiv 1$. Denoting this special solution by $U(x,t)$, the general solution of (1) is given by

$$u(x,t) = \frac{\partial}{\partial t} \int_0^t U(x, t-\tau)\, \varphi(\tau) d\tau. \tag{2}$$

In order to outline the idea of the following considerations, let us consider the BVP

$$\begin{aligned} &u_t = u_{xx} \\ &u(0,t) = 0,\ u(1,t) = 0 \\ &u(x,0) = f(x) \end{aligned} \tag{3}$$

in the same strip $0 \leq x \leq 1,\ t \geq 0$.

Usually, this problem is solved by the Fourier method using the Fourier sine-transform. However, from the standpoint of the numerical analysis this method is not quite satisfactory, since it includes time-consuming operations, such as Fourier series expansion of the function $f(x)$ and the numerical summation of the series obtained for $u(x,t)$ in many points. It is well known that these series are very slow convergent. Using an analogue of (2) (see Example 1), it could be avoided the both time-consuming stages.

2 Extension of the Duhamel principle to space variables

In order to make clear the basic idea of the approach, we shell consider a rather general nonlocal BVP with a Stieltjes boundary value condition of the following type.

Let P be a polynomial of one variable, and let us consider the evolution equation

$$u_t = P\left(\frac{\partial^2}{\partial x^2}\right) u \tag{4}$$

in the strip $0 \leq x \leq 1$, $t \geq 0$. Let Φ be a non-zero linear functional in $C^1[0,1]$. Then we are looking for a solution of (4) satisfying the boundary values conditions

$$\frac{d^{2j}}{dx^{2j}} u(0,t) = 0, \ \Phi_\xi\{\frac{d^{2j}}{dx^{2j}} u(\xi,t)\} = 0, (j = 0,1,2,\ldots,\deg P - 1) \tag{5}$$

and the initial condition

$$u(x,0) = \chi(x),$$

where $\chi(x)$ is a given function from $C^1[0,1]$.

As it is well known, each functional Φ in $C^1[0,1]$ can be represented in the form:

$$\Phi\{f\} = af(0) + \int_0^1 f'(\xi) d\alpha(\xi),$$

where a is a constant and α is a function with bounded variation.

Further we consider only the special cases:

$$\Phi(f) = f(1) \text{ (local case) and } \Phi(f) = \int_0^1 f(\xi) d\xi \text{ (nonlocal case)}.$$

All our further considerations are based on the following

Theorem 1. *Let Φ be a linear functional in $C^1[0,1]$, such that $\Phi_\xi\{\xi\} = 1$. Then the operation*

$$(f \star g)(x) = -\frac{1}{2}\Phi_\xi \left\{ \int_0^\xi h(x,\xi) d\xi \right\}, \tag{6}$$

where

$$h(x,\xi) = \int_{\xi}^{x} f(x+\xi-\eta)g(\eta)d\eta - \int_{-\xi}^{x} f(|x-\xi-\eta|)g(|\eta|)sgn(x-\xi-\eta)\eta d\eta \quad (7)$$

is a bilinear, commutative and associative operation in $C[0,1]$ *such that the right inverse operator* L *of* d^2/dx^2 *which satisfies the boundary value conditions* $(Lf)(0) = 0, \Phi(Lf) = 0$ *has the form* $Lf = \{x\} \star f$.

Operation (6) bears the name *convolution of the operator* L .

For a proof, see [Dim1] , pp. 176-177.

By means of the convolution (7) it can be proposed the following extension of the classical Duhamel principle for problem (4)-(5):

Theorem 2. *Let* $U(x,t)$ *be a solution of* (4) *for the special choice* $\chi(x) \equiv x$. *Then*

$$u(x,t) = \frac{\partial^2}{\partial x^2}(U(x,t) \star \chi(x)) \quad (8)$$

is a solution of (4) *provided* χ *satisfies the boundary value conditions of* (4) .

The proof can be obtained either by a direct check, or using operational calculus approach (see [Dim2], p. 140).

3 Examples

Examples illustrating the application of the presented approach are described. All computations related to them are performed with the computer algebra system *Mathematica* [SW].

3.1 Example 1

We are looking for the solution of the BVP

$$\begin{aligned} &u_t = u_{xx}, \ \ 0 \leq x \leq 1, \ t \geq 0 \\ &u(0,t) = 0, \ u(1,t) = 0 \\ &u(x,0) = f(x). \end{aligned}$$

From Theorem 2, when $\Phi_\xi \{u(\xi,t)\} = u(1,t)$ we obtain

$$u(x,t) = \int_0^1 [U(1-x-\xi,t) - U(1+x-\xi,t)]\, f(\xi)d\xi,$$

where

$$U(x,t) = \sum_{n=1}^{\infty} (-1)^n \exp(-n^2\pi^2 t)\cos n\pi x.$$

A numerical solution using the above formulas for $f(x) = xsin(\pi x)$ has the following graphical image shown by Figure 1.

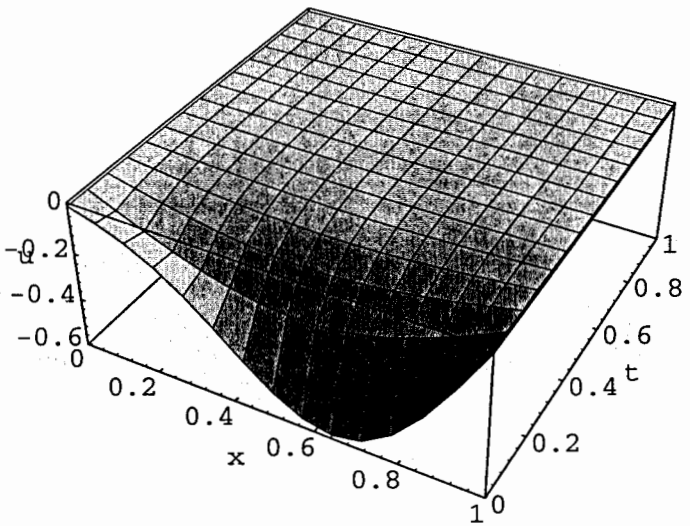


Figure 1: Example 1, $f(x) = xsin(\pi x)$

A similar solution of this problem is considered in [Wid] for the strip $0 \leq x \leq \pi, t > 0$.

Widder obtains the representation

$$u(x,t) = \int_0^\pi [\theta(x-y,t) - \theta(x+y,t)]\, f(y)dy,$$

where $\theta(x,t)$ is the classical θ-function

$$\theta(x,t) = \frac{1}{2\pi} + \frac{1}{p}\sum_{n=1}^{\infty}(-1)^n exp(-n^2\pi^2 t)\cos n\pi x.$$

This representation is not essentially different from our representation. A graphical image of such a numerical solution for $f(x) = \sin(x)$ follows (Figure 2).

Examples 2 and 4 are connected with the same boundary value functional $\Phi\{f\} = f(1)$.

3.2 Example 2

This example is described in [Lat], p.47. It is obtained by the quasi-reversibilily method of Lattes and Lions, applied to the backward heat equation.

We are looking for the solution $v = v(x,t)$ of the fourth order PDE

$$\frac{\partial v}{\partial t} + \frac{\partial^2 v}{\partial x^2} + \varepsilon\frac{\partial^4 v}{\partial x^4} = 0, 0 \leq x \leq 1,\ t \geq -\tau,$$

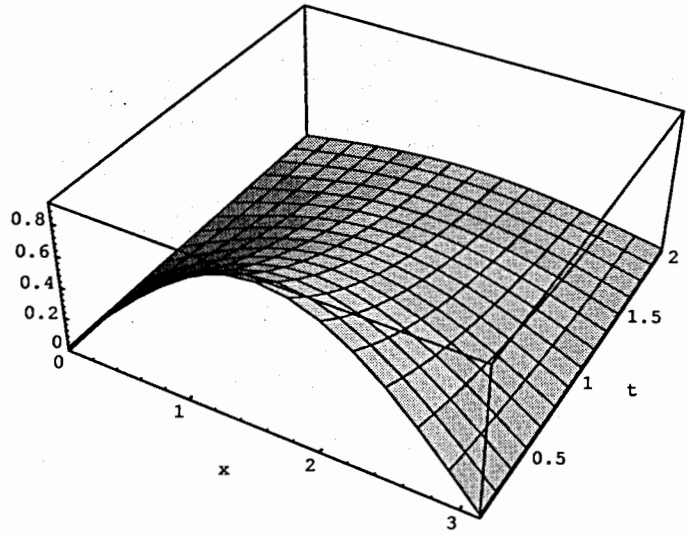


Figure 2: Example 1, $f(x) = \sin(x)$

under the boundary and initial value conditions

$$v(0,t) = v(1,t) = 0,$$

$$\frac{\partial^2 v}{\partial x^2}(0,t) = \frac{\partial^2 v}{\partial x^2}(1,t) = 0,$$

$$v(x,0) = \chi(x).$$

Let $V(x,t)$ be the solution of the above BVP under the special choice $\chi(x) \equiv x$. It is easy to obtain

$$V(x,t) = \frac{2}{\pi} \sum_{m=1}^{\infty} \frac{(-1)^{m-1}}{m} \exp m^2\pi^2(1 - \varepsilon m^2\pi^2)t \sin m\pi x .$$

By representation (8) we have

$$v(x,t) = \frac{\partial^2}{\partial x^2}(V \star \chi)$$

$$= -\frac{1}{2}\frac{\partial}{\partial x}\int_0^1 [V(1+x-\eta,t) + V(1-x-\eta,t)]\,\chi(\eta)d\eta$$

$$+\frac{1}{2}\frac{\partial}{\partial x}\int_0^x [V(1+x-\eta,t)+V(1-x+\eta,t)]\,\chi(\eta)d\eta\,.$$

If we denote

$$\Omega(x,t)=\frac{\partial V}{\partial x}=\frac{2}{\pi}\sum_{m=1}^{\infty}(-1)^{m-1}\exp(m^2\pi^2(1-\varepsilon m^2\pi^2)t)\cos m\pi x,$$

we obtain

$$v(x,t)=-\frac{1}{2}\int_0^1 [\Omega(1+x-\eta,t)-\Omega(1-x-\eta,t)]\,\chi(\eta)d\eta$$

i.e. almost the same representation as in Example 1 but with $\Omega(x,t)$ instead of $U(x,t)$.

The computed numerical solution using these formulas for the special choice $\chi(x)=x$ and $\varepsilon=0.01$ has the following graphical image (Figure 3).

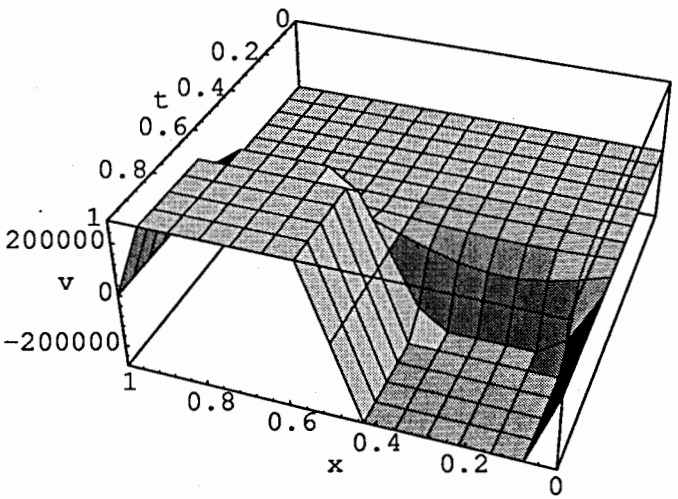


Figure 3: Example 2

3.3 Example 3

We are looking for the solution $u=u(x,t)$ of the following Samarskii-Ionkin problem [Ion], which bears the name "Samarskii-Ionkin problem":

$$u_t = u_{xx}$$

$$u(0,t) = 0, \int_0^1 u(x,\tau)d\tau = 0$$

$$u(x,0) = f(x)$$

As it is shown in [Dim1] (Theorem 3.4.8 on p. 174), the solution $u(x,t)$ can be represented in the form

$$u(x,t) = -2\int_0^x U(x-\xi,t)f(\xi)d\xi - \int_x^1 U(1+x-\xi,t)f(\xi)d\xi$$
$$+ \int_{-x}^1 U(1-x-\xi,t)f(|\xi|)sgn\xi d\xi,$$

where

$$U(x,t) = \sum_{n=1}^{\infty}\{-2x\cos 2n\pi x + 8\pi nt \sin 2n\pi x\}\exp(-4n^2\pi^2 t).$$

Here $U(x,t)$ is the solution of the same problem for $\chi(x) = x$.

This representation can be obtained from the representation (8) (Theorem 2 for $\Phi\{f\} = \int_0^1 f(\xi)d\xi$).

In [Baz] a generalization of the Samarskii-Ionkin problem for the fractional diffusion-wave equation is considered.

A comparison of the numerical solution of the problem for $f(x) = \sin 2\pi x + 3x\cos 2\pi x$ with the exact solution

$$(3\,x\,\cos(2\,\pi\,x) + (1. - 12.\,\pi\,t)\,\sin(2\,\pi\,x))\,e^{4\pi^2 t}$$

was made. Accuracy of order 10^{-14} was achieved.

A visualization of the computed numerical solution follows (Figure 4).

3.4 Example 4

Find the solution of the time-nonlocal BVP

$$\frac{\partial w}{\partial t} + \frac{\partial^2 w}{\partial x^2} + \varepsilon\frac{\partial^4 w}{\partial x^4} = 0$$

$$\int_{-\tau}^{\tau} w(x,t)dt = \chi(x)$$

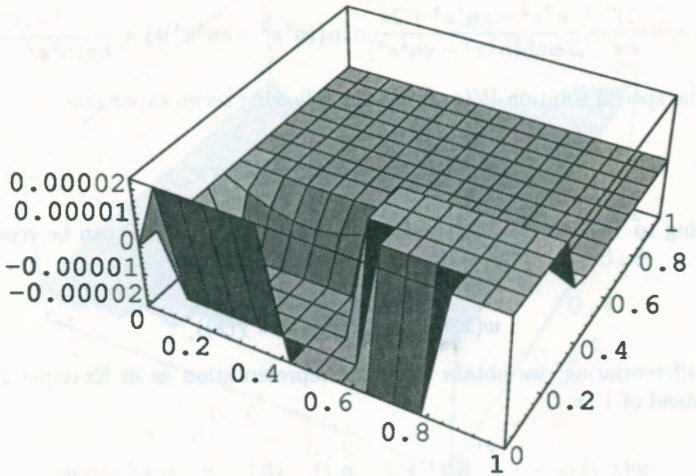


Figure 4: Example 3

$$w(0,t) = w(1,t) = \frac{\partial^2 w}{\partial x^2}(0,t) = \frac{\partial^2 w}{\partial x^2}(1,t) = 0$$

in the strip $0 \le x \le 1,\ t \ge -\tau$.

In the book of Lattes and Lions [Lat] a difference approach to the problem is developed. We propose an analytic approach, based on Theorem 2.

First, we shall find a solution $W(x,t)$ of the described problem for the special choice $\chi(x) = x$. To this end, we use the finite Fourier sine-transform

$$S_n\{f\} = \int_0^1 f(\xi) \sin n\pi\xi d\xi, \quad n = 1,2,\dots$$

Denoting $W_n(t) = \int_0^1 W(\xi,t) \sin n\pi\xi d\xi$, we obtain the ordinary differential equation

$$\frac{dW_n}{dt} - (n^2\pi^2 - \varepsilon n^4\pi^4)W_n = \frac{(-1)^{n-1}}{n\pi},\ n = 1,2,\dots,$$

where $W_n(t)$ should satisfy the nonlocal BVC:

$$\int_{-\tau}^{\tau} W_n(\sigma)d\sigma = \int_0^1 \xi \sin n\pi\xi d\xi = \frac{(-1)^{n-1}}{n\pi}.$$

By an easy calculation one obtains

$$W_n(t) = \frac{(-1)^{n-1}}{n\pi} \frac{n^2\pi^2 - \varepsilon n^4\pi^4 + 2\pi}{2\sinh(n^2\pi^2 - \varepsilon n^4\pi^4)} \exp\{(n^2\pi^2 - \varepsilon u^4\pi^4)t\} + \frac{(-1)^{n-1}}{n\pi(n^2\pi^2 - \varepsilon u^4\pi^4)}.$$

Then the special solution $W(x,t)$ has the following series expansion

$$W(x,t) = 2\sum_{n=1}^{\infty} W_n(t) \sin n\pi x.$$

According to Theorem 2, the solution of the considered BVP can be represented in the form

$$w(x,t) = \frac{\partial^2}{\partial x^2}(W(x,t) \star \chi(x)).$$

After differentiating, we obtain the same representation as in Example 2, but with $W(x,t)$ instead of $V(x,t)$:

$$w(x,t) = -\frac{1}{2}\int_0^1 [\Omega(1+x-\eta,t) - \Omega(1-x-\eta,t)]\,\chi(\eta)d\eta$$

where

$$\Omega(x,t) = \frac{\partial W}{\partial x} = 2\pi\sum_{n=1}^{\infty} W_n(t) \cos n\pi x.$$

Graphical image of the computed numerical solution for $\chi(x) = x$ and $\varepsilon = 0.01$ follows (Figure 5).

4 Concluding remarks

The use of Duhamel-type representations of the solutions of linear BVP for partial differential equations has evident advantages in comparison with the known numerical methods. We would like to point out some of them:

- Compairing with the use of difference methods:
  - i) the values of the solution can be obtained for the points where they are needed only;
  - ii) the accuracy can be controled by the chosen quadratic formulas parameters;
  - iii) never any numerical instability occur.
- Compared with the Fourier's method, the time-consuming operations mentioned above are avoided. In such a way, the approach proposed here has all the advantages of the Fourier methods and avoids most of it shortcommings.

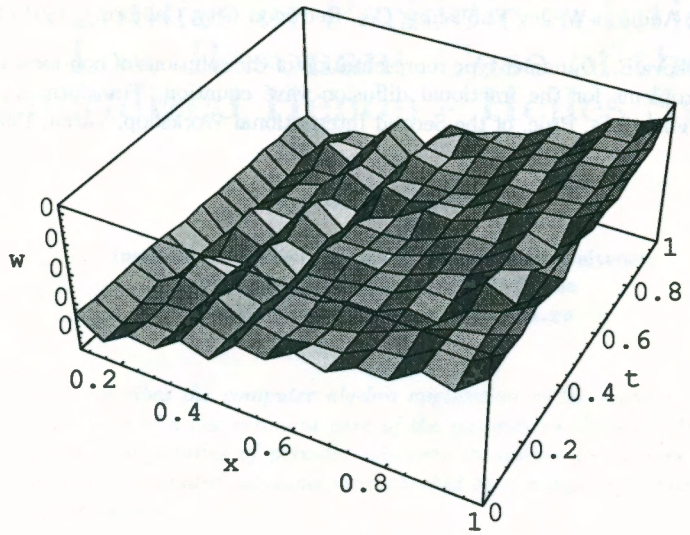


Figure 5: Example 4

- The performance of the computations related to the application of the presented approach in the environment of the computer algebra system *Mathematica* gives additional advantages: high accuracy of the numerical computations, visualization of the resuts and convenient use of the implemented approach.

References

[Dim1] Dimovski I., Convolutional Calculus, Kluwer Academic Publishers, Dordrecht, 1990

[Dim2] Dimovski I., Duhamel-Type Representation of the Solutions of Non-Local Boundary Value Problems, Differential Equations and Applications (I), Proc. of the Second Conference, Rousse'81, Bulgaria, 1982, pp. 240-247

[Wid] Widder D.V., The Heat Equation, Academic Press, New York, 1975

[Lat] Lattes R. and J.-L. Lions, The Method of Quasi-Reversibility. Applications to Partial Differential Equations, American Elsevier Publ. Co., Inc., New York, 1969

[Ion] Ionkin N. I., Numerical Solution of a Nonclassical Boundary Value Problem for the Heat Equation, J. Differential Equations, V. 13, No 2, 1977, pp.294-304 (in Russian)

[SW] Wolfram S., Mathematica. A System for Doing Mathematics by Computer, Second Edition, Addison-Wesley Publishing Co., Redwood City, California, 1991

[Baz] Bazhlekova E., Duhamel-type representation of the solutions of non-local boundary value problems for the fractional diffusion-wave equation, Transform Methods & Special Functions, Proc. of the Second International Workshop, Varna, 1996, pp.32-40

On Families of Periodic Solutions of Low Resonant Case of the Generalized Henon - Heiles System

Victor F. Edneral[1]

Institute for Nuclear Physics of Moscow State University,
Vorobievi Gori, Moscow, 119899 Russia
e-mail: edneral@theory.sinp.msu.ru

The paper describes the computer algebra application of the normal form method to bifurcation analysis of a low resonant case of the generalized Henon - Heiles system. A behavior of all local families of periodic solutions in system parameters is determined. Corresponding approximated solutions were checked by a comparison with the numerical solutions of the system.

1 Introduction

Normal form methods use a nonlinear change of variables to transform a nonlinear system of ordinary differential equations to a simpler form. In this paper we use an algorithm based on an approach developed by A.D. Bruno [1] for computing the resonant normal form. An important advantage of this approach is its algorithmic simplicity: there are direct recurrence formulas for coefficients of the transformation and of the transformed system, and thus the storage of large intermediate results is not necessary. This approach does not require solving any intermediate systems and there are no restrictions on low resonance cases.

2 The Generalized Henon–Heiles System

The generalized Henon–Heiles system is a couple of second order differential equations:

$$\begin{aligned} \ddot{x} + l_1 \cdot x + 2 \cdot d \cdot x \cdot y = 0\,, \\ \ddot{y} + l_2 \cdot y + d \cdot x^2 - c \cdot y^2 = 0\,. \end{aligned} \tag{1}$$

This system is known to be integrable [3] when:

1. $l_1 = l_2, \quad c/d = -1;$
2. $c/d = -6;$

[1]Supported by the RFBR grant for the Scientific Schools support (the School by Academician A.A.Logunov)

3. $16l_1 = l_2, \quad c/d = -16.$

Below we discuss the case with the same pure imaginary eigenvalues of the linear part of (1). I.e. we suppose that $l_1 = l_2 > 0$. By changing the time variable we choose $l_1 = l_2 = 1$ and thus rewrite (1) to the "low resonant form".

Take into account the fact that if $d = 0$ the system can be integrated in an analytical form because equations for x and y will be independent:

$$\ddot{x} + x = 0\,, \quad \ddot{y} + y - c \cdot y^2 = 0\,.$$

The first equation above has an exponential solution and the second one has a solution in terms of elliptic functions [9], example 6.10:

$$\begin{aligned} x(t) &= C_1 \cdot \exp(it) + C_2 \cdot \exp(-it), \qquad C_1, C_2 \in \mathbb{C}\,, \\ t &= \int \frac{dt}{\sqrt{\frac{2}{3} \cdot c \cdot y^3 - y^2 + C_3}}, \qquad C_3 \in \mathbb{R}\,. \end{aligned} \tag{2}$$

Thus, let us assume that $d \neq 0$. Then with changing $d \cdot x \to x$, $d \cdot y \to y$ and $c/d \to c$ we obtain (1) in the form:

$$\begin{aligned} &\ddot{x} + x + 2 \cdot x \cdot y = 0\,, \\ &\ddot{y} + y + x^2 - c \cdot y^2 = 0\,, \end{aligned} \tag{3}$$

with the Hamiltonian:

$$h = \frac{1}{2}[(\dot{x})^2 + (\dot{y})^2 + x^2 + y^2] + x^2 y - \frac{c}{3} y^3\,. \tag{4}$$

A linear change of variables:

$$\begin{aligned} x &= y_1 + y_2\,, \quad \dot{x} = -i\,(y_1 - y_2) \\ y &= y_3 + y_4\,, \quad \dot{y} = -i\,(y_3 - y_4)\,, \end{aligned} \tag{5}$$

transforms (3) to the form required by the method:

$$\begin{aligned} \dot{y}_1 &= -i\ y_1 - i\,(y_1 + y_2)\,(y_3 + y_4)\,, \\ \dot{y}_2 &= \ \ i\ y_2 + i\,(y_1 + y_2)\,(y_3 + y_4)\,, \\ \dot{y}_3 &= -i\ y_3 - \tfrac{i}{2}\,[(y_1 + y_2)^2 - c(y_3 + y_4)^2]\,, \\ \dot{y}_4 &= \ \ i\ y_4 + \tfrac{i}{2}\,[(y_1 + y_2)^2 - c(y_3 + y_4)^2]\,, \end{aligned} \tag{6}$$

The eigenvalues of this system are two pairs of complex conjugate imaginary units: $\Lambda = (-i, i, -i, i)$ [2]. So this is a deeply resonant problem, i.e. the most difficult (and interesting) type of a problem.

[2]Remark that in paper [6] the other order of variables is used. The corresponding vector of eigenvalues is $\Lambda = (-i, -i, i, i)$ there. It corresponds to interchanging $y_2 \leftrightarrow y_3$ for agreement with notation of the present paper.

3 A Normal Form for the Generalized Henon–Heiles System

For a pure resonance of the generalized Henon–Heiles system the ratios of all pairs of eigenvalues are ± 1 and the normal form for (6) is:

$$\dot{z}_k = z_k G_k \stackrel{\text{def}}{=} \lambda_k z_k + z_k \sum g_{k,q_1,q_2,q_3,q_4,p}\, z_1^{q_1} z_2^{q_2} z_3^{q_3} z_4^{q_4} c^p, \quad k = 1,\ldots,4. \tag{7}$$
$$q_k \geq -1,$$
$$q_1,\ldots,q_{k-1},q_{k+1},\ldots,q_4 \geq 0,$$
$$q_1+q_3 = q_2+q_4 > 0$$
$$0 \leq p \leq q_1+q_2+q_3+q_4$$

$G_k = G_k(\mathbf{z},c)$ are series in $z_1,\ldots,z_4$ and polynomials in c, $g_{k,q_1,q_2,q_3,q_4,p}$ are numeric coefficients which can be calculated by the LISP based program NORT [4].

The set $\mathcal{A}$ in a phase space of system (7) is defined by the system of equations [1], [7]:

$$\mathcal{A} = \{z_1,z_2,z_3,z_4 : \lambda_k z_k \omega = \lambda_k z_k + z_k \sum g_{k,q_1,q_2,q_3,q_4,c}\, z_1^{q_1} z_2^{q_2} z_3^{q_3} z_4^{q_4} c^p\}, \quad k = 1,\ldots,4, \tag{8}$$

where ω is a series in $z_1,\ldots,z_4$ and polynomials in c. ω does not depend on index k along the set $\mathcal{A}$.

Searching for *all* local periodic families of solutions of system (6) is equivalent to searching for the set $\mathcal{A}$ which contains all of them. It is important that along the set $\mathcal{A}$ formal series in formulae above have a basis which consists of convergent power series in z_k variables [3] [1]. So the families of periodic solutions of (8) can be expressed (approximated) in terms of convergent series.

Equations (8) can be recast (by eliminating ω which is non zero for non-trivial solutions) in the form:

$$\begin{aligned} P_1 &\stackrel{\text{def}}{=} z_1 z_2 \cdot [G_1(\mathbf{z},c) + G_2(\mathbf{z},c)] = 0, \\ P_2 &\stackrel{\text{def}}{=} z_3 z_4 \cdot [G_3(\mathbf{z},c) + G_4(\mathbf{z},c)] = 0, \\ P_3 &\stackrel{\text{def}}{=} z_1 z_4 \cdot [G_1(\mathbf{z},c) + G_4(\mathbf{z},c)] = 0, \\ P_4 &\stackrel{\text{def}}{=} z_2 z_3 \cdot [G_2(\mathbf{z},c) + G_3(\mathbf{z},c)] = 0. \end{aligned} \tag{9}$$

Of course no more than 3 equations above are independent, but the form (9) is symmetric.

Because of (8) all families of periodic solutions of (7) have the form:

$$z_j = a_j \exp(-i\omega t), \quad z_{j+1} = a_{j+1} \exp(i\omega t), \quad j = 1,3. \tag{10}$$

The a_j above are integration constants and ω is time independent and plays the role of frequency. It depends on constants c and a_j only.

With (10) we can rewrite (9) as an algebraic problem of solving a system of equations

[3] The parameter c is not supposed to be small.

over the ring of formal power series in a_k:

$$\begin{aligned} P_1 &\stackrel{\text{def}}{=} a_1 a_2 \cdot [G_1(a_1,\ldots,a_4,c) + G_2(a_1,\ldots,a_4,c)] = 0\,, \\ P_2 &\stackrel{\text{def}}{=} a_3 a_4 \cdot [G_3(a_1,\ldots,a_4,c) + G_4(a_1,\ldots,a_4,c)] = 0\,, \\ P_3 &\stackrel{\text{def}}{=} a_1 a_4 \cdot [G_1(a_1,\ldots,a_4,c) + G_4(a_1,\ldots,a_4,c)] = 0\,, \\ P_4 &\stackrel{\text{def}}{=} a_2 a_3 \cdot [G_2(a_1,\ldots,a_4,c) + G_3(a_1,\ldots,a_4,c)] = 0\,. \end{aligned} \tag{11}$$

Searching for all local families of periodic solutions of system (7) is equivalent to determining all solutions of system (11).

The original system (3) is real, thus real solutions of (7) satisfy the reality conditions:

$$\begin{aligned} &z_{j+1} = \overline{z_j}, \quad j = 1,3 \\ &\text{or} \\ &a_{j+1} = \overline{a_j}, \quad j = 1,3 \end{aligned} \tag{12}$$

so we can fix (by neglecting a trivial time shift) $a_2 = a_1$ as pure real.

Because system 3 is even in time all families of solutions arise by couples.

4 Calculation of Results

By using our program NORT for system (6) we have calculated the normalizing transformation and normal form till the 10^{th} order in z_k (i.e. till 12^{th} in a_k for P_k series). We calculated periodic solutions and compared them with corresponding numerical solutions at different values of parameters. Below we discuss the bifurcation picture and the phase portrait of system (3) following this analysis [4].

It is proved in [2] that for any reversible resonant system of the 4^{th} order, both the series P_1 and P_2 have the same factor:

$$\begin{aligned} P_1(\mathbf{z}) &= (z_1^r z_4^s - z_2^r z_3^s) \cdot Q_1(\mathbf{z})\,, \\ P_2(\mathbf{z}) &= (z_1^r z_4^s - z_2^r z_3^s) \cdot Q_2(\mathbf{z})\,, \end{aligned}$$

where r and s are smallest positive integers which satisfy the equation $\lambda_1 \cdot r = \lambda_3 \cdot s$.

For our case $s = r = 1$, but the system has an additional symmetry and we can find by factoring that the P_1 and P_2 series calculated by the NORT program have a more complicated factor:

$$\begin{aligned} P_1 = \; & 2i \cdot \; (a_1^2 a_4^2 - a_2^2 a_3^2) \cdot \\ & \cdot \left[1 + \frac{1}{6}c + \left(\frac{133}{18} - \frac{95}{108}c - c^2\right) a_1 a_2 + \right. \\ & \left. + \left(\frac{4}{9} - \frac{172}{27}c + \frac{23}{18}c^2 + \frac{89}{108}c^3\right) a_3 a_4 + O(a^4)\right]\,, \\ P_2 = -2i \cdot \; & (a_1^2 a_4^2 - a_2^2 a_3^2) \cdot \\ & \cdot \left[1 + \frac{1}{6}c + \left(\frac{13}{2} - \frac{95}{108}c - \frac{1}{9}c^2\right) a_1 a_2 + \right. \\ & \left. + \left(\frac{20}{9} - \frac{148}{27}c - \frac{1}{2}c^2 - \frac{7}{108}c^3\right) a_3 a_4 + O(a^4)\right]\,, \end{aligned} \tag{13}$$

[4] All calculations below were carried out for the mechanical energy $h = 1/12$.

where (and in some places below) we adduce for simplicity first terms of the calculated series only.

Recall that we are interested only in local solutions, i.e. in such solutions of (11) which can include the stationary point $a_k = 0$, $k = 1, \ldots, 4$ as a particular case.

Thus, instead of the first pair of equations in (11), we have the equation:

$$a_1^2 a_4^2 - a_2^2 a_3^2 = 0\,,$$

because the brackets in (13) cannot add any nontrivial local solution as they contain constant terms at $c \neq -6$, and at $c = -6$ these both brackets are proportional to $a_1 a_2 + 4a_3 a_4$, which is a sum of squares of modules (see (12)).

So, if we now fix $a_1 = a_2 = a$ as a pure real, then the first couple of equations (13) has two solutions:

1. $a_3 = a_4 = b$ has a pure real value;

2. $a_3 = -a_4$ has a pure imaginary value. Let $a_3 = ib$.

4.1 Case of Pure Real a_3

In this case the second pair of equations (13) gives a single equation:

$$\begin{aligned} P_3 \;&= -P_4 = \alpha_1 \cdot a \cdot b \cdot [a^2 - (c+2)b^2] \cdot \left[c - 1 - \left(\frac{2431}{180} - \frac{29}{90}c + \frac{217}{60}c^2\right) a^2 - \right. \\ &\left. - \left(\frac{67}{90} - \frac{1289}{180}c + \frac{233}{45}c^2 - \frac{157}{36}c^3\right) b^2 + O(a^4) + O(b^4) + O(a^2 b^2)\right] = 0\,. \end{aligned} \tag{14}$$

α_1 here is a nonzero numerical constant. Let us discuss the families of periodic solutions which correspond to zeroing each factor of the product above.

A couple of families of periodic solutions which corresponds to $a = 0$ exists at any values of c and lies in the plane $x = 0$. This is a family with a single internal parameter. We choose the mechanical energy h from (4) as this parameter. At $c = 1$ this family corresponds to family 5 of the classic Henon–Heiles system [8], see Fig. 1 and paper [6].

In Fig. 1, the intersections of periodic solutions of the Henon–Heiles system with Surface Of Section (SOS) [8], which is defined by equations $\mathrm{SOS} = \{x = 0,\ \dot{x} = \dot{x}(x, y, \dot{y}, h) > 0\}$, are displayed in coordinates y, $\dot{y}$. The periodic solutions of families 5 lie entirely in the plane $x = \dot{x} = 0$. For this case there is an analytical solution of type 2 in elliptic functions.

The families which correspond to $b = 0$ also exist at any values of c. They look like family 4 of the Henon – Heiles system (Fig. 1). The corresponding intersection flows slowly from left to right at increasing c. The frequency of these periodic families is:

$$\begin{aligned} \omega_4 = \;& 1 - 5\rho/3 + \rho^2(-281 + 504c)/108 + \\ & + \rho^3(-13913 + 645024c - 323488c^2)/19440 + \\ & + \rho^4(33903721 + 134318856c - 137045376c^2 + 59393664c^3)/699840 \\ & + \rho^5(103971857615 + 172223295216c - 402212367472c^2 + \\ & + 294216077568c^3 - 105272265984c^4)/220449600 + O(\rho^6)\,, \end{aligned}$$

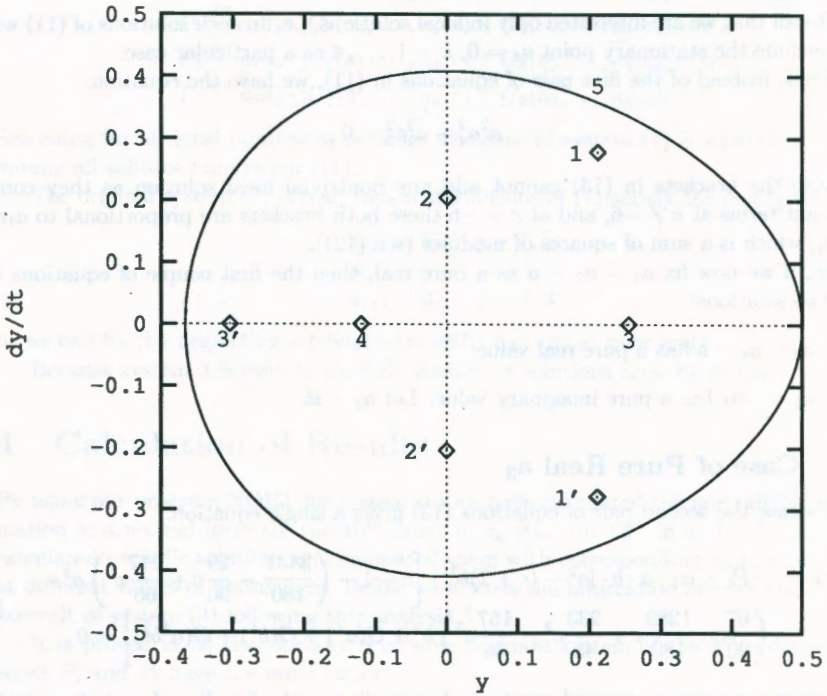


Figure 1: Intersections of periodic solutions of the Henon–Heiles system ($c = 1$) with the surface of section (SOS) at the energy level $h = \frac{1}{12}$.

where:

$$\begin{aligned}\rho = \; & h/2 + 7h^2/72 + h^3(-1099 - 1184c)/3456 + \\ & + h^4(-1830221 - 1187297c + 208324c^2)/777600 + \\ & + h^5(-64702312929 - 10993351152c + 19241890208c^2 - \\ & - 4382098944c^3)/6718464000 + O(h^6)\,.\end{aligned}$$

These are one-parametric families.

Let us define a relative error as the maximum of relative difference between the tabulated series (of tenth order) $x(t)_{app}, y_{app}(t), \dot{x}_{app}(t), \dot{y}_{app}(t)$ and the corresponding numerical solution during one period:

$$ferr \stackrel{\text{def}}{=} \sup_{t \in [0, 2\pi/\omega_i]}$$

$$\sqrt{\frac{(x_{num}(t)-x_{app}(t))^2+(y_{num}(t)-y_{app}(t))^2+(\dot{x}_{num}(t)-\dot{x}_{app}(t))^2+(\dot{y}_{num}(t)-\dot{y}_{app}(t))^2}{x^2_{num}(t)+y^2_{num}(t)+\dot{x}^2_{num}(t)+\dot{y}^2_{num}(t)}}$$

The relative error of the approximation of this case is about 6% at $c = -1$; 0.3% at $c = 1$ and 0.23% at $c = 2$.

The next case is $a^2 = (c+2) \cdot b^2$. At $c = 1$ it corresponds to 2 and 2' families in fig. 1. The families are real if $c \geq -2$ only. It is one parametric case also. With increasing c it slowly flows to the abscissa axis. At $c = -2$ it coincides with curve 5. The frequency is:

$$\begin{aligned}\omega_2 = \; & 1 - 20h_1/3 - 380h_1^2/3 - 878960h_1^3/243 - \\ & -88121780h_1^4/729 - 3203319760h_1^5/729 + O(h_1^6)\end{aligned}$$

where $h_1 = h/(2(3+c))$. The relative error is about 4% at $c = 0$; 0.4% at $c = 1$ and 0.06% at $c = 2$.

Zeroing the last brackets in (14) is possible near the point $c = 1$, when the corresponding distance $c - 1$ is about a square of amplitudes a or b, or, say, about energy h. So we can suppose that $c = 1 + \varepsilon \cdot h$ and the corresponding solution exists at least at ε lower or about 1. This case agrees with families 1 and 1' in Fig. 1. The intersection with SOS flows to the abscissa axis and from right to left with increasing ε. These are one parametric families. The frequency is: $\omega_1 = 1 - 25\varepsilon h/252 + O(h^2)$. The relative error here is 0.09% at $\varepsilon = -1$; 0.09% at $\varepsilon = 1$ and 0.2% at $\varepsilon = 2$. A small value of error may say about possibility of spreading this case into wider domain of the parameter ε.

4.2 Case of Pure Imaginary a_3

Let $a_3 = ib$. In this case the second pair of equations (13) also gives a single equation:

$$\begin{aligned}P_3 = P_4 = \; & \alpha_2 \cdot a \cdot b \cdot (c+1) \cdot \\ & \cdot [7a^2 - (5c+2)b^2 + O(a^4) + O(b^4) + O(a^2b^2)] = 0\,.\end{aligned} \tag{15}$$

α_2 here is a nonzero numerical constant. Let us one more discuss families of periodic solutions, which correspond to zeroing factors of the product (15).

Cases $a = 0$ and $b = 0$ have been discussed. Case $c = -1$ is a specific one. It is an essential two parametric case, but it is better to choose parameters from a physical meaning. Let us introduce the energy as one of parameters. We will suppose that the amplitudes a and b connected with the energy h in this way:

$$a^2 + b^2 = p(h), \quad a^2 = \beta \cdot p(h), \quad \beta \in [0, 1]\,,$$

where $p(h)$ is the series:

$$\begin{aligned}p(h) = \; & h/2 + 7h^2/72 + 85h^3/3456 - \\ & -2173h^4/3888 - 240679781h^5/53747712 + O(h^6)\,.\end{aligned}$$

This parameterization is good because it displays that frequency depends on $p(h)$, i.e. on energy only:

$$\omega_{-1} = 1 - 5p(h)/3 - 785p^2(h)/108 - 65495p^3(h)/1296 - \\ -59370835p^4(h)/139968 - 4971155135p^5(h)/1259712 + O(p^6(h)).$$

At the beginning of the interval ($\beta = 0$) a behavior of these periodic families looks like solution 5 in fig. 1. Then the solutions cross the SOS look like 3 and 3′ on the segment $\beta \in (0,1)$ and at the end of the interval ($\beta = 1$) these 3 and 3′ join each other in the point which corresponds to solution 4. Relative error here equals 6% at $\beta = 0$; 5% at $\beta = 0.5$ and again 6% at $\beta = 1$.

The case when the last bracket in (15) is equal to zero is close to the last case of the previous subsection. It is one-parametric case and it is real only when $c \geq -\frac{2}{5}$. The case corresponds to families 3 and 3′ in the Figure. Frequency is:

$$\omega_3 = 1 + 2/3h(1+6c)/(9+5c) + \\ +1/3h^2(-135 - 1051c - 2694c^2 - 1503c^3 - \\ -105c^4)/(729 + 1215c + 675c^2 + 125c^3) + O(h^3).$$

The families start from a shape of 5 at $c = -2/5$, and then go along abscissa to the origin likewise couple 3, 3′. Relative errors equal $6 \cdot 10^{-6}$% at $c = -2/5$; $6 \cdot 10^{-5}$% at $c = 0$; $4 \cdot 10^{-3}$% at $c = 1$ and 0.08% at $c = 1.5$.

There is also a couple of two-parametric local families of complex periodic solutions at zero energy h with frequencies equal to ± 1. Such solutions were discussed in [5], [6].

5 Conclusions

In the normal form method a bifurcation analysis for local families of periodic solutions was carried out for the low resonant case of the generalized Henon–Heiles system (3). We have found 6 pairs of real families of local periodic solutions. In terms of Figure 1 they are :

1. a couple of families looks like solutions 5 in the Figure, with one internal parameter (mechanical energy h). Here and below we neglect dependence on a trivial time shift. These families exist at any value of external parameter c;

2. a couple of families looks like solutions 4 in the Figure, with one internal parameter. The families exist at any value of external parameter c;

3. a couple of families looks like solutions 2 and 2′ in the Figure, with one internal parameter. The couple exists at the value of external parameter $c \geq -2$;

4. a couple of families looks like solutions 1 and 1' in the Figure, with one internal parameter. The couple exists at least near the value of external parameter $c = 1$;

5. a couple of families looks like solutions 3 and 3′ in the Figure, with two internal parameters. The couple exists at the single value of external parameter $c = -1$;

6. a couple of families looks like number 3 and 3′ in the Figure also, with one internal parameter. The couple exists at the value of external parameter $c \geq -2/5$.

There are two local two-parametric complex families of periodic solutions at the energy $h = 0$ also.

Acknowledgements

The author is very grateful to Prof. R. I. Bogdanov and Dr. S. Yu. Vernov for interesting discussions and important advices.

References

[1] Bruno, A.D.: *Local Method in Nonlinear Differential Equations*, Springer-Verlag, Berlin (1989)

[2] Bruno, A.D., Soleev, A.: *Local analysis of a singularity of a reversible ODE system. Complicated cases*; Preprint # 47 of the Keldysh Institute of Applied Mathematics of RAS, Moscow (1995)(in Russian)

[3] Chang, Y.F., Tabor, M. and Weiss, J.: Analytic structure of the Henon-Heiles Hamiltonian in integrable and nonintegrable regimes. *J. Math. Phys.* **23** (1982) 531–538

[4] Edneral, V.F.: Computer generation of normalizing transformation for systems of Nonlinear ODE. In: *Proceedings of the 1993 International Symposium on Symbolic and Algebraic Computation (Kiev, Ukraine, July 6-8, 1993)*, M. Bronstein (ed.), ACM Press, New York (1993) 14–19

[5] Edneral, V.F.: Complex periodic solutions of autonomous ODE systems with analytical right-hand sides near an equilibrium point. *Fundamentalnaya i prikladnaya matematika* **1** (1995) #2 393–398 (in Russian)

[6] Edneral, V.F.: A symbolic approximation of periodic solutions of the Henon–Heiles system by the normal form method. *J.Mathematics and Computers in Simulation* **45** (1998) 445–463

[7] Edneral, V.F.: About normal form method. In: *Proc. Second Workshop on Computer Algebra in Scientific Computing (CASC'99, Munich, Germany, 1999)*, Ganzha et al. (eds.), Springer (1999) 51–66

[8] Henon, M., Heiles, C.: The applicability of the third integral of motion: Some numerical experiments. *Astronomical J.* **69** (1964) 73–79

[9] Kamke, E.: *Differentialgleichungen. I*, Verbesserte Auflage, Leipzig (1959)

Computer Algebra at KELDYSH Institute of Applied Mathematics

G.B.Efimov, I.B.Tshenkov, E.Yu.Zueva

Keldysh Institute of Applied Mathematics RAS, Moscow;
e-mail: efimov@applmat.msk.su

Brief retrospective review and main references concerned Computer Algebra (CA) researches and applications at famous Keldysh Institute of Applied Mathematics (Russia Academia of Science) are presented. At the Institute CA was used in various areas: Applied Celestial Mechanics, Mathematics, Robotics, Hydromechanics, Applied Calculation Methods. Some original program sys-tems were elaborated. During several years the work devoted to classification of existing systems and their possibilities was done. Conferences and seminars concerned CA applications in mechanics were organized. A language of recur-sive functions, REFAL, was generated at the Institute. Later it was modified and intensively used in various fields, in particular CA. Many ideas and results, which one can find in researches on CA and close fields implemented at Keldych Institute, are still interesting today. The work was supported by RFBI, grants No 01-01-00015 and No 00-15-96036.

Keldysh Institute of Applied Mathematics of Russian Academia of Science (RAS) was founded in 1952 by M.V.Keldysh for solving difficult scientific problems of national importance, such as nuclear physics, cybernetics, space mechanics and others. At the Institute experts on different areas - mathematicians, physicists, mechanicians, computer scientists - were working together, in close contacts with each others. Such famous scientists as A.N.Tichonov, K.I.Babenko, I.M.Gelfand, A.A.Liapunov, I.B.Zeldovich, D.E.Okhozimsky, A.A.Samarsky, V.S.Yablonsky, M.R.Shura-Bura, A.N.Miamlin, M.L.Lidov, V.V.Beletsky, S.P.Kurdiumov, T.M.Eneev were among them.

Numerous difficult problems required various mathematical methods for their solution. Active pioneer's using of first computers, enthusiasm caused by early successes in this new field - all this gave many interesting ideas. The idea of symbolic computations arose just after computer's appearance, as a will to learn computer to interact with the human person via ordinary mathematical language, to facilitate the job of physicist or mecanician. Reviews of early soviet researches on Computer Algebra (or Symbol Manipulations, as it was called initially) are presented in the publications [1-13]. These old works may be interesting -not all of them were known sufficiently at the time when they were carried out; some ideas are still valuable now.

First experiments with Symbol Manipulations were made in Russia as early as the beginning of sixties. In 1956 at the first National Conference devoted to computer science A.A.Dorodnitcin formulated the task which we can consider as the beginning of Symbol Manipulation in Soviet Union [14]. This task was concerned with the investigation of

nuclear explosion in the atmosphere. It was proposed to built the solution in the form of two asymptotic power series (near the center of the explosion and far from it). These assimptotics should be conjugated by common numerical solution in the regular area. Thus it was the combined analytical/numerical approach. D.E.Okhozimsky applied this approach to cosmodynamics problem [15]. He also built two assimptotics conjugated by common numerical part. Assimptotics construction on one of the first soviet computers "Strela" was pioneer too.

At the same time in Leningrad department of Steklov Mathematical Institute Nobel laureate L.V.Kantorovich investigated some types of symbolic notation for Computer Algebra. His follower T.N.Petrova created the program "Polynomial Prorab", used later for the tasks of theory of elasticity and others [16].

The formulation of Dorodnitcin is referred by A.A.Stogny [17], the follower of V.A. Glushkov in Kiev. He proposed an algorithm to obtain the polynomial solution of differential equation. It was probably the first result of the famous Kiev Computer Algebra school.

N.N.Yanenko investigated Cartan's methods of analysis of compatibility of systems of differential equations in partial derivations and realized it on computer "Strela" [18].

Solutions in series of some problems of Celestial Mechanics were built in Leningrad by V.A.Brumberg and his colleages [19]. This approach was elaborated and extended later. Similar work was done by I.B.Zadyhajlo and Z.P.Vlasova in KIAM, Moscow. The similar approach to the tasks of the theory of elasticity was elaborated by the group of L.V.Kantorovich in Leningrad and V.K.Kabulov in Tashkent [20].

Let us return to KIAM. Here the first attempts to use Symbol Manipulations were done in the area of Applied Celestial Mechanics. As it was mentioned above, manipulations with trigonometric and power series were implemented on soviet computer "Strela" by Z.P.Vlasova and I.B.Zadyhajlo (non-published). In 1964 D.E.Okhozimsky initiated the work on semi-analytical solution of one cosmodynamics problem, the low-thrust flight in central field of gravitation [15]. G.B.Efimov realized this approach for simplest Poisson series (1970) [21].

From 1970 A.P.Markeev used CA for Gamilton's systems normalization and periodic solution stability analysis [22]. The next steps in this direction were done by his follower A.G.Sokolsky [23-25]. This work was continued by the same group in MAI and later by A.G. Sokolsky in ITA RAS (Leningrad) [26-29], where the well known school of V.A.Brumberg did work already. In MAI Computer Algebra applications to education were developed as well [30].

V.A.Saryshev and S.A.Gutnik used CA for the problem of sputnik equilibrium stability (1984) [31-32]. Some important results were obtained last years by A.D.Bruno and his group (see below).

From 1963 M.L.Lidov with his group did numerous experiments concerned CA application to sputnik dynamics problems [33]. A method united both analytical and numerical approaches was proposed [34-35]. For elliptic orbits and distortions of different sorts, analytical approach was used to construct the Hamilton disturbing function H*. Then, coordinate transformation and calculation of right parts of disturbed motion equations in every step of integration, was done via Hamilton function differentiation. This approach provides high accuracy method of motion calculation and allows to avoid labor-consuming

calculations [36-38]. Infortunately, requirements to CA systems to be used in this scheme were rather high, and available CA systems were not capable to satisfy them. Thus, these very interesting experiments didn't produce practically usable integrated (analytically-numerical) system. From the other hand, it was impossible to use ready CA systems of general purpose, very primitive at this time, to solve such complicated problems. As a result, very sophisticated special algorithms and programs were elaborated, but these investigations didn't stimulate the development of CA systems for common usage.

The very interesting page in russian cybernetics is the history of REFAL. In 1969 V.F.Tourchin created an original computer language REFAL based on new principle of programming - associative text processing on the base of recursive function theory, without directly addressed control of the program [39-41]. From the beginning Computer Algebra was among potential areas of REFAL applications [42]. However, first REFAL realization was rather "scientific" then practical, since it was isolated and not compatible with "ordinary" software - numerical packages, library support, memory allocation and so on. Additional efforts of many people required to make REFAL modifications practically usable, in particular for Computer Algebra Applications. S.N.Florentcev, S.A.Romanenko, A.V.Klimov were the first who wrote high-effective compiler from REFAL.

The first REFAL program realized some CA features was elaborated by V.F. Tourchin and others to solve some problems of Nuclear Physics in series [43]. I.B. Tshenkov elaborated general purpose SAM named SANTRA and later modified it [44-45]. On this base the special applied system DISPLAN was made for processing non-standard difference schemes, by I.B.Tshenkov and M.Yu.Shashkov [46]. M.L.Lidov and L.M. Bakuma were among the first REFAL's users in applied area. They did some attempts to process Poisson series. Analogous attempts in the group theory were done by H.C.Ibragimov and I.B.Tshenkov.

Some enthusiasts of REFAL were united by KIAM and worked in close contact with it. V.L.Topunov with his colleagues from Moscow Institute of Pedagogy used REFAL-based CA system in differential geometry [47]. Together with V.P.Shapeev and the others followers of N.N.Yanenko (Novosibirsk) they realized method of H.Cartane and investigated the characteristics of difference schemes [48,17]. L.V.Provorov in N.Jukovsky Central Air-Hydrodynamics Institute and in Bauman High Technology University used CA system ALCOR in large area applications in engineering [49]. In mechanics, O.M.Gorodeskiy (Grodno) and A.V.Korlyukov created CA system for simulation in dynamics of Gyroscope systems (used by ac. D.M.Klimov) [50]. As a first stage of the work, the equations of the motion were automatically derived. Usage of REFAL allowed to present the equations in convenient form near to human presentation. L.F.Belous and I.R.Akselrod (in Kharcov) used REFAL for integration several different programming systems and numerical packages into a united system, including in this system the well known REDUCE and domestic Computer Algebra package SIRIUS [51].

In Keldysh Institute the possibilities to create REFAL-processor were investigated. A.N.Miamlin, I.B.Zadyhajlo and V.K.Smirnov were the leaders of the project [52-54]. L.K. Eisymont analyzed REFAL efficiency for both program and apparatus realization, in particular from the CA application point of view [55,56]. REFAL processor EC-2702 was simulated by V.K.Smirnov with his group on the computer EC-2635 with microprogramming [54,57-60]. This processor was compatible with EC-series computers, it was

used for CA problems as well as for translators [61].

REFAL was initially planned as a "meta language". It was used indeed in wide scope of textual tasks, more or less near to CA. V.A.Fisun, A.I.Choroshilov and others realized computer languages Simula-1, DYNAMO [62,63], complex for cosmic training TRIKS, ALGOL-FORTRAN converter for physical packages - all based on REFAL. Yu.F.Golubev simulated the work of on-board sputnik computer. In this direction the automate scaling of calculation was introduced, to compensate the lost of accuracy caused by fixed point usage. This work was carried out by several authors, in particular L.K.Eisymont and I.B.Zadyhajlo. A.N.Andrianov and K.H. Efimkin automated the calculation of difference schemes in NORMA system [64].

At the beginning of eighties, CA popularization and systems comparison became important, in particular for mechanical tasks which required computer experiment for their investigation and solution [5,7-8,65,66]. CA was successively used for difference schemes construction in the area of non-regularity [67]. The result of the last work was positively evaluated by A.A.Samarsky, leader in soviet mathematical modeling. With the aid of A.A. Samarsky KIAM became one of the main organizer of the First National Conference on Computer Algebra Applications in Mechanics in Gorky (Nijniy Novgorod) in 1984 [6]. The results of about 20 years research, as well as future plans and perspective directions, were discussed.

To generalize the experience of common work of mathematicians, mechanicians and programmers, some classification work was done by G.B.Efimov and M.V.Grosheva [68-71]. There were reviews of CA systems and CA applications for mechanical problems [5,7,8,66,72-75]. Tables of CA systems features were presented for users [5,8,68,69]. These reviews provided convenient tool for CA systems comparison and selection for potential users - experts in applied areas. Such analysis was useful for CA developers as well. First CA systems were usually specialized and elaborated for concrete task solution. For example, in the area of dynamics of multi-bodies systems many various CA systems were created, so the classification and comparison of their features became important [69,71]. Later the well known modern universal systems, such as REDUCE and others, appeared and were used for various algorithms realization. For example, S.A.Gutnik, M.Yu.Shashkov and others used REDUCE [31,76,85-87,90]. Currently CA is used as necessary standard tools in big program complex, often without any special mention.

A number of problems were investigated and solved in KIAM with aid of CA during about 25 years. The investigation of difference schemes was continued [45,67], partly in contact with the followers of N.N.Yanenko [75]. In the area of fluid and gas dynamics some works were carried out. Several Hydrodynamics problems were resolved by I.B.Tshenkov and Ya.M. Kajdan with aid of REFAL-based CA program and by M.Yu.Shashkov and L.N.Platonova in REDUCE [76,77]. In the area of dynamics of complicated multi-bodies systems (robotics, spacecraft) CA is used for deriving the equations of motion, stability investigation, automatic generation of program of numerical analysis, and others. The PAS and others systems were created which was used for deriving the equations of motion and for solving some problems of the theory of control [78-82]. Computer Algebra applications to mechanical education were developed by D.Yu.Pogorelov, the follower of V.V.Beletsky [83,84]. Reviews of the works, concerning CA application in mechanics, dynamics of multi-bodies systems and theory of control were issued [5,7,8,66,72-75].

Most important results of last years at Keldysh Institute are obtained by A.D.Bruno and his followers. It concerned some algorithms of normalization in Hamilton systems and Newton polyhedra investigation [85-90]. S.Yu.Sadov and V.P.Varin investigated stability of motion for celestial mechanics problems [86,87,90]. There are several publications devoted to the history of CA and its applications in KIAM [13,91-93].

References

Almost all publications are in Russian, except noticed by "*". The "Cosmic Research" and other Journals were translated in English since about 1975. At some numbers the references to Russian Reference Journal in Mathematics and Mechanics was done - RJ Math., RJ Mech.

1. Calculus mathematics and computer facilities. All-Union seminar. - Inst. of Low Temperature Physics AS UkrSSR, 1972, N 3, 152 pp.

2. Analytical calculations on the Computer and their application in theoretical Physics. Materials of International conference. Dubna, 1979. - Dubna: JINR, 1980, 187 pp. 1983, 260 pp. 1985, 420 pp.

3. Gerdt V.P., Tarasov O.V., Shirkov D.V.- Uspehi Phys. Nauk, 1980, v. 30, N 1, 113-147.

4.* Miola A.M., Pottosin I.V. - A bibliography of soviet works in algebraic manipulations. - SIGSAM Bull., 1981, v. 15, N 1, 5-7.

5. Systems of analytical calculations on the Computer (Applied Analytical Packages). Materials of seminar of Institute Mechanics MSU, 1981. - Informator. Kedysh Institute of Applied Mathematics AS USSR, 1983, N 1, 65 pp.

6. Systems for Analytical Manipulation in Mechanics. Thesis the Reports of All-Union Conference. Gorky, 1984. - Gorky: GSU, 1984, 147 pp.

7. Applied Program Packages. Analitical manipulations. - M., Nauka, 1988. 156 pp.

8. Grosheva M.V., Efimov G.B. Systems for analytical calculation in Computer. - In: Applied programs pakages. Analytical manipulations. M., Nauka, 1988, 5-30.

9.* Brumberg V.A., Tarasevich S.V., Vasiliev N.N. Specialized Celestial Mechanics systems for Symbolic Manipulations.- Celestial Mech., 1988/89, v. 45, N 1-3, 145-162.

10. Klimov D.M., Rudenko V.M. Computer Algebra Method's in Mechanical problems. - M., Nauka, 1989. 215 pp.

11.* Computer Algebra in Physical Research. Int. Conf. Computer Algebra Phys. Res., Dubna, USSR, 1990. Memorial Volume for N.N.Govorun. - World Scientific, 1991, 453 p. Thesis, Dubna-1990. - Dubna: JINR, 1990.

12. Abramov S.A., Zima E.B., Rostovzev V.A. Computer Algebra. - Programirovanie, 1992, N 5, 4-25.

13. The History of Computer Algebra Applications. Session. 4-the Intern. IMACS conference on Applications of Computer Algebra.- IMACS ACA'98, Prage, Aug.9-11, 1998.- M., 1998, 35 pp.

14. A.A.Dorodnitcin. Solution of mathematical and logical tascs in computer. All-Uniom Confer. Moscow, 12-17 March 1956. Plenary reports. - VINITI. M., 1956.

15. Okhozimsky D.E. Cosmic Research. 1964, v.2, N 6, 817-842.-. - RJ Mech., 1965, 11A29

16. Kantorovich L.V. About one mathematical symbolic system convinient for computer calculations. Doklady AS USSR, 1957, v. 113, N 4, 738-739; Kantorovich L.V., Petrova L.T. 3th All-Union Mathemat. Congress. 1956. Rep. V.2, M., 151; Smirnova T.N. Trans. Steklov Mathematical Institute. - M.- L., 1962.; Pervozvanskaya T.N. - Ibid.; Smirnova T.N. Polinomial PRPRAB. - L., Nauka. 1967.

17. Stogny A.A. In.: Problems of Cybernetic. - M., Nauka, 1962 N 7, 189-200.

18. Shurygin V.A., Yanenko N.N. In.: Problems of Cybernetic. - M., Nauka, 1961 N 6, 33-43.- RJ Math., 1962, 7B332.

19. Brumberg V.A. Bull. ITA AS USSR, 1963, v.9, N 4, 234-256; Brumberg V.A. Reports of Inst. Theor. Astronomy (ITA AS USSR). L., 1966, N 11, 3-88; Polozova N.G., Shor V.A. Probl. of motion artif. spacecrafts of Celestial bodies. - M., 1963.

20. Kabulov V.K. Doklady AS UzbekSSR, Tashkent, 1963, N 9; Tolok V.A. In.: Probl. comp. Math. & Thechn. Tashkent, 1964, N 3; Kabulov V.K. Algorithmization in elasticity and deformational plasticity theory.- "Fan" AS UsbekSSR, Tashkent, 1966.

21. Efimov G.B. Cosmic Research.1970, v.8, N 1, 26-47.- RJ Mech., 1970, 7A28; KIAM Publication. 1970. 110 pp.

22. Markeev A.P. KIAM Publication. 1970. 163 pp.

23. Markeev A.P., Sokolsky A.G. KIAM Preprint N 10. 1975.

24. Markeev A.P., Sokolsky A.G. KIAM Preprint N 31. 1976. RJ Math., 1976, B1097.

25. Markeev A.P., Sokolsky A.G. Reports of Inst. Theor. Astronomy. L., 1978, N 17, 62-68.

26. Sokolsky A.G., Shevchenko I.I. Preprint N 8 Inst. Theor. Astronomy. L., 1990.

27. Sokolsky A.G. Contemporant Problems of Computer Algebra in Celestial Mechanics - 7 All-Union Congress of Ther. & Applied Mechanics, M., 1991. Abstract. M., 1991, p. 323. RJ Mech., 1991, 12A125.

28. Sokolsky A.G. On the Problems of Computerization of Celestial Mechanics.- Computer Algebra Applications. Intrn. Worcshop ISSAC Kiev, Ucrain, July 1993.- Abstract. S.-Peterburg. 1993. pp.41-45.

29. Sokolsky A.G., Vakhidov A.A., Vasiliev N.N. Development of a Motion Theory for Satellites with large eccentricities by means of Computer Algebra.-Russia-Romania Simposium. Bucurest, 7-12 sent. 1995. Abstract., S.-Petersburg. 1995. 39.

30. Veretennikov V.G., Markeev A.P., Karpov I.I. et all. Theoretical Mechanics. Generation and Analysis of equetion of motions.- M., "Visshaya shkola", 1990.

31. Sarychev V.A., Gutnik S.A. Cosmic Research.1984, v.22, N 3, 323-326.

32. Sarychev V.A., Gutnik S.A. KIAM Preprint N84 1990.

33. Dovjenko Z.P. Diploma. Moscow University (lided by Lidov M.L.) 1963.

34. Lidov M.L., Lyachva V.A., Solovyev A.A. Cosmic Research.1975, v.13, N 3, 335-346.- KIAM Preprint N 69-70. 1974.

35. Lidov M.L. Reports of Inst. Theor. Astronomy. Leningrad, 1978, N 17, 54-61.

36. Gordeeva Yu.F. KIAM Preprint N 26-27. Lidov M.L., Gordeeva Yu.F. Cosmic Research.1974, v.12, N 4, 491.

37. Lidov M.L., Solovyev A.A. Cosmic Research.1978, v.16, N 6, 806; Cosmic Research.1975, v.13, N 3, 303.- KIAM Preprint N 69-70. 1974.

38. Vashkovyak M.A. Cosmic Research.1972, v.10, N 2, 147-158; Cosmic Re-search.1983, v.21, N 6, 819. Cosmic Research.1985, v.23, N 3, 335-346.

39. Turchin V.F. Metaalgortmic Langvich. - Cybernetika. 1968, N4, 45-54. RJ Math., 1971, 2V86.

40. Turchin V.F., Serdobolsky V.I. - Cybernetika. 1969, N3, 58-62.

41. Turchin V.F. KIAM Preprint N41,43,44,48,49. 1971. RJ Math., 1972, 1V993-996.

42. Turchin V.F. - In.: Calculus mathematics and computer facilities. All-Union seminar. - Inst. of Low Temperature Physics AS UkrSSR, 1972, N 3, 28.

43. Budnik A.P., Turchin V.F., Tshenkov I.B. et all. Nuclear Phys., 1971, v.14, N2, 304-314.

44. Tshenkov I.B. In.: Systems for Analytical Manipulation in Mechanics. Thesis the Reports of All-Union Conference. Gorky, 1984. - Gorky: GSU, 1984, 47-49.

45. Tshenkov I.B. KIAM Preprints N 1, 7, 21 1989, N 14, 15 1991.

46. Shashkov M.Yu., Tshenkov I.B. KIAM Preprint N 48 1983, N 23 1985.

47. Topunov V.L. In.: Applied Math. and Math. Phys. M., 1975, N 2, 158-172, 148-152.

48. Arais E.A., Shapeev V.P., Yanenko N.N. Doklady AS USSR, 1974, v. 214, N 4, 737-738. RJ Math., 1974, 6V1149. Ganja V.J., Meleshko S.V., Murzin F.A., Shapeev V.P., Yanenko N.N. Doklady AS USSR, 1981, v. 261, N 5, 1044-1046. RJ Math., 1982, 4B1206.

49. Provorov L.V., Shtarkman V.S. KIAM Preprint N 61, 166 1982. Provorov L.V. et all. Bauman Moscow Technical University. M., 1985.

50. Klimov D.M., Gorodetsky O.M., Korlyukov O.M. Inst. Probl. Mach. AS USSR. Preprint N 233, 1985, 55 pp.

51. Belous L.F. In.: Systems for Analytical Manipulation in Mechanics. Thesis the Reports of All-Union Conference. Gorky, 1984. - Gorky: GSU, 1984, 21-22.

52. Zadyhajlo I.B., Kamynin S.S., Lyubimsky E.Z. - KIAM Preprint N 1971.

53. Zadyhajlo I.B., Myamlin A.N., Smirnov V.K., Eisymont L.K. - In: Atificial Intelligence. Result & Perspective.- M. 1974. 157-165. Zadyhajlo I.B., Kotov E.I., Krasovsky A.G., Myamlin A.N., Smirnov V.K. KIAM Preprint N 15 M., 1975.

54. Zadyhajlo I.B., Kotov E.I., Myamlin A.N., Pozdnyakov L.A., Smirnov V.K. - KIAM Preprint N 41 M., 1975.

55. Eisymont L.K. - Control Systems & Computers. 1977, N 2, 56-64.

56. Eisymont L.K., Platonova L.N. - In.: Analytical calculations on the Computer and their application in theoretical physics. Materials of International conference. Dubna, JINR, 1980, p. 19-33.

57. Myamlin A.N., Zadyhajlo I.B., Smirnov V.K. - In.: All-Union Conference. Parallel Programming. Part 1. Novosibirsk: Comp.Centr.Sib. Dep. AS USSR.1980. 54-63.

58. Zadyhajlo I.B., Eisymont L.K. Systems for Analytical Manipulation in Mechanics. Thesis the Reports of All-Union Conference. Gorky, 1984. - Gorky: GSU, 1984, 23-24.

59. Myamlin A.N., Smirnov V.K., Golovkov S.L. A Specialized Symbol Processor. Fifth Generation Architecture. - Ed. by J.V.Woods. New-Holland, 1986.

60. Myamlin A.N., Smirnov V.K., Golovkov S.L. A specialized processor ES-2702. - In: Elaboration of Computer of new generation. - Ed. by A.P.Ershov, V.E.Kotov. Novosibirsk, Comp. Centr. Sib. Dep. AS USSR, 1986, p.39-47.

61. Smirnov V.K., Topunov V.L., Shishkov D.L. - KIAM Preprint N 62, 1991.

62. Bychkov S.P., Horoshilov A.I. et all. - KIAM Preprint N 118, 1975.

63. Bychkov S.P., Zelenetsky S.D., Fisun V.A., Horoshilov A.I. - KIAM Preprint N 118, 1976.

64. Andrianov A.N., Efimkin K.N., Zadyhajlo I.B. - In: Langviche & parallele Computers. M.: Nauka. 1990. 3-37; - Programmirovaniye. 1991, N2, pp. 80-94.

65. Samarsky A.A., Shashkov M.Yu. - In.: Systems for Analytical Manipulation in Mechanics. Thesis the Reports of All-Union Conference. Gorky, 1984. - Gorky: GSU, 1984, 23-24; In: Optimization & Simulation in SAPR. Gorky, 1985, 33-41.

66. Efimov G.B. Analytical Manipulations in cycle of Computer experiments. - Ibid. 58-63.

67. Efimov G.B., Tishkin V.F., Shashkov M.Yu., Tshenkov I.B. KIAM Preprint N 20 1982;- In.: Analytical calculations on the Computer and their application in theoretical physics. Materials of International conference. Dubna, JINR, 1983. pp. 74-84.

68. Grosheva M.V., Efimov G.B. In.: Applied Program Package. Nauka. Novosibirsk, 1983. pp. 174-182.

69. Efimov G.B. Systems of Symbolic manipulations and it's classification. - Preprint of Leningrad branch of Institute of Machines-management AS USSR, N 16,1989, pp. 5-6.

70. Efimov G.B., Zueva E.Yu. Computer analogy and Symbolic manipulations.- Algorithms and Programs in Celestial mechanics. Thesis reports of All-Union Work-shop. Leningrad. - Inst. of Theor. Astronomy, 1991, pp. 63-64.

71. Efimov G.B., Grosheva M.V. - On Using Computer Algebra in Mechanics. - Computer Algebra in Scientific Computing. Intern. Conference, St.Petersburg, April 20-24,1998. Extended abstracts. St.Petrsb. 1998, 37-40.

72. Grosheva M.V., Efimov G.B. On using symbolic manipulations in Mechanics.- Computer Algebra Applications. Intrn. Worcshop ISSAC Kiev, Ucrain, July 1993. Abstract. S.-Peterburg. 1993. P. 11-14.

73. Grosheva M.V., Efimov G.B., Samsonov V.A. On using symbolic manipulations in Mechanics.- Computer Algebra Applications. Intrn. Worcshop ISSAC Gamburg, Germany, Oct. 1995. Abstract. S.-Peterburg. 1995. p. 11-14.

74. Grosheva M.V., Efimov G.B., Samsonov V.A. - Simbolic Manipulations in Control Theory. - Izvestiya Akademii Nauk. The Control Theory & Systems, 1998, N 3, 80-91.

75. Valiullin A.N., Ganja V.J., Murzin F.A., Shapeev V.P., Yanenko N.N. Inst. Appl. & Theor.Mechanics. Sib.Dep. AS USSR, Preprint N 7, Novosibirsk, 1981. Valiullin A.N., Ganja V.J., Ilyin V.P., Shapeev V.P., Yanenko N.N. Doklady AS USSR, 1984, v. 275, N 3, 528-532. RJ Math., 1984, 3G341.

76. Platonova L.N., Shashkov M.Yu. - KIAM Preprint N 134 1983.

77. Kajdan Ya.M., Tshenkov I.B. - Jornal Appl.Mechan.& Technic Phys. (PMTF), N1 (173), Novosibirsk, 1989, 41-49.

78. Sklyarenko (Zueva) E.Yu. Diploma. (leaded by Sadov Yu.A.). Moscow Phys. Technic. Inst., M., 1972.

79. Efimov G.B. Konikova N.S. Tes. 2-the All-Union Conference of Control Theory & Optimization in Mechanics. Kazan, 1977. 18.

80. Grosheva M.V., Efimov G.B., Konikova N.S., Samsonov V.A. Abstracts. Aii-Union Work-shop in Robotic systems.Vladimir-1978. M., 1978, 196.

81. Klimov D.M., Efimov G.B., Rudenko V.M., Samsonov V.A. - In: Optimization & Simulation in SAPR. Gorky, 1985,3-33.

82. Grosheva M.V., Efimov G.B., Abdrahmanov A.T. - In.: Systems for Analytical Manipulation in Mechanics. Thesis the Reports of All-Union Conference. Gorky, 1984. - Gorky: GSU, 1984, 73-74.

83. Pogorelov D. On Coding of Symbolic Expressions by Generation of Motion Equations of Multibody Systems. - EUROMECH 343, Computer Symbolic Manipulation in Mechanics. International Colloquium. Technical University Hamburg, Hamburg, Germany, October 9-13, 1995. Abstracts, p. 21.

84. Pogorelov D.Yu., Efimov G.B. KIAM Preprint N 72. 1993.

85. Soleev A., Aranson A.B. KIAM Preprint N 36 1994.

86. Bruno A.D., Varin V.P. KIAM Preprint N 124, 128 1995.

87. Alexander D. Bruno, Victor P.Varin. The limit problems for equation of oscillations of a satellite. - Celestial Mechanics and Dynamical Astronomy, 1997, v. 67.1-40.

88. Bruno A.D., Edneral V.F., Stanly Steinberg. Foreword.- Mathematics and Computers in Simulation. 1998, v.45, p.409-411.

89. Bruno A.D. Normal Forms. - Mathematics and Computers in Simulation. 1998, v.45, p.413-427.

90. Sadov S.Yu. KIAM Preprint N 37. 1998.

91. Grosheva M.V., Efimov G.B., Samsonov V.A. On the History of use the Computer Symbolical Manipulation's in Mechanics. - Inst.of Mech. Moscow Lomonosov University, Preprint N 16-95, M., 1995. 44 pp.

92. Smirnov V.K. KIAM Preprint, M., 2001.

93. Efimov G.B., Zueva E.Yu., Tshenkov I.B. Computer Algebra in Keldysh Institute of Applied Mathematics.- Mathematical Simulation. 2001, N 6.

Conception of Universality of the Huygens Resonance Synchronization Principle and Model for Structural Pecularities of Superconducting Systems and Biomolecules

F.A. Gareev[a], G.F. Gareeva[b]

[a]BLTPH, [b]FLNPH JINR 141980 Dubna, Russia
e-mail: gareev@thsun1.jinr.ru

Schrödinger wrote that an interaction between microscopic physical objects is controlled by specific resonance laws. According to these laws, the difference between two eigenenergies (eigenfrequencies) in one system should be equal to each other: $h\nu_1 - h\nu_1' = h\nu_2 - h\nu_2'$, $\nu_1 - \nu_1' = \nu_2 - \nu_2'$, which can be generalized for a complex system as $\sum q_{ij}\nu_j = 0$ (q_{ij} is the integer number matrix). Therefore, the eigenfrequencies are additive. In other words, the resonance condition is formulated in the following way: Oscillations participating in an interaction process should be constituents of the same frequency. Thus, we come to the conclusion: In a whole interacting self-consistent wave system the hierarchy of frequencies is established. So, the sum of all partial frequencies is an integral of motion. Any interaction in a microscopic hierarchic wave system exhibits the resonance character. Due to the above-said the corresponding partial motions are determinate. As the resonance condition arises from the fundamental energy conservation law, the rhythms and synchronization of the majority of phenomena to be observed are the reflection of the universal property of self-organization of the Universe. The Huygens synchronization principle is substantiated at the microscopic level and its universality is established. The universal Huygens resonance synchronization principle is independent of substance, fields, matter and interactions for micro- and macrosystems. It is well known in optics (in quantum mechanics too) that the transition coefficient of light through the layer is equal to one if the following relations between the thickness d of layer and wave length λ exist

$$d = (n/4)\lambda, \; n = 1, 2, 3, ... \tag{1}$$

It is interesting to note that: 1)the Tomasch quantization conditions for tunneling are the same as (1), 2)the Bohr quantization conditions $\lambda_N = N\lambda_e$ for a hydrogen atom and the quantization conditions $\lambda_N = N\lambda_{^4He}$ for superfluid ^4He coincide with (1) if $N = n/4$. We carry out a systematic analysis of interatomic distances for a huge number of systems using (1) in which $\lambda = \lambda_e$ is the electron wave length in the ground state of hydrogen atoms. We come to the conclusion that the superconductivity can be explained by the assumption: channel motions in such systems and electron motion in the ground state of a hydrogen atom are exactly synchronous. Therefore, superconductive systems represent coherent

synchronized states — complex of coupled resonators with the commensurable frequencies. The parameter-free formula for interatomic distances in biomolecules, superconductors and sizes of nanostructures has been obtained which establishes some bridge between the structures of different phenomena (conductivity, superconductivity, insulator-metal transitions, quantum Hall effect, superfluidity, quantization of nanostructure cluster sizes, sizes of biomolecules). This connection can be considered as indication of existence of the same physical phenomena in the structures of the superconducting and living systems. The electron wave length in these systems plays a role of a standard one for distances between atoms in such systems.

Interest of our results is not only in their closeness to experimental data, but also in the derivation of a formula from the fundamental laws of Nature — the conservation law of energy-momentum and Huygens resonance synchronization principle. These observations allow us to formulate a strategy of experimental searches for a new class of high-temperature superconductors. Thereby, we bring some arguments in favour to mechanism — ORDER from ORDER, declared by Schrodinger.

It means that our method can be used in a different fields of the fundamental research and also in applications: construction of a new materials, say, high-temperature superconductors (control T_c), medicine, a new devices in analogy with biomolecules.

Analytic Calculations for Some ODE with Quadratic Nonlinearity by Continuous-Group Methods and Vector-Field Analysis

A.G. Galperin, V.M. Dubovik, V.S. Richvitsky

Laboratory of Information Technologies,
Laboratory of High Energies,
Joint Institute for Nuclear Research, 141980 Dubna,Russia;
e-mail: rqvtsk@cv.jinr.ru

Analytic calculations allow a systematic analysis of various kinds of the Lotka-Volterra ODE together with classical systems (the Euler top, etc.) by the continuous ODE group calculation and the vector field analysis. Owing to the equations being homogeneous (or by restating the equation to a similar one), the analysis of a vector field leads to the graphic representation of its behavior on a sphere at all the characteristic points.

1 The Model

The system of equations

$$\begin{aligned} \dot{x} &= x(az - by), \\ \dot{y} &= y(bx - cz), \\ \dot{z} &= z(cy - ax) \end{aligned} \tag{1}$$

was introduced in the paper [1] as a model of the pre-election fight of two parties with participation of non-party people. As was discovered further, the system also describes the lines of the potential level in Henon-Hayles equations [8], the boundary of the 3-particle decay diagram [9], the formal kinetic [3] of chemical (abstract autocatalytic) reactions.

$$\begin{aligned} z + x &\xrightarrow{a} 2x, \\ x + y &\xrightarrow{b} 2y, \\ y + z &\xrightarrow{c} 2z. \end{aligned} \tag{2}$$

System (1) has essential features as compared with the Lotka-Volterra system of predator-prey on the one hand, and the Euler top equations on the other hand, in spite of great external similarity. The conservation law for the sum of the components is built in the system according to the sense of the model.

For instance, interpretation of the discussed model as that of the pre-election fight leads us to recognition that the non-party electors could be considered as the third party. We postulate that the number of participants of the election campaign is constant, considering that the latter is short-term and without mortality.

We are interested in studying discrete and continuous symmetries of the system (1) and the full classification of its solutions in [5] (at the rate the authors succeeded in it). It is noticeable that the considered system, unlike others (for example, the Lotka - Volterra system of predator-prey) could not have limit cycles, i.e. self-oscillating solutions, but contains systems of cycles only.

2 First integrals and singular points

Further study of the system is based on the knowledge of the first integrals

$$\begin{array}{c} I_1 = x + y + z = A, \\ I_2 = x^c y^a z^b = C \end{array} \tag{3}$$

and the analysis of its behavior in the vicinities of the singular points.

The homogeneity of the system (1) in phase variables and parameters allows one to determine the first continuous symmetry - the similarity of the phase curves (homothety) with the group generator

$$x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y} + z\frac{\partial}{\partial z} \tag{4}$$

and their independence of homothety of the space of parameters with the generator of the group

$$a\frac{\partial}{\partial a} + b\frac{\partial}{\partial b} + c\frac{\partial}{\partial c}. \tag{5}$$

This allows one to make two conclusions, leading to two different approaches.

First, it is enough to consider phase space in two invariant planes: $I_1 = x + y + z = 1$ and $I_1 = x + y + z = 0$. Second, it is natural to switch to homogeneous coordinates in both the spaces, i.e. to the projection spaces.

2.1 Evolution on the invariant planes

If we rewrite the system (1) as follows

$$\dot{\boldsymbol{r}} = \boldsymbol{F}(\boldsymbol{r}), \tag{6}$$

where

$$\begin{array}{c} \boldsymbol{r} = (x, y, z)^T, \\ \boldsymbol{F}(\boldsymbol{r}) = (axz - bxy, byx - cyz, czy - azx)^T \end{array} \tag{7}$$

is the list of the singular points, the solution of the system of equations

$$\boldsymbol{F}(\boldsymbol{r}) = 0 \tag{8}$$

easily comes into the system of analytic calculations as well as the solution of the system of the characteristic equations

$$det\left(\frac{d\boldsymbol{F}}{d\boldsymbol{r}} - \lambda I\right) = 0, \tag{9}$$

in the vicinities of the mentioned singularities.

The solutions of the characteristic equations

$$\begin{gathered}\{x=A,y=0,z=0\},\ \lambda=\{-a,b\},\\ \{x=0,y=A,z=0\},\ \lambda=\{-b,c\},\\ \{x=0,y=0,z=A\},\ \lambda=\{-c,a\},\\ \{x=\frac{c}{a+b+c},y=\frac{a}{a+b+c},z=\frac{b}{a+b+c}\},\ \lambda^2=-\frac{abc}{a+b+c},\end{gathered} \tag{10}$$

show us that there are always (in general) three singular points, two of which are of the node type, and one is of the saddle type, and also the fourth point - the centre or the saddle, which depends in what region of the state space it hits. According to this, in fact, space partition into the regions occurs. Since it is connected with the set of parameters, the space of parameters is divided into the regions exactly in the same way, and the conformity between the regions of these spaces is determined.

The knowledge of the first integrals allows us to calculate the generator of the diffeomorphism group that transforms certain phase curves on the invariant plane into the others when parameter values are constant, according to the formulae

$$(0,1,0)\,(\boldsymbol{F},\nabla_{\boldsymbol{r}}I_2,\nabla_{\boldsymbol{r}}I1)^{-1}\left(\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial z}\right)^T, \tag{11}$$

It also allows one to calculate three generators of the diffeomorphism group that connect the change of the parameters a, b, c with the change of the phase curves

$$\begin{gathered}(0,\tfrac{\partial I_2}{\partial a},0)\,(\boldsymbol{F},\nabla_{\boldsymbol{r}}I_2,\nabla_{\boldsymbol{r}}I1)^{-1}\,(\tfrac{\partial}{\partial x},\tfrac{\partial}{\partial y},\tfrac{\partial}{\partial z})^T+\tfrac{\partial}{\partial a},\\ (0,\tfrac{\partial I_2}{\partial b},0)\,(\boldsymbol{F},\nabla_{\boldsymbol{r}}I_2,\nabla_{\boldsymbol{r}}I1)^{-1}\,(\tfrac{\partial}{\partial x},\tfrac{\partial}{\partial y},\tfrac{\partial}{\partial z})^T+\tfrac{\partial}{\partial b},\\ (0,\tfrac{\partial I_2}{\partial c},0)\,(\boldsymbol{F},\nabla_{\boldsymbol{r}}I_2,\nabla_{\boldsymbol{r}}I1)^{-1}\,(\tfrac{\partial}{\partial x},\tfrac{\partial}{\partial y},\tfrac{\partial}{\partial z})^T+\tfrac{\partial}{\partial c}.\end{gathered} \tag{12}$$

2.2 Evolution in the projection space

The equation (1) does not provide the consistent motion in the homogeneous coordinates, but the time rescaling (which depends on coordinates)

$$\dot{\boldsymbol{r}}=\boldsymbol{F}_1(\boldsymbol{r})=\frac{1}{\sqrt{x^2+y^2+z^2}}\boldsymbol{F}(\boldsymbol{r}), \tag{13}$$

allows us to coordinate them and thus to speak about the motion equations (13) in the projection space.

The covering manifold of a 2-dimensional projection space is the 2-dimensional sphere $x^2+y^2+z^2=1$. The vector field projection of equations (13) onto a sphere results in a projection onto a sphere of phase curves together with singularities to which fixed points corresponding to the solution asymptotes are added.

By means of analytic calculations there was obtained the vector field projection onto a sphere

$$\boldsymbol{F}_s(\boldsymbol{r})=\tfrac{1}{|\boldsymbol{r}|}(\boldsymbol{F}_1(\boldsymbol{r})-\tfrac{\boldsymbol{r}\boldsymbol{F}_1(\boldsymbol{r})}{\boldsymbol{r}\boldsymbol{r}}\boldsymbol{r}). \tag{14}$$

Coordinates and types of singular points together with the figures of the vector field in the vicinity of each singularity (when a=1, b=1, c=1) are represented in Appendix A.

3 Classification of solutions

The most usual type of a manifest representation of the solution evolution on the phase space is the phase portrait. It is obviously implied that other pictures are similar to the given one. It is most likely true only in the vicinity of the produced solution. Similarity relation - the relation of equivalence - defines the partition of the phase space into the classes of equivalence, i.e. regions, within which the phase trajectories are similar. The meaning of the similarity conception will be defined while answering specific questions.

When the parameters are present, the similarity of the phase curves is asserted in the same way, at least, for a certain vicinity of the parameter space. The mentioned similarity is also the ratio of equivalence in the space of parameters, and it divides the latter into the regions - classes of equivalence of the similar behavior of the system. Thus, the ensembles of systems are chosen.

When looking at the system (1), one can notice that the discriminant set a,b,c, a+b+c [1] of the population (10) of types of all the singular points gives the parameter space partition into the regions (it is easy to see that there are 14 of them).

The coordinates of the fourth singularity from the population (10) naturally set one-to-one correspondence between the space of parameters a:b:c and the phase space x:y:z. Thus, partition of the space of parameters into the regions is transferred on the phase space.

One can notice that the phase curves in the phase space turn out to be kept within one of the regions (i.e. there are no curves that proceed from one region to the other, which is not obvious because of the partition construction). The latter becomes clear by virtue of that the boundaries between the regions themselves are phase curves, invariant under any parameter changes (this is verified by the direct substitution).

When defining (see above) the meaning of the similarity, we notice that the curves within each region (of the phase space) are related in the sense that when the parameters a:b:c are fixed, there exists the family of the diffeomorphisms with the generator (11), that transfers each phase curve of a certain (and each) region of the phase space into a certain phase curve of the same region; when the set of the parameters a:b:c is changed continuously within the region (of the parameters), there exists the family of the diffeomorphisms with the generator of the group (12).

The relationship of the phase curves of various regions is revealed in the sense that the diffeomorphism of the parameter space conserving the partitions will be the diffeomorphism of the phase curves in case the latter is a composition of the permutation of phase space regions and the diffeomorphisms of the phase curves within the regions described earlier. In this case, the permutation of regions of the phase space is induced by one of the regions of the parameter space so that the singularity types remain constant in mapping.

There are fixed points corresponding to the asymptotes of the initial phase space in the projection phase space x:y:z.

The vector field will be fully described if the plot of it in the vicinity of each point with the indication of the characteristic curve leading to each adjacent singularity is given. In this case, the latter task is simplified by that the separatrixes are characteristic in the

[1] Each time when one of the variables a, b, c or the sum a+b+c alternates, the type of one of the particular points is changed.

case of a saddle (at the fourth point), and so are the boundary phase curves (straight lines in our case) that are stable at any parameter values and connect the rest points - knots and saddles.

4 Appendix A

The projection interpretation of the vector field of the equations (1) has singular points on a sphere and the corresponding characteristic equations

$$\begin{array}{cc} 0:0:1 & (\lambda-a)(\lambda+c) \\ 0:1:0 & (\lambda-c)(\lambda+b) \\ 1:0:0 & (\lambda-b)(\lambda+a) \\ c:a:b & \lambda^2+abc(a+b+c) \\ -1:0:1 & (\lambda\sqrt{2}+a)(\lambda\sqrt{2}+(a+b+c)) \\ -1:1:0 & (\lambda\sqrt{2}-b)(\lambda\sqrt{2}-(a+b+c)) \\ 0:-1:1 & (\lambda\sqrt{2}-c)(\lambda\sqrt{2}-(a+b+c)). \end{array} \tag{15}$$

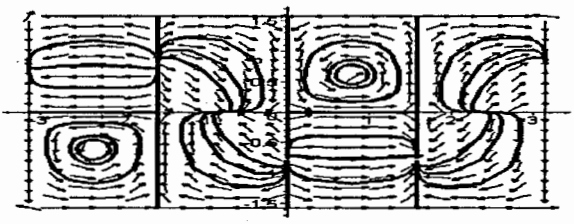


Figure 1: The development of the sphere. Phase portraits when a=1, b=1, c=1

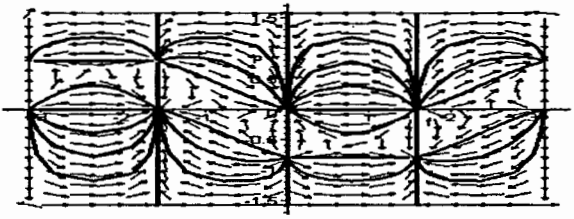


Figure 2: The development of the sphere. Phase portraits when a=1, b=-1, c=1

It is obvious that the discriminating expressions that divide the parameter sphere $a^2+b^2+c^2=1$ into the regions of different behavior of the system are a,b,c, a+b+c.

5 Appendix B

Euler equations

$$\begin{aligned} a\dot{x} &= (b-c)yz, \\ b\dot{y} &= (c-a)xz, \\ c\dot{z} &= (a-b)xy \end{aligned} \tag{16}$$

are of the same homogeneity as (1) and could be studied by the same methods. In particular, there are groups of the similarity of the phase space and the parameter space with the generators of the groups (4) and (5).

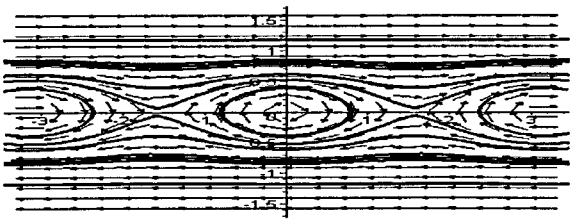


Figure 3: The development of the sphere. Phase portraits when a=1, b=2, c=3

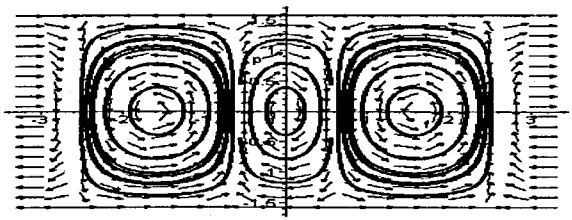


Figure 4: The development of the sphere. Phase portraits when a=1, b=2, c=1.5

The projection interpretation of a vector field of the equations (16) has the following singular points

$$\begin{aligned} 0:0:1 \quad & \lambda^2 + \tfrac{(b-c)(a-c)}{ab} \\ 0:1:0 \quad & \lambda^2 + \tfrac{(b-c)(b-a)}{ac} \\ 1:0:0 \quad & \lambda^2 + \tfrac{(a-c)(a-b)}{bc} \end{aligned} \tag{17}$$

It is obvious that the discriminating expressions that divide parameter sphere $a^2 + b^2 + c^2 = 1$ into the regions of the diverse behavior of the system are a, b, c, a-b, b-c, a-c.

The first integrals of the equations are

$$\begin{aligned} I_1 &= a(c-a)x^2 - b(b-c)y^2 = C_1, \\ I_2 &= b(a-b)y^2 - c(c-a)z^2 = C_2, \end{aligned} \tag{18}$$

that allow one to calculate the generators of the groups of the phase curve similarity according to formulae (11) and (12).

References

[1] A.G. Galperin, V.M.Dubovik, V.S.Richvitsky. Nonlinear dynamic model of a 3-component system conserving total number of objects. Int Conf. CMCP96, JINR, D5 11-96-300, p69, Dubna 1996.

[2] A.G. Galperin, V.M.Dubovik, V.S.Richvitsky. Nonlinear dynamic model of evolution of the 3-component system conserving total number of objects. P4-97-340, Dubna, 1997, in Russian .

[3] V.M.Dubovik, A.G. Galperin, V.S.Richvitsky. The correspondence between chemical schemes and dynamical Lotka-Volterra systems. E4-97-417, Dubna, 1997.

[4] V.M.Dubovik, A.G. Galperin, V.S.Richvitsky, S.K. Slepnyov. The conditions of existence of first integrals and hamiltonian structures of the Lotka-Volterra equations. Comment on some of the resent papers. E4-97-416, Dubna, 1997.

[5] V.S.Richvitsky, A.G. Galperin, V.M.Dubovik. The full classification (atlas) of the solutions of some 3D-equation of Lotka-Volterra with the skew-symmetric matrix of interactions. P4-98-42, Dubna, 1998, in Russian.

[6] V.M.Dubovik, A.G. Galperin, V.S.Richvitsky, S.K. Slepnyov. The conditions of existence of first integrals and Hamiltonian structures of Lotka-Volterra and Volterra systems, Phys. Atom.Nucl, 2000, v.63, N4, pp.629-634 (from 'Yad. Fiz', 2000, v.63, N4, pp.695-700).

[7] V.M.Dubovik, A.G. Galperin, V.S.Richvitsky. Around Lotka-Volterra Kind Equations and Nearby Problems. Nonlinear Phenomena and complex systems (NPCS) N4 V3 2000.

[8] M.Hénon, and C.Heiles, Astronomical J.,**69**, p.73-79 (1964). "The Applicability of The Third Integral Of Motion: Some Numerical Experiments",

[9] V.B.Berestetsky, E.M.Livshits, P.P.Pitaevsky. Relativistic quantum theory. Moscow, Nauka, 1971 (in Russian)

Janet Bases of Toric Ideals

Vladimir P. Gerdt[a], Yuri A. Blinkov[b]

[a]Laboratory of Information Technologies,
Joint Institute for Nuclear Research,
141980 Dubna, Russia;
e-mail: gerdt@jinr.ru
[b]Department of Mathematics and Mechanics,
Saratov State University,
410071 Saratov, Russia;
e-mail: blinkovua@info.sgu.ru

In this paper we present a version of the general polynomial involutive algorithm for computing Janet bases specialized to toric ideals. The relevant data structures are Janet trees which provide a very fast search for a Janet divisor. We broach also efficiency issues in view of application of the algorithm presented to computation of toric ideals.

1 Introduction

We consider the problem of computing a Janet basis of a *toric* ideal $\mathcal{I}_\mathcal{A}$ in $\mathbb{K}[\mathbf{x}] \equiv \mathbb{K}[x_1, \ldots, x_n]$ generated by binomials of the form [1]

$$\mathcal{I}_\mathcal{A} = \{\, \mathbf{x}^\mathbf{u} - \mathbf{x}^\mathbf{v} \mid \mathbf{u}, \mathbf{v} \in \mathbb{N}^n,\ \pi(\mathbf{u}) = \pi(\mathbf{v}),\ \gcd(\mathbf{x}^\mathbf{u}, \mathbf{x}^\mathbf{v}) = 1 \,\}.$$

Here $\mathbf{u}, \mathbf{v} \in \mathbb{N}^n$ and π is the semigroup homomorphism

$$\pi\ :\ \mathbb{N}^n \to \mathbb{Z}^d, \quad \mathbf{u} = \{u_1, \ldots, u_n\} \to u_1\mathbf{a}_1 + \cdots + u_n\mathbf{a}_n$$

where $\mathbf{a}_i \in \mathbb{Z}^d$ $(1 \le i \le n)$.

Given a set of binomials generating a toric ideals, the problem of constructing its Gröbner basis is usually (except small problems) rather expensive from the computational point of view [2]. In practice, for this particular problem, one typically deals with a large number n of variables and their degrees. If d is the maximal degree of the initial binomials, then the degree of a reduced Gröbner basis is bounded by [3]

$$2 \cdot \left(\frac{d^2}{2} + d\right)^{2^{n-1}}.$$

But for all that the reduced Gröbner basis is also binomial since the binomial structure is preserved during the Buchberger algorithm [4, 5]. Similarly, the involutive algorithms [6] based on the sequential multiplicative reductions of nonmultiplicative prolongations of the intermediate polynomials preserve the binomial structure. The output involutive basis which is also a Gröbner basis though generally redundant.

Thus, unlike construction of reduced Gröbner bases or involutive bases for general polynomial ideals, the integer arithmetics which may take most of computing time is not important for binomial ideals. In this case a fast search of monomial divisors for performing reductions of S-polynomials may become crucial in acceleration of computations.

Recently [8, 8] we designed and implemented involutive algorithms specialized to constructing Janet bases of monomial and polynomial ideals. Janet division as well as any other involutive division [6] provides uniqueness of an involutive divisor in a polynomial set with co-prime leading monomials. This allows one to organize a very fast search for a Janet divisor using special data structures for intermediate polynomial sets called Janet trees.

The main goal of this paper is to discuss the issue of practical efficiency in computing Janet bases of toric ideals based on the use of Janet trees. Since one of the most important applications of toric ideals is integer programming we shortly describe this application [9] in the next section.

2 Toric Ideals and Integer Programming

Let $\mathcal{A}$ be a matrix of dimension $m \times n$ with integer entries and $\mathbf{b} \in \mathbb{Z}^m$, $\mathbf{c} \in \mathbb{Z}^n$ be vectors. The following optimization problem

$$\min\{\, \mathbf{c}^T\mathbf{x} \mid \mathbf{x} \in \mathbb{N}^n,\, \mathcal{A}\mathbf{x} = \mathbf{b} \,\}$$

is called a problem of *integer programming*.

We shall assume that $\mathbf{c} \in \mathbb{N}^n$. If there exists vector $\mathbf{x}_0$ satisfying $\mathcal{A}\mathbf{x}_0 = \mathbf{b}$, $\mathbf{x}_0 \in \mathbb{N}^n$, then the problem of finding minimum of function $\mathbf{c}^T\mathbf{x}$ can be reduced to all kinds of transformation of the initial vector state $\mathbf{x}_0$ using $\ker(\mathcal{A})$.

The problem of determining $\ker(\mathcal{A})$ can be formulated in terms of toric ideals. Indeed, every vector $\mathbf{u} \in \ker(\mathcal{A})$ may be uniquely represented as $\mathbf{u} = \mathbf{u}^+ - \mathbf{u}^-$ where both $\mathbf{u}^+$ and $\mathbf{u}^-$ are nonnegative and have disjoint support. Associate symbol v_i with the i-th column of matrix $\mathcal{A}$. Then the ideal

$$\mathcal{I}_\mathcal{A} = \{\, \mathbf{v}^{\mathbf{u}^+} - \mathbf{v}^{\mathbf{u}^-} \mid \mathbf{u}^+ - \mathbf{u}^- = \mathbf{u} \in \ker(\mathcal{A}) \,\}$$

associated with $\ker(\mathcal{A})$ is toric. Given the initial solution $\mathbf{x}_0$, the optimal solution can be found as follows [9]:

1. Construct a basis of the toric ideal $\mathcal{I}_\mathcal{A}$.
2. Construct a reduced Gröbner basis or an involutive basis of $\mathcal{I}_\mathcal{A}$ with respect to the admissible monomial ordering $\succ_\mathbf{c}$ generated by vector $\mathbf{c}$.
3. Reduce monomial $\mathbf{v}^{\mathbf{x}_0}$ modulo the constructed basis to obtain the optimal solution.

Therefore, the reduced Gröbner basis or any involutive basis of the associated toric ideal $\mathcal{I}_\mathcal{A}$ provide an algorithmic tool for solving the problem of integer programming.

3 Janet Bases of Toric Ideals

3.1 Definition of Janet Basis

In our papers [6] the Gröbner bases of special type, called *involutive* and based on the concept of *involutive division* were introduced. Given a set of coprime monomials and an involutive division, any monomial may have at most one involutive divisor in the set. This property of the involutive division allows one to design an efficient search for the involutive divisor using the method of separative monomials [10] for a general involutive division or Janet trees [7] for Janet division.

Because of a larger number of variables and unimportance of integer arithmetical operations over coefficients of the binomials, the practical complexity of an algorithm for construction of Gröbner or Janet bases is caused by an enormous number of binomials arising in computation of the basis. A faster search for divisors may accelerates the computation substantially.

By definition of Janet division [6] (which formalizes the pioneering ideas of Janet [11]) induced by the order

$$x_1 \succ x_2 \succ \ldots \succ x_n \tag{1}$$

on x, a polynomial set F is partitioned into the groups labeled by non-negative integers $d_1, \ldots, d_i$:

$$[d_1, \ldots, d_i] = \{\ f\ \in F \mid d_j = \deg_j(\mathrm{lm}(f)),\ 1 \le j \le i\ \}$$

where $\deg_i(u)$ denotes the degree of x_i in monomial u and $\mathrm{lm}(f)$ denotes the leading monomial of f. A variable x_i is called *(Janet) multiplicative* for $f \in F$ if $i = 1$ and

$$\deg_1(\mathrm{lm}(f)) = \max\{\ \deg_1(\mathrm{lm}(g)) \mid g \in F\ \},$$

or if $i > 1$, $f \in [d_1, \ldots, d_{i-1}]$ and

$$\deg_i(\mathrm{lm}(f)) = \max\{\ \deg_i(\mathrm{lm}(g)) \mid g \in [d_1, \ldots, d_{i-1}]\ \}.$$

If a variable is not multiplicative for $f \in F$, it is called *(Janet) nonmultiplicative* for f. In the latter case we shall write $x_i \in NM_J(f, F)$. $u \in \mathrm{lm}(F)$ is a *Janet divisor* of $w \in \mathbb{M}$, if $u \mid w$ and monomial w/u contains only multiplicative variables for u. In this case we write $u \mid_J w$.

Let $\mathrm{lm}(F) = \{\ \mathrm{lm}(f) \mid f \in F\ \}$. Then a polynomial set F is called *Janet autoreduced* if each term in every $f \in F$ has no Janet divisors among $\mathrm{lm}(F) \setminus \mathrm{lm}(f)$. A polynomial h is said to be in the *Janet normal form modulo* F if every term in h has no Janet divisors in $\mathrm{lm}(F)$. In that follows $NF_J(f, F)$ denotes the Janet normal form f modulo F.

A Janet autoreduced set F is called a *Janet basis* if

$$(\forall f \in F)\ (\forall x \in NM_J(f, F))\ \ [\ NF_J(f \cdot x, F) = 0\ ]. \tag{2}$$

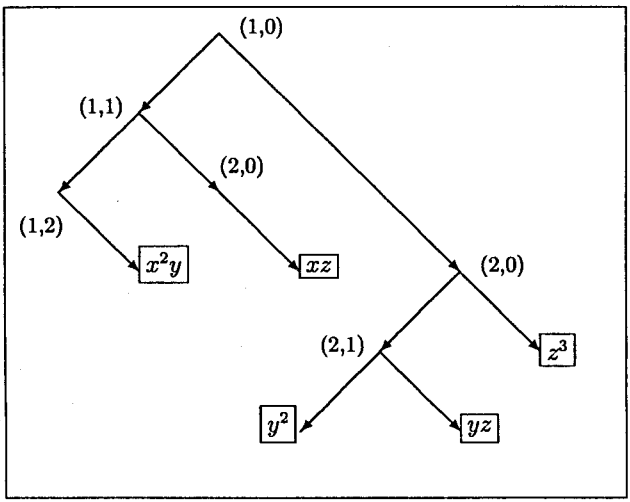
A Janet basis G is called *minimal* if for any other Janet basis F of the same ideal the inclusion $lm(G) \subseteq \mathrm{lm}(F)$ holds. If both G and F are monic this inclusion implies $G \subseteq F$. A Janet basis is a Gröbner one, though generally not reduced. However, similarly to a reduced Gröbner basis, a monic minimal Janet basis is uniquely defined by an ideal and a monomial order. In that follows we deal with minimal Janet bases only and omit the word "minimal".

3.2 Janet Trees and Search for Janet Divisor

Consider now a binary Janet tree [7] whose structure reflects the above partition of elements in U into the groups which sorted in the degrees of variables within every group. Before description of the general structure of Janet trees we explain it in terms of the concrete example [7]

$$U = \{x^2y, xz, y^2, yz, z^2\}, \qquad (x \succ y \succ z)$$

and portray it in the form of Janet tree as shown below. In doing so, the monomials in set U are assigned to the leaves of the tree. The monomial with increased by one degree of the current variable is assigned to the left child whereas the right child points at the next variable with respect to chosen ordering. In contrast to Janet tree presented in paper [7], the below tree takes into account sparseness of monomials that is inherent in toric ideals. The related information is given in pairs of integers placed in brackets where the first element represents the number of current variable and the second one represents its degree.



Consider now the structure of Janet tree of the general form as a set $JT := \cup\{\nu\}$ of internal nodes and leaves which corresponds to a nonempty binomial set. To every element ν of the tree we shall assign the set of five elements $\nu = \{v, d, nd, nv, nb\}$ with the following structure:

var(ν)	=	v	is the index of the current variable
dg(ν)	=	d	is the degree of the current variable
ndg(ν)	=	nd	is the pointer to the next node in degree
nvr(ν)	=	nv	is the pointer to the next node in variable
bnm(ν)	=	bn	is the pointer to binomial

In the absence of a child we shall assign the value *nil* to the corresponding pointer. Wherever it does not lead to misunderstanding we shall identify the pointers nd and nv with the nodes they point out. To the root of JT we assign ν_0 with $\mathrm{var}(\nu_0) = 1$ in accordance with labeling (1) and $\mathrm{dg}(\nu_0) = 0$.
The internal nodes and leaves of tree JT are characterized by the states:

$$\begin{aligned} \textbf{Internal node:} \quad & ((nv \neq nil \wedge v < \mathrm{var}(nv)) \vee (nd \neq nil \wedge d < \mathrm{dg}(nd)) \\ & \wedge\, bn = nil \\ \textbf{Leaf:} \quad & nv = nil \wedge nd = nil \wedge bn \neq nil \wedge d = \mathrm{dg}(\mathrm{lm}(bn)). \end{aligned}$$

For a fast search for Janet divisor in the given tree one can use the following algorithm **J-divisor** which is an adaptation to the above structure of Janet tree of the algorithm described in [7].

Algorithm: J-divisor(JT, w)

Input: JT, a Janet tree; w, monomial
Output: bn, a binomial such that $\mathrm{lm}(bn) \mid_J w$,
or *nil*, otherwise

```
1: ν := ν0
2: while deg_var(ν)(w) ≥ dg(ν) do
3:    while ndg(ν) and deg_var(ndg(ν))(w) ≥ dg(ndg(ν)) do
4:       ν := ndg(ν)
5:    od
6:    if nvr(ν) then
7:       ν := nvr(ν)
8:    elif ndg(ν) then
9:       return nil
10:   else
11:      return bnm(ν)
12:   fi
13: od
14: return nil
```

Apparently, the next theorem formulated and proved in [7] is valid for the adapted algorithm as well.

Theorem 1. *Let d be the maximal total degree of the leading monomials of binomials in n variables which constitute the finite set U. Then the complexity bound of the algorithm* **J − divisor** *and the binary search algorithm is given by*

$$\begin{aligned} t_{\textbf{J-divisor}} &= O(d+n), \\ t_{\textbf{BinarySearch}} &= O(n((d+n)\log(d+n) - n\log(n) - d\log(d))). \end{aligned}$$

Thus, the complexity bound for the search of Janet divisor is $O(n+d)$ where n is the number of variables and d is the maximal degree of the leading monomials in the binomial basis. Since this bound is even lower than that for the binary search algorithm, one can expect that the involutive completion of binomial ideals may be faster than the reduced Gröbner basis completion.

3.3 Algorithms for Binomial Janet Bases

Given the generating binomial set F of a toric ideal $\mathcal{I}_A$, the following algorithm **BinomialJanetBasis** which is a special form of the general polynomial algorithm [6, 8] constructs a Janet basis of $\mathcal{I}_A$.

Algorithm: BinomialJanetBasis(F, $\prec$)

Input: $F \in \mathbb{R} \setminus \{0\}$, a finite binomial set; $\prec$, an admissible ordering
Output: G, a Janet basis of the ideal generated by F

choose $f \in F$ with the lowest $\mathrm{lm}(f)$ w.r.t. $\succ$
$T := \{f, \mathrm{lm}(f), \emptyset\}$
$Q := \{\ \{q, \mathrm{lm}(q), \emptyset\} \mid q \in F \setminus \{f\}\ \}$
$Q :=$**JanetReduce**(Q, T)
while $Q \neq \emptyset$ **do**
  choose $p \in Q$ with the lowest $\mathrm{lm}(\mathrm{bin}(p))$ w.r.t. $\succ$
  $Q := Q \setminus \{p\}$
  if $\mathrm{lm}(\mathrm{bin}(p)) = \mathrm{anc}(p)$ **then**
    for all $\{\ r \in T \mid lm(\mathrm{bin}(r)) \succ \mathrm{lm}(\mathrm{bin}(p))\ \}$ **do**
      $Q := Q \cup \{r\}$; $T := T \setminus \{r\}$
    od
    $p := NF_J(\mathrm{bin}(p), T)$
  fi
  $T := T \cup \{p\}$
  for all $q \in T$ **and** $x \in NM_J(\mathrm{bin}(q), T) \setminus \mathrm{nmp}(q)$ **do**
    $Q := Q \cup \{\ \{\mathrm{bin}(q) \cdot x, \mathrm{anc}(q), \emptyset\}\ \}$
    $\mathrm{nmp}(q) := \mathrm{nmp}(q) \cap NM_J(\mathrm{bin}(q), T) \cup \{x\}$
  od
  $Q :=$**JanetReduce**(Q, T)
od
return $G := \{\ \mathrm{bin}(f) \mid f \in T\ \}$

As well as in [8] to apply the involutive criteria and avoid repeated prolongations we shall endow with every binomial $f \in F$ the triple structure

$$p = \{f,\ u,\ vars\}$$

such that

$$\begin{array}{lcl} \mathrm{bin}(p) & = & f \text{ is binomial itself,} \\ \mathrm{anc}(b) & = & u \text{ is the leading monomial of a binomial ancestor of } f \text{ in } F \\ \mathrm{nmp}(p) & = & vars \text{ is a (possible empty) subset of variables.} \end{array}$$

Here the *ancestor* of f is a polynomial $g \in F$ with $u = \mathrm{lm}(g)$ and such that $u \mid \mathrm{lm}(p)$. Moreover, if $\deg(u) < \deg(\mathrm{lm}(p))$, then every variable occurring in the monomial $\mathrm{lm}(p)/u$

is nonmultiplicative for g. Besides, for the ancestor g the equality $\mathrm{anc}(g) = \mathrm{lm}(g)$ must hold. These conditions mean that polynomial p was obtained in the course of the below algorithm **BinomialJanetBasis** from g by a sequence of nonmultiplicative prolongations. This tracking of the history in the algorithm allows one to use the involutive analogues of Buchberger's criteria to avoid unnecessary reductions.

The set *vars* contains those nonmultiplicative variables that have been already used in the algorithm for construction of nonmultiplicative prolongations. This set serves to prevent repeated prolongations.

In order to provide minimality of the output Janet basis we separate [6, 8] the whole polynomial data into two subsets which are contained in sets T and Q. Set T is a part of the intermediate binomial basis. Another part of the intermediate basis is contained in set Q together with all the nonmultiplicative prolongations of polynomials in T which must be examined in accordance to the above definition (2) of Janet bases. In so doing, after every insertion of a new element p in T all elements $r \in T$ such that

$$\mathrm{lm}(\mathrm{bin}(r)) \succ \mathrm{lm}(\mathrm{bin}(p))$$

are moved from T to Q as the **for**-loop 6-11 in algorithm **BinomialJanetBasis** does. Such a displacement guaranties that the output basis is minimal [6].

It should also be noted that for any triple $p \in T$ the set *vars* must always be a subset of the set of nomultiplicative variables for $\mathrm{bin}(p)$

$$vars \subseteq NM_J(\mathrm{bin}(p), T)\,. \tag{3}$$

In the description of algorithm **JanetBinomialBases** we use the contractions:

$$NM_J(\mathrm{bin}(p), T) \equiv NM_J(\mathrm{bin}(p), \{\mathrm{bin}(f) \mid f \in T\})\,,$$
$$NF_J(\mathrm{bin}(p), T) \equiv NF_J(\mathrm{bin}(p), \{\mathrm{bin}(f) \mid f \in T\})\,,$$

The insertion of a new polynomial in T may generate new nonmultiplicative prolongations of elements in T which are added to Q in line 16. To avoid repeated prolongations the set $\mathrm{nmp}(q)$ of Janet nonmultiplicative variables for g has been used to construct its prolongations is enlarged with x in line 17. The intersection placed in this line preserves the condition (3).

The subalgorithms **JanetReduce** and $\mathbf{NF_J}$ perform Janet reduction of polynomials in Q modulo polynomials in T and presented below. In addition to reductions in lines 4 and 19, the Janet normal form computation is placed in line 12. This is because the replacement of elements from T to Q may lead to the tail reducibility of the binomial in p. Such a reducibility may be caused by converting of some nonmultiplicative variables for binomials in T into multiplicative due to the replacement.

In subalgorithm **JanetReduce** computation of the Janet normal form h is done in line 6 for every binomial $\mathrm{bin}(p)$ in T. If h is nonzero, then line 8 checks if $\mathrm{lm}(\mathrm{bin}(p))$ was subjected by reduction. If the reduction took place $\mathrm{lm}(h)$ cannot be multiple [6] of any monomial in the set

$$\{\ \mathrm{lm}(\mathrm{bin}(g)) \mid g \in T\ \}.$$

Therefore, one has to insert the triple with h in the output set Q as shown in line 9 as h cannot have ancestors among polynomials in T and one must also examine all nonmultiplicative prolongations of h. If $\mathrm{lm}(\mathrm{bin}(p))$ is Janet irreducible modulo $\{\mathrm{lm}(\mathrm{bin}(g)) \mid g \in T\}$, then the triple $\{h, \mathrm{anc}(p), \mathrm{nmp}(p)\}$ is added to Q in line 11.

Algorithm: JanetReduce(Q, T)

Input: Q and T, sets of triples
Output: Q whose polynomials are Janet head reduced modulo T

```
1:  S := Q
2:  Q := ∅
3:  while S ≠ ∅ do
4:    choose p ∈ S
5:    S := S \ {p}
6:    h := NF_J(p, T)
7:    if h ≠ 0 then
8:      if lm(bin(p)) ≠ lm(h) then
9:        Q := Q ∪ {h, lm(h), ∅}
10:     else
11:       Q := Q ∪ {h, anc(p), nmp(p)}
12:     fi
13:   fi
14: od
15: return Q
```

Subalgorithm $\mathbf{NF}_J(p, T)$ performs the Janet reduction of a binomial $g = \mathrm{bin}(p)$ modulo polynomial set in T.

For the head reducible input binomial $\mathrm{bin}(f)$ the two criteria are verified in line 5:

- **CriterionI**(f, g) is true iff $\mathrm{anc}(f) \cdot \mathrm{anc}(g) \mid \mathrm{lm}(\mathrm{bin}(f))$.
- **CriterionII**(f, g) is true iff $\deg(\mathrm{lcm}(\mathrm{anc}(f) \cdot \mathrm{anc}(g))) < \deg(\mathrm{lm}(\mathrm{bin}(f))$.

These criteria are the Buchberger criteria [12] adapted to the involutive completion procedure. If any of the two criteria is true, then $NF(\mathrm{bin}(f), T) = 0$ [8].

It should be noted that the Janet normal form is uniquely defined and, hence, uniquely computed by the above subalgorithm. This uniqueness hold because of the uniqueness of a Janet divisor among the leading terms of binomials in T at every step of intermediate computations [6].

Algorithm: $\mathbf{NF_J}(f,T)$

Input: $f = \{\text{bin}(f), \text{anc}(f), \text{nmp}(f)\}$, a triple; T, a set of triples
Output: $h = NF_J(\text{bin}(f), T)$, the Janet normal form of the binomial in f modulo binomial set in T

```
1: G := {bin(g) | g ∈ T}
2: h := bin(f)
3: if lm(h) is Janet reducible modulo G then
4:    choose g ∈ T such that lm(bin(g)) |_J lm(h)
5:    if lm(h) ≠ anc(f) and
        CriterionI(f,g) or CriterionII(f,g) then
6:       return 0
7:    fi
8: else
9:    while h ≠ 0 and h has a term t Janet reducible modulo G do
10:      choose q ∈ G such that lm(q) |_J t
11:      h := h − q · t/lm(q)
12:   od
13: fi
14: return h
```

4 Examples

As we emphasized in Sect.3.1, in the course of involutive completion of the initial binomial generators for a toric ideal the reduction can be performed very fast due to the fast search for a Janet divisor, This fast search is provided by the use of the Janet tree structures for intermediate binomial set. Our computer experiments with C/C++ codes implementing polynomial algorithms for Janet bases [8] perfectly strengthen this theoretical fact. In particular this fast reduction in addition to suppressing swell of intermediate integer coefficients results in high computational speed observed for the benchmark collection used for testing Gröbner bases software [8]. These benchmarks, however, are not very "sparse" with respect to degrees of variables occurring in the generating set. By contrast, the generating binomial sets for toric ideals especially for those arising in integer programming problem are usually highly sparse. This may lead to much larger cardinality of a Janet basis than that of the reduced Gröbner basis and thereby annihilate the advantages of involutive reduction.

Consider the example taken from [13]

$$\mathcal{I}_A = \{\, x_0x_1x_2x_3x_4 - 1, x_2^{29}x_3^5 - x_1^{14}x_4^{20}, x_1^{39} - x_2^{25}x_3^{14} \,\}.$$

Our C++ package [8] generates the degree-reverse-lexicographical Janet basis of $\mathcal{I}_A$ with 7769 binomials whose sorting with respect to the ordering chosen gives

$$\{\, x_0x_1^3x_3x_4^{281} - x_1x_2^{280}, x_0x_2^{61}x_3^2x_4^{221} - x_1x_2^{279}, x_0x_1^2x_3x_4^{281} - x_2^{280}, \ldots, x_0x_1x_2x_3x_4 - 1 \,\}$$

where we explicitly show only three highest ranking binomials and the lowest one. The computing time on a Pentium III 700 Mhz based PC running under RedHat Linux 6.2 is 6 seconds that is noticeably larger than the running time for direct computation of the reduced Gröbner basis which contains 19 binomials only:

$$\{\ x_0{x_1}^2x_3{x_4}^{281} - {x_2}^{280}, {x_2}^{281} - x_1{x_4}^{280}, x_0{x_3}^2{x_4}^{221} - x_1{x_2}^{218}, {x_1}^2{x_2}^{219} - x_3{x_4}^{220},$$
$$x_0{x_3}^3{x_4}^{161} - {x_1}^4{x_2}^{156}, {x_1}^5{x_2}^{157} - {x_3}^2{x_4}^{160}, x_0{x_3}^4{x_4}^{101} - {x_1}^7{x_2}^{94}, {x_1}^8{x_2}^{95} - {x_3}^3{x_4}^{100},$$
$$x_0{x_1}^4{x_4}^{61} - {x_2}^{61}, {x_2}^{62}x_3 - {x_1}^3{x_4}^{60}, x_0{x_3}^5{x_4}^{41} - {x_1}^{10}{x_2}^{32}, {x_1}^{11}{x_2}^{33} - {x_3}^4{x_4}^{40},$$
$$x_0{x_2}^{26}{x_3}^{15}x_4 - {x_1}^{38}, {x_1}^{39} - {x_2}^{25}{x_3}^{14}, x_0{x_1}^{15}{x_4}^{21} - {x_2}^{28}{x_3}^4, {x_2}^{29}{x_3}^5 - {x_1}^{14}{x_4}^{20},$$
$$x_0{x_3}^{10}{x_4}^{21} - {x_1}^{24}{x_2}^3, {x_1}^{25}{x_2}^4 - {x_3}^9{x_4}^{20}, x_0x_1x_2x_3x_4 - 1\ \}.$$

Accordingly, such a computer algebra system as *Singular* [17] needs much less than 1 second to compute this Gröbner basis on the same computer.

Having ascertained this drawback of the involutive method with respect to the Gröbner basis one in computing toric ideals we designed another algorithmic approach to computing Gröbner bases [14]. This approach preserves the Janet-like tree structure and uniqueness of a divisor though underlying division is not involutive since it does not satisfy the axioms in [6]. On the other hand the resulting bases unlike Janet bases are often reduced as Gröbner bases and their cardinality is always less or equal to the cardinality of Janet bases. For toric ideals the new bases are much more compact then Janet bases. We have not implemented yet the new algorithm and so we demonstrate the compactness of its output in comparison with algorithm **BinomialJanetBasis** by the following simple example taken from [2]:

$$\mathcal{I}_{\mathcal{A}} = \{\ x^7 - y^2z, x^4w - y^3, x^3y - zw\ \}.$$

The reduced Gröbner basis and Janet basis of this toric ideal for the degree-reverse-lexicographic order induced by $x \succ y \succ z \succ w$ are

$$\{\ x^7 - y^2z, x^4w - y^3, x^3y - zw, y^4 - xzw^2\ \}$$

and

$$\{\ x^7 - y^2z, x^6y - x^3zw, x^6w - x^2y^3, x^5y - x^2zw, x^2y^4 - x^3zw^2, x^5w - xy^3,$$
$$x^4y - xzw, x^2zw^2 - xy^4, x^4w - y^3, x^3y - zw, y^4 - xzw^2\ \},$$

respectively. Their cardinalities are 4 and 11. The new basis contains 5 elements

$$\{\ x^7 - y^2z, x^4y - xzw, x^4w - y^3, x^3y - zw, y^4 - xzw^2\ \}$$

and contains only single extra element in comparison with the reduced Gröbner basis.

It should be noted that there are also a number of other efficient algorithms computing Gröbner bases of toric ideals (see, for example, [2, 15, 16]) which are differ greatly from just completion of a generating binomial set to a Gröbner basis. After implementation of our new algorithm we are planning to run the underlying code for collection of large examples given in [1, 2] and other references.

Acknowledgements

The work was supported in part by the RFBR grants 00-15-96691, 01-01-00708 and by grant Intas 99-1222.

References

[1] Sturmfels, B.: *Gröbner bases and convex polytopes.* University Lecture Series **8**, Providence, RI, American Mathematical Society, 1996.

[2] Bigatti, A.M., La Scala, R., Robbiano, L.: Computing toric ideals. *J. Symb. Comp.* **27** (1999) 351-365.

[3] Koppenhagen, U., Mayr, E.W.: An Optimal Algorithm for Constructing the Reduced Gröbner Basis of Binomial Ideals. *Proceedings of ISSAC'96*, ACM Press, 1996, 55-62.

[4] Adams, W.W., Loustaunau, P.: *An Introduction to Gröbner Bases.* Graduate Studies in Mathematics **3**, American Mathematical Society, 1994.

[5] Pottier, L.: Gröbner Bases of Toric Ideals. *Rapport de recherche* **2224** (1997), INRIA Sophia Antipolis.

[6] Gerdt, V.P., Blinkov, Yu.A.: Involutive Bases of Polynomial Ideals. *Math. Comp. Simul.* **45** (1998) 519-542; Minimal Involutive Bases. *Math. Comp. Simul.* **45** (1998) 543-560.

[7] *Gerdt V.P., Blinkov Yu.A., Yanovich D.A.* Construction of Janet Bases. I. Monomial Bases. In: *Computer Algebra in Scientific Computing / CASC'01*, V.G.Ganzha, E.W.Mayr and E.V.Vorozhtsov (Eds.), Springer-Verlag, Berlin, 2001, pp.233-247.

[8] Gerdt, V.P., Blinkov, Yu.A., Yanovich, D.A.: Construction of Janet Bases. II. Polynomial Bases. In: *Computer Algebra in Scientific Computing / CASC'01*, V.G.Ganzha, E.W.Mayr and E.V.Vorozhtsov (Eds.), Springer-Verlag, Berlin, 2001, pp.249-263.

[9] Conti, P., Traverso, C.: Buchberger algorithm and integer programming. *Proceedings of AAECC-9*, Springer LNCS **539** (1991) 130-139.

[10] Blinkov, Yu.A.: Method of Separative Monomials for Involutive Divisions. *Programming and Computer Software* **3** (2001) 43-45.

[11] Janet, M.: *Leçons sur les Systèmes d'Equations aux Dérivées Partielles*, Cahiers Scientifiques **IV**, Gauthier-Villars, Paris, 1929.

[12] Buchberger, B.: Gröbner Bases: an Algorithmic Method in Polynomial Ideal Theory. In: *Recent Trends in Multidimensional System Theory*, N.K. Bose (ed.), Reidel, Dordrecht (1985) 184–232.

[13] Morales, M.: Equations des variétés monomiales. *Preprint*, Université de Grenoble I, France.

[14] Gerdt, V.P., Blinkov, A.Yu.: Involutive-like Gröbner Bases. *In preparation.*

[15] Di Biase, F., Urbanke, R.: An algorithm to calculate the kernel of certain polynomial ring homomorphisms. *Experimental Mathematics* 4 (1995) 227-234.

[16] Hosten, S., Sturmfels, B.: GRIN: An implementation of Groebner basis for integer programming. In: *Integer Programming and Combinatorial Optimization*, Balas, E., Clausen J., (Eds.), LNCS **920**, Springer-Verlag, New York, 1995, pp.267-276.

[17] Greuel, G.-M., Pfister, G., Schöenemann, H.: *Singular: A Computer Algebra System for Polynomial Computation*, Department of Mathematics, University of Keiserslautern (2001) http://www.singular.uni-kl.de/Manual/2-0-0/.

Analysis of Constraints in Light-Cone Version of $SU(2)$ Yang-Mills Mechanics

V. P. Gerdt [a], A. M. Khvedelidze [a,b], D. M. Mladenov [c]

[a] Laboratory of Information Technologies,
Joint Institute for Nuclear Research,
141980 Dubna, Russia;
e-mail: gerdt@jinr.ru
[b] A.Razmadze Mathematical Institute,
GE-380093 Tbilisi, Georgia;
e-mail: khved@thsun1.jinr.ru
[c] Bogoliubov Laboratory of Theoretical Physics,
Joint Institute for Nuclear Research,
141980 Dubna, Russia;
e-mail: mladim@thsun1.jinr.ru

We study the classical dynamics of mechanical model obtained from the light-cone version of $SU(2)$ Yang-Mills field theory under the supposition of the gauge potential dependence only on "time" along the light-cone direction. The computer algebra system Maple was used strongly to compute and separate the complete set of constraints. In contrast to the instant form of the Yang-Mills mechanics the constraints here represent a mixed form of first and second-class constraints and reduce the number of the physical degrees of freedom up to four canonical one.

1 Introduction

Notion of the evolution of observables is the key element in analyzing of the physical properties of any relativistic field theory. After Dirac's famous work entitled *"Forms of Relativistic Dynamics"* [1] it has been recognized that the different choices of the time evolution parameter can drastically change the content and interpretation of the theory. The simplest and well-known example illustrated this observation is the light-cone dynamics of free scalar field. In contrast to the corresponding instant time model, in this case, owing to the choice of time evolution parameter along the light-cone characteristics, theory becomes degenerate, the corresponding Hessian is zero [2]. Dealing with gauge theories on the light-cone we encounter much more complicated description than for the ordinary instant form dynamics (see e.g. recent reviews [3]-[7]).

In the present talk we would like to state some results concerning the light-cone description of simple mechanical model originated from the $SU(2)$ Yang-Mills theory under assumption of spatial homogeneity of the fields on the light-cone. This means that we

shall consider the light-cone action for $SU(2)$ Yang-Mills model with the gauge potential only light-cone time depending. The dynamical system, obtained under such a supposition contain finite number of degrees of freedom and possesses gauge invariance. Our aim is to study its Generalized Hamiltonian dynamics [2, 8, 9] and to compare it with the corresponding instant form of the Yang-Mills mechanics, intensively studed during the last decades (see e.g [11]-[17] and references therein).

Using the Generalized Hamiltonian formalism for degenerate systems [2, 8, 9] and exploiting the Maple package [10] implementing algorithm Dirac-Gröbner for computation and separation of constraints we found the complete set of constraints and performed their separation into the sets of first and second-class constraints.

Our calculations demonstrate that the light-cone version of Yang-Mills mechanics differs from its instant form counterpart in the character of the local gauge invariance and therefore the corresponding unconstrained Hamiltonian systems describe different canonically non-equivalent models.

2 Description of the model

Let us start with a general formulation of the Yang-Mills theory on four-dimensional Minkowski space M_4, endowed with some arbitrary metric g_P, tensor field of type $(0,2)$,

$$g_P = g_{\mu\nu}\,\omega^\mu \otimes \omega^\nu . \tag{1}$$

Here we use a basis $\omega^\mu, \mu = 0,\ldots,3$ of 1-forms in the cotangent space $T^*_P(M_4)$. The metric defines a inner product between two vectors in the tangent space $T_P(M_4)$ and if one fixes a basis e_μ in $T_P(M_4)$, dual to the basis of 1-forms ω^μ, the components of the metric tensor are given as

$$g_{\mu\nu} = g_P(e_\mu, e_\nu) .$$

Using this geometric settings, the action of the Yang-Mills field theory can be represented in coordinate free form

$$I := \frac{1}{g^2}\int_{M_4} \operatorname{tr} F \wedge *F . \tag{2}$$

Here the $SU(2)$ algebra valued curvature 2-form

$$F := dA + A \wedge A$$

is constructed from the connection 1-form $A = A_\mu\,\omega^\mu$. The connection and curvature as Lie algebra valued quantities are expressed in terms of the Pauli matrices $\sigma^a, a = 1,2,3$ [1]

$$A = A^a\frac{\sigma^a}{2i}, \qquad F = F^a\frac{\sigma^a}{2i} .$$

[1]The Pauli matrices are defined to satisfy

$$[\sigma^a,\sigma^b] = 2i\,\varepsilon^{abc}\sigma^c, \qquad \operatorname{tr}\sigma^a\sigma^b = 2\delta^{ab} .$$

The metric g enters the action through the dual field strength tensor defined in accordance to the Hodge star operation

$$*F_{\mu\nu} = \frac{1}{2}\,\epsilon_{\mu\nu\alpha\beta}\,F^{\alpha\beta}\,.$$

If one fixes Lorentzian coordinates in Minkowski space M_4 $x^\mu = (x^0, x^1, x^2, x^3)$ and choose a coordinate basis for the tangent vectors $e_\mu = \frac{\partial}{\partial x^\mu}$, we have the conventional Minkowskian metric $g = \|1, -1, -1, -1\|$ and the corresponding action (2) will provide Yang-Mills equations in the instant form with a time variable $t = x^0$.

To formulate the light-cone version of the theory let us introduce basis vectors in the tangent space $T_P^*(M_4)$

$$e_\pm := \frac{1}{\sqrt{2}}\,(e_0 \pm e_3)\,, \quad e_\perp := (e_k, k = 1, 2)\,.$$

The first two vectors are tangent to the light-cone and the corresponding coordinates are referred usually as the light-cone coordinates $x^\mu = \left(x^+, x^-, x^\perp\right)$ with

$$x^\pm := \frac{1}{\sqrt{2}}\,\left(x^0 \pm x^3\right)\,, \qquad x^\perp := \left(x^k, k = 1, 2\right)\,.$$

The light-cone basis vectors $(e_\pm, e_k)$ determine, according to (1), the so-called light-cone metric, whose non-zero elements are

$$g_{+-} = g_{-+} = -g_{11} = -g_{22} = 1$$

and thus the 1-form connection in the light-cone formulation is given as

$$A = A_+\,dx^+ + A_-\,dx^+ + A_k\,dx^k\,. \tag{3}$$

Now we are ready to define the Lagrangian corresponding to the light-cone Yang-Mills mechanics. By definition the Lagrangian of Yang-Mills mechanics follows from the Lagrangian of Yang-Mills theory if one suppose that 1-form connection A depends only on light-cone "time variable" x^+

$$A = A(x^+)\,.$$

Using the definition (2) and (3) we find the Lagrangian of the Yang-Mills light-cone mechanics

$$L := \frac{1}{2g^2}\,\left(F^a_{+-}\,F^a_{+-} + 2\,F^a_{+k}\,F^a_{-k} - F^a_{12}\,F^a_{12}\right)\,, \tag{4}$$

where the field-strength tensor light-cone components are

$$\begin{cases} F^a_{+-} = \frac{\partial A^a_-}{\partial x^+} + \epsilon^{abc}\,A^b_+\,A^c_-\,, \\ F^a_{+k} = \frac{\partial A^a_k}{\partial x^+} + \epsilon^{abc}\,A^b_+\,A^c_k\,, \\ F^a_{-k} = \epsilon^{abc}\,A^b_-\,A^c_k\,, \\ F^a_{ij} = \epsilon^{abc}\,A^b_i\,A^c_j\,, \quad i, j, k = 1, 2\,. \end{cases} \tag{5}$$

3 Hamiltonian formulation of $SU(2)$ Yang-Mills mechanics on the light-cone

In this section we present the main results of this paper. The underlying computations were done with the Maple package implementing algorithm Dirac-Gröbner [10] for computation and separation of constraints for Lagrangian dynamical systems of polynomial type. Some computational details are described in the next section.

The choice of the light-cone time variable

$$\tau = x^+$$

as the evolution parameter prescribes a certain Legendre transformation of the dynamical variables (A_+, A_-, A_k) [2]

$$\begin{cases} \pi_a^- = \frac{\partial L}{\partial \dot{A}_-^a} = \frac{1}{g^2}\left(\dot{A}_-^a + \epsilon^{abc} A_+^b A_-^c\right), \\ \pi_a^k = \frac{\partial L}{\partial \dot{A}_k^a} = \frac{1}{g^2}\epsilon^{abc} A_-^b A_k^c. \end{cases} \tag{6}$$

Given this set of equations the Dirac-Gröbner algorithm designed in [10] and leads to the the primary constraints

$$\varphi_a^{(1)} := \pi_a^+ = 0, \tag{7}$$

$$\chi_k^a := g^2 \pi_k^a + \epsilon^{abc} A_-^b A_k^c = 0. \tag{8}$$

Then, the canonical Hamiltonian is given by

$$H_C = \frac{g^2}{2}\pi_a^- \pi_a^- - \epsilon^{abc} A_+^b \left(A_-^c \pi_a^- + A_k^c \pi_a^k\right) + V(A_k) \tag{9}$$

with a potential term in (9)

$$V(A_k) = \frac{1}{2g^2}\left[\left(A_1^b A_1^b\right)\left(A_2^c A_2^c\right) - \left(A_1^b A_2^b\right)\left(A_1^c A_2^c\right)\right].$$

The non vanishing Poisson brackets are

$$\{A_\pm^a, \pi_b^\pm\} = \delta_b^a,$$
$$\{A_k^a, \pi_b^l\} = \delta_k^l \delta_b^a.$$

With respect to these fundamental Poisson brackets the primary constraints $\left(\varphi_a^{(1)}, \chi_k^a\right)$ obey the algebra

$$\{\varphi_a^{(1)}, \varphi_b^{(1)}\} = 0,$$
$$\{\varphi_a^{(1)}, \chi_k^b\} = 0,$$
$$\{\chi_i^a, \chi_j^b\} = -2\, g^2 \epsilon^{abc} A_-^c\, g_{ij}.$$

[2] To simplify the formulas we shall use overdot to denote derivative of functions with respect to light-cone time variable x^+. Further we shall treat in equal footing the up and down isotopic indexes, denoted with a, b, c, d.

According to the Dirac's prescription, the dynamics for degenerate theories is governed by the total Hamiltonian which differs from the canonical one by linear combination of the primary constraints. In case of the light-cone Yang-Mills mechanics the total Hamiltonian has the form

$$H_T = H_C - 2\,\mathrm{tr}\left(U(\tau)\,\varphi^{(1)}\right) - 2\,\mathrm{tr}\left(V_k(\tau)\,\chi_k\right)\,, \tag{10}$$

where $U(\tau)$ and $V_k(\tau)$ are arbitrary $SU(2)$ valued functions of the light-cone time $\tau = x^+$. Using this Hamiltonian it is necessary to check the dynamical self-consistence of the primary constraints. ¿From the requirement of conservation in time of the primary constraints $\varphi_a^{(1)}$ the following equations follows

$$0 = \dot{\varphi}_a^{(1)} = \{\pi_a^+\,, H_T\} = \epsilon^{abc}\left(A_-^b\pi_c^- + A_k^b\pi_c^k\right)\,.$$

It follows that there are three secondary constraints $\varphi_a^{(2)}$

$$\varphi_a^{(2)} := \epsilon_{abc}\left(A_-^b\pi_c^- + A_k^b\pi_c^k\right) = 0\,, \tag{11}$$

which obey the $SO(3,\mathbb{R})$ algebra

$$\{\varphi_a^{(2)}\,, \varphi_b^{(2)}\} = \epsilon_{abc}\,\varphi_c^{(2)}\,.$$

The same procedure for the primary constraints χ_k^a gives

$$0 = \dot{\chi}_k^a = \{\chi_k^a\,, H_C\} - 2\,g^2\,\epsilon^{abc}\,V_k^b\,A_-^c\,. \tag{12}$$

Because the matrix $\|\epsilon^{abc}\,A_-^c\|$ is degenerate, its rank is

$$\mathrm{rank}\|\,\epsilon^{abc}\,A_-^c\,\| = 2\,,$$

one can determine among the Lagrange multipliers V_b^k only four ones. The unit null vector of the matrix $n^a\|\,\varepsilon^{abc}\,A_-^c\,\| = 0$ is

$$n^a = \frac{A_-^a}{\sqrt{(A_-^1)^2 + (A_-^2)^2 + (A_-^3)^2}}\,.$$

Using this null vector one can can decompose the six primary constraints χ_k^a

$$\begin{aligned}\chi_{k\perp}^a &:= \chi_k^a - \left(n^b\chi_k^b\right)\,n^a\,,\\ \psi_k &:= n^a\chi_k^a\,.\end{aligned}$$

Constraints $\chi_{k\perp}^a$ are functionally dependent due to the conditions

$$n^a\,\chi_{k\perp}^a = 0 \tag{13}$$

and choosing among them any four independent constraints we are able to determine four Lagrange multipliers $V_{b\perp}^k$. The two constraints ψ_k satisfy the Abelian algebra

$$\{\psi_i\,, \psi_j\} = 0\,.$$

One can verify that the Poisson brackets of ψ_k and $\varphi_a^{(2)}$ with the total Hamiltonian on the constraint surface (CS) are zero

$$\{\psi_k, H_T\}_{|CS} = 0, \tag{14}$$
$$\{\varphi_a^{(2)}, H_T\}_{|CS} = 0 \tag{15}$$

and thus there are no ternary constraints. One can now summarize: we arrive at the set of constraints $\left(\varphi_a^{(1)}, \psi_k, \varphi_a^{(2)}, \chi_{k\perp}^b\right)$. The Poisson bracket algebra of the three first one is

$$\{\varphi_a^{(1)}, \varphi_a^{(1)}\} = 0, \tag{16}$$
$$\{\psi_i, \psi_j\} = 0, \tag{17}$$
$$\{\varphi_a^{(2)}, \varphi_b^{(2)}\} = \epsilon_{abc}\varphi_c^{(2)}, \tag{18}$$
$$\{\varphi_a^{(1)}, \psi_k\} = \{\varphi_a^{(1)}, \varphi_b^{(2)}\} = \{\psi_k, \varphi_a^{(2)}\} = 0. \tag{19}$$

The constraints $\chi_{k\perp}^b$ satisfy the algebra

$$\{\chi_{i\perp}^a, \chi_{j\perp}^b\} = -2g^2\epsilon^{abc}A_-^c g_{ij} \tag{20}$$

and the Poisson brackets between these two sets of constraints are

$$\{\varphi_a^{(2)}, \chi_{k\perp}^b\} = \epsilon^{abc}\chi_{k\perp}^c, \tag{21}$$
$$\{\varphi_a^{(1)}, \chi_{k\perp}^b\} = \{\psi_i, \chi_{j\perp}^b\} = 0. \tag{22}$$

From this algebra of constraints we conclude that we have eight first-class constraints $\left(\varphi_a^{(1)}, \psi_k, \varphi_a^{(2)}\right)$ and four second-class constraints $\chi_{k\perp}^a$. According to counting of the degrees of freedom eliminated by all these constraints, after reduction to the unconstrained phase space, instead of 24 degrees of freedom possessing the Yang-Mills mechanics on the light-cone we arrive at $24 - 4 - 2(3+3+2) = 4$ unconstrained degrees of freedom.

Thus one can conclude that in contrast to the instant form of the Yang-Mills mechanics, where the number of the unconstrained canonical pairs is 12, in the light-cone version we have only 4 physical canonical variables. It is important to note that such a decreasing of the numbers of the physical coordinates has two reasons: as well as the presence of the second-class constraints as the additional first-class constraint. As it is well-known the presence of the first-class constraints in the theory means the existence of a certain gauge symmetry. Our analysis shows that in the light-cone Yang-Mills mechanics the original $SU(2)$ gauge symmetry of the field theory, after supposition of the gauge fields homogeneity, transforms into $SU(2) \times U(1) \times U(1)$ symmetry.

4 Computational aspects

The system of Euler-Lagrange equations for the Lagrangian (4) with the generalized coordinates (5) as well as the system of Hamiltonian equations which are obtained by the Legendre transformation (6) is under-determined. Just under-determinacy is inherent in constrained dynamical systems [1, 2, 8, 9], and to study such systems one has to compute

their "hidden" constraints. For the quantization purposes one has also to separate the constraints into disjoint sets of first-class and second-class constraints.

In the paper [10] a general algorithm for computing and separating constraints for polynomial Lagrangians was devised. The algorithm combines the constructive ideas of Dirac [8] with the Gröbner bases techniques and called Dirac-Gröbner algorithm. Its implementation was done in Maple and in this section we characterize briefly the main computational steps one needs to obtain the results of the previous section as they were done by the Maple code. In so doing the below described computational steps is nothing else than concretization of the Dirac-Gröbner algorithm to our model described in Sect.2.

Denote by q_i and $\dot{q}_i$ $(1 \leq i \leq 12)$, respectively, the generalized Lagrangian coordinates in (4) listed as

$$A^1_+, A^2_+, A^3_+, A^1_1, A^2_1, A^3_1, A^1_2, A^2_2, A^3_2, A^1_-, A^2_-, A^3_-$$

and their velocities (time derivatives). Then momenta are

$$p_i = \frac{\partial L}{\partial \dot{q}_i} \qquad (1 \leq i \leq 12)\,. \tag{23}$$

To compute the primary constraints it suffices to eliminate velocities $\dot{q}_i$ from the system (23) polynomial in $\dot{q}_i, q_i, p_i$. The elimination are performed by computing a Gröbner basis [18, 19] for the generating polynomial set

$$\{\; p_i - \frac{\partial L}{\partial \dot{q}_i} \;\mid\; 1 \leq i \leq 12 \;\}$$

for an ordering (in Maple`lexdeg`) eliminating velocities $\dot{q}_i$. In the obtained set all algebraically dependent constraints [18] are ruled out. Thus (7)-(8) is the algebraically independent set.

The canonical Hamiltonian (9) is determined as reduction of

$$p_i\dot{q}_i - L$$

modulo the Gröbner basis computed. Then computation of the Poisson brackets between the Hamiltonian variables (generalized coordinates and momenta) as well as computation of the total Hamiltonian (10) is straightforward.

Next step is construction of the secondary constraints (11)-(12). It is done by reduction of the Poisson brackets of the primary constraints with the total Hamiltonian modulo the set of primary constraints. Again the Gröbner basis technique provides the right algorithmic tool for doing such computations. Thus, the complete set of algebraically independent constraints consists of twelve elements

$$\mathcal{F} = \{\; \varphi^{(1)}_a, \psi_k, \varphi^{(2)}_a, \chi^b_{k\perp} \;\}\,, \qquad (a,b = 1,2,3;\; k = 1,2)\,, \tag{24}$$

where from the six constraints $\chi^b_{k\perp}$ only four algebraically independent are included in (24) in accordance with the two relations (13).

Next, to separate the complete set of constraints into first and second classes one computes the 12×12 Poisson bracket matrix on the constraint surface

$$M := \| \{f_i, f_j\}_{|CS} \|\,,$$

where $f_i, f_j \in \mathcal{F}$. Since $\mathrm{rank}\|M\| = 4$ the complete constraint set $\mathcal{F}$ can be separated in four second-class constraints and eight first-class ones. To select the first-class constraints it suffices to compute a basis

$$\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_8\}$$

of the null space for the matrix $\|M\|$ and then construct the first-class constraints as

$$(\mathbf{a}_i)_k f_k \qquad (1 \le i \le 8)\,.$$

To extract the second-class constraints from $\mathcal{F}$ one constructs 8×12 matrix $\| (\mathbf{a}_i)_j \|$ from the components of the vectors in $\mathcal{A}$ and finds a basis

$$\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_4\}$$

of the null space of the constructed matrix. Then every vector $\mathbf{b} \in \mathcal{B}$ yields a second-class constraint:

$$(\mathbf{b}_i)_k f_k \qquad (1 \le i \le 4)\,.$$

As a result, eight first-class constraints are $\left(\varphi_a^{(1)}, \psi_k, \varphi_a^{(2)}\right)$, whereas four algebraically independent constraints from $\chi_{k\perp}^a$ are of the second-class.

Relations (16)-(19) revealing the structure of the gauge group generated by the first class constraints can also be computed fully algorithmically. To do this we extended of Maple package [10] with a general procedure that computes the Poisson bracket of any two first-class constraints f_i and f_j as linear combination of elements in the set of first-class constraints:

$$\{f_i\,, f_j\} = c^k_{ij}\, f_k\,. \tag{25}$$

With that end in view and in order to cope the most general case we implemented the extended Gröbner basis algorithm [19]. Given a set of polynomials $Q = \{q_1, \ldots, q_m\}$ generating the polynomial ideal $< Q >$, this algorithm outputs the explicit representation

$$g_i = h_{ik}\, q_k \tag{26}$$

of elements in a Gröbner basis $G = \{g_1 \ldots, g_n\}$ of this ideal in terms of the polynomials in Q. Having computed a Gröbner basis G for the ideal generated by the first-class constraints and the corresponding polynomial coefficients h_{ik} for the elements in G as given in (26), the local group coefficients c^k_{ij} (which may depend on the generalized coordinates and momenta) in (25) are easily computed by reduction [18, 19] of the Poisson brackets modulo Gröbner basis expressed in terms of the first-class constraints.

However, the use of this universal approach may be very expensive from the computational point of view. For this reason our Maple package tries first to apply the multivariate polynomial division algorithm [18] modulo the set of first-class constraints. Due to the special structure of the primary first-class constraints that usually include those linear in momenta as in (7), this algorithm often produces the right representation (26); but unlike the extended Gröbner basis algorithm does it very fast. Correctness of the output is easily verified by computing of the reminder. If the last vanishes, then the output of the division algorithm is correct. Otherwise the extended Gröbner basis algorithm is applied.

In our case the division algorithm just produces the correct formulas (16)-(19) for the Poisson brackets of the first-class constraints $\left(\varphi_a^{(1)}, \psi_k, \varphi_a^{(2)}\right)$. Similarly, one obtains the formulas (20)-(22).

Acknowledgements

The work was supported in part by the RFBR grant 01-01-00708. Contribution of V.P.G. was also partially supported by the RFBR grant 00-15-96691 and by grant INTAS 99-1222. A.M.K. acknowledges INTAS for providing financial support, grant 00-00561.

References

[1] P.A.M. Dirac, Forms of relativistic dynamics, Rev. Mod. Phys. **21** (1949) 392-399.

[2] K. Sundermeyer, *Constrained Dynamics*, Lecture Notes in Physics **169**, (Springer Verlag, Berlin-Heidelberg-New York, 1982).

[3] T. Heinzl, *Light cone quantization: Foundations and applications*, Lecture Notes Physics **572** 55-142, (Springer Verlag, Berlin-Heidelberg-New York, 2001). [arXiv: hep-th/0008096].

[4] G.A. Miller, *Light front quantization: A technique for relativistic and realistic nuclear physics*, Prog. Part. Nucl. Phys. **45** (2000) 83-155 ; [arXiv: nucl-th/0002059].

[5] S.A. Paston, V.A. Franke and E.V. Prokhvatilov, Constructing the light front QCD Hamiltonian, Theor. Math. Phys. **120** (1999) 1164-1181, (Teor. Mat. Fiz. **120** (1999) 417-437); [arXiv: hep-th/0002062].

[6] H.C. Pauli, *Discretized light cone quantization and the effective interaction in hadrons*, AIP Conf. Proc. **494**, (1999) 80-139; [arXiv: hep-ph/9910203].

[7] P.P. Srivastava, *Perspectives of light front quantum field theory: Some new results*, arXiv: hep-ph/9908492.

[8] P.A.M. Dirac, *Lectures on Quantum Mechanics*, Belfer Graduate School of Science, (Yeshiva University Press, New York, 1964).

[9] M. Henneaux, C. Teitelboim, *Quantization of Gauge Systems*, (Princeton University Press, Princeton, NJ, 1992).

[10] V.P. Gerdt and S.A. Gogilidze, *Constrained Hamiltonian Systems and Gröbner Bases*. In: Computer Algebra in Scientific Computing / CASC'99, V.G. Ganzha, E.W. Mayr and E.V. Vorozhtsov (Eds.), pp. 139-146, (Springer-Verlag, Berlin, 1999).

[11] S.G. Matinyan, G.K. Savvidy, and N.G. Ter-Arutyunyan-Savvidy, *Classical Yang-Mills mechanics. Nonlinear colour oscillations*, Sov. Phys. JETP **53** (1981) 421-429.

[12] H.M. Asatryan and G.K. Savvidy, *Configuration manifold of Yang-Mills classical mechanics*, Phys. Lett. **A 99** (1983) 290-292.

[13] M.A. Soloviev, *On the geometry of classical mechanics with nonabelian gauge symmetry*, Teor. Mat. Fyz. **73** (1987) 3-15.

[14] M.J. Gotay, *Reduction of homogeneous Yang-Mills fields*, J. Geom. Phys. **6** (1989) 349-365.

[15] B. Dahmen and B. Raabe, *Unconstrained $SU(2)$ and $SU(3)$ Yang-Mills classical mechanics*, Nucl. Phys. **B 384** (1992) 352-380.

[16] S.A. Gogilidze, A.M. Khvedelidze, D.M. Mladenov, and H.-P. Pavel, *Hamiltonian reduction of $SU(2)$ Dirac-Yang-Mills mechanics*, Phys. Rev. **D 57** (1998) 7488-7500.

[17] A.M. Khvedelidze and D.M. Mladenov, *Euler-Calogero-Moser system from $SU(2)$ Yang-Mills theory*, Phys. Rev. D **62**, (2000) 125016(1-9) .

[18] D. Cox, J. Little, D. O'Shea, *Ideals, Varieties and Algorithms*, Second Edition, (Springer-Verlag, New York, 1996).

[19] T. Becker, V. Weispfenning, H. Kredel, *Gröbner Bases. A Computational Approach to Commutative Algebra.* Graduate Texts in Mathematics **141**, (Springer-Verlag, New York, 1993).

Parallelism in Computing Janet Bases

Vladimir P. Gerdt, Denis A. Yanovich

Laboratory of Information Technologies,
Joint Institute for Nuclear Research,
141980 Dubna, Russia;
e-mail: gerdt@jinr.ru yan@jinr.ru

In this paper we show that the involutive algorithm for computation of Janet bases for polynomial ideals admits an efficient parallelism based on the shared memory architecture. We present explicit computational results obtained for parallel modular computations on an two-processor computer. As benchmarks we use those widely exploited for testing Gröbner bases software.

1 Introduction

The goal of this paper is to study efficiency of parallelism of the algorithm developed in [1] for computing Janet bases of polynomial ideals. Janet bases, whose structure goes back to the original Janet concept of involutive systems of partial differential equations [2], are typical representatives of involutive bases [3, 4, 5, 6, 7]. They have found a number of applications to mathematics and physics [9].

As well as other involutive bases, Janet bases are (generally redundant) Gröbner bases [3] and, hence, have an exponential and even subexponential complexity bound [10]. Though in practice, the modern computational algorithms for these bases terminate on many problems much quicker than one might expect from the worst-case estimation, the running time and storage space are still very high for polynomial systems of real practical interest. By this reason parallelization of these algorithms is a problem of great practical importance.

In the last decade quite a number of attempts have been made to parallelize (see, for example [11, 12, 13]) the classical Buchberger algorithm for computing Gröbner bases [15]. It turned out that a reasonable scalability cannot be achieved. One of the main reasons is a very high sensitivity in the course of the Buchberger algorithm on the selection strategy for critical pairs ($S-$polynomials). Such a strong dependence inhibits effectiveness of the parallelization.

In the present paper, by example of Janet bases, we argue that unlike the Buchberger algorithm, the involutive algorithm [1] admits an effective parallelism.

2 Polynomial Janet bases

In this paper we use notations and definitions given in [1, 8]: $\mathbb{N}$ is the set of non-negative integers; $\mathbb{M} = \{x_1^{d_1} \cdots x_n^{d_n} \mid d_i \in \mathbb{N}\}$ is the set of monomials in the polynomial ring $\mathbb{R} = \mathbb{K}[x_1, \ldots, x_n]$ over zero characteristic field $\mathbb{K}$; $\deg_i(u)$ is the degree of x_i in $u \in \mathbb{M}$; $\deg(u) = \sum_{i=1}^{m} \deg_i(u)$ is the total degree of u; $\succ$ is an admissible [10, 15] monomial ordering compatible with

$$x_1 \succ x_2 \succ \cdots \succ x_n .$$

If monomial u divides monomial v we write $u \mid v$. A divisor u of a monomial v is called a proper divisor if $deg(u) < deg(v)$. The leading monomial and the leading term of the polynomial $f \in \mathbb{R}$ with respect to $\succ$ are denoted by $\mathrm{lm}(f)$ and $lt(f)$, respectively. If $F \subset \mathbb{R}$ is a polynomial set, then $\mathrm{lm}(F)$ denotes the leading monomial set for F, and $Id(F)$ denotes the ideal in R generated by F.

Given a finite polynomial set $F \subset \mathbb{R}$ and a polynomial $f \in F$, the Janet separation of variables into multiplicative and nonmultiplicative with respect to f is done as follows. For each $1 \leq i \leq n$ divide F into groups labeled by non-negative integers $d_1, \ldots, d_i$:

$$[d_1, \ldots, d_i] = \{\ f\ \in F \mid d_j = \deg_j(\mathrm{lm}(f)),\ 1 \leq j \leq i\ \}.$$

A variable x_i is called *(Janet) multiplicative* for $f \in F$ if $i = 1$ and

$$\deg_1(\mathrm{lm}(f)) = \max\{\deg_1(\mathrm{lm}(g)) \mid g \in F\},$$

or if $i > 1$, $f \in [d_1, \ldots, d_{i-1}]$ and

$$\deg_i(\mathrm{lm}(f)) = \max\{\deg_i(\mathrm{lm}(g)) \mid g \in [d_1, \ldots, d_{i-1}]\}.$$

If a variable is not multiplicative for $f \in F$, it is called *(Janet) nonmultiplicative* for f. In the latter case we shall write $x_i \in NM_J(f, F)$. $u \in \mathrm{lm}(F)$ is a *Janet divisor* of $w \in \mathbb{M}$, if $u \mid w$ and monomial w/u contains only multiplicative variables for u. In this case we shall write $u \mid_J w$.

A finite polynomial set F is called *Janet autoreduced* if each term in every $f \in F$ has no Janet divisors among $\mathrm{lm}(F) \setminus \mathrm{lm}(f)$. A polynomial $h \in \mathbb{R}$ is said to be in the *Janet normal form modulo* F if every term in h has no Janet divisors in $\mathrm{lm}(F)$. In that follows $NF_J(f, F)$ denotes the Janet normal form of polynomial f modulo F. If the leading monomial $\mathrm{lm}(f)$ of f has no Janet divisors among elements in $\mathrm{lm}(F)$, then we say that f is in the *Janet head normal form* modulo F and write $f = HNF_J(f, F)$.

A Janet autoreduced set F is called a *Janet basis* of $Id(F)$ if any nonmultiplicative *prolongation* (multiplication by a nonmultiplicative variable) of any polynomial in F has vanishing Janet normal form modulo F:

$$(\forall f \in F)\ (\forall x \in NM_J(f, F))\ \ [\ NF_J(f \cdot x, F) = 0\ ]. \tag{1}$$

A Janet basis G of ideal $Id(G)$ is called *minimal* if for any other Janet basis F of the ideal the inclusion $lm(G) \subseteq \mathrm{lm}(F)$ holds. If both G and F are monic this inclusion implies $G \subseteq F$. A Janet basis is a Gröbner one, though generally not reduced. However, similarly to a reduced Gröbner basis a monic minimal Janet basis is uniquely defined by an ideal

and a monomial order. In that follows we deal with minimal Janet bases only and omit the word "minimal".

The below described algorithm **JanetBasis** for computation of polynomial Janet bases derived from that described in [1], where one can find more algorithmic details. For purposes of parallelism we added one more input variable: a maximal number of simultaneous reductions to be done in parallel on an available environment.

As well as in [1] to apply the involutive criteria and avoid repeated prolongations we shall endow with every polynomial $f \in F$ the triple structure

$$p = \{f, u, vars\}$$

such that

$$\begin{aligned} \mathrm{pol}(p) &= f \text{ is polynomial } f \text{ itself,} \\ \mathrm{anc}(p) &= u \text{ is the leading monomial of a polynomial ancestor of } f \text{ in } F, \\ \mathrm{nmp}(p) &= vars \text{ is a (possible empty) subset of variables.} \end{aligned}$$

Here the *ancestor* of f is a polynomial $g \in F$ with $u = \mathrm{lm}(g)$ and such that $u \mid \mathrm{lm}(p)$. Moreover, if $\deg(u) < \deg(\mathrm{lm}(p))$, then every variable occurring in the monomial $\mathrm{lm}(p)/u$ is nonmultiplicative for g. Besides, for the ancestor g the equality $\mathrm{anc}(g) = \mathrm{lm}(g)$ must hold. These conditions mean that polynomial p was obtained in the course of the below algorithm **JanetBasis** from g by a sequence of nonmultiplicative prolongations. This tracking of the history in the algorithm allows one to use the involutive analogues of Buchberger's criteria to avoid unnecessary reductions.

The set *vars* contains those nonmultiplicative variables which have been already used in the algorithm for construction of nonmultiplicative prolongations. This set serves to prevent repeated prolongations.

In order to provide minimality of the output Janet basis we separate [1, 4] the whole polynomial data into two subsets which are contained in sets T and Q. Set T contains a part of the intermediate basis. Another part of the intermediate basis is contained in set Q together with all the nonmultiplicative prolongations of polynomials in T which must be examined in accordance to the above definition (1) of Janet bases. In so doing, after every insertion of a new element p in T all elements $r \in T$ such that

$$\mathrm{lm}(\mathrm{pol}(r)) \succ \mathrm{lm}(\mathrm{pol}(p))$$

are moved from T to Q. Such a displacement guaranties that the output basis is minimal [4].

It should also be noted that for any triple $p \in T$ the set *vars* must always be a subset of the set of nomultiplicative variables of $\mathrm{pol}(p)$

$$vars \subseteq NM_J(\mathrm{pol}(p), T). \tag{2}$$

In the description of algorithm **JanetBases** we use the following contractions

$$\begin{aligned} NM_J(\mathrm{pol}(p), T) &\equiv NM_J(\mathrm{pol}(p), \{\mathrm{pol}(f) \mid f \in T\}), \\ NF_J(\mathrm{pol}(p), T) &\equiv NF_J(\mathrm{pol}(p), \{\mathrm{pol}(f) \mid f \in T\}), \\ HNF_J(\mathrm{pol}(p), T) &\equiv HNF_J(\mathrm{pol}(p), \{\mathrm{pol}(f) \mid f \in T\}). \end{aligned}$$

Algorithm: JanetBasis(F, $\prec$, J_{MP})

Input: $F \in \mathbb{R} \setminus \{0\}$, a finite polynomial set; $\prec$, an admissible ordering; J_{MP}, the maximal number of simultaneous reductions to be done in parallel

Output: G, a Janet basis of $Id(F)$

1: **choose** $f \in F$ with the lowest $\mathrm{lm}(f)$ w.r.t. $\succ$
2: $T := \{f, \mathrm{lm}(f), \emptyset\}$
3: $Q := \{\{q, \mathrm{lm}(q), \emptyset\} \mid q \in F \setminus \{f\}\}$
4: $Q :=$**JanetHeadReduce**(Q, T, J_{MP})
5: **while** $Q \neq \emptyset$ **do**
6: **choose** $p \in Q$ such that $\mathrm{lm}(\mathrm{pol}(p))$ has no proper divisors among $\{\mathrm{lm}(\mathrm{pol}(q)) \mid q \in Q \setminus \{p\}\}$
7: **if** $\mathrm{lm}(\mathrm{pol}(p)) = 1$ **then**
8: **return** $\{1\}$
9: **else**
10: $Q := Q \setminus \{p\}$
11: **if** $\mathrm{lm}(\mathrm{pol}(p)) = \mathrm{anc}(p)$ **then**
12: **for all** $\{r \in T \mid lm(\mathrm{pol}(r)) \succ \mathrm{lm}(\mathrm{pol}(p))\}$ **do**
13: $Q := Q \cup \{r\}$; $T := T \setminus \{r\}$
14: **od**
15: **fi**
16: $\mathrm{pol}(p) := \mathbf{NF_J}(\mathrm{pol}(p), T)$
17: **fi**
18: $T := T \cup \{p\}$
19: **for all** $q \in T$ **and** $x \in NM_J(\mathrm{pol}(q), T) \setminus \mathrm{nmp}(q)$ **do**
20: $Q := Q \cup \{\{\mathrm{pol}(q) \cdot x, \mathrm{anc}(q), \emptyset\}\}$
21: $\mathrm{nmp}(q) := \mathrm{nmp}(q) \cap NM_J(\mathrm{pol}(q), T) \cup \{x\}$
22: **od**
23: $Q :=$**JanetHeadReduce**(Q, T, J_{MP})
24: **od**
25: **return** $G := \{\mathrm{pol}(f) \mid f \in T\}$

The initialization step is done in lines 1-4. The next subalgorithm **JanetHeadReduce** performs Janet reduction of the leading terms of polynomials in Q modulo polynomials in T and under specification of the number of threads for parallel reductions.

In the main loop 5-22 an element in Q is selected in line 6. The correctness of this selection strategy proved in [14]. In practice the number of elements in Q at intermediate steps of the algorithm is rather large and easily runs up to hundreds and thousands. At the same time there may be different polynomials in Q with identical leading monomials. Therefore, the restriction in line 6 still admits some arbitrariness. In our implementation in [1] for the degree-reverse-lexicographical ordering a triple $p \in Q$ with the minimal $\deg(\mathrm{lm}(\mathrm{pol}(p)))$ was chosen. In the case of several such polynomials in Q that with the

minimal number of terms was picked up. In our parallel implementation described in the next section we use similar selection strategy.

Line 8 break computations in the case when inconsistency is revealed during the head term reduction in Q and returns the unit basis. In line 16 the tail Janet reduction is done and then the Janet reduced polynomial in p is inserted in T which is released from the higher ranking polynomials in loop 12-14 if any. Actually the release takes place only if polynomial in p has been subjected by the head term reduction in line 23. Otherwise, $\mathrm{pol}(p) \succ \mathrm{pol}(r)$ holds for any $r \in T$. The insertion of a new polynomial in T may generate new nonmultiplicative prolongations of elements in T which are added to Q in line 20. To avoid repeated prolongations the set $\mathrm{nmp}(q)$ of Janet nonmultiplicative variables for g has been used to construct its prolongations is enlarged with x in line 21. The intersection placed in this line preserves the condition (2).

Since the Janet head reduction of an element in Q does not exert influence upon reduction of other its elements, the reduction in lines 4 and 20 admits a natural and easy parallelism adjusted to a number of processors available. It is remarkable that just Janet reduction done in lines 4 and 20 spends overwhelming majority of running time for examples large enough as our experiments with the sequential code [1] show.

The following algorithm **JanetHeadReduce** realizes these facilities for parallel reductions. It is part of algorithm **JanetBasis** and uses light weigth processes or threads.

Algorithm: JanetHeadReduce(Q, T, J_{MP})

Input: Q and T, sets of triples; J_{MP}, a number of threads
Output: Q whose polynomials are Janet head reduced modulo T

```
 1: create J_MP threads for computing HNF_J
 2: S := Q
 3: Q := ∅
 4: while S ≠ ∅ do
 5:   while number of free threads ≠ 0 do
 6:     choose p ∈ S
 7:     S := S \ {p}
 8:     send p in a free thread to compute h := HNF_J(p, T)
 9:     if h ≠ 0 then
10:       if lm(pol(p)) ≠ lm(h) then
11:         Q := Q ∪ {h, lm(h), ∅}
12:       else
13:         Q := Q ∪ {p}
14:       fi
15:     fi
16:   od
17: od
18: return Q
```

Every particular thread is used to compute the Janet head normal form h that is done in line 8 for the polynomial $\mathrm{pol}(p)$ in a triple p chosen in line 6. If h is nonzero then line

10 checks if $\mathrm{pol}(p)$ was subjected to the head term reduction. If the reduction took place, then $\mathrm{lm}(h)$ is not multiple of any monomial in the set $\{\mathrm{lm}(\mathrm{pol}(g)) \mid g \in T\}$ [3]. Therefore, one has to insert the triple with h in the output set Q as shown in line 11 since h cannot have ancestors among polynomials in T. One must also examine all nonmultiplicative prolongations of h as the empty set in the triple signals. If $\mathrm{lm}(\mathrm{pol}(p))$ is Janet irreducible modulo $\{\mathrm{lm}(\mathrm{pol}(g)) \mid g \in T$, then p itself is added to Q in line 13.

Subalgorithm $\mathbf{HNF_J}(f,T)$ of algorithm **JanetHeadReduce** performs the Janet head reduction of the polynomial in f modulo polynomial set in T:

Algorithm: $\mathbf{HNF_J}(f,T)$

Input: $f = \{\mathrm{pol}(f), \mathrm{anc}(f), \mathrm{nmp}(f)\}$, a triple; T, a set of triples
Output: $h = HNF_J(\mathrm{pol}(f), T)$, the Janet head normal form of the polynomial in f modulo polynomial set in T

```
1:  G := {pol(g) | g ∈ T}
2:  if lm(pol(f)) is Janet irreducible modulo G then
3:    return f
4:  else
5:    h := pol(f)
6:    choose g ∈ T such that lm(pol(g)) |_J lm(h)
7:    if lm(h) ≠ anc(f) then
8:      if CriterionI(f,g) or CriterionII(f,g) then
9:        return 0
10:     fi
11:   else
12:     while h ≠ 0 and lm(h) is Janet reducible modulo G do
13:       choose q ∈ G such that lm(q) |_J lm(h)
14:       h := h − q · lt(h)/lt(q)
15:     od
16:   fi
17: fi
18: return h
```

For the head reducible input polynomial $\mathrm{pol}(f)$ the two criteria are verified in line 8:

- **CriterionI**(f,g) is true iff $\mathrm{anc}(f) \cdot \mathrm{anc}(g) \mid \mathrm{lm}(\mathrm{pol}(f))$.
- **CriterionII**(f,g) is true iff $\deg(\mathrm{lcm}(\mathrm{anc}(f) \cdot \mathrm{anc}(g))) < \deg(\mathrm{lm}(\mathrm{pol}(f)))$.

These criteria are nothing else than the Buchberger criteria [15] adapted to the involutive completion procedure. If any of the two criteria is true, then $HNF(\mathrm{pol}(f), T) = 0$ [1].

The next subalgorithm $\mathbf{NF_J}$ performs the Janet tail reduction of a polynomial with the irreducible leading term. Its output is the (full) Janet normal form $NF_J(f,T)$ of the input polynomial f modulo polynomial set containing in T. This subalgorithm is called in line 16 of the main algorithm **Janet basis** and performs a chain of elementary involutive

reductions until every term in the obtained polynomial becomes Janet irreducible modulo polynomials in T.
It should be noted that both the full Janet normal form and the Janet head normal form are uniquely defined and, hence, uniquely computed by the corresponding subalgorithms. This uniqueness hold because of the uniqueness of a Janet divisor among the leading terms of polynomials in T at every step of intermediate computations [4].

Algorithm: $\mathbf{NF_J}(f,T)$

Input: f, the polynomial in a triple p such that $f := HNF_J(p,T)$;
T, a set of triples
Output: $h = NF_J(f,T)$, the full Janet normal form of h
modulo polynomial set in T

```
1: G := {pol(g) | g ∈ T}
2: h := f
3: while h ≠ 0 and h has a term t Janet reducible modulo G do
4:    choose g ∈ G such that lm(g) |_J t
5:    h := h − g · t/lt(g)
6: od
7: return h
```

3 Benchmarking

As it was mentioned in the previous section we used the light weigth processes for parallelizing. The above described parallel version of algorithm **JanetBasis** have been implemented in C. We exploited the Janet trees [1, 8] as data structures for T and unsorted lists for Q. The running times were measured for the degree-reverse-lexicographical monomial ordering compatible with $x_1 \succ x_2 \succ \cdots \succ x_n$ on a 2 processors Pentium III 700 Mhz with 1Gb RAM computer running under Red Hat Linux release 6.2. Parallelization was done by means of the Linux Threads package along with the Intel C/C++ compiler version 5.0.1 for compiling the C code.

Unfortunately, all C packages implementing arithmetics over long integers and available at our disposal are not suitable for parallel computation since they have global buffers and special optimizations. Thus we were able to run the parallel code only for modular arithmetics. More preciously, in computing Janet bases we manipulated with integer coefficients modulo prime number 31013. As an admissible monomial ordering we chosen the degree-reverse-lexicographical order.

To compare the sequential [1] and parallel version of our C code we used the benchmarks taken from the collection [16, 17] and listed in the first column of the below table. The second column contains timings obtained with our sequential code highly optimized just for sequential computing. The other five columns show, respectively, timings for one-thread computation, two-thread computation, speedup rate of the latter with respect to the former, four-thread process and its speedup rate with respect to the one-thread computation. Every of these timings was determined as difference between the astronomical

running time and the system costs such as swapping, context switching, etc.

To provide correct work of the parallel code one had to turn down some optimizations including the global variables among other things and to add some extra computational costs: working with a circular polynomial buffer, blocking of processing at semaphores and mutexes. That is why the optimized sequential code runs notably faster than one-thread version.

Switching to a large number of threads does not increase the parallelism overhead. Therefore, the running time is to decrease with a rate equal to or greater than the number of threads in action. One can see from the table that this decreasing is especially affected large examples. Small examples needing rather short running time have the parallelism overhead which comes up to doubled value of the proper computing time. Thus, our parallel code is best for large problems: the maximal speedup obtained on our two-processor machine is about 38% that is in accordance with the portion of performing Janet reductions which for the modular computation aggregates about 80% of the total running time. In the case of computation over integers one can expect even greater gain since in this case a portion of Janet reduction often exceeds 90%.

One more advantage of parallelism is its more beneficial selection strategy for polynomials coming into play. If several Janet normal forms are computed at the same time, then shorter of them (those with less number of terms and smaller coefficients) will be earlier computed. Then these normal forms come first into play and give rise to reduction of more lengthy polynomials. As an example of such behavior of the parallel computational process we point out **extcyc7**. The speedup factor obtained for this example is above 50%.

In [13] a nondeterministic (chaotic) behavior in parallel implementation of the Buchberger algorithm for computing Gröbner bases was experimentally observed. This multiprocessor chaotic state led to runs which sometimes terminated in seconds while other runs of the same examples must be killed due to memory overflow after hours. In our implementation of parallel Janet completion we did not observe such chaotic behavior, and variation in timing for different runs does not exceed a fraction of a percent.

Acknowledgements

The work was supported in part by the RFBR grants 00-15-96691, 01-01-00708 and by grant INTAS 99-1222. The authors are grateful to Yuri Blinkov for useful discussions and comments.

example	C_{opt}	1 thr	2 thr's	%speedup	4 thr's	%speedup
assur44	1.20	2.16	2.68	-24.07	4.33	-100.46
chemkin	2.70	4.48	4.17	6.92	5.53	-23.44
cohn3	3.55	6.10	6.10	0.00	7.02	-15.08
cyclic6	0.13	0.42	0.66	-57.14	0.97	-130.95
cyclic7	19.68	29.66	24.93	15.95	26.52	10.59
cyclic8	751.96	831.12	614.21	26.10	659.37	20.66
dl	113.90	181.85	165.14	9.19	192.68	-5.96
discret3	2.76	4.46	4.20	5.83	6.15	-37.89
eco8	0.28	0.57	0.74	-29.82	1.32	-131.58
eco9	3.11	4.35	4.02	7.59	5.90	-35.63
eco10	30.29	36.43	26.61	26.96	34.62	4.97
eco11	278.82	307.40	194.6	36.69	218.90	28.79
eco12	3458.96	4215.27	2923.12	30.65	3213.76	23.76
extcyc5	0.41	1.03	1.29	-25.24	1.84	-78.64
extcyc6	29.91	41.95	32.21	23.22	38.23	8.87
extcyc7	12919.30	23094.80	10651.80	53.88	12975.10	43.82
f744	4.20	8.88	9.72	-9.46	12.90	-45.27
f855	35.99	59.42	54.17	8.84	68.56	-15.38
hcyclic7	21.10	32.55	28.61	12.10	33.59	-3.20
hcyclic8	1206.65	1455.21	1213.87	16.58	1245.89	14.38
hietarinta1	1.11	2.63	3.57	-35.74	4.99	-89.73
i1	1.54	2.97	3.48	-17.17	5.62	-89.23
ilias_k_2	18.83	31.46	26.30	16.40	31.81	-1.11
ilias_k_3	28.14	40.29	34.89	13.40	41.24	-2.36
ilias12	203.86	258.92	205.35	20.69	224.64	13.24
ilias13	47.98	65.63	55.72	15.10	65.17	0.70
katsura7	0.73	1.32	1.25	5.30	1.97	-49.24
katsura8	11.06	14.37	10.02	30.27	13.41	6.68
katsura9	107.56	125.88	79.85	36.57	92.64	26.41
katsura10	1035.79	1191.83	733.66	38.44	759.82	36.25
noon6	0.95	1.69	2.50	-47.93	3.34	-97.63
noon7	17.34	23.56	26.36	-11.88	32.25	-36.88
noon8	560.39	621.92	600.62	3.42	642.09	-3.24
redcyc6	0.10	0.36	0.61	-69.44	0.90	-150.00
redcyc7	37.68	60.05	51.1	14.90	56.33	6.19
redeco10	12.22	16.26	13.24	18.57	17.50	-7.63
redeco11	113.70	133.67	92.13	31.08	108.96	18.49
redeco12	1107.71	1246.92	826.37	33.73	895.12	28.21
reimer5	0.33	0.61	0.78	-27.87	1.22	-100.00
reimer6	10.88	14.61	12.15	16.84	14.47	0.96
reimer7	818.02	978.62	781.69	20.12	805.27	17.71

References

[1] Gerdt, V.P., Blinkov, Yu.A., Yanovich, D.A.: Construction of Janet Bases. II. Polynomial Bases. In: *Computer Algebra in Scientific Computing / CASC'01*, V.G.Ganzha, E.W.Mayr and E.V.Vorozhtsov (Eds.), Springer-Verlag, Berlin, 2001, pp.249-263.

[2] Janet, M.: *Leçons sur les Systèmes d'Equations aux Dérivées Partielles*, Cahiers Scientifiques, IV, Gauthier-Villars, Paris, 1929.

[3] Gerdt, V.P., Blinkov, Yu.A.: Involutive Bases of Polynomial Ideals. *Math. Comp. Sim.* **45** (1998) 519-542.

[4] Gerdt, V.P., Blinkov, Yu.A. Minimal Involutive Bases. *Math. Comp. Simul.* **45** (1998) 543-560.

[5] Apel, J.: Theory of Involutive Divisions and an Application to Hilbert Function. *J. Symb. Comp.* **25** (1998) 683-704.

[6] Gerdt, V.P.: Completion of Linear Differential Systems to Involution. In: *Computer Algebra in Scientific Computing / CASC'99*, V.G.Ganzha, E.W.Mayr and E.V.Vorozhtsov (Eds.), Springer-Verlag, Berlin, 1999, pp. 115-137.

[7] Calmet, J., Hausdorf, M., Seiler, W.M.: A Constructive Introduction to Involution. *Proc. Int. Symp. Applications of Computer Algebra - ISACA 2000*, Allied Publishers, New Delhi, 2001, pp. 33-50.

[8] Gerdt V.P., Blinkov Yu.A., Yanovich D.A.: Construction of Janet Bases. I. Monomial Bases. In: *Computer Algebra in Scientific Computing / CASC'01*, V.G.Ganzha, E.W.Mayr and E.V.Vorozhtsov (Eds.), Springer-Verlag, Berlin, 2001, pp.233-247.

[9] Gerdt, V.P.: On the Relation Between Pommaret and Janet Bases. In: *Computer Algebra in Scientific Computing / CASC'00*, V.G.Ganzha, E.W.Mayr and E.V.Vorozhtsov (Eds.), Springer-Verlag, Berlin, 2000, pp.167-181.

[10] Cox, D., Little, J., O'Shea, D.: *Ideals, Varieties and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra.* 2nd Edition, Springer-Verlag, New-York, 1996.

[11] Attardi, G., Traverso, C.: A Strategy-Accurate Parallel Buchberger Algorithm. In: *PASCO'94, Lecture Notes Series in Computing* **5**, World Scientific, Singapore, 1994, pp. 49-54.

[12] Faugère, J.C.: Parallelization of Gröbner Basis. In: *PASCO'94, Lecture Notes Series in Computing* **5**, World Scientific, Singapore, 1994, pp. 124-132.

[13] Amrheim, B., Gloor, O., Küchlin, W.: A Case Study of Multi-Threaded Gröbner Basis Completion. *Proceedings of ISSAC'96*, ACM Press, 1996, 95-102.

[14] Gerdt, V.P.: On an Algorithmic Optimization in Computation of Involutive Bases. *Programming and Computer Software* **2** (2002), to appear.

[15] Buchberger, B.: Gröbner Bases: an Algorithmic Method in Polynomial Ideal Theory. In: *Recent Trends in Multidimensional System Theory*, N.K. Bose (ed.), Reidel, Dordrecht (1985) 184–232.

[16] Bini, D., Mourrain, B.: *Polynomial Test Suite*, 1996.
http://www-sop.inria.fr/saga/POL.

[17] Verschelde, J.: The Database with Test Examples.
http://www.math.uic.edu/~jan/demo.html.

On Algebraic Geometric and Computer Algebra Aspects of Mirror Symmetry

N.M. Glazunov

Glushkov Institute of Cybernetics NAS,
03187 Glushkov prospect 40, Kiev-187 , Ukraine;
e-mail: glanm@yahoo.com

We survey some algebraic geometric aspects of mirror symmetry and duality in string theory. Some applications of computer algebra to algebraic geometry and string theory are shortly reviewed.

1 Introduction

This paper aims to be accessible to those with no previous experience in algebraic geometry; only some basic familiarity with linear and polynomial algebra, group theory, topology and category theory will be assumed.
Development of theoretical physics in direction of string theory enlarged the context of symmetry considerations and included in it the notion of duality. String theory has following ingredients: (i) base space (open or closed string) Σ; (ii) target space M; (iii) fields: $X \to \Sigma \to M$; (iv) action $S = \int \mathcal{L}(X, \varphi)$. where $\mathcal{L}$ is a Lagrangian [1]. Let G be a group such that $G \supset SU(3) \times SU(2) \times U(1)$. Recall that if $\mathcal{L}(G\Phi) = \mathcal{L}(\Phi)$ then $\mathcal{L}$ is $G-$invariant, or $G-$symmetry. In string theory [1] one of the beautiful symmetries is the radius symmetry $R \to 1/R$ of circle, known as $T-$duality [2, 3] and [4] and references there in. Authors of papers [5, 6] conjectured that a similar duality might exist in the context of string propagation on Calabi-Yau (CY) manifolds, where the role of the complex deformation on one manifold get exchanged with the Kähler deformation on the dual manifold. A pair of manifolds satisfying this symmetry is called *mirror pair*, and this duality is called *mirror symmetry*. From the point of view of physicists which did the remarkable discovery mirror symmetry is a type of duality that means that we may take two types of string theory and compactify them in two different ways and achieve "isomorphic" physics [7]. Or in the case of a pair of Calabi-Yau threefolds (X, Y) P. Aspinwall are said [8] that X and Y to be a *mirror pair* if and only if the type IIA string compactified on X is "isomorphic to" the $E_8 \times E_8$ heterotic string compactified on Y. If in the case X is Calabi-Yau threefold then Y is the product of a $K3$ surface and elliptic curve and the following data specifies the theory [8].

1. A Ricci-flat metric on Y.
2. A $B-$field $\in H^2(K3 \times E, \mathbf{R}/\mathbf{Z})$.

3. A vector bundle $V \to (K3 \times E)$ with a connection satisfying the Yang-Mills equations and whose structure group $\subseteq E_8 \times E_8$.

4. A dilation + axion, $\Phi \in \mathbf{C}$.

C. Vafa defines the notion of mirror of a Calabi-Yau manifold with a stable bundle. Lagrangian and special Lagrangian submanifolds appear in this situation. Mathematicians also work hard upon the problems of mirror symmetry, although it is difficult in some cases to attribute to a researcher the identifier "mathematician" or "physicist". V. Batyrev gives construction of mirror pairs using Gorenstein toric Fano varieties and Calabi-Yau hypersurfaces in these varieties [9]. M. Kontsevich in his talk at the ICM'94 gave a conjecturel interpratation of mirror symmetry as a "shadow" of an equivalence between two triangulated categories associated with $A_\infty-$ categories [10]. His conjecture was proved in the case of elliptic curves by A. Polishchuk and E. Zaslow [11]. The aim of the paper is to provide a short and gentle survey of some algebraic aspects of mirror symmetry and duality with examples - without proofs, but with (a very restricted) guides to the literature.

2 Algebraic geometric preliminaries

This section gives a basic introduction to algebraic geometric aspects of mirror symmetry beginning with a description of how Calabi-Yau manifolds arise from ringed spaces. A more detailed treatment of algebraic geometric material of this section may be found in [12, 13], the terminology and notation of which will be followed here.
Let X be a topological space and $Cov(X)$ an open covering of X. It is well known [14] that $Cov(X)$ forms the category. Let Cat be a category (of sets, abelian groups, rings, modules). The *presheaf* is a contravariant functor $\mathcal{F}$ from $Cov(X)$ to Cat. If for example Cat is the category Ab of abelian groups then $\mathcal{F} : Cov(X) \to Ab$ is the presheaf of abelian groups. Elements $f \in \mathcal{F}(\mathcal{U})$ is called *sections* of the presheaf $\mathcal{F}$.
If $i : U \subset V$ then we shall denote by ρ_U^V the morphism $\mathcal{F}(i) : \mathcal{F}(V) \to \mathcal{F}(U)$. Functor $\mathcal{F}$ is the contravariant and we apply morphisms from the left to the rigth. Hence for any open sets $U \subset V \subset W$

$$\rho_U^W = \rho_U^V \rho_V^W.$$

Let $U \subset X$ be any open subset of X and $\bigcup U_\alpha = U$ it's open covering. A presheaf $\mathcal{F}$ on a topological space X is called the *sheaf* if the following conditions are satisfied:
1) if $\rho_{U_\alpha}^U s_1 = \rho_{U_\alpha}^U s_2$ for $s_1, s_2 \in \mathcal{F}(U)$ and for any U_α, then $s_1 = s_2$;
2) if $s_\alpha \in \mathcal{F}(U_\alpha)$ are such that $\rho_{U_\alpha \cap U_\beta}^{U_\alpha} s_\alpha = \rho_{U_\alpha \cap U_\beta}^{U_\beta} s_\beta$, then there exists $s \in \mathcal{F}(U)$ such that $s_\alpha = \rho_{U_\alpha}^U s$ for all U_α.

The *ringed space* is the pair $(X, \mathcal{O})$, where X is a topological space and $\mathcal{O}$ is a sheaf of rings on X. A *morphism of ringed spaces*

$$\varphi : (X, \mathcal{O}_X) \to (Y, \mathcal{O}_Y)$$

is the class of maps (φ, ψ_U), where φ is the continuous map $\varphi : X \to Y$, and ψ_U is a homomorphism $\psi_U : \mathcal{O}_Y(U) \to \mathcal{O}_X(\varphi^{-1}(U))$ for any open set $U \subset Y$ such that the

diagram

$$\begin{array}{ccc} \mathcal{O}_X(\varphi^{-1}(V)) & \xrightarrow{\rho_U^V} & \mathcal{O}_X(\varphi^{-1}(U)) \\ \downarrow \psi_V & & \downarrow \psi_U \\ \mathcal{O}_Y(V) & \xrightarrow{\rho_U^V} & \mathcal{O}_Y(U) \end{array}$$

is commutative for any U and V, $U \subset V \subset Y$. For a ringed space $(X, \mathcal{O}_X)$ and an open $U \subset X$ the restriction of the sheaf $\mathcal{O}_X$ on U defines the ringed space $(U, \mathcal{O}_{X|U})$.
Let $\mathcal{O}_X$ be the sheaf of smooth functions on a topological space X. Then any smooth manifold X with the sheaf $\mathcal{O}_X$ is a ringed space $(X, \mathcal{O}_X)$. Respectively let X be a Hausdorff topological space and $\mathcal{O}_X$ a sheaf on X. Let it satisfies conditions: (i) $\mathcal{O}_X$ is the sheaf of algebras over $\mathbf{C}$; (ii) $\mathcal{O}_X$ is a subsheaf of the sheaf of continuous complex valued functions. Let W be a domain in $\mathbf{C}^n$ and $\mathcal{O}_{an}$ the sheaf of analytical functions on W. The ringed space $(X, \mathcal{O}_X)$ is called the *complex analytical manifold* if for any point $x \in X$ there exists a neighbourhood $U \ni x$ such that $(U, \mathcal{O}_{X|U}) \simeq (W, \mathcal{O}_{an})$ (here $\simeq$ denotes the isomorphism of ringed spaces).

We shall use in contrast to [1] some another definition of Calabi-Yau (CY) manifold. The definition is based on the theorem of Yau who proved Calabi's conjecture that a complex Kähler manifold of vanishing first Chern class admits a Ricci-flat metric.

Definition 1. *A complex Kähler manifold is called Calabi-Yau (CY) manifold if it has vanishing first Chern class.*

Examples of the CY-manifolds include, in particular, elliptic curves E, $K3-$surfaces and their products $E \times K3$. Let (X, ω, Ω) be a complex manifold (real dimension $=2n$) with

$$\omega^n/n! = (-1)^{n(n-1)/2}(i/2)^n \cdot \Omega \wedge \overline{\Omega}.$$

It is said that a $n-$dimensional submanifold $L \subset X$ is *special Lagrangian (s-lag)* $\Leftrightarrow$

$$Re(\Omega|_L) = Vol|_L \Leftrightarrow \omega|_L = 0, Im(\Omega|_L) = 0.$$

Example 1. *Let X be an elliptic curve E. Then $\omega = c(i/2)dz \wedge d\bar{z}$, $\Omega = cdz$. S-lag $L \subset E$ are straight lines with slope determined by* $\arg c$.

Let $(U, \mathcal{O}_{X|U}) \simeq SpecA$ for a commutative ring A. In the case the neighbourhood $U \ni x$ is called the *affine neighbourhood* of the point x.
The *scheme* S is the ringed space $(X, \mathcal{O}_X)$ with the condition: for any point $x \in X$ there is an affine neighbourhood $V \ni x$ such that $(V, \mathcal{O}_{X|V}) \simeq SpecA$.

2.1 Blow-ups

Blowing up is a well known method of constructing complex manifolds M. There are points on the manifolds that are not divisors on M. Blow up is the construction that transforms points of complex manifolds to divisors. For instance in the case of two dimensional complex manifolds (complex surface) N it consists of replacing a point $p \in N$ by a projective line $\mathbf{CP}(1)$ considered as the set of limit directions at p.

Example 2. *Let $\pi : M_2 \to \mathbf{C}^2$ be the blow-up of $\mathbf{C}^2$ at the point $0 \in \mathbf{C}^2$. Then M_2 is a two dimensional complex manifold that defined by two local charts. In coordinates $\mathbf{C}^2 = (z_1, z_2), \mathbf{CP}(1) = [l_0, l_1]$ the manifold M_2 is defined in $\mathbf{CP}(1) \times \mathbf{C}^2$ by quadratic equations $z_i l_j = z_j l_i$. Thus M_2 is a line bundle over Riemann sphere $\mathbf{CP}(1)$. $\pi^{-1}(0) = \mathbf{CP}(1)$ is called the divisor of the blow up (the exceptional divisor).*

Recently a large class of CY orbifolds in weighted projective spaces have been proposed. C. Vafa have predicted and S. Roan [17] have computed the Euler number of all the resolved CY hypersurfaces in a weighted projective space $\mathbf{WCP}(4)$.

3 Complexes, homotopy categories, cohomologies and quasiisomorphisms

Here we recall the relevant properties of complexes, derived categories, cohomologies and quasimorphisms referring to [13, 20] for details and indication of proofs.
Cochein complex

$$(K^\bullet, d) = \{K^0 \xrightarrow{d} K^1 \xrightarrow{d} K^2 \xrightarrow{d} \cdots\}$$

is the sequence of abelian groups and differentials $d : K^p \to K^{p+1}$ with the condition $d \circ d = 0$. For a category Cat we denote by $Ob\,Cat$ it's objects and by $Mor\,Cat$ it's morphisms.

Let A be an abelian category, $Kom(A)$ the category of complexes over A. Furthermore, there are various full subcategories of $Kom(A)$ whose respective objects are the complexes which are bounded below, bounded above, bounded in both sides. Now recall (by [20]) the notion of homotopy morphism.

Lemma-definition 1. *(i) Let $K^\bullet, L^\bullet$ be two complexes over abelian category A, $k = k^i$, $k^i : K^i \to L^{i-1}$ a sequence of morphisms between elements of the complexes. Then the maps*

$$h^i = k^{i+1} d_K^i + d_L^{i-1} k^i : K^i \to L^i$$

form the morphism of complexes

$$h = kd + dk : K^\bullet \to L^\bullet.$$

The morphism h is called homotopic *to zero ($h \sim 0$).*
(ii) morphisms $f, g : K^\bullet \to L^\bullet$ is called homotopic, if $f - g = kd + dk \sim 0$ ($f \sim g$), k is called homotopy.
(iii) If $f \sim g$, then $H^\bullet(f) = H^\bullet(g)$, where the map $H^\bullet$ is induced on cohomologies of complexes.

The *homotopy category* $K(A)$ is defined by the following way:

$$Ob\,K(A) = Ob\,Kom(A),$$

$Mor\,K(A) = Mor\,Kom(A)$ by the module of homotopy equivalence.

Let X be a topological space and $\mathcal{K}^\bullet, \mathcal{L}^\bullet$ be complexes of sheaves over X. *Quasiisomorphism* is the map

$$f : \mathcal{K}^\bullet \to \mathcal{L}^\bullet$$

which induces the isomorphism of cohomological sheaves

$$f_* : \mathcal{H}^q(\mathcal{K}^\bullet) \to \mathcal{H}^q(\mathcal{L}^\bullet),\ q \geq 0.$$

4 Connections

Consider the connection in the context of algebraic geometry. Let S/k be the smooth scheme over field k, U an element of open covering of S, $\mathcal{O}_S$ the structure sheaf on S, $\Gamma(U, \mathcal{O}_S)$ the sections of $\mathcal{O}_S$ on U. Let $\Omega^1_{S/k}$ be the sheaf of germs of 1−dimension differentials, $\mathcal{F}$ be a coherent sheaf. The *connection* on the sheaf $\mathcal{F}$ is the sheaf homomorphism

$$\nabla : \mathcal{F} \to \Omega^1_{S/k} \otimes \mathcal{F},$$

such that if $f \in \Gamma(U, \mathcal{O}_S)$, $g \in \Gamma(U, \mathcal{F})$ then

$$\nabla(fg) = f\nabla(g) + df \otimes g.$$

There is the dual definition. Let $\mathcal{F}$ be the locally free sheaf, $\Theta^1_{S/k}$ the dual to sheaf $\Omega^1_{S/k}$, $\partial \in \Gamma(U, \Theta^1_{S/k})$. The *connection* is the homomorphism

$$\rho : \Theta^1_{S/k} \to End_{\mathcal{O}_S}(\mathcal{F}, \mathcal{F}),$$

$$\rho(\partial)(fg) = \partial(f)g + f\rho(\partial).$$

4.1 Integration of connections

Let $\Omega^i_{S/k}$ be the sheaf of germs of i−dimensional differential forms on S. Particularly $\Omega^1_{S/k}$ is the cotangent bundle over S. Let $\omega \in \Omega^i_{S/k}$, $f \in \Gamma$ and

$$\nabla_i(\omega \otimes f) = d\omega \otimes f + (-1)^i \omega \wedge \nabla(f).$$

Hence, ∇_i define the sequence of homomorphisms of sheaves

$$\mathcal{F} \xrightarrow{\nabla} \Omega^1_{S/k} \otimes \mathcal{F} \xrightarrow{\nabla_1} \Omega^2_{S/k} \otimes \mathcal{F} \longrightarrow \cdots$$

The sequence is the *complex* if $\nabla \circ \nabla_1 = 0$. In this case the connection ∇ is integrable.

Example 3. *Let $\mathcal{F} = \mathcal{O}_S$ be the structural sheaf on S. Then*

$$\nabla : \mathcal{O}_S \to \Omega^1_{S/k} \otimes \mathcal{O}_S \simeq \Omega^1_{S/k}.$$

Hence $\nabla(f) = df$, $\rho : \Theta_{S/k} \to \mathcal{O}_S$. This connection ∇ is integrable because it defines the de Rham complex

$$\Omega^\bullet_{S/k} : \mathcal{O}_S \to \Omega^1_{S/k} \to \Omega^2_{S/k} \to \cdots$$

Example 4. *Let $\mathcal{LC}$ be a locally constant sheaf on S/k such that $\mathcal{LC} \simeq k^n$ (local coefficients) as sheaves. Let $\mathcal{F} = \mathcal{LC} \otimes \mathcal{O}_S$, $v \in \Gamma(U, \mathcal{LC})$, $f \in \Gamma(U, \mathcal{F})$. Then there is a* canonical connection $\nabla(v \otimes f) = df \otimes v$:

$$\nabla : \mathcal{F} \to \Omega^1_S \otimes \mathcal{F}.$$

For a connection $\nabla : \mathcal{F} \to \Omega^1_S \otimes \mathcal{F}$ a section $s \in \Gamma(U, \mathcal{F})$ is *horizontal* if $\nabla(s) = 0$.

Let now S be a complex manifolds, $ShC(S/k)$ the category of sheaves with a connection and $LC(S)$ the category of local coefficients over S. Let $(\mathcal{F}, \nabla) \in ShC(S/k)$ be a sheaf with connection. We can define the functor

$$Fn : (\mathcal{F}, \nabla) \mapsto \{\text{the sheaf of germs of horizontal sections of } \mathcal{F}\}.$$

and it's inverse Fn^{-1}.

Proposition 1. *The functors Fn and Fn^{-1} give the equivalence of category $ShC(S/k)$ and $LC(S)$.*

5 Moduli spaces in string theory

Mirror symmetry connects with geometrical deformations of complex and Kähler structures on CY-manifolds. So we have to know moduli spaces of complex and Kähler structures on CY-manifolds.

5.1 Moduli spaces

The theory of moduli spaces [15, 16] has, in recent years, become the meeting ground of several different branches of mathematics and physics - algebraic geometry, instantons, differential geometry, string theory and arithmetics. Here we recall some underlieing algebraic structures of the relation. In previous section we have reminded the situation with vector bundles on projective algebraic curves X. On X any first Chern class $c_1 \in H^2(X, \mathbf{Z})$ can be realized as c_1 of vector bundle of prescribed rank (dimension) r. How to classify vector bundles over algebraic varieties of dimension more than 1? This is one of important problems of algebraic geometry and the problem has closed connections with gauge theory in physics and differential geometry. Mamford [15] and others have formulated the problem about the determination of which cohomology classes on a projective variety can be realized as Chern classes of vector bundles? Moduli spaces are appeared in the problem. What is moduli? Classically Riemann claimed that $3g-3$ (complex) parameters could be for Riemann surface of genus g which would determine its conformal structure (for elliptic curves, when $g = 1$, it is needs one parameter). From algebraic point of view we have the following problem: given some kind of variety, classify the set of all varieties having something in common with the given one (same numerical invariants of some kind, belonging to a common algebraic family). For instance, for an elliptic curve the invariant is the modular invariant of the elliptic curve.

Let $\mathbf{B}$ be a class of objects. Let S be a scheme. A family of objects parametrized by the S is the set of objects

$$X_s : s \in S, X_s \in \mathbf{B}$$

equipped with an additional structure compatible with the structure of the base S. Parameter varieties is a class of moduli spaces. These varieties is very convenient tool for computer algebra investigation of objects that parametrized by the parameter varieties. We have used the approach for investigation of rational points of hyperelliptic curves over prime finite fields [22].

Example 5. *Let* $\omega_1, \omega_2 \in \mathbf{C}$, $Im(\omega_1/\omega_2) > 0$, $\Lambda = n\omega_1 + m\omega_2$, $n, m \in \mathbf{Z}$ *be a lattice. Let* H *be the upper half plane. Then* $H/\Lambda = E$ *be the elliptic curve. Let*

$$y^2 = x^3 + ax + b = (x - e_1)(x - e_2)(x - e_3),\ 4a^3 + 27b^2 \neq 0,$$

be the equation of E. *Then the differential of first kind on* E *is defined by formula*

$$\omega = dx/y = dx/(x^3 + ax + b)^{1/2}.$$

Periods of E :

$$\pi_1 = 2\int_{e_1}^{e_2} \omega,\ \pi_2 = 2\int_{e_2}^{e_3} \omega.$$

The space of moduli of elliptic curves over $\mathbf{C}$ *is* $\mathbf{A}^1(\mathbf{C})$. *Its complation is* $\mathbf{CP}(1)$.

For $K3-$surfaces the situation is more complicated but in some case is analogous [19]:

Theorem 1. *The moduli space of complex structure on market* $K3-$*surface (including orbifold points) is given by the space of possible periods.*

Some computational aspects of periods and moduli spaces are considered in author's note [23].

6 Some categorical constructions

Every compact symplectic manifold Y, ω with vanishing first Chern class, one can associate a $A_\infty-$category whose objects are essentially the Lagrangian submanifolds of Y, and whose morphisms are determined by the intersections of pairs of submanifolds. This category is called Fukaya's category and is denoted by $\mathcal{F}(Y)$ [10]. Let (X, Y) be a mirror pair. Let M be any element of the mirror pair. The bounded derived category $D^b(M)$ of coherent sheaves on M is obtained from the category of bounded complexes of coherent sheaves on M [20]. In the case of elliptic curves A. Poleshchuk and E. Zaslov have proved [11]:

Theorem 2. *The categories* $D^b(E_q)$ *and* $\mathcal{F}^0(\overline{E^q})$ *are equivalent.*

Recently A. Kapustin and D. Orlov have suggested that Kontsevich's conjecture must be modified: coherent sheaves must be replaced with modules over Azumaya algebras, and the Fukaya category must be "twisted" by closed 2-form [21].

7 Computer algebra aspects

Computer algebra applications to classical algebraic geometry are well known. Most of them are based on the method of Gröbner bases [24, 25]. They include the decomposition of algebraic varieties, rational parametrization of curves and surfaces [26, 27], inversion of birational maps [28], the normalization of affine rings [18]. Some computer algebra results presented on CAAP-2001 can be used for computations in algebraic geometry and string theory. These are results on computation of toric ideals presented by V. Gerdt [29] and on computation of cohomology presented by V. Kornyak [30]. Talks of V. Gerdt include also result on computation in Yang-Mills mechanics [31]. Some recent papers include description of efficient algorithms computing the homology of commutative differential graded algebras [32] and computing the complite Hopf algebra structure of the 1-homology of purely quadratic algebra [33].
For the future research let us mention that it might be as well to have a tool, namely computer algebra for computation with (i) various moduli spaces; (ii) deformations (deformation of complex structure and deformation of Kähler structure); (iii) $A_\infty-$categories; (iv) geometric Fourier transform.

Conclusions

In the paper we tried to give an algebraic geometric framework for some aspects of mirror symmetry. This framework includes rather restricted context of mirror symmetry and string theory. But it is based on a simple and unified mathematical base. Some applications of computer algebra to algebraic geometry and string theory are shortly reviewed.

Acknowledgements

I would like to thanks the organizers of the CAAP'2001 (Dubna) and SAGP'99 (Mirror Symmetry in String Theory, CIRM, Luminy) for providing a very pleasent environment during the conferences.

References

[1] B. Greene, String theory on Calabi-Yau manifolds, *Preprint,* 1997, hep-th/9702155.

[2] L. Dixon, Lecture at the 1987 ICTP Summer Workshop in High Energy Physics and Cosmology. *Trieste*, 1987.

[3] W. Lerche, C. Vafa, and N. Warner, Chiral rings in N=2 superconformal theories, *Nucl. Phys.*, 1989, B324, pp. 421-430.

[4] A. Giveon, M. Porrati,E. Rabinovici, Target Space Duality in String Theory, *Phys.Rept.*, 1994, N 244, pp. 77-202.

[5] P. Candelas, M. Lynker, R. Schimmrink. *Nucl. Phys.*, 1990, B330, p. 49.

[6] B. Greene, R. Plesser, *Nucl. Phys.*, 1990, B338, p. 15.

[7] C. Vafa, Geometric Physics, *Proceedinds of ICM'98, Berlin, Documenta Math. J. DMV*, 2000, pp. 430-439.

[8] P. Aspinwall, String theory and duality, *ibid*, pp. 229-238.

[9] V. Batyrev, Mirror symmetry and toric geometry, *ibid*, pp. 239-248.

[10] M. Kontsevich, Homological algebra of mirror symmetry, *Proceedinds of ICM'94, Zurich, Birkhauser*, 1995, pp. 120-139.

[11] A. Polishuk, E. Zaslow, Categorical mirror symmetry: the elliptic curves, *Adv.Theor.Math.Phys.*, 1998, N 2, pp. 443-470.

[12] I.R. Shafarevitch. Foundations of Algebraic Geometry. *Moscow: Nauka*, 1988, v.1,v.2 (in Russian).

[13] Ph. Griffiths, J. Harris. Principles of algebraic geometry. *A Wiley Int. Pub. N.Y.*, 1978.

[14] R. Godement. Topologie algebrique et theorie des faisceaux. *Hermann, Paris.*

[15] D. Mamford, Towards an enumirative geometry of the moduli space of curves. *Arithmetic and Geometry* Vol.II, Progress in Math., 1983.

[16] J. Harris, J. Morrison, Moduli of curves, *Berlin-N.Y, Springer*, 1998.

[17] S. Roan, Algebraic geometry and physics, *Preprint*, 2000, math-ph/0011038

[18] W. Decker, T. de Jong, G. Greuel, G. Pfister, The normalization: A new algorithm, implementation and comparisons. *"Computational methods for representations of groups and algebras". Proceedings of the Euroconference, (April 1-5, 1997, Essen) Basel, Birkhäuser*, 1999, pp. 177-185

[19] W. Barth, C. Peters, and van der Ven., Compact complex surfaces, *New York, Springer*, 1984.

[20] S.I. Gelfand, Yu. I. Manin, Methods of homological algebra, *Moscow, Nauka*, 1988.

[21] A. Kapustin, D. Orlov, Remarks on A-branes, Mirror Symmetry, and the Fukaya category, *Preprint*, 2001, hep-th/0109098

[22] N.M. Glazunov, On moduli spaces, equidistribution, bounds and rational points of algebraic curves. *Ukrainian Math. Journ.*, 2001, N 9, pp. 1174-1183.

[23] N.M. Glazunov, Computation of Moduli, Periods and Modular Symbols. *IMACS-ACA'98*. Session "Algebra and Geometry". Prague, 1998.
http://math.unm.edu/ACA/1998/sessions/geom/glazunov/index.html

[24] Buchberger B. An algerithm for finding a bases. *PhD thesis, Universitst Innsbruck, Institut fur Mathematik*, 1965.

[25] Buchberger B. Groebner bases: an algorithmic mathod in polynomial ideal theory. *Recent Trends in Multidimensional Systems Theory, chapter 6. D. Riedel Publ. Comp*, 1985.

[26] J. Sendra, F. Winkler. Symbolic parametrization of curves. *J. Symb. Comput.*, 1991, 12, pp. 607-632.

[27] J. Schicho. Rational parametrization of surfaces. *J. Symb. Comput.*, 1998, 26, pp. 1-29.

[28] J. Schicho. Inversion of Birational Maps with Gröbner Bases. *Lond. Math. Soc. Lect. Note Ser. Cambridge: Cambridge Univ. Press.* 1998, 251, pp. 495-503.

[29] V.P. Gerdt, Yu.A. Blinkov. Janet trees in computing of toric ideals. *Computer algebra and its applications to physics (CAAP-2001), Dubna*, 2001, p. 19.

[30] V.V. Kornyak. Extraction of "minimal" cochain subcomplexes. ibid, p. 10.

[31] V.P. Gerdt, A.M. Khvedelidze, D.M. Mladenov. On light-cone vertion of $SU(2)$ Yang-Mills mechanics. ibid, p. 12.

[32] V. Álvarez, J. Armario, P. Real, B. Silva. Homological perturbation theory ond computability of Hochschild and cyclic homologies of CDGAS. *Int. Conf. on Secondary calculus and cohomological physics, Moscow*, 1997, pp. 1-15.

[33] M. Jimenez, P. Real. "Coalgebra" structures on 1-homological models for commutative differential graded algebras. *preprint*, 2000, 15 p.

Differential Gröbner walk

Oleg Golubitsky

Moscow State University,
Laboratory of Computational Methods,
Faculty of Mechanics and Mathematics, MSU 119899
e-mail: oleg@shade.msu.ru

An algorithm which transforms a characteristic set of an ideal in the differential ring of differential polynomials w.r.t. one ranking into the characteristic set w.r.t. another ranking is proposed. The algorithm is a generalization of the Gröbner walk algorithm for transformation of Gröbner bases for polynomial ideals from one monomial ordering to another.

The work was supported by INTAS, grant 99-1222.

1 Introduction

In this paper, we propose an algorithm which, given a characteristic set $\mathcal{A}$ of a prime differential ideal I w.r.t. some ranking $\leq_1$, and another ranking $\leq_2$, constructs a characteristic set $\mathcal{B}$ of I w.r.t $\leq_2$.

The problem addressed in this paper can be solved using either of the Rosenfeld-Gröbner algorithm [19] or the Specialized Rosenfeld-Gröbner algorithm [21]. In both cases, however, the algorithms proceed by direct computation of the target characteristic set w.r.t. ranking $\leq_2$. Since this may be very inefficient, especially in case when $\leq_2$ is an elimination ranking, an algorithm (called DFGLM) based on an adaptation of the FGLM method [7] to differential algebra is proposed in [20]. However, as it is pointed out in [20], this algorithm is only applicable to differential systems whose solutions depend on finitely many constants. The Kähler algorithm [20] is applicable to any differential systems but, as it is noticed in [20], it is less efficient than DFGLM.

The case when solutions of a differential system depend on finitely many constants is completely analogous to the case of a zero-dimensional ideal in a polynomial ring R, to which the original FGLM algorithm is applicable. But even in this restricted case, the complexity of the algorithm grows with the dimension of quotient vector space R/I. Experiments in [9, 10] show that apart from very simple cases the Gröbner walk algorithm [8] is at least as fast as FGLM, and as the dimension of R/I grows, the Gröbner walk becomes much faster. Besides, the Gröbner walk approach is applicable to any polynomial ideals.

The key concepts needed for Gröbner walk are the ones of parametrization of all possible monomial orders by weight matrices [3, 11], monomial preorders defined by weight vectors [3, 11, 18], and the Gröbner fan [4, 11] of a polynomial ideal. In the differential case, the parametrization of rankings, in general, is very complex [18, 17]. Therefore, we restrict ourselves to the case of Riquier rankings [12, 18] which includes important cases

of orderly and elimination rankings [15, 20]. The Theorem of Mora and Robbiano about the finiteness of the Gröbner fan [4, 11] cannot be directly generalized to the differential case — in fact, the differential Gröbner fan may be infinite. Despite that, we will show that differential Gröbner walk always terminates.

We generalize the key Gröbner walk Lemmas [8, Lemmas 3.1–3.3] to the differential case, where they acquire a significantly different form. In particular, the polynomial ideal generated by the set of initial forms plays a key role for the polynomial Gröbner walk, but it cannot be generalized directly to the differential case. Instead, we consider the sets of initial forms themselves. However, the computation of characteristic sets for them is still algorithmically possible.

In the end, we discuss some open problems which should be addressed before the differential Gröbner walk can be implemented.

2 Differential algebra

Here we only give a short presentation of the basic concepts in differential algebra and refer the reader to [13, 14, 15].

Let R be a commutative ring. A *derivation* over R is a mapping $\delta : R \to R$ which for every $a, b \in R$ satisfies

$$\delta(a+b) = \delta(a) + \delta(b), \qquad \delta(ab) = \delta(a)b + a\delta(b).$$

A *differential ring* is a commutative ring endowed with a finite set of derivations $\Delta = \{\delta_1, \ldots, \delta_m\}$ which commute pairwise. The commutative monoid generated by the derivations is denoted by Θ. Its elements are *derivation operators* $\theta = \delta_1^{i_1} \cdots \delta_m^{i_m}$, where $i_1, \ldots, i_m$ are nonnegative integer numbers.

A *differential ideal* I of differential ring R is an ideal of R stable under derivation, i.e.

$$\forall A \in I, \delta \in \Delta \quad \delta A \in I.$$

For a subset $\mathcal{A} \subset R$, denote by $[\mathcal{A}]$ the smallest differential ideal containing $\mathcal{A}$. For $A, B \in R$, and differential ideal I, we shall write $A \equiv B \bmod I$, if $A - B \in I$.

Let $U = \{u_1, \ldots, u_n\}$ be a finite set called the set of *differential indeterminates*. Derivation operators apply to differential indeterminates giving *derivatives* θu. We denote by ΘU the set of all derivatives.

Let $\mathcal{K}$ be a differential field of characteristic zero. The differential ring of *differential polynomials* $\mathcal{K}\{U\}$ is the ring of polynomials of infinitely many variables $\mathcal{K}[\Theta U]$ endowed with the set of derivations Δ.

3 Rankings

Let m be a nonnegative integer and n be a positive integer. Let

$$\mathbb{N} = \{0, 1, 2, \ldots\}, \quad \mathbb{N}_n = \{1, \ldots, n\}.$$

A *ranking* is a total order $\leq$ of $\mathbb{N}^m \times \mathbb{N}_n$ such that for all $a, b, c \in \mathbb{N}^m$, $i, j \in \mathbb{N}_n$,

- $(a,i) \leq (b,j) \iff (a+c,i) \leq (b+c,j)$
- $(a,i) \geq (0,i)$.

Rankings on $\mathbb{N}^m \times \mathbb{N}_n$ correspond to those on the set of derivatives ΘU:

$$\delta_1^{i_1} \cdots \delta_m^{i_m} u_j \leq \delta_1^{k_1} \cdots \delta_m^{k_m} u_l \iff (i_1, \ldots, i_m, j) \leq (k_1, \ldots, k_m, l).$$

A ranking is a *Riquier ranking*, if for all $a, b \in \mathbb{N}^m$, $i, j \in \mathbb{N}_n$

$$(a,i) \leq (b,i) \iff (a,j) \leq (b,j).$$

Note that $\mathbb{N}^m \times \mathbb{N}_n$ may be embedded into $\mathbb{N}^{n+m}$ by

$$(i_1, \ldots, i_m, j) \longmapsto (i_1, \ldots, i_m, 0, \ldots, \overset{(m+j)}{1}, \ldots, 0).$$

Using this embedding, we can characterize Riquier rankings.

Theorem 1. *[18, Theorem 6] A Riquier ranking is a ranking $\leq$ for which there exists a positive integer s and an $s \times (m+n)$ real matrix M such that*

- *for $k = 1, \ldots, m$, k^{th} column c_k of M satisfies*

$$c_k \geq_{\text{lex}} (0, \ldots, 0) \tag{1}$$

- *$(i_1, \ldots, i_m, j) \leq (k_1, \ldots, k_m, l)$ if and only if*

$$M\big(i_1, \ldots, i_m, 0, \ldots, \overset{(m+j)}{1}, \ldots, 0\big) \leq_{\text{lex}} M\big(k_1, \ldots, k_m, 0, \ldots, \overset{(m+l)}{1}, \ldots, 0\big).$$

Vice versa, any $s \times (m+n)$ real matrix M of rank $m+n$ satisfying (1) defines a Riquier ranking $\leq_M$.

In a particular algorithm, for a ranking $\leq$, one makes finitely many comparisons w.r.t. $\leq$. In this case, ranking $\leq$ may be defined by an integer matrix, since the following theorem holds.

Theorem 2. *[18, Theorem 8] Let $\leq$ be a Riquier ranking on $\mathbb{N}^m \times \mathbb{N}_n$, and let $\mathcal{R}$ be a finite subset of $(\mathbb{N}^m \times \mathbb{N}_n)^2$ such that for all $(u,v) \in \mathcal{R}$, $u \leq v$. Then there exists a positive integer s and an $s \times (m+n)$ integer matrix M satisfying (1) such that for all $(u,v) \in \mathcal{R}$, $u \leq_M v$.*

4 Characteristic sets

Let $\leq$ be a Riquier ranking on the set of derivatives ΘU, $A \in \mathcal{K}\{U\}$, $A \notin \mathcal{K}$. The derivative θu_j of highest rank present in A is called the *leader* of A (denoted as $\mathrm{ld}_{\leq} A$). Let $\mathbf{u} = \mathrm{ld}_{\leq} A$ and $d = \deg_{\mathbf{u}} A$. Then $A = \sum_{i=0}^{d} I_i \mathbf{u}^i$, where $I_0, \ldots, I_d$ are uniquely defined polynomials free of $\mathbf{u}$. The differential polynomial $I_A = I_d$ is called the *initial* of A, and the differential polynomial $S_A = \sum_{i=1}^{d} i I_i \mathbf{u}^{i-1}$ is called the *separant* of A. The *leading polynomial* $\mathrm{lp}_{\leq} A$ is $I_A(\mathrm{ld}_{\leq} A)^d$.

Let $A, B \in \mathcal{K}\{U\}$. We shall say that A has *lower rank than* B (and write $A < B$) if either $A \in \mathcal{K}$, $B \notin \mathcal{K}$, or $\mathrm{ld}_{\leq} A \leq \mathrm{ld}_{\leq} B$, or $\mathrm{ld}_{\leq} A = \mathrm{ld}_{\leq} B = \mathbf{u}$ and $\deg_{\mathbf{u}} A < \deg_{\mathbf{u}} B$. In case when neither $A < B$, nor $B < A$, we will write $\mathrm{rk}_{\leq} A = \mathrm{rk}_{\leq} B$.

Let $A, F \in \mathcal{K}\{U\}$, $A \notin \mathcal{K}$. The differential polynomial F is called *partially reduced* w.r.t. A, if F is free of every proper derivative $\theta \mathbf{u}$ of the leader $\mathbf{u} = \mathrm{ld}_{\leq} A$ of A. If F is partially reduced w.r.t. A and $\deg_{\mathbf{u}} F < \deg_{\mathbf{u}} A$, then F is said to be *reduced* w.r.t. A. A nonempty subset $\mathcal{A} \subset \mathcal{K}\{U\}$ is called *autoreduced* if any element of $\mathcal{A}$ is reduced w.r.t. any other element of $\mathcal{A}$. Every autoreduced set is finite. If $\mathcal{A} = \{A_1, \ldots, A_r\}$ is an autoreduced set, then any two leaders $\mathrm{ld}_{\leq} A_i$, $\mathrm{ld}_{\leq} A_j$ for $1 \leq i \neq j \leq r$ are distinct, and we shall suppose that elements of any autoreduced set are arranged in order of increasing rank of their leaders $\mathrm{ld}_{\leq} A_1 < \mathrm{ld}_{\leq} A_2 < \cdots < \mathrm{ld}_{\leq} A_r$.

Let $\mathcal{A}$ be an autoreduced subset of $\mathcal{K}\{U\}$ w.r.t. $\leq$. If $F \in \mathcal{K}\{U\}$, then there exists a differential polynomial F_0 called the *remainder of* F (denoted $\mathrm{rem}_{\leq}(F, \mathcal{A})$) and $r_A, t_A \in \mathbb{N}$ $(A \in \mathcal{A})$ such that F_0 is reduced w.r.t. $\mathcal{A}$, the rank of F_0 is no higher than that of F, and $\prod_{A \in \mathcal{A}} I_A^{r_A} S_A^{t_A} F \equiv F_0 \bmod [\mathcal{A}]$. For a subset $\mathcal{B} \subset \mathcal{K}\{U\}$, denote by $\mathrm{rem}_{\leq}(\mathcal{B}, \mathcal{A})$ the set of remainders of the differential polynomials in $\mathcal{B}$ w.r.t. $\mathcal{A}$.

Let $\mathcal{A} = \{A_1, \ldots, A_r\}, \mathcal{B} = \{B_1, \ldots, B_s\}$ be two autoreduced sets. We shall say that $\mathcal{A}$ has *lower rank than* $\mathcal{B}$ and write $\mathcal{A} < \mathcal{B}$, if either there exists $k \in \mathbb{N}$ such that $\mathrm{rk}_{\leq} A_i = \mathrm{rk}_{\leq} B_i$ $(1 \leq i < k)$ and $A_k < B_k$, or $r > s$ and $\mathrm{rk}_{\leq} A_i = \mathrm{rk}_{\leq} B_i$ $(1 \leq i \leq s)$. If $r = s$ and $\mathrm{rk}_{\leq} A_i = \mathrm{rk}_{\leq} B_i$ $(1 \leq i \leq s)$, then $\mathcal{A}$ is said to *have the same rank as* $\mathcal{B}$ $(\mathrm{rk}_{\leq} \mathcal{A} = \mathrm{rk}_{\leq} \mathcal{B})$. Any nonempty set of autoreduced subsets contains an autoreduced set of the lowest rank. For a subset $I \subset \mathcal{K}\{U\}$, an autoreduced subset of I of the lowest rank is called a *characteristic set of* I. An autoreduced set $\mathcal{A}$ is a characteristic set of I if and only if I contains no nonzero elements reduced w.r.t. $\mathcal{A}$. If I is a differential ideal, and $\mathcal{A}$ is a characteristic set of I, then for each nonzero $F \in I$, $\mathrm{lp}_{\leq} F$ is not reduced w.r.t. $\mathcal{A}$.

5 Differential Gröbner fan

At this point, we are interested in a description of all possible characteristic sets w.r.t. different rankings. Contrary to the polynomial case, there are infinitely many characteristic sets of a differential ideal for a fixed ranking. However, the following lemma holds.

For a subset $\mathcal{A} \subset \mathcal{K}\{U\}$, let

$$\mathrm{ld}_{\leq}(\mathcal{A}) = \{\mathrm{ld}_{\leq}(A) \mid A \in \mathcal{A}\}$$

be the set of leaders of elements of $\mathcal{A}$.

Lemma 1. *Let $\leq$ be a (Riquier) ranking. Then all characteristic sets have the same sets of leaders.*

Proof. Let $\mathcal{A} = \{A_1, \ldots, A_r\}, \mathcal{B} = \{B_1, \ldots, B_s\}$ be two characteristic sets of a differential ideal I w.r.t. $\leq$. Then $\mathrm{rk}_{\leq} \mathcal{A} = \mathrm{rk} \leq \mathcal{B}$, i.e. $r = s$ and $\mathrm{rk}_{\leq} A_i = \mathrm{rk}_{\leq} B_i$ $(1 \leq i \leq s)$. Thus, $\mathrm{ld}_{\leq} A_i = \mathrm{ld}_{\leq} B_i$ $(1 \leq i \leq s)$ and $\mathrm{ld}_{\leq}(\mathcal{A}) = \mathrm{ld}_{\leq}(\mathcal{B})$. □

Let

$$\mathrm{Ld}(I) = \{\mathrm{ld}_{\leq}(\mathcal{A}) \mid \leq \text{ is a Riquier ranking and } \mathcal{A} \text{ is a characteristic set of } I \text{ w.r.t. } \leq\}.$$

Theorem 3. *For each differential ideal $I \subset \mathcal{K}\{U\}$, the set $\mathrm{Ld}(I)$ is finite.*

Proof. Suppose that $\mathrm{Ld}(I)$ is an infinite set. For each $N \in \mathrm{Ld}(I)$, let $\leq_N$ be the corresponding Riquier ranking. Then set $\Sigma = \{\leq_N \mid N \in \mathrm{Ld}(I)\}$ is infinite.

Let $A_0 \in I$ be a differential polynomial and $\mathcal{A}_0 = \{A_0\}$. Since A_0 contains only a finite number of terms, by the pigeonhole principle, there exists an infinite subset $\Sigma_1 \subset \Sigma$ such that $\mathrm{ld}_{\leq} A_0 = \mathrm{ld}_{\leq_1} A_0$ for all $\leq, \leq_1 \in \Sigma_1$.

If $\mathcal{A}_0$ is a characteristic set of I w.r.t some ranking $\leq_1 \in \Sigma_1$, then $\mathcal{A}_0$ is a characteristic set of I w.r.t. every $\leq \in \Sigma_1$:

1. $\mathcal{A}_0$ is autoreduced w.r.t. $\leq$.

2. All differential polynomials in I are reduced w.r.t. $\mathcal{A}_0$, $\leq$.

However, this cannot be the case since the original set of rankings Σ was chosen so that characteristic sets corresponding to distinct rankings in Σ have distinct sets of leaders. Hence, there must be some $A_1 \in I$ reduced w.r.t. $\mathcal{A}_0$ and $\leq_1$. Note that A_1 is also reduced w.r.t. $\mathcal{A}_0$ and any $\leq \in \Sigma_1$.

Let $\mathcal{A}_1 = \mathrm{rem}_{\leq_1}(\mathcal{A}_0, \{A_1\})$. Then $\mathcal{A}_1$ is autoreduced. Now we apply the pigeonhole principle again to find an infinite subset $\Sigma_2 \subseteq \Sigma_1$ such that $\mathrm{ld}_{\leq}(\mathcal{A}_1) = \mathrm{ld}_{\leq_2}(\mathcal{A}_1)$ for all $\leq, \leq_2 \in \Sigma_2$. The above argument shows that $\mathcal{A}_1$ cannot be a characteristic set of I w.r.t. any of the rankings in Σ_2 and there exists a differential polynomial $A_2 \in I$ reduced w.r.t. $\mathcal{A}_1$ and every $\leq \in \Sigma_2$. Note that A_2 is also reduced w.r.t. $\{A_0, A_1\}$ and any $\leq \in \Sigma_2$.

Proceeding in the same way, we construct an infinite sequence of differential polynomials $A_0, A_1, A_2, \ldots$ and rankings $\leq_1, \leq_2, \ldots$ such that A_i is reduced w.r.t. $\{A_0, \ldots, A_{i-1}\}$ and $\leq_i$ $(i > 0)$, and $\mathrm{ld}_{\leq_j} A_i = \mathrm{ld}_{\leq_i} A_i$ for all $j > i$. Therefore, for all $j > i$, $\mathrm{ld}_{\leq_j} A_j \,/\, |\mathrm{ld}_{\leq_i} A_i$.

One can find an infinite subsequence $i_1, i_2, \ldots$ such that $\mathrm{ld}_{\leq_{i_1}} A_{i_1}, \mathrm{ld}_{\leq_{i_2}} A_{i_2}, \ldots$ are derivatives of the same differential indeterminate $u \in U$. Let $\theta_{i_1}, \theta_{i_2}, \ldots$ be the sequence of derivation operators such that $\theta_{i_k} u$ is the leader of A_{i_k} $(k > 0)$. Then for all $0 < k < l$, we cannot have that $\theta_{i_l} | \theta_{i_k}$, and this contradicts the Dickson's lemma. □

A *marked characteristic set* of a differential ideal I is a set of differential polynomials $\bar{\mathcal{A}}$, together with identified leaders in each $\bar{A} \in \bar{\mathcal{A}}$ such that $\bar{\mathcal{A}}$ is a characteristic set w.r.t. some ranking selecting those leaders. For a marked characteristic set $\bar{\mathcal{A}}$, the leader of a polynomial $\bar{A} \in \bar{\mathcal{A}}$ will be denoted by $\mathrm{ld}\,\bar{A}$. The set of leaders of $\bar{\mathcal{A}}$ will be denoted by $\mathrm{ld}\,\bar{\mathcal{A}}$. Our next goal is to understand the set of all rankings for which $\bar{\mathcal{A}}$ is a marked characteristic set of I.

By Theorem 1, each Riquier ranking $\leq$ is defined by some matrix M. Thus, when we determine the leader of a polynomial, we first compare the derivatives using the first row of the matrix, i.e. a weight vector $\mathbf{w}$.

Let $\mathbf{w} \in (\mathbb{R}^{m+n})^+$ be a weight vector. For a derivative $\mathbf{u} = \delta_1^{i_1} \cdots \delta_m^{i_m} u_j$, let the $\mathbf{w}$-*degree of* $\mathbf{u}$ be defined as the following inner product:

$$\deg_{\mathbf{w}} \mathbf{u} = \mathbf{w} \cdot (i_1, \ldots, i_m, 0, \ldots, \overset{(m+j)}{1}, \ldots, 0).$$

For a differential monomial $\tau = c \prod_\alpha \mathbf{u}_\alpha \notin \mathcal{K}$, define

$$\deg_{\mathbf{w}} \tau = \max_\alpha \deg_{\mathbf{w}} \mathbf{u}_\alpha.$$

Let $\bar{\mathcal{A}}$ be a marked characteristic set of I. We define

$$C_{\bar{\mathcal{A}}} = \{\mathbf{w} \in (\mathbb{R}^{m+n})^+ \mid \deg_{\mathbf{w}}(\mathrm{ld}\, \bar{A}) \geq \deg_{\mathbf{w}} \mathbf{u} \text{ for all derivatives } \mathbf{u} \text{ present in } \bar{A}, \bar{A} \in \bar{\mathcal{A}}\}.$$

The elements of $C_{\bar{\mathcal{A}}}$ correspond to all possible first rows of matrices M defining rankings $\leq_M$ such that $\bar{\mathcal{A}}$ is a marked characteristic set w.r.t. $\leq_M$. It is easy to see from the definition that $C_{\bar{\mathcal{A}}}$ is an intersection of closed half-spaces in $\mathbb{R}^{m+n}$, hence is a closed convex polyhedral cone contained in the positive orthant. We will call $C_{\bar{\mathcal{A}}}$ a *differential Gröbner cone*, and the set of all such cones for a differential ideal I the *differential Gröbner fan* of I.

Contrary to the polynomial case, a differential Gröbner fan may be infinite. For example, let $I = [\frac{\partial u}{\partial x}]$. Then $\bar{\mathcal{A}}_0 = \{\frac{\partial u}{\partial x}\}$ is a characteristic set of I w.r.t. lexicographic ranking with $x > y$. Also, for every $i > 0$,

$$\bar{\mathcal{A}}_i = \left\{ \frac{\partial^i u}{\partial y^i} \cdot \overline{\frac{\partial u}{\partial x}} \right\}$$

is a marked characteristic set w.r.t. the same ranking, and

$$\mathrm{Ld}(\bar{\mathcal{A}}_i) = \left\{ \frac{\partial u}{\partial x} \right\}.$$

The differential Gröbner cone for $\bar{\mathcal{A}}_i$ is

$$C_{\bar{\mathcal{A}}_i} = \{(w_1, w_2, w_3) \in (\mathbb{R}^3)^+ \mid w_1 > w_2 i\},$$

hence

$$C_{\bar{\mathcal{A}}_1} \supset C_{\bar{\mathcal{A}}_2} \supset \ldots,$$

where all inclusions are strict. In Section 8, we will see that despite the total number of differential Gröbner cones for a differential ideal may be infinite, the differential Gröbner walk still terminates in finitely many steps.

6 Characteristic set conversion

We say that a weight vector $\mathbf{w}$ is *compatible* with a ranking $\leq$, if for all derivatives $\mathbf{u}_1, \mathbf{u}_2 \in \Theta U$, $\deg_{\mathbf{w}} \mathbf{u}_1 < \deg_{\mathbf{w}} \mathbf{u}_2$ implies $\mathbf{u}_1 < \mathbf{u}_2$.

Let $A = \tau_1 + \ldots + \tau_k$ be a differential polynomial represented as a sum of differential monomials. Let $J = \{i_1, \ldots, i_l\}$ be a subset of $\{1, \ldots, k\}$ such that for all $j, j' \in J, i \in \{1, \ldots, k\} \setminus J$,

$$\deg_{\mathbf{w}} \tau_j = \deg_{\mathbf{w}} \tau_{j'}, \qquad \deg_{\mathbf{w}} \tau_j > \deg_{\mathbf{w}} \tau_i.$$

We will call differential polynomial

$$\mathrm{in}_{\mathbf{w}} A = \sum_{j \in J} \tau_j$$

the $\mathbf{w}$-*initial form* of A. Intuitively, $\mathrm{in}_{\mathbf{w}} A$ is the sum of all differential monomials present in A having the highest $\mathbf{w}$-degree. For a subset $\mathcal{A} \subset \mathcal{K}\{U\}$,

$$\mathrm{in}_{\mathbf{w}}(\mathcal{A}) = \{\mathrm{in}_{\mathbf{w}}(A) \mid A \in \mathcal{A}\}.$$

Lemma 2. *If* $\mathbf{w}$ *is compatible with* $\leq$ *and* $A \in \mathcal{K}\{U\}$ *then*

$$\begin{aligned}
\mathrm{ld}_{\leq} A &= \mathrm{ld}_{\leq}(\mathrm{in}_{\mathbf{w}} A) && (2)\\
\mathrm{lp}_{\leq} A &= \mathrm{lp}_{\leq}(\mathrm{in}_{\mathbf{w}} A) && (3)\\
I_A &= I_{\mathrm{in}_{\mathbf{w}} A} && (4)\\
S_A &= S_{\mathrm{in}_{\mathbf{w}} A} && (5)\\
\mathrm{rk}_{\leq} A &= \mathrm{rk}_{\leq}(\mathrm{in}_{\mathbf{w}} A) && (6)
\end{aligned}$$

Proof. Let $A = \tau_1 + \ldots + \tau_k$ be a representation of A as a sum of differential monomials where τ_1 contains the highest degree of $\mathrm{ld}_{\leq} A$. Then $\deg_{\mathbf{w}} \tau_1 \geq \deg_{\mathbf{w}} \tau_i$ $(i \in \{1, \ldots, k\})$ since $\mathbf{w}$ is compatible with $\leq$. Hence τ_1 is present in $\mathrm{in}_{\mathbf{w}} A$ and we obtain (2) and (6). Moreover, all differential monomials that contain $\mathrm{ld}_{\leq} A$ are present in $\mathrm{in}_{\mathbf{w}} A$, therefore (3), (4), and (5) hold. $\square$

A differential polynomial $A = \tau_1 + \ldots + \tau_k$ is called $\mathbf{w}$-*homogeneous*, if $\deg_{\mathbf{w}} \tau_1 = \ldots = \deg_{\mathbf{w}} \tau_k$, and by definition $\deg_{\mathbf{w}} A = \deg_{\mathbf{w}} \tau_1$. If A, B are two $\mathbf{w}$-homogeneous differential polynomials such that $\deg_{\mathbf{w}} A = \deg_{\mathbf{w}} B$, and τ is a differential polynomial such that $\deg_{\mathbf{w}} \tau \leq \deg_{\mathbf{w}} A$, then $A + B$ and τA are also $\mathbf{w}$-homogeneous. However, θA, where $\theta \in \Theta$, is not necessarily $\mathbf{w}$-homogeneous. For example, let $\Delta = \{\delta\}$, $A = uv \in \mathcal{K}\{u, v\}$, $\mathbf{w} = (1, 1, 0)$. Then A is $\mathbf{w}$-homogeneous, since A is a monomial, but $\delta A = v\delta u + u\delta v$ is not, since

$$\begin{aligned}
\deg_{\mathbf{w}}(v\delta u) &= \max(\deg_{\mathbf{w}}(v), \deg_{\mathbf{w}}(\delta u)) = \max(0, 2) = 2\\
\deg_{\mathbf{w}}(u\delta v) &= \max(\deg_{\mathbf{w}}(u), \deg_{\mathbf{w}}(\delta v)) = \max(1, 1) = 1
\end{aligned}$$

Lemma 3. *Let* $\leq$ *be a ranking and let* $\mathbf{w}$ *be compatible with* $\leq$. *Let* $\mathcal{A} = \{A_1, \ldots, A_r\}$ *be a characteristic set for an ideal* I *w.r.t.* $\leq$. *Then* $\mathrm{in}_{\mathbf{w}}(\mathcal{A})$ *is a characteristic set of* $\mathrm{in}_{\mathbf{w}}(I)$ *w.r.t.* $\leq$.

Proof. It follows from (2) that $\mathrm{in}_{\mathbf{w}}(\mathcal{A})$ is autoreduced.

If $\bar{F} \in \mathrm{in}_{\mathbf{w}}(\mathcal{A})$, then there exists $F \in \mathcal{A}$ such that $\mathrm{in}_{\mathbf{w}} F = \bar{F}$. According to (3), $\mathrm{lp}_{\leq} F = \mathrm{lp}_{\leq} \bar{F}$. Since $\mathcal{A}$ is a characteristic set for I w.r.t. $\leq$, $\mathrm{lp}_{\leq} F$ is not reduced w.r.t. $\mathcal{A}$, i.e. it is reducible by some $A \in \mathcal{A}$. It follows from (2) that $\mathrm{lp}_{\leq} F$ is then reducible by $\mathrm{in}_{\mathbf{w}} A$, and hence $\bar{F}$ is reducible by $\mathrm{in}_{\mathbf{w}} A$. $\square$

We are now ready to introduce the *differential Gröbner walk*. Assume that we have a characteristic set $\mathcal{A}_0$ for I w.r.t. some ranking $\leq_s$. We call $\leq_s$ the *starting ranking* for the walk, and we will assume that we have some matrix $M_s = \begin{pmatrix} \mathbf{w}_0 \\ N \end{pmatrix}$ representing $\leq_s$. The goal is to compute a characteristic set for I w.r.t. some other given *target ranking* $\leq_t$ represented by matrix $M_t = \begin{pmatrix} \mathbf{w}_t \\ N' \end{pmatrix}$. Consider the segment $[\mathbf{w}_0, \mathbf{w}_t] = \{\xi \mathbf{w}_t + (1 - \xi)\mathbf{w}_0 \mid \xi \in [0,1]\}$ — though that is not always the best choice and one might consider more complex "perturbed" paths from $\mathbf{w}_0$ to $\mathbf{w}_t$ as in [10]. The Gröbner walk consists of a sequence of conversion steps performed each time the path crosses the boundary of a differential Gröbner cone, yielding a characteristic set of I corresponding to each new cone in turn. By the time we reach the end of the path we have the characteristic set w.r.t. the target ranking.

The conversion step is justified by the following Lemma.

Lemma 4. *Let $\mathcal{A}_{old}$ be a characteristic set for a differential ideal I w.r.t. some ranking $\leq_{old}$. Let $\mathbf{w}$ be a weight vector compatible with $\leq_{old}$ satisfying the following condition: For all $A \in \mathcal{A}_{old}$, $\theta \in \Theta$, $\theta \neq 1$,*

$$\deg_{\mathbf{w}}(I_A) < \deg_{\mathbf{w}}(\mathrm{ld}_{\leq_{old}} A), \qquad \deg_{\mathbf{w}}(S_A) < \deg_{\mathbf{w}}(\theta\, \mathrm{ld}_{\leq_{old}} A), \tag{7}$$

where initial and separant are taken w.r.t. $\leq_{old}$.[1] Let $\leq_{new}$ be any other monomial ordering compatible with $\mathbf{w}$. By Lemma 3, since $\mathbf{w}$ is compatible with $\leq_{old}$, $\mathrm{in}_{\mathbf{w}}(\mathcal{A}_{old})$ is a characteristic set for $\mathrm{in}_{\mathbf{w}}(I)$ w.r.t. $\leq_{old}$. Let $\bar{\mathcal{B}} = \{\bar{B}_1, \ldots, \bar{B}_s\}$ be a characteristic set for $\mathrm{in}_{\mathbf{w}}(I)$ w.r.t. $\leq_{new}$.[2] Then, one can efficiently compute a characteristic set $\mathcal{A}_{new}$ for I w.r.t. $\leq_{new}$.

Proof. For each $i \in \{1, \ldots, s\}$, since $\mathrm{in}_{\mathbf{w}}(\mathcal{A}_{old})$ is a characteristic set for $\mathrm{in}_{\mathbf{w}}(I)$ w.r.t. $\leq_{old}$, $\bar{B}_i$ is reducible w.r.t. $\mathcal{A}_{old}$, $\leq_{old}$. So, there exists a differential monomial of the form $\tau\theta(\mathrm{ld}_{\leq_{old}} A)$ present in $\bar{B}_i$ ($\theta \in \Theta$, τ is a differential monomial). There are two cases possible:

1. $\theta \neq 1$. Take $\bar{B}'_i = S_A \bar{B}_i - \mathrm{in}_{\mathbf{w}}(\tau\theta A)$. Since by assumption (7) $\deg_{\mathbf{w}}(S_A) < \deg_{\mathbf{w}}(\theta\, \mathrm{ld}_{\leq_{old}} A)$, we have that $\deg_{\mathbf{w}}(S_A) < \deg_{\mathbf{w}} \bar{B}_i = \deg_{\mathbf{w}}(\tau\theta A)$. Since $\bar{B}_i \in \mathrm{in}_{\mathbf{w}}(I)$, we have that $\bar{B}_i = \mathrm{in}_{\mathbf{w}} B_i$ for some $B_i \in I$, and $\mathrm{in}_{\mathbf{w}}(S_A B_i) = S_A \bar{B}_i$. Therefore, $\bar{B}'_i = \mathrm{in}_{\mathbf{w}}(S_A B_i) - \mathrm{in}_{\mathbf{w}}(\tau\theta A)$ and $\deg_{\mathbf{w}}(S_A B_i) = \deg_{\mathbf{w}}(\tau\theta A)$, hence $\bar{B}'_i \in \mathrm{in}_{\mathbf{w}}(I)$.

2. $\theta = 1$. Take $\bar{B}'_i = I_A \bar{B}_i - \mathrm{in}_{\mathbf{w}}(\tau A)$. Since by (7) $\deg_{\mathbf{w}}(I_A) < \deg_{\mathbf{w}}(\mathrm{ld}_{\leq_{old}} A)$, we have similarly that $\bar{B}'_i \in \mathrm{in}_{\mathbf{w}}(I)$.

[1] If assumptions (7) do not hold, the conversion step cannot be performed as described below. Then one can either compute the characteristic set w.r.t. $\leq_{new}$ directly and then proceed along the Gröbner walk, or perturb the path and find another $\mathbf{w}$ such that conditions (7) hold.

[2] The question, how to compute $\bar{\mathcal{B}}$, will be briefly discussed in Section 9.

Continuing to "reduce" $\bar{B}_i'$ in the same way, we finally obtain zero since every reduction chain is finite and zero is the only element in $\mathrm{in}_{\mathbf{w}}(I)$ irreducible w.r.t. $\mathcal{A}_{old}$. Therefore, there exist polynomials q, p_α, τ_α ($\alpha \in \mathcal{I}$) such that $\deg_{\mathbf{w}} q < \deg_{\mathbf{w}} \bar{B}_i$, $\deg_{\mathbf{w}} p_\alpha < \deg_{\mathbf{w}}(\tau_\alpha \theta_\alpha A_\alpha)$ and

$$\begin{aligned} q\bar{B}_i &= \textstyle\sum_{\alpha\in\mathcal{I}} p_\alpha \,\mathrm{in}_{\mathbf{w}}(\tau_\alpha\theta_\alpha A_\alpha) \\ &= \textstyle\sum_{\alpha\in\mathcal{I}} \mathrm{in}_{\mathbf{w}}(p_\alpha\tau_\alpha\theta_\alpha A_\alpha) \\ &= \mathrm{in}_{\mathbf{w}}(\textstyle\sum_{\alpha\in\mathcal{I}} p_\alpha\tau_\alpha\theta_\alpha A_\alpha). \end{aligned} \tag{8}$$

Take

$$B_i = \sum_\alpha p_\alpha\tau_\alpha\theta_\alpha A_\alpha, \quad \mathcal{B} = \{B_1, \ldots, B_s\}.$$

Then $\mathrm{ld}_{\leq_{new}} B_i = \mathrm{ld}_{\leq_{new}} \bar{B}_i$, hence every polynomial in $\mathrm{in}_{\mathbf{w}}(I)$ is reducible w.r.t. $\mathcal{B}$, $\leq_{new}$. Moreover, $\mathrm{in}_{\mathbf{w}} B_i$ is reduced w.r.t. $\mathcal{B} \setminus \{B_i\}$, for each $i \in \{1, \ldots, s\}$, since $\bar{\mathcal{B}}$ is autoreduced. Hence, set

$$\mathcal{A}_{new} = \{\mathrm{rem}_{\leq_{new}}(B_i, \mathcal{B} \setminus \{B_i\}) \mid 1 \leq i \leq s\}$$

is a characteristic set of I w.r.t. $\leq_{new}$, and the conversion step is completed. □

The characteristic set $\mathcal{A}_{new}$ w.r.t. $\leq_{new}$ computed from characteristic set $\mathcal{A}_{old}$ w.r.t. $\leq_{old}$ by the above procedure will be denoted by $\mathrm{conv}(\mathcal{A}_{old}, \leq_{old}, \leq_{new})$.

7 Differential Gröbner walk

The differential Gröbner walk consists of the following conversion steps.

First, let $\leq_1$ be a ranking defined by the matrix $\begin{pmatrix} \mathbf{w}_0 \\ M_t \end{pmatrix}$. Since both $\leq_1$ and $\leq_0$ are compatible with $\mathbf{w}_0$, we can apply Lemma 4 and compute

$$\mathcal{A}_1 = \mathrm{conv}(\mathcal{A}_0, \leq_0, \leq_1).$$

Each next step is performed as follows. Suppose, we have computed the characteristic set $\mathcal{A}_i$ w.r.t. $\leq_i$ defined by matrix $\begin{pmatrix} \mathbf{w}_{i-1} \\ M_t \end{pmatrix}$. Let $\mathbf{w}_i$ be the closest to $\mathbf{w}_{i-1}$ point on the segment $(\mathbf{w}_{i-1}, \mathbf{w}_t]$ such that the leader of some differential polynomial in $\mathcal{A}_i$ and some other derivative present in that polynomial have the same $\mathbf{w}$-degree. Since $\mathcal{A}_i$ is finite, and every $A \in \mathcal{A}_i$ involves finitely many derivatives, there are finitely many (or zero) possibilities for $\mathbf{w}_i$, and one can always choose the minimal one out of them. If such a point does not exist, $\mathcal{A}_i$ already is a characteristic set w.r.t. $\leq_t$, and the algorithm stops. Otherwise, let $\leq_i'$ be a ranking defined by the matrix

$$\begin{pmatrix} \mathbf{w}_i \\ \mathbf{w}_{i-1} \\ M_t \end{pmatrix}.$$

Then $\leq_i'$ is compatible with $\mathbf{w}$.

Lemma 5. *$\mathcal{A}_i$ is a characteristic set of I w.r.t. $\leq_i'$.*

Proof. In order to prove that $\mathcal{A}_i$ is a characteristic set w.r.t. $\leq_i'$, we shall prove that for every $A \in \mathcal{A}_i$, $\mathrm{ld}_{\leq_i'}(A) = \mathrm{ld}_{\leq_i}(A)$ and then use the fact that $\mathcal{A}_i$ is a characteristic set w.r.t. $\leq_i$. Suppose, there exists $A \in \mathcal{A}_i$ such that $\mathbf{u}' = \mathrm{ld}_{\leq_i'}(A) \neq \mathrm{ld}_{\leq_i}(A) = \mathbf{u}$. Then,

$$\deg_{\mathbf{w}_i} \mathbf{u} < \deg_{\mathbf{w}_i} \mathbf{u}' \quad (9)$$

$$\deg_{\mathbf{w}_{i-1}} \mathbf{u} \geq \deg_{\mathbf{w}_{i-1}} \mathbf{u}' \quad (10)$$

There are two cases possible:

1. $\deg_{\mathbf{w}_{i-1}} \mathbf{u} > \deg_{\mathbf{w}_{i-1}} \mathbf{u}'$. Then there exists a point $\mathbf{w} \in (\mathbf{w}_{i-1}, \mathbf{w}_i)$ such that $\deg_{\mathbf{w}} \mathbf{u} = \deg_{\mathbf{w}} \mathbf{u}'$ which contradicts to our choice of $\mathbf{w}_i$.

2. $\deg_{\mathbf{w}_{i-1}} \mathbf{u} = \deg_{\mathbf{w}_{i-1}} \mathbf{u}'$. Then $\deg_{\mathbf{w}_t} \mathbf{u} \geq \deg_{\mathbf{w}_t} \mathbf{u}'$. Moreover, we cannot have $\deg_{\mathbf{w}_t} \mathbf{u} = \deg_{\mathbf{w}_t} \mathbf{u}'$ since there is at most one point $\mathbf{w}$ on the segment $[\mathbf{w}_{i-1}, \mathbf{w}_t]$ having the property $\deg_{\mathbf{w}} \mathbf{u} = \deg_{\mathbf{w}} \mathbf{u}'$. Thus $\deg_{\mathbf{w}_t} \mathbf{u} > \deg_{\mathbf{w}_t} \mathbf{u}'$. Since $\deg_{\mathbf{w}} \mathbf{u}$ is a linear function of $\mathbf{w}$, for every $\mathbf{w} \in (\mathbf{w}_{i-1}, \mathbf{w}_t]$, $\deg_{\mathbf{w}} \mathbf{u} > \deg_{\mathbf{w}_t} \mathbf{u}'$ which again contradicts to our choice of $\mathbf{w}_i$.

□

Let $\leq_{i+1}$ be the ranking defined by the matrix $\begin{pmatrix} \mathbf{w}_i \\ M_t \end{pmatrix}$. Compute

$$\mathcal{A}_{i+1} = \mathrm{conv}(\mathcal{A}_i, \leq_i', \leq_{i+1}),$$

and then proceed to the next step.

In the next Section, we show that the Gröbner walk algorithm always stops after a finite number of conversion steps.

8 Termination of Gröbner walk

During the Gröbner walk procedure, we compute a sequence of characteristic sets $\mathcal{A}_1, \mathcal{A}_2, \ldots$ w.r.t. rankings $\leq_1, \leq_2, \ldots$, where $\leq_i$ are represented by matrices of the form $\begin{pmatrix} \mathbf{w}_{i-1} \\ M_t \end{pmatrix}$ $(i > 0)$, and $\{\mathbf{w}_i\}$ is a sequence of consecutive distinct points on the segment $(\mathbf{w}_0, \mathbf{w}_t]$.

Theorem 4. *The sequences $\mathcal{A}_1, \mathcal{A}_2, \ldots$ and $\mathbf{w}_1, \mathbf{w}_2, \ldots$ are finite.*

Proof. Assume that the sequences are infinite. For every $i > 0$, consider set

$$K_i = \{k > 0 \mid \mathrm{ld}_{\leq_i}(\mathcal{A}_i) = \mathrm{ld}_{\leq_k}(\mathcal{A}_k)\}.$$

Obviously, for every $i, j > 0$, either $K_i = K_j$, or $K_i \cap K_j = \varnothing$, and $\cup_{i=1}^{\infty} K_i = \mathbb{N} \setminus \{0\}$. According to Theorem 3, the set $\{K_j\}_{j \in \mathbb{N} \setminus \{0\}}$ is finite. We shall prove that every K_j $(j \in \mathbb{N} \setminus \{0\})$ is also finite and thus show that our assumption about the infiniteness of the sequences above was wrong.

Take the minimal element $i \in K_j$. If K_j is infinite, then $\mathbf{w}_i \neq \mathbf{w}_t$ and thus $\mathbf{w}_i \in (\mathbf{w}_0, \mathbf{w}_t)$. Then, for any $k > i$, $k \in K_j$, there exists differential polynomial $A_k \in \mathcal{A}_k$, and derivative $\mathbf{u}_k$ present in A_k such that

$$\deg_{\mathbf{w}_k}(\mathbf{u}_k) = \deg_{\mathbf{w}_k}(\mathrm{ld}_{\leq_k} A_k), \tag{11}$$

and $\mathbf{w}_k$ is the closest to $\mathbf{w}_{k-1}$ vector along the segment $(\mathbf{w}_{k-1}, \mathbf{w}_t]$ such that the equality (11) holds. That is, for every $\mathbf{w} \in (\mathbf{w}_{k-1}, \mathbf{w}_k)$,

$$\deg_{\mathbf{w}}(\mathbf{u}_k) < \deg_{\mathbf{w}}(\mathrm{ld}_{\leq_k} A_k). \tag{12}$$

Since $\deg_{\mathbf{w}}(\mathbf{u}_k)$ is a linear function of $\mathbf{w}$, it follows from (11,12) that

$$\deg_{\mathbf{w}_i}(\mathbf{u}_k) < \deg_{\mathbf{w}_i}(\mathrm{ld}_{\leq_k} A_k). \tag{13}$$

Let $J = \{j \in \{1, \ldots, m+n\} \mid \mathbf{w}_0^j = \mathbf{w}_t^j = 0\}$. For any point $\mathbf{w} \in (\mathbf{w}_0, \mathbf{w}_t)$,

$$\mathbf{w}^j = 0 \iff j \in J.$$

The coordinates of $\mathbf{w}_k$ that do not belong to J are uniquely determined by the corresponding components of the $(n+m)$-dimensional natural-valued vector representing $\mathbf{u}_k$, according to (11). But, according to (13), for every $l \notin J$,

$$|\mathbf{u}_k^l| \leq \frac{\deg_{\mathbf{w}_i}(\mathrm{ld}_{\leq_k} A_k)}{\mathbf{w}_i^l},$$

and $\mathbf{w}_i^l \neq 0$, since $\mathbf{w}_i \in (\mathbf{w}_0, \mathbf{w}_t)$. So, there are finitely many choices for the components of $\mathbf{u}_k$ which do not belong to J, and hence, there are finitely many vectors $\mathbf{w}_k$. Since for $j \neq k$, $\mathbf{w}_j \neq \mathbf{w}_k$, we obtain that K_i is finite. □

9 Open problems

The first problem to solve before the differential Gröbner walk can be implemented is how to compute efficiently the characteristic set $\bar{\mathcal{B}}$ of $\mathrm{in}_{\mathbf{w}}(I)$ w.r.t. $\leq_{new}$ having a characteristic set $\mathrm{in}_{\mathbf{w}}(\mathcal{A})$ of $\mathrm{in}_{\mathbf{w}}(I)$ w.r.t. $\leq_{old}$. This computation is possible despite $\mathrm{in}_{\mathbf{w}}(I)$ is not a differential ideal: one can compute a characteristic set $\mathcal{B}$ of I w.r.t. $\leq_{new}$, then by Lemma 3 $\mathrm{in}_{\mathbf{w}}(\mathcal{B})$ is a characteristic set of $\mathrm{in}_{\mathbf{w}}(I)$ w.r.t. $\leq_{new}$. However, this argument does not help from the practical point of view, since following this algorithm we would not need consider $\mathrm{in}_{\mathbf{w}}(I)$ at all. Nevertheless, we hope that, as in case of a polynomial ideal, there is an algorithm which computes $\bar{\mathcal{B}}$ directly. Our future research is directed towards the analysis of the Rosenfeld-Gröbner algorithm and its adjustment for $\mathrm{in}_{\mathbf{w}}(I)$.

Second, one can extend the algorithm to arbitrary (non-Riquier) rankings, since they also have a matrix characterization [18, 17]. One will need to refine the notion of weight vector and initial form.

Finally, the choice of segment $[\mathbf{w}_0, \mathbf{w}_t]$ as the Gröbner walk path is not always the best choice, as it follows from [9, 10]. One can either apply the path perturbation strategy described in these papers, or use pre-rankings [18] compatible with rankings which generalize weight vectors.

References

[1] Buchberger B. An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal (in German). PhD Thesis, Univ. of Insbruck, Inst. for Math, 1965.

[2] Becker T., Weispfenning V. with H. Kredel. *Gröbner Bases: a computational approach to Commutative Algebra,* Springer-Verlag, 1993.

[3] Robbiano L. Term orderings on the polynomial ring. *Proc. EUROCAL'85,* 513–517, Linz, Austria, 1985.

[4] Mora T., Robbiano L. The Gröbner fan of an ideal. *JSC,* 6:183–208, 1988.

[5] Schwarz F. Monomial orderings and Gröbner bases. *ACM SIGSAM Bulletin 25/1,* pp. 10–23, 1991.

[6] Carrà-Ferro G., Sit W.Y. On term-orderings and rankings. *Lecture notes in pure and applied Math., Computational Algebra, Dekker, 151,* 1992.

[7] Faugére J.C., Gianni P., Lazard D., Mora T. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *JSC,* 16:329–344, 1993.

[8] Collart S., Kalkbrener M., Mall D. The Gröbner walk. Dept. of Math., Swiss Federal Inst. of Tech, 8092 Zürich, Switzerland, 1993.

[9] Amrhein B., Gloor O., Küchlin W. How fast does the walk run. *Proc. of RWCA'96,* 1996.

[10] Amrhein B., Gloor O., Küchlin W. Walking faster. *Proc. of DISCO'96,* Karlsruhe, Germany, 1996.

[11] O'Shea D. Gröbner fan and walk. *Springer-Verlag electronic production,* 2001.

[12] Riquier C. *Les systèmes d'èquations aux dérivées partielles.* Gauthier-Villars, Paris, 1910.

[13] Ritt J. *Differential Algebra.* Dover, 1950.

[14] Kolchin E.R. *Differential Algebra and Algebraic Groups.* Academic press, 1973.

[15] Kondratieva M.V., Levin A.B., Mikhalev A.V., Pankratiev E.V. *Differential and Difference Dimension Polynomials.* Klüver publishers, 1999.

[16] Mansfield E. Differential Gröbner bases. Ph.D. Thesis, University of Sydney, 1991.

[17] Weispfenning V. Differential term-orders. *Proc. of ISAAC'93.* Kiev, ACM press, 1993.

[18] Rust C.J., Reid G.J. Rankings of partial derivatives. 1998.

[19] Boulier F., Lazard D., Ollivier F., Petitot M. Representation for the radical of a finitely generated differential ideal. *Proc. ISAAC 1995*, ACM Press, 158–166.

[20] Boulier F. Efficient computation of regular differential systems by change of rankings using Kähler differentials. Tech. rep., Université Lille I, 1999.

[21] Boulier F., Lemaire F., Maza M.M. PARDI! Publication LIFL 2001-01, 2001.

Application of Maple Package to Analysis of Fluid Dynamics and Mathematical Biology Problems

V. Govorukhin

Rostov State University,
e-mail: vgov@math.rsu.ru

This talk is devoted to application of computer algebra package Maple for analysis of planar filtrational convection problem and "active predator - prey" mathematical model studying. We use computer algebra manipulation to obtaining finite-dimensional approximation of PDE and for analysis of ODE-systems. Analytical form of approximation permits to increase precision and rate of calculations. The possibilities of computer algebra allow to carry out the stability analysis of stationary regimes and to receive critical values of bifurcation parameter only with an error of Galerkin or lines methods. Maple is using for MATLAB and Pascal code generation with following numerical investigation. These two examples can be a sample for other scientific problem solution by Maple.

The Application of the Computing Algebra in Cosmic Dynamical Problems

E.A. Grebenikov, M. Jakubiak, D. Kozak–Skoworodkin

University of Podlasie, Siedlce, Poland

In this paper we demonstrate the KAM–theory possibility to Lyapunov stability of new cosmic models by help methods of Computing Algebra.

Poincaré Problem

Consider the $2n$–dimensional Hamiltonian System

$$\frac{dp}{dt} = -\frac{\partial H}{\partial q}\,, \qquad \frac{dq}{dt} = \frac{\partial H}{\partial p}\,, \tag{1}$$

where the Hamiltonian $H(p,q)$ is of the type

$$H(p,q) \equiv H_0(p) + \mu H_1(p,q)\ , 0 \leq \mu < 1\,, \tag{2}$$

where H_1 satisfies

$$H_1(p,q) \equiv H_1(p, q + (2\pi)). \tag{3}$$

In addition we assume $H(p,q)$ to be 2π–periodical on $q_1, q_2, ..., q_n$ and analytic on $2n$–dimensional symplectic manifold

$$G_{2n} = \{p \in G_n, \|Imq\| < \rho < 1, \|Imq\| = \sum_{s=1}^{n} |\ Imq_s\ |\}\ , \tag{4}$$

where G_n denotes a n–dimensional torus manifold in euclidean space. The variables (p,q) usually are refereed to as "action – angle" coordinates.

The problem of integrability of the system (1), which means to find a nondegenerate canonical mapping $G_{2n} \to G^*_{2n}, (p,q) \to (P,Q)$, that reduces the system (1) to the following one:

$$\frac{dP}{dt} = 0\,, \qquad \frac{dQ}{dt} = \frac{\partial H^*}{\partial P}\,. \tag{5}$$

It follows from this, that in G^*_{2n} one has

$$H^*(P) \equiv H(p,q)\,. \tag{6}$$

Understanding the difficulty of this programme, Poincaré considered the following question to be solved first:

Problem. To find all the equilibrium points of the equations (1), to classify them and to investigate their orbital and Lyapunov stability.

We analyse two aspects of this problem, which can be solve by help Computing Symbol System (CSS) "Mathematica":

Aspect 1.1. The existence and linear stability problem of the equilibrium solutions of the hamiltonian systems, describing the many – body ($n > 3$) restricted problems.

Aspect 1.2. The adaptability of KAM – theory (in particularly, of Arnold – Moser theorem) for hamiltonian solution Lyapunov stability in restricted Newtonian many – body problem, in particularly.

Aspect 1.1.

One from property of symplectic phase marks, that linear hamiltonian systems in environment of whichever equilibrium point (p_0, q_0) are describe by symplectic constant matrix. The basic property of such matrix is, that all eigenvalues have form $\pm\alpha \pm \beta i$. Then the point (p_0, q_0) is stable, if all $\alpha \equiv 0$. Therefore by methods of classical stability theory the stability of nonlinear systems is unsolvable and can be studied only in frames of KAM – theory.

In restricted newtonian problem of $N(N = n + 2 > 3)$ bodies it is investigate the movement of zero – mass in symmetrical field, definited by gravitation powers of other masses $m_1 = m_2 = ... = m_n \neq 0$, creating the regular polygon, turning round centre m_0 of all masses. B. Elmabsout and E. Grebenicov showed, that similar configuration is exact particular solution of newtonian problem of many bodies with equal masses. Similarly as in classical restricted problem of three bodies, here also it can study the restricted problem of N–bodies: circular, elliptic, hyperbolic and parabolic.

The differential equations of restricted circular of N body problem in uniformly turning coordinates system P_0xyz have form:

$$\begin{aligned} \frac{d^2x}{dt^2} - 2\omega_n\frac{dy}{dt} &= -\frac{m_0 x}{r^3} + \frac{\partial R}{\partial x}, \\ \frac{d^2y}{dt^2} + 2\omega_n\frac{dy}{dt} &= -\frac{m_0 y}{r^3} + \frac{\partial R}{\partial y}, \\ \frac{d^2z}{dt^2} &= -\frac{m_0 z}{r^3} + \frac{\partial R}{\partial z}, \end{aligned} \tag{7}$$

$$R(x,y,z) = \frac{\omega_n^2}{2}(x^2+y^2) + m\sum_{k=1}^{n}\left[\frac{1}{\Delta_k} - \frac{xx_k + yy_k + zz_k}{r_k^3}\right], \tag{8}$$

$$\Delta_k^2 = (x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2 ,$$
$$r^2 = x^2 + y^2 + z^2,\ r_k^2 = x_k^2 + y_k^2 + z_k^2,\ k = 1, ..., n ,$$
$$x_k = a_0 \cos\frac{2\pi(k-1)}{n},\ y_k = a_0 \sin\frac{2\pi(k-1)}{n},\ z_k = 0,\ k = 1, .., n , \tag{9}$$
$$\omega_n = \sqrt{\frac{1}{a_0^3}\left[m_0 + \frac{m}{4}\sum_{k=2}^{n}\left(\sin\frac{\pi(k-1)}{n}\right)^{-1}\right]}$$
$$n = N - 2 ,$$

ω_n–the angle speed of coordinate system P_0xyz in relation to original inertial system, and also is the angle speed of regular polygon $P_1P_2...P_n$ in tops of which are masses $m_1 = m_2 = ... = m_n \neq 0$, round central body P_0 with mass m_0. Using classical transformations, it is possible to write the equations (7) in form (1).

Determination of equilibrium positions of system (7) brings to solutions of non–linear, functional equation system:

$$\frac{dx}{dt} = \frac{dy}{dt} = \frac{dz}{dt} = 0 ,$$
$$-\frac{m_0 x}{r^3} + \frac{\partial R}{\partial x} = -\frac{m_0 y}{r^3} + \frac{\partial R}{\partial y} = -\frac{m_0 z}{r^3} + \frac{\partial R}{\partial z} = 0 , \tag{10}$$

or

$$\omega_n^2 x - \frac{m_0 x}{r^3} + m\sum_{k=1}^{n}\left[\frac{x_k - x}{\Delta_k^3} - \frac{1}{a_0^2}\cos\frac{2\pi(k-1)}{n}\right] = 0 ,$$
$$\omega_n^2 y - \frac{m_0 y}{r^3} + m\sum_{k=1}^{n}\left[\frac{y_k - y}{\Delta_k^3} - \frac{1}{a_0^2}\sin\frac{2\pi(k-1)}{n}\right] = 0 , \tag{11}$$
$$-\frac{m_0 z}{r^3} + m\sum_{k=1}^{n}\frac{z}{\Delta_k^3} = 0 .$$

In (11) the sizes x, y, z are unknowns.

Last equation from (11) for $z = 0$ is always realized. Then all equilibrium points of system (7) are found in plane P_0xy. It can show, that for any $n \geq 2$ the system (11) is equivalent to system:

$$\omega_n^2 x - \frac{m_0 x}{r^3} + m\sum_{k=1}^{n}\frac{x_k - x}{\Delta_k^3} = 0 ,$$
$$\omega_n^2 y - \frac{m_0 y}{r^3} + m\sum_{k=1}^{n}\frac{y_k - y}{\Delta_k^3} = 0 . \tag{12}$$

In particularly, for famous restricted 3-body problem ($n = 1$) the equations (11) are form

$$\omega_1^2 x - \frac{m_0 x}{r^3} + m\left(\frac{1-x}{\Delta_1^3} - 1\right) = 0\,,$$
$$\omega_1^2 y - \frac{m_0 y}{r^3} - \frac{my}{\Delta_1^3} = 0\,. \tag{13}$$

For $y = 0$ first equation has three Euler collinear solutions. For $y \neq 0$ the system (13) has two Lagrange solutions, which determinate two equilateral triangles P_0P_1P. It is known, that the collinear solutions are unstable in first approximation for arbitrary values of m.

Research of Lagrange triangle stability has 200-year history. At first G. Gascheau, E. Routh and A. Lyapunov studied the triangle stability in first approximation. The conditions of this stability is

$$0 \leq m < m^* = \frac{9 - \sqrt{69}}{18} = 0.0385209....$$

The stability in Lyapunov sense studied H. Poincaré, A. Lyapunov, G. Birkhoff, C. Siegel, V. Arnold, A. Deprit, J. Moser, A. Leontovich, A. Markeev, A. Sokolski, V. Sebehely and ultimate results were got on base of KAM-theory.

Using CSS "Mathematica", we count the coordinates of equilibrium positions in restricted problems 4, 5, 6, 7 – of bodies.

Here, the first algebraic problem, which appears at research of equilibrium positions, is the determination of all solutions of nonlinear functional equations (12) for $n = 2, 3, 4, 5$. The next algebraic problem is the construction of linear hamiltonian system in environment of the equilibrium points and research eigenvalues of the their matrices.

We demonstrated that the radial stationary points are unstable in first approximation for all values m and the bisectorial stationary points are stable in first approximation for $0 \leq m < m^*$.

We have the board

N	$0 \leq m < m^*$
4	$0 \leq m < 0.085...$
5	$0 \leq m < 0.023...$
6	$0 \leq m < 0.0094...$
7	$0 \leq m < 0.0047...$

For all values $0 \leq m < m^*$ the eigenvalues of matrix of linear hamiltonian equations are the numbers of type $\pm\beta i$. Also one can prove, that all equilibrium positions are the homografic solutions in Lagrange – Wintner sense of differntial equations (7).

Aspect 1.2.

For research of Lyapunov stability of stationary points, stable in first approximation, one used Arnold – Moser theorem. In order to use this theorem, one should be realized the

operation of Birkhoff normalization of hamiltonians of these four problems with accuracy to fourth degree of local coordinates. All algebraic operations and calculations one realized by help CSS "Mathematica". We will show, that in intervals $0 \leq m < m^*$ (with exception so-called of "resonanse points"), the equilibrium positions of restricted problems are stable in Lyapunov sense.

If we will transfer the origin of coordinate system from point P_0 to stable in first approximation stationary point with coordinates x^*, y^* by help of expressions

$$X = x - x^*,$$
$$Y = y - y^*,$$
$$P_X = p_x - p_{x^*},$$
$$P_Y = p_y - p_{y^*},$$

and we will pass to canonical variables (X, Y, P_X, P_Y), using classical transformations, we will receive, for example, hamiltonian $H(7)$ of restricted problem of seven – bodies $(n = 5)$ in form (14):

$$\begin{aligned} H(7) = \omega_5(yp_x - xp_y) + \frac{1}{2}(p_x^2 + p_y^2) - (x^2 + y^2)^{-1/2} - m\Bigg(\Big((x-1)^2 + y^2\Big)^{-1/2} + \\ + \left(\left(\frac{1}{4}(1+\sqrt{5}) + x\right)^2 + \left(-\frac{1}{2}\sqrt{\frac{1}{2}(5-\sqrt{5})} + y\right)^2\right)^{-1/2} + \\ + \left(\left(\frac{1}{4}(1+\sqrt{5}) + x\right)^2 + \left(\frac{1}{2}\sqrt{\frac{1}{2}(5-\sqrt{5})} + y\right)^2\right)^{-1/2} + \\ + \left(\left(\frac{1}{4}(1-\sqrt{5}) + x\right)^2 + \left(-\frac{1}{2}\sqrt{\frac{1}{2}(5+\sqrt{5})} + y\right)^2\right)^{-1/2} + \\ + \left(\left(\frac{1}{4}(1-\sqrt{5}) + x\right)^2 + \left(\frac{1}{2}\sqrt{\frac{1}{2}(5+\sqrt{5})} + y\right)^2\right)^{-1/2}\Bigg). \end{aligned} \tag{14}$$

Hamiltonian differential equations have form (1). They have, of course, particular solution

$$X = Y = P_X = P_Y = 0. \tag{15}$$

Unfortunately they are not comfortable for research of Lyapunov stability, because that hamiltonian (14) is not positively definite function of phase variables (X, Y, P_X, P_Y). In enough little environment of the coordinate origin (15) of hamiltonian (14) one can represent by help of convergent power serie:

$$H = H_2(X, Y, P_X, P_Y) + H_3(X, Y) + H_4(X, Y) + ..., \tag{16}$$

where H_k–homogeneous form of k–degree.

On beginning by help non–singular canonical transformation
$(X, Y, P_X, P_Y) \to (q_1, q_2, p_1, p_2)$

$$[X, Y, P_X, P_Y]^T = A \cdot [q_1, q_2, p_1, p_2]^T, \tag{17}$$

where matrix A one shuld determine so, in order to transformed hamiltonian K $(H(X, Y, P_X, P_Y) \to K(q_1, q_2, p_1, p_2))$ had form

$$K(q_1, q_2, p_1, p_2) = K_2(q_1, q_2, p_1, p_2) + K_3(q_1, q_2, p_1, p_2) + K_4(q_1, q_2, p_1, p_2). \tag{18}$$

For equilibrium point with coordinates $x^* = 0.809317,\ y^* = 0.588003$, expressions K_2, K_3, K_4 have forms:

$$K_2 = \frac{1}{2}\sigma_1(p_1^2 + q_1^2) - \frac{1}{2}\sigma_2(p_2^2 + q_2^2) = 0.485052(p_1^2 + q_1^2) - 0.115864(p_2^2 + q_2^2), \tag{19}$$

$$\begin{aligned} K_3 = {} & -2.52816p_1^3 - 2.51376p_2^3 + 1.19723q_1^3 + 0.339633q_2^3 + 2.35999q_1^2q_2 + \\ & +1.55067q_1^2q_2^2 + p_1^2(-12.0813p_2 - 0.600941q_1 - 0.39486q_2) + p_2^2(-14.7484p_1 + \\ & +16.6356q_1 + 10.9308q_2) + p_1(8.12181p_2q_1 + 5.5534q_1^2 + 5.33659p_2q_2 + \\ & +7.29794q_1q_2 + 2.39762q_2^2) + p_2(11.6697q_1^2 + 15.3356q_1q_2 + 5.03827q_2^2), \end{aligned} \tag{20}$$

$$\begin{aligned} K_4 = {} & -0.599766p_1^4 - 77.3193p_2^4 + 1.9228q_1^4 + 0.358408q_2^4 - p_1^3(16.9111p_2 + \\ & +14.8001q_1 + 9.72468.q_2) - p_2^3(139.675p_1 + 107.938q_1 + 70.9224q_2) + \\ & -p_1^2(82.2071p_2^2 + 97.3317p_2q_1 + 16.6075q_1^2 + 63.9536p_2q_2 + 21.8245q_1q_2 + \\ & +7.1701q_2^2) + p_2^2(-188.407p_1q_1 - 20.6904q_1^2 - 123.796p_1q_2 - 27.1901q_1q_2 + \\ & -8.93287q_2^2) + p_1(-44.955p_2q_1^2 + 2.0817q_1^3 + 4.10347q_1^2q_2 - 59.077p_2q_1q_2 + \\ & +19.8283q_1^2q_2 + 13.0286q_1q_2^2 + 2.85356q_2^3) + 5.05365q_1^3q_2 + 4.9809q_1^2q_2^2 + \\ & +2.18186q_1q_2^3. \end{aligned} \tag{21}$$

Transformation (17) permits to reset in K_2 expressions $P_XY - P_YX$. From formula (19) one can see, that K_2 does not have expressions $q_1p_2, q_2p_2, q_2p_1, q_1p_1$. The solution of the system (17) with 16 unknowns is possible only by help CSS "Mathematica".

This is the first important step in research of Lyapunov stability. Canonical variables (q_1, q_2, p_1, p_2) are not the variables of type "actin – angle", because K_2 depends not only from impulses, but depends and from coordinates q_1, q_2. Farther one should be realized the Birkhoff transformation in form

$$\begin{aligned} q_1 &= \sqrt{2\tau_1}\sin\Theta_1, \\ q_2 &= \sqrt{2\tau_2}\sin\Theta_2, \\ p_1 &= \sqrt{2\tau_1}\cos\Theta_1, \\ p_2 &= \sqrt{2\tau_2}\cos\Theta_2, \end{aligned} \tag{22}$$

which in square parts of new hamiltonian F

$$K(q_1, q_2, p_1, p_2) \to F(\Theta_1, \Theta_2, \tau_1, \tau_2)$$

without coordinates Θ_1, Θ_2. If we represent hamiltonian F in form

$$F(\Theta_1, \Theta_2, \tau_1, \tau_2) = F_2(\tau_1, \tau_2) + F_3(\Theta_1, \Theta_2, \tau_1, \tau_2) + F_4(\Theta_1, \Theta_2, \tau_1, \tau_2) + ..., \tag{23}$$

then F_2, F_3, F_4 have forms :

$$F_2 = \sigma_1\tau_1 - \sigma_2\tau_2 = 0.970104\tau_1 - 0.231729\tau_2, \tag{24}$$

$$\begin{aligned} F_3 = &-1.4362\tau_1^{3/2}\cos\Theta_1 - 17.4666\tau_1^{1/2}\tau_2\cos\Theta_1 - 5.71453\tau_1^{3/2}\cos 3\Theta_1 + \\ &-1.28015\tau_1^{1/2}\tau_2\cos(\Theta_1 - 2\Theta_2) - 11.6341\tau_1\tau_2^{1/2}\cos(2\Theta_1 - \Theta_2) + \\ &-0.582121\tau_1\tau_2^{1/2}\cos\Theta_2 - 1.7699\tau_2^{3/2}\cos\Theta_2 - 5.34009\tau_1^{3/2}\cos 3\Theta_2 + \\ &-21.9549\tau_1\tau_2^{1/2}\cos(2\Theta_1 + \Theta_2) - 22.968\tau_1^{1/2}\tau_2\cos(\Theta_1 + 2\Theta_2) + \\ &+2.11478\tau_1^{3/2}\sin\Theta_1 + 25.7193\tau_1^{1/2}\tau_2\sin\Theta_1 - 1.2715\tau_1^{3/2}\sin 3\Theta_1 + \\ &+6.89313\tau_1^{1/2}\tau_2\sin(\Theta_1 - 2\Theta_2) + 7.69096\tau_1\tau_2^{1/2}\sin(2\Theta_1 - \Theta_2) + \\ &+2.77911\tau_1\tau_2^{1/2}\sin\Theta_2 + 8.44968\tau_2^{3/2}\sin\Theta_2 + 7.48905\tau_2^{3/2}\sin 3\Theta_2 + \\ &+3.79502\tau_1\tau_2^{1/2}\sin(2\Theta_1 + \Theta_2) + 14.4402\tau_1^{1/2}\tau_2\sin(\Theta_1 + 2\Theta_2), \end{aligned} \tag{25}$$

$$\begin{aligned} F_4 = &-6.3191\tau_1^2 - 105.087\tau_1\tau_2 - 119.908\tau_2^2 - 5.04513\tau_1^2\cos 2\Theta_1 + \\ &-73.6677\tau_1\tau_2\cos 2\Theta_1 + 8.9652\tau_1^2\cos 4\Theta_1 - 74.8189\tau_1^{1/2}\tau_2^{3/2}\cos(\Theta_1 - 3\Theta_2) + \\ &-54.2213\tau_1\tau_2\cos(2\Theta_1 - 2\Theta_2) - 51.1759\tau_1^{3/2}\tau_2^{1/2}\cos(\Theta_1 - \Theta_2) + \\ &-229.539\tau_1^{1/2}\tau_2^{3/2}\cos(\Theta_1 - \Theta_2) + 0.582859\tau_1^{3/2}\tau_2^{1/2}\cos(3\Theta_1 - \Theta_2) + \\ &-100.708\tau_1\tau_2\cos 2\Theta_2 - 155.355\tau_2^2\cos 2\Theta_2 - 34.014\tau_2^2\cos 4\Theta_2 + \\ &-44.5123\tau_1^{3/2}\tau_2^{1/2}\cos(\Theta_1 + \Theta_2) - 208.894\tau_1^{1/2}\tau_2^{3/2}\cos(\Theta_1 + \Theta_2) + \\ &+27.461\tau_1^{3/2}\tau_2^{1/2}\cos(3\Theta_1 + \Theta_2) + +4.85566\tau_1\tau_2\cos(2\Theta_1 + 2\Theta_2) + \\ &-45.4469\tau_1^{1/2}\tau_2^{3/2}\cos(\Theta_1 + 3\Theta_2) - 12.7184\tau_1^2\sin 2\Theta_1 + \\ &-185.71\tau_1\tau_2\sin 2\Theta_1 - 8.4409\tau_1^2\sin 4\Theta_1 + 1.71031\tau_1^{1/2}\tau_1^{3/2}\sin(\Theta_1 - 3\Theta_2) + \\ &-53.6605\tau_1\tau_2\sin(2\Theta_1 - 2\Theta_2) - 21.0421\tau_1^{3/2}\tau_2^{1/2}\sin(\Theta_1 - \Theta_2) + \\ &-0.582121\tau_1\tau_2^{1/2}\cos\Theta_2 - 1.7699\tau_2^{3/2}\cos\Theta_2 - 5.34009\tau_1^{3/2}\cos 3\Theta_2 + \\ &-94.3798\tau_1^{1/2}\tau_2^{3/2}\sin(\Theta_1 - \Theta_2) - 46.7812\tau_1^{3/2}\tau_2^{1/2}\sin(3\Theta_1 - \Theta_2) + \\ &-44.1253\tau_1\tau_2\sin 2\Theta_2 - 68.0689\tau_2^2\sin 2\Theta_2 - 36.888\tau_2^2\sin 4\Theta_2 + \\ &-46.1126\tau_1^{3/2}\tau_2^{1/2}\sin(\Theta_1 + \Theta_2) - 216.404\tau_1^{1/2}\tau_2^{3/2}\sin(\Theta_1 + \Theta_2) + \\ &-60.6094\tau_1^{3/2}\tau_2^{1/2}\sin(3\Theta_1 + \Theta_2) - 137.442\tau_1\tau_2\sin(2\Theta_1 + 2\Theta_2) + \\ &-122.676\tau_1^{1/2}\tau_2^{3/2}\sin(\Theta_1 + 3\Theta_2). \end{aligned} \tag{26}$$

In Arnold – Moser theorem hamiltonian should have the form:

$$W(\psi_1, \psi_2, T_1, T_2) = W_2(T_1, T_2) + W_4(T_1, T_2) + W_5(\psi_1, \psi_2, T_1, T_2) + ..., \tag{27}$$

where

$$W_2(T_1, T_2) = F_2(T_1, T_2), W_4(T_1, T_2) = c_{20}T_1^2 + c_{11}T_1T_2 + c_{02}T_2^2,$$

eigenvalues of linear system

$$\frac{dT_1}{dt} = -\frac{\partial W_2}{\partial \psi_1} = 0, \frac{d\psi_1}{dt} = \frac{\partial W_2}{\partial T_1},$$
$$\frac{dT_2}{dt} = -\frac{\partial W_2}{\partial \psi_2} = 0, \frac{d\psi_2}{dt} = \frac{\partial W_2}{dT_2}, \tag{28}$$

are the numbers $\pm i\sigma_1, \pm i\sigma_2$,

$$n_1\sigma_1 + n_2\sigma_2 \neq 0, \text{ for } 0 < |n_1| + |n_2| \leq 4, \tag{29}$$

$$c_{20}\sigma_2^2 + c_{11}\sigma_1\sigma_2 + c_{02}\sigma_1^2 \neq 0. \tag{30}$$

Then equilibrium point

$$T_1 = T_2 = \psi_1 = \psi_2 = 0$$

is stable in Lyapunov sense.

In order to use this theorem, one should construct yet one canonical transformation, $(\Theta_1, \Theta_2, \tau_1, \tau_2) \to (\psi_1, \psi_2, T_1, T_2)$, which would transform $F_3(\Theta_1, \Theta_2, \tau_1, \tau_2)$ in $W_3(\psi_1, \psi_2, T_1, T_2) = 0$ and $F_4(\Theta_1, \Theta_2, \tau_1, \tau_2)$ in $W_4(T_1, T_2)$.

Stays yet to construct last canonical transformation

$$\begin{aligned}
\Theta_1 &= \psi_1 + V_{13}(\psi_1, \psi_2, T_1, T_2) + V_{14}(\psi_1, \psi_2, T_1, T_2), \\
\Theta_2 &= \psi_2 + V_{23}(\psi_1, \psi_2, T_1, T_2) + V_{24}(\psi_1, \psi_2, T_1, T_2), \\
\tau_1 &= T_1 + U_{13}(\psi_1, \psi_2, T_1, T_2) + U_{14}(\psi_1, \psi_2, T_1, T_2), \\
\tau_2 &= T_2 + U_{23}(\psi_1, \psi_2, T_1, T_2) + U_{24}(\psi_1, \psi_2, T_1, T_2),
\end{aligned} \tag{31}$$

where $U_{13}, U_{23}, U_{14}, U_{24}, V_{13}, V_{23}, V_{14}, V_{24}$ one should be determinate from the linear differential partial equations.

For example, we have

$$\frac{\partial U_{13}}{\partial \psi_1}\sigma_1 - \frac{\partial U_{13}}{\partial \psi_2}\sigma_2 = A_{13}(\psi_1, \psi_2, T_1, T_2), \tag{32}$$

where A_{13} is the trigonometrical polynomial of variables ψ_1, ψ_2.

The solution of (32) is also the trigonometrical polynomial form relative to ψ_1, ψ_2.

Similar expressions we have for other functions $U_{13}, U_{23}, U_{14}, U_{24}, V_{13}, V_{23}, V_{14}, V_{24}$. Ultimate hamiltonian form for W is such, that the application of Arnold – Moser theorem are realized. From this it results that equilibrium point with coordinates $x^* = 0.809317, y^* = 0.588003$, stable in first approximation, is stable in Lyapunov sense. The similar result is correct for other bisectorial equilibrium points.

References

[1] V.I. Arnold. Teoria rownan rozniczkowych, –Warszawa: PWN, 1983, 299 p.

[2] S. Wolfram. The Mathematica – Book, –Cambridge: University Press, 1996, 1403 p.

[3] K.R. Meyer, G.R. Hall. Introduction to Hamiltonian Dynamical Systems and the N–Body Problem, –New York: Springer – Verlag, 1992, 292 p.

[4] J. Moser. On invariant curves of area preserving mappings of an Annulus, –Gottingen: Nachr. Wiss. Math -Phys, KI, I 1962.

[5] G.E.O. Giacaglia. Perturbation methods in non-linear systems, –New York: Springer – Verlag, 1972, 520 p.

[6] E. Grebenicov. Two New Dynamical Models in Celestial Machanics, –Bucharest: Rom. Astron. J., vol.8, Nr 1, 1998, pp. 13–19.

[7] D. Kozak, E. Onishk. Application of the "Mathematica" computer system to the determination of the coordinates of equilibrium positions in the Grebenicov–Elmabsout gravitation model of four bodies, –Kyiv: Nonlinear oscillations, 1999, pp.16–18.

[8] M. Jakubiak. Sufficient conditions of linear stability for equilibrium points in the Newtonian gravitational model of six bodies,–Kyiv: Nonlinear oscillations, 1999, pp.138–144.

[9] E. Grebienicov, D. Kozak, M. Jakubiak. The Algebraic Problems of the Normalization in Hamiltonian Theory, –Moscow: Proceeding of the Second International Workshop on Mathematica System in Teaching and Reserch, Siedlce'2000, 2000, pp. 73–90.

[10] D. Kozak, M. Jakubiak. The Birkhoff Hamiltonian Normalization Using the "Mathematica", –Odessa: Abstracts of "DIFIN–2000", p. 342.

Stability of Equilibrium Points in Lagrange - Wintner Models

E. A. Grebenikov, G. Olszanowski, A. Siluszyk

University of Podlasie, Siedlce, Poland;
e-mail: agnesd@imif.ap.siedlce.pl

In the papers [1, 2, 3] it is demonstrated the existence of the equilibrium solutions of differential equations for restricted Lagrange - Wintner gravitational models. In these models it is studied the motion of the passive gravitate mass in Newtonian gravitational field, what is created by rotating gravitational regular polygon, in the vertices which there are gravitational masses all equal m.

The numbers of vertices is arbitrary. Independently from of number of polygon vertices n the differential equations of the motion of passive gravitate mass in space R^3 have six order and on the plane the order of this equation system is four. It is possible to prove, that all the equilibrium points in these models there are in the plane of regular polygon.

In this paper we prove, that the equilibrium points of circular restricted Lagrange - Wintner gravitational models for $n = 7, ..., 20$ are unstable. The differential equations of the motion of the point P in the gravitational field of the bodies $P_k (k = 1, 2, .., n)$ in the barycentric Cartesian system (Gxy) have the form [4]:

$$\begin{cases} \frac{d^2x}{dt^2} - 2\omega_n \frac{dy}{dt} - \omega_n^2 x = f\, m \sum\limits_{k=1}^{n} \frac{x_k - x}{\Delta_k^3}, \\ \frac{d^2y}{dt^2} + 2\omega_n \frac{dx}{dt} - \omega_n^2 y = f\, m \sum\limits_{k=1}^{n} \frac{y_k - y}{\Delta_k^3}, \end{cases} \tag{1}$$

where

$$\omega_n^2 = \frac{m}{4} \sum_{k=1}^{n} \left(\sin \left(\frac{\pi (k-1)}{n} \right) \right)^{-1}, \tag{2}$$

$$\Delta_k^2 = (x - x_k)^2 + (y - y_k)^2, \tag{3}$$

f- gravitational constant ($f = 1$), m- mass of the body P_k.

For determining of equilibrium points of system (1), after transformations we finally have solving system of functional equations [4]:

$$\begin{cases} \Phi_1 (x, y) \equiv \omega_n^2 x + \sum\limits_{k=1}^{n} \frac{x_k - x}{\Delta_k^3} = 0, \\ \Phi_2 (x, y) \equiv \omega_n^2 y + \sum\limits_{k=1}^{n} \frac{y_k - y}{\Delta_k^3} = 0. \end{cases} \tag{4}$$

In the article [5] have been shown that for $n = 3, 4, 5, 6$ and for arbitrary mass m all the equilibrium points (its number is equal $3n$) are unstable. Our calculations show, that

for $n = 7, .., 20$ all the equilibrium points for arbitrary mass m also are unstable. Really, if to substitute in the equations (4) the expression (2) for ω_n^2 and to reduce the size of m, then we have the equations:

$$\begin{cases} \frac{m}{4}\sum\limits_{k=1}^{n}\left(\sin\left(\frac{\pi(k-1)}{n}\right)\right)^{-1} + \sum\limits_{k=1}^{n}\frac{x_k - x}{\Delta_k^3} = 0, \\ \frac{m}{4}\sum\limits_{k=1}^{n}\left(\sin\left(\frac{\pi(k-1)}{n}\right)\right)^{-1} + \sum\limits_{k=1}^{n}\frac{x_k - x}{\Delta_k^3} = 0. \end{cases} \tag{5}$$

These equations demonstrate, that the coordinates of equilibrium points do not depend from gravitational parameter m, but depend only from values n. We solved the equations (5) for $n = 7, ..., 20$ with the help of system Mathematica [6] and we found all the equilibrium points of these models. The numbers of these points are $21, 24, 27, ..., 60$.

The linearization of the system equations (1) in neighborhood of every equilibrium point gives the system of differential equation:

$$\frac{dx}{dt} = Ax, \tag{6}$$

where A -matrix of the order 4×4 has the form:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_{31} & a_{32} & 0 & 2 \\ a_{41} & a_{42} & -2 & 0 \end{bmatrix}. \tag{7}$$

Here the elements $a_{31}, a_{32}, a_{41}, a_{42}$, are constants. The calculation show, that among eigenvalues of matrix A for $n = 7, .., 20$ and for all the equilibrium points always exist eigenvalues $\lambda = \pm a$, where a-real number.

Hence, the equilibrium points of circular restricted Lagrange - Wintner gravitational models for $n = 7, 8, ..., 20$ are unstable in the first approximation and it means, that the points are unstable in Lapunov sense[7].

References

[1] Kozak D., Oniszk E., Equilibrium Points in the Restricted Four - Body Problem: Sufficient Conditions for Linear Stability, -Bucharest: Rom. Astron. J., vol. 8, No 1, 1998, p. 27 -31.

[2] Olszanowski G., Maruszewski Z., A Restricted Four - Body Problem in Lagrange - Wintner Model, "Mathematica" System in Teaching and Research, -M.:, 2000, p. 205-211.

[3] Dziak A., The existence of Jacobi integral for averaging equations of restricted, elliptic three-body problem, "Mathematica" System in Teaching and Research, -M.:, 2000, p. 191-194.

[4] Grebenicov E. A., Kozak-Skoworodkin D., Jakubiak M., The methods of computer algebra in the many body problem, -M.:, Ed. RUDN, 2001,213p.

[5] Grebenicov E. A., Olszanowski G., Siluszyk A., On stability of symmetrical configurations of many body problem,-Grodno: University VII International Mathematic Conference,28-30 May 2001,250p.

[6] Wolfram S., The Mathematica Book, - Cambridge University Press, 1996, p. 1403.

[7] Stepanov V.V., Kurs differencialnych uravnenii,-M.: Nauka, 1968.

Symbolic Computation Systems and the Many-Body Problem

E.A. Grebenikov[a], A.N. Prokopenya[b]

[a] University of Podlasie, Siedlce, Poland
[b] Brest Technical University, Belarus;
e-mail: prokopenya@belpak.brest.by

Usage of the computer algebra systems for solving the many-body problem is discussed. Stability of a new class of the exact symmetric solutions of the Newton's gravitational $(n+1)$-body problem that are represented by regular, scale-similarly to itself varying polygon, rotating with variable angular velocity about its center is analyzed.

Introduction

The many-body problem is one of the most widely studied problems of classical mechanics. Many outstanding mathematicians and mechanicians starting from Newton have been dealing with this problem but only in the case of two interacting bodies the corresponding general solution is found. Moreover, it turned out the differential equations of the many-body problem are in general not integrable. Thus, the problem formulated by Poincare - to find new classes of the exact particular solutions of the many-body problem and to investigate their stability - has retained a high degree of topicality.

Solving the first part of this problem is connected with a search of the central configurations of the n-body system. The necessary and sufficient conditions of the existence of central configurations of Newton's gravitational n-body problem may be written as a system of 3n nonlinear algebraic equations for 3n coordinates of the bodies with masses $m_1, ..., m_n$ considered as parameters [1]. Analysis of a solvability of this system is very complicated algebraic problem and it has been completely fulfilled only for two and three interacting bodies. Nevertheless, there is a hypothesis proposed by A.Wintner [2] that in the case of $n > 3$ the central configurations for the Newton's n-body problem always exist but the amount of such configurations is restricted. This hypothesis in general form is not proved till present time. And it seems that for $n > 3$ this problem may be analyzed only with using modern symbolic computation systems, for example, Mathematica or Maple that essentially increase our ability in doing both numerical and symbolic calculations.

When some exact particular solution of the many-body problem is found, the next step is to investigate its stability. But this problem turned out to be the most complicated problem of the qualitative theory of differential equations. For example, solving the stability problem of the Lagrange triangular solutions has taken about 200 years, whereas stability of the homographic and homothetic solutions in the three-body problem still

remains unsolved. Analysis of this problem requires using both the methods of classical theory of stability and the KAM-theory. In the second case it is also necessary to construct the Birkhoff transformation for the hamiltonian normalization. Realization both of these approaches is connected with doing a lot of tremendous analytical calculations that can not be done by hands. So the further progress in solving the many-body problem and generally in the theory of dynamical systems is associated with using modern computer algebra systems.

Linearization of the Equations of Motion

In [3] B. Elmabsout found a new exact particular solution of the plane Newton's many-body problem where the n bodies $P_1, P_2, ...,\ P_n$ of equal masses m are at the vertices of a regular polygon with n sides that rotates uniformly around the body P_0 of a non-zero mass m_0 being at the center of polygon. Then it was shown [4, 5], that there exist a whole class of the exact symmetric solutions of the problem of $(n+1)$ bodies which are represented by regular, scale-similarly to self varying polygons, rotating with variable angular velocity around their centers. Stability of the invariant polygon was completely investigated in linear approximation in [6]. Here we study the stability problem in linear approximation in general case of the varying regular polygon with n sides. The corresponding equations of motion of the bodies $P_1, P_2, ...,\ P_n$ of equal masses m in relative Cartesian coordinates can be written as [4] |

$$\frac{d^2\overline{r}_j}{dt^2} + f\,(m_0+m)\,\frac{\overline{r}_j}{r^3{}_j} = f\,m\sum_{i=1\ (i\neq j)}^{n}\left(\frac{\overline{r}_i-\overline{r}_j}{|\overline{r}_i-\overline{r}_j|^3}-\frac{\overline{r}_i}{r_i{}^3}\right)\quad (j=\overline{1,n}), \tag{1}$$

where m_0 and f are a mass of the body P_0 and gravitational constant respectively. It is supposed that the body P_0 is in the origin of Cartesian frame $P_0\,xyz$.

Equations (1) can be written in cylindrical coordinates as

$$\frac{d^2\rho_j}{dt^2} - \rho_j\left(\frac{d\phi_j}{dt}\right)^2 + f(m_0+m)\frac{\rho_j}{r_j{}^3} =$$

$$= f\,m\sum_{i=1\ (i\neq j)}^{n}\left(\frac{\cos(\phi_i-\phi_j)\rho_i-\rho_j}{r_{i,j}{}^3}-\frac{\cos(\phi_i-\phi_j)\rho_i}{r_i{}^3}\right),$$

$$\rho_j\,\frac{d^2\phi_j}{dt^2} + 2\frac{d\rho_j}{dt}\,\frac{d\phi_j}{dt} = f\,m\sum_{i=1\ (i\neq j)}^{n}\left(\frac{1}{r_{i,j}{}^3}-\frac{1}{r_i{}^3}\right)\sin(\phi_i-\phi_j)\,\rho_i, \tag{2}$$

$$\frac{d^2z_j}{dt^2} + f(m_0+m)\frac{z_j}{r_j^3} = f\,m\sum_{i=1\ (i\neq j)}^{n}\left(\frac{z_i-z_j}{r_{i,j}{}^3}-\frac{z_i}{r_i{}^3}\right)\ (j=\overline{1,n})$$

where

$$r_i^2=\rho_i^2+z_i^2,\ r_{i,j}{}^2=|\overline{r}_i-\overline{r}_j|^2=\rho_i^2+\rho_j^2-2\rho_i\rho_j\,\cos(\phi_i-\phi_j)+(z_i-z_j)^2.$$

It is easy to check that equations (2) have a solution

$$\rho_j(t) = \rho_1(t) \equiv \rho(t),\ \phi_j(t) = \phi(t) + \frac{2\pi}{n}\, j,\ z_j(t) = 0\ \ (j = \overline{1,n}), \tag{3}$$

where the functions $\rho(t)$ and $\phi(t)$ are connected with two relationships

$$\rho(\phi) = \frac{p}{1 + e\ \cos\ \phi},\ \ \rho^2\ \frac{d\phi}{dt} = c \equiv const.$$

Parameter p is connected with the constant c according to the relationship

$$p = \frac{c^2}{f\ M_n},\ M_n = m_0 + \frac{m}{4} \sum_{i=1}^{n-1} \frac{1}{\sin\ (\pi\ i/n)}.$$

Solution (3) determines the plane motion of the bodies on a similar trajectories being in the $P_0\ xy$ plane. The shape of the trajectories is completely determined with two parameters p and e. And in the case of $e = 0$ all bodies move on the circle of radius p with angular velocity

$$\omega = \sqrt{\frac{f\ M_n}{p^3}}.$$

To investigate stability of solution (3) it is convenient to transform equations (2) into Nechvil's phase space according to the rule

$$\rho_j(t) \to \frac{p}{1 + e\ \cos\ \phi}\ \rho_j(\phi),\ \ z_j(t) \to \frac{p}{1 + e\ \cos\ \phi}\ z_j(\phi),$$

where p and e are some constants, and the polar angle ϕ is considered as a new independent variable. Then we obtain the equations of motion in the form

$$\frac{d^2\rho_j}{d\phi^2} - \rho_j\Big(\frac{d\phi_j}{d\phi}\Big)^2 + \frac{e\ \cos\phi}{1 + e\ \cos\phi}\ \rho_j + \frac{m_0 + m}{M_n(1 + e\ \cos\phi)}\ \frac{\rho_j}{r_j{}^3} =$$

$$= \frac{m}{M_n(1 + e\ \cos\phi)} \sum_{i=1\ (i\neq j)}^{n} \Big(\frac{\cos(\phi_i - \phi_j)\rho_i - \rho_j}{r_{i,j}{}^3} - \frac{\cos(\phi_i - \phi_j)\rho_i}{r_i{}^3}\Big),$$

$$\rho_j\ \frac{d^2\phi_j}{d\phi^2} + 2\frac{d\rho_j}{d\phi}\ \frac{d\phi_j}{d\phi} = \frac{m}{M_n(1 + e\ \cos\phi)} \sum_{i=1\ (i\neq j)}^{n} \Big(\frac{1}{r_{i,j}{}^3} - \frac{1}{r_i{}^3}\Big)\ \sin(\phi_i - \phi_j)\ \rho_i,$$

$$\frac{d^2z_j}{d\phi^2} + \frac{e\ \cos\phi}{1 + e\ \cos\phi}\ z_j + \frac{m_0 + m}{M_n(1 + e\ \cos\phi)}\ \frac{z_j}{r_j^3} =$$

$$= \frac{m}{M_n(1 + e\ \cos\phi)} \sum_{i=1\ (i\neq j)}^{n} \Big(\frac{z_i - z_j}{r_{i,j}{}^3} - \frac{z_i}{r_i{}^3}\Big)\ (j = \overline{1,n}), \tag{4}$$

Solution (3) can be written in Nechvil's variables as

$$\rho_j(\phi) = \rho \equiv 1,\ \phi_j(\phi) = \phi + \frac{2\pi}{n}\, j,\ z_j(\phi) = 0\ \ (j = \overline{1,n}). \tag{5}$$

In order to study the equations (4) at the neighborhood of solution (5) we put

$$\rho_j(\phi) \to 1 + u_j(\phi), \quad \phi_j(\phi) \to \phi + \frac{2\pi}{n} j + \gamma_j(\phi).$$

Considering functions $u_j(\phi)$, $\gamma_j(\phi)$, $z_j(\phi)$ as small perturbations of solution (5) we expand equations (4) in powers of u_j, γ_j, z_j neglecting all terms of order superior or equal to 2. As a result we obtain a linearized system of differential equations in the form

$$\frac{d^2u_j}{d\phi^2} - 2\frac{d\gamma_j}{d\phi} = \frac{8 + 12\mu + S_1}{(1 + e\cos\phi)(4\mu + S_1)} u_j -$$

$$-\frac{1}{4(1 + e\cos\phi)(4\mu + S_1)} \sum_{i=1\ (i\neq j)}^{n} \frac{1}{|\sin(\pi(i-j)/n)|^3} ((-1 + 3\cos(\frac{2\pi}{n}(i-j)))u_j +$$

$$+(-3 + \cos(\frac{2\pi}{n}(i-j)) - 32\cos(\frac{2\pi}{n}(i-j))|\sin(\frac{\pi}{n}(i-j))|^3)u_i -$$

$$-(1 + 16|\sin(\frac{\pi}{n}(i-j))|^3)\sin(\frac{2\pi}{n}(i-j))(\gamma_i - \gamma_j)),$$

$$\frac{d^2\gamma_j}{d\phi^2} + 2\frac{du_j}{d\phi} = -\frac{1}{4(1 + e\cos\phi)(4\mu + S_1)} \sum_{i=1\ (i\neq j)}^{n} \frac{1}{|\sin(\pi(i-j)/n)|^3} ((1 -$$

$$-32|\sin(\frac{\pi}{n}(i-j))|^3)\sin\frac{2\pi}{n}(i-j))u_i + 3\sin\frac{2\pi}{n}(i-j))u_j +$$

$$+(3 + \cos(\frac{2\pi}{n}(i-j)) + 16\cos(\frac{2\pi}{n}(i-j))|\sin(\frac{\pi}{n}(i-j))|^3)(\gamma_i - \gamma_j)),$$

$$\frac{d^2z_j}{d\phi^2} + \frac{4(1+\mu) + (4\mu + S_1)e\cos\phi}{(1 + e\cos\phi)(4\mu + S_1)} z_j = \tag{6}$$

$$= \frac{1}{2(1 + e\cos\phi)(4\mu + S_1)} \sum_{i=1(i\neq j)}^{n} \frac{1}{|\sin(\pi(i-j)/n)|^3} ((1 - 8|\sin(\frac{2\pi}{n}(i-j))|^3)\ z_i - z_j),$$

where

$$\mu = \frac{m_0}{m}, \quad S_1 = \sum_{j=1}^{n-1} \frac{1}{\sin(\pi j/n)}.$$

Now the problem of linear stability of solution (5) is reduced to studying stability of a trivial solution of the system (6) that is a linear system of differential equations with periodic coefficients.

Studying Linear Stability of the Solutions for the Perpendicular Perturbations

According to system (6) the functions z_j determining perturbations of the bodies trajectories in perpendicular to the plane P_0xy direction do not depend on u_j and γ_j . So in linear approximation we can study stability of solution (5) in respect to the perpendicular perturbations z_j and the perturbations u_j and γ_j contained in the plane of orbits P_0xy separately. Let us start with studying linear stability of solutions (5) for perpendicular perturbations. It is convenient to rewrite the last equation of system (6) determining the functions z_j in vector notations. So we introduce a n-dimensional vector Z with components $z_j(\phi)$ and n x n matrixes B and K that have the next elements:

$$B_{jk} = \frac{1}{|\sin(\pi(j-k)/n|^3} \quad for \quad j \neq k \quad and \quad B_{jj} = -\sum_{k=1(\neq j)}^{n} B_{jk};$$

$$K_{jk} = 1 \quad (j,k = \overline{1,n}).$$

Then we can rewrite the last equation of system (6) as

$$\frac{d^2Z}{d\phi^2} = \frac{1}{2(1+e\ \cos\phi)(4\mu+S_1)}\ (B - 8K - 8\mu I - 2(4\mu+S_1)e\ \cos\phi\ I)Z, \qquad (7)$$

where I is an nxn identity matrix. It is easy to verify [6] that the matrix B has eigenvectors E_r with components

$$E_{k,r} = \frac{1}{\sqrt{n}}\ e^{\frac{2\pi\ r}{n}\ i\ k} \quad (k,r = \overline{1,n}) \qquad (8)$$

where $i = \sqrt{-1}$ and the first index k denotes the number of component. The corresponding eigenvalues are $(-\lambda_r)$ where

$$\lambda_r = 2\sum_{k=1}^{n-1} \frac{\sin^2(\pi rk/n)}{\sin^3(\pi k/n)}.$$

Vectors (8) are also the eigenvectors of the matrix K that has two eigenvalues: 0 and n. Actually,

$$K\ E_r = 0 \quad for \quad r = \overline{1,n-1} \quad and \quad K\ E_r = n\ E_n.$$

So introducing the matrix Q with elements $Q_{k,r} = E_{k,r}$ we can easily diagonalize the matrixes B and K according to the transformation rules

$$B \rightarrow\ Q^+BQ,\ \ K \rightarrow\ Q^+KQ,$$

and reduce equation (7) to the normal form

$$\frac{d^2z}{d\phi^2} = -\frac{a+e\ \cos\phi}{1+e\ \cos\phi}\ z, \qquad (9)$$

where

$$a = \frac{\lambda_r + 8\mu + 8n\delta_{r,n}}{2(4\mu+S_1)}.$$

Equation (9) is the second order differential equation that is sometimes called a Hill's equation. In the case of $e = 0$ it reduces to the equation of simple harmonic oscillator of frequency $\sqrt{a}$. It's apparent that $a > 0$ for any $r = \overline{1,n}$ and so in this case the solution (5) is linearly stable for any perpendicular to the plane of the bodies trajectories perturbations.

Equation (9) is apparently equivalent to a linear system of two first order differential equations with periodic matrix. General analysis of such systems shows that depending on the values of parameters a and e its trivial solution may be stable or unstable [7]. The boundaries between the domains of stability and instability in the ae-plane are some curves $a = a(e)$ which characterized by the presence of periodic solutions with period 2π and 4π . Hence, we can attempt to determine these boundaries directly by seeking a solution of equation (9) in the form

$$z = c_0 + \sum_{k=1}^{\infty} (c_k \cos(\frac{k}{2}\phi) + d_k \sin(\frac{k}{2}\phi)). \qquad (10).$$

Although this is a Fourier series for the function $z = z(\phi)$ of period 4π, it can also be used to obtain the solution with period 2π by setting to zero the Fourier coefficients corresponding to k being an odd integer. On substituting (10) into equation (9) and setting coefficients of $\cos(\frac{k}{2}\phi)$ and $\sin(\frac{k}{2}\phi)$ to zero we have obtained the following infinite sequence of equations determining coefficients of the Fourier series (10).

$$a\, c_0 = 0,$$

$$ec_0 + (a-1)c_2 - \frac{3}{2}\, ec_4 = 0,$$

$$\cdots\cdots\cdots$$

$$-\frac{k(k-2)}{2}\, ec_{2k-2} + (a-k^2)c_{2k} - \frac{k(k+2)}{2}\, ec_{2k+2} = 0. \qquad (11)$$

$$(a - \frac{1}{4} + \frac{3}{8}\, e)c_1 - \frac{5}{8}\, ec_3 = 0,$$

$$\cdots\cdots\cdots$$

$$-\frac{(2k-5)(2k-1)}{8}\, ec_{2k-3} + (a-(k-\frac{1}{2})^2)c_{2k-1} - \frac{(2k-1)(2k+3)}{8}\, ec_{2k+1} = 0. \qquad (12)$$

$$(a-1)\, d_2 - \frac{3}{2}\, ed_4 = 0,$$

$$\cdots\cdots\cdots$$

$$-\frac{k(k-2)}{2}\, ed_{2k-2} + (a-k^2)d_{2k} - \frac{k(k+2)}{2}\, ed_{2k+2} = 0. \qquad (13)$$

$$(a - \frac{1}{4} - \frac{3}{8}\, e)\, d_1 - \frac{5}{8}\, e\, d_3 = 0,$$

$$\cdots\cdots\cdots \qquad (14)$$

$$-\frac{(2k-5)(2k-1)}{8}\, e\, d_{2k-3} + (a-(k-\frac{1}{2})^2)\, d_{2k-1} - \frac{(2k-1)(2k+3)}{8}\, e\, d_{2k+1} = 0.$$

It can be seen that in fact there are four infinite subsequences of linear homogeneous equations. Two of these (11) and (13) are for the coefficients $c_0, c_2, \ldots, c_{2k}$ and $d_2, \ldots, d_{2k}$ respectively and represent solution (10) with period 2π. For a solution to exist, the corresponding determinants of infinite systems (11), (13) must vanish, thus determining the stability boundaries in the ae-plane. These boundaries obviously reduce to $a = (2k)^2$ $(k = 0, 1, 2, ...)$ when $e \to 0$. The remaining two subsequences of equations (12) and (14) are for $c_1, c_3, \ldots, c_{2k+1}$ and $d_1, \ldots, d_{2k+1}$, and correspond to those stability boundaries which reduce to $a = \frac{(2k-1)^2}{4}$ $(k = 1, 2, \ldots)$ when $e \to 0$.

Of course, it's impossible to calculate a determinant of the infinite matrix. So to find the stability boundaries $a = a(e)$ we should truncate the infinite subsequences of equations (11)-(14) after the k-th term, where k is a suitably large number. The corresponding determinant of the system (11), for example, can be written as

$$D_k = \begin{vmatrix} a & 0 & 0 & 0 & \ldots & 0 \\ c & a-1 & -\frac{3}{2}e & 0 & \ldots & 0 \\ 0 & 0 & a-4 & -4e & \ldots & 0 \\ 0 & 0 & -\frac{3}{2}e & a-9 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 0 & -\frac{k(k-2)}{2}e & a-k^2 \end{vmatrix}. \tag{15}$$

Setting determinant D_k to zero we obtain an algebraic equation giving us an approximation for the stability boundary $a_k = a_k(e)$. An exact expression for the boundary is obtained when $k \to \infty$. Determinant (15) is best evaluated from the following recurrence relation

$$D_k = (a-k^2)D_{k-1} - \frac{e^2}{2}(k-2)(k-1)k(k+1)D_{k-2} \quad (k = 3, 4, \ldots) \tag{16}$$

which is readily established from (15). To start the iterative process we observe that

$$D_1 = a, \quad D_2 = a(a-1).$$

A similar procedure can be followed for the other systems (12)-(14). For instance, the determinant of system (13) is just the same as (15) with the first row and column deleted. The recurrence relation is again (16) for $k \geq 3$, but the starting values are now given by

$$D_1 = a - 1, \quad D_2 = (a-1)(a-4).$$

The corresponding recurrence relation for the determinants of systems (12), (14) is

$$D_k = (a - \frac{(2k-1)^2}{4})\, D_{k-1} - \frac{e^2}{64}(2k-5)(2k-3)(2k-1)(2k+1)\, D_{k-2} \tag{17}$$

with the starting values

$$D_1 = a - \frac{1}{4} \pm \frac{3e}{8}, \quad D_2 = (a - \frac{1}{4} \pm \frac{3e}{8})(a - \frac{9}{4}) + \frac{15e^2}{64}.$$

It's apparent from (16), (17) that in the case of $e = 0$ determinants of systems (11)-(14) will equal to zero when $a = \frac{1}{4}k^2$ $(k = 0, 1, 2, ..)$. It means the stability boundaries should cross the $e = 0$ axis in the ae-plane at these points. For sufficiently small e we can represent the stability boundaries $a = a(e)$ in the vicinity of the points $a = \frac{1}{4}k^2$ as a power series in e. As a result we obtain the curves in the form

$$a = 0,\ a = \frac{1}{4} \mp \frac{3}{8}\, e + \frac{15}{128}\, e^2 \mp \frac{45}{2048}\, e^3 + \frac{885}{32768}\, e^4,\ a = 1,$$

$$a = \frac{9}{4} - \frac{135}{256}\, e^2 \mp \frac{45}{2048}\, e^3 - \frac{34695}{262144}\, e^4,\ \ a = 4 - \frac{6}{5}\, e^2 - \frac{39}{125}\, e^4,\ \ldots \qquad (18)$$

where the error term is $O(e^5)$.

It should be emphasized that zones of instability are bounded only by the curves crossing the $e = 0$ axis in the points $a = \frac{(2k-1)^2}{4}$ $(k = 1, 2 \ldots)$ and these zones are quite narrow. Besides, there are curves of instability crossing the $e = 0$ axis in the points $a = k^2$ $(k = 0, 1, 2, \ldots)$. Thus, if parameters a and e are in a zone of instability then the trivial solution of equation (9) is unstable.

Conclusion

In present paper we have analyzed the linear stability of the exact particular solutions of the plane Newton's many-body problem found in [4,5]. It was shown that stability of the solutions in respect to the perpendicular perturbations and the perturbations contained in the plane of bodies orbits may be studied separately. Linear stability of the solutions for perpendicular perturbations depends on the ellipticity of the orbits e, the number on bodies n and parameter $\mu = m_0/m$ because parameter a is determined with n and μ according to (9). And if for a fixed value of e parameter a turns out to be in a zone of instability then the corresponding solution (5) is unstable. All calculations are done with computer algebra system *Mathematica*.

References

1. E.A.Grebenikov. About necessary and sufficient conditions of the existence of homographic solutions of Newton's many-body problem in noninertial frame of reference. *Nonlinear Analysis and Homographic Dynamics*, Iss. 1. Moscow, Computer Center RAS (1999) pp. 5-19.

2. A.Wintner. The Analytical Foundations of Celestial Mechanics. Princeton: Princeton Univ. Press, 1941.

3. B.Elmabsout. Sur l'existence de certaines configurations d'equilibre relatif dans le probleme des N corps. *Celestial Mechanics*, 41 (1988), 131-151.

4. E.A.Grebenikov. The existence of the exact symmetric solutions in the plain Newton's problem of many bodies, *Mathematical modeling*, 10, No 8 (1998) pp. 74-80.

5. E.A.Grebenikov. New exact solutions in the planar symmetrical (n+1)-body problem. *Romanian Astronomical Journal*, 7 (1997), No. 2, 151-156.

6. B.Elmabsout. Stability of some degenerate positions of relative equilibrium in the n-body problem. *Dynamics and Stability of Systems*, 9 (1994), No. 4, 305-319.

7. R.Grimshaw. Nonlinear Ordinary Differential Equations, CRC Press, 2000, 328 pp.

In memoriam William Schelter

TEXmacs Interfaces to Maxima, MuPAD and REDUCE

A. G. Grozin

Budker Institute of Nuclear Physics,
630090 Novosibirsk, Russia;
e-mail: A.G.Grozin@inp.nsk.su

GNU TEXmacs is a free wysiwyg word processor providing an excellent typesetting quality of texts and formulae. It can also be used as an interface to Computer Algebra Systems (CASs). In the present work, interfaces to three general-purpose CASs have been implemented.

1 TEXmacs

GNU TEXmacs [1] is a free (GPL) word processor which

- typesets texts and mathematical formulae with very high quality (like LATEX),
- emphasizes the logical structure of a document rather than its appearance (like LATEX),
- is easy to use and intuitive (like typical wysiwyg word processors),
- can be extended by a powerful programming language (like Emacs),
- can include PostScript figures (as well as other figures which can be converted to PostScript),
- can export LATEX, and import LATEX and `html`,
- supports a number of languages based on Latin and Cyrillic alphabets.

It uses TEX fonts both on screen and when printing documents. Therefore, it is truly wysiwyg, with equally good quality of on-screen and printed documents (in contrast to LyX which uses X fonts on screen and calls LATEX for printing). There is a similar commercial program called Scientific Workplace (for Windows).

TEXmacs can also be used as an interface to any CAS which can generate LATEX output. It renders LATEX formulae on the fly, producing CAS output with highest typesetting quality (better than, e.g., *Mathematica*, which uses fixed-width fonts for formula output). A user can utilize editing facilities of TEXmacs: copy (a part of) a previous input into the new one, edit it and sent to the CAS, copy a result derived using the CAS into a paper,

etc. In the present talk, I give some examples of using *Maxima*, *MuPAD* and *REDUCE* via TEXmacs. It is not my aim to describe these powerful and complex CASs; I only show examples of typesetting produced by TEXmacs. This talk has been written in TEXmacs and exported to LATEX.

2 Maxima

Macsyma is one of the oldest and most mature CASs. It was developed at MIT during the end of sixties – beginning of seventies. Later, it was owned by various commercial companies. Now it seems practically dead.

Fortunately, a free CAS *Maxima* is now under GPL. It is based upon the *Macsyma* code base from seventies, with a number of later enhancements. It incorporates a lot of mathematical knowledge, is stable and well tested. From its very beginning, *Macsyma* (and *Maxima*) pays much attention to mathematical correctness; for example, if the form of an integral depends on the sign of a parameter, it will ask the user about it, or use an assumption – other systems only recently incorporated similar facilities. It is an excellent platform for research projects, because it provides a solid foundation, and it cannot vanish into thin air as commercial systems (e.g., *Macsyma*) can do at any moment. Its text-based interface now looks somewhat old-fashioned. When combined with a nice graphical interface provided by TEXmacs, it can compete with commercial CASs like *Mathematica* and *Maple*, and even produce higher-quality output. And it is difficult for commercial vendors to beat the price :–)

Here is a sample *Maxima* session within TEXmacs.

```
GCL (GNU Common Lisp) Version(2.4.0) Tue May 15 15:03:11 NOVST
2001 Licensed under GNU Library General Public License Contains
Enhancements by W. Schelter Maxima 5.6 Tue May 15 15:03:08 NOVST
2001 (with enhancements by W. Schelter). Licensed under the GNU
Public License (see file COPYING)

(C1) (x^2-y^2)/(x^2+y^2)+sin(alpha)^2;
```

(D1) $\frac{x^2-y^2}{y^2+x^2}+\sin^2\alpha$

```
(C2) expand((x+y-1)^5);
```

(D2) $y^5+5xy^4-5y^4+10x^2y^3-20xy^3+10y^3+10x^3y^2-30x^2y^2+30xy^2-10y^2$
$+5x^4y-20x^3y+30x^2y-20xy+5y+x^5-5x^4+10x^3-10x^2+5x-1$

```
(C3) solve(a*x^2+b*x+c,x);
```

(D3) $\left[x=-\frac{\sqrt{b^2-4ac}+b}{2a},x=-\frac{b-\sqrt{b^2-4ac}}{2a}\right]$

```
(C4) integrate(sqrt(x^2+a),x);
Is a positive or negative?negative;
```

(D4) $\frac{a\log\left(2\sqrt{x^2+a}+2x\right)}{2}+\frac{x\sqrt{x^2+a}}{2}$

```
(C5) assume(a>0);
```

(D5) $[a>0]$

```
(C6) integrate(sqrt(x^2+a),x);
```

$$\text{(D6)}\quad \frac{a\,\text{ASINH}\left(\frac{x}{\sqrt{a}}\right)}{2}+\frac{x\sqrt{x^2+a}}{2}$$

```
(C7) integrate(exp(sin(x)),x,0,%pi);
```

$$\text{(D7)}\quad \int_0^{\pi} e^{\sin x}\,dx$$

```
(C8) diff(f(x),x,2);
```

$$\text{(D8)}\quad \frac{d^2}{dx^2}f(x)$$

```
(C9) g:gamma(1+x);
(D9) Γ(x+1)
(C10)  taylor(g,x,0,3);
```

$$\text{(D10)}\quad 1-\gamma x+\frac{\left(6\gamma^2+\pi^2\right)x^2}{12}-\frac{\left(2\gamma^3+\pi^2\gamma+4\zeta\left(3\right)\right)x^3}{12}+\cdots$$

```
(C11) m:entermatrix(2,2);
Is the matrix 1. Diagonal 2. Symmetric 3. Antisymmetric
4. General
Answer 1, 2, 3 or 4 :  4;
Row 1 Column 1:  a;
Row 1 Column 2:  b;
Row 2 Column 1:  c;
Row 2 Column 2:  d;
Matrix entered.
```

$$\text{(D11)}\quad \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

```
(C12) m^^(-1);
```

$$\text{(D12)}\quad \begin{pmatrix} \frac{d}{ad-bc} & -\frac{b}{ad-bc} \\ -\frac{c}{ad-bc} & \frac{a}{ad-bc} \end{pmatrix}$$

```
(C13) fac(n):=if n=0 then 1 else n*fac(n-1);
```

$$\text{(D13)}\quad \text{fac}(n):=\text{if } n=0 \text{ then } 1 \text{ else } n\,\text{fac}(n-1)$$

```
(C14) trace(fac);
(D14)  [fac]
(C15) fac(5);
1 Enter fac  [5]
 2 Enter fac  [4]
  3 Enter fac  [3]
   4 Enter fac  [2]
    5 Enter fac  [1]
     6 Enter fac  [0]
     6 Exit fac 1
    5 Exit fac 1
   4 Exit fac 2
  3 Exit fac 6
 2 Exit fac 24
1 Exit fac 120
(D15) 120
```

```
(C16) f(x):=block([a,z:0],a:x+1,a:a/z,a+1);
```
(D16) $f(x) := \textbf{block}\left([a, z : 0], a : x + 1, a : \frac{a}{z}, a + 1\right)$
```
(C17) debugmode(true);
(D17) true
(C18) f(u);
Division by 0
-- an error.  Entering the Maxima Debugger dbm f(x=u)
(dbm:1) a;
```
$u + 1$
```
(dbm:1) z;
0
(dbm:1) :q
(C19) plot2d(sin(x)/x,[x,-10,10]);
(D19) 0
(C20) f(x,y):=sin(sqrt(x^2+y^2))/sqrt(x^2+y^2);
```
(D20) $f(x, y) := \frac{\sin\sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2}}$
```
(C21) plot3d(f(x,y),[x,-10,10],[y,-10,10]);
(D21) 0
```

Plots appear in separate windows (Fig. 1). When the mouse is over such a window, its coordinates are continuously displayed at the upper left corner (in the 3d case, also z of the surface at the mouse position (x, y) is shown). When the mouse is at the upper left corner, a menu appears. It allows the user to control the plot: zoom, rotate (in the 3d case), print, save as PostScript, etc.

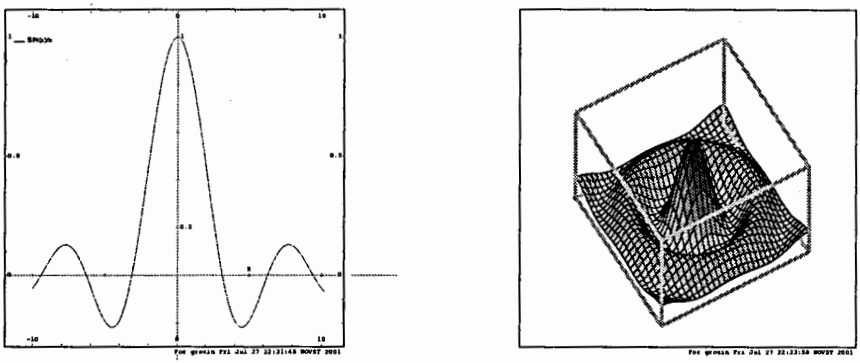


Figure 1: *Maxima* plots

The toolbar icon showing the question mark shows the *Maxima* documentation. The *Maxima* manual is in html; it is imported into TEXmacs and shown in a new buffer.

Hyperlinks work with double click. It is easy to arrange things in such a way that you do calculations in one TEXmacs window, and read the manual in another window.

3 MuPAD

MuPAD [3] is the most recent addition to the family of universal CASs. It is being developed at University of Paderborn, and commercially distributed by SciFace. In some cases, it can be obtained free of charge (see their web site), but it is not free software. It is rather similar to *Maple*, but designed from scratch, and some new fundamental ideas were incorporated. The library is not so extensive as those of older systems, but *MuPAD* is progressing fast. Its interface is text-based (except the Windows version). Therefore, adding a high-quality graphical formula output provided by TEXmacs is useful.

Here is a sample *MuPAD* session within TEXmacs.

```
  *-----*    MuPAD 2.0.0 -- The Open Computer Algebra System
 /|   /|
*-----* |    Copyright (c)  1997 - 2000  by SciFace Software
| *--|-*                   All rights reserved.
|/   |/
*-----*      Licensed to:   Andrey Grozin
```

```
≫ (x^2-y^2)/(x^2+y^2)+sin(alpha)^2
```

$$\sin(\alpha)^2 + \frac{(x^2-y^2)}{(x^2+y^2)}$$

```
≫ expand((x+y-1)^5)
```

$$5x+5y-20xy-10x^2+10x^3-10y^2-5x^4+10y^3+x^5-5y^4+y^5+30xy^2$$
$$+30x^2y-20xy^3-20x^3y+5xy^4+5x^4y-30x^2y^2+10x^2y^3+10x^3y^2-1$$

```
≫ solve(a*x^2+b*x+c=0,x)
```

$$\begin{cases} \mathbb{C} & \text{if } a=0 \wedge b=0 \wedge c=0 \\ \{\} & \text{if } a=0 \wedge b=0 \wedge c\neq 0 \\ \left\{-\frac{c}{b}\right\} & \text{if } a=0 \wedge b\neq 0 \\ \left\{\frac{-\frac{b}{2}-\frac{\sqrt{b^2-4ac}}{2}}{a}, \frac{\frac{\sqrt{b^2-4ac}}{2}-\frac{b}{2}}{a}\right\} & \text{if } a\neq 0 \end{cases}$$

```
≫ int(sqrt(x^2+a),x)
```

$$\frac{x\sqrt{a+x^2}}{2} + \frac{a\ln\left(x+\sqrt{a+x^2}\right)}{2}$$

```
≫ i1:=int(exp(sin(x)),x=0..PI); float(i1)
```

$$\int_0^{\pi} \exp(\sin(x))\,dx$$

6.208758036

```
≫ diff(f(x),x,x)
```

$$\frac{\partial^2}{\partial x^2} f(x)$$

```
≫ g:=gamma(1+x)
```

$$\gamma(x+1)$$

```
≫ series(g,x=0,4)
```

$$1 - x\gamma + x^2\left(\frac{\pi^2}{12} + \frac{\gamma^2}{2}\right) + x^3\left(-\frac{\zeta(3)}{3} - \frac{\gamma^3}{6} - \frac{\pi^2\gamma}{12}\right) + O\left(x^4\right)$$

```
≫ M:=matrix([[a,b],[c,d]]); 1/M
```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$\begin{pmatrix} -\frac{d}{bc-ad} & \frac{b}{bc-ad} \\ \frac{c}{bc-ad} & -\frac{a}{bc-ad} \end{pmatrix}$$

```
≫ plotfunc2d(sin(x)/x,x=-10..10)
Warning:  Dumb terminal:  Plot data saved in binary file save.mp
[plot]; during evaluation of 'plot2d'
≫ f:=(x,y)->sin(sqrt(x^2+y^2))/sqrt(x^2+y^2)
(x, y) ->sin(sqrt(x^2+ y^2))/sqrt(x^2+ y^2)
≫ plotfunc3d(f(x,y),x=-10..10,y=-10..10)
Warning:  Dumb terminal:  Plot data saved in binary file save.mp
[plot]; during evaluation of 'plot3d'
≫ quit
The end
```

Plots appear in separate windows (Fig. 2). Spurious warnings about dumb terminal may be ignored. Plots are displayed by the program `vcam` which is distributed with *MuPAD*. They can be controlled (and saved to PostScript files) via menus.

Figure 2: *MuPAD* plots

The question mark icon opens the help menu. *MuPAD* documentation is extensive and high-quality. It is written in an extension of LaTeX with hyperlinks, and displayed by an extended `dvi` viewer distributed with *MuPAD*. Therefore, it has high typesetting quality.

4 REDUCE

REDUCE is one of the older CASs (it was somewhat influenced by *Macsyma*). It is a commercial system. It is stable and efficient, and can solve larger problems in a given memory than, say, *Mathematica*. Its text-based interface looks old-fashioned; it has also an X interface, which provides 2-dimensional formula output (not of a very high quality) and a convenient on-line help.

Here is a sample *REDUCE* session within TEXmacs.

```
Loading image file :/opt/reduce/lisp/psl/linux/red/reduce.img
REDUCE 3.7, 15-Apr-1999, patched to 14-Jun-2001 ...

1:  (x^2-y^2)/(x^2+y^2)+sin(alpha)^2;
```

$$\frac{\sin(\alpha)^2 x^2 + \sin(\alpha)^2 y^2 + x^2 - y^2}{x^2 + y^2}$$

```
2:  (x+y-1)^5;
```

$$x^5 + 5x^4y - 5x^4 + 10x^3y^2 - 20x^3y + 10x^3 + 10x^2y^3 - 30x^2y^2 + 30x^2y - 10x^2 + 5xy^4 - 20xy^3 + 30xy^2 - 20xy + 5x + y^5 - 5y^4 + 10y^3 - 10y^2 + 5y - 1$$

```
3:  solve(a*x^2+b*x+c=0,x);
```

$$\left\{x = \frac{\sqrt{-4ac+b^2} - b}{2a},\ x = \frac{-\left(\sqrt{-4ac+b^2} + b\right)}{2a}\right\}$$

```
4:  int(sqrt(x^2+a),x);
```

$$\frac{\sqrt{a+x^2}\,x + \log\left(\frac{\sqrt{a+x^2}+x}{\sqrt{a}}\right) a}{2}$$

```
5:  int(exp(sin(x)),x);
```

$$\int e^{\sin(x)}\,dx$$

```
6:  df(f(x),x,2);
Declare f operator ?  y
```

$$\frac{\partial^2 f(x)}{\partial x^2}$$

```
7:  taylor(sin(x),x,0,10);
```

$$x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7 + \frac{1}{362880}x^9 + O\left(x^{11}\right)$$

```
8:  m:=mat((a,b),(c,d));
```

$$m := \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

```
9:  1/m;
```

$$\begin{pmatrix} \frac{d}{ad-bc} & \frac{-b}{ad-bc} \\ \frac{-c}{ad-bc} & \frac{a}{ad-bc} \end{pmatrix}$$

```
10:  plot(sin(x)/x,x=(-10 ..  10));
11:  procedure f(x,y); sin(sqrt(x^2+y^2))/sqrt(x^2+y^2);
f
12:  plot(f(x,y),x=(-10 ..  10),y=(-10 ..  10),hidden3d,points=40);
13:  bye;
Quitting
The end
```

Plots appear in separate windows, they are displayed by `gnuplot` (Fig. 3). Unfortunately, it is not possible to control them interactively, one has to use options in the `plot` procedure. In order to save a plot to a PostScript file, the options

```
terminal="postscript eps",output="filename.eps"
```

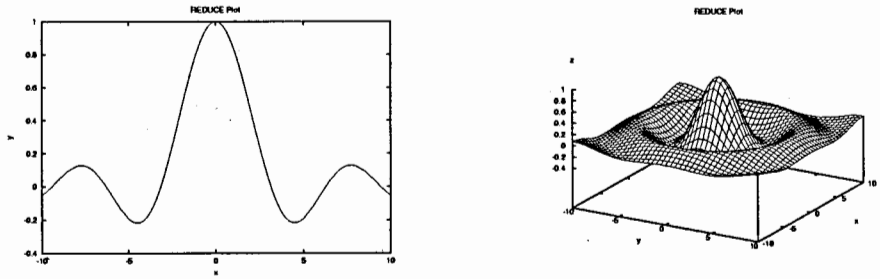


Figure 3: *REDUCE* plots

are used.

The question mark icon displays help menu. *REDUCE* manual (written in LaTeX) is imported into TeXmacs. Some LaTeX constructs are not handled correctly, but, nevertheless, the manual is quite readable.

All interfaces described in this talk are preliminary, and require more work. It is not difficult to implement TeXmacs interfaces with more CASs. TeXmacs progresses fast; in the future, it can become a complete scientist's work place, suitable both for writing articles and for doing calculations using various external systems, within the same comfortable environment.

I am grateful to Joris van der Hoeven for numerous discussions about TeXmacs and CAS interfaces; to William Schelter for his great help with *Maxima* and its LaTeX generation; to Ralf Hillebrand for similar help with *MuPAD* and for providing an improved LaTeX generation library; to Winfried Neun for useful discussions about *REDUCE*.

Note added: Professor William Schelter died soon after the workshop. Let *GNU Common Lisp* and *Maxima*, the only free CAS, be his living memory.

References

[1] J. van der Hoeven, http://www.texmacs.org/ http://www.texmacs.org/

[2] W. F. Schelter, http://www.ma.utexas.edu/maxima.html
http://www.ma.utexas.edu/maxima.html

[3] *MuPAD* team, http://www.mupad.com/
http://www.mupad.com/

[4] A. C. Hearn, http://www.uni-koeln.de/REDUCE/
http://www.uni-koeln.de/REDUCE/

Multiloop Calculations in Heavy Quark Effective Theory

A. G. Grozin

Budker Institute of Nuclear Physics,
630090 Novosibirsk, Russia;
e-mail: A.G.Grozin@inp.nsk.su

Heavy Quark Effective Theory (HQET) is a new approach to QCD problems with a single heavy quark, when all characteristic momenta are much less than its mass. It simplifies treatment of such problems, and reveals new symmetries hidden in QCD. Recently, an algorithm of calculation of arbitrary three-loop propagator diagrams in HQET has been constructed, and implemented as a REDUCE package Grinder. It is about 3000 lines long, and its testing required more than a CPU-month on good workstations. Large parts of this package were also ported to Axiom. Finding QCD/HQET matching coefficients requires calculation of on-shell diagrams with a massive quark in QCD. An algorithm of three-loop calculations of single-mass on-shell propagator diagrams has also been constructed, and implemented in FORM. A substantial progress was also achieved in calculation of diagrams with more than two legs in lower loops.

Maple Implementing Algebraic Perturbation Theory Algorithm: Hydrogen Atom in Weak Electric Fields

Alexander Gusev[1], Valentin Samoilov[1],
Vitaly Rostovtsev[2], Sergue Vinitsky[3]

[1]Scientific Center for Applied Research,
[2]Laboratory of Information Technologies,
[3]Bogoliubov Laboratory of Theoretical Physics,
Joint Institute for Nuclear Research,
141980 Dubna, Moscow Region, Russia;

Algorithms for evaluation of eigenvalues and eigenfunctions of a hydrogen atom in electric fields in framework of perturbation theory is implemented on MAPLE. A program description as well as input and output files are represented.

1 Introduction

In articles [3, 4, 5] algorithms for calculating eigenfunctions and eigenvalues of hydrogen atom in the homogeneous electric field and in the field of distant charge were represented. They have been implemented by means of CAS REDUCE.

Algorithms of this sort are needed to construct of combined program package for computing both atomic systems properties in external fields and effective potentials in quantum mechanical three body problem [2, 1].

We have been generalizing our algorithms and implementing them in CAS Maple. In this case, we wrote the program without using rewrite rules. These algorithm and program are represented in the article. In addition, we perform a comparison of the packages efficiency and results of calculation of energy spectrum up to 9 order by small parameter.

2 Setting of the Problem

Concider Shroedinger equation of a hydrogen-like atom with the charge Z_a in weak electric field

$$\left(\frac{1}{2}\mathbf{p}^2 - \frac{Z_a}{r} + V(r, x_3) - E\right)|\overline{\Phi}\rangle = 0, \qquad \langle\overline{\Phi}|\overline{\Phi}\rangle = 1 \tag{1}$$

where E is an energy and $|\overline{\Phi}\rangle$ are wave functions of hydrogen atom. $V(r, x_3)$ is potential, which can be represented as a series

$$V = \sum_{k=1} \varepsilon^k V_k, \quad \varepsilon \ll 1.$$

The wave function $|\overline{\Phi}\rangle$ and the energy $E = E(R)$ are sought for in the form

$$|\overline{\Phi}\rangle = \sum_{k=0} \varepsilon^k |\overline{\Phi}^{(k)}\rangle, \quad E = \sum_{k=0} \varepsilon^k E^{(k)}. \tag{2}$$

Using these expansions we rewrite equation (1) in form

$$\left(\frac{1}{2} r\mathbf{p}^2 - Z_a - \sum_{k=1} \varepsilon^k V^{(k)}(x_3, r) - rE^{(0)}\right)|\overline{\Phi}\rangle = 0. \tag{3}$$

Then we can write (3) in notations generators L_{56}, $A_3 = L_{34}$, $L_3 = L_{12}, x_3 = L_{35} - L_{34}$, $r = L_{56} - L_{46}$ of $so(4,2)$ algebra [3]. Making use of the tilting transformation $\sqrt{\frac{\sqrt{-2E^{(0)}}}{n}} U$ [3], we go over from the hydrogen states $|\overline{\Phi}\rangle$ to the basis states $|\Phi\rangle$, we arrive at the following equation:

$$\left[L_{56} - \frac{Z_a}{\sqrt{-2E^{(0)}}} - \frac{1}{\sqrt{-2E^{(0)}}} \sum_{k=1} \varepsilon^k V^{(k)}\left(\frac{x_3}{\sqrt{-2E^{(0)}}}, \frac{r}{\sqrt{-2E^{(0)}}}\right)\right]|\Phi\rangle = 0. \tag{4}$$

The normalization condition for basis states $|\Phi\rangle$ follows from(1) for states $|\overline{\Phi}\rangle$

$$\langle\overline{\Phi}|\overline{\Phi}\rangle = \frac{\sqrt{-2E^{(0)}}}{n}\langle\Phi|UrU^{-1}|\Phi\rangle =$$

$$= \frac{\sqrt{-2E^{(0)}}}{n}\langle\Phi|\frac{r}{\sqrt{-2E^{(0)}}}|\Phi\rangle = \frac{1}{n}\langle\Phi|r|\Phi\rangle = 1. \tag{5}$$

Hence, it follows that in the course of the passage from the wave functions $|\Phi\rangle$ to the basis states $|\overline{\Phi}\rangle$, the first one should be normalized by the condition

$$\langle\Phi|r|\Phi\rangle = n, \tag{6}$$

which becomes an identity for states $|\Phi^{(0)}\rangle$.

3 The Scheme of Perturbation Theory and Algorithm

We look for a solution to equation (4) in the form of the perturbation series

$$|\Phi\rangle = \sum_{k=0}^{k_{max}} \varepsilon^k |\Phi^{(k)}\rangle. \tag{7}$$

The unknown coefficients $|\Phi^{(k)}\rangle$ satisfy the system of inhomogeneous differential equations

$$L(n)|\Phi^{(0)}\rangle = (L_{56} - n)|\Phi^{(0)}\rangle = 0 \equiv f^{(0)}, \tag{8}$$

$$L(n)|\Phi^{(1)}\rangle = \frac{r}{(-2E^{(0)})}[E^{(1)} - Z_b(Z_a - 1)]|\Phi^{(0)}\rangle \equiv f^{(1)}, \tag{8a}$$

$$L(n)|\Phi^{(k)}\rangle = \frac{r}{(-2E^{(0)})}\sum_{p=0}^{k-1} V^{(k-p)}\left(\frac{x_3}{\sqrt{-2E^{(0)}}}, \frac{r}{\sqrt{-2E^{(0)}}}\right)|\Phi^{(p)}\rangle \equiv f^{(k)} \tag{8b}$$

As the basis, which realizes the infinite-dimensional irreducible representation of the algebra $so(4,2)$ in x-space, we take the eigenfunctions $\langle y|s,t\rangle$ of commuting operators L_{56}, A_3 and L_3 distinguished from the basis functions $\langle y|n_1+s, n_2+t, m\rangle$ only by the normalization factor and coincide with them at $s = t = 0$

$$\langle y|s,t\rangle = \frac{C_{n_1 n_2 |m|}}{C_{n_1+s, n_2+t, |m|}}\langle y|n_1+s, n_2+t, m\rangle. \tag{9}$$

The operators L_{56}, x_3, r and L_3 on the functions $\langle y|s,t\rangle$ are defined by the relations without fractional powers of parabolic quantum numbers

$$L_{56}|s,t\rangle = (n_1+n_2+|m|+1+s+t)|s,t\rangle = (n+s+t)|s,t\rangle, \tag{10}$$

$$A_3|s,t\rangle = (-(n_1+s)+(n_2+t))|s,t\rangle, \quad L_3|s,t\rangle = |m||s,t\rangle, \tag{10a}$$

$$\begin{aligned} r|s,t\rangle &= (n+s+t)|s,t\rangle - \tfrac{1}{2}((n_1+s+|m|)|s-1,t\rangle + (n_1+s+1)|s+1,t\rangle \\ &+(n_2+t+|m|)|s,t-1\rangle + (n_2+t+1)|s,t+1\rangle), \end{aligned} \tag{10c}$$

$$\begin{aligned} x_3|s,t\rangle &= \tfrac{1}{2}(-(n_1+s+|m|)|s-1,t\rangle - (n_1+s+1)|s+1t\rangle \\ &+(n_2+t+|m|)|s,t-1\rangle + (n_2+t+1)|s,t+1\rangle) - (-(n_1+s)+(n_2+t))|s,t\rangle. \end{aligned} \tag{10d}$$

Suppose the azimutal quantum number m is positive $m = |m|$. Applying relations (10), we expand the right-hand side $f^{(k)}$ and solutions $|\Phi^{(k)}\rangle$ of the system (8) over basis states $|s,t\rangle$ (9):

$$f^{(k)} = \sum_{s,t} f^{(k)}_{st}|s,t\rangle, \quad |\Phi^{(k)}\rangle = \sum_{s,t} b^{(k)}_{st}|s,t\rangle. \tag{11}$$

Substituting (11) into (8) and taking into account the relation

$$L(n)|s,t\rangle = (s+t)|s,t\rangle$$

and orthogonality condition (4) of basis (9), we obtain the system of linear algebraic equations for unknown coefficients $b^{(k)}_{st}$ and perturbation corrections $E^{(k)}$, for each $k \geq 1$

$$(s+t)b^{(k)}_{st} - f^{(k)}_{st} = 0, \quad min(|s|,|t|,|s+t|) \leq k. \tag{12}$$

It enables us to find the coefficients $b^{(k)}_{st}$ using known coefficients $f^{(k)}_{st}$ from the above definitions. In the second order $k = 2$, we obtain the energy correction $E^{(2)} = 3ndZ_b/2Z_a$ at $s = t = 0$ and eight coefficients $b^{(2)}_{st}$ at $s + t \neq 0$:

$$b^{(2)}_{0,\mp 1} = \mp\frac{n^3 Z_b}{8Z_a^3}(d \mp m - n + 1)(d + 2n \mp 2)),$$

$$b^{(2)}_{0,\mp 2} = \mp\frac{n^3 Z_b}{32 Z_a^3}(d \mp m - n \mp 1)(d \mp m - n \pm 3)),$$

$$b^{(1)}_{1,-1} = \frac{n}{8Z_a}(d - m + n + 1)(d - m - n + 1)),$$

$$b^{(2)}_{1,-1} = -\frac{n^2}{8Z_a^2}(d - m + n + 1)(d - m - n + 1)(d + 1)),$$

$$b^{(2)}_{2,-2} = \frac{n^2}{128 Z_a^2}(d - m + n + 3)(d - m + n + 1)(d - m - n + 3)(d - m - n + 1)).$$

For Stark effect, coefficients $b^{(1)}_{1,-1}$, $b^{(1)}_{-1,1}$ are equal to zero and $b^{(1)}_{0,j}$ are equal to corresponding coefficients of hydrogen atom in the field of the distant charge $b^{(2)}_{0,j}$. The remaining coefficients are obtained up to the sign by interchanging n_1 by n_2 ($d = n_2 - n_1 \to -d$), $b^{(k)}_{st}(d) = (-1)^{k+1} b^{(k)}_{ts}(-d)$, $b^{(k)}_{s,-s}(d) = (-1)^k b^{(k)}_{-s,s}(-d)$. Indeed, at each k the functions $f^{(k)}_{st}$ depend on the unknown correction $E^{(k)}$ and known coefficients $E^{(p)}$ and $b^{(p)}_{st}$ for $p = 0, 1, \cdots, k-1$, which are evaluated from previous $k-1$ equations. The coefficients $b^{(p)}_{st}$ also depend on the corrections $E^{(q)}$ and $b^{(q)}_{st}$ with $q = 0, 1, \cdots, p-1$, which are evaluated from previous equations (12) by recurrence. Thus, in each order ($k \geq 1$) we calculate step-by-step the needed corrections $E^{(k)}, b^{(k-2)}_{s,-s}, b^{(k)}_{s,t}$ by solving the following algebraic equations:

$$f^{(k)}_{00}\left(E^{(k)}, E^{(p)}, b^{(p)}_{s't'}, 0 \leq p \leq k-1\right) = 0 \to E^{(k)}, \tag{13}$$

$$f^{(k)}_{s-s}\left(E^{(k)}, E^{(p)}, b^{(p)}_{s't'}, 1 \leq p \leq k-1\right) = 0 \to b^{(k-2)}_{s-s}, \tag{13a}$$

$$b^{(k)}_{s,t} = (s+t)^{-1} f^{(k)}_{st}\left(E^{(k)}, E^{(p)}, b^{(p)}_{s't'}, 0 \leq p \leq k-1\right). \tag{13b}$$

The initial conditions for the recurrence procedure are given by

$$E^{(0)} = -\frac{Z_a^2}{2n^2}, \quad b^{(0)}_{0,0} = 1, \quad b^{(0)}_{s,t} = 0 \quad \text{for} \quad s, t \neq 0, \quad b^{(k)}_{0,0} = 0. \tag{14}$$

To obtain the normalized wave function Φ by condition (6) up to the kth order, we must redefine the coefficient $b^{(k)}_{0,0}$ by the following relation:

$$b^{(k)}_{00} = -\frac{Z_a}{2n^2}\sum_{p=0}^{k}\sum_{s',t'}\sum_{s,t} b^{(k-p)}_{st}\langle s, t|\frac{r}{\sqrt{-2E^{(0)}}}|s', t'\rangle b^{(p)}_{s't'}. \tag{15}$$

In particularly, the coefficient $b^{(1)}_{0,0} = 0$ and $b^{(2)}_{0,0}$ is defined by

$$b^{(2)}_{0,0} = \frac{n^2}{64 Z_a^2}(-d^4 + (2d^2 - m^2)(m^2 + n^2 - 3) + m^2(3n^2 - 1) - (n^2 - 1)^2) + \frac{3dn^3 Z_b}{4Z_a^3}. \tag{16}$$

Then the algorithm is evaluated step by step as follows.

1. "Potential". Coefficients of potential and energy expansions are calculated with respond to (2). The step has two branch of calculations: in case of Stark effect and in case of point charge.

2. "Action of potential by ket". The results of actions of the operators r and x_3 with respect of (10c,10d) are defined in this step. Sequence of the calculations is described in the next section.

3. Step "finding eigenvalues" realizes the main goal of the calculations. Indeed, in this step we are calculated the corrections to $f^{(k)}$ and $E^{(k)}$ by means of solving equations (13), (13a), (13b) up to preassigned order k_{max}.

4 Program MW3STARK and Results

In this Section we adduce the main program **MW3STARK** as a pilot implementation of the above algorithm in MAPLE VR5. First of all we defined next parameters:

- `stark0_point1` – program switch: 0-stark, 1-point charge;
- `kmax` – a number of spectrum corrections;
- `bbound` – a boundary of summation range in expansions (11);
- `-bbound`$\leq s,t \leq$`bbound`;
- `fdE,fdB` are names of output files.

The program consists of blocks in respect to the algorithm described in previous section.

1. In block "POTENTIAL", we denote by V the potential multiplied by r and $ee(k)$ and the energy $E^{(k)}$ (2). Program variables `vr, vx3` and `vr0, vx30` are the variables r and x_3 in (3). The Coice of calculation branch is realised by switch `stark0_point1`.
2. The block "ACTION OF POTENTIAL BY KET" consist of the following parts.
   (a) In block "action r and x_3 by ket" results of action of the operators r and x_3 are defined.
   (b) In block "action v by ket" coefficients `actket(k,x1,x2)` of an expansion "$V(k)ket(N1,N2,m) = \sum actket(k,x_1,x2) * ket(N1+x_1,N2+x_2,m)$" are defined and evaluated in following parts:
      i. "preparation": set all `actket(k,x1,x2)` to zero. Then set `actket(k,0,0)` to value of potentials `V(k,vx3,xr)`, and assign a degree of `V(k,vx3,xr)` over `(vx3,vr)` to a variable `vmax(k)`;
      ii. "action of operator r by ket": application of (10) for excluding r in calculation of potential action. In order that, we are using an auxiliary variable `ketvspom`. Terms containing the variable `vr` are transferred from `actket(k,x1,x2)` to `ketvspom`. Then we factorized `vr` and act on `ket`, accordingly to step 2a. Boundaries of region `(x1,x2)` and a number of application of this procedure for $k-th$ order are defined by means of variable `vmax(k)`;

iii. "action of operator x_3 by ket": An application of the formula (10d) to exclude the variable x_3 is used under calculation of the potential action;

iv. "test": are transfer to new quantum numbers "NN, ND".

3. The subroutine "finding eigenvalues realizes" consist of the following parts.

   (a) "Preparation": the zero values are assigned to all `b(k,js,jt)` but `b(0,0,0)=1`;

   (b) "Calculations": extracting eigenvalues and eigenfunctions. By means of loop under `k` from `sbros` to `kmax`, we are accomplished the following actions:

      i. "calculating f": by (8) and (11) we evaluate coefficients `b(k,js,jt)`;

      ii. "calculating ee(k)": we solve equation (13);

      iii. "calculating b(k-sbros,s,-s)": an auxiliary block;

      iv. "add v(sbros)*b(k-sbros,s,-s)": we are taking account over contribution of the action `v(sbros)` of the diagonal components `b(k-sbros,js,-js)`;

      v. `calculating b(k,s,t)`: we solve equation (13b).

4. Print(b(k,s,t)).

5. Print time.

```
#program mw3stark;
restart;timestart:=time():
stark0_point1:=1; # 1 for atom in the field of the point charge,
#stark0_point1:=0; # 0 for stark effect
kmax:=11;
bbound:=20;
fdE:=open("e.out",WRITE): fdB:=open("b.out",WRITE):

#1. potential
 if stark0_point1=0 then
  V(1,vx3,vr):=expand(subs(vr0=vr*n/Za,vx30=vx3*n/Za
      ,(ee(1)*vr0-vr0*vx30)*n/Za));
  for k from 2 to kmax do   V(k,vx3,vr):=
   expand(subs(vr0=vr*n/Za,vx30=vx3*n/Za,ee(k)*vr0*n/Za));
  od:
  fprintf(fdE,"Hydrogen atom in homogeneous
    electric field\n kmax="):
  fprintf(fdB,"Hydrogen atom in homogeneous
   electric field\n kmax="):
  fprintf(fdE,cat(convert(kmax,string),";\n")):
  fprintf(fdB,cat(convert(kmax,string),";\n")):
  sbros:=1;
 fi:
 if stark0_point1=1 then
```

```
  V(1,vx3,vr):=expand(subs(vr0=vr*n/Za,vx30=vx3*n/Za
       ,n/Za*expand(vr0*(ee(1)-Zb*(Za-1)))));
  for k from 2 to kmax do
    V(k,vx3,vr):=expand(subs(vr0=vr*n/Za,vx30=vx3*n/Za
       ,n/Za*expand(vr0*(ee(k)+Zb*vr0^(k-1)
        *orthopoly[P](k-1,-vx30/vr0)))));
  od;
  fprintf(fdE,"Hydrogen atom in the field
   of the distant charge\n kmax="):
  fprintf(fdB,"Hydrogen atom in the field
   of the distant charge\n kmax="):
  fprintf(fdE,cat(convert(kmax,string),";\n")):
  fprintf(fdB,cat(convert(kmax,string),";\n")):
  sbros:=2;
  ee(1):=Zb*(Za-1);
  fprintf(fdE,"ee(1) := "):
  fprintf(fdE,cat(convert(ee(1),string),";\n")):
 fi:
#2.actions of potential by ket #2a.action r and x3 by ket rul1:=
 rket ( 0, 0):=(N1+N2+m+1):  x3ket ( 0, 0):=-(-N1+N2):
 rket (-1, 0):=-1/2*(N1+m):  x3ket (-1, 0):=-1/2*(N1+m):
 rket ( 1, 0):=-1/2*(N1+1):  x3ket ( 1, 0):=-1/2*(N1+1):
 rket ( 0,-1):=-1/2*(N2+m):  x3ket ( 0,-1):=1/2*(N2+m):
 rket ( 0, 1):=-1/2*(N2+1):  x3ket ( 0, 1):=1/2*(N2+1):

#2b. action v by ket #2b1.podgotovka
 for k from 0 to sbros-1 do vmax(k):=0 od:
 for k from sbros to kmax do
  vmax(k):=degree(subs(vx3=vx,vr=vx,V(k,vx3,vr)),vx);
  for js from -vmax(k) to vmax(k) do
   for jt from -vmax(k)+abs(js) to vmax(k)-abs(js) do
    actket(k,js,jt):=0;
  od od:
  actket(k,0,0):=V(k,vx3,vr);
 od:

#2b2. action of operator r by ket
 for k from sbros to kmax do
  for j from 0 to vmax(k)-1 do
   for js from -j to j do
    for jt from -j+abs(js) to j-abs(js) do
     ketvspom:=simplify((actket(k,js,jt)
         -subs(vr=0,actket(k,js,jt)))/vr);
     actket(k,js,jt):=subs(vr=0,actket(k,js,jt));
     for a in [0,0],[0,1],[1,0],[0,-1],[-1,0] do
```

```
      as:=op(1,a);
      at:=op(2,a);
      actket(k,js+as,jt+at):=simplify(actket(k,js+as,jt+at)
       +subs(N1=N1+js,N2=N2+jt,rket(as,at))*ketvspom);
 od:od:od:od:od:

#2b3. action of operator x3 by ket
 for k from sbros to kmax do
  for j from 0 to vmax(k)-1 do
   for js from -j to j do
    for jt from -j+abs(js) to j-abs(js) do
     ketvspom:=simplify((actket(k,js,jt)
      -subs(vx3=0,actket(k,js,jt)))/vx3);
     actket(k,js,jt):=subs(vx3=0,actket(k,js,jt));
     for a in [0,0],[0,1],[1,0],[0,-1],[-1,0] do
      as:=op(1,a); at:=op(2,a);
      actket(k,js+as,jt+at):=simplify(actket(k,js+as,jt+at)
       +subs(N1=N1+js,N2=N2+jt,x3ket(as,at))*ketvspom);
 od:od:od:od:od:

#2b4. test
 for k from sbros to kmax do
  for js from -vmax(k) to vmax(k) do
   for jt from -vmax(k)+abs(js) to vmax(k)-abs(js) do
   actket(k,js,jt):=collect(subs(N1=(NN+ND-m-1)/2
    ,N2=(NN-ND-m-1)/2,actket(k,js,jt)),ee,simplify);
 od:od:od:

#3. finding eigenvalues #3a. podgotovka
 for k from 0 to kmax do
 for js from -bbound to bbound do
 for jt from -bbound to bbound do
 b(k,js,jt):=0;
 od:od:od: b(0,0,0):=1:
#3b. calculations
 for k from sbros to kmax do

  #3b1. calculating f (now b(k,s,t)=f(k,s,t))
  for kpb from 0 to k-sbros do
  for jsb from -bbound to bbound do
  for jtb from -bbound to bbound do
   if b(kpb,jsb,jtb)<>0 then
    for js from -vmax(k-kpb) to vmax(k-kpb) do
    for jt from -vmax(k-kpb)+abs(js) to vmax(k-kpb)-abs(js) do
      if (abs(js+jsb)>bbound) or (abs(jt+jtb)>bbound)
```

```
      then print("bbound is very small") fi:
   b(k,js+jsb,jt+jtb):=collect(b(k,js+jsb,jt+jtb)
   +b(kpb,jsb,jtb)*subs(NN=n+jsb+jtb,ND=d+jsb-jtb
    ,actket(k-kpb,js,jt)),ee,simplify);
  od: od:
 fi:
 od:od:od:

#3b2. calculation ee(k):
ee(k):=
 simplify(-coeff(b(k,0,0),ee(k),0)/coeff(b(k,0,0),ee(k),1)):
fprintf(fdE,cat( "\n ee(",convert(k,string)));
fprintf(fdE,cat( ") := ",convert(ee(k),string)));
fprintf(fdE,";\n ");

#3b3 calculation b(k-sbros, s,-s)
for js from 1 to bbound do
if((simplify(b(k,js,-js))<>0)or(simplify(b(k,-js,js)<>0)))
 then
 b(k-sbros, js,-js):=simplify(
  -b(k,js,-js)/subs(NN=n,ND=d+2*js,actket(sbros,0,0))):
 b(k-sbros, -js,js):=simplify(
  -b(k,-js,js)/subs(NN=n,ND=d-2*js,actket(sbros,0,0))):
fi:
od:

#3b4. add  v(sbros)*b(k-sbros...) to f(k,s,t)
for j from -bbound to bbound do
if simplify(b(k,j,-j))<>0 then
 for js from -vmax(sbros) to vmax(sbros) do
 for jt from -vmax(sbros)+abs(js) to vmax(sbros)-abs(js) do
  if (abs(js+j)>bbound) or (abs(jt-j)>bbound)
   then print("bbound is very small") fi:
  b(k,js+j,jt-j):=simplify(b(k,js+j,jt-j)
   +b(k-sbros,j,-j)*subs(NN=n,ND=d+2*j,actket(sbros,js,jt)));
 od:od:
fi:
od:

#3b5. calculation b(k, s, t) (now b(k,s,t))
for js from -bbound to bbound do
for jt from -bbound to bbound do
 if js<>-jt then
   b(k,js,jt):=simplify(b(k,js,jt)/(js+jt)):
 fi:
```

```
 od:od:
 print(k);
od:

#4. print(b);
for k from 0 to kmax-sbros do
 for js from -bbound to bbound do
  for jt from -bbound to bbound do
   if b(k,js,jt)<>0 then
    fprintf(fdB,cat( "\n b(",convert(k,string)));
    fprintf(fdB,cat( ",",convert(js,string)));
    fprintf(fdB,cat( ",",convert(jt,string)));
    fprintf(fdB,cat( "):=",convert(b(k,js,jt),string)));
    fprintf(fdB,";\n");
   fi:
od:od:od:

#5.
fprintf(fdE,cat("\n\nTime:=",convert(time()-timestart,string))):
fprintf(fdE," s;\n"):
close(fdE):close(fdB):
```

To check the algorithm and program we have been calculating the corrections to $b_{st}^{(k)}$ and $E^{(k)}$ up to 11-th order. The results are very space consuming so we cannot adduce then here. This test was calculated on computer PC-2 350MHz 64MB memory. In the above formulas, we use the following notations: $\mathbf{n} \equiv n = n_1 + n_2 + |m| + 1$, $\mathbf{m} \equiv |m|$, $\mathbf{d} \equiv n_1 - n_2$, $\mathbf{za} \equiv Z_a$, $\mathbf{zb} \equiv Z_b$. Note that our results coincide up to the seventh order with [3, 4, 5] (the orders 8-11 was not calculated in the sited works because technical limits).

5 Conclusion

We have demonstrated the efficiency of the proposed recursive symbolic algorithm in the framework of an algebraic version of the conventional perturbation theory without assumption on separation of independent variables in both parabolic and spherical representations, which are needed for applications [5]

Acknowledgements

We are grateful to Professor V.P. Gerdt for his support of this work and useful discussion. The authors (AG, SI) appreciate support of Russian Foundation for Basic Research (grant No. 02-02-17617).

References

[1] Faifman, M.P., Ponomarev, L.I., Vinitsky, S.I.: Asymptotic form of effective potential of the Coulomb three-body problem in the adiabatic representation *J. Phys. B: Atom. Molec. Phys.* **9** (1976) 2255–2268.

[2] Abrashkevich, A.G., Puzynin, I.V., Vinitsky, S.I.: ASYMPT: a program for calculating asymptotics of hyperspherical potential curves and adiabatic potentials. *Computer Physics Communications* **125** (2000) 259–281.

[3] Gusev, A.A., Samoilov, V.N., Rostovtsev, V.A., Vinitsky, S.I.: Symbolic Algorithm of Algebraic Perturbation Theory of a Hydrogen Atom: the Stark effect. In: *Computer Algebra in Scientific Computing*, V.G.Ganzha, E.W. Mayr, E.V. Vorozhtsov (Eds.), Springer-Verlag, Berlin, 2000, pp. 219–231.

[4] A.A. Gusev, V.N. Samoilov, V.A. Rostovtsev, S.I. Vinitsky, Algebraic perturbation theory for hydrogen atom in weak Electric Fields (in Russian) Programing v1, p27-31 (2001).

[5] A.A. Gusev,V.N. Samoilov, V.A. Rostovtsev, S.I. Vinitsky: Symbolic Algorithms of Algebraic Perturbation Theory: Hydrogen Atom in the Field of Distant Charge Proc. of Fourth Workshop on "Computer Algebra in Scientific Computing", (Konstanz, September 22 - 26, 2001), CASC-2001, Eds. V.G. Ganzha, E.W.Mayr, E.V. Vorozhtsov, Springer-Verlag, 2001, pp. 309-322.

Completion to Involution and Symmetry Analysis

Marcus Hausdorf, Werner M. Seiler

Lehrstuhl für Mathematik I, Universität Mannheim,
68131 Mannheim, Germany;
e-mail: {hausdorf,werner.seiler}@math.uni-mannheim.de
http://www.math.uni-mannheim.de/~wms

We discuss some applications of the theory of involutive systems in the Lie symmetry analysis of partial differential equations. Special emphasis is put on non-classical symmetries.

Supported by Deutsche Forschungsgemeinschaft, Landesgraduiertenförderung Baden - Württemberg and INTAS grant 99-1222.

1 Introduction

Lie symmetry analysis represents one of the most important techniques for tackling differential equations, in particular non-linear ones. It often provides the only possibility to find solutions in closed form. But many other applications exist, too; perhaps best known is the construction of conservation laws via Noether's law. By now, a number of textbooks on symmetry analysis are available (see e. g. [3, 9, 13]); a survey of computer algebra tools for symmetry analysis was given by Hereman [5, 6].

The theory of involutive systems appears at many places in symmetry analysis. The best known application is, of course, the completion of the determining systems in order to facilitate its integration in closed form. In fact, this problem has been a major impetus for the renewal of interest in involution in the last, say, twenty years. In this article we will, however, concentrate on other, less known applications.

In the literature one can find several different notions of involutive (or passive or complete) systems. We will use the so-called formal theory [14, 18]. It is a geometric approach and may be formulated completely intrinsically. This makes its application in symmetry theory very natural.

This article is organised as follows. The next section briefly reviews some basic notions of the formal theory. Then we study the role involution plays for the very definition of symmetries. Section 4 contains some formal results about measuring the size of the solution space of differential equations. Then we analyze symmetry reductions; this leads in a natural manner to non-classical symmetries. Finally, some conclusions are given.

2 Involution

For lack of space, we cannot give here a detailed introduction into the formal theory or the underlying jet bundle formalism. Our presentation follows [18]; a general reference is the book of Pommaret [14].

For simplicity, we will use local coordinates. Let $\boldsymbol{x} = (x^1, \ldots, x^n)$ and $\boldsymbol{u} = (u^1, \ldots, u^m)$ be an adapted coordinate system on the fibred manifold $\pi : \mathcal{E} \to \mathcal{X}$. Coordinates on the qth order jet bundle $J_q\mathcal{E}$ are obtained by adding all derivatives

$$p^\alpha_\mu = \partial^{|\mu|} u^\alpha / \partial (x^1)^{\mu_1} \cdots \partial (x^n)^{\mu_n}$$

with $|\mu| \leq q$. Here $\mu = [\mu_1, \ldots, \mu_n]$ is a multi index and $|\mu| = \mu_1 + \cdots + \mu_n$ its length. Thus the jet bundle $J_q\mathcal{E}$ may be interpreted as the space of all Taylor series truncated at order q.

We define a *differential equation*[1] of order q intrinsically as a fibred submanifold $\mathcal{R}_q \subseteq J_q\mathcal{E}$. Locally, $\mathcal{R}_q$ is described by a system of equations

$$\mathcal{R}_q : \left\{ \Phi^\tau(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}) = 0\,, \quad \tau = 1, \ldots, p\,. \right. \tag{1}$$

A *solution* is a (local) section $\sigma : \mathcal{X} \to \mathcal{E}$ such that the image of its prolongation $j_q\sigma : \mathcal{X} \to J_q\mathcal{E}$ completely lies in $\mathcal{R}_q$. Locally, a section is of the form $\sigma(\boldsymbol{x}) = \big(\boldsymbol{x}, \boldsymbol{s}(\boldsymbol{x})\big)$ with some functions $\boldsymbol{s} = (s^1, \ldots, s^m)$. For its prolongation we differentiate these functions and σ is a solution, if the substitution $p^\alpha_\mu = \partial^{|\mu|} s^\alpha / \partial (x^1)^{\mu_1} \cdots \partial (x^n)^{\mu_n}$ in (1) yields an identity.

The *prolongation* $\mathcal{R}_{q+1} \subseteq J_{q+1}\mathcal{E}$ is obtained by formally differentiating every equation in (1) with respect to all independent variables $\boldsymbol{x}$, i. e. by adding the differential equations

$$D_i \Phi^\tau = \frac{\partial \Phi^\tau}{\partial x^i} + \frac{\partial \Phi^\tau}{\partial u^\alpha} p^\alpha_i + \frac{\partial \Phi^\tau}{\partial p^\alpha_\mu} p^\alpha_{\mu+1_i} = 0\,, \qquad i = 1, \ldots, n\,. \tag{2}$$

Iteration yields higher order prolongations $\mathcal{R}_{q+r} \subseteq J_{q+r}\mathcal{E}$. A (smooth) solution of $\mathcal{R}_q$ is automatically also a solution of $\mathcal{R}_{q+r}$ and vice versa.

The *symbol* $\mathcal{M}_q$ of the differential equation $\mathcal{R}_q$ locally represented by the system (1) is the solution space of the following linear system of (algebraic!) equations in the unknowns v^α_μ

$$\mathcal{M}_q : \left\{ \sum_{\alpha=1}^{m} \sum_{|\mu|=q} \left(\frac{\partial \Phi^\tau}{\partial p^\alpha_\mu} \right) v^\alpha_\mu = 0\,. \right. \tag{3}$$

(By abuse of language, we will refer to both the linear system and its solution space as the symbol). The place-holders v^α_μ are coordinates of a finite-dimensional vector space; we introduce one for each derivative of order q. For a linear differential equation, the symbol is simply obtained by considering only the highest order or principal part and substituting v^α_μ for p^α_μ.

Let us make a power series ansatz for the solutions of the differential equation $\mathcal{R}_q$ expanding around some point $\boldsymbol{x}_0 \in \mathcal{X}$. Substituting into (1) and evaluating at $\boldsymbol{x} = \boldsymbol{x}_0$

[1] Note that we do not distinguish between scalar equations and systems!

yields a system of algebraic equations for the Taylor coefficients up to order q. Substituting the ansatz into $\mathcal{R}_{q+r}$ and evaluating again at $x = x_0$ yields an inhomogeneous linear system for the coefficients of order $q+r$. Its homogeneous part is given by the matrix of the prolonged symbol $\mathcal{M}_{q+r}$, i. e. the symbol of the prolonged equation $\mathcal{R}_{q+r}$.

This order by order construction of a formal power series solution fails, if integrability conditions occur. They pose additional conditions on coefficients of lower order and must all be known in order to pursue the above described procedure. A differential equation containing all its integrability conditions is called *formally integrable*, as it possesses a formal power series solution. Unfortunately, no finite criterion for formal integrability is known.

For an intrinsic formulation of integrability conditions we exploit the natural projection maps between jet bundles of different order (they simply forget the higher order derivatives). For $r > q$ we write $\pi_q^r : J_r\mathcal{E} \to J_q\mathcal{E}$ and define $\mathcal{R}_q^{(r)} = \pi_q^{q+r}(\mathcal{R}_{q+r}) \subseteq \mathcal{R}_q$, i. e. we first prolong r times and then project back r times. Integrability conditions appear during these prolongations, if and only if $\mathcal{R}_q^{(r)} \subsetneq \mathcal{R}_q$.

Definition 1. *The differential equation $\mathcal{R}_q$ is* formally integrable, *if for all integers $r > 0$ the equality $\mathcal{R}_{q+r}^{(1)} = \mathcal{R}_{q+r}$ holds.*

It follows from the construction above that the arbitrariness of the general formal power series solution is reflected by the dimensions of the prolonged symbols, because at each order $\dim \mathcal{M}_{q+r}$ coefficients are not determined by the differential equations but can be chosen freely. Formal integrability does, however, not suffice to determine these dimensions without explicitly constructing the prolonged symbols.

A jet variable p_μ^α is said to be of *class* k, if μ_k is the first non-vanishing entry of its multi index. We order the columns of the symbol $\mathcal{M}_q$ by class (highest class first) and compute a row echelon form. If v_μ^α is the leading term of an equation in this solved form of the symbol, then p_μ^α is a *principal* derivative. All other derivatives are *parametric*. We define the indices $\beta_q^{(k)}$ of the symbol $\mathcal{M}_q$ as the number of principal derivatives of class k.

Definition 2. *The symbol $\mathcal{M}_q$ is called* involutive, *if*

$$\operatorname{rank} \mathcal{M}_{q+1} = \sum_{k=1}^{n} k\beta_q^{(k)} . \tag{4}$$

The differential equation $\mathcal{R}_q$ is involutive, *if it is formally integrable and if its symbol $\mathcal{M}_q$ is involutive.*

In contrast to formal integrability, involution can be checked in a finite manner. To decide whether a symbol is involutive requires only some linear algebra. For differential equations we note the following very important criterion for involution that needs only one prolongation instead of infinitely many ones as the definition of formal integrability.

Theorem 1. *Let $\mathcal{R}_q$ be a differential equation with involutive symbol $\mathcal{M}_q$. Then $\mathcal{R}_q$ is involutive, if and only if $\mathcal{R}_q^{(1)} = \mathcal{R}_q$.*

Two important results on involutive equations are the *Cartan–Kuranishi* and the *Cartan–Kähler Theorem*. The former one states that any (regular) differential equation $\mathcal{R}_q$ can be completed in a finite number of steps to an equivalent differential equation $\mathcal{R}_{q+r}^{(s)}$ that is involutive; discussions of the algorithmic realisation of this completion can be found in [4, 17]. The latter is an existence and uniqueness theorem in the analytic category; if $\mathcal{R}_q$ is an analytic involutive differential equation, then a certain initial value problem for it has a unique analytic solution for analytic initial data. Thus it generalises the well-known Cauchy–Kovalevskaya Theorem.

3 Basic Symmetry Theory

Symmetry theory [3, 9, 13] provides us with some of the most important techniques for the analysis of differential equations, in particular for non-linear equations. A somewhat subtle and often overlooked point is the importance of the local solvability of the analysed differential equation. Essentially, this concerns the equivalence of two different definitions of what a symmetry of a differential equation actually is.

Definition 3. *A* symmetry *of a differential equation is a transformation that maps any of its solutions into another solution.*

This probably represents the most general definition of a symmetry of a differential equation. It is still somewhat vague, as it is not clearly specified what kinds of transformations or solutions are considered. We restrict in this article to point symmetries, i.e. to diffeomorphisms $\phi : \mathcal{E} \to \mathcal{E}$, and to formal power series solutions. Definition 3 is very hard to apply in practice, as it requires the knowledge of the full solution space. Thus usually the following, more geometric approach is taken.

Definition 4. *A diffeomorphism $\phi : \mathcal{E} \to \mathcal{E}$ is a* point symmetry *of the differential equation $\mathcal{R}_q \subseteq J_q\mathcal{E}$, if its prolongation $\phi^{(q)} : J_q\mathcal{E} \to J_q\mathcal{E}$ leaves the submanifold $\mathcal{R}_q$ invariant.*

The infinitesimal version of this definition forms the basis of all concrete computations with Lie symmetries. One considers a one-parameter group of diffeomorphisms ϕ_ϵ generated by a vector field X over $\mathcal{E}$. Given a local description of $\mathcal{R}_q$ in the form (1), a linearisation of the condition above leads to the well-known infinitesimal symmetry criterion

$$X^{(q)}\Phi^\tau\big|_{\mathcal{R}_q} = 0 \tag{5}$$

where the prolonged vector field $X^{(q)}$ on $J_q\mathcal{E}$ is the generator of $\phi_\epsilon^{(q)}$. $X^{(q)}$ and $\phi_\epsilon^{(q)}$ are easily determined by the chain rule.

With respect to our adapted coordinates, an arbitrary vector field X over $\mathcal{E}$ may be locally written in the form $X = \xi^i(\boldsymbol{x}, \boldsymbol{u})\partial_{x^i} + \eta^\alpha(\boldsymbol{x}, \boldsymbol{u})\partial_{u^\alpha}$. Evaluation of the criterion (5) yields an overdetermined linear system of partial differential equations for the coefficients ξ^i and η^α, the *determining system* $\mathcal{D}_q$ (it has the same order as the equation $\mathcal{R}_q$ to be analysed).

It is often assumed that Definitions 3 and 4 are equivalent, but this holds only under certain assumptions on the differential equation $\mathcal{R}_q$. We must assume that for every point $\rho \in \mathcal{R}_q$ at least one solution $\sigma : \mathcal{X} \to \mathcal{E}$ exists such that $\rho \in j_q(\sigma)$. This property is known as *local solvability*. It is trivial for scalar equations or systems in Cauchy–Kovalevskaya form but becomes an issue for overdetermined systems.

We illustrate the problem with a very simple example. Consider the following first order equation:

$$\mathcal{R}_1 : \begin{cases} u_z + y u_x = 0\,, \\ u_y = 0\,. \end{cases} \tag{6}$$

Since y appears explicitly, the manifold $\mathcal{R}_1$ does not remain invariant under y-translations and thus they are *not* symmetries in the sense of Definition 4. Cross-differentiations show that $\mathcal{R}_1^{(1)}$ is defined by the equations $u_x = u_y = u_z = 0$. Hence the general solution of our equation is $u(x, y, z) = const$ and y-translations *are* symmetries in the sense of Definition 3. The geometric approach "loses" this symmetry, as it requires that the whole manifold $\mathcal{R}_1$ should remain invariant and not only its submanifold $\mathcal{R}_1^{(1)}$. Since the image of any prolonged solution $j_1\sigma$ lies in $\mathcal{R}_1^{(1)}$, the second definition imposes a stronger condition than the first one.

A less trivial example are the incompressible Navier–Stokes equations. They are not formally integrable; the integrability condition is the well-known Poisson equation for the pressure. But in this special case, the hidden equation has no influence on the symmetry analysis. Thus the classical results on the symmetry group of the Navier–Stokes equations are correct, although their derivations neglect the integrability condition.

The problem disappears, as soon as we consider involutive differential equations. The Cartan–Kähler Theorem implies that any involutive equation is locally solvable. Note that as we are only considering formal power series solutions, the analyticity assumptions of this theorem play no role here.

4 Measuring the Size of Solution Spaces

It was already mentioned in Section 2 that the dimensions of the prolonged symbols $\dim \mathcal{M}_{q+r}$ measure the size of the formal solution space. For involutive equations these dimensions can be easily determined explicitly. Recall that the indices $\beta_q^{(k)}$ count the number of principal derivatives of order q and class k. Similarly, the *Cartan characters*

$$\alpha_q^{(k)} = m\binom{q+n-k-1}{q-1} - \beta_q^{(k)}\,, \quad k = 1, \ldots, n \tag{7}$$

count the parametric derivatives of order q and class k.

Associating with each equation whose principal derivative is of class k its *multiplicative variables* $x^1, \ldots, x^k$, one can show that if we prolong each equation with respect to its multiplicative variables only, we get algebraically independent equations. By (4), no further independent equations of order $q+1$ exist in the case of an involutive symbol. This implies a simple way to compute the indices of the prolonged symbol: $\beta_{q+1}^{(k)} = \sum_{i=k}^{n} \beta_q^{(k)}$

and correspondingly $\alpha_{q+1}^{(k)} = \sum_{i=k}^{n} \alpha_q^{(k)}$. Iteration yields

$$\alpha_{q+r}^{(k)} = \sum_{i=k}^{n} \binom{r+i-k-1}{r-1} \alpha_q^{(i)} . \tag{8}$$

Since $\dim \mathcal{M}_{q+r} = \sum_{k=1}^{n} \alpha_{q+r}^{(k)}$, we see that if (and only if) the symbol $\mathcal{M}_q$ is involutive, we can trivially determine the size of the formal solution space based on the Cartan characters $\alpha_q^{(k)}$. For more details see [19].

A trivial application of these results consists in measuring the size of the symmetry group. As the dimension of a Lie group coincides with the dimension of its Lie algebra, we perform an involution analysis of the determining system $\mathcal{D}_q$. The simplest case arises, if the symmetry group is finite-dimensional. It follows from the discussion above that then all Cartan characters of the involutive completion $\mathcal{D}_{q+r}^{(s)}$ must vanish and the dimension of the group is simply $\dim \mathcal{D}_{q+r}^{(s)}$. In the case of an infinite-dimensional group, the Cartan characters of $\mathcal{D}_{q+r}^{(s)}$ still encode much information about the size and, in particular, permit an easy comparison of different groups. These results are not restricted to Lie point symmetries but may obviously be applied in the same manner to generalised symmetries.

A less trivial application, which we cannot discuss here, is that one may deduce not only the size but also the structure of Lie symmetry algebra by a completely formal analysis. In the case of a finite-dimensional algebra this is rather simple; for infinite-dimensional algebras one needs deeper results from their structure theory. More details can be found in [7, 16].

One interesting feature of this approach is the possibility to formally "subtract" some effects. We will consider here only as a fairly trivial example superposition symmetries. A much more important application concerns gauge symmetries; formal methods allow for a simple analysis of gauge fixing conditions or for an intrinsic definition of the number of degrees of freedom. For lack of space we can only refer to the literature [21, 22].

All linear differential equations possess the superposition symmetry: if $u^\alpha = f^\alpha(x)$ is any solution of $\mathcal{R}_q$, then $X = f^\alpha(x)\partial_{u^\alpha}$ is an infinitesimal symmetry. This reflects the elementary fact that linear combinations of solutions are again solutions; the solution space has the structure of a vector space. Thus one always finds an infinite-dimensional symmetry algebra for linear equations. But usually the other symmetries are of much more interest. Thus one would like to know the size of the remaining algebra.

We illustrate the procedure for the heat equation, although it is trivial to compute its symmetries explicitly. The heat equation $u_t = u_{xx}$ defines a five-dimensional submanifold $\mathcal{R}_2$ in a six-dimensional jet bundle. Its Cartan characters are $\alpha_2^{(2)} = 0$ and $\alpha_2^{(1)} = 2$.

The determining system of the heat equation is also of second order:

$$\mathcal{D}_2 : \left\{ \begin{array}{llll} \tau_u = 0, & \tau_{ux} + \xi_u = 0, & \eta_{uu} - \xi_{ux} = 0, \\ \tau_x = 0, & \tau_{xx} + 2\xi_x - \tau_t = 0, & \eta_{ux} - \frac{1}{2}\xi_{xx} + \frac{1}{2}\xi_t = 0, \\ \tau_{uu} = 0, & \xi_{uu} = 0, & \eta_{xx} - \eta_t = 0. \end{array} \right. \tag{9}$$

Its involutive completion is $\mathcal{D}_3^{(4)}$ and generated by the following equations (plus all their

prolongations up to third order):

$$\mathcal{D}_3^{(4)}: \left\{ \begin{array}{llll} \tau_{ttt} = 0, & \eta_{xx} - \eta_t = 0, & \xi_u = 0, \\ \eta_{uu} = 0, & \xi_{tt} = 0, & \xi_x - \frac{1}{2}\tau_t = 0, \\ \eta_{ux} + \frac{1}{2}\xi_t = 0, & \tau_u = 0, & \tau_x = 0, \\ \eta_{ut} + \frac{1}{4}\tau_{tt} = 0. \end{array} \right. \tag{10}$$

Thus $\mathcal{D}_3^{(4)}$ defines a 13-dimensional submanifold. Its Cartan characters are given by $\alpha_3^{(3)} = \alpha_3^{(2)} = 0$ and $\alpha_3^{(1)} = 2$.

As the last character does not vanish, we have an infinite-dimensional symmetry group. The question is whether an infinite-dimensional subgroup besides the superposition symmetry exists. But from a comparison of the Cartan characters of the determining system and the heat equation, we see that the non-vanishing ones are identical. Hence the infiniteness stems solely from the superposition symmetry.

In order to determine the number of remaining symmetries, we must compare the dimension of the heat equation and of its determining system as submanifolds, as the superposition symmetry manifests itself in the appearance of the equation $\eta_{xx} - \eta_t = 0$ in $\mathcal{D}_3^{(4)}$. As the involutive completion of the determining system is of third order, we must also prolong the heat equation to third order.[2] The number of remaining symmetries is hence given by $\dim \mathcal{D}_3^{(4)} - \dim \mathcal{R}_3$ and one readily computes that this yields in our example $13 - 7 = 6$. Thus we recover the well-known result that besides superposition the heat equation has a six-dimensional symmetry group.

Another simple application of such formal considerations consists in measuring how "special" solutions are that have been obtained with symmetry methods. It is well-known that in the case of partial differential equations, it is not possible to construct the general solution using symmetry theory. Thus one would like to know how large the part of the solution space is that one has constructed with the help of symmetry reductions.

For lack of space, we cannot go here into details. The basic idea is to perform an involution analysis of the combined system consisting of the original equation and the relevant invariant surface condition (see below). An example for the application of this technique to the case of generalised symmetries of a first order differential equation in Cauchy–Kovalevskaya form can be found in [20] (see also [8]).

5 Symmetry Reductions

The fundamental idea behind symmetry reductions is to construct solutions of a given differential equation $\mathcal{R}_q$ that are invariant under the action of a group. Given a symmetry of $\mathcal{R}_q$, one usually rewrites the differential equation in terms of group invariants. Geometrically, this corresponds to "factoring" the submanifold $\mathcal{R}_q$ by the group action. Computationally, this leads to equations in a lower number of independent variables.

For our purposes another approach is more useful. A solution remains invariant under the action of the one-parameter group generated by the vector field $X = \xi^i(\boldsymbol{x}, \boldsymbol{u})\partial_{x^i} +$

[2] In principle, such an adjustment of the order would also have been necessary for the above comparison of the Cartan characters. In our example we could ignore this, as only the last Cartan character did not vanish and thus the Cartan characters are not affected by a prolongation.

$\eta^\alpha(x,u)\partial_{u^\alpha}$, if and only if it is a solution of the corresponding *invariant surface condition*

$$\xi^i(x,u)\frac{\partial u^\alpha}{\partial x^i} = \eta^\alpha(x,u)\,, \qquad \alpha = 1,\dots,m\,. \tag{11}$$

Obviously, this represents a first order system of differential equations. In the sequel we will assume $\xi^n(x,u) \not\equiv 0$, so that we can normalise it to one. As the numbering of the independent variables x^i is arbitrary, this assumption amounts to a transversality condition: not all coefficients ξ^i may vanish identically or, equivalently, the action of the group generated by X must be transversal to the fibration of $\mathcal{E}$.

If the differential equation $\mathcal{R}_q$ has the local representation (1), then any solution of it invariant under the group generated by X must solve the augmented differential equation $\bar{\mathcal{R}}_q$ generated by the equations in (1) and in (11) (and the prolongations of the latter ones up to order q)

$$\bar{\mathcal{R}}_q : \begin{cases} \Phi^\tau(x,u,p) = 0\,, & \tau = 1,\dots,p\,, \\ \xi^i(x,u)p^\alpha_i = \eta^\alpha(x,u)\,, & \alpha = 1,\dots,m\,. \end{cases} \tag{12}$$

Even if $\mathcal{R}_q$ is a scalar equation or a system in Cauchy–Kovalevskaya form, the augmented equation $\bar{\mathcal{R}}_q$ will be an overdetermined equation. Hence the question whether it possesses solutions or equivalently whether $\mathcal{R}_q$ has group-invariant solution is non-trivial.

Following the discussion in Section 3, we assume throughout that $\mathcal{R}_q$ is an involutive equation. It is easy to see that (under the made transversality assumption) the system (11) is involutive, too. Hence integrability conditions in (12) may only arise from cross-derivatives between equations of these two subsystems. In a straightforward calculation one shows [15, 21] that these conditions can be concisely expressed in the form $X^{(q)}\Phi^\tau = 0$.

Assume now that the vector field X generates a one-parameter group of Lie point symmetries of $\mathcal{R}_q$. Then (5) entails that all integrability conditions vanish modulo the equations contained in (1). Hence we always find group-invariant solutions for transversal point symmetries.

Obviously, more possibilities for group-invariant solutions exist. Assume for example that the integrability conditions $X^{(q)}\Phi^\tau = 0$ do not vanish modulo (1) alone, thus X does *not* generate symmetries of $\mathcal{R}_q$ in the sense of Definition 4, but modulo (1) *and* (11). Then the augmented equation $\bar{\mathcal{R}}_q$ is formally integrable and has at least formal power series solutions. Hence $\mathcal{R}_q$ does possess solutions invariant under the group generated by X, although $\mathcal{R}_q$ itself does not remain invariant under the group. In fact, this is nothing but the *non-classical method* of Bluman and Cole [2].

But the non-classical method does not end here — in contrast to what is claimed in most articles on it. Olver and Rosenau [12] were probably the first to point this out introducing the concept of a *weak symmetry group* as a group of transformations such that invariant solutions exist. Later, this observation was again emphasised by Pucci and Saccomandi [15] and one of the present authors [21]. In the language of the formal theory we may formulate it as follows: it is *not* necessary that $\bar{\mathcal{R}}_q$ is formally integrable (as required by Bluman and Cole), but it must be possible to complete $\bar{\mathcal{R}}_q$ to a formally integrable equation without encountering inconsistencies.

We distinguish non-classical symmetries of different *levels*. The symmetries of level 0 are the classical ones and those of Bluman–Cole; here $\bar{\mathcal{R}}_q$ is already formally integrable. The vector field X generates a one-parameter group of symmetries of level s, if s is

the minimal number of projections required to render $\bar{\mathcal{R}}_q$ formally integrable; i. e. the completion leads to an equation of the form $\bar{\mathcal{R}}^{(s)}_{q+r_s}$.

Let us consider a simple example of a non-classical symmetry of level 1 for the heat equation (a number of further examples can be found in [12, 15]). It is generated by the vector field $X = t\partial_t - x\partial_x - 3x^3\partial_u$. The augmented system (12) has now the following form (we omit to write down the prolongations of the second equation to second order)

$$\bar{\mathcal{R}}_2 : \begin{cases} u_t - u_{xx} = 0\,, \\ tu_t - xu_x + 3x^3 = 0\,. \end{cases} \tag{13}$$

$\bar{\mathcal{R}}_2$ is not formally integrable because of the appearance of the integrability condition $u_t = 6x$ (after simplification); hence X generates neither a classical point symmetry nor a non-classical symmetry in the sense of Bluman–Cole. Note that $X^{(2)}(u_t - u_{xx}) = 18x - u_t - 2u_{xx}$ and modulo the heat equation this is indeed equivalent to our integrability condition. As $\bar{\mathcal{R}}^{(1)}_2$ is formally integrable (in fact, even involutive), X generates a non-classical symmetry of level 1. The direct integration of the completed system $\bar{\mathcal{R}}^{(1)}_2$ is trivial and yields the corresponding group-invariant solutions of the heat equation: $u(x,t) = x^3 + 6xt + c$ with an arbitrary constant c.

6 Conclusions

The goal of this article was to exhibit some applications of the formal theory of differential equations within the realm of symmetry methods. We demonstrated the importance of completing a given differential equation before it is analysed, as otherwise one might overlook important symmetries. The formal analysis of the determining systems as studied in Section 4 is probably of interest only in some special situations; symmetry methods lose a lot of their charm, if one is not able to explicitly determine the symmetries.

Any solution of a differential equation $\mathcal{R}_q$ is invariant under some Lie group of point transformations. Thus, in principle, *any* solution can be constructed via non-classical symmetry reductions [12, Theorem 5]. Unfortunately, in practice this seems hardly possible. The determining systems of the non-classical method are non-linear and for higher levels the systems comprise less equations which are, in addition, more complicated. Thus the brute force approach of setting up the determining system and solving it explicitly will almost always fail. Furthermore, it is a priori not clear up to which level one needs non-classical symmetries.

Nevertheless, non-classical symmetries of higher level have already been applied successfully to compute explicit solutions. The method is probably most useful when one has already some ideas what groups might be of particular interest (for example for physical reasons). Then one can impose strong restrictions on the form of the generator X and has a chance to obtain manageable determining systems.

Apparently, the `DETools` library [1] of *MuPAD* represents currently the only available computer algebra software for setting up the determining systems for non-classical symmetries of higher level. It allows for prescribing special ansätze for the symmetry generators. The library also contains procedures for the completion to involution. Detailed descriptions of these packages will appear elsewhere.

The non-classical method may be extended to generalised symmetries. The role of the invariant surface condition is then played by the characteristic of the symmetry generator [12, Lemma 2]. Basically, this means that we augment $\mathcal{R}_q$ by some arbitrary differential equations. Thus the non-classical method for generalised symmetries coincides with what is often called the *method of differential constraints* (comprising also many other reduction methods [10]). Again the practical applicability of the approach is greatly reduced by the unsolved problem of finding reasonable ansätze for the differential constraints.

Our analysis underlines again the correctness of a remark made already in 1986 by Olver and Rosenau [11]:

> The most important conclusion to be drawn from this approach is that the unifying theme behind finding special solutions to partial differential equations is *not*, as is commonly supposed, group theory, but rather the more analytic subject of overdetermined systems of partial differential equations.

The importance of group theory lies in the fact that it permits the systematic construction of "good" differential constraints.

References

[1] J. Belanger, M. Hausdorf, and W.M. Seiler. A *MuPAD* library for differential equations. In V.G. Ghanza, E.W. Mayr, and E.V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing — CASC 2001*, pages 25–42. Springer-Verlag, Berlin, 2001.

[2] G.W. Bluman and J.D. Cole. The general similarity solution of the heat equation. *J. Math. Mech.*, 18:1025–1042, 1969.

[3] G.W. Bluman and S. Kumei. *Symmetries and Differential Equations.* Applied Mathematical Sciences 81. Springer-Verlag, New York, 1989.

[4] M. Hausdorf and W.M. Seiler. An efficient algebraic algorithm for the geometric completion to involution. Preprint Universität Mannheim, 2000.

[5] W. Hereman. Review of symbolic software for the computation of Lie symmetries of differential equations. *Euromath Bull.*, 2:45–82, 1994.

[6] W. Hereman. Symbolic software for Lie symmetry analysis. In N.H. Ibragimov, editor, *CRC Handbook of Lie Group Analysis of Differential Equations, Volume 3: New Trends in Theoretical Development and Computational Methods*, chapter 13. CRC Press, Boca Raton, Florida, 1995.

[7] I.G. Lisle, G.J. Reid, and A. Boulton. Algorithmic determination of structure of infinite Lie pseudogroups of symmetries of PDEs. In A.H.M. Levelt, editor, *Proc. ISSAC '95*, pages 1–6. ACM Press, New York, 1995.

[8] S.V. Meleshko. Differential constraints and one-parameter Lie-Bäcklund transformation groups. *Sov. Math. Dokl.*, 28:37–41, 1983.

[9] P.J. Olver. *Applications of Lie Groups to Differential Equations.* Graduate Texts in Mathematics 107. Springer-Verlag, New York, 1986.

[10] P.J. Olver. Direct reduction and differential constraints. *Proc. Roy. Soc. Ser. A*, 444:509–523, 1994.

[11] P.J. Olver and P. Rosenau. The construction of special solutions to partial differential equations. *Phys. Lett. A*, 114:107–112, 1986.

[12] P.J. Olver and P. Rosenau. Group-invariant solutions of differential equations. *SIAM J. Appl. Math.*, 47:263–278, 1987.

[13] L.V. Ovsiannikov. *Group Analysis of Differential Equations.* Academic Press, New York, 1982.

[14] J.F. Pommaret. *Systems of Partial Differential Equations and Lie Pseudogroups.* Gordon & Breach, London, 1978.

[15] E. Pucci and G. Saccomandi. On the weak symmetry groups of partial differential equations. *J. Math. Anal. Appl.*, 163:588–598, 1992.

[16] G.J. Reid. Finding abstract Lie symmetry algebras of differential equations without integrating determining equations. *Eur. J. Appl. Math.*, 2:319–340, 1991.

[17] J. Schü, W.M. Seiler, and J. Calmet. Algorithmic methods for Lie pseudogroups. In N. Ibragimov, M. Torrisi, and A. Valenti, editors, *Proc. Modern Group Analysis: Advanced Analytical and Computational Methods in Mathematical Physics*, pages 337–344. Kluwer, Dordrecht, 1993.

[18] W.M. Seiler. *Analysis and Application of the Formal Theory of Partial Differential Equations.* PhD thesis, School of Physics and Materials, Lancaster University, 1994.

[19] W.M. Seiler. On the arbitrariness of the general solution of an involutive partial differential equation. *J. Math. Phys.*, 35:486–498, 1994.

[20] W.M. Seiler. Arbitrariness of the general solution and symmetries. *Acta Appl. Math.*, 41:311–322, 1995.

[21] W.M. Seiler. Involution and symmetry reductions. *Math. Comp. Model.*, 25:63–73, 1997.

[22] W.M. Seiler. Involution and constrained dynamics III: Intrinsic degrees of freedom count. *Tech. Mech.*, 20:137–146, 2000.

Chiral Lagrangian Approach to $J/\psi + \pi \to \bar{D} + D^*$ Process: Computer Algebra Calculations

Valery V. Ivanov

Laboratory of Information Technologies,
Joint Institute for Nuclear Research,
141980 Dubna, Russia

The anomalous suppression of J/ψ in high-energy heavy-ion collisions may indicate on the quark-gluon plasma (QGP) formation in laboratory conditions [1]. This effect can be explained both by the QGP formation [2] and by more conventional mechanism based on J/ψ absorption by comoving hadrons [3]. In order to understand the anomalous charmonium suppression, it is important to have better knowledge of the cross sections of the J/ψ absorption by hadrons.

Various approaches have been developed for the evaluation of the charmonium absorption by hadrons [4]. However, all of them give significantly different values for $\sigma_{\pi\psi}$ (and for $\sigma_{\rho\psi}$).

In [5] the $SU(4)$ chiral Lagrangian approach has been applied to the calculation of J/ψ breakup by hadrons impact together with specially chosen hadronic form factors. In this work we present the results of study of the J/ψ absorption by π and ρ mesons applying MATHEMATICA [6] and FORM [7] packages.

References

[1] T. Matsui and H. Satz, *Phys. Lett.* B **178** (1986) 416. Gonin M.*et al*(NA50 Collaboration) *Nucl. Phys.* A **610** (1996) 404; Abreu M. C.*et al* (NA50 Collaboration) *Phys. Lett.* B **450** (1999) 456.

[2] J.P. Blaizot and J.Y. Ollitrault, *Phys. Rev. Lett.* **77** (1996) 1703. C.Y. Wong, *Nucl. Phys.* A **630** (1998)487;

[3] W. Cassin and C.M. Ko, *Phys. Lett.* B **396** (1997) 39; N. Armesto and A. Capella, *Phys. Lett.* B **430** (1998) 23.

[4] C.Y. Wong, E.S. Swanson and T. Barnes, Phys. Rev. **C 62**, 045201 (2000). D. Kharzeev and H. Satz, *Phys. Lett.* B **334** (1994)155. S.G. Matinyan and B. Müller, *Phys. Rev.* C **58** (1998) 2994. K. Haglin, *Phys. Rev.* C **61** 031902 (2000).

[5] V.V. Ivanov, Yu.L. Kalinovsky, D. Blaschke, G. Burau, Chiral Lagrangian approach to the J/ψ breakup cross section, In: *Exporing Quark Matter, Proceedings of the Workshops on "Quark Matter in Astro- and Particle Physics"*, Rostock (Germany),

November 2000 and "Dynamical Aspects of the QCD Phase Transition" Trento (Italy) March 2001, pp. 149-164, Editors G.Burau, D.Blaschke, S.Schmidt, University Rostock, Rostock, 2001.

[6] Mathematica 3.0, Wolfram research.

[7] J.A.M. Vermaseren, FORM, November 21, 1989.

The FABULA System: Implementation on the Java Platform

N.A. Kalinina, D.M. Prudnikov

Novosibirsk State University,
Novosibirsk, Russia;
e-mail: kalinina@ccmath.nsu.ru

The FABULA system [1] intended for solving transformation and optimization problems of Boolean algebras is described.

The new implementation is carried out in language Java of the version JDK 1.2. The main rezult of modernization shows that Java has useful potential for providing possibilities of algebra systems.

The FABULA system being the instrumental one oriented towards theoretical research supporting for decision making problem in conditions of interval and stochastic indefiniteness. In particular the system suppose to be used in logic derivation procedure applied to diagnostic expert system and when researching conditional and fuzzy events algebraic models.

The main function of FABULA system is analitical transformation of symbolic Boolean formulas and solving of optimization problems and systems of linear equations in Boolean algebras.

The input language of FABULA system is an advanced programming one containing means for expressions, functions, equations and other language objects definition and operating with them as well. The language has Pascal-like control constructions and rather wide toolkit for operating with Boolean functions.

The system is designed using traditional translator-systems design methods, the latest computer algebra system constructing methods and methods of the object-oriented programming.

The work is supported by Russian Foundation for Basic Research, grant N 00-07 90322.

References

[1] N.A.Kalinina, V.A.Klimov, M.I.Nechepurenko. Design and realization of computer algebra system for Boolean algebra problems. In: Komputernaya logika, algebra i intellektnoe upravlenie. Problemy analiza i strategicheskoy stabilnosti. V.1, Irkutsk: IrVTs SORAN, 1994, pp.89-100.

Iterative Evaluation of Functions over a Number of Points

V.V. Kislenkov

CMC dept., Moscow State University
e-mail: kislenkov@mail.ru

A problem of evaluation of functions over a number of points is considered. Usual approach, employed in many Computer Algebra applications, performs independent computation at each point. The approaches, described here, reduce the amount of operations and allow to expedite such computations.

Using previously computed value

This approach is based on substitution of an independent computation of function at a new point x_i with a simpler approximate computation, based on earlier computed value of the function at point x_j, where $j < i$ and x_i lies close enough to x_j. Note that computation at x_j is performed independently, using standard system routines and therefore has standard accuracy. This approach allows to expedite computations (the experiments, comparing this approach with independent evaluation of elementary functions $exp(x)$, $ln(x)$, $sin(x)$, $tg(x)$ at all given points, indicate a 1.4-3.3 times speedup on PentiumII). Note that grid doesn't need to be regular as it is required in some other optimization approaches, including application of chains of recurrences.

Configuring computational method w.r.t. the range of argument values

More general approach to evaluation of functions uses configurable methods. By method configuration we assume a set of values of parameters, which adjust the method. When method receives an argument value at first time, it configures itself for a whole range of argument values, which contains the received value. Being computed once, the configuration remains the same during computations until another argument value gets out of the range. At this point method generates a new configuration for a new range, containing the new argument value. Hence, after configuration has been done once, all further computations in the selected range can be performed faster. Moreover, a transition to a new interval of argument values can be performed using data (e.g. polynom coefficients) from previous configuration.

Implementation

The approaches, described here, can be implemented as a library of procedures which allows one to use grid-oriented procedures for evaluation of functions in addition to standard computational means. Besides, such algorithms for computation of elementary functions, being implemented at microcode level and combined with knowledge about architecture of the particular processor, could give a significant speedup for grid computations comparing with traditional independent evaluation.

Computer Algebra Systems for Initial Boundary Value Problem with Parameter

E. Komarova

Russian People's Friendship University,
Mathamatics and physics,
Moscow, Russia;
e-mail: evkomarova@mail.ru

For parametric initial problem with parameter which takes on the values from 0 to 1 it is constructed interpolation procedure of the solution with help of package Maple 6.

Consider the follow problem

$$\varepsilon\dot{x} = F(x,y,t), \qquad \dot{y} = f(x,y,t), \tag{1}$$

$$x(0,\varepsilon) = x^0, \qquad y(0,\varepsilon) = y^0 \tag{2}$$

where $x \in R^n$, $y \in R^m$ with $\varepsilon \in [0,1]$ and $t \in [0,1]$. This work is the development of the results made by M.G. Dmitriev and N.P. Belyaeva [1], [2]. The (1), (2) solution can be found in the following form

$$x_{[1/1]} = \frac{a_0(t) + \tilde{a}_0(\tau) + \varepsilon(a_1(t) + \tilde{a}_1(\tau)) + ... + \varepsilon^n(a_n(t) + \tilde{a}_n(\tau))}{1 + \varepsilon b_1(t) + ... + b_n(t)} \tag{3}$$

here $\tau = \frac{t}{\varepsilon}$ and $\varepsilon \in [0,1]$. The coefficients from (3) - are unknown smooth function of there arguments, which can be found from the asimptotical equality of the expression (3) to the series $\varepsilon \to 0$ and to the $\varepsilon \to 1$.

It was used mathematical package Maple version 6 which let to get formal solution based on computer resources. After some computations we get the systems for $a_0(t), ..., a_n(t)$, $\tilde{a}_0(\tau), ...\tilde{a}_n(\tau)$ and for $b_1(t), ..., b_n(t)$. To simplify it was used the case $n = 2$. Such order is enough to applications.

References

[1] Belyaeva N., Dmitriev M. Asimptotics joining of solution of initial boundary value problem with parameter based on Pade - approximation. // Programmnie sistemi (Russian edition), Moscow, 1999.

[2] Belyaeva N. The elaboration of algorithms for construction trajectory of dynamic systems based on Pade - approximation and asymptotic expansions. // Abstract of Ph.D. thesis (Russian edition), 1999, Yaroslavl State university, 19 p.

[3] Vasil'eva A.B., Butuzov V.F. Asymptotic expanding solutions of singular perturbation equations//(Russian edition), Moscow, Nauka, 1973, 272 p.

Examples of Calculations of the Generators of Differential Ideal by Its Characteristic Set

M.V. Kondratieva

Moscow State University
Faculty of Mechanics and Mathematics, LCM;
e-mail: kondra_m@shade.msu.ru

An algorithm for representation of a radical differential ideal in a ring of differential polynomials as an intersection of a finite number of regular differential ideals was proposed in [2]. This algorithm is called the Rosenfeld-Gröbner algorithm. However, the representation obtained by the above algorithm may not be minimal. This problem could have been solved if we could solve the following Ritt problem: given characteristic sets of two prime differential ideals, determine, whether one of the ideals is a subset of the other one. (see [1]) The latter problem is currently far from being solved. The Ritt problem is a particular case of the problem of finding of the generators for a prime differential ideal: given $P = [\mathcal{A}] : H_{\mathcal{A}}^{\infty}$, where $\mathcal{A}$ is a characteristic set, find elements $A_1, \ldots, A_n$ such that $P = \{A_1, \ldots, A_n\}$.

In this paper we calculate the generators of some prime differential ideals. We suppose that the characteristic set consists of one irreducible ordinary differential equation (note that even in this case the Ritt problem is non-trivial).

Using computer algebra system Maple-V.5, we calculated the generating system for prime differential ideals whose characteristic sets are $y''^2 + 2y$, $y''^2 y' + 2y'^2 + y + 1$, $yy'' + y'^2 + y$. For these ideals the generating system consists of two elements. We suggest a method which allows to obtain the generating system for some prime differential ideals and illustrate this method on ideals whose characteristic sets are $y''^2 + y$, $y''^3 + y''^2 y + y'$ (and others).

Also, a sufficient condition for a differential ideal to be generated by its characteristic set is proved. This condition holds, for example, for ideals whose characteristic sets are $yy'' + y'$ and $yy'' + y' + y$.

References

[1] E.R. Kolchin, Differential Algebra and Algebraic Groups, New York, Academic Press, 1973.

[2] F. Boulier F., D. Lazard, F. Ollivier, M. Petitot, Representation for the radical of a finitely generated differential ideal. *ISSAC'95*, pp. 158–166.

Extraction of "Minimal" Cochain Subcomplexes for Computing Cohomologies of Lie Algebras and Superalgebras

V. V. Kornyak

Laboratory of Information Technologies,
Joint Institute for Nuclear Research,
141980 Dubna, Russia;
email: kornyak@jinr.ru

A new algorithmic approach to computation of cohomologies of Lie (super)algebras is described. The approach is based on splitting the full cochain complex into "minimal" subcomplexes in some non-invariant (i.e., dependent on the choice of coordinates) sense. Such splitting makes the computation much more efficient, because usually the dimensions of cochain spaces included in the full cochain complex are very high. The pseudocode texts of algorithms are presented. The proposed approach is illustrated by two visually graspable examples.

This work was partially supported by the grants RFBF 01-01-00708 and INTAS 99-1222.

1 Introduction

The dual constructions called *homology* and *cohomology* are the main tools for investigation of different mathematical objects and physical models by means of algebraic topology.

Homological or cohomological theory arises if one can construct *chain complex*

$$0 \leftarrow C_0 \xleftarrow{\partial_0} \cdots \xleftarrow{\partial_{k-2}} C_{k-1} \xleftarrow{\partial_{k-1}} C_k \xleftarrow{\partial_k} C_{k+1} \xleftarrow{\partial_{k+1}} \cdots \tag{1}$$

or *cochain complex*

$$0 \rightarrow C^0 \xrightarrow{d^0} \cdots \xrightarrow{d^{k-2}} C^{k-1} \xrightarrow{d^{k-1}} C^k \xrightarrow{d^k} C^{k+1} \xrightarrow{d^{k+1}} \cdots, \tag{2}$$

respectively. Here C_k and C^k are linear spaces (more generally, abelian groups), graded by the integer number k, called *dimension* or *degree* (depending on the context). The elements of the spaces C_k and C^k are called *chains* and *cochains*, respectively.

The linear mappings ∂_k and d^k are called *boundary operator* (or *codifferential*) and *differential* (or *coboundary operator*), correspondingly. The main property of these mappings is "their squares are equal to zero": $\partial_{k-1} \circ \partial_k = d^k \circ d^{k-1} = 0$. Note that the differential increases the dimension k, whereas the boundary operator decreases it.

The elements of the space $Z_k = \text{Ker } \partial_{k-1}$ ($Z^k = \text{Ker } d^k$) are called *cycles* (*cocycles*). The elements of the space $B_k = \text{Im } \partial_k$ ($B^k = \text{Im } d^{k-1}$) are called *boundaries* (*coboundaries*). Note that $B_k \subseteq Z_k$ and $B^k \subseteq Z^k$.

The kth *homology* and *cohomology* are the quotient spaces

$$H_k = Z_k/B_k \equiv \text{Ker } \partial_{k-1}/\text{Im } \partial_k$$

and

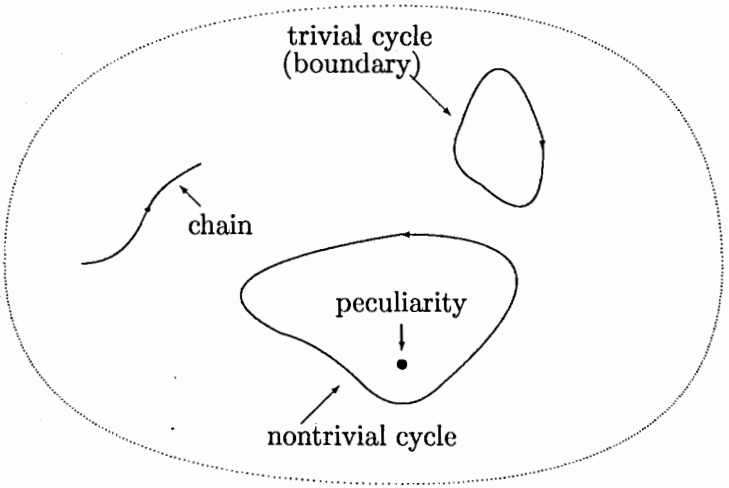
$$H^k = Z^k/B^k \equiv \text{Ker } d^k/\text{Im } d^{k-1},$$

respectively.

There are many homological and cohomological theories designed for investigation of different mathematical structures. The only difference between these theories lies in the constructions of the chain (cochain) spaces and boundary operator (differential). These constructions depend on the underlying mathematical structures.

We list here some examples of cohomological theories mentioning the underlying mathematical structures: the de Rham cohomology (differential forms on manifolds), cohomology of singular cochain comlexes, cohomology of groups, Hochschield (and cyclic) cohomology (associative algebras), Cech cohomology (certain coverings of topological spaces), Spencer cohomology (systems of differential equations).

Figure 1: Chains and cycles



To make the above introduced notions more intuitive, let us illustrate them with the help of the theory of homology of manifolds and the dual theory of the de Rham cohomology. For these theories (see two-dimentional Figure 1) the chains are arbitrary combinations of submanifolds (with or without boundaries). On the set of chains the

structure of abelian group (and even of linear space) is imposed in some appropriate way. The cycles are combinations of submanifolds without boundaries. Some of the cycles contain peculiarities within. These cycles are called *non-trivial* and just they are topologically interesting. To single out these non-trivial cycles up to the trivial ones (boundaries), one should construct the corresponding quotient space. The dual elements to the chains, cochains, are exterior forms of different degrees on the manifold and the differential is the exterior differential acting on these forms. The duality for this pair of theories is provided by the generalized Stokes' theorem stating that the integral of differential of form over the manifold is equal to the integral of form over the boundary of manifold: $\int_M d\omega = \int_{\partial M} \omega$. (This theorem generalizes the well known theorems of the integral calculus given by Newton–Leibniz, Stokes, Gauss-Ostrogradski etc.)

The cohomology of the Lie (super)algebra A in the module X is defined via cochain complex (2) in which (see, e. g., [1]) the cochain spaces $C^k = C^k(A;X)$ consist of superskewsymmetric k-linear mappings $A \times \cdots \times A \to X$, $C^0 = X$ by definition. Superskewsymmetry means symmetry with respect to swapping of two adjacent *odd* cochain arguments and antisymmetry for any other combination of parities for adjacent pair.

The differential d^k takes the form

$$\begin{aligned}(d^k c)(a_0,\ldots,a_k) &= -\sum_{0\le i<j\le k}(-1)^{s(a_i)+s(a_j)+p(a_i)p(a_j)}c([a_i,a_j],a_o,\ldots,\widehat{a_i},\ldots,\widehat{a_j},\ldots,a_k)\\ &\quad -\sum_{0\le i\le k}(-1)^{s(a_i)}a_i c(a_o,\ldots,\widehat{a_i},\ldots,a_k),\end{aligned} \tag{3}$$

where the functions $c(\ldots)$ are elements of cochain spaces; $a_i \in A$; $p(a_i)$ is the parity of a_i; $s(a_i) = i$, if a_i is even element and $s(a_i)$ is equal to the number of even elements in the sequence $a_0,\ldots,a_{i-1}$, if a_i is odd element. In the case of trivial module (i.e., if $ax = 0$ for all $a \in A$ and $x \in X$) one uses as a rule the notation $H^k(A)$.

In papers [2, 3, 4, 5] we presented an algorithm for computation of Lie (super)algebra cohomologies. These papers contain also description of its C implementation and some results obtained with the help of codes designed. This algorithm computes cohomology of Lie (super)algebra A over module X in a straightforward way, i. e., for cochain complex (2) the algorithm constructs the full set of basis monomials forming the space C^k, generates subsequently all basis monomials in the space C^{k+1}, computes the differentials corresponding to these monomials to obtain the set of linear equations determining the space of cocycles

$$Z^k = \mathrm{Ker}\ d^k = \{C^k \mid dC^k = 0\}, \tag{4}$$

constructs the space of coboundaries

$$B^k = \mathrm{Im}\ d^{k-1} = \{C^k \mid C^k = dC^{k-1}\}. \tag{5}$$

Finally, the algorithm constructs the basis elements of quotient space

$$H^k(A;X) = Z^k/B^k. \tag{6}$$

This last step is based on the Gauss elimination procedure.

The main difficulty in computing cohomology comes from the very high dimensions of the spaces C^k: for n-dimensional ordinary Lie algebra and p-dimensional module

$$\dim C^k = p\binom{n}{k},$$

and for $(n|m)$-dimensional Lie superalgebra

$$\dim C^k = p\binom{n}{k} + p\sum_{i=1}^{k}\binom{n}{k-i}\binom{m+i-1}{i}.$$

In many cases it is possibly to extract some easier to handle subcomplexes of the full cochain complex (2). The partition of cochain complex for a graded algebra and module into homogeneous components is a typical example. In many papers (see, e. g., [6, 7, 8]) more special subcomplexes were used successfully to obtain new results in the theory of cohomology of Lie (super)algebras.[1]

The main idea of the new algorithms presented in this paper is to extract the minimal possible (in the sense explained below) subcomplexes from complex (2) and to carry computations within these subcomplexes.

2 Description of Algorithms

In this section we describe shortly two algorithms: **ComputeCohomology** and **SearchCohomology.** The algorithm **ComputeCohomology** is applicable to the cases when it is possible to handle the full space of cochains C^k, i. e., $\dim C^k$ should be moderate enough to keep the basis of C^k in the memory of a computer. This algorithm allows to obtain the full set of basis elements of cohomology for a given cochain complex.

On the other hand, the algorithm **SearchCohomology** allows to obtain partial information about cohomology, i.e., to search for some cohomological classes, in high-dimensional complexes. In some cases this algorithm can be applied even to infinite-dimensional complexes.

The input data for both algorithms should include:

1. The Lie (super)algebra A over the module X. A and X should be presented as sets of basis elements, their commutator and action tables. Our C implementation of the algorithms is able to construct these basis elements and tables for most important algebras and modules.

2. Integer non-negative number k, which is the degree (or dimension) of cohomology.

3. Integer number g giving the grade of cohomology. Most important algebras and modules are graded (if not, one can always prescribe zero grade to all elements of algebra and module) and this grading induces grading in the cohomology.

[1]The main trick consists in imposing some restrictions on the elements of C^k and proving the invariance of these restrictions with respect to the differential.

For the algorithm **SearchCohomology** a positive number n should be additionally inputed. It restricts the number of attempts to find non-trivial cohomological classes.

The output for both algorithms is a set BH_g^k of basis elements of cohomology. This set is full for the algorithm **ComputeCohomology** and partial for the algorithm **SearchCohomology**.

Let $\{\alpha_i\}$ and $\{\xi_\iota\}$ be sets of basis elements of Lie (super)algebra A and module X, correspondingly. The following super skew-symmetric monomials

$$C(\alpha_{i_1},\dots,\alpha_{i_k};\xi_\iota) \equiv C(\alpha_{i_1}) \wedge \cdots \wedge C(\alpha_{i_k}) \otimes \xi_\iota \equiv \alpha'_{i_1} \wedge \cdots \wedge \alpha'_{i_k} \otimes \xi_\iota$$

form basis of the cochain space C^k. Here $i_1 \le \cdots \le i_k$ and α'_i is the dual to α_i element. We use the notation m_g^k for such monomials in both algorithms.

Algoritm: ComputeCohomology

Input: A, Lie (super) algebra; X, module;
k, cohomology degree; g, grade

Output: BH_g^k, set of basis cohomological classes

Local: M_g^k, full set of k-cochain monomials (basis of C_g^k);
s, current subcomplex: $C_{g,s}^{k-1} \xrightarrow{d_{g,s}^{k-1}} C_{g,s}^k \xrightarrow{d_{g,s}^k} C_{g,s}^{k+1}$;
$m_g^k \in M_g^k$, starting monomial for constructing subcomplex s;
$M_{g,s}^k$, set of k-cochain monomials involved in subcomplex s;
$BH_{g,s}^k$, set of basis cohomological classes in subcomplex s

1: $BH_g^k := \emptyset$
2: $M_g^k := GenerateMonomials(A, X, k, g)$
3: **while** $M_g^k \neq \emptyset$ **do**
4: $\quad m_g^k := ChooseMonomial(M_g^k)$
5: $\quad \{s, M_{g,s}^k\} := ConstructSubcomplex(m_g^k)$
6: $\quad M_g^k := M_g^k \setminus M_{g,s}^k$
7: $\quad BH_{g,s}^k := ComputeCohomologyInSubcomplex(s)$
8: $\quad$ **if** $BH_{g,s}^k \neq \emptyset$ **then**
9: $\quad\quad BH_g^k := BH_g^k \cup BH_{g,s}^k$
10: $\quad$ **fi**
11: **od**
12: **return** BH_g^k

Both algorithms construct the following local (working) objects:

1. The subcomplex s constructed by the subalgorithm **ConstructSubcomplex.**

2. The k-cochain monomial m_g^k. This monomial is input for the subalgorithm **ConstructSubcomplex.** The algorithm **ComputeCohomology** takes the monomials m_g^k from the set M_g^k by the subalgorithm **ChooseMonomial,** whereas the algorithm

Algoritm: SearchCohomology

Input: A, Lie (super) algebra; X, module;
k, cohomology degree; g, grade;
$n > 0$, number of generations of subcomplexes

Output: BH_g^k, set of basis cohomological classes

Local: s, current subcomplex: $C_{g,s}^{k-1} \xrightarrow{d_{g,s}^{k-1}} C_{g,s}^{k} \xrightarrow{d_{g,s}^{k}} C_{g,s}^{k+1}$;
m_g^k, starting monomial for constructing subcomplex s;
$M_{g,s}^k$, set of k-cochain monomials involved in subcomplex s;
$BH_{g,s}^k$, set of basis cohomological classes in subcomplex s

1: $BH_g^k := \emptyset$
2: **do**
3: $m_g^k := NewMonomial()$
4: $\{s, M_{g,s}^k\} := ConstructSubcomplex(m_g^k)$
5: $BH_{g,s}^k := ComputeCohomologyInSubcomplex(s)$
6: **if** $BH_{g,s}^k \neq \emptyset$ **and** $BH_{g,s}^k \not\subseteq BH_g^k$ **then**
7: $BH_g^k := BH_g^k \cup BH_{g,s}^k$
8: **fi**
9: $n := n - 1$
10: **od while** $n \neq 0$
11: **return** BH_g^k

SearchCohomology generates m_g^k by the subalgorithm **NewMonomial** (based, e.g., on a random number generator).

3. The set $M_{g,s}^k$ of monomials involved in the current subcomplex s. This set is subtracted from the set M_g^k in the algorithm **ComputeCohomology** (and is one of constituent parts of s in the algorithm **SearchCohomology**).

4. The set $BH_{g,s}^k$ of basis elements of cohomology in the subcomplex s.

For the work of algorithm **ComputeCohomology** the full set M_g^k of basis monomials for the space of k-cochains is generated by the subalgorithm **GenerateMonomials** at the start of computation.

The subalgorithms **ConstructSubcomplex** and **ComputeCohomologyInSubcomplex** are central parts of both algorithms. The subalgorithm **ComputeCohomologyInSubcomplex** computes the set of basis elements of cohomology in a given subcomplex in a standard way as is described in Introduction.

The most important part of our approach is the subalgorithm **ConstructSubcomplex.** Starting with arbitrary monomial m_g^k this subalgorithm constructs the minimal subcomplex s of the form

$$C_{g,s}^{k-1} \xrightarrow{d_{g,s}^{k-1}} C_{g,s}^{k} \xrightarrow{d_{g,s}^{k}} C_{g,s}^{k+1} \tag{7}$$

involving the monomial m_g^k. The subalgorithm **ConstructSubcomplex** is based on formula (3) for the differential d^k. This formula is a set of linear relations connecting k- and $(k+1)$-monomials. The relations containing the monomial m_g^k contains also other monomials and we can add into consideration the relations for these monomials. After such connected relations for k-monomials are all constructed we should carry out an analogous procedure with the differential d^{k-1}. This can lead (in practice, fairly rare) to appearance of new k-monomials and then we should return to the procedure for the differential d^k. This cyclic process terminates if any given basis element α_i of algebra appears only in a finite number of the right hand parts of commutators and similar property (with respect to action of algebra on module) holds for basis elements of module ξ_ι. Note that the subalgorithm **ConstructSubcomplex** can construct finite-dimensional subcomplexes in some cases when grading does not allow finite-dimensional subcomplexes, e. g., when computing cohomology in adjoint module for infinite-dimensional graded Lie (super)algebras. Note also that the subcomplexes constructed by the subalgorithm **ConstructSubcomplex** are not invariant, i. e., they depend on the choice of bases for algebra and module, and they, apparently, not always can be extended to full sequence (2) in such a way that part (7) remains unchanged, but for our purposes this is not important.

3 Examples: Computation of $H^3(\mathrm{Poincare}(1,3))$ and $H^3(\mathrm{SH}(0|4))$

Both examples of computations in this section are carried out in zero grade complexes and we shall omit subscripts $g=0$ everywhere below. As the first example we consider computation of third cohomology in the trivial module for 10-dimensional Poincaré algebra. The basis of this algebra contains the following generators of: translations P_0, P_1, P_2, P_3; rotations R_1, R_2, R_3; Lorentz transformations L_1, L_2, L_3. The commutator table for the Poincaré algebra takes the form:

$$[P_\mu, P_\nu] = 0, \quad \mu,\nu = 0,\ldots,3;$$

$$[P_0, L_i] = P_i, \quad [P_i, L_j] = \delta_{ij} P_0, \quad [P_i, R_j] = \epsilon_{ijk} P_k,$$

$$[R_i, R_j] = \epsilon_{ijk} R_k, \quad [R_i, L_j] = \epsilon_{ijk} L_k, \quad [L_i, L_j] = -\epsilon_{ijk} R_k, \quad i,j,k = 1,\ldots,3.$$

Here ϵ_{ijk} is the permutation symbol with $\epsilon_{123} = 1$.

We consider the part

$$C^2 \xrightarrow{d^2} C^3 \xrightarrow{d^3} C^4 \tag{8}$$

of cochain complex (2). The spaces of cochains have the following dimensions

$$\dim C^2 = 45, \quad \dim C^3 = 120, \quad \dim C^4 = 210.$$

The system of equations for 3-cocycles $d^3(C^3) = 0$ contains initially 187 equations for 120 variables. The Gauss elimination process shows that the rank of this system is equal to 83, i. e., the 3-cocycles form 37-dimensional subspace in 120-dimensional space C^3. The expressions $d^2(C^2)$ determine parametrically 35-dimensional subspace of coboundaries in

the subspace of 3-cocycles, i. e., we get finally the cohomology as a 2-dimensional quotient space of these subspaces.

The algorithm **ComputeCohomology** divides complex (8) into 28 subcomplexes as shown in Table 1. As one can see from Table 1 nontrivial cohomological classes arise

Table 1: Subcomplex structure for complex (8)

s	Starting monomial	$\dim C^3_s$	$\dim Z^3_s$	$\dim B^3_s$	$\dim H^3_s$	BH^3_s
1	$c(P_0, P_1, P_2)$	1	0	0	0	
2	$c(P_0, P_1, P_3)$	1	0	0	0	
3	$c(P_0, P_2, P_3)$	1	0	0	0	
4	$c(P_1, P_2, P_3)$	1	0	0	0	
5	$c(L_1, L_2, R_1)$	2	1	1	0	
6	$c(L_1, L_2, R_2)$	2	1	1	0	
7	$c(L_1, L_3, R_1)$	2	1	1	0	
8	$c(L_1, R_1, R_2)$	2	1	1	0	
9	$c(L_1, R_1, R_3)$	2	1	1	0	
10	$c(L_2, R_1, R_2)$	2	1	1	0	
11	$c(P_0, P_1, L_2)$	4	1	1	0	
12	$c(P_0, P_1, L_3)$	4	1	1	0	
13	$c(P_0, P_1, R_2)$	4	1	1	0	
14	$c(P_0, P_1, R_3)$	4	1	1	0	
15	$c(P_0, P_2, L_3)$	4	1	1	0	
16	$c(P_0, P_2, R_3)$	4	1	1	0	
17	$c(L_1, L_2, L_3)$	4	4	3	1	h^3_{17}
18	$c(L_1, L_2, R_3)$	4	1	0	1	h^3_{18}
19	$c(P_0, L_1, R_2)$	6	2	2	0	
20	$c(P_0, L_1, R_3)$	6	2	2	0	
21	$c(P_0, L_2, R_3)$	6	2	2	0	
22	$c(P_1, L_2, R_3)$	6	2	2	0	
23	$c(P_0, P_1, L_1)$	6	0	0	0	
24	$c(P_0, P_1, R_1)$	6	0	0	0	
25	$c(P_0, L_1, L_2)$	9	3	3	0	
26	$c(P_0, L_1, L_3)$	9	3	3	0	
27	$c(P_0, L_1, R_1)$	9	3	3	0	
28	$c(P_0, L_2, L_3)$	9	3	3	0	

only in subcomplexes 17 and 18, generated by the function **ConstructSubcomplex** starting with the monomials $c(L_1, L_2, L_3)$ and $c(L_1, L_2, R_3)$, correspondingly. Dimensions of subspaces C^3_{17} and C^3_{18} are equal to 4.

The set of equations for cocycles $d^3(C^3_{17}) = 0$ is empty and expressions for coboundaries

$d^2(C^2_{17})$ take the form

$$\begin{aligned} c(L_1,L_2,L_3) &= -c(L_1,R_1)-c(L_2,R_2)-c(L_3,R_3),\\ c(L_1,R_2,R_3) &= c(L_1,R_1)-c(L_2,R_2)-c(L_3,R_3),\\ c(L_2,R_1,R_3) &= c(L_1,R_1)-c(L_2,R_2)+c(L_3,R_3),\\ c(L_3,R_1,R_2) &= -c(L_1,R_1)-c(L_2,R_2)+c(L_3,R_3), \end{aligned}$$

where 2-cochains should be treated as arbitrary parameters. These relations determine 3-dimensional subspace and basis element h^3_{17} of cohomology for subcomplex 17 can be expressed in the form $h^3_{17} = c(L_1,L_2,L_3) - c(L_1,R_2,R_3) + c(L_2,R_1,R_3) - c(L_3,R_1,R_2)$.

For subcomplex 18 equations $d^3(C^3_{18}) = 0$ determining cocycles take the form

$$\begin{aligned} c(L_1,L_2,R_1,R_2) &= -c(L_1,L_2,R_3)-c(L_1,L_3,R_2)+c(L_2,L_3,R_1)+c(R_1,R_2,R_3)=0,\\ c(L_1,L_3,R_1,R_3) &= c(L_1,L_2,R_3)+c(L_1,L_3,R_2)+c(L_2,L_3,R_1)+c(R_1,R_2,R_3)=0,\\ c(L_2,L_3,R_2,R_3) &= c(L_1,L_2,R_3)-c(L_1,L_3,R_2)-c(L_2,L_3,R_1)+c(R_1,R_2,R_3)=0. \end{aligned}$$

This system of equations determines 1-dimensional subspace. Expressions for coboundaries $d^2(C^2_{18})$ determine 0-dimensional subspace taking the form

$$c(L_1,L_2,R_3) = c(L_1,L_3,R_2) = c(L_2,L_3,R_1) = c(R_1,R_2,R_3) = 0,$$

and basis element h^3_{18} of cohomology can be chosen as $h^3_{18} = c(R_1,R_2,R_3)$ or as linear combination

$$h^3_{18} = \alpha c(R_1,R_2,R_3) + \beta c(L_1,L_2,R_3) + \gamma c(L_1,L_3,R_2) + \delta c(L_2,L_3,R_1),$$

with $\alpha-\beta+\gamma-\delta \neq 0$.

Let us consider also an example of computation that is typical for Lie superalgebras. Computation of the third cohomology in the trivial module and in zero grading for special Hamiltonian algebra SH(0|4) is a task of approximately the same complexity as in the above example with the Poincaré algebra. The (6|8)-dimensional superalgebra SH(0|4) has the following even $E_1 = \theta_1\theta_2$, $E_2 = \theta_1\theta_3$, $E_3 = \theta_2\theta_3$, $E_4 = \theta_1\theta_4$, $E_5 = \theta_2\theta_4$, $E_6 = \theta_3\theta_4$ and odd $O_1 = \theta_1$, $O_2 = \theta_2$, $O_3 = \theta_3$, $O_4 = \theta_4$, $O_5 = \theta_1\theta_2\theta_3$, $O_6 = \theta_1\theta_2\theta_4$, $O_7 = \theta_1\theta_3\theta_4$, $O_8 = \theta_2\theta_3\theta_4$ basis elements. Here θ_1, θ_2, θ_3, θ_4 are Grassmann variables and Lie product coincides with Poisson bracket $\{\cdot,\cdot\}_{Pb}$. For this particular case $\{f,g\}_{Pb} = -(-1)^{p(f)}\sum_{i=1}^4 \partial f/\partial\theta_i \partial g/\partial\theta_i$, where f and g are functions of $\theta_1,\dots\theta_4$ and $p(f)$ is parity of function f. In the case of Lie superalgebra cochain complex can be split into two subcomplexes, even and odd. In our example only even subspace of C^3 contains nontrivial cocycle. The spaces of even subcomplex have the following dimensions

$$\dim C^2 = 31, \qquad \dim C^3 = 116, \qquad \dim C^4 = 355.$$

The initial number of determining equations for 3-cocycles is equal to 351. This system of equations determines 25-dimensional space of cocycles. The expressions for coboundaries describe 24-dimensional subspace, hence the cohomology is 1-dimensional.

The algorithm **ComputeCohomology** generates eight subcomplexes. Six of them have the following characteristics:

$$\dim C^3_s = 14, \quad \dim Z^3_s = \dim B^3_s = 3, \quad \dim H^3_s = 0.$$

There are also two subcomplexes with $\dim C^3_s = 16$, one of them has $\dim Z^3_s = \dim B^3_s = \dim H^3_s = 0$. And finally, the remaining subcomplex (with 30 equations for 3-cocycles) contains 1-dimensional cohomology: $\dim Z^3_s = 7, \dim B^3_s = 6, \dim H^3_s = 1$. We do not present here the basis element of this cohomology explicitly because its expression is rather long.

4 Conclusion

As one can see from the two above, chosen at random, examples, cochain complexes can be effectively divided into smaller subcomplexes. One can show that generally the efficiency of such division grows with increase of cochain degree k, i. e., the dimensions of subcomplexes grow slower than the dimension of full complex. We are implementing the algorithms **ComputeCohomology** and **SearchCohomology** as different regimes of one C program.

References

[1] Fuks, D.B.: *Cohomology of Infinite Dimensional Lie Algebras.* Consultants Bureau, New York, 1987.

[2] Kornyak, V.V.: A program for Computing Cohomologies of Lie Superalgebras of Vector Fields. *Zapiski nauchnyh seminarov POMI. St.Petersburg.* **258** (1999) 148-160.

[3] Kornyak, V.V.: Cohomology of Lie Superalgebras of Hamiltonian Vector Fields: Computer Analysis. In: *Computer Algebra in Scientific Computing / CASC'99*, V.G.Ganzha, E.W.Mayr and E.V.Vorozhtsov (Eds.), Springer-Verlag, Berlin, 1999, pp. 241-249; math.SC/9906046.

[4] Kornyak, V.V.: Computation of Cohomology of Lie Superalgebras of Vector Fields. *Int. J. of Mod. Phys. C.* **11** (2000) 397-414; arXiv: math.SC/0002210.

[5] Kornyak, V.V.: Computation of Cohomology of Lie Superalgebras: Algorithm and Implementation. *Russian Journal for Computer Science ("Programmirovanie").* **3** (2001) 46-50 (in Russian).

[6] Gel'fand, I.M., Kalinin, D.I., Fuks, D.B.: On Cohomology of Lie Algebra of Hamiltonian Formal Vector Fields. *Funkts. Anal. Prilozhen.* **6** (1972) 25-29 (in Russian).

[7] Perchik, J.: Cohomology of Hamiltonian and related formal vector fields Lie algebras. *Topology.* **15**, 4 (1976) 395-404.

[8] Guillemin, V.M., Shnider, S.D.: Some stable results on the cohomology of classical infinite dimensional Lie algebras. *Trans. Amer. Math. Soc.* **179** (1973) 275-280.

Computation of the Characteristic Sets for the Euler Equations for Different Rankings

N.A. Makarevich

Moscow State University,
Faculty of Mechanics and Mathematics, MSU, 119899
e-mail: makarevich_n@mail.ru

The aim of this study is to investigate the dependence of the characteristic set of the system of the Euler equations on the number of the variables and on the ranking. This system is of interest for computation due to a number of reasons. It is one of treatable systems of nonlinear differential equations and the corresponding differential ideal is prime.

At the same time, the results obtained for various rankings are very different. The computations were made by the Rosenfeld—Gröbner algorithm implemented in CAS MAPLE. This algorithm represents the least radical differential ideal containing initial ideal set by a finite number of differential polynomials as a finite intersection of regular differential ideals J_i:

$$\{\Sigma\} = J_1 \cap \cdots \cap J_s$$

1 Some definitions and notations

Let K be a differential field of characteristic zero with a set of derivations $\Delta = \{d_1, \ldots, d_m\}$. Consider the differential ring $R = K\{y_1, \ldots, y_n\}$. The order of an operator $\theta = {d_1}^{i_1} \ldots {d_m}^{i_m}$ is equal to the sum of the exponents i_j. We call the y_i by *letters* and the θy_i by *derivatives*. The degree of a polynomial $f \in R$ or $\deg f$ is the degree of f as a polynomial in an infinite number of variables θy_i and its order $\operatorname{ord} f$ is the maximal order of the derivatives that are present in f.

Let I be an ideal and T be a multiplicatively stable family of $R = K\{y_1, \ldots, y_n\}$. The symbol $I : T$ stands for the ideal of all elements p of R such as for some $t \in T$ the element tp belongs to I.

A differential ideal I of $R = K\{y_1, \ldots, y_n\}$, $I \neq R$ is said to be prime if for any $p_1, p_2 \in R$ the inclusion $p_1 p_2 \in I$ implies either $p_1 \in I$ or $p_2 \in I$.

A differential ideal I of $R = K\{y_1, \ldots, y_n\}$ is called radical if the inclusion $p^n \in I$ ($p \in R$, n is a positive integer) implies the inclusion $p \in I$. The least radical differential ideal containing a finite set $\Sigma \subset R$ is denoted by $\{\Sigma\}$.

A total order $\mathcal{R}$ on the set of the derivatives y_j is said to be a ranking if the following conditions hold:

1. $d_i\theta y_j >_{\mathcal{R}} \theta y_j$ for all derivations d_i, all operators θ and all letters y_j;

2. $\theta_1 y_i >_{\mathcal{R}} \theta_2 y_j \Rightarrow d_k\theta_1 y_i >_{\mathcal{R}} d_k\theta_2 y_j$ for all derivations d_k, all operators θ_1, θ_2 and all letters y_i, y_j.

Let $p \in K\{y_1, \ldots, y_n\}$, $p \notin K$ and $\mathcal{R}$ be a ranking. The leader u of p is defined as the highest derivative with respect to the ranking $\mathcal{R}$ that appears in p. Let d denote the degree of u in p. The initial I_p of p is the coefficient of u^d in p. The separant S_p is the partial derivative of p with respect to u.

Let p and q be two polynomials, u be the leader of p, and d be its degree in p. The polynomial q is said to be partially reduced with respect to p if no proper derivative of u appears in q. The polynomial q is said to be reduced with respect to p if q is partially reduced with respect to p and its degree in u is less than d.

A set of polynomials is said to be autoreduced if each element of the set is reduced with respect to every other its element. Let A be an autoreduced set. Let H_A be the product of all initials and separants of A. The symbol H_A^∞ denotes the set of all powers of the element H_A.

An autoreduced subset C of a set A of polynomials is called a characteristic set of A if A contains no non-zero elements reduced with respect to C. A characteristic set C of an ideal I reduces to zero all elements of I. If the ideal is prime, then C reduces to zero only the elements of I and we have $I = [C] : H_C^\infty$.

Let p and q be two polynomials in an autoreduced set A, whose leaders $\theta_1 y_i$ and $\theta_2 y_i$ are derivatives of one and the same letter y_i. Let θ be the operator of minimal order, ϕ_1 and ϕ_2 be two derivation operators such that $\phi_1\theta_1 = \phi_2\theta_2 = \theta$. The Δ-polynomial between p and q is the polynomial $\Delta = S_q\phi_1 p - S_p\phi_2 q$. A set A is said to be coherent if it reduces to zero all Δ-polynomials between any two elements of A.

Let Σ be a differential polynomial system of equations and inequations $\{p_1 = 0, \ldots, p_k = 0, q_1 \neq 0, \ldots, q_l \neq 0\}$, where p_i, $q_j \in K\{y_1, \ldots, y_n\}$ for $i = 1 \ldots k$ and $j = 1 \ldots l$. A differential model of Σ is a morphism of the differential K-algebras $K\{y_1, \ldots, y_n\} \to L$ into a differential field L that annihilates the equations but not the inequations of Σ.

A system of differential equations and inequations is said to be regular with respect to a ranking $\mathcal{R}$, if the set of its equations is autoreduced and coherent, the initial and separant of each equation appear among the inequations and if its other inequations are partially reduced with respect to the equations:

$$\Omega \begin{cases} p_1 = 0 & \\ \quad\vdots & A = \{p_1, \ldots, p_s\} \text{ is autoreduced and coherent} \\ p_{s'} = 0 & \\ I_1 \neq 0 & \\ \quad\vdots & \text{the initials and separants of } p_i \\ S_s \neq 0 & \\ q \neq 0 & q \text{ is partially reduced w.r.t. } A \end{cases}$$

A differential ideal J is said to be regular if there exists a regular system Ω such as $J = [A] : H_\Omega^\infty$, where A is the set of equations of Ω.

The input data for the Rosenfeld–Gröbner package are a system of differential equations and inequations Σ and a ranking $\mathcal{R}$. Its output is a finite family (Ω_i) of consistent (with models) regular systems whose differential models form a partition of the differential models of Σ. Let A_i denote the set of equations for each Ω_i and H_{Ω_i} be the product of all inequations of Ω_i. Denoting by $H_{\Omega_i}^{\infty}$ the set of all powers of the H_{Ω_i}, we can represent the result of the run of the program as

$$\{\Sigma\} = [A_1] : H_{\Omega_1}^{\infty} \cap \cdots \cap [A_s] : H_{\Omega_s}^{\infty}$$

2 Computations

In the case of three space variables, the system of the Euler equations is:

$$p1 := v1_{x1} + v2_{x2} + v3_{x3}$$

$$p2 := v1_t + v1\ v1_{x1} + v2\ v1_{x2} + v3\ v1_{x3} + p_{x1}$$

$$p3 := v2_t + v1\ v2_{x1} + v2\ v2_{x2} + v3\ v2_{x3} + p_{x2}$$

$$p4 := v3_t + v1\ v3_{x1} + v2\ v3_{x2} + v3\ v3_{x3} + p_{x3}$$

Theorem 1. *If A is a coherent autoreduced set such that the ideal $(A) : H_A^{\infty}$ of $K\{y_1, \ldots, y_n\}$ is prime and contains no nonzero elements reduced with respect to A, then A is a characteristic set of a prime Δ-ideal of the ring $K\{y_1, \ldots, y_n\}$.*

The proof of this theorem can be found in [2].

For $t > x1 > x2 > x3$, $v1 > v2 > v3 > p$, and the ordering of total degree then lexicographic, the set A

$$p1 := v1_{x1} + v2_{x2} + v3_{x3}$$

$$p2 := v1_t + v2\ v1_{x2} + v3\ v1_{x3} + p_{x1} - v1\ v2_{x2} - v1\ v3_{x3}$$

$$p3 := v2_t + v1\ v2_{x1} + v2\ v2_{x2} + v3\ v2_{x3} + p_{x2}$$

$$p4 := v3_t + v1\ v3_{x1} + v2\ v3_{x2} + v3\ v3_{x3} + p_{x3}$$

$$p5 := p_{x1,x1} + p_{x2,x2} + p_{x3,x3} + v2_{x1}\ v1_{x2} + 2\,v3_{x1}\ v1_{x3} + 2\,{v2_{x2}}^2$$

$$+2\,v3_{x2}\ v2_{x3} + 2\,{v3_{x3}}^2 + 2\,v2_{x2}\ v3_{x3}$$

satisfies the conditions of the theorem; hence, the ideal corresponding to the system of the Euler equations is prime.

For some rankings, the program computes not only the required system whose set of equations forms the characteristic set of the ideal, but also some redundant components. In this case and in some situations when the direct computations fail, the program allows one to compute this system using the results obtained for another ranking.

So, the possible situations are the following:

1. the characteristic set can be computed directly;

2. the program stops the computation because the arguments of some intermediate functions become too large, but the characteristic set can be determined using the result obtained for another ranking;

3. both attempts of computing the characteristic set fail.

When the initials and separants of a charasteristic set belong to the field of constant, = [A] and in this case A is an analogue of the Gröbner basis, in particular, the elements A and their derivatives generate the ideal.

Three cases of the Euler equations are considered: with one, two, and three space ariables. It is natural that the first case is the simplest one. The initial system is:

$$v_x, \quad v_t + vv_x + p_x.$$

After reducing the second equation with respect to the first one, the system becomes near and the program computes the Gröbner basis of the system. Due to its special rm, the basis can be obtained for any ranking and it is equal either to

$$v_x, \quad v_t + p_x, \quad p_{xx}$$

r to

$$v_x, \quad p_x + v_t.$$

The cases of two and three space variables are much more complicated. For two space ariables, the initial system is:

$$p1 := v1_t + v1\ v1_x + v2\ v1_y + p_x$$

$$p2 := v2_t + v1\ v2_x + v2\ v2_y + p_y$$

$$p3 := v1_x + v2_y$$

The system is symmetric with respect to the space variables; therefore, we can exclude ome orders on the letters and derivations from consideration. For the order of total degree hen lexicographic, the results are obtained directly. Nevertheless, they depend on the rder of derivatives. If $t > x > y$, then in both cases $p > v1 > v2$ and $v1 > v2 > p$ edundant components appear.

If $x > y > t$, $v1 > v2 > p$, then we obtain

$$P := [regular, regular, regular]$$

$$[[v1\ v2\ p_{x,x} + v1\ v2\ p_{y,y} + 2\,v2_t\,v1_t + 2\,v2_t\,p_x - 2\,v2_t\,v2_y\,v1 + 2\,v2\ v2_y\,v1_t + 2\,v2\ v2_y\,p_x + 2\,p_y\,v1_t + 2\,p_y\,p_x - 2\,p_y\,v2_y\,v1,\ v1_x + v2_y,\ v1_t + v2\ v1_y + p_x - v2_y\,v1,\ v2_t + v1\ v2_x + v2\ v2_y + p_y],\ [p_{x,x},\ v1_x,\ v1_t + p_x,\ p_y,\ v2],\ [p_{y,y},\ v2_y,\ v2_t + p_y,\ p_x,\ v1]]$$

However, if $t > x > y$ and $v1 > v2 > p$, then we have only one component

$$[[2\,v2_x\,v1_y + p_{x,x} + 2\,{v2_y}^2 + p_{y,y},\ v1_t + v2\,v1_y + p_x - v2_y\,v1,\ v1_x + v2_y,$$
$$v2_t + v1\,v2_x + v2\,v2_y + p_y]]$$

For the ranking where the derivatives $d_1^{i_1} \ldots d_n^{i_n} y_j$ are ordered in the lexicographical order of the vectors $(i_1, \ldots, i_n, j)$, the situation is the same. If $t > x > y$, then we directly obtain the characteristic sets which are quite simple. If $x > y > t$, then we obtain redundant components.

1. The case $x > y > t$ and $p > v1 > v2$:

$$J := [regular,\ regular]$$

$$[[v1_t + v2\,v1_y + p_x - v2_y\,v1,\ v1_x + v2_y,\ v2_t + v1\,v2_x + v2\,v2_y + p_y],$$
$$[v2_{x,t},\ p_x,\ v2_t + p_y,\ v2_y,\ v1]]$$

2. The case $t > x > y$ and $p > v1 > v2$:

$$[[v1_t + v2\,v1_y + p_x - v2_y\,v1,\ v2_t + v1\,v2_x + v2\,v2_y + p_y,$$
$$2\,v2_x\,v1_y + p_{x,x} + 2\,{v2_y}^2 + p_{y,y},\ v1_x + v2_y]]$$

The worst case is the case of the lexicographical order. Neither direct computations, nor computations using previous results can handle this system.

In the case of three space variables, one of the simplest sets corresponds to the order of total degree then lexicographic, $t > x1 > x2 > x3$ and $v1 > v2 > v3 > p$. This case was considered earlier. The output of the program is

$$[[2\,v2_{x1}\,v1_{x2} + 2\,v3_{x1}\,v1_{x3} + p_{x1,x1} + 2\,{v2_{x2}}^2 + 2\,v3_{x2}\,v2_{x3} + p_{x2,x2} + 2\,{v3_{x3}}^2$$
$$+p_{x3,x3} + 2\,v2_{x2}\,v3_{x3},\ v1_t + v2\,v1_{x2} + v3\,v1_{x3} + p_{x1} - v1\,v2_{x2} - v1\,v3_{x3},$$
$$v1_{x1} + v2_{x2} + v3_{x3},\ v2_t + v1\,v2_{x1} + v2\,v2_{x2} + v3\,v2_{x3} + p_{x2},$$
$$v3_t + v1\,v3_{x1} + v2\,v3_{x2} + v3\,v3_{x3} + p_{x3}]]$$

Unlike the case of two space variables, if we change the order of derivatives for $x1 > x2 > x3 > t$, then the characteristic sets cannot be obtained directly, but can be computed with the help of the previous results:

$$[[2\,v2\,v3\,v1_{x3}\,v2_{x2} - 2\,v1\,v2_t\,v2_{x2} - 2\,v2\,v3\,v3_{x3}\,v1_{x3} + 2\,v2\,v1_t\,v2_{x2}$$
$$- 2\,v1\,v3\,v2_{x3}\,v3_{x3} + 2\,v2\,v1\,v3_{x2}\,v2_{x3} - 2\,v1\,v3\,v2_{x3}\,v2_{x2}$$
$$- 2\,v1\,p_{x2}\,v3_{x3} + 2\,v2\,v1\,{v3_{x3}}^2 + v2\,v1\,p_{x2,x2} + 2\,{v3}^2\,v2_{x3}\,v1_{x3}$$
$$+ 2\,v3\,v2_{x3}\,p_{x1} + 2\,p_{x2}\,v3\,v1_{x3} + 2\,v2_t\,v3\,v1_{x3} - 2\,v1\,p_{x2}\,v2_{x2}$$
$$+ v2\,v1\,p_{x3,x3} - 2\,v1\,v2_t\,v3_{x3} - 2\,v2\,v1_{x3}\,p_{x3} + 2\,v3\,v2_{x3}\,v1_t$$
$$+ 2\,v2\,v2_{x2}\,p_{x1} + v2\,v1\,p_{x1,x1} - 2\,v2\,v3_t\,v1_{x3} - 2\,{v2}^2\,v3_{x2}\,v1_{x3} + 2\,p_{x2}\,p_{x1}$$
$$+ 2\,p_{x2}\,v1_t + 2\,v2_t\,p_{x1} + 2\,v2_t\,v1_t,\ v1_{x1} + v2_{x2} + v3_{x3},$$
$$v1_t + v2\,v1_{x2} + v3\,v1_{x3} + p_{x1} - v1\,v2_{x2} - v1\,v3_{x3},$$
$$v2_t + v1\,v2_{x1} + v2\,v2_{x2} + v3\,v2_{x3} + p_{x2},$$
$$v3_t + v1\,v3_{x1} + v2\,v3_{x2} + v3\,v3_{x3} + p_{x3}]]$$

The result in the case, where $p > v1 > v2 > v3$ and $t > x1 > x2 > x3$, just as in the first case, can be obtained directly; however, its size is quite large:

$$
\begin{aligned}
&[[-v2_{x1}\, v3_{x1,x3} - v2_{x1}\, v1_{x2,x2} + v1\, v2_{x1,x2,x2} - v2_{x2}\, v2_{x1,x1} - v2_{x2}\, v2_{x2,x2} \\
&\quad - v2_{x2}\, v3_{x2,x3} \\
&\quad + v1\, v2_{x1,x1,x1} + v1\, v3_{x1,x2,x3} + v3\, v2_{x1,x1,x3} + v1_{x2}\, v3_{x1,x3} + 2\, v3_{x1}\, v2_{x1,x3} \\
&\quad + v3_{x1,x1}\, v2_{x3} - 2\, v2_{x1,x1}\, v3_{x3} + v2_{x1}\, v2_{x1,x2} - v3_{x1}\, v1_{x2,x3} - 2\, v3_{x3}\, v2_{x2,x2} \\
&\quad + v2\, v2_{x2,x2,x2} + v3_{x2}\, v2_{x2,x3} + v2\, v3_{x2,x2,x3} + v3\, v2_{x2,x2,x3} + v3\, v3_{x2,x3,x3} \\
&\quad + v2\, v2_{x1,x1,x2} - 2\, v3_{x3}\, v3_{x2,x3} - v3_{x1,x2}\, v1_{x3} + v3_{x2}\, v3_{x3,x3} + v2_{t,x2,x2} \\
&\quad + v2_{t,x1,x1} \\
&\quad + v3_{t,x2,x3}, v3_{x1}\, v1_{x2,x2} + v2\, v3_{x2,x3,x3} + v3_{x1}\, v3_{x1,x3} + v3_{t,x2,x2} + v3_{t,x1,x1} \\
&\quad + v3_{t,x3,x3} + v3_{x2,x2}\, v3_{x3} - v3_{x2}\, v2_{x3,x3} - v3_{x3}\, v3_{x3,x3} - v3_{x3}\, v3_{x1,x1} \\
&\quad + v3\, v3_{x3,x3,x3} \\
&\quad - 4\, v2_{x2,x3}\, v2_{x2} - 2\, v2_{x2}\, v3_{x3,x3} + v2_{x1,x1}\, v3_{x2} - v3_{x1}\, v2_{x1,x2} - 2\, v3_{x1,x1}\, v2_{x2} \\
&\quad - v3_{x1}\, v1_{x3,x3} - 2\, v2_{x1}\, v1_{x2,x3} + v2\, v3_{x1,x1,x2} + 2\, v2_{x1}\, v3_{x1,x2} - 2\, v2_{x1,x3}\, v1_{x2} \\
&\quad + v1\, v3_{x1,x3,x3} + v1\, v3_{x1,x1,x1} + v3\, v3_{x1,x1,x3} + 2\, v2_{x2}\, v3_{x2,x2} \\
&\quad + v2\, v3_{x2,x2,x2} - 2\, v3_{x3}\, v2_{x2,x3} + v1\, v3_{x1,x2,x2} + v3\, v3_{x2,x2,x3} + 2\, v3_{x2}\, v3_{x2,x3} \\
&\quad + 2\, v3_{x1,x2}\, v1_{x2} + v2_{x2,x2}\, v3_{x2}, v1_{t,x2} + v2\, v1_{x2,x2} + v3_{x2}\, v1_{x3} + v3\, v1_{x2,x3} \\
&\quad - v1\, v2_{x2,x2} - v1_{x2}\, v3_{x3} - v1\, v3_{x2,x3} - v2_{t,x1} - v1\, v2_{x1,x1} - v2\, v2_{x1,x2} \\
&\quad - v3_{x1}\, v2_{x3} - v3\, v2_{x1,x3} + v2_{x1}\, v3_{x3}, v1_{t,x3} + v2_{x3}\, v1_{x2} + v2\, v1_{x2,x3} \\
&\quad + v3\, v1_{x3,x3} - v1_{x3}\, v2_{x2} - v1\, v2_{x2,x3} - v1\, v3_{x3,x3} - v3_{t,x1} - v1\, v3_{x1,x1} \\
&\quad - v2_{x1}\, v3_{x2} - v2\, v3_{x1,x2} - v3\, v3_{x1,x3} + v3_{x1}\, v2_{x2}, v2_{t,x3} + v1_{x3}\, v2_{x1} \\
&\quad + v1\, v2_{x1,x3} + v2_{x3}\, v2_{x2} + v2\, v2_{x2,x3} + v3_{x3}\, v2_{x3} + v3\, v2_{x3,x3} - v3_{t,x2} \\
&\quad - v3_{x1}\, v1_{x2} - v1\, v3_{x1,x2} - v2_{x2}\, v3_{x2} - v2\, v3_{x2,x2} - v3_{x2}\, v3_{x3} - v3\, v3_{x2,x3}, \\
&\quad v1_t + v2\, v1_{x2} + v3\, v1_{x3} + p_{x1} - v1\, v2_{x2} - v1\, v3_{x3}, \\
&\quad v2_t + v1\, v2_{x1} + v2\, v2_{x2} + v3\, v2_{x3} + p_{x2}, \\
&\quad v3_t + v1\, v3_{x1} + v2\, v3_{x2} + v3\, v3_{x3} + p_{x3},\ v1_{x1} + v2_{x2} + v3_{x3}]]
\end{aligned}
$$

Changing the order $t > x1 > x2 > x3$ for $x1 > x2 > x3 > t$, we obtain the most difficult case among the computable ones. It cannot be computed directly. The characteristic set consists of only 9 differential polynomials, but they have 81, 34, 14, 13, 6, 5, 5, and 3 terms, respectively, and occupy about three pages and a half.

For the ranking where the derivatives $d_1^{i_1} \dots d_n^{i_n} y_j$ are ordered in the lexicographical order of the vectors $(i_1, \dots, i_n, j)$, the computations are quite similar to the case of two space variables; i.e., the same as for the order of total degree then lexicographic. The computation complexity depends only on the order of the derivatives: if $t > x1 > x2 > x3$, then the basis is obtained directly; if $x1 > x2 > x3 > t$, then results for another ranking should be used. But, in this case, the set itself has a very simple form, even simpler than in the first case considered. However, in that case the computation time is short and the results can be obtained directly. If $x1 > x2 > x3 > t$ and $p > v1 > v2 > v3$, then the set consists only of the reduced initial polinomials:

$$
\begin{aligned}
&[[v1_t + v2\, v1_{x2} + v3\, v1_{x3} + p_{x1} - v1\, v2_{x2} - v1\, v3_{x3},\ v1_{x1} + v2_{x2} + v3_{x3}, \\
&\quad v2_t + v1\, v2_{x1} + v2\, v2_{x2} + v3\, v2_{x3} + p_{x2}, \\
&\quad v3_t + v1\, v3_{x1} + v2\, v3_{x2} + v3\, v3_{x3} + p_{x3}]]
\end{aligned}
$$

However, it cannot be obtained directly.

Finally, for the lexicographical order, we did not manage to obtain any results even in the case of two space variables.

In conclusion I would like to express my gratitude to my scientific adviser E.V. Pankratiev for valuable remarks while preparing this paper.

References

[1] F. Boulier, D. Lazard, F. Ollivier, M. Petitot "Representation for the radical of a finitely generated differential ideal", ISSAC 1995, AMC Press, 158-166

[2] M.V. Kondratieva, A.B. Levin, A.V. Mikhalev, E.V. Pankratiev "Differential and Difference Dimension Polynomials", 1999, Kluwer Academic Publishers

Computer Algebra with Haskell: Applying Functional-Categorial-'Lazy' Programming

Serge D. Mechveliani

Program Systems Institute,
Pereslavl-Zalessky, Russia;
e-mail: mechvel@botik.ru

We give an outline of a computer algebra program written in a functional language Haskell and implementing certain piece of commutative algebra.

1 Introduction

The project of algebraic Domain Constructor DoCon [Me1] has grown into somewhat an advanced system: permutations, linear algebra, polynomial arithmetics, GCD, factorization, Gröbner bases, symmetric functions package, and some other tools.

It is written in Haskell language [Ha] (Miranda family), which we declare as supporting the FTCL approach: Functional-Typeful-Categorial-LazyEvaluation style in programming.

Also certain simplified version is presented as the BAL program [Me2] — basic algebra library for Haskell. For anyone going to deal with DoCon we recommend to read first the paper [Me2] discussing the main points of Haskell's relation to computer algebra.

Why designing another CA system, is not the `Axiom` program [Je] sufficient?

I experimented with the FTCL approach because **(a)** of importance of pure functionality, **(b)** of 'lazy' evaluation elegance, **(c)** I aimed to study the question of fitness to programming of mathematics of several known languages: Refal, ML, Haskell, term-rewriting logical languages ... **(d)** for maintainability, free-with-source program tools, described at all levels, are preferable.

And Haskell does provide several free-with-source implementations, among which the recent DoCon exploits GHC [GH].

DoCon is a program package. The general notions of a class, instance, polymorphic type, and such — are of the Haskell language. And DoCon is a library of particular algebraic categories, their instances and other items.

The **functionality** feature allows to treat a program (brought to internal form) as symbolic expression, like say, a polynomial in mathematics. This enables (in principle) the compiler to simplify such a program symbolically.

But it was often said that functional programming reduces the efficiency, that there is a necessity to use a pointer, a mutable array, and so on.

DoCon tries to shows that pure functionality and 'lazy' evaluation are practicable. The benchmarks reveal a good performance. These tests are described in the Section 'pe' of DoCon Manual [Mel]; they compare the CA programs DoCon-2.02, Axiom-2.2, MuPAD-1.4.2 on several algorithmically complex tasks.

We also discuss the design **difficulties** (Section 6). The most important one is, probably, common to all 'typeful' systems and concerns modelling a domain depending on a dynamic value. With existing Haskell, we find certain way out in a sample argument approach — see Section 4.

Acknowledgements. The author is grateful to *Dr. S.V.Duzhin* for his administrative support of the project, *Russian Foundation for Basic Research* (grant 98-01-00980) and
INTAS foundation (coordinator Professor B.Jacob) — for some part of DoCon was supported by these grants, CA center *MEDICIS* <http//www.medicis.polytechnique.fr> for making it possible to compare the chosen CA systems on one computer.

2 Mathematics implemented in DoCon-2.02

DoCon-2.02 includes in its functionality:

Methods:

Permutation group: composition, inversion, decomposition to cycles.

Fraction field over a gcd-ring: its arithmetic.

Linear algebra over an Euclidean domain: vector, matrix arithmetic (dense form), reduction to staircase and diagonal form of matrix, solving a system.

Polynomial arithmetic and g.c.d. in $P = R[x_1, \ldots, x_n]$, R a commutative ring. Four possible representations for P, free module over P and symmetric functions are given by the constructors `UPol`, `Pol`, `RPol`, `EPol`, `SymPol`; factorization in $k[x, y]$, k a finite field.

Gröbner basis, normal form, syzygy generators functions for $P = R[x_1, \ldots, x_n]$, R an Euclidean ring, and in a free module over P [Bu, Mo, MoM].

Symmetric function package: decomposition into elementary symmetrics, to other bases, operations with partitions [16].

Category hierarchy expressed partially via the data classes of Haskell:
`Set`, `Semigroup`, `Group`, `Ring`, `LinSolvRing` ... `LeftModule` ...,
some operations with the description terms of `Subset`, `Subgroup`, `Subring`, `Ideal`.

Domain constructors: `Permutation`; `Fraction` field for a gcd-ring, `Residue` ring by the ideal, and others.

Property processing: evaluation of certain small number of important algebraic domain property values is supported: `Finite`, `IsCyclicGroup`, `FactorialRing`, and the such. They serve as the correctness conditions for various methods, besides, present an important information by themself.

But many users need to do approximate computations, compute with differential operators, integration, and so on. With this respect, we point out that DoCon-2.02 is a free-with-source program that can be taken as basis and developed by adding user libraries.

3 About Haskell language

The main Haskell features are:
(F) pure functionality and high-order functions, (L) 'lazy' model of evaluation,
(M) evaluation by pattern matching is widely usable,
(T) recursive polymorphic parametric types with user-defined constructors, polymorphic functions, Hindley-Milner type system with static type resolution,
(C) categorial approach to types.

We could provide many examples on the 'lazy' programming elegance. But let us point it out: **sometimes 'laziness' leads to expenses.** In the worst case, it increases the cost from 'N steps of time + constant memory' to 'N steps of time + N cells of memory'.

The recursive polymorphic types are good because

They reflect the mathematical need to describe a domain of a map preventing senseless computations. The types describe certain part of the correctness conditions which is relatively easy for the compiler to check.

What is categorial programming.

Haskell provides the type classes and their *instances* to express the idea somewhat similar to object programming and to algebraic categories. For example, the declaration

```
class CommutativeRing a => GCDRing a where  gcd      : a -> a -> a
                                            canAssoc : a -> a
```

describes abstractly all the domains (types) `'a'` supplied with the operations `gcd`, `canAssoc`, these operations having the specified types. Also it puts that the operations of the category `CommutativeRing` (defined earlier) are inherited. The compiler controls the types and instance match for the data but not the properties of operations, such as associativity, opposite existence, and so on. Further, the instance declaration

```
instance GCDRing Integer where  canAssoc n = if n < 0 then -n else n
                                gcd n 0 = n
                                gcd n m = gcd m (mod n m)
```

tells the compiler in what way `Integer` is an instance of `GCDRing`. Then, the declarations

```
data Fraction a = a :/ a ...
instance GCDRing a => AdditiveSemigroup (Fraction a)
  where
  (x :/ y) + (x' :/ y') = ... usual way to sum fractions

instance GCDRing a => Field (Fraction a) where ...
```

define the arithmetic of a domain `Fraction a` via arbitrary instance `GCDRing a`. After this, our program is ready to operate uniformly with the fractions of integers, of integer polynomials and such.

4 Particular points of DoCon design

These are

(sa) sample argument (SA) approach to representing domain depending on a value,

(ic) implicit categories given by attribute values (see Item (bo) and Section 5),

(bo) 'base' operations (related to SA),

(ct) casting to domain of sample via `class Cast a b`,

(gx) so-called gx-ring method to operate with ideals given by generator lists

(dt) possibility to transform domains isomorphically by means of instances,

Here follow some comments.

Item (sa). The SA approach puts that an algebraic domain `D = D(s)` inside a type `T` is defined by any data element `s :: T`. The meaning of `D(s)` is defined individually for each known constructor `C` applied to form `s`. `D(s)` is defined according to `C` and the parameter values contained in `s`. `D(s)` does not depend on the choice of `s` in the same domain. For example, the vectors

```
s = Vec [0,1],  Vec [0,0],  Vec [-1,3,0]  :: Vector Integer
```

define the vector domains `D`, `D`, `D'` respectively, where `D` is the 2-dimensional vector domain over `Integer`, `D'` — the 3-dimensional one.

Each created element can be used further as a sample denoting a domain. With the SA approach, the correctness conditions provided by a sample are not checked by the compiler, they can be checked on demand by the user program. Ignoring such value dependent domains would make many computation methods impossible. For example, a natural design for the popular Chinese remainder method for the polynomial g.c.d. requires operating in different residue domains $R/(p)$, the needed set of the elements $p \in R$ is determined at the run-time. We could give other important examples.

Item (bo). The 'base' operations relate closely to the SA approach. Most explicit algebraic categories in DoCon export the corresponding 'base' operation: `baseSet`, `baseRing`, and others. Given any domain element `s :: T`, a 'base' operation builds the *domain description term* `tD = tD(s)` which gathers several attributes of a domain.

Item (ct). If `s` is an element of a certain domain, then one can use the construction `cast s x` to convert various data `x` to corresponding canonical values in the domain defined by `s`. For example, if `s ∈ Z[x]` is a polynomial over `Integer`, then the expressions

```
cast s 2, cast s (2,3)
```

give the constant 2, considered as a polynomial, and a one-term polynomial equal to $2 * x^3$.

Items (gx), (dt). The gx operations generalize the structures of the Gröbner basis and Gröbner reduction operations, Euclidean extended g.c.d and remainder division and also include the syzygy module generators finding. For example, one cannot compute the arithmetics and the ideal inclusion in a ring $P = (Integer/(4))[x, y, z]$ by, say, direcly appying Gröbner basis or Gröbner reduction. Instead, DoCon has the category instances for defining first the gx operations for *Integer* via the extended Euclidean

g.c.d. algorithm. Also the arithmetic and gx operations are defined for the domain $P' = Integer[x,y,z]$, as the generic `LinSolvRing` instance for $E[x_1,\ldots,x_n]$, the latter basing on the (weak) Gröbner basis method over an Euclidean ring E. Further, the gx structure is defined for the domain $R = P'/(4)$. This is done via the Gröbner reduction in P' by the ideal (4) and via the correspondence of the ideals in $P'/(4)$ to their counter-images in P'. And it remains to exploit the computable isomorphism

$$P = (Integer/(4))[x,y,z] \longleftrightarrow R = (Integer[x,y,z])/(4)$$

This scheme has to work for the domains of kind $(R/I)[x_1,\ldots,x_n]$. But DoCon has not yet implemented the above isomorhism. We tried first to use such kind of domain transformations for setting the gx operations for the domain $P = a[x_1,\ldots,x_n][y]$, where a is any domain qualified by the instance `LinSolvRing (Pol a)` — this asserts that the domain $a[x_1,\ldots,x_n]$ has a gx structure. And in this case, DoCon uses successfully the isomorphism $P \longleftrightarrow P' = a[y,x_1,\ldots,x_n]$
to port the `LinSolvRing` instance from the ring P' to P.

5 Example of programming

Given two polynomials `f, g` with coefficients in `Z = Integer`, find the list of the greatest common divisors for `f, g`, when their coefficients are reduced modulo several different integers `m` belonging to a list `ms`:

```
[gcd(f'm,g'm) | m <- ms],   ms :: [Z]
```

Here `f'm` denotes `f` with coefficients projected modulo `m`.
Let us program this in the BAL library. Its type denotations for this example are

```
f,g :: UPol Z,                  <--> Z[x]
f'm, g'm :: UPol (ResidueE Z)  <--> (Z/(m))[x]
```

The idea is that for different `m` the domain of the coefficients of `f'm` is different.
Below, the items `UPol`, `ResidueE`, `gcd`, `mapCoef`, `cast`, `rse`, `upol`, `smParse` are of BAL.

```
gcdMods :: EuclideanRing a => UPol a-> UPol a-> [a]-> [UPol (ResidueE a)]
gcdMods                       f       g       ms =

 [gcd [mapCoef (cast r) f, mapCoef (cast r) g] | r <- resSamples]
   where
   resSamples = [rse 'f' un m [] | m <- ms]       -- :: [ResidueE a]
   un         = unity $ head ms                   -- sample for 'a'

 -- Example of usage:
 let x1 = upol [(1,0)] "x"  :: UPol Z             -- unity polynomial: serves
                                                  -- also as a sample for Z[x]
     f = smParse x1 "(x^2 +2*x +3)*(3*x +1)"  -- parse value from string
     g = smParse x1 "(x^2 +2*x +3)*(x +2)^3"  -- using  x1  as  sample
```

```
                                        -- defining destination domain
    gcs      = gcdMods f g [2,3,5]
    gl       = last gcs                 --look into domain description
    (pS,pR) = (justBaseSet gl, justBaseRing gl)  -- of gl -for curiosity
in  (gcs, (osetCard pS, subringProps pR))
```

This expression is output as

```
( [x^2 + 1,  x^2 + 2*x,  x^3 + 4*x^2 + 2*x + 1],
  ( Infinity,  [(IsField,No),(Factorial,Yes),...] )
)
```

Here are some explanations: first on the Haskell programming in general, then — on this particular example:

`X :: T` denotes that data `X` is of type `T`. `[]` is the type constructor for List. `':'` is a data constructor of prepending an element to a list. `(x,y)` is a pair, it has a type `(a,b)` for `x :: a, y :: b`.

`f x y === ((f x) y)` denotes a function application `f(x,y)`. So, the above `cast` is a function of two arguments, `cast r` is a function of one argument,
`[gcd (f r) (g r) | r <- resSamples]` is the list of gcd-s when `r` runs through the list `resSamples`.

Haskell also uses the `$` denotation to save the parentheses in the function applications. For example, the expression `f x $ g y $ h z` converts to `f x (g y (h z))`.

Certain *indentation* agreement on the program lines serves to save the program structural parentheses `(; { } ('begin','end'))`.

The second line of the output of this program displays several attributes of the domain related to `gl`. `ResidueE` is an abstract data type. Some initial residue element

```
r :: EuclideanRing a => ResidueE a
```

can be created by the function `rse`.

For example, `rse 'c' 4 3 [(3,1)] :: ResidueE Z` creates the residue of 4 in the domain $Z/(3)$, 4 is being reduced modulo 3 to the internal representation 1; 3 goes to internal representation and is called a 'base'. Here `rse` treats the list `[(3,1)]` as the given factorization of the base; its usage is shown at the page 209. Another example: `(rse 'c' 2 3 [])`, `(rse 'c' 4 3 [])` create the two elements of the domain $Z/(3)$ that look internally as `Rse 2 3 []`, `Rse 1 3 []`.

The mode `'c'` makes the `rse` function to reduce canonically the first argument by the base. The mode `'f'` instructs it to apply `'c'` and also to prepare the factorization of the base — taking it as ready if it is given in the argument. The factorization can be skipped by setting it with the empty list `[]`. The parts of `r` can be extracted by the functions `rseRepr`, `rseBase`, `rseFt`. Once the programmer created any element `s :: ResidueE a`, other elements `x :: a` can be 'projected' standardly to the domain of `s` by applying `(cast s x)`.

The operation `mapCoef` is of the BAL class `PolLike`. `(mapCoef f p)` applies the function `f` to each coefficient of polynomial `p` bringing the result to the canonical form and returning a polynomial.

The library supplies the constructor `ResidueE` with

```
instance EuclideanRing a => Additive (ResidueE a) where ...
```

and other instances, up to `CommutativeRing` — with usual instances for the operations `+, *, ...` The library also defines the instances

```
instance EuclideanRing a => GCDRing (ResidueE a) where ...
instance EuclideanRing a => Field (ResidueE a)
```

The above instances support the arithmetic in `R= Z/(m)`. And since the proposed standard instance `GCDRing (UPol a)` defines gcd in R[x] via the arithmetic of R, all this enables automatically the above program to compute
`gcd [f'm, g'm] :: UPol (ResidueE Z)`.
The expression `(osetCard pS, subringProps pR)` shows explicitly the two attributes of the domain `R[x]` given by the last of gcd's as by the sample. To find these attributes, BAL analyses the domain tower `Z -- Z/(m) -- (Z/(m))[x]` deducing the attributes of each domain from the ones of simpler domains.

Now suppose that we append the base 4 to the list in the above program:

```
gcs = gcdMods f g [2,3,5,4]
```

The program passes the compilation, starts running, prints the result for 2,3,5, and then breaks reporting

```
Error:  gcd [f,g]  <- R[x],   R = Z/(4)   R is not a GCD Ring
```

This is because we model the algebraic domain $R = R(m) = Z/(m)$ depending on the parameter `m :: Z`. To compute `gcd` in `R[x]` a meaningful `gcd` algorithm in `R` is needed. For `R = Z/(m)`, such algorithm has sense only for a prime `m`. In such case `Z/(m)` is a *field* and a `GCDRing`, and `(gcd f'm g'm)` computes by the method 'over a GCDRing'. To reflect all this, the library provides the instances

```
instance EuclideanRing a => GCDRing (ResidueE a) where ...
instance EuclideanRing a => Field (ResidueE a)
```

The elements `resSamples :: [ResidueE Z]` belong to the domains `Z/(m1), Z/(m2), ...` inside the same type `ResidueE Z`. The instances `GCDRing, Field` match formally each domain `Z/(m)`, but these instances are valid only for a prime `m`.

The function call `gcdMods f g [2,3,5,4]` applies `(gcd f'm g'm)`. Then, the operation `gcd` extracts a coefficient sample `r` from `f'm` and tests `(detectGCDRing r)`. If it is not `Yes`, `gcd` sets the error break. In general, the programmer may skip any `detectCategory` test. But then, it is up to the user to apply the related methods only to the domains with meaningful parameters.

Each category in BAL is accompanied by its 'detect' function. For example, `GCDRing` has the detector `detectGCDRing :: GCDRing a => a -> PropValue`

The *sample* `r` is a residue element, it defines the current domain `R = Z/(m)`, it contains a parameter `ft = rseFt r`. The operation `gcd` applies `detectGCDRing` to test whether `Z/(m)` is really a GCD ring. The function `detectGCDRing` applies `baseGCDRing` ... and looks into the above factorization part `ft`. At the fourth application

in our example, it finds `ft = [(2,2)]`, showing that `m` is not prime, and `gcd` sets the break.

The domains like `Z`, `(Z, UPol Z)` *are* modelled by static type instances. In our example, however, `Z/(m)` cannot be expressed so. The SA approach by DoCon recognises a category instance partially by class instances and partially by analysing the parameters in a sample element.

This is illustrated by the behavior of our example program on the list `ms=[2,3,5,4]`.

6 Difficulties

The main problems encountered in the project concern
(ul) cases of unneeded 'laziness' (Section 3), (sa) domain depending on a value,
(le) language standard and implementation, (ct) implicit casting between domains.

Item (sa) was discussed earlier. Maybe, the *dependent types* feature ([Au]) can help the compiler to understand more of the value dependent domains.
Item (le). DoCon is written not precisely in the standard Haskell-98 but in a language including certain version of overlapping instances and some minor extensions. Mathematical needs require some further extension. But there arises a question of the standard agreement.
Item(ct). Casting between domains via the instances of `class Cast a b`
is not sufficient for really nice transformations. Some other function has to set automatically the operation `cast` of this class in appropriate places with the appropriate samples.

References

[Au] L. Augustsson. *Cayenne — a language with dependent types.* In Proceedings of International Conference on Functional Programming (ICFP'98). ACM Press, September 1998.

[Bu] B. Buchberger. *Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory.* CAMP. Publ. No.83-29.0 November 1983.

[GH] *The Glasgow Haskell Compiler.* `<http://haskell.org>`

[Ha] *Haskell 98: A Non-strict, Purely Functional Language.* Report of February 1999. `<http://www.haskell.org>`

[Je] R. D. Jenks, R. S. Sutor, et al. *Axiom, the Scientific Computation System.* Springer-Verlag, New York–Heidelberg–Berlin (1992).

[Ma] I. G. Macdonald. *Symmetric Functions and Hall Polynomials.* Clarendon Press, Oxford, 1979.

Me1] S. D. Mechveliani. *DoCon the Algebraic Domain Constructor. Manual and program* Pereslavl-Zalessky, 1998-2000,
<ftp.botik.ru:/pub/local/Mechveliani/docon/>

Me2] S. D. Mechveliani. *BAL — basic algebra library for Haskell.* (paper manuscript and source program) Pereslavl-Zalessky (2000-2001).
<http://www.botik.ru/pub/local/Mechveliani/basAlgPropos/>

Mo] H. M. Moeller. *On the Construction of Gröbner Bases Using Syzygies.* Journal of Symbolic Computation (1988), Vol. 6.

MoM] H. M. Moeller, F. Mora. *New Constructive Methods in Classical Ideal Theory.* Journal of Algebra, v 100, pp.138-178 (1986)

Mu] *MuPAD the Multi Processing Algebra Data Tool.* <http://www.mupad.de>

On an Implementation of Desingularization of Linear Recurrence Operators with Polynomial Coefficients

A. M. Mitichkina

Moscow State University,
Dept. of Computational Mathematics and Cybernetics,
Russia, Moscow 119899;
e-mail allam@dis.muh.ru

We present in this paper an implementation of the algorithms to desingularize linear difference operators with polynomial coefficients of the form

$$P = f_d(n)E^d + \cdots + f_1(n)E + f_0(n),\ f_i(n) \in \mathbb{Q}[n],\ n \in \mathbb{Z},\ i = 0 \ldots d$$

where the trailing and the leading coefficients $f_0(n)$ and $f_d(n)$ are not identically zero and have integer roots. The algorithms to solve the problem for the case where the trailing coefficient $f_0(n)$ vanishes for some integer n are presented in [1]. Based on the relation between the desingularization of the leading and the trailing coefficients, we present an algorithm to solve the problem for the case where the leading coefficient $f_d(n)$ have integer roots. We also describe an implementation of the algorithms [1] to extend a sequence which satisfies a given operator. This can be used to compute any element of the sequence including those in the set of the integer roots of the trailing and/or the leading coefficients.

1 Introduction

For a given linear difference operator with polynomial coefficients

$$P = f_d(n)E^d + \cdots + f_1(n)E + f_0(n),\ f_i(n) \in \mathbb{Q}[n],\ n \in \mathbb{Z},\ i = 0 \ldots d, \tag{1}$$

where E denotes the shift operator, defined by $E(f(n)) = f(n+1)$, *the problem points* of P is the set of all integer roots of either the trailing coefficient $f_0(n)$ or the leading coefficient $f_d(n)$ of P, and the desingularization of the operator P involves the elimination of the problem points for either $f_0(n)$ or $f_d(n)$. This is attained via the construction of an operator $Q \in \mathbb{Q}[n, E]$ which is "similar" to P and which does not contain the problem points.

The algorithms as presented in [1] allow one to construct the operator Q which is a composition of a difference operator $H \in \mathbb{Q}(n)[E]$ and the given operator P, i.e.,

$$Q = H \circ P, \tag{2}$$

where the integer roots of the trailing coefficient $f_0(n)$ in (1) are excluded.

The process of constructing such an operator where the problem points are eliminated is named *the desingularization* (either by the trailing or by the leading coefficient) of the given operator P, and the operator Q in (2) is named *the desingularizing operator* for P. A description of the implementation of these algorithms is presented in section 2.1. In section 2.2, we describe an algorithm to construct the operator Q which excludes the integer roots of the leading coefficient. The algorithm consists of the application of some transformations to the input operator P in order to reduce the problem to that of excluding the integer roots of the trailing coefficient. Section 3 shows description of a program to desingularize a given sequence, i.e., to extend the sequence based on some given elements. For a given sequence, the *problem elements* are defined to be the elements with the number from the set of the integer roots of either the trailing coefficient or the leading coefficient, and the remaining elements are named *regular elements*. One can use the program to compute both the problem and the regular elements of the sequence.

2 Desingularization of Operators

2.1 An implementation of the algorithms to desingularize operators and the ε-criterion.

For a given linear difference operator with polynomial coefficients P of the form (1), the algorithm for the desingularization by the trailing coefficient of P [1] returns an answer in one of the following three forms:

1) a desingularizing operator where all integer roots of the trailing coefficient $f_0(n)$ are excluded;

2) a desingularizing operator where at least one integer root of $f_0(n)$ is excluded;

3) no operator, i.e., it is not possible to exclude even one root of $f_0(n)$.

The ε-criterion as established in [1] determines if the operator Q of the form (2) exists. It also helps to determine the number of integer roots and which roots can be excluded during the construction of Q. The main idea of the ε-criterion consists of the substitution of n by $n + \varepsilon$ where ε is an indeterminate parameter, the application of some transformations, and finally, the back substitution of ε by 0 and the analysis of the resulting operator. The procedure `IsDesingularizable` is an implementation of both the ε-criterion and of the algorithm to construct the desingularizing operator. It has the calling sequence

$$\text{IsDesingularizable}(P, E, n, Q);$$

where P is a linear difference operator of the form (1), E is the shift operator w.r.t. n, and Q is an optional argument which can be any unassigned name. The output is the result of applying the ε-criterion, i.e., it returns *true* if the desingularizing operator Q can be constructed and all the integer roots in the trailing coefficient of P can be excluded, *false* if it is not possible to exlude any integer root, and FAIL(*ints*) if it is possible to

exclude at least one integer root. In the last case, *ints* is a sequence of integers specifying the integer roots which cannot be excluded. If the optional argument Q is specified, the constructed desingularizing operator will be assigned to Q, provided that it exists.

Example 1. Consider the operator

```
> P := (n-3)*(n-2)*E+n*(n-1);
```

$$P := (n-3)(n-2)E + n(n-1)$$

```
> IsDesingularizable(P,E,n);
```

$$true$$

This means that all integer roots of the trailing coefficient can be excluded during the construction of the desingularizing operator.

Example 2. Consider the operator

```
> P := (n-5)*(n-2)*E+n*(n-1)^2;
```

$$P := (n-5)(n-2)E + n(n-1)^2$$

```
> IsDesingularizable(P,E,n,'Q');
```

$$\mathrm{FAIL}(0)$$

The above result shows that the integer roots will be partially excluded during the construction of the desingularizing operator Q in (2). To be more specific, it is possible to exclude the root at $n = 1$. However, it is not possible to exclude the root at $n = 0$.

```
> Q;
```

$$\frac{1}{12}n\,E^3 - \left(\frac{1}{12}n^2 - \frac{43}{12}n + \frac{59}{6}\right)E^2 + (4n^2 + 7n + 6)\,E + n$$

The constructed operator Q can be written as $Q = H \circ P$ where

$$H = \frac{1}{1440}\frac{1}{n-1}E^4 - \frac{1}{36}\frac{1}{n-1}E^3 - \frac{1}{2}\frac{1}{n-1}E^2 + \frac{1}{2}\frac{15n-19}{n^2-2n+1}E + \frac{1}{n^2-2n+1}.$$

Note that the operator H in (2) should not have polynomial coefficients.

2.2 The desingularization of the operators by the leading coefficient

Consider the problem of excluding the integer roots of the leading coefficients of the operator P in (1). An algorithm to solve this problem which is analogous to the one for the trailing coefficient is discussed in [1]. This requires the development of two independent programs. Therefore, it would be desirable if we are able to transform the problem for the leading coefficient to that for the trailing coefficient.

Denote by $*$ the transformation to linear difference operators with polynomial coefficients, defined by the substitution of n by $-n$, and of E^a by E^{-a}, $a \in \mathbb{Z}$. The following properties are needed subsequently. The proofs are simple and hence omitted.

Proposition 1. *For all* $P, P_1, P_2 \in \mathbb{Q}[n, E]$,

1) $(P_1 + P_2)^* = P_1^* + P_2^*$,

2) $(P^*)^* = P$,

3) $(P_1 \circ P_2)^* = P_1^* \circ P_2^*$.

Applying the transformation * to (1) results in

$$P^* = f_d(-n)E^{-d} + \cdots + f_1(-n)E^{-1} + f_0(-n). \tag{3}$$

Define the operator P_1 as follow.

$$P^* = P_1 \circ E^{-d},\ P_1 = f_d(-n) + \cdots + f_1(-n)E^{d-1} + f_0(-n)E^d. \tag{4}$$

Notice that the polynomial $f_d(-n)$ is the trailing coefficient of the operator P_1. If we establish a relationship between the problem of excluding the integer roots of the polynomial $f_d(-n)$ in P_1 and that of the polynomial $f_d(n)$ in P, which we are interested in, we can reduce the problem of desingularization by the leading coefficient of the operator P to the problem of desingularization by the trailing coefficient of the operator P_1.

Recall that the application of the algorithm for the desingularization by the trailing coefficient to P_1 gives two possible results: the exclusion of some roots of the trailing coefficients (all or at least one), and the exclusion of no root (no desigularizing operator exists). Both cases are considered in the following theorem.

Theorem 1. *For a given operator* $P \in \mathbb{Q}[n, E]$ *of the form (1),* $\operatorname{ord} P = d$, *let* $\{l_1, \ldots, l_m\}$, $l_1 > \cdots > l_m$ *be the integer roots of the leading coefficient* $f_d(n)$ *of* P. *Let the operator* P_1 *be as defined in (4). For* $1 \le k \le m$, *let* $W \in \mathbb{Q}[n, E]$ *be a desingularizing operator for* P *where the integer roots of the leading coefficients* $\{l_k, \ldots, l_m\}$ *are excluded, and* $U \in \mathbb{Q}[n, E]$, $\operatorname{ord} U = d_1$ *be a desingularizing operator for* P_1 *where the integer roots of the trailing coefficient* $\{-l_k + d_1 - d, \ldots, -l_m + d_1 - d\}$ *are excluded. Then the existence of* W *is equivalent to the existence of* U.

Proof: Suppose that the operator U exists, and can be written as

$$U = u_d(-n) + u_{d-1}(-n)E + \cdots + u_0(n)E^d + \cdots + u_{d_1}(-n)E^{d_1},\ d_1 > d.$$

Since U is a desingularizing operator for P_1, $U = S \circ P_1$ for some $S \in \mathbb{Q}(n)[E]$. Therefore, $U \circ E^{-d} = S \circ P_1 \circ E^{-d}$. It follows from (4) that

$$U \circ E^{-d} = S \circ P^*. \tag{5}$$

Applying the transformation $*$ to both sides of (5) yields $H = U^* \circ E^d = S^* \circ P$, which can be written in the form

$$H = u_{d_1}(n)E^{d-d_1} + \cdots + u_0(n) + \cdots + u_{d-1}(n)E^{d-1} + u_d(n)E^d.$$

Since $d_1 > d$, the coefficients of some terms E^j, $j < 0$ in H might not vanish. They can be removed by multiplying H from the left by E^{d_1-d}. This yields $E^{d_1-d} \circ H = E^{d_1-d} \circ S^* \circ P$. Set $W = E^{d_1-d} \circ S^* \circ P$. It is easy to see that W is a desingularizing operator for P, and

$$W = u_{d_1}(n+d_1-d)+\cdots+u_0(n+d_1-d)E^{d_1-d}+\cdots+u_{d-1}(n+d_1-d)E^{d_1-1}+u_d(n+d_1-d)E^{d_1}.$$

Since the desingularizing operator U for P_1 does not have in the trailing coefficient $u_d(-n)$ the roots $\{-l_k+d_1-d,\ldots,-l_m+d_1-d\}$, by definition of the transformation $*$, the leading coefficient of U^* does not have the roots $\{l_k-d_1+d,\ldots,l_m-d_1+d\}$. Consequently, the leading coefficient of H does not have these roots either. Since W is obtained by applying E^{d_1-d} to H, by definition of the shift operator E, the leading coefficient of W does not have the roots $\{l_k,...,l_m\}$.

Suppose that the operator W exists, and can be written as

$$W = w_0(n) + w_1(n)E + \cdots + w_d(n)E^d + \cdots + w_{d_2}(n)E^{d_2},\ d_2 > d.$$

Since W is a desingularizing operator for P, $W = T \circ P$ for some $T \in \mathbb{Q}(n)[E]$. Therefore,

$$W \circ E^{-d} = T \circ P \circ E^{-d}. \tag{6}$$

Applying the transformation * to both sides of (6) yields $W^* \circ E^d = T^* \circ P^* \circ E^d$. It follows from (4) that $H = W^* \circ E^d = T^* \circ P_1$, which can be written as

$$w_0(-n)E^d + w_1(-n)E^{d-1} + \cdots + w_d(-n) + \cdots + w_{d_2}(-n)E^{d-d_2}.$$

Since $d_2 > d$, the coefficients of some terms E^j, $j < 0$ in H might not vanish. They can be removed by multiplying H from the left by E^{d_2-d}. This yields $E^{d_2-d} \circ W^* \circ E^d = E^{d_2-d} \circ T^* \circ P_1$. Set $U = E^{d_2-d} \circ W^* \circ E^d$. U is right divisible by P_1, and has the form

$$w_{d_2}(-n+d_2-d)+\cdots+w_d(-n+d_2-d)E^{d_2-d}+\cdots+w_1(-n+d_2-d)E^{d_2-1}+w_0(-n+d_2-d)E^{d_2}. \tag{7}$$

Since the desingularizing operator W for P does not have in the leading coefficient w_{d_2} the roots $\{l_k,\ldots,l_m\}$, by definition of the the transformation *, the trailing coefficient of W^* does not have the roots $\{-l_k,\ldots,-l_m\}$. Consequently, the trailing coefficient of H does not have these roots either. Since U is obtained by applying E^{d_2-d} to H, by definition of the shift operator E, the trailing coefficient $w_{d_2}(-n+d_2-d)$ of U does not have the roots $\{-l_k-d_2+d,\ldots,-l_m-d_2+d\}$. It follows from (7) that U has the order d_2. Therefore, U is a desingularizing operator for P_1. ∎

Note that the function `IsDesingularizable` also accepts an optional argument which is a name and has the value either `lead` or `trail`. This argument is used to determine whether the desingularization should be done to the leading or the trailing coefficient. The default if `trail`.

Example 3.

```
> P := (n-2)*(n^3+2)*E+n;
```

$$P := (n-2)\,(n^3+2)\,E + n$$

```
> IsDesingularizable(P,E,n,Q,'lead');
```

$$true$$

```
> Q;
```

$$-(n^3+6n^2+12n+10)E^3-\left(\frac{2}{3}n^3+\frac{4}{3}n^2+1\right)E^2-\left(\frac{1}{6}n^3-\frac{1}{3}n^2+1\right)E-\frac{1}{6}$$

Note that Q can be written as $H \circ P$ where

$$H=\frac{6}{n}E^2+\frac{4}{n}E+\frac{1}{n}.$$

2.3 On the construction of the multipliers

The algorithm of desingularization builds the resulting operator based on a set $\mathcal{S}$ of constructed operators each of which excludes one root in the leading or trailing coefficient *coeff*. If *coeff* has only one root or it is possible to exclude only one root from the whole set of the roots of *coeff*, the operator constructed after the first step is the final result. In the case when *coeff* has more than one roots, and it is possible to exclude m roots, $m>1$, i.e., the set $\mathcal{S}$ consists of m operators $T_1,\ldots,T_m$ each of which is right divisible by the given operator and each of which has one root being excluded from *coeff*, then the resulting operator with m roots in *coeff* being excluded is constructed as a linear combination of the m constructed operators, i.e.,

$$T=\mu_1T_1+\mu_2T_2+\cdots+\mu_mT_m. \tag{8}$$

The construction of the multipliers μ_i in (8) is based on the condition in [1] which relates to the absence of integer roots in the polynomial

$$\sum_{j=1}^{m}\mu_j(n-l_1)^{\gamma_1}\cdots(n-l_{j-1})^{\gamma_{j-1}}(n-l_{j+1})^{\gamma_{j+1}}\cdots(n-l_m)^{\gamma_m}, \tag{9}$$

where $\{l_1,\ldots,l_m\}$ is the set of integer roots being excluded, and γ_i is the multiplicity of the root l_i, $1\le i\le m$. If $\gamma_i=1$ for $1\le i\le m$, we can compute the multipliers by setting the sum in (9) to 1. Otherwise, we use the method of random fitting multipliers, i.e., assigning an arbitrary value to μ_i and checking the condition on the absence of the integer roots. In this case, it is possible that the computed operators are different for the same input.

Example 4. In this example, the multipliers were computed by setting the sum in (9) to 1.

```
> P := (n-5)^2*(n-9)*E^3+n*(n+1);
```

$$P:=(n-5)^2(n-9)E^3+n(n+1)$$

```
> IsDesingularizable(P,E,n,H); H;
```

$$true$$

$$-\frac{5}{14}(n+4)^2E^{12}+\left(\frac{807}{7}n^2-\frac{3383}{14}n-\frac{4637}{14}\right)E^9$$
$$+\left(\frac{165}{4}n^2-\frac{5001}{14}n+\frac{22314}{7}\right)E^6+\left(\frac{161}{4}n+\frac{995}{4}\right)E^3-1.$$

Example 5. In this example, the multipliers were selected at random.

```
> P := (n-3)*(n-2)*(n-1)*E+n^2*(n-1);
```

$$P := (n-3)(n-2)(n-1)E+n^2(n-1)$$

```
> IsDesingularizable(P,E,n,H);
```

$$true$$

```
> H;
```

$$\left(\frac{1}{28}n-\frac{1}{7}\right)E^{12}+\left(\frac{165}{28}n-\frac{547}{28}\right)E^9-\left(\frac{9}{28}n-\frac{1821}{14}\right)E^6-\left(\frac{25}{4}n+\frac{139}{4}\right)E^3-1$$

Second call to `IsDesingularizable` with the same input.

```
> IsDesingularizable(P,E,n,H);
```

$$true$$

```
> H;
```

$$-\frac{5}{222}(n+1)(n+2)E^4+\left(\frac{655}{222}n^2-\frac{65}{111}n-\frac{255}{74}\right)E^3$$
$$+\left(\frac{110}{37}n^2+\frac{420}{37}n+\frac{1180}{37}\right)E^2+\left(\frac{99}{67}n^2-\frac{25685}{2479}n+\frac{6568}{2479}\right)E$$
$$+\left(\frac{99}{67}n^2+\frac{5}{37}n-\frac{5}{37}\right)$$

3 Desingularization of Sequences

The second program is for desingularization of sequences. For the given several terms of a sequence which satisfy a given operator of the form (1), the main problem is to construct the extension of the sequence. Of most interest is the case where the direction of extension passes through the problem points. Note that the construction of a desingularizing operator for (1) is a variant of the solution to this problem. It is shown in [1] that even though there might exist different desingularizing operators, the sequence and its extensions, obtained by the construction of new operators, will satisfy the given operator, and the extensions from different operators will be the same.

In order to make the extension, there is no need to use the expensive algorithm to construct the desingularizing operator. The ε-criterion clearly allows one to determine

if it is possible to construct the desingularizing operators, and in terms of sequence, which members we can pass through and continue the sequence. Some modification of the ε-criterion allows one to continue the sequence (if it is possible) without the need to construct the operator Q in (2).

It is obvious that the members of the sequence starting from $\alpha + 1$ where

$$\alpha = \max_{n \in \mathbb{Z}} \{n | f_d(n) = 0\}$$

to the direction of the increasing indices do not pose any problem. It is also well-known that if we have d elements of the sequence, we can calculate any regular element. The idea of the ε-criterion helps to remove the problem elements temporarily if we make the transition from the given operator P to the operator P_ε by replacing n by $n + \varepsilon$. After that we can compute elements of the sequence as rational functions of ε, evaluate the unknown terms step by step via $c(\alpha + 1), \ldots, c(\alpha + d + 1)$. After the transformations we make the substitution $\varepsilon = 0$.

The procedure `DesingularizedExtensionSequence` is an implementation of this algorithm. It has the calling sequence

$$\text{DesingularizedExtensionSequence}(P, E, n, ini\_sys, c, N);$$

where P is a linear difference operator with polynomial coefficients, E is the shift operator w.r.t. n, *ini_sys* is a set of several terms of the given sequence that satisfy P, c is the name of the sequence, and N is an integer denoting the last/first term of the desirable extension. The output from `DesingularizedExtensionSequence` is a list of the computed terms of the sequence.

Example 6. Consider the operator P from Example 1.

$$P := (n-3)(n-2)E + n(n-1)$$

We would like to compute the elements of the sequence from -4 to 4 from the value of one element of the sequence (note that the order of the operator P is 1).

```
> DesingularizedExtensionSequence(P,E,n,{c(4)=1},c,-4);
```

$$[c(-4) = 105, c(-3) = -50, c(-2) = 20, c(-1) = -6,$$
$$c(0) = 1, c(1) = 0, c(2) = 0, c(3) = 0, c(4) = 1]$$

Notice that the problem terms $c(0)$ and $c(1)$ are computed. Since the regular elements are on the right of the roots of the trailing coefficients in the positive direction of the abscissa line, the problem points are encountered only in the case of extending the sequence to the left, i.e. when $N < \alpha$. Obviously, the described algorithm is correct for the case of extending the sequence to the right, i.e., when $N > \alpha$ without problem terms. If $\alpha < N < \min_{n \in \mathbb{Z}} \{n | f_0(n) = 0\}$, we calculate the elements analogously. The difference is that the calculation will be to the direction of the decreasing indices.

Example 7. Let us return one more time to the operator P from Example 1.

```
> DesingularizedExtensionSequence(P,E,n,{c(-1)=1},c,4);
```

$$[c(-1) = -6,\ c(0) = 1,\ c(1) = 0,\ c(2) = 0,\ c(3) = 0,\ c(4) = 1]$$

Note that the algorithm gives a result without any error only in the case when

- the ε-criterion gives the positive answer concerning the calculation of the problem terms;
- the elements of the sequence from $\alpha + 1$ to $\alpha + d + 1$ are given.

In the remaining cases, the procedure can be invoked; however, there is no guarantee of success.

If a user provides fewer than d elements, then the remaining elements will be represented symbolically such as $c(n)$ where n is an integer, and c is the name of the sequence.

Example 9. Consider the operator

```
> P := (n-1)*E^2+(1/2-n^2)*E+(1/4)*n*(2*n-1);
```

$$P := (n-1)E^2 + \left(\frac{1}{2} - n^2\right) E + \frac{1}{4} n\,(2n-1)$$

The order of the operator is 2. However, we provide only 1 element of the sequence.

```
> DesingularizedExtensionSequence(P,E,n,{c(5)=a},c,0);
```

$$[c(0) = -\frac{8}{13}\,a + \frac{53}{13}\,c(2),\ c(1) = 2\,c(2),\ c(2) = c(2),\ c(3) = \frac{51}{104}\,c(2) + \frac{1}{13}\,a,$$
$$c(4) = \frac{45}{208}\,c(2) + \frac{7}{26}\,a,\ c(5) = a]$$

In this example $c(2)$ was chosen by the system as the parameter.

References

[1] S.A. Abramov, M. van Hoeij, Desingularization of Linear Difference Operators with Polynomial Coefficients. *Proceedings ISSAC'99*, 1999, pp. 269-275 .

Involutive and Classical Gröbner Bases Construction from the Computational Viewpoint

V.A. Mityunin, A.I. Zobnin, A.I. Ovchinnikov, A.S.Semyonov

Moscow State University,
Department of Mechanics and Mathematics,
Russia, Moscow 119899;
e-mail: vmit@metric.ru al_zobnin@mail.ru
cathoduv@mtu-net.ru semyonov@mccme.ru

It is known that standard basis computation is a problem of great computational complexity both in time and in space. Modern computer algebra systems and software for computation of standard bases are able to compute only rather small systems. There are at least two obstacles that prohibit the us from moving further. The first problem in the standard bases computation is the coefficient growth. The second one consist in the difficulty of parallelization of the known algorithms. There are two well-known approaches to computation of standard bases: the classical Buchberger and the involutive basis construction algorithm. They have some obvious advantages and disadvantages but the nature of this difference can hardly be understood. We have developed and implemented an interpretation of the classical Buchberger and involutive bases computation algorithms using the C++ and investigated the bases sizes, computational times and some other parameters of the algorithms. From our point of view such a comparison is useful for a better understanding of the algorithms. We have also developed a parallel version of this interpretation of the Buchberger algorithm.

The work was supported by INTAS(project no. 99-1222).

1 Introduction

There is some freedom of choice of some parameters in the Buchberger algorithm. The most important parameter is the selection strategy. Probably, the best choice is the sugar one, which was presented and discussed in the [1]. Another good step is to homogenize the basis. Both approaches have some disadvantages. The sugar strategy often works not very well. Homogenizing basis, however, is good for strategy, but not so good for the basis to be computed, as it was noticed in [1]. We combine these options of algorithm avoiding the explicit homogenization of the basis by simulating the homogeneous algorithm described below. This version of the algorithm was never published before. The results obtained were compared with the best freely available implementation of the Buchberger algorithm in CAS Singular and show better computational time in many examples. We also compare the results with the implementation of the involutive basis construction algorithm implemented by prof. V.P.Gerdt and collaborators, for detailed description see

[2]. Parallelization of pseudo-homogeneous version is also presented. There are numerous papers described parallelization of the Bucheberger algorithm, see for example [3], [4], [5], [6], but the results were never very good. J.C.Fauger showed that the Buchberger algorithm is very sequentional one. In our parallelization we tried not to break the strategy because it is very important task as was noticed by C.Traverso and introduced some way to estimate the parallelization quality in addition to the computational time.

2 Buchberger algorithm

There are many interpretations of the Buchberger algorithm. We will consider the version described by Buchberger, see [7], [8]. The principal (and essential) improvement in this algorithm was the use of the criteria for avoiding useless S-pairs computation. There is a nice version of the algorithm one can use in the homogeneous case. This modification is not so well known. One can refer to paper [5] for detailed description of the homogeneous version of the Buchberger algorithm. The main idea is to slowly rise the total degree of polynomials and S-pairs. Experiments have shown that the coefficient growth is not so rapid in the homogeneous case for most examples. We combine these strategies. Here we present our interpretation of this algorithm applied to the inhomogeneous case, Pseudo-Homogeneous Gröbner basis. As one can see, we use sugar for simulating homogeneous computational process while not explicitly homogenizing the ideal. We slowly rise the total degree of the polynomials and S-pairs w.r.t. their sugar. From the definition of sugar, it is obvious, that the prove of the correctness and termination of the algorithm is the same as for the homogeneous version.

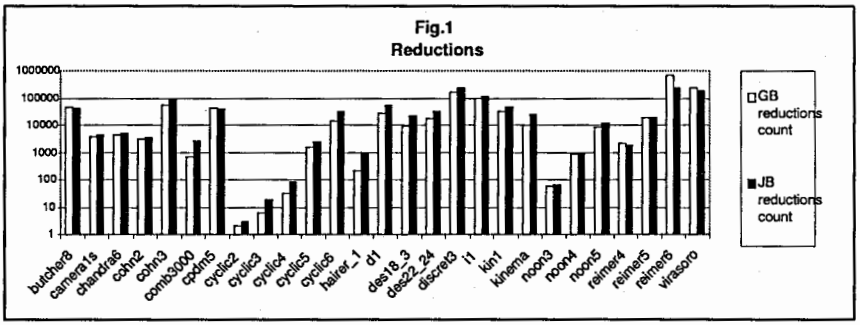
As the criteria for avoiding unnecessary reductions in the the algorithm we used Gebauer-Möller (see [9] and [10]), implemented as procedure syzBasis(). When computing the normal form, we chose double-sugar strategy described in [1]. The main idea behind is to perform the reduction only if it does not rise the sugar of the polynomial. This strategy was implemented in the sugarnormalf() procedure. We tried to avoid intermediate basis autoreductions due to the time complexity. Instead of this we just add every new non-zero normal form to the intermediate basis and apply procedure finalreduce() after the computation of the basis to find redundant polynomials. This procedure is very simple, it just discards polynomials with the leading monomial that is a multiple of the leading monomial of any other polynomial from the basis , and then performs full autoreduction, which is usually very fast. Beside this, the basis before finalreduce() often contains exactly the same number of elements as the true Gröbner basis, and is never much more, and from our viewpoint this proves that the computation is efficient.

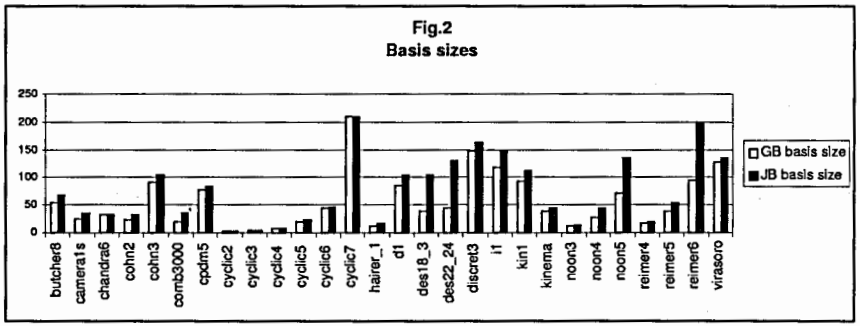
For detailed description of other auxiliary procedures like sugar computation, autoreducing and S-polynomial computations refer to [8] and [1].

We also implemented the involutive basis construction algorithm as it was described in [11], [12], [13] and [14]. Currently there is a much more efficient version, for detailed description refer to [2]. Our goal was to investigate the number of reductions as well as the computation time of this algorithm compared with classical Buchberger algorithm.

Algorithm **Pseudo–Homogeneous GröbnerBasis**
Input: $G, \prec$ - admissible ordering
Output: GB, a Gröbner basis of [G]
begin
$G \longleftarrow autoreduce(G)$
$g \longleftarrow$ select minimal element of G w.r.t. sugar
$currentdegree \longleftarrow sugar(g)$
$G \longleftarrow G \setminus \{g\}$
$GB \longleftarrow g$
while $B \neq \emptyset$ **and** $G \neq \emptyset$ **do**
$(f, g) \longleftarrow$ select element from B with a minimal sugar of the $spoly(f, g)$
$s \longleftarrow spoly(f, g)$
while $sugar(s)$ is equal to a $currentdegree$ **do**
$h \longleftarrow sugarnormalf(s, GB)$
if $h \neq 0$
then $GB \longleftarrow GB \cup \{h\}$
$B \longleftarrow syzbasis(h, B)$
end if
$(f, g) \longleftarrow$ select element from B with a minimal sugar of the $lcm(f, g)$
$s \longleftarrow spoly(f, g)$
end while
$f \longleftarrow$ select minimal element from G w.r.t. sugar
$G \longleftarrow G \setminus \{f\}$
while $sugar(f)$ is equal to a $currentdegree$ **do**
$h \longleftarrow sugarnormalf(f, GB)$
if $h \neq 0$
then $GB \longleftarrow GB \cup \{h\}$
$B \longleftarrow syzbasis(h, B)$
end if
$f \longleftarrow$ select minimal element from G w.r.t. sugar
$G \longleftarrow G \setminus \{f\}$
end while
$currentdegree \longleftarrow currentdegree + 1$
end while
$(GB) \longleftarrow finalreduce(GB)$
return(GB)
end

The results obtained show that the criteria are not so important in the course of the involutive algorithm. This is the same result as in study [2]. As a test suite we used polynomial systems collections presented by Prof. Jan Verschelde in his homepage. It is known, that the involutive basis has the same or greater size as the corresponding classical Gröbner basis. It can be very large in some cases. But surprisingly, there are some examples ("eco" and "reimer") where the involutive basis size is considerably





larger than that of the corresponding Gröbner basis and the same holds for the number of reductions, but for the computational time the reverse is true. On the other hand, there is an example ("noon") where computational time and size of the involutive basis is rather big. In the Fig.1 one can see the number of reductions for some systems from the test suite, and the number of reductions reductions is usually almost the same for the involutive and for the Gröbner bases. In the Fig.2 one can see the basis sizes for this systems. It is interesting, that in practice there are only few cases when the involutive basis is significantly larger than the corresponding Gröbner basis.

As for the computational time, our implementation is currently approximately two times slower than the best free implementation of the classical Buchberger algorithm in the CAS Singular for modular computations and is significantly faster in a number of examples with the big integers. From our viewpoint this demonstrate that the Pseudo-Homogeneous version is better than the classical version of Gröbner basis w.r.t. the coefficient growth.

In our further research we will continue investigation of selection strategies and other options of algorithms, since some of them remain unclear.

3 Parallelization

It is not easy to parallelize the classical Gröbner basis algorithm. It seems that the algorithm described above can be parallelized with better quality. The most successful attempt to parallelize the classical Buchberger algorithm was made by J.C.Faugere in his FGb software. The parallel version of the involutive basis construction algorithm was described in [15], and the results were better w.r.t. classical version.

In this paper we present two attempts to parallelize the Pseudo-Homogeneous Buchberger algorithm of computation of Gröbner bases. As Faugere has showed, the problem of Gröbner basis computation is very sequential and, therefore, every attempt to parallelize essentially breaks the original algorithm. The main reason is that the result of the polynomial reduction often heavily depends on the previously found basis polynomials. Nevertheless, modifying a reduction strategy we can achieve some acceleration.

We explore two principle ideas. Assume, that at a current step we have a polynomial p to be reduced. Let us denote the list of previously reduced (and added to the basis) polynomials by P. The sequential version reduces p by P at this step and adds it to P in the non-zero case.

What can we do? The list P can be divided into some parts, which can be distributed on different workstations. Let us denote these portions by $P_1, \ldots, P_n$. Thus we can send p to the first workstation and reduce it by P_1. At the next step we can send the result to the workstation, that keeps P_2. And at the same time we can send the next polynomial to be reduced, q, to P_1. This is the main idea of the Pipeline algorithm (Fig.3).

The second idea is just the opposite. We distribute the whole P to all workstations and simultaneously reduce the queue of polynomials $p_1, \ldots, p_n$ by P. This is the Conveyer strategy (Fig.4).

Main ideas of the Pipeline strategy were developed in [4], and these of the Conveyer strategy - in [5]. Of course, this brief discussion leaves many questions to be cleared further. Experiments show that the Conveyer version is much more rapid than the Pipeline is. Nevertheless, the Pipeline should be studied, as some questions of the character of its operation remain unclear.

Algorithm **processpipeline**
Input: GB - polynomials list, B - the set of critical pairs
Output: modified set GB and B
begin
  $h \longleftarrow$ receive polynomial from the pipeline
  if $h \neq 0$ **then**
    if h was modified by reductions **then**
      add h to the *pendinglist*
    else
      $GB \longleftarrow GB \cup \{h\}$, $B \longleftarrow$ *syzbasis*(h, B)
      select the next slave and send h to it as new basis element
    end if
  end if
  return (GB, B)
end

Algorithm **Pipeline GröbnerBasis**
Input: $G, \prec$ - admissible ordering
Output: GB, a Gröbner basis of [G]
begin
  $G \longleftarrow autoreduce(G)$
  $g \longleftarrow$ select minimal element of G w.r.t. sugar
  $currentdegree \longleftarrow sugar(g)$
  $G \longleftarrow G \setminus \{g\}$
  $GB \longleftarrow g$
  select the first slave and send g to it as new basis element
  $pendinglist \longleftarrow \{\}$
  while $B \neq \emptyset$ **and** $G \neq \emptyset$ **do**
    $(f, g) \longleftarrow$ select element from B with a minimal sugar of the $spoly(f, g)$
    $s \longleftarrow spoly(f, g)$
    while $sugar(s)$ is equal to a *currentdegree* and *pendinglist* is not empty **do**
      if *pendinglist* is not empty **then**
        select element from the *pendinglist* and send it to the pipeline
      else
        send s to the pipeline
      end if
      $(GB, B) \longleftarrow processpipeline(GB, B)$
      $(f, g) \longleftarrow$ select element from B with a minimal sugar of the $lcm(f, g)$
      $s \longleftarrow spoly(f, g)$
    end while
    $f \longleftarrow$ select minimal element from G w.r.t. sugar
    $G \longleftarrow G \setminus \{f\}$
    while $sugar(f)$ is equal to a *currentdegree* and *pendinglist* is not empty **do**
      if *pendinglist* is not empty **then**
        select element from the *pendinglist* and send it to the pipeline
      else
        send f to the pipeline
      end if
      $(GB, B) \longleftarrow processpipeline(GB, B)$
      $f \longleftarrow$ select minimal element from G w.r.t. sugar
      $G \longleftarrow G \setminus \{f\}$
    end while
    $currentdegree \longleftarrow currentdegree + 1$
  end while
  $(GB) \longleftarrow finalreduce(GB)$
  return(GB)
end

Algorithm **Reductors Conveyer GröbnerBasis**
Input: $G, \prec$ - admissible ordering
Output: GB, a Gröbner basis of [G]

```
begin
  G ⟵ autoreduce(G), g ⟵ select minimal element of G w.r.t. sugar
  currentdegree ⟵ sugar(g), G ⟵ G \ {g}, GB ⟵ g, pendinglist ⟵ {}
  send g to slaves as new basis element
  while B ≠ ∅ and G ≠ ∅ do
    (f, g) ⟵ select element from B with a minimal sugar of the spoly(f, g)
    s ⟵ spoly(f, g)
    while sugar(s) is equal to a currentdegree and pendinglist is not empty do
      send the next S-polynomials with degree(lcm(f, g)) = currentdegree to slaves
      R ⟵ receive reduced S-polynomials from slaves
      while R ≠ ∅ do
        h ⟵ select element from R with a minimal sugar
        R ⟵ R \ {h}
        if h ≠ 0 then
          B ⟵ syzbasis(h, B) and send h to slaves as new basis element
        end if
      end while
      (f, g) ⟵ select element from B with a minimal sugar of the spoly(f, g)
      s ⟵ spoly(f, g)
    end while
    f ⟵ select minimal element from G w.r.t. sugar, G ⟵ G \ {f}
    while sugar(f) is equal to a currentdegree and pendinglist is not empty do
      send the next polynomials from G with degree(f) = currentdegree to slaves
      R ⟵ receive reduced polynomials from slaves
      while R ≠ ∅ do
        h ⟵ select element from R with a minimal sugar
        R ⟵ R \ {h}
        if h ≠ 0 then
          B ⟵ syzbasis(h, B) and send h to slaves as new basis element
        end if
      end while
      f ⟵ select minimal element from G w.r.t. sugar, G ⟵ G \ {f}
    end while
    currentdegree ⟵ currentdegree + 1
  end while
  (GB) ⟵ finalreduce(GB)
  return(GB)
end
```

Fig.3

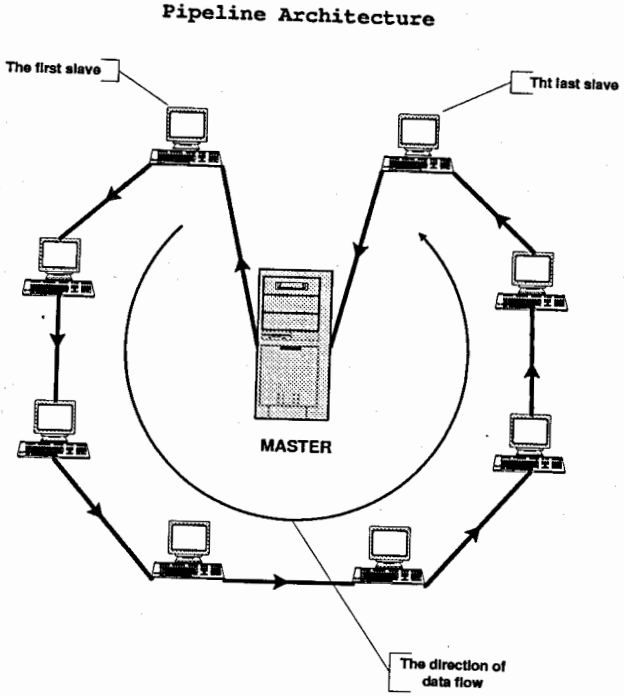
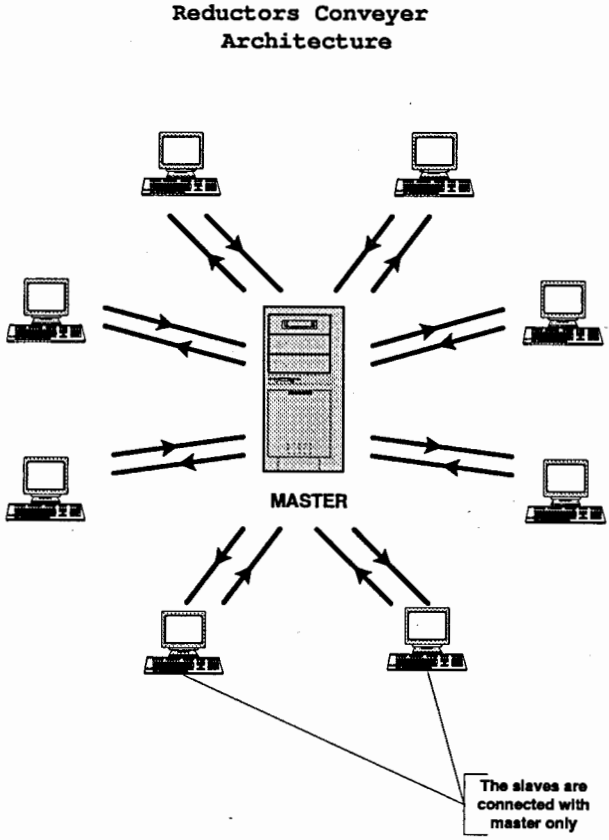

Pipeline Architecture
The first slave
Tht last slave
MASTER
The direction of
data flow

Fig.4


Reductors Conveyer
Architecture
MASTER
The slaves are
connected with
master only

References

[1] A.Giovanni, T.Mora, G. Niesi, L. Robbiano, C. Traverso, One sugar cube, please or Selection Strategies in Buchberger Algorithm, *Proceedings of ISSAC '91, ACM*, 1991, 49-54

[2] Gerdt, V.P., Blinkov, Yu.A., Yanovich, D.A.: Construction of Janet Bases. II. Polynomial Bases. In: *Computer Algebra in Scientific Computing / CASC'01*

[3] J.C.Fauger, Parallelization of Grobner Basis, *Proceedings of PASCO'94*

[4] Kurt Siegl, A Parallel Factorization Tree Grobner Basis Algorithm, *Proceedings of PASCO'94*

[5] Alyson A. Reeves, A Parallel Implementation of Buchberger's Algorithm over Zp for $p \leq 31991$, *J.Symbolic Computation*, 1998, 229-244

[6] Giuseppe Attardi, Carlo Traverso, A strategy-accurate parallel Buchberger algorithm, *Proceedings of PASCO'94*

[7] B. Buchberger, A Criterion for Detecting Unnecessary Reductions in the Construction of Grobner Bases, *Proceedings of the 1979 European Symposium on Symbolic and Algebraic Computation, Lecture Notes in Computer Science 72*, Springer-Verlag, Berlin, Heidelberg, 3-21, 1979

[8] B. Buchberger, Grobner Bases : An Algorithmic Method in Polynomial Ideal Theory, In N. K. Bose, editor, *Multidimensional Systems Theory* , pages 184–232. D. Reidel Publishing Company, 1985

[9] T.Mora, G.Pfister, C.Traverso, An introduction to the tangent cone algorithm, *Advances in Computing research, Issues in Robotics and nonlinear geometry* , 1992, 6, 199-270

[10] R.Gebauer, H.M.Möller, On an Installation of Buchberger's Algorithm, *J. Symbolic Computation*, (1988) 6, 275-286.

[11] V.P.Gerdt, Yu.A.Blinkov, Involutive Polynomial Bases, *Publication IT-271*, Laboratoire d'Informatique Fondamentale de Lille, Lille, 1995

[12] V.P.Gerdt, Yu.A.Blinkov, Involutive Bases of Polynomial Ideals, *Mathematics and Computers in Simulation 45*, 1998, 519-542.

[13] V.P.Gerdt, Yu.A.Blinkov, Minimal Involutive Bases, *Mathematics and Computers in Simulation 45*, 1998, 543-560

[14] Vladimir P. Gerdt, Involutive Division Technique: Some Generalizations and Optimizations, submitted to International Conference "Computer Algebra in Scientific Computing" (April 20-24, St.Petersburg, Russia).

[15] Denis A. Yanovich, On Parallelization of Involutive Janet Bases Construction Algorithm, 2001, submitted to journal *Programmirovanie*

Computer Analysis of Hypergeometric Series: a Project of an Instruction Set Duplicating Operations of the Factorization Method

A.W.Niukkanen, O.S.Paramonova

Vernadsky Institute (GEOKHI),
RAS, Kossygin st. 19, 117975 Moscow, Russia;
e - mail: elkor@geokhi.msk.su; niukkanen@tula.net

The operator factorization method (see, e.g., [1]) greatly facilitating the study of multiple, including simple, hypergeometric series is the main object of our further interest.

The main goal of the paper is to outline a project of a universal "formula sythesizer" in the theory of hypergeometric series. The main idea of the project is to convert the basic operations of the operator factorization method into a complete set of commands serving us as a symbolic manipulation superstructure over a computer algebra system. Presently we do not try to carry out a program implementation of **this** *part of the project (***other** *parts are well underway; see Sec. 1.5 and 4). Our only intention is to present, explicitly, a complete list of* **the main operations** *inherent in the factorization method.*

The research has been supported by Russian Foundation for Basic Research, grant 01-01-00380.

1 Computer analysis and the factorization method

1.1. *Computer algebra or computer analysis?* Being related to mathematical physics by origin and to a great number of problems in a variety of sciences, by application, the hypergeometric series do have most direct relation to mathematical analysis, by the methods used for their study. Computer – aided approach to the study of hypergeometric series relates, obviously, to *computer analysis* rather than to *computer algebra.*

Analysis is the heart of mathematics and the concept of function is the heart of analysis. Functions in pure mathematics are the immaterial entities which are deprived of all properties except those endowed at will of mathematician. As to applied analysis the hypergeometric series serve as a universal substitute for what we call a function. In contrast to their "pure analogs" they show a fantastic abundance of properties endowed by their explicit structure. Therefore any algorithm efficient enough to tackle multiple hypergeometric series of arbitrarily complicated structure can be looked upon as a versatile solver capable to handle with almost any problem relating to functions of practical interest.

1.2. *The central idea of the operator method* is connected with an introduction of a new simple differential operation of *"Ω-multiplication"* $w = u * v$ over the functions $u = u(x_1, \ldots, x_N)$ and $v = v(x_1, \ldots, x_N)$:

$$\langle u * v | x_1, \ldots, x_N \rangle = u\,(d/ds_1, \ldots, d/ds_N)\; v(x_1 s_1, \ldots, x_N s_N)\;|_{\forall s_n = 0} \qquad (1)$$

The fundamental importance of the Ω-multiplication is that it allows any series having complicated structure to be directly expressed through simpler series thus permitting us to use the properties of the simple series to analyze any property of the initial complicated series. Let ${}^N F[A; x_1, \ldots, x_N]$, ${}^N F[B; x_1, \ldots, x_N]$ and ${}^N F[A, B; x_1, \ldots, x_N]$ be the series of the power functions $x_1^{i_1} \ldots x_N^{i_N}/(i_1! \ldots i_N!)$ with coefficients $A(i_1, \ldots, i_N)$, $B(i_1, \ldots, i_N)$ and $A(i_1, \ldots, i_N)B(i_1, \ldots, i_N)$, respectively. The $i_n (n = 1, \ldots, N)$ are summation indices of the series $(i_n = 0, 1, 2, \ldots)$.

The general factorization formula for the series ${}^N F[A, B]$ reads

$$ {}^F N[A, B; x_1, \ldots, x_N] = \langle {}^N F[A] *^N F[B] | x_1, \ldots, x_N \rangle \qquad (2)$$

In short, $F[A \times B] = F[A] * F[B]$. This formula conveys the *property of Ω - representablility* of multiplication operation over coeffifients of an *arbitrary power series.* It shows also that the factorization method can be looked upon as a *disguised form of algebraization of the theory of hypergeometric series.*

1.3. *Conceptual basics of the method.* First, any hypergeometric series is expressed only through hypergeometric series (*closure property*). No necessity in employment of any other auxiliary representation is arisen.

Second, the functional relation $f * f_1 = f * f_2$ will be called Ω-equivalent to the relation $f_1(x_1, \ldots, x_N) = f_2(x_1, \ldots, x_N)$. The concept of *Ω-equivalence* allows *classes* of Ω-equivalent relations to be introduced. In each class a *simplest relation* which will be called a *proto-relation* can be chosen. Having proved the proto-relation we thus *prove all formulas* belonging to the class.

Third, using Ω-multiplication Ω-equivalent operators $F_1 \Leftrightarrow F_2$ can be introduced. They defined by

$$F_1\,(d/ds, s)\,\Psi(xs)\big|_{s=0} = F_2\,(d/ds, s)\,\Psi(xs)\big|_{s=0}, \qquad (3)$$

where Ψ is an arbitrary function. Note that F_1 and F_2 are not necessarily identical to one another. The possibility to substitute F_1 for F_2 in an Ω-product is an inportant technical expedient of the method.

Fourth, by analogy with arithmetically identical expressions the algebraic expressions connected by finite number of arithmetic operations and Ω-multiplication operations will be called *Ω-identical expressions.* Transformation of an expression to an Ω-identical form is another important technical expedient.

1.4. *Three approaches to the use of factorization method.* Altogether we can see three approaches. *The first one is to derive all formulas manually* being dispensed completely with the necessity of working with computer. Even in this case the factorization method offers great advantages over traditional methods. *The second approach implies the manual derivation of basis sets of formulas* with subsequent application of symbolic computer programs for exaustive search of all different combinations of the basis formulas. *The third way consists in full-scale computerization of all operations* inherent in the factorization

method which would open up the possibility for computer–aided derivation of any formula relating to the theory of hypergeometric functions. One of the goals of the present paper is to substantiate, theoretically, the feasibility of the third approach.

1.5. *An experience in development of symbolic computer programs.* All programs developed so far [2, 3] are based on the second (intermediate) approach. These programs utilize not the factorization method *per se*, but the result of its application to a certain class of formulas with the aim to derive manually, without using a computer, a set of basic relationships playing the role of building blocks for a chosen class of formulas. Applying such a "bounded–universal" procedure to one or another of several score classes of formulas making up the backbone of the theory of hypergeometric series one can obtain, in principle, sufficiently complete computerized version of the theory indirectly connected with the factorization method. The initial steps in this direction covering the transformation theory of hypergeometric series are made, with sufficient completeness, in the programs announced in [2, 3].

The notable success of this approach was corroborated by the computer–aided derivation [4] of an important reduction formula given by Gelfand at al. [5] and its numerous non-trivial generalizations.

1.6. *An outline of the project.* The accumulated experience suggests the desirability of expanding the programs' potentialities, in the spirit of the third approach (see Section 1.4).

The central part of the core (CPC) of the projected program complex is conceived to perform, directly, all operations of the operator factorization method thus placing at the user's disposal a sort of a universal interactive "formula synthesizer".

The peripheral part of the core (PPC) is planned to consist of macro-commands implementing the sets of basic relations obtained with the help of the CPC. Many of these relations have been already obtained manually [4, 6]. Very large sets of relations can be obtained with the help of the PPC in an automatic mode.

If the relations presenting little interest for derivation of new formulas are consided to be valuable on their own they will be placed at date base surrounding (DBS). The DBS will play the role of an information reference system. The DBS is expected to be of moderate size for the main bulk of relations will be generated in PPC.

1.7. *How the method would work in the program.* The functionality of the method in the program would not differ much from the work using hand and pencil. There are two general schemes for carrying out calculations with the help of operator factorization method.

The first scheme consists of four steps.

Step 1 (analysis) breaks up a series into an Ω-product of simpler series. **Step 2 (simple series transformations)** utilizes the known properties of the simpler series for transformation of factorized terms. **Step 3 (auxiliary transformations)** uses a finite set of auxiliary identities converting the resultant expression into a form allowing application of a (possible new) factirization formula. **Step 4 (synthesis)** transforms the operator expression into an algebraic form with the help of a suitable factorization formula.

The second scheme is based on the concept of Ω-equivalence (see Section 1.3). Introducing a simple relation we can multiply its both sides by an Ω-factor thus obtaining a new relation.

Despite the seemingly exotic nature of these approaches, they prove in practice to be

quite simple, universal and effective, and thus fairly suitable for the role of a superstructure over one of the existing analytical manipulation systems.

2 Instruction set underlying the central part of the core (CPC) of the proposed program complex

The instruction set duplicates the main operations of the factorization method. Each *definition* of a formula giving us a project of a future entry of the instruction set is supplied with *capital* boldface label. Sometimes mnemonic synonyms of the instructions names are indicated in parentheses. The numbers attached to the labels of kindred formulas are to be subsituted, in course of the program implementation, by exact specific indications of the commands formats.

The *references* to the definitions will be denoted by *lower case* boldface labels. The whole set of formulas given below constitutes a project of a specialized algorithmic language. This language may give an origin to a program complex aimed at a full-scale computerization of a substantial part of applied mathematical analysis. For convenience, all formulas are subdivided into several classes.

Sometimes, if a formula written for the case of one variable can be extended directly to the case of several variables we do not present the latter explicitly.

For notation see Refs. [1] and [13].

2.1. *Factorization formulas*

FACT1. Factorization of the series $F[\mathbf{d}; x]$

$$F[\mathbf{d}_1\,;\, x_1\, d(s)]\, F[\mathbf{d}_2; x_2 s]|_{s=0} = F[\mathbf{d}_1, \mathbf{d}_2\,;\, x_1\, x_2]\,, \qquad d(s) = d/ds \tag{4}$$

FACT2. Factorization of the series in one variable containing compound parameter

$$F[\mathbf{d}_1\,;\, x_1\, d(s)]\, F[\mathbf{d}_2; x_2 s^m]|_{s=0} = F[< \mathbf{d}_1 \mid m >, \mathbf{d}_2\,;\, x_1^m x_2] \tag{5}$$

FACT3. General factorization of the series ${}^N F$

$$\begin{aligned} &{}^N F[L_1,\, L_2\,;\, x_1, \ldots, x_N] \\ &\quad =^N F\,[L_1\,;\, d(s_1), \ldots, d(s_N)]\, {}^N F[L_2\,;\, x_1 s_1, \ldots, x_N s_N]\,\big|_{\forall s_n = 0} \end{aligned} \tag{6}$$

FACT4. Special factorization of the series ${}^N F$

$$\begin{aligned} &{}^N F[< \mathbf{d} \mid m_1, \ldots, m_N >, L\,;\, x_1, \ldots, x_N] \\ &\quad = F\,[\mathbf{d}\,;\, d(s)]\, {}^N F[L\,;\, x_1 s^{m_1}, \ldots, x_N s^{m_N}]\,\big|_{s=0} \end{aligned} \tag{7}$$

FACT5. Factorization of ${}^N F$ containing the glueing operator $F[\mathbf{d}_0\,;\, x d^m(s)]$

$$\begin{aligned} &F[\mathbf{d}_0\,;\, x d^m(s)]\, F[\mathbf{d}_1\,;\, x_1 s^m] \cdots F[\mathbf{d}_N\,;\, x_N s^m]|_{s=0} \\ &\quad = F\left[\mathbf{d}_0, \frac{1}{m}, \ldots, \frac{m-1}{m}\,:\, \mathbf{d}_1\,;\, \ldots\,;\, \mathbf{d}_N\,;\, x x_1 m^m, \ldots, x x_N m^m\right] \end{aligned} \tag{8}$$

FACT6. Factorization of multiple series containing constant arguments

$$\Omega =^{N+P} F[\langle \mathrm{d}|m_1, \dots, m_{N+P}\rangle, \dots : x_1\frac{d}{ds_1}, \dots, x_N\frac{d}{ds_N}, u_1, \dots, u_P]$$

$$\times^{N+Q} F[\langle \mathrm{d}'|l_1, \dots, l_{N+Q}\rangle, \dots : y_1 s_1, \dots, y_N s_N, v_1, \dots, v_Q]\,|_{\forall s_n=0} \tag{9}$$

$$=^{N+P+Q} F[\langle \mathrm{d}|m_1, \dots, m_N, m_{N+1}, \dots, m_{N+P}, \underbrace{0, \dots, 0}_{Q}\rangle,$$

$$\langle \mathrm{d}'|l_1, \dots, l_N, \overbrace{0, \dots, 0}^{P}, l_{N+1}, \dots, l_{N+Q}\rangle, \dots :$$

$$x_1 y_1, \dots, x_N y_N, u_1, \dots, u_P, v_1, \dots, v_Q] \tag{10}$$

2.2. *General properties of the Ω-multiplication operation*

OMEGA 1. Commutativity property $u * v = v * u$ **(COMM)**

$$u(d(s))v(xs)|_{s=0} = v(d(s))u(xs)|_{s=0} \tag{11}$$

OMEGA 2. Coupling rule **(COUP)**

$$u(d(s))v(xs)|_{s=0} = u(xd(s))v(s)|_{s=0} \tag{12}$$

OMEGA 3. Associativity property **(ASSOC)**

$$< w * (u * v)|x > = < (w * u) * v|x > \tag{13}$$

OMEGA 4. $\exp(x)$ plays the role of Ω-unit **(OMUN)** that is $\exp * f = f * \exp = f$, or

$$\exp(d(s))f(xs)|_{s=0} = f(d(s))\exp(xs)_{s=0} = f(x) \tag{14}$$

OMEGA 5. The Ω-"unitarity" can be interpreted as "renaming" (s for x) property **(REN)**

$$\exp xd(s)f(s)|_{s=0} = f(x) = f(x)|_{s\Rightarrow x} \tag{15}$$

2.3. *Ω-equivalent operators*

The case of **EQUIV1.** an arbitrary operator multiplied by power function

$$F(d(s))s^n\,\Psi(s)|_{s=0} = F^{(n)}(d(s))\,\Psi(s)|_{s=0} \tag{16}$$

The case of **EQUIV2.** an arbitrary operator multiplied by exponential function

$$F(d(s))e^{xs}\,\Psi(s)|_{s=0} = F(d(s)+x)\,\Psi(s)|_{s=0} \tag{17}$$

The case of **EQUIV3.** the binomial operator multiplied by exponential function

$$F_0^1[a; d(s)]e^{xs}\,\Psi(s)|_{s=0} = (1-x)^{-a}F_0^1\,[a; d(s)/1-x]\;\Psi(s)\big|_{s=0} \tag{18}$$

2.4. *Relationships containing operators (without setting differentiation variable to zero)*

OPER1. Shift operator identity **(SHIFT)**

$$\exp(u\,d(x))f(x) = f(x+u) \tag{19}$$

OPER2. Similarity transformation (F and f are arbitrary functions, A is an arbitrary operator) **(SIMIL)**

$$f^{-1}\,F(A)\,f = F(f^{-1}A\,f) = F(A + f^{-1}[A, f]) \tag{20}$$

OPER3. Operator argument displacement formula **(DISP)**

$$\exp(-vx)F[d(x)]\exp(vx) = F[d(x) + v] \tag{21}$$

OPER4. Applying of a differential operator to $\exp(x)$ **(OPEXP)**

$$F(d(x))\,\exp(ux) = F(u)\,\exp(ux) \tag{22}$$

OPER5. Generalized Leibnitz rule **(LEIB)**

$$F(d(x))\,f_1(x)f_2(x) = F(d(x_1) + d(x_2))f_1(x_1)f_2(x_2)|_{x_1=x_2=x} \tag{23}$$

OPER6. Differentiation of simple hypergeometric series **(DIFHYP)**

$$d^n(x)\,F[\mathbf{d};\,ux] = u^n(\mathbf{d}, n)\,F[\mathbf{d} + n;\,ux] \tag{24}$$

2.5. *Elementary reduction formulas*

RED1. Reduction of the exponential series F_0^0 **(REDEXP)**

$$F_0^0[*//*; x] = \exp(x) \tag{25}$$

RED2. Reduction of the binomial series F_0^1 **(REDBIN)**

$$F_0^1[a//*; x] \equiv F[a; x] = (1 - x)^{-a} \tag{26}$$

RED3. Reduction of an infinite geometrical progression **(GEOINF)**

$$1 + x + x^2 + \cdots \equiv F_0^1[1; x] = (1 - x)^{-1} \tag{27}$$

RED4. Reduction of the finite geometrical progression **(GEOFIN)**

$$1 + x + x^2 + \cdots + x^N = (1 - x^{N+1})/(1 - x) \tag{28}$$

RED5. Reduction of the series ${}^NF$ with empty glueing set

$${}^NF[*//* : \mathbf{d}_1; \ldots; \mathbf{d}_N;\, x_1, \ldots, x_N] = F[\mathbf{d}_1;\, x_1] \cdots F[\mathbf{d}_N;\, x_N] \tag{29}$$

RED6. Reduction of the series ${}^NF$ with empty individual sets

$${}^NF[\mathbf{d} : *; \ldots; *;\, x_1, \ldots, x_N] = F[\mathbf{d}\,;\, x_1 + \cdots + x_N] \tag{30}$$

2.6. *Auxiliary algebraic identities*

ALG1 Gauss-Legendre multiplication formula for the Pochhammer symbol **(MULT(m))**

$$(\alpha, mi) = m^{mi}\left(\frac{\alpha}{m}, i\right)\left(\frac{\alpha+1}{m}, i\right)\cdots\left(\frac{\alpha+m-1}{m}, i\right) \tag{31}$$

ALG2. Inversion formula for Pochhammer symbol **(INVER)**

$$(a, -I) = (-1)^I (1-a, I)^{-1} \tag{32}$$

ALG3. Cancellation or, *vica versa*, introduction of equal parameters in numerator and denominator of a series **(CANC, INTRO)**

$$F[\mathbf{d}; x] = F[a, \mathbf{d}//a; x] \tag{33}$$

ALG4. Vertical transfer of parameters **(VERT)**

$${}^N F[\langle a|m_1, \ldots, m_N\rangle, L; \mathbf{x}] = {}^N F[L//\langle 1-a|\bar{m}_1, \ldots, \bar{m}_N\rangle; (-1)^{\mathbf{m}}\mathbf{x}], \tag{34}$$

$\mathbf{x} = [x_1, \ldots, x_N]$, $(-1)^{\mathbf{m}}\mathbf{x} = [(-1)^{m_1}x_1, \ldots, (-1)^{m_N}x_N]$

ALG5. Any series ${}^N F$ is symmetric with respect to simultaneous permutation of arguments $x_i \rightleftharpoons x_j$, individual sets of parameters $\mathbf{d}_i \rightleftharpoons \mathbf{d}_j$ and all corresponding spectral numbers $m_i \rightleftharpoons m_i$, $l_i \rightleftharpoons l_j$, etc. **(PERM)**

ALG6. Uniformization of the argument of the binomial series $F_0^1[a; x+u]$ **(UNIF)**

$$F_0^1[a; x+u] = (1-x-u)^{-a} = (1-u)^{-a} F_0^1[a; x/(1-u)] \tag{35}$$

ALG7. Factorization of geometrical progression $F_0^1[1; x]$ into a product of two progressions **(PROG)**

$$F_0^1[1; x] = \left(\sum_{r=0}^{N-1} x^r\right) F_0^1[1; x^N] \tag{36}$$

ALG8. Decomposition of e^x into a sum of N series $F_{N-1}^0(x^N)$ **(DEXP(N))**

$$e^x = \sum_{r=0}^{N-1} \frac{x^r}{r!} F\left[\begin{matrix} 1; x^N \\ \langle 1+r|N\rangle \end{matrix}\right] = \sum_{r=0}^{N-1} \frac{x^r}{r!} F\left[\begin{matrix} *; & (x/N)^N \\ \frac{1+r}{N}, \cdots, & \frac{N-1}{N}, \frac{N+1}{N}, \cdots, \frac{N+r}{N} \end{matrix}\right] \tag{37}$$

ALG9. Addition formula for binimial series **(ADDBIN)**

$$F_0^1[a; x_1+x_2] = \sum_{n=0}^{\infty} \frac{(a,n)}{n!} x_1^n F_0^1[a+n; x_1] x_2^n F_0^1[a+n; x_2] \tag{38}$$

3 Examples suggestive of functionality specifics of the proposed CPC commands

3.1. *An elementary example* can be seen from comparison between relationships **prog** and **dexp(N)** which prove to be Ω-equivalent one to another! The elementary formula **prog** follows from the formulas **geofin** and **geoinf**. Thus it is just the **prog** plays the role of proto-relation (see Sec. 1.3). Applying the operator $F_1^0[*//1; z\,d(x)]|_{x=0}$ to the both parts of **prog** we use the operations **fact1, canc** and **redexp** in the left-hand part and the operations **equiv1, difhyp, fact2, mult(m)** and **canc** in the right-hand part. Then the **dexp(N)** follows immediately. More general relations belonging to this class can be

obtained in analogous way if we apply $F[\mathbf{d};\, zd(x)]|_{x=0}$, instead of the F_1^0, to the both parts of **prog**.

3.2. *Already published examples.* In fact, we are delivered from the necessity of giving many examples of how the method could work in practice. Numerous examples of the kind are given in the already published papers [1], [7]-[13]. Many simple examples illustrating application of operations (4) - (38) are given is Ref. [1]. The same operations were emlpoyed for derivation of new recurrence relations [7]. Some new generating functions for the Laguerre polynomials were presented in Ref. [8]. More general generating functions, as well as a complete set of Meixner-type formulas and a new class of Lagrangean polynomials were introduced in [9]. Very important special transformations of the Appel F_4 and the Horn H_1 and G_2 functions were obtained in [10]. A new approach to derivation and generalization of involved Burchnall and Chaundy expansions playing a particularly important role in the theory of double hypergeometric series was found in [11]. A sophisticated analysis of many particular problems originated from contemplation of a classical relation between Bessel functions was given in [12]. A heavy use of the operations (4) - (38) was made in [13] for analysis of linearization relations and addition formulas including a generaliztion of an important Koornwinder formula of the Jacobi polynomials. Special attention has been given in Ref. [13] to the details of the new technology of analytical transformations based on the operations (4) - (38).

3.3. *An additional example.* The references given in Sec. 3.2 relate mostly to the main scheme involving the four steps mentioned in Sec. 1.7. The second scheme based on the concept of Ω-equivalency was paid lesser attention in the above examples and needs therefore a little bit more substantiation. To this end we introduce the notation

$$u_1 = 1 - z + \xi_1 z, \quad u_2 = 1 - z + \xi_2 z$$

and consider the elementary relationship

$$\mathcal{L} \equiv F_0^1[c;\, (1-\xi_1)(1-\xi_2)z] = (1-z)^c (u_1 u_2)^{-c}\, F_0^1[c;\, \xi_1\xi_2 z/u_1u_2] \equiv \mathcal{R}. \tag{39}$$

which is readily verified by using the reduction rule **redbin**. We then transform the proto-relation (39) to an Ω-identical form facilitating transition to Ω-equivalent relations. Using twice the operation **fact1** we get the preliminary Ω-identical (see Sec. 1.3) transformation

$$F_0^1\left[c;\, \frac{\xi_1\xi_2 z}{u_1u_2}\right] = \prod_{n=1}^{2} F_0^1\left[c;\, \frac{\xi_n(1-z)}{u_n}\frac{d}{ds_n}\right]\, F_1^0\left[\begin{matrix}*;\\ c\end{matrix}\, \frac{s_1s_1z}{(1-z)^2}\right]\Bigg|_{s_1=s_2=0} \tag{40}$$

To simplify dependence on ξ_1, ξ_2 we multiply eq. (40) by $u_1^{-c}u_2^{-c}$ (see eq. (39)), transform the both operator series F_0^1 with the help of **redbin**, allow for definitions of u_1, u_2, make some elementary algebraic manipulations, use, inversely, the **redbin** and employ **equiv2**. Thus we have

$$\begin{aligned} u_n^{-c}\, F_0^1\left[c;\, \frac{\xi_n(1-z)}{u_n}\frac{d}{ds_n}\right] &= (1-z)^{-c}\, F_0^1\left[c;\, \xi_n\left(\frac{d}{ds_n} + \frac{z}{z-1}\right)\right] \\ &\Leftrightarrow (1-z)^{-c}\, F_0^1[c;\, \xi_n d(s_n)]\exp[zs_n(z-1)^{-1}]. \end{aligned} \tag{41}$$

Inserting (40) and (41) into (39) we finally have the desired Ω-identical representation of

$\mathcal{R}$:

$$\mathcal{R} = (1-z)^{-c}\, F_0^1[c;\, \xi_1 d(s_1)] F_0^1[c;\, \xi_2 d(s_2)] \\ \times \exp\left(\frac{zs_1}{z-1}\right) \exp\left(\frac{zs_2}{z-1}\right) F_1^0 \left[\begin{matrix} * \,;\, \dfrac{s_1 s_2 z}{(1-z)^2} \\ c \end{matrix}\right]\Bigg|_{s_1=s_2=0} . \tag{42}$$

With the help of **redbin** the left-hand side $\mathcal{L}$ can be writhen as:

$$\mathcal{L} = \sum_{n=0}^{\infty} \frac{(c,n}{n!}\, F_0^1[-n, \xi_1] F_0^1[-n, \xi_2]\, z^n . \tag{43}$$

Then we apply the operator

$$F_1^0[*//c;\, x_1 d(\xi_1)] F_1^0[*//c;\, x_2 d(\xi_2)]\,|_{\xi_1=\xi_2=0}$$

to the both sides of the identity $\mathcal{L} = \mathcal{R}$ where $\mathcal{L}$ and $\mathcal{R}$ are given by eqs. (42) and (43) respectevely. In case of the $\mathcal{L}$ the only operation **fact1** is needed. In case of the $\mathcal{R}$ we apply **fact1, canc, redexp** and **ren**, consecutively. The result

$$\sum_{n=0}^{\infty} \frac{(c,n)}{n!}\, F_1^1 \left[\begin{matrix} -n\,;\, x_1 \\ c \end{matrix}\right] F_1^1 \left[\begin{matrix} -n\,;\, x_2 \\ c \end{matrix}\right] z^n \\ = (1-z)^{-c} \exp\left[\frac{(x_1+x_2)z}{z-1}\right] F_1^0 \left[\begin{matrix} *\,;\, \dfrac{x_1 x_2 z}{(1-z)^2} \\ c \end{matrix}\right] \tag{44}$$

is equivalent to the Hille-Hardi bilinear generating function for Laguerre polynomials (see [14], vol.1).

Applying to the both sides of eq. (44) the operator product

$$F_0^1[a_1;\, \xi_1 d(x_1)] F_0^1[a_2;\, \xi_2 d(x_2)]|_{x_1=x_2} ,$$

using twice **fact1** on the left and **equiv3** (see eq. (41)) and **fact1** on the right we obtain the apparently new bilinear generating functions for Gaussian polynomials $F[-n, a//c;\, \xi]$:

$$\sum_{n=0}^{\infty} \frac{(c,n)}{n!}\, F_1^2 \left[\begin{matrix} -n, a_1\,;\, \xi_1 \\ c \end{matrix}\right] F_1^2 \left[\begin{matrix} -n, a_2\,;\, \xi_2 \\ c \end{matrix}\right] z^n \\ = (1-z)^{a_1+a_2-c}\, u_1^{-a_1} u_2^{-a_2} F[a_1, a_2//c;\, \xi_1\xi_2 z/u_1 u_2] . \tag{45}$$

An attempt to apply eq.(45) to the Gegenbauer polynomials $C_n^\lambda(x)$ may seem to make no sense whatever because any of the known hypergeometric representations of $C_n^\lambda(x)$ in the form of F_1^2 contains *two* parameters dependent on n whereas each of the F_1^2 in eq. (45) contains only *one* such parameter. Being sure that the list of representations for the $C_n^\lambda(x)$ given in literature is not complete we looked into a question of how many different formulas for the $C_n^\lambda(x)$ may exist. We used the linear ($x \to x^{-1}$, $x \to 1-x$, $x \to x/(x-1)$) and quadratic transformations (see [14], vol. 2 and the Sec. 4.2 below) conserving the polynomial structure of the transformed functions. We found altogether 18 different representations. The formula

$$C_n^\lambda(x) = \frac{(2\lambda, n)}{n!} [x + (x^2-1)^{1/2}]^n\, F_1^2 \left[\begin{matrix} -n, \lambda\,;\, \dfrac{2(x^2-1)^{1/2}}{x+(x^2-1)^{1/2}} \\ 2\lambda \end{matrix}\right] \tag{46}$$

which is incidentally absent in literature is of prime interest for applications.

This formula is remarkable for that the dependence on the order n of the polynjmial $F[-n, \lambda//2\lambda]$ in the definition (46) in the same as in the case of the polynomials $F[-n//\alpha+1]$ occuring in the definition of the Laguerre polynomials. The structural similarity of the $C_n^\lambda(x)$ and L_n^α shows that solution of the problems where dependence on n plays an essential role would have, formally, much in common for the $C_n^\lambda(x)$ and L_n^α despite the fact that the Gegenbauer polynomials are a natural particular case of the Jacobi polynomials which have nothing to do with the Laguerre polynomials. Beyond the factorization method this formal observation would hardly be of any significance. On the contrary, within the factorization method the simple observation gives us a powerful tool for obtaining new interesting results. For example, letting $a_1 = a_2 = \lambda$, $c = 2\lambda$ in eq. (45) and using eq. (46) to express the resultant F_1^2 polynomials through $C_n^\lambda(x)$ we can readily obtain a seemingly new bilinear generating function for the Gegenbauer polynomials. Derivation of bilateral, for the L_n^α and $C_n^\lambda(x)$, generating function can be also performed with ease.

4 Examples of macro-commands constituiting the peripheral part of the core (PPC) of the proposed program complex

As was already mentioned above (see Sec. 1.5 and 1.6) along with the universal set of "low-level" derivation rules (4) - (38) we are going to use the generators of formula classes consisting of a few specialized "high-level" basic relationships. In distinction to the CPC operations many of the PPC macro-commands have been already programmed [2, 3] and applied for analysis of multiple hypergeometric series [4, 15].

We first give typical instances of macro-commands and then present some examples of using these macro-commands (all necessary definitions and notation are given in the refs. [1, 13]).

4.1. *Linear transformations*

LIN(K). Linear transformation connecting two series having the Kummer type (1//1) with respect to x_0 (the L^* symbolizes the coefficients independent of summation index i_0):

$$F\left[\begin{matrix}\langle\nu_1|1,\mathbf{m}_1\rangle\,,\ L^*;\ x_0,\mathbf{x}\\ \langle\nu_0|1,\mathbf{m}_0\rangle\end{matrix}\right] = e^{x_0}\,F\left[\begin{matrix}\langle\nu_{0\bar{1}}|1,\mathbf{m}_{0\bar{1}}\rangle, \langle\nu_1|0,\mathbf{m}_1\rangle,\ L^*;\ -x_0,\mathbf{x}\\ \langle\nu_0|1,\mathbf{m}_0\rangle\ , \langle\nu_{0\bar{1}}|0,\mathbf{m}_{0\bar{1}}\rangle\end{matrix}\right] \tag{47}$$

LIN(G). Three linear transformations linking the series having the Gauss type (2//1) with respect to x_0. For each series we use a canonical representation [1, 13], where all spectral numbers connected with x_0 are equal to 1. The three transformations change, consecutively, the first (G01), the second (G02) and the both (G00) numerator parameters. The symbol following G in (GOQ) is the number of argument.

$$F\left[\begin{matrix}\langle\nu_1|1,\mathbf{m}_1\rangle\,,\ \langle\nu_2|1,\mathbf{m}_2\rangle\,,\ L^*;\ x_0,\mathbf{x}\\ \langle\nu_0|1,\mathbf{m}_0\rangle\end{matrix}\right] =$$

LIN(G01) $= L_1^0 F \equiv (1-x_0)^{-\nu_2} \times$

$$\times F\left[\begin{matrix}\langle \nu_{0\bar{1}}|1,\mathbf{m}_{0\bar{1}}\rangle\,,\ \langle\nu_2|1,\mathbf{m}_2\rangle\,,\ \langle\nu_1|0,\mathbf{m}_1\rangle\,,\ L^*;\ \dfrac{x_0}{x_0-1},\dfrac{\mathbf{x}}{(1-x_0)^{\mathbf{m}_2}}\\ \langle\nu_0|1,\mathbf{m}_0\rangle\,,\ \langle\nu_{0\bar{1}}|0,\mathbf{m}_{0\bar{1}}\rangle\end{matrix}\right] \tag{48}$$

LIN(G02) $= L_2^0 F \equiv (1-x_0)^{-\nu_1} \times$

$$\times F\left[\begin{matrix}\langle \nu_1|1,\mathbf{m}_1\rangle\,,\ \langle\nu_{0\bar{2}}|1,\mathbf{m}_{0\bar{2}}\rangle\,,\ \langle\nu_2|0,\mathbf{m}_2\rangle\,,\ L^*;\ \dfrac{x_0}{x_0-1},\dfrac{\mathbf{x}}{(1-x_0)^{\mathbf{m}_1}}\\ \langle\nu_0|1,\mathbf{m}_0\rangle\,,\ \langle\nu_{0\bar{2}}|0,\mathbf{m}_{0\bar{2}}\rangle\end{matrix}\right] \tag{49}$$

LIN(G00) $= L_0^0 F \equiv (1-x_0)^{\nu_{0\bar{1}\bar{2}}} \times$

$$\times F\left[\begin{matrix}\langle \nu_{0\bar{1}}|1,\mathbf{m}_{0\bar{1}}\rangle,\ \langle\nu_{0\bar{2}}|1,\mathbf{m}_{0\bar{2}}\rangle,\ \langle\nu_1|0,\mathbf{m}_1\rangle\,,\langle\nu_2|0,\mathbf{m}_2\rangle\,,\ L^*;\ x_0,\mathbf{X}\\ \langle\nu_0|1,\mathbf{m}_0\rangle,\ \langle\nu_{0\bar{1}}|0,\mathbf{m}_{0\bar{1}}\rangle,\ \langle\nu_{0\bar{2}}|0,\mathbf{m}_{0\bar{2}}\rangle\end{matrix}\right], \tag{50}$$

$$\mathbf{X} = \mathbf{x}(1-x_0)^{-\mathbf{m}_{\bar{0}12}}.$$

4.2. *Quadratic transformations*

The "classical" theory of quadratic transformations even in case of simple series lacks simplicity and transparency of structure to say nothing of multiple case. The relationships which follow are applicable to *any multiple series* satisfying some necessary conditions. Moreover instead of 9 functions we can confine ourselves, at the first step, but to 3 functions:

$$F_1 \equiv F_1[\langle\nu_1|1,\mathbf{m}_1\rangle, \langle\nu_1+1/2|1,\mathbf{m}_1\rangle, L^*//\langle\nu_0|1,\mathbf{m}_0\rangle; x_0, \mathbf{x}], \tag{51}$$

$$F_2 \equiv F_2[\langle\nu_1|1,\mathbf{m}_1\rangle, \langle\nu_2|1,\mathbf{m}_2\rangle, L^*//\langle 1+\nu_{1\bar{2}}|1,\mathbf{m}_{1\bar{2}}\rangle; x_0, \mathbf{x}], \tag{52}$$

$$F_3 \equiv F_3[\langle\nu_1|1,\mathbf{m}_1\rangle, \langle\nu_2|1,\mathbf{m}_2\rangle, L^*//\langle 2\nu_2|1,2\mathbf{m}_2\rangle; x_0, \mathbf{x}]. \tag{53}$$

Just these functions occur in the following three basic quadratic transformations:

QUAD32. The transformation relating F_3 to F_2 is

$$F_3 = [(2/(2-x_0)]^{\nu_1} F_1\left[\begin{matrix}\langle\frac{\nu_1}{2}|1,\frac{\mathbf{m}_1}{2}\rangle, \langle\frac{\nu_1+1}{2}|1,\frac{\mathbf{m}_1}{2}\rangle, L^*; x_{13}, \mathbf{x}_{13}\\ \langle\nu_2+1/2|1,\mathbf{m}_2\rangle\end{matrix}\right] \tag{54}$$

$$x_{13} = x_0^2(2-x_0)^{-2},\ \mathbf{x}_{13} = 4^{\mathbf{m}_{12}}\mathbf{x}\,(2-x_0)^{-\mathbf{m}_1}.$$

QUAD21. The transformation linking F_2 and F_1 has the form

$$F_2 = (1+x_0)^{-\nu_1} F_1\left[\begin{matrix}\langle\frac{\nu_1}{2}|1,\frac{\mathbf{m}_1}{2}\rangle, \langle\frac{\nu_1+1}{2}|1,\frac{\mathbf{m}_1}{2}\rangle, \langle\nu_2|0,\mathbf{m}_2\rangle, L^*; x_{12}, \mathbf{x}_{12}\\ \langle 1+\nu_1-\nu_2|1,\mathbf{m}_1-\mathbf{m}_2\rangle\end{matrix}\right] \tag{55}$$

$$x_{12} = 4x_0(1+x_0)^{-2},\ \mathbf{x}_{12} = 2^{\mathbf{m}_1}\mathbf{x}(1+x_0)^{-\mathbf{m}_1}.$$

QUAD32. The transformation expressing F_3 through F_2 is

$$F_3 = [2/(1+\sqrt{1-x_0})]^{2\nu_1} F_2\left[\begin{matrix}\langle\nu_1|1,\mathbf{m}_1\rangle, \langle\nu_{1\bar{2}}+\frac{1}{2}|1,\mathbf{m}_{1\bar{2}}\rangle, L^*; x_{23}, \mathbf{x}_{23}\\ \langle\nu_2+1/2|1,\mathbf{m}_2\rangle\langle\nu_{1\bar{2}}+\frac{1}{2}|0,\mathbf{m}_{1\bar{2}}\rangle\end{matrix}\right] \tag{56}$$

$x_{23} = (1-\sqrt{1-x_0})^2(1+\sqrt{1-x_0})^{-2}$, $\mathbf{x}_{23} = 4^{\mathbf{m}_{12}}\mathbf{x}(1+\sqrt{1-x_0})^{-2\mathbf{m}_1}$.

All other quadratic transformations follow from eqs. (54)-(56) by using the three operations **lin(G)** for F_1, F_2, F_3 occuring in (54)-(56). This adds 6 new functions. Letting

$N = 0$ we thus obtain a complete systematic set of quadratic transformations of the Gauss function F_1^2.

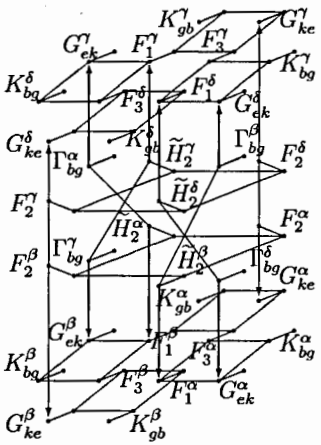
4.3. *Examples.* Many interesting examples can be found in ref. [4] where the macro-commands **lin(G)** were applied to Gelfand functions on grassmanians $G_{2,4}$ and $G_{3,6}$. These functions depend on three and four variables, respectively. The linear transformations permitted us to use a new algorithm of finding reducible cases of these functions. The idea of the algorithm lies in transforming the functions to the form allowing one out of 6 *elementary* self-explanatory reduction rules to be used. The list of the elementary reductions is given in ref. [4].

To conclude with, we give, without going into details, the results of computer analysis of the special Appell function $F_4[a_1, a_2, a_1, b_2; x_1, x_2]$. We first used a representation of the general F_4 function through a complete series of the third order [10] and then, confining ourselves to the special case, we obtained two different expressions of the general F_4 in the form of the following non-Hornian functions:

$$K_{gb} = F\begin{bmatrix} \alpha : a_1, a_1' \,;\, a_2 \,;\, x_1, x_2 \\ \beta : \quad b_1 \quad ;\, * \end{bmatrix}, \tag{57}$$

$$\Gamma_{bg} = F\begin{bmatrix} \langle \alpha_1 | 1, \bar{1} \rangle \langle \alpha_2 | \bar{1}, 1 \rangle : a_1 \,;\, a_2, a_2' \,;\, x_1, x_2 \\ : * \,;\quad b_2 \end{bmatrix}. \tag{58}$$

The processing of these functions consisted in using all possible linear commands **lin(G)** along with an auxiliary bilinear transformation applicable to the functions containing an "indefinite" parameter $\langle 0|1, \bar{1}\rangle$. The process performed in automatic mode gave us the following 5 functions:



$$G_{ek} = F\begin{bmatrix} \alpha_1, \alpha_2 : * \,;\, a_2 \,;\, x_1, x_2 \\ \beta \quad : * \,;\, b_2 \end{bmatrix} \tag{59}$$

$$F_1 = F\begin{bmatrix} \alpha : a_1 \,;\, a_2 \,;\, x_1, x_2 \\ \beta : * \,;\, * \end{bmatrix} \tag{60}$$

$$F_2 = F\begin{bmatrix} \alpha : a_1 \,;\, a_2 \,;\, x_1, x_2 \\ * : b_1 \,;\, b_2 \end{bmatrix} \tag{61}$$

$$F_3 = F\begin{bmatrix} * : a_1, a_1' \,;\, a_2, a_2' \,;\, x_1, x_2 \\ \beta : \quad * \quad ;\quad * \end{bmatrix} \tag{62}$$

$$\widetilde{H}_2 = F\begin{bmatrix} \langle \alpha | \bar{1}, 1 \rangle : a_1, a_1' \,;\, a_2 \,;\, x_1, x_2 \\ : \quad * \quad ;\, b_2 \end{bmatrix} \tag{63}$$

The result of all transformation is represented at the diagram. Black nodes at the ends of short segments denote arguments of the double series. Long lines symbolize linear transformations. For more details see Ref.[15].

5 Concluding remarks

The program implementation of the "central part of the core" (see Sec. 2) would allow us, instead of using a "manual", with pen and paper, calculation technique, to perform all analytical transformations with the help of computer in an interactive mode, in this way relieving the researcher of the tedious copying of cumbersome formulas and placing at his disposal a universal "formula synthesizer" of a sort.

An addition of new macro-commands to the "peripheral part of the core" (see Sec. 4) would give us an access to hundreds and thousands of new relationships, whose publication in the traditional form of books and periodicals would hardly have been practicable.

An inappropriately cumbersome user interface may well happen to become a substantial practical obstacle to the effective program realization of the global approach. A detailed look at the work with the formulas' "screen images" and sophisticated investigation of different variants of its organization seems to be obligatory condition for the effective man–machine formula interface. We hope that recourse to "semantics-oriented" tools, like XML, OpenMath, etc., may help us to solve the problem.

References

[1] A.W.Niukkanen A new theory of multiple hypergeometric series and its prospects for computer algebra programming (in Russian), *Fundamentalnaya i prikladnaya matematika*, 1999, **5**, 1–29.

[2] O.S.Paramonova, A.W.Niukkanen, Analytical transformations of hypergeometric series (in Russian), *Programmirovanie*, 1998, No 6, 25–26.

[3] O.S.Paramonova, A.W.Niukkanen, New algorithms of finding reduction formulas of multiple hypergeometric series (in Russian), *Programmirovanie*, 2000, No 1, 62–63.

[4] A.W.Niukkanen, O.S.Paramonova, Computer generation of complicated transformations for multiple hypergeometric series, *Comput. Phys. Commun.*, 2000, **126**, 141–148.

[5] I.M. Gel'fand, M.I. Graev, V.S. Retakh, General hypergeometric equations systems and hypergeometric type series (in Russian), *Uspekhi Matem. Nauk*, 1992, **47**, 3–82.

[6] A. W. Niukkanen, Transformation theory of multiple hypergeometric series and computer aided symbolic calculations, *Proceedings of the 9th Conference on Computational modelling and computing in physics*, Dubna, 1997, 219–223.

[7] A.W.Niukkanen, Factorization method and recurrence relations (in Russian), *Integral transforms and special functions.* Computer center, 1997, **1**, No 2, 10-15.

[8] A. W. Niukkanen, A new method in the theory of the hypergeometric series (in Russian), *Uspekhi Matematicheskikh Nauk*, 1988, **43**, 191–193.

[9] A. W. Niukkanen, A new approach to the theory of the hypergeometric series (in Russian), *Russian Matematicheskie zametki*, 1991, **50**, 65–73.

[10] A.W.Niukkanen, Factorization method and special transformations of the functions F_4, H_1 and G_2 (in Russian), *Uspekhi Matem. Nauk*, 1999, **54**, 169–170.

[11] A.W.Niukkanen, New algoritms for the hypergeometric series and the Burchnall and Chaundy expansions (in Russian), *Programmirovanie*, 2000, No 2, 67–69.

[12] A.W.Niukkanen, I.V.Perevozchikov, V.A.Lurie, A generalization of a classical relation between $J_{\nu+n}(z)$, $J_\nu(z)$ and $J_{\nu-1}(z)$, *Fractional calculus and applied analysis*, 2000, **3**, No 2, 119-132.

[13] A.W.Niukkanen, Operator factorization method and addition formulas for hypergeometric functions, *Integral transforms and special functions*, 2001, **11**, 25-48.

[14] A. Erdélyi, W. Magnus, F. Oberhettinger, F. Tricomi, *Higer transcendental functions*, (McGraw-Hill, N.-Y., 1953).

[15] O.S.Paramonova, A.W.Niukkanen, Computer analysis of transformation formulas of the Appell and the Horn functions, submitted to *Programmirovanie*.

Investigation of Nonlinear Second and Third Orders Differential Equations of P-Type with CAS MATHEMATICA

A.N.Prokopenya[a], A.V.Chichurin[b]

[a]Brest Technical University, Belarus;
[b]Brest State University, Belarus;
e-mail: prokopenya@belpak.brest.by

Usage of the computer algebra system Mathematica for solving some problems of analytical and general theory of nonlinear ordinary differ ential equation of the second and the third orders is discussed.

Introduction

There are some problems arising on investigating of the nonlinear second and third orders ordinary differential equations of P-type (solutions of such equations have not movable critical singular points) that can be successfully solved with computer algebra system *Mathematica* ([1,2]). In present paper we'll consider several such problems and show how they can be solved using the corresponding *Mathematica* codes. In the case of the second order differential equations of P-type it is possible to make refer to the next problems.

Problem 1. Construction the systems that are equivalent to the six non - reducible Painleve equations.

We start from construction of the equivalent system for the third Painleve equation (P_3) that can be defined as

$$\mathbf{eq = w''[z] == \frac{w'[z]}{w[z]} - \frac{w'[z]}{z} + \frac{1}{z}(\alpha\ w[z]^2 + \beta) + \gamma\ w[z]^3 + \frac{\delta}{w[z]};}$$

If such system would be constructed then we could build the Baklund's transformation for this equation. Let us seek this system in the form ([3])

$$\mathbf{eq1 = w'[z] == a_0[z] + a_1[z]w[z] + a_3[z]w[z]^2 + a_6[z]w[z]^2v[z];}$$

$$\mathbf{eq2 = v'[z] == b_0[z] + b_1[z]w[z] + b_2[z]v[z] + b_4[z]w[z]v[z] + b_6[z]v[z]^2w[z];}$$

where $a_j(z), b_i(z)$ $(j = 0, 1, 3, 6;\ i = 0, 1, 2, 4, 6)$ are some functions. Solving $eq1$ on $v(z)$ and substituting the result into $eq2$ we obtain the next second order differential equation:

$$\mathbf{sol1 = DSolve[eq1, v, z]//Flatten;}$$

$$\mathbf{eq3 = w''[z] == (w''[z]/.Solve[eq2/.sol1, w''[z]][[1]]);}$$

In order the system of equations $eq1, eq2$ to be equivalent to the second order differential equation of P-type $eq3$ should coincide with (P_3)-equation. To satisfy this requirement we should introduce some restrictions on coefficients $a_j(z)$ and $b_i(z)$. First we equalize denominators of the right sides of $eq3$ and eq.

$$\mathbf{eq4 = Denominator[Together[eq3[[2]] == Denominator[Together[eq[[2]]}$$

$$w[z]\ a_6[z] == z\ w[z]$$

Solving equation $eq4$ we obtain the function $a_6(z)$ $\mathbf{(sol4 = DSolve[eq4, a_6, z][[1]]}$ $\{a_6 \to (\#1\&)\})$. Then we define numerators of the right sides of $eq3$ and eq taking into account the solution $sol4$.

$$\mathbf{eq5 = Collect[Numerator[Together[eq3[[2]]/.sol4]], \{w'[z], w[z]\}];}$$

$$\mathbf{eq6 = Numerator[Together[eq[[2]]]];}$$

Equalizing coefficients of $w'(z)^2$ in $eq5$ and $eq6$ we can find the function $b_6(z)$.

$$\mathbf{eq7 = Coefficient[eq5, w'[z], 2] == Coefficient[eq6, w'[z], 2]}$$

$$2\ z + \ b_6[z] == \ z$$

$$\mathbf{sol7 = DSolve[eq7, b_6, z]//Flatten}$$

$$\{b_6 \to \ (-\#1\ \&)\}$$

Then we equalize coefficients of $w'(z)w(z)^2$ in $eq5$ and $eq6$ and obtain the function $b_4(z)$.

$$\mathbf{eq8 = Coefficient[eq5/.sol7, w'[z]w[z]^2] == Coefficient[eq6, w'[z]w[z]^2]}$$

$$2\ z\ a_3[z]\ +\ z\ b_4[z]\ ==\ 0$$

$$\mathbf{sol8 = DSolve[eq8, b_4, z][[1]]}$$

$$\{b_4\ \to\ (-2\ a_3[\#1\ ]\ \&)\}$$

Substituting functions b_4 and b_6 determined in $sol7$ and $sol8$ into expression $eq5$ and equalizing the corresponding coefficients of $w'w$, w^n $(n = \overline{0,4})$ in $eq5$ and $eq6$ we can find functions b_2, b_1, b_0 a_0 and a_1. This algorithm is realized below.

$$\mathbf{eq9 = Coefficient[eq5/.sol7/.sol8, w'[z]w[z]] == Coefficient[eq6, w'[z]w[z]]}$$

$$1 + \ z\ a_1[z] + \ z\ b_2[z] == \ -1$$

$$\mathbf{sol9 = DSolve[eq9, b_2, z][[1]]}$$

$$\{\ b_2 \to \ (-\frac{2 + \#\ 1\ a_1[\#\ 1]}{\#\ 1}\ \&\ )\ \}$$

$$\mathbf{eq10 = Coefficient[eq5/.sol7/.sol8/.sol9, w[z]^4] == Coefficient[eq6, w[z]^4]}$$

$$z\ a_3[z]^2 + \ z^2\ b_1[z] == \ z\ \gamma$$

$$\mathbf{sol10 = DSolve[eq10, b_1, z][[1]]}$$

$$\{\, b_1 \to \; (-\frac{-\gamma + \; a_3[\#\; 1]^2}{\#\; 1}\; \&\;)\;\}$$

$$\mathbf{eq11 = (eq5/.w \to (0\;\&)/.sol7) == (eq6/.w \to \;(0\;\&))} \quad \{-z a_0[z]^2 == z\;\delta\}$$

$$\mathbf{sol11 = DSolve[eq11, a_0, z][[1]]}$$

$$\{\, a_0 \to \; (-I\;\sqrt{\delta}\;\;\&\;)\;\}$$

$$\mathbf{eq12 = Coefficient[eq5/.sol7/.sol8/.sol9/.sol10/.sol11, w[z]^3] =}$$

$$\mathbf{Coefficient[eq6, w[z]^3]}$$

$$-a_3[z] + (2 + \; z\; a_1[z])\; a_3[z] + z^2\; b_0[z] + z\; a_3'[z] == \; \alpha$$

$$\mathbf{sol12 = DSolve[eq12, b_0, z][[1]]}$$

$$\{\, b_0 \to \; (-\frac{-\alpha + a_3[\#\; 1] + \#1\; a_1[\#]\; a_3[\#] \; + \; \#1\; a_3'\;[\#1]}{\#\; 1^2}\; \&\;)\;\}$$

$$\mathbf{eq13 = Coefficient[eq5/.sol7/.sol8/.sol9/.sol10/.sol11/.sol12, w[z]^2] =}$$

$$\mathbf{= Coefficient[eq6, w[z]^2]}$$

$$-a_1[z] - \; z\; a_1[z]^2 + \; a_1[z]\; (2 + z\; a_1[z]) + z\; a_1'[z] == \; 0$$

$$\mathbf{sol13 = DSolve[eq13, a_1, z][[1]]}$$

$$\{\, a_1 \to \; (\frac{C[1]}{\#1}\; \&\;)\;\}$$

$$\mathbf{eq14 = Coefficient[eq5/.sol7/.sol8/.sol9/.sol10/.sol11/.sol12/.sol13, w[z]] =}$$

$$\mathbf{= Coefficient[eq6, w[z]/.w'[z] \to \; 0}$$

$$I\;\sqrt{\delta} + 2I\;\sqrt{\delta}\;C[1] - \; I\sqrt{\delta}\;(2 + C[1]) \; == \; \beta$$

$$\mathbf{sol14 = DSolve[eq14, C[1]][[1]]}$$

$$\{\, C[1] \to \; -\frac{I\;(\beta + I\sqrt{\delta})}{\sqrt{\delta}}\;\}$$

Now we can substitute all functions found above into $eq3$ and rewrite it as

$$\mathbf{w''[z] = (eq3[[2]]/.sol4/.sol7/.sol8/.sol9/.sol10/.sol11}$$

$$\mathbf{/.sol12/.sol13/.sol14//Collect[\#,\; w'[z], w[z],\; Simplify]\;\&)}$$

$$w''[z] == \frac{\beta}{z} + \frac{\delta}{w[z]} + \frac{\alpha\; w[z]^2}{z} + \gamma\; w[z]^3 - \frac{w'[z]}{z} + \frac{w'[z]^2}{w[z]}$$

We see that this equation exactly coincides with the third Painleve equation. So the system of equations $eq1, eq2$ with functions $a_j(z), b_i(z)$ found above is equivalent to (P_3) equation and its generalizes the corresponding system obtained in [3].

With functions $a_j(z), b_i(z)$ found above we can rewrite this system as

$$\mathbf{\{eq1, eq2\}/.sol4/.sol7/.sol8/.sol9/.sol10/.sol11}$$

/.sol12/.sol13/.sol14///Simplify

$$\{w'[z] == -I\sqrt{\delta} + (1 - \frac{I\beta}{\sqrt{\delta}})\ \frac{w[z]}{z} + (z\ v[z] + a_3[z])w[z]^2,\ v'[z] ==$$

$$\frac{1}{z^2}\ (\alpha\ - z^3 v[z]^2 w[z] - 2a_3[z] + \frac{I\beta\ a_3[z]}{\sqrt{\delta}}\ +$$

$$+\ z\ v[z]\ (-3 + \frac{I\beta}{\sqrt{\delta}} - 2z\ w[z]\ a_3[z]) + z\ w[z](\gamma - a_3[z]^2) - z\ a_3'[z])\ \}$$

Now let us consider the sixth Painleve equation (P_6) [4]. To investigate characteristics of non-movable singular points and also to solve some other problems it is better to use an equivalent system instead of (P_6) equation. Such system consists of two differential equations of the first order. Let the first equation has a form

$$w' = F_1(z,w) + F_2(z,w)v. \qquad (P_6)$$

The second equation is determinated with the form of equation (P_6). Differentiating equation (P_6) and equating the right part of the equation obtained to the right part of (P_6) (that is given in a normal form) after some transformations we have obtained the second equation of the equivalent system in the next form

$$F_2\ v' + \phi_0 + \phi_1\ v + \phi_2\ v^2\ = 0,$$

where $\phi_2 = (\ \frac{\partial\ F_2}{\partial\ w} - l(z,w)\ F_2)\ F_2,$

$$\phi_1 = \frac{\partial\ F_2}{\partial\ z} - m(z,w)\ F_2\ - 2\ l(z,w)\ F_1\ F_2 +\ F_2\ + \frac{\partial\ F_1}{\partial\ w} + F_1\ \frac{\partial\ F_2}{\partial\ w},$$

$$\phi_0 = \frac{\partial\ F_1}{\partial\ z} - n(z,w)\ - m(z,w)\ F_1\ -\ l(z,w)\ F_1^2 +\ F_1\ \frac{\partial\ F_1}{\partial\ w},$$

$$l(w,z)\ =\ \frac{1}{2}\ (\frac{1}{w} + \frac{1}{w-1} + \frac{1}{w-z}),\ \ m(z,w) = -\ (\ \frac{1}{z} + \frac{1}{z-1} + \frac{1}{w-z}),$$

$$n(z,w)\ =\ \frac{w(w-1)(w-z)}{z^2(z-1)^2}\ (\alpha + \beta\ \frac{z}{w^2} + \gamma\ \frac{z-1}{(w-1)^2} + \delta\ \frac{z(z-1)}{(w-z)^2}\ ),$$

where α, β, γ, δ are some constants. In general case functions F_1 and F_2 may be arbitrary. Consequently realization of this algorithm is very complicated. It requires a lot of calculations, which can not be done without CAS (for example, *Mathematica*)([4]). Thus, using *Mathematica* it's easy to build an equivalent system for the equation (P_6).

Using the results obtained for problem 1 the next two problems can be solved.

Problem 2. Reduction of the system obtained to the corresponding equation of geodesic lines and investigation of geometrical properties of its solution ([5,6])

Let us solve the problem for the more general system that include the equivalent to (P_6) system as partial. We define the system in the form

$$\mathbf{ur1 = z(z-1)w'[z] == a[w[z], v[z]]z + b[w[z], v[z]];} \qquad (1)$$

$$\mathbf{ur2 = z(z-1)v'[z] == c[w[z], v[z]]z + d[w[z], v[z]];} \tag{2}$$

From equation (2) we have found $v'[z]$: $\mathbf{sol2 = Solve[ur2, v'[z]][[1]]}$

$$\{v'[z] \to -\frac{-zc[w[z], v[z]] - d[w[z], v[z]]}{z(z-1)}.\}$$

Dividing equation (1) on (2) and considering v as a new independent variable we have found $\frac{dw}{dv}$ in the form

$$\mathbf{ur3 = w'[v] == (ur1[[2]]/ur2[[2]]/.\ \phi\_[z] \to \phi)}$$

$$\{w'[v] == \frac{z\ a[w,v] + b[w,v]}{z\ c[w,v] + d[w,v]}\} \tag{3}$$

Let us find z solving equation (3): $\mathbf{sol3 = Solve[ur3, z][[1]]}$

$$\{z \to \frac{-b[w,v] + d[w,v]\ w'[v]}{a[w,v] - c[w,v]\ w'[v]}\} \tag{4}$$

The right part of (4) is a function of three variables, i.e. $f(w, v, \frac{dw}{dv})$. And we can calculate a derivative $v'[z]$: $\mathbf{\phi[v\_] = z/.sol3/.\phi\_[w, v] \to \phi[w[v], v];}$

$$\mathbf{dv = v'[z]/.sol2/.\phi\_[w[z], v[z]] \to \phi[w[v], v]/.z \to \phi[v]//Simplify}$$

$$\frac{(b[w[v],v]\ c[w[v],v] - a[w[v],v]\ d[w[v],v])(-a[w[v],v] + c[w[v],v]w'[v])}{(b[w[v],v] - d[w[v],v]\ w'[v])(a[w[v],v] + b[w[v],v] - (c[w[v],v] + d[w[v],v])w'[v]))}$$

Now let us consider the main equation $\mathbf{eq4 = dv\ \partial_v\ \phi[v] == 1;}$

$$\mathbf{sol4 = Solve[eq4, w''[v]][[1]]//Simplify;}$$

From the equation obtained we can find the coefficient $k(w, v)$

$$\mathbf{k[w\_, v\_] = -1/((w''[v]/.sol4//Factor)[[2]])/.w[v] \to w}$$

$$-(b[w,v]\ c[w,v] - a[w,v]\ d[w,v])^2$$

In a similar way we can find other coefficients $e[w, v], f[w, v], g[w, v], h[w, v]$

$$\mathbf{rest = (w''[v]/.sol4//Factor)[[3]]//Collect[\#, w'[v], Simplify]\&;}$$

$$\mathbf{e[w\_, v\_] = Coefficient[rest, w'[v], 3]/.w[v] \to w//Simplify}$$

$$(a[w,v]\ d[w,v]^2 c^{(1,0)}[w,v] + c[w,v]d[w,v](d[w,v] - b[w,v]c^{(1,0)}[w,v] -$$

$$-a[w,v]\ d^{(1,0}[w,v]) + c[w,v]^2(d[w,v] + b[w,v]d^{(1,0)}[w,v])$$

$$\mathbf{f[w\_, v\_] = Coefficient[rest, w'[v], 2]/.w[v] \to w//Simplify}$$

$$b[w,v]^2 c[w,v]c^{(1,0)}[w,v] + a[w,v]d[w,v](d[w,v](-1 + c^{(0,1)} - a^{(1,0)}[w,v]) +$$

$$+c[w,v](-2 - d^{(0,1)}[w,v] + b^{(1,0)}[w,v]) + a[w,v]d^{(1,0)}[w,v]) +$$

$$+b[w,v](c[w,v]^2(-1+d^{(0,1)}[w,v]-b^{(1,0)}[w,v])-a[w,v]d[w,v]c^{(1,0)}[w,v]-$$
$$-c[w,v](d[w,v](2+c^{(0,1)}[w,v]-a^{(1,0)}[w,v])+a[w,v]d[w,v]c^{(1,0)}[w,v]-$$
$$-c[w,v](d[w,v](2+c^{(0,1)}[w,v]-a^{(1,0)}[w,v])+a[w,v]d^{(1,0)}[w,v]))$$
$$\mathbf{g[w\_,v\_]=Coefficient[rest,w'[v],1]/.w[v]\to w//Simplify}$$
$$b[w,v]c[w,v](d[w,v]a^{(0,1)}[w,v]-c[w,v]b^{(0,1)}[w,v]+b[w,v](1+c^{(0,1)}-a(1,0)[w,v]))+$$
$$+a[w,v]^2d[w,v](1+d^{(0,1)}[w,v]-b^{(1,0)}[w,v])+a[w,v](d[w,v]$$
$$(-d[w,v]a^{(0,1)}[w,v]+c[w,v]b^{(0,1)}[w,v])+b[w,v](d[w,v](2-c^{(0,1)}[w,v]+$$
$$+a^{(1,0)}[w,v])+c[w,v](2-d^{(0,1)}[w,v]+b^{(1,0)}[w,v])))$$
$$\mathbf{h[w\_,v\_]=Coefficient[rest,w'[v],0]/.w[v]\to w//Simplify}$$
$$-b[w,v]^2c[w,v]a^{(0,1)}[w,v]+a[w,v]b[w,v](-b[w,v]+d[w,v]a^{(0,1)}[w,v]+$$
$$+c[w,v]b^{(0,1)}[w,v])-a[w,v]^2(b[w,v]+d[w,v]b^{(0,1)}[w,v])$$

Thus we have built the second order differential equation of geodesic lines of the next kind

$$k[w,v]\frac{d^2w}{dv^2}=e[w,v](\frac{dw}{dv})^3+f[w,v](\frac{dw}{dv})^2+g[w,v]\frac{dw}{dv}+h[w,v]. \qquad (5)$$

Let us substitute now the values of $a[w,v]z+b[w,v], c[w,v]z+d[w,v]$ for (P_6)

$$\mathbf{a[w,v]z+b[w,v]=\lambda z+((r-\lambda)z-(1+\lambda+q))w+c_2w^2+c_1w(w-1)(w-z)v;}$$

$$\mathbf{c[w,v]z+d[w,v]=\mu-((r-\lambda)z-(1+\lambda+q))v-2c_2wv-\frac{c_1}{2}(3w^2-2zv-2w+z)v^2;}$$

where $\mu\equiv\frac{2\alpha-c_2^2}{2c_1}$, α,λ,r,c_1,c_2 are constants ([7]). Then coefficients of equation (5) for P_6 are determined as $e[w,v], f[w,v], g[w,v], h[w,v], k[w,v]$. For example, $k[w,v]$ has a form

$$k[w,v]=-\frac{1}{4}(vw(-2r+2\lambda+c_1v(2v-1))(-1-q-\lambda+c_1vw(w-1)+c_2w)+$$
$$+(rw+\lambda-\lambda w-c_1vw(w-1))(v^2w(3w-2)c_1-2(v+qv+\lambda v+\mu-2c_2vw)))^2.$$

Problem 3. Construction Backlund's transformation for Painleve equations

Using the functions $a_0(z), a_1(z), a_3(z), a_6(z), b_0(z), b_1(z), b_2(z), b_4(z), b_6(z)$ obtained above we can introduce the next transformations for equation P_3:

$$H:\ w\ \to\ v\ =\ \frac{w'-a_0-a_1\ w-a_3\ w^2}{a_6\ w^2}\ ;$$

$$G:\ v\ \to\ w\ =\ \frac{v'-b_0-b_1\ v}{b_1+b_4\ v+b_6\ v^2}\ ,$$

$$T:\ w\to\ \sigma_1\ w^{k_1},\ z\to\ \sigma_2\ z^{k_2}\ (\sigma_1,\sigma_2,k_1,k_2\neq\ 0).$$

Choosing parameters σ_1, σ_2 ($k_1 = k_2 = 1$) we can fix two nonzero parameters in (P_3) equation. Let $\phi(z)$ is a solution of (P_3). Then we can construct new solutions of (P_3) using the next transformations ([7])

$$T_1: \ \phi \to \ \sigma_1 \ \phi(\sigma_2 z, \alpha \ \sigma_1^{-1} \ \sigma_2, \beta \ \sigma_1 \ \sigma_2, \ \gamma \ \sigma_1^{-2} \ \sigma_2^2, \ \delta \ \sigma_1^2 \ \sigma_2^2),$$

$$T_2: \ \phi \to \ \phi^{-1}(z, -\beta, -\alpha, \ -\delta, \ -\gamma),$$

$$T_3: \ \phi(z, \alpha, \beta, 0, 0) \to \ \phi^{1/2}(z^2, 0, 0, 2 \ \alpha, 2 \ -\beta),$$

Actually, these new solutions $\tilde{w}(z, \tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta})$ can be built according to the scheme:

$$w \ \xrightarrow{H} \ v \ \xrightarrow{T_j} \ \tilde{v} \ \xrightarrow{G} \ \tilde{w} \ \ (j = 1, 2, 3).$$

Each transformation T_j gives us the corresponding Baklund's transformation. Such procedure may be also realized for (P_6) equation ([7]). In the case of the third order differential equations of P-type let us consider

Problem 4.

To find one-parametrical family of solutions (for example, in the form of solutions of Riccati equation) for differential equation of the third order with six singular points

$$y''' = \sum_{k=1}^{6} \frac{(y' - a_k')(y'' - a_k'') + A_k(y' - a_k')^3 + B_k(w' - a_k')^2 + C_k(y' - a_k')}{y - a_k} +$$

$$+ Dy'' + Ey' + \prod_{i=1}^{6}(y - a_k) \ \sum_{k=1}^{6} \frac{F_k}{y - a_k}, \tag{7}$$

where A_k, B_k, C_k, F_k $(k = \overline{1,6})$, D, E are functions of x and a_k $(k = \overline{1,6})$ are constants. J.Chazy found the necessary and sufficient conditions of the solution of equation (7) to have no movable singular critical points, namely, functions A_k, B_k, C_k, F_k D, E and a_k $(k = \overline{1,6})$ have to satisfy the system of 31 algebraic and differential equations. Using this result we rewrite equation (7) in the form ([8])

$$p(y)(y''' - Dy'' - Ey') = \sum_{i=1}^{6} q_i(y) v_i(y'', y') + p(y) \ \sum_{i=1}^{6} q_i(y) \ F_i, \tag{8}$$

where $$p(y) = y^6 + \sigma_2 y^4 - \sigma_3 y^3 + \sigma_4 y^2 + \sigma_6,$$

$$q_i(y) = y^5 + a_i y^4 + \omega_{2i} y^3 - \omega_{3i} y^2 + \omega_{4i}(y + a_i),$$

$$\omega_{2i} = \sigma_2 + a_i^2, \ \omega_{3i} = \sigma_3 - a_i \sigma_2 - a_i^3, \ \omega_{4i} = \sigma_4 - a_i \sigma_3 + a_i^2 \sigma_2 + a_i^4,$$

$$v_i(y'', y') = y' y'' - \frac{1}{a_i} \ y'^3 + B_i \ y'^2 + C_i \ y' \ \ (i = \overline{1,6}),$$

σ_j $(j = 2, 3, 4, 6)$ are elementary symmetric polynomial composed of the elements a_k $(k = \overline{1,6})$. We double differentiate an expression $y' = \alpha(x) \ y^2 + \beta(x) \ y + \gamma(x)$ and substitute

y', y'', y''' into equation (8). Comparing the coefficients of powers y^n $(n = \overline{0,11})$ we obtain a system of equations for determination of the unknown functions $\alpha(x)$, $\beta(x)$, $\gamma(x)$.

For example, for certain coefficients conditions of equation (8) the equations

$$y' = \alpha(y^2 \pm \frac{1}{6\sigma_2}(\mp 4\sigma_2^2 + 6\sigma_4 + \sqrt{4\sigma_2^4 - 27\sigma_2\sigma_3^2 - 24\sigma_2^2\sigma_4 + 36\sigma_4^2}))$$

determine one-parametric family of the equation (8) solutions.

The authors would like to thank Prof. N.A.Lukashevich for the interesting discussion of the considered problems.

References

1. *Prokopenya A. N., Chichurin A. V.* Usage of the System Mathematica for Solving Ordinary Differential Equations. Minsk, BSU, 1999. - 265 p.

2. *Prokopenya A. N., Chichurin A. V.* Computer algebra system Mathematica and solving of ordinary differential equations // Procedings of second international scientific conference CAS-99, Minsk, 20-24.09.1999. - Minsk, BSU, 1999. P. 96-98.

3. *Prokopenya A. N., Chichurin A. V.* Baklund's Transformation Search with Computer Algebra System Mathematica // Proceedings of the Second International Workshop on Mathematica System in Teaching and Research, Siedlce, Poland, Jan. 28-30, 2000. - Moscow, Publ. S. Lavrova, 2000. P.165-177.

4. *Prokopenya A. N., Chichurin A. V.* Construction of the systems that are equivalent to the equation // Proceedings of International Mathematical Conf. "VIII Byelorussian Mathematical Conference", Minsk, June 19-24, 2000. - Minsk, IM NASB, 2000. V. 3. P. 176.

5. *Lukashevich N. A., Chichurin A. V.* About one transformation III and V Painleve equations // Proc. of the Acad. of Sc. of Belarus. 1995. No 2, P.9-12.

6. *Prokopenya A. N., Chichurin A. V.* Investigation of the equation of geodesic lines corresponding to the third Painleve equation // Proc. of Int. Conf. "The Third Scientific Readings on Ordinary Differential Equations on the 80th Anniversary of Yu.S.Bogdanov Birth", Minsk, Feb. 26-March 1, 2001. - Minsk, BSU, 2001. P. 14-15.

7. *Gromak V. I., Lukashevich N. A.* The Analytic Solutions of the Painleve equations. Minsk. Universitetskoye, 1990.

8. *Chichurin A. V.* Third Order P-type Equations with Six Poles // Vestnik Brest University, 2001. No 2. P. 55-61.

Cosmological Creation of Vector Bosons and Integrals of Motion in General Relativity

D. Proskurin, V.N. Pervushin

Bogoliubov Laboratory of Theoretical Physics,
Joint Institute for Nuclear Research,
Dubna, Russia

The cosmological creation of primordial vector bosons and fermions is described in the Standard Model of strong and electro-weak interactions given in a space-time with the relative standard of measurement of geometric intervals. Using the reparametrization - invariant perturbation theory and the holomorphic representation of quantized fields we derive equations for the Bogoliubov coefficients and distribution functions of created particles. The main result is the intensive cosmological creation of longitudinal Z and W bosons (due to their mass singularity) by the universe in the rigid state. We introduce the hypothesis that the decay of the primordially created vector bosons is the origin of the Cosmic Microwave Background radiation.

Analytical Design of SIMD Computer Application Software

V.S. Richvitsky, A.P. Sapoznikov, A.G. Galperin

Laboratory of Information Technologies,
Laboratory of High Energies,
Joint Institute for Nuclear Research,
141980 Dubna, Russia;
e-mail: rqvtsk@cv.jinr.ru

1 APE-100 supercomputer

One of the aim of the APE100 group [2, 3] is the realization of a supercomputer specially designed to solve specific problem arising from QCD community, but suitable also for other applications.

The family of computers APE is array-parallel computing systems with distributed memory and SIMD architecture. The basic distinctive feature of architecture SIMD (Single Instruction, Multiple Data) is the fact, that all processors, included in structure of the computing system, synchronously carry out the same program, and each processor processes the complete set of local data.

To the present time APE-100 is successfully maintained in Italy, Great Britain and Germany. A 32-processor complete set (the configuration 8x2x2) is established and in JINR [1].

The basic programming language in APE-100 is language TAO of a high level with fortran-like syntax and built-in support of parallel architecture of the machine.

2 APE-100 programming

The programming of computers with SIMD architecture is very difficult. Some Fortran compilers has regimes of automatic parallelization of usual programs. The present article gives another approach.

The original mathematical problem is to be solved must seemed contain some properties to be parallelizable. For example, if may be the set of processes with the identical program over data of the same structure, that are not communicate each other, such as Monte-Carlo simulation.

In other cases the processes may be parallel and execute the identical instruction stream but communicate at some points. If some big row of data is distributed among the number of processors for summation, the total sum will be obtained when processors communicate their partners their partial sums and after that each of them can calculate total sum.

Task of data distribution is may be solvable due to some symmetries in calculations such as some permutation group of assignments, that not affect on results. Mentioned group property is thought be more the property of the original mathematical task, than of the numerical method or the derived algorithm in language of programming.

So, if the formulas of task solution derived with analytical tools such as Maple, this analytical tool can find the symmetries for distribution the data among processors and such methods of their communications that give the total results.

Usually, the programmer mentions, that each method of programming may be given by some macro, so that the task from initial formulation on some formal language using formal syntax and formal context rules may be decomposed into sequence of applications of macro to primitive objects and to the preceding ones. The primitive objects are the data distribution and the programming forms.

Our approach is to discover the symmetries in the original task formulation, that preserving in derived formulas and programming statements. Then the analytical tool may fill the structures of data distribution in memory with additional structures, that distribute the data among processors, parallelize calculations and collect results of them.

The Maple power is enough for analyzing the structure of formulas, the relations between and assignments 'producer-consumer', group calculations and writing the text of output program in some suitable language of parallel programming to file.

2.1 Program control

The control of the program may be global and local. The usual control operators - branching and cycles - concern to global, i.e. work simultaneously on all processors (since a flow of teams one).

The local branching is carried out on each processor individually. Actually, the branch, which should not be run, running on all the processors, but the results of calculations are not writing down to a memory.

Local conditions is usual logic expression. The global condition will be formed from local with the some specific logic functions.

2.2 Data control

Every processing node can access data stored on the nodes at its right, left, up, down, front and rear. The RIGHT, LEFT, UP, DOWN, FRONT and REAR keywords can be added to the address of an array in order to access the data stored in one of the neighbouring nodes.

Example 1. The procedure summarizes same variable s1 on all processors and leaves a copy of a sum in variable sum also on each processor.

```
subroutine SumAllLocals(real s1, real sum) real t[1]  !! must be
declared as array because we use "indexes"
 t[0]=s1
 sum=s1                !! start from own local value
 do i=2,imachinelx !! summation in x-direction
```

```
  t[0]=t[left]+sum
 enddo
 sum=t[0]
 do j=2,imachinely !! summation in y-direction
  t[0]=t[back]+sum
 enddo
 sum=t[0]
 do k=2,imachinelz !! summation in z-direction
  t[0]=t[up]+sum
 enddo
 sum=t[0]              !! final result
end
```

3 Analytical calculations in automatic program construction

The problem of SIMD is such distribution of data between processors, that it "was divided without reminder".

Methods of data distribution must be packed into Maple procedures, that analyze the variable dependence and try any possible allocations of them including division arrays on parts and duplication of simple variables in memories of different processor nodes.

Example 2. Scalar product of vectors v1 [160] and v2 [160] is calculated which are placed on sixteen processors, connected in a cyclically closed square (i.e. torus). The known way of summation is divided into two parts: local summation and summation between processors.

```
real v1[10],v2[10] real s[1] sum=0. do i=1,10
 sum=sum+v1[i]*v2[i]
enddo s[0]=sum do i=2,4
 s[0]=s[left]+sum
enddo sum=s[0] do i=2,4
 s[0]=s[down]+sum
sum=s[0] enddo
```

Backtracking algorithms for retrieve combinatorial schemes are known in Prologue language [5, 6], where backtracking is fundamental mechanism of proof.

4 The analysis of structure of calculations and program construction

We shall assume, that the problem is described by the disordered list of calculations of a kind

$$var = formula, ...$$

It is required to determine dependence between variables and to order assignments, by receiving thus program.

4.1 Ordering calculations

Each calculation creates the relation of preceding between variables. Variable X directly precedes variable Y (X ¡ Y), if the formula for calculation Y contains X. The relation X ¡ Y is transitive closure of direct precedence.

The precedence derives a lattice with operations $\vee$ and $\wedge$: $Z = X \wedge Y$, if Z is greatest such, that $Z < X$ and $Z < Y$. Similarly $Z = X \vee Y$, if Z is least such, that $X < Z$ and $Y < Z$. If the lattice is incomplete, it should be filled up with greatest and least (by definition) symbols.

An ideal in a lattice U is a set B, that a) if $x \in B$ and $y \in U$, than $x \vee y \in B$, b) if $x \in B$ and $y < x$, than $y \in B$.

The greatest ideal in U will be set $\{z = x \vee y : x, y \in U\}$.

Ordering calculations is reached by executing of the following algorithm:

```
U:=set of all variables;
 While U <> {} do
 B:=greatest ideal in U;
 Print out calculations of variables of U-B;
 U:= B
End.
```

It is the not most effective algorithm, but it is useful, as it is a basis of all further circuits of calculations.

The procedure Symlist builds for each expression the list variable, from which it depends:

```
 Symlist:=proc(x) local s,i,a,b,t:
   if type(x,symbol) then {x} else
   if type(x,atomic) and whattype(x)<>indexed then {} else
   if op(0,x)=`::`then
     i:=op(1,op(1,x)):  a:=op(1,op(2,op(1,x))):
     b:=op(2,op(2,op(1,x))):  t:=op(2,x):
    Symlist(t) union Symlist(a) union Symlist(b) minus {i}
   else
    s:={}: for i from 1 to nops(x) do:
     s:=s union Symlist(op(i,x)):
    od:
    s
   fi
   fi
   fi
end:
```

Example 3.

$$Symlist(2 * cos(phi)/d + r * sin(phi))$$

results

$$\{r, d, phi\}.$$

The procedure Stream orders calculations:

```
Stream:=proc(s) local rr,rl,rp,ru,r0,re,rs,rv,r:
 rr:=Union(map(x->Symlist(rhs(x)),s));
  rl:=Union(map(x->Symlist(lhs(x)),s));
  rp:=rr minus rl; ru:=rr union rl minus rp;
  r0:=s: r:=[]: rs:=[]; while ru <> {} do:
   re:=select(x->(Symlist(rhs(x)) intersect ru)={},r0);
   r:=[r,re];  rv:=map(x->lhs(x),re);
   r0:=r0 minus re;  ru:=ru minus rv;
  od:
  List(r)
end:

List:=proc(s) local i:
 if type(s,set) or type(s,list) then seq(List(s[i]),i=1..nops(s))
 else s
 fi
end:
```

Example 4.

$$Stream(\{a = b + c, c = d/e, b = e\})$$

results

$$c = d/e,\ b = e,\ a = b + c.$$

The procedure Translate builds the program under the list of the formulas:

```
Translate:=proc(S) local i,a,b,t:
 if op(0,S)='::' then
  i:=op(1,op(1,S)):
  a:=op(1,op(2,op(1,S))):
  b:=op(2,op(2,op(1,S))):
  t:=op(2,S):
  print(cat('for ',i,':=',a,' to ',b,' do begin'));
  Translate(t);
  print('end;');
 else
  if type(S,list) then
   for i from 1 to nops(S) do:
    Translate(S[i]);
   od:
```

```
  else
   if type(S,set) then
    Translate([Stream(S)]);
   else
    print(S);
   fi
  fi
 fi
end:
```

Example 5.

$$S := \{(i = 0..3) :: z[i] = r[i] - b, r[i] = f[i] + i, b = c/d\} : \quad Translate(S);$$

results

```
b = c/d do i=0,3 r[i] = f[i] + i z[i] = r[i] - b enddo
```

References

[1] Sapojnikov A.P., " APE Project in JINR ", Second International Conference " Modern Trends in Computational Physics ", July 24-29, 2000, Dubna, Russia. D11-2000-155, p144. *Http : //www.jinr.ru/ tsap/Koi/ape/ape_conf.htm*

[2] The Ape Project: a Gigaflop Parallel Processor for Lattice Calculations, P. Bacilieri et al., L.O. Hertzberger and W. Hoogland (Editors), Computing in High Energy Physics 85. Elsevier Science Publishers B.V., North-Holland, 1986.

[3] The Ape Computer: an Array Processor Optimized for Lattice Gauge Theory Simulations, M. Albanese et al., Comp. Phys. Comm. 45(1987)345.

[4] Burow B.D., " DRAFT: Task System and Item Architecture (TSIA) ", DESY 99-066 June 1999.

[5] Language a Prologue in the fifth generation the COMPUTER. The collection of the articles, translation with English. Moscow, "World", 1988.

[6] Hogger K. " Introduction in logic programming ". Moscow, World, 1988.

Some Formal Solutions of LODE

A. Ryabenko[1]

Computing Center of the Russian Academy of Science
Vavilova 40, Moscow 117967 Russia;
e-mail: ryabenko@cs.msu.su

Let be given a linear homogeneous ordinary differential equation $Ly(x) = 0$ with coefficients which are polynomial over a field K of characteristic 0. There is a method [3] to build a local fundamental system of formal solutions for the equation in a neighbourhood of $x = x_0$.

A formal solution has the form

$$y(t) = e^{Q(1/t)} t^{\lambda} \Phi(t), \quad x(t) - x_0 = \Lambda t^p,$$

where the *regular part* $\Phi(t) = \Phi_s(t) + \Phi_{s-1}(t)\log(t) + \cdots + \Phi_0(t)\log^s(t)$ can be built by the well known Frobenius' algorithm. The power series $\Phi_0(t) = \sum_{n=0}^{\infty} c_n^{(0)} t^n$ has coefficients which satisfy a linear homogeneous recurrence $Rc^{(0)} = 0$ with polynomial coefficients. The others series $\Phi_i(t) = \sum_{n=0}^{\infty} c_n^{(i)} t^n$, $1 \le i \le s$, have coefficients which satisfy homogeneous or inhomogeneous recurrences: $Rc^{(i)} = f_i(n)$. Using these recurrences, we can compute formal solutions with arbitrary accuracy.

We present a method to select a subspace of formal solutions that content only such power series which are polynomial (i.e. $c_n^{(i)} = 0$ for all large enought n) or m-hypergeometric (i.e. $c_{n+m}^{(i)}/c_n^{(i)} \in \tilde{K}(n)$, $m \ge 1$, for all large enought n). In this case we can write series from formal solutions in the closed form. To arrive at our goal we use the algorithms from [1] and [2] for finding all m-hypergeometric solutions of the homogeneous and inhomogeneous recurrence.

References

[1] M. Petkovšek, Hypergeometric solutions of linear recurrences with polynomial coefficients. *J. Symb. Comput.*, 1992, vol. 14, pp. 243-264.

[2] M. Petkovšek, B. Salvy, Finding all hypergeometric solutions of linear differential equations, *Proc. ISSAC'93*, 1993, pp. 27-33.

[3] E. Pflugel, DESIR-II, *RT 154, IMAG Grenoble*, 1996.

[1]Supported in part by grant from the French-Russian Lyapunov Institute, Project No 98-03

Solving an Inverse Problem on Lattice by Using CAS REDUCE

S.I. Serdyukova

JINR, Dubna, Russia;
e-mail: sis@cv.jinr.ru

An algorithm of numerical solution of the inverse problem for two-dimensional Shrödinger equation had been worked out [1]. The problem reduces to reconstruction of five-diagonal $M \times N$ matrix with given spectrum and given first N components for each of basic eigenvectors. But all N components can't be chosen arbitrary. It's stated that they must satisfy $(N-1)^2(M-1)$ additional conditions and N conditions of compatibility. We succeeded to clear the statement of the problem to the end in the process of concrete calculating. It had been shown that the lacking components can be determined by solving the system of the polynomial equations. Deriving and solving huge polynomial system had been performed on SPP by using CAS REDUCE 3.6. This time we succeeded to overcome difficulties [1], arising in solving, by choosing proper variables and proper initial data. In our report we discuss as well the inverse problem, when symmetry of basic eigenvectors is reserved. Then the problem reduces to solving polynomial system not so big order (36 instead 62 in concrete example). But additional difficulties connected with the statement of the problem arise. The symmetry of basic eigenvectors leads to symmetry of blocks of found matrix. As result we can't even disturb spectrum arbitrarily. First N components must satisfy additionally some "symmetry conditions". In our report we show how such inverse problem is stated and solved in concrete case.

1 General statement of the problem

Wave motion on the lattices in discrete quantum mechanics is described [2] by finite-difference Schrödinger equation:

$$-\frac{\psi_{i-1,j}-2\psi_{i,j}+\psi_{i+1,j}}{h_x^2}-\frac{\psi_{i,j-1}-2\psi_{i,j}+\psi_{i,j+1}}{h_y^2}+U_{ij}\psi_{ij}=\lambda\psi_{i,j}.$$

We consider the problem in rectangle $1 \le i \le M, \quad 1 \le j \le N$, with zero boundary conditions: $\psi_{0,j}=\psi_{M+1,j}=\psi_{i,0}=\psi_{i,N+1}=0$.

In the case of zero potential ($U_{i,j}=0$) the eigenvalues and basic eigenvectors are determined by formulae

$$\lambda_{m,n} == \frac{4}{h_x^2}\sin^2\frac{\pi m}{2(M+1)}+\frac{4}{h_y^2}\sin^2\frac{\pi n}{2(N+1)},$$

$$V_{m,n}=||v_{m,n}(i,j)||=2\sqrt{h_x h_y}||\sin\frac{\pi m i}{M+1}\sin\frac{\pi n j}{N+1}||,$$

$$1 \le i,m \le M,\ 1 \le j,n \le N,\ l = MN.$$

We arrange the eigenvalues in order of increasing:

$$\lambda_1 \le \lambda_2 \le \lambda_3 ... \le \lambda_l.$$

Consider vectors $E_\xi = [e_1(\xi), ..., e_l(\xi)] =$

$$= [v_{mn}(1,1), ..., v_{mn}(1,N), ..., v_{mn}(M,1), ..., v_{mn}(M,N)].$$

λ_ξ, E_ξ (determined for zero potential and $h_x = h_y = 1$) are solution the spectral problem for symmetric five-diagonal matrix

$$C = \begin{bmatrix} A_1 & D_1 & 0 & \dots & 0 \\ D_1 & A_2 & D_2 & \dots & 0 \\ \dots & \dots & \dots & \dots & D_{M-1} \\ 0 & 0 & \dots & D_{M-1} & A_M \end{bmatrix}, \quad A_i = \begin{bmatrix} 4 & -1 & 0 & \dots & 0 \\ -1 & 4 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & -1 \\ 0 & 0 & \dots & -1 & 4 \end{bmatrix},$$

and on the lateral diagonals $D_i = -I$ blocks stay, where I are the unit matrices of N order.

The problem of reconstruction of perturbed discrete Schrödinger operator from spectral data is stated. What is equivalent to the problem of reconstruction of C perturbed five-diagonal matrix: on the lateral diagonals of C diagonal blocks

$$D_i = \begin{bmatrix} -1+v_{qi+1} & 0 & \dots & 0 \\ 0 & -1+v_{qi+2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & -1+v_{qi+N} \end{bmatrix}, \ i = 1, ..., M-1,$$

$qi = (i-1)N$, stay. And on the main diagonal

$$A_i = \begin{bmatrix} 4+\theta_{qi+1} & -1+u_{qi+1} & \dots & 0 \\ -1+u_{qi+1} & 4+\theta_{qi+2} & \dots & 0 \\ \dots & \dots & \dots & -1+u_{qi+N-1} \\ 0 & \dots & -1+u_{qi+N-1} & 4+\theta_{qi+N} \end{bmatrix}, \ i = 1, ..., M,$$

blocks stay. In what follows, we note

$$4+\theta_i = tei, \ -1+u_i = uui, \ -1+v_i = -vvi.$$

The orthonormality of basic eigenvectors implies [3] the orthonormality of e_i vectors constituted of i-th components of basic eigenvectors: $e_i = [e_i(1),\ e_i(2), ..., e_i(l)]$. From this a simple algorithm (presented in Section 5 of [1]) for computing C symmetric five-diagonal matrix of given spectrum and given first N components for each basic eigenvector arises. C matrix has lacuna between the second and $(N+1)$-th diagonals. As result all N components can't be given arbitrary.

These components besides $N(N+1)/2$ orthonormality conditions must satisfy $(N-1)^2(M-1)$ additional conditions, guaranteeing orthonormality of the following components of basic eigenvectors:

$$(e_1, e_j) = 0, ..., (e_{j-1}, e_j) = 0,\ j = N+1, ..., l,$$

computed simultaneously with matrix. In addition compatibility conditions guaranteeing satisfying N low equations of $CE_j = \lambda_j E_j$ spectral system for all j must be fulfilled.

Inverse problem for two-dimensional discrete Schrödinger equation reduces [1] to reconstruction of symmetric five-diagonal matrix of given spectrum and given $k(M,N)$, $1 \leq k < N$, *first components for each basic eigenvector.*

Elements of matrix are computed simultaneously with "lacking" $(N-k)$ components by solving polynomial equations system, which along with the orthonormality conditions, the additional conditions and the compatibility conditions contain relations defining matrix elements as functions of λ_j and components of basic eigenvectors.

2 New system and new initial approximation

The results presented in [1] were stated in process of numerical experiments with first spectra point $\lambda_1 = 0.96775...$ was disturbed to $\tilde{\lambda}_1 = 0.96$. The lacking components of basic eigenvectors had been found by solving the system of 62 polynomial equations with 62 variables. The polynomial system contains:

- 9 conditions of orthonormality,
- 18 additional conditions, providing orthogonality of computed components of basic eigenvectors,
- 4 compatibility conditions,
- 28 relations,determining 28 elements of found matrix and 3 relations $sui = \sum_{j=1}^{l} (\lambda_j e_i(j))^2$, defining sui, $i = 2,3,4$.

62 variables are:

- 31 the lacking components: $e_2(j)$, $j = 1:7$; $e_3(j), e_4(j)$ $j = 1:l$;
- 28 elements of matrix: tei, $i = 2:l$; uui, $i = 1:3,5:7,9:11$; $vvi = 1:8$;
- 3 variables $su2$, $su3$, $su4$.

Original values corresponding to non-perturbed matrix had been chosen as initial data. In such way we had perfect result [1] by using NUMERIC package of REDUCE 3.6 [4] for $\tilde{\lambda}_1 = 0.96$ and $\tilde{\lambda}_1 = 0.973$. But for $\tilde{\lambda} = 0.975$ the Newton type iterations realized in NUMERIC diverge.

In the case of $\tilde{\lambda} = 0.975$ we succeeded to solve the inverse problem by including of $e_5(j), ..., e_{12}(j)$, $j = 1:l$ in the set of variables. The relations, determining new variables recursively, complete the system of 62 equations to the system of 158 equations. It's nothing but the equations of spectral system, starting from the 5th, for all given eigenvalues. This time original initial data are corrected: matrix elements and sui are

recomputed of given disturbed eigenvalues and original basic eigenvectors components. In addition $te12$ is determined by spur equation and relations

$$(e_9, e_7) = 0, \quad (e_{10}, e_8) = 0, \quad (e_{12}, e_9) = 0$$

are used instead of the corresponding additional conditions which contain multipliers

$$(te1 + te2 + te3),\ (te2 + te3 + te4),\ (te1 + te2 + te3 + te4)$$

and are fulfilled not so good after correction of initial data. Remark, that equations of new system have much more simple structure. As a matter of fact this system of 62 equations is obtained of the system of 158 equation by excluding new variables using the recursive relations defining new variables. As result the times of computing are the same approximately:

$$(0.975, 25, 9, 13), \quad SPP, \quad t = 2853020, \quad 158 \text{ equations},$$

$$(0.96, 25, 15, 10), \quad SPP, \quad t = 2871840, \quad 62 \text{ equations}.$$

Computing 13 iterations takes 2583020ms in the case of 158 equations and computing 10 iterations takes 2871840ms in the case of 62 equations. We believe that this (when all components of basic eigenvectors are variables) is more natural way of solving the considered inverse problem. But attempts to solve in such way the inverse problem with $\tilde{\lambda}_1 = 0.96$ gave no result. The Newton type iterations realized in NUMERIC diverge.

3 Inverse problem with reserving symmetry conditions

In this section we discuss the inverse problem when the symmetry of basic eigenvectors are reserved. With such symmetry we can prolong the eigenvalues, determining in the rectangle on the whole plain. We consider again $m = 3$, $n = 4$. This time problem reduces to solving polynomial system of order 36 only instead of 62. But new difficulties appear in statement of the problem. The symmetry of basic vectors lead to special symmetry block structure of found matrix. As result we can't even to disturb spectrum points arbitrarily. We discuss solving such problem in concrete example only. When $m = 3$, $n = 4$,

$$\begin{array}{ccccccc}
4 & \triangleleft & 8 & \triangleleft & 12 & \triangleleft \\
 & \vdots & \dots & \vdots & \dots & \vdots \\
3 & \triangleleft & 7 & \triangleleft & 11 & \triangleleft \\
 & \vdots & \dots & \vdots & \dots & \vdots \\
2 & \triangleleft & 6 & \triangleleft & 10 & \triangleleft \\
 & \vdots & \dots & \vdots & \dots & \vdots \\
1 & \triangleleft & 5 & \triangleleft & 9 & \triangleleft
\end{array}$$

Basic eigenvectors of original problem have such symmetries:

$$e_1(j) = (-1)^{j-1} e_4(j),\ j = 1 : 12; \tag{1}$$

$$e_5(j) = (-1)^{j-1} e_8(j),\ j = 1:12, \tag{2}$$

$$e_5(i) = e_8(i) = 0,\ i = 5:8; \tag{3}$$

$$e_9(j) = (-1)^{j-1} e_{12}(j),\ j = 1:12; \tag{4}$$

$$e_2(j) = \begin{cases} e_{10}(j),\ j = 1:4,\ 9:12, \\ -e_{10}(j),\ j = 5:8; \end{cases} \tag{5}$$

$$e_3(j) = \begin{cases} e_{11}(j),\ j = 1:4,\ 9:12, \\ -e_{11}(j),\ j = 5:8; \end{cases} \tag{6}$$

$$e_1(j) = \begin{cases} e_9(j),\ j = 1:4,\ 9:12, \\ -e_9(j),\ j = 5:8; \end{cases} \tag{7}$$

$$e_4(j) = \begin{cases} e_{12}(j),\ j = 1:4,\ 9:12, \\ -e_{12}(j),\ j = 5:8; \end{cases} \tag{8}$$

As $tei = \sum_{j=1}^{12} \lambda_j e_i(j)^2$, $sui = \sum_{j=1}^{12} (\lambda_j e_i(j))^2$, $vvi = \sum_{j=1}^{12} \lambda_j e_i(j) e_{i+4}(j)$, the symmetry conditions (1),(2) give $te1 = te4$, $su1 = su4$ $vv1 = vv4$. And taking in account relations $su1 = te1^2 + uu1^2 + vv1^2$, $su4 = te4^2 + uu3^2 + vv4^2$ we find, that $uu1 = uu3$. Using additionally the first and the fourth equations of the spectral system

$$\begin{cases} (te1 - \lambda_j)e_1(j) + uu1 \cdot e_2(j) + vv1 \cdot e_5(j) = 0, \\ uu1 \cdot e_1(j) + (te2 - \lambda_j)e_2(j) + uu2 \cdot e_3(j) + vv2 \cdot e_6(j) = 0, \\ uu2 \cdot e_2(j) + (te3 - \lambda_j)e_3(j) + uu3 \cdot e_4(j) + vv3 \cdot e_7(j) = 0, \\ uu3 \cdot e_3(j) + (te4 - \lambda_j)e_4(j) + vv4 \cdot e_8(j) = 0, \\ vv1 \cdot e_1(j) + (te5 - \lambda_j)e_5(j) + uu5 \cdot e_6(j) + vv5 \cdot e_9(j) = 0, \\ vv2 \cdot e_2(j) + uu5 \cdot e_5(j) + (te6 - \lambda_j)e_6(j) + uu6 \cdot e_7(j) + vv6 \cdot e_{10}(j) = 0, \\ vv3 \cdot e_3(j) + uu6 \cdot e_6(j) + (te7 - \lambda_j)e_7(j) + uu7 \cdot e_8(j) + vv7 \cdot e_{11}(j) = 0, \\ vv4 \cdot e_4(j) + uu7 \cdot e_7(j) + (te8 - \lambda_j)e_8(j) + vv8 \cdot e_{12}(j) = 0, \\ vv5 \cdot e_5(j) + (te9 - \lambda_j)e_9(j) + uu9 \cdot e_{10}(j) = 0, \\ vv6 \cdot e_6(j) + (te10 - \lambda_j)e_{10}(j) + uu10 \cdot e_{11}(j) = 0, \\ vv7 \cdot e_7(j) + (te11 - \lambda_j)e_{11}(j) + uu11 \cdot e_{12}(j) = 0, \\ vv8 \cdot e_8(j) + uu11 \cdot e_{11}(j) + (te12 - \lambda_j)e_{12}(j) = 0, \end{cases} \tag{9}$$

we get new symmetry

$$e_2(j) = (-1)^{j-1} e_3(j). \tag{10}$$

So it's natural, when we find the lacking of 4 first components, to suppose that they satisfying (1) and (10). This leads immediately to the following relations $te1 = te4$, $te2 = te3$, $uu1 = uu3$, $su1 = su4$, $su2 = su3$, $vv1 = vv4$, $vv2 = vv3$, $e_5(j) = (-1)^{j-1}e_8(j)$. Taken in account additionally and the second and the third equations of the spectral system (9), we get one more symmetry $e_6(j) = (-1)^{j-1}e_7(j)$. This and the symmetry condition (2) give $te5 = te8$, $te6 = te7$, $uu5 = uu7$, $su5 = su8$, $su6 = su7$.

These, the previous found relations and

$$su5 = vv1^2 + te5^2 + uu5^2 + vv5^2,\ su8 = vv4^2 + uu7^2 + te8^2 + vv8^2$$

$$su6 = vv2^2 + te5^2 + uu5^2 + te6^2 + uu6^2 + vv6^2,\ su7 = vv3^2 + uu6^2 + te7^2 + uu7^2 + vv7^2$$

give $vv5 = vv8$, $vv6 = vv7$. Using 5th and 8th equations of (9) we have, that

$$e_9(j) = (-1)^{j-1}e_{12}(j), \quad te9 = te12 = te1. \tag{11}$$

With additional request (3) and symmetry conditions (5), (7) we obtain, that $vv1 = vv5$. Using the fifth equation of the spectral system (9) we find, that $e_6(j) = 0$, $j = 5:8$, which implies $vv2 = vv6$. Using 6th and 7th equations of the spectral system (9), relations $vv2 = vv3$, $uu5 = uu7, te6 = te7$, $vv6 = vv7$ and symmetries:

$$e_2(j) = (-1)^{j-1}e_3(j), \quad e_5(j) = (-1)^{j-1}e_8(j), \quad e_6(j) = (-1)^{j-1}e_7(j)$$

we get, that $e_{11} = (-1)^{j-1}e_{10}(j)$. The last and (11) imply $te10 = te11, uu9 = uu11$.

As result we proved, that disturbed matrix, whose basic eigenvectors satisfy the symmetry conditions (1),(5),(7) and additional request (3), has special symmetric block-diagonal structure:

$$M = \begin{bmatrix} A & W & 0 \\ W & B & W \\ 0 & W & A \end{bmatrix}, \quad A = \begin{bmatrix} a_1 & b_1 & 0 & 0 \\ b_1 & a_2 & b_2 & 0 \\ 0 & b_2 & a_2 & b_1 \\ 0 & 0 & b_1 & a_1 \end{bmatrix}, \quad B = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & 0 \\ \beta_1 & \alpha_2 & \beta_2 & 0 \\ 0 & \beta_2 & \alpha_2 & \beta_1 \\ 0 & 0 & \beta_1 & \alpha_1 \end{bmatrix}$$

and $W = diag(d_1, d_2, d_3, d_4)$. Eigenvalues of M are solutions of

$$Det[-(A - \lambda I)(2 - W^{-1}(B - \lambda I)W^{-1}(A - \lambda I))] = 0,$$

where I is the unit matrix of order 4. The spectrum of such matrix can't be arbitrary. We can't to disturb the spectrum of original matrix arbitrarily and hope to find matrix having given disturbed spectrum and reserving the symmetric block-diagonal structure.

The eigenvectors $E = [E_1, E_2, E_3, E_4]^T$, $E_i -$ column-vectors of order 4, satisfy a system

$$\begin{cases} (A - \lambda I)E_1 + WE_2 = 0, \\ WE_1 + (B - \lambda I)E_2 + WE_3 = 0, \\ WE_2 + (A - \lambda I)E_3 = 0. \end{cases}$$

By subtracting the third equation from the first we get

$$(A-\lambda I)E_1 = (A-\lambda I)E_3.$$

So, if λ is not eigenvalue of A, then $E_1 = E_3$. For original matrix λ_5, λ_6, λ_7, λ_8 form the spectrum of A. We succeeded to solve the inverse problem for two disturbed spectra.

First we put $\delta = 0.003$, $\Delta a_1 = \Delta a_2 = \delta$, $\Delta a_3 = 1 - \sqrt{1+\delta-\delta^2}$, $\Delta a_4 = 2\delta$, the other elements of matrix stay nonperturbed. Then λ_i with odd i are disturbed and with even i stay nonperturbed.

Second we put $\delta = 0.001$, $\Delta a_1 = \delta$, $\Delta a_2 = 3\delta$, $\Delta a_3 = -\delta$, $\Delta a_4 = -2\delta$, $\Delta b_1 = 2\delta$, $\Delta b_3 = 2\delta$, $\Delta b_4 = 3\delta$, $\Delta c_1 = -2\delta$, $\Delta c_2 = -3\delta$, $\Delta c_3 = -2\delta$, $\Delta b_4 = -3\delta$, the other elements of matrix stay nonperturbed. Then all λ_i were disturbed.

In both cases perturbed matrix $\tilde{M}$, whose basic eigenvectors reserve the symmetry, was reconstructed of given spectrum by solving a system of 36 polynomial equations with 36 variables:

$$e_1(j), e_2(j),\ j = 1:12;\ te1, te2, te5, te6, uu1, uu2, uu5, uu6, vv1, vv2; su1, su2.$$

The system consists of 6 orthonormality conditions, 6 additional conditions (instead of 18 in [1]), 4 compatibility conditions and 8 symmetry conditions.

Orthonormality conditions are

$$||e_1|| = 1,\ \sum_{j=1}^{12} e_1(j)^2 = 1; \quad ||e_2|| = 1,\ \sum_{j=1}^{12} e_2(j)^2 = 1;$$

$$(e_1, e_2) = 0,\ \sum_{j=1}^{12} e_1(j)e_2(j) = 0; \quad (e_1, e_3) = 0,\ \sum_{j=1}^{12} e_1(j)e_2(j)(-1)^j = 0;$$

$$(e_1, e_4) = 0,\ \sum_{j=1}^{12} e_1(j)^2(-1)^j = 0; \quad (e_2, e_3) = 0,\ \sum_{j=1}^{12} e_2(j)^2(-1)^j = 0.$$

Additional conditions are

$$e_5 \perp e_3,\ \sum_{j=1}^{12} \lambda_j e_1(j)e_3(j) = 0; \quad e_5 \perp e_4,\ \sum_{j=1}^{12} \lambda_j e_1(j)e_4(j) = 0;$$

$$e_6 \perp e_5,\ \sum_{j=1}^{12} \lambda_j^2 e_1(j)e_2(j) = uu1(te1 + te2);$$

$$e_7 \perp e_5,\ \sum_{j=1}^{12} \lambda_j^2 e_1(j)e_3(j) = uu1 \cdot uu2;$$

$$e_7 \perp e_6,\ \sum_{j=1}^{12} \lambda_j^2 e_2(j)e_3(j) = uu2(te2 + te3); \quad e_8 \perp e_5,\ \sum_{j=1}^{12} \lambda_j^2 e_1(j)e_4(j) = 0.$$

Compatibility conditions are $e_5(j) = 0$, $j = 5 : 8$. Symmetry conditions are $e_1(j) = e_9(j)$, $j = 1 : 4$; $e_2(j) = e_{10}(j)$, $j = 1 : 4$. The rest 12 equations of the system determine matrix elements and $su1$, $su2$. The first 8 equations (9) determine recursively $e_i(j)$, $i = 5 : 12$, as functions of chosen 36 variables. The corresponding functions are substituted in the right parts of

$$tei = \sum_{j=1}^{12} \lambda_j e_i(j)^2; \quad uui = \sum_{j=1}^{12} \lambda_j e_i(j) \cdot e_{i+1}(j); \quad sui = \sum_{j=1}^{12} (\lambda_j e_i(j))^2.$$

The obtained system of 36 equations is solved by using NUMERIC, REDUCE 3.6. The original values of variables, corresponding to nonperturbed matrix are used as initial data. We have perfect results in both cases: when all eigenvalues are disturbed and when only λ_i with odd i are disturbed.

```
Inverse problem on lattice, odd eigenvalues are disturbed$ Time:
42340 ms  plus GC time: 180 ms la1; {0.973752448877015,15,9,10}$
Given spectrum.    Found spectrum.          Original spectrum. spi
:= {            sp := {                  spo:={ 0.973752448877015,
la=0.973752448877015,   0.967752448877010102993724 3,
1.96775244887701,  la=1.96775244887_677,
1.967752448877010102993724, 3.2098204263768,
la=2.38796601125_285,   3.203820426376799799402898,
4.2038204263768,   la=3.20982042_594845,
4.203820426376799799402898, 2.38796601125011,
la=3.38196601_231218,   2.38196601125010515179 5413,
3.38196601125011,  la=3.80217957_13511,
3.381966011250105151795413, 4.6240339887499,
la=4.2038204_3090632,   4.618033988749894848204587,
5.61803398874989,  la=4.6240339_7973626,
5.618033988749894848204587, 3.80217957362321,
la=4.7961795_8022541,   3.796179573623200200597102,
4.7961795736232,   la=5.618033988_01261,
4.796179573623200200597102, 6.038247551123,
la=6.038247551_38763,   6.032247551122989897006276,
7.03224755112299}$ la=7.0322475511_1343}$
7.032247551122989897006276}$ rsys := {0.0240180000000204,18}$
```

Here rsys - result of substitution of initial data into the system. The maximum of absolute value of errors is in the 18th equation. 10 iterations were produced. Precision was 15, accuracy - 9. The found 4 first components of basic eigenvectors were used to compute by primitive algorithm matrix with given spectrum presented in first column. The spectrum of found matrix is presented in the second column. The given and found spectra coincide in 7 digits after the point. In the third column the spectrum of original nonperturbed matrix is presented.

```
Inverse problem on lattice, all eigenvalues are disturbed$ Time:
```

```
60120 ms  plus GC time: 330 ms la1;
{0.96656910271957537947299 45,25,15,10}$ Given spectrum.
Found spectrum. spi := {                          sp := {
0.96656910271957537947299 45,      la=0.96656910271957537947_29_854,
1.96647984359861708513680 7,       la=1.96647984359861708513_6_768,
2.38207149468788668275895 ,        la=2.38207149468788668275_9_9047,
3.20193017848495282921303 9,       la=3.20193017848495282912_9_82246,
3.38317635316983951448421 5,       la=3.38317635316983951_4_559909,
3.80270423139162485327946 9,       la=3.80270423139162485_3_128874,
4.20101452641397898156672 6,       la=4.20101452641397898_1_837853,
4.61992850531211331724120 5,       la=4.61992850531211331_7_772882,
4.80134803417438716407653 1,       la=4.80134803417438716_4_00907,
5.62282364683016048551578 5,       la=5.62282364683016048_5_484107,
6.03679648740384693803452 7,       la=6.03679648740384693_8_044792,
7.03715759581301676921990 5}$      la=7.03715759581301676921_9_662}$
rsys := {0.02630705693190862570881944,23}$
```

This time maximum of absolute value of errors is in the 23th equation. 10 iterations were produced. Precision was 25, accuracy - 15. The given and found spectra coincide in 18 digits after the point. The calculations had been produced on SPP.

4 References

1. Serdyukova S.I., Russ.J. Numer. Anal. Math. Modelling, Vol.15, No.5, pp. 455-468(2000)).
2. B.N.Zakhariev, A.A.Suzko - Direct and Inverse Problems. Potentials in Quantum Scattering. Springer, Heidelberg,1990, Sec.2.4.
3. F.R.Gantmakher - Matrix Theory. Nauka, Moscow, 1967 (in Russian)
4. Hearn A.C. - REDUCE User's Manual. Version 3.6. Rand, Santa Monica, 1995. CA 90707-2138.

On Investigation of Involutivity of Differential Equation Systems in CAS 'MATEMATICA'

V.P. Shapeev[1]

ITAM SB RAS,
630090, Novosibirsk, Russia;
e-mail: shapeev@itam.nsc.ru

Compatibility analysis of partial differential equation systems is a kind of problem where Computer Algebra Systems (CAS) can be very helpful [4, 5, 6, 7, 8, 9, 10, 11]. There is mathematical definition of two equivalent algorithms in this problem: Cartan's one [1, 2] and one of Janet-Spencer-Kuranishi) [3]. In applications this problem demands long symbolic computations.

In terms of symbolic computations, compatibility analysis consists of a certain sequence of algebraic differential procedures. Given system PDE (S) we differentiate it by all independent variables and add to it all differential consequences, thus constructing the prolonged system (PS) (prolongation operation). Then, we calculate dimensions of the embedded linear spaces sequence, related to ranks of matrices of coefficients (PS). Coordinates in these spaces are dependent and independent variables in (S) and all derivatives of the latter in (PS). Then, we use the Cartan criterion that is expressed numerically through dimensions of the embedded spaces sequence. If Cartan criterion is not fulfilled, (PS) is taken for the given system and the next prolongation is made. It continues this way until Cartan criterion is fulfilled (system (PS) in involution) or contradiction is obtained in the form of equation of independent variables (system (S) is contradictory). The theory guarantees that one of these results is reached within a finite number of prolongations. Only solutions of system in involution (PS) are solutions of system (S).

Realisation in CAS of an algorithm, which would be suitable for analysis in a general case, is a very complicated problem. The main difficulty is that in a nonlinear case, while calculating ranks of the matrices, one has to make assumptions, that should be tested on a variety defined by (PS) in the space of all variables and derivatives. Here, in a nonlinear case the algorithm can branch out into many cases, and it is difficult to describe them all formally. However, particular cases show that the last difficulty can be overcome in the dialogue mode, because in practice the number of subcases can be small. Analysis of PDE compatibility as one of the algorithm's steps is a part of the Differential Constraints Method (DCM) or non-classical symmetries and of group analysis method for systems of partial differential equations (PDE). If the initial system (S) contains equations whose forms are indefinite because of the "indefinite elements" in them (for example, the form of

[1]This work was partially supported by the Russian Foundation for Basic Research (grants 99-01-00515 and 00-01-00370) and INTAS (proposal 1222).

quasi-linear equation coefficients is indefinite), then the Cartan criterion can be fulfilled by applying restrictions to the ranks of corresponding matrices of (PS) coefficients. In general case this will be expressed in the form of differential equations on the indefinite elements. If one manages to find solutions for such equations, thus he finds the form for the indefinite elements, which make the initial system (S) compatible. This form has to be defined in the differential constraints method while looking for constraints compatible with a given system, and also in the method of intermediate integral.

In the present paper MATHEMATICA is used first of all for formulating the compatibility conditions, whose operator form can be obtained before using CAS. For example, in the case of overdetermined systems of quasi-linear hyperbolic equations, the form of the compatibility conditions is shown in the terms of full differentiation operators [7]. A similar situation can take place in the case of overdetermined systems of linear equations.

Compatibility analysis of overdetermined uniform linear systems of partial differential equations of one unknown function is carried out with help of the well known Poisson brackets algorithm. Despite the relatively simple form, its applications lead to huge symbolic computations. In the present paper a program in MATHEMATICA system is realised, whose input data is the number of equations, number and designation of the variables, the form of the equations. It uses the standard functions of the system. The program makes one prolongation and finds number of algebraically independent equations in the prolonged system (PS) by calculating the ranks of the corresponding matrices. In the dialogue mode the number of prolongations is limited by the computer resources only. If system (S) has indefinite elements, then by demanding the corresponding matrices determinants to be zero, one can obtain conditions for the indefinite elements, which would make the prolonged system complete (i.e. in involution).

The following is a sample problem solved by the program. One-dimensional equations of polytropic gas dynamics in Lagrangian variables

$$u_t + p_x = 0, \quad \psi(x)p^{-k}p_t + u_x = 0, \quad \psi_t = 0 \tag{1}$$

by introducing potential ϕ by formulas $u = -\phi_x,\ p = \phi_t$, are reduced to equation

$$\psi(x)(\phi_t)^{-k}\phi_{tt} - \phi_{xx} = 0. \tag{2}$$

We seek intermediate integral for it in the form (variable t is not among the coefficients (2))

$$F(x, \phi, \phi_t, \phi_x) = C. \tag{3}$$

Equation (2) contains indefinite element $\psi(x)$ that is a function pertaining to the entropy and defining its gas particle distribution as a function of Lagrangian variable x. It follows from the known definition that intermediate integral (3) for equation (2) exists if in respect to function $F(x, \phi, \phi_t, \phi_x)$ two linear equations are fulfilled

$$\begin{aligned} L_1 &= F_{\phi_x} \pm \phi_t^{k/2}(\psi(x)^{-1/2})F_{\phi_t} = 0, \\ L_2 &= F_x\phi_t^{k/2-1}\psi(x)^{-1/2} \pm (1 \pm \phi_t^{k/2-1}\psi(x)^{-1/2}\phi_x)F_\phi = 0. \end{aligned} \tag{4}$$

In the dialogue mode two system prolongations are made (4). From the demand of the prolonged system involution (corresponding determinants equal zero) we obtain an

ordinary differential equation of the second order on indefinite element $\psi(x)$ with general solution $\psi = (c_1 x + c_2)^{k-4}$. Taken the latter into account, we can solve system (PS) and thus obtain intermediate integral (3). Moreover, in this case one can obtain solutions of equation (3). These solutions will have one-functional arbitrariness of the initial equation (2) and thus they will be solutions of gas dynamics equations (1). They generalize the known isoentropic Riemann waves on the case of non-isoentropic flows with two constants arbitrariness in entropy distribution on Lagrangian variable x. We do not have enough space here to go into details of this example.

As the other examples in the present paper, we sought conditions for existence of one and a system of intermediate integrals for equations of the second order with three independent variables. Similar to the definition in the two-dimensional case in the classical theory of differential equations, by intermediate integral for equation of the second order

$$F(x_1, x_2, x_3, u, u_1, u_2, u_3, u_{11}, ..., u_{33}) = 0 \tag{5}$$

in the three-dimensional case we will call differential equation of the first order

$$V(x_1, x_2, x_3, u, u_1, u_2, u_3) = c_1, \tag{6}$$

whose any non-special solution is a solution of equation (5). Here $u(x_1, x_2, x_3)$ is an unknown function, $u_i = \frac{\partial u}{\partial x_i}$, $u_{ij} = \frac{\partial u_i}{\partial x_j}$, c_1 is an arbitrary constant. System of equations (5), (6) is overdetermined and needs to be analysed for compatibility. Let us differentiate (6) once by each variable x_1, x_2, x_3:

$$\begin{aligned} &v_{x_1} + u_1 v_u + v_{u_1} u_{11} + v_{u_2} u_{21} + v_{u_3} u_{31} = 0, \\ &v_{x_2} + u_2 v_u + v_{u_1} u_{12} + v_{u_2} u_{22} + v_{u_3} u_{32} = 0, \\ &v_{x_3} + u_3 v_u + v_{u_1} u_{13} + v_{u_2} u_{23} + v_{u_3} u_{33} = 0. \end{aligned} \tag{7}$$

On non-special solution intermediate integral (6) can be resolved in respect to u_{i_0} which is one of u_i (i=1,2,3), while system (7) can be resolved in respect to three corresponding $u_{i_0 j}$, (j=1,2,3). By definition, substitution of found u_i in (5) turns it to identity.

Consider quasi-linear equation

$$\sum_{i,j=1}^{3} a_{ij} u_{ij} + l = 0 \qquad (a_{ij} = a_{ji}), \tag{8}$$

where coefficients a_{ij}, l are functions of $x_1, x_2, x_3, u, u_1, u_2, u_3$. In order to turn (8) to identity due to equations (6), all minors of the fourth order in the matrix of second derivative coefficients and free terms in system (7), (8)

$$\begin{pmatrix} v_{u_1} & v_{u_2} & v_{u_3} & 0 & 0 & 0 & v_{x_1} + u_1 v_u \\ 0 & v_{u_1} & 0 & v_{u_2} & v_{u_3} & 0 & v_{x_2} + u_2 v_u \\ 0 & 0 & v_{u_1} & 0 & v_{u_2} & v_{u_3} & v_{x_3} + u_3 v_u \\ a_{11} & 2a_{12} & 2a_{13} & a_{22} & 2a_{23} & a_{33} & l \end{pmatrix}$$

must equal zero. Hence, we obtain four proportions between v_{u_i}, v_{x_i}, that can be formulated in the following way

$$\begin{aligned} &\mu_{21} v_{u_2} = v_{u_1}, \qquad \mu_{31} v_{u_3} = v_{u_1}, \qquad \mu_{32} v_{u_3} = v_{u_2}, \\ &a_{11} v_{x_1} + a_{22} \mu_{21} v_{x_2} + a_{33} \mu_{31} v_{x_3} + (a_{11} u_1 + a_{22} \mu_{21} u_2 + a_{33} \mu_{31} u_3) v_u - l v_{u_1}, \end{aligned} \tag{9}$$

where $\mu_{ij} = (a_{ij} \pm \sqrt{a_{ij}^2 - a_{ii}a_{jj}})/a_{ii}$. Moreover, from this we obtain $\mu_{31} = \mu_{32}\mu_{21}$ — restriction on coefficients a_{ij}, $(i,j = 1,2,3)$. Equation system (9) is linear and uniform in respect to function v. Its compatibility is investigated with help of Poisson brackets. In doing this, the program in MATHEMATICA system described above essentially helps a mathematician to carry out the symbolic computations. Example. For equation

$$a(z)u_{11} + 2b(z)(u_{12} + u_{13}) + c(z)(u_{22} + 2u_{23} + u_{33}) = 0, \qquad z = u_2 + u_3$$

after one prolongation system (9) becomes complete (in involution). By its integration we find intermediate integral

$$v \equiv u_1 + \int \frac{cdz}{b \pm \sqrt{b^2 - ac}} = c_1.$$

In two-dimensional case a separate intermediate integral selects for an equation of the second order a class of solutions with one arbitrary function of one argument. In the three-dimensional case a separate intermediate integral selects for an equation of the second order a class of solution with one arbitrary function of two arguments. But a more essential distinction of the three-dimensional case from the two-dimensional one consists of the fact that an equation of the second order with three independent variables can have system of two intermediate integrals, which selects a class of solutions with one arbitrary function of one argument.

By a system of intermediate integrals for equations of the second order (5) we will call system of differential equations of the first order

$$\Psi(x_1, x_2, x_3, u, u_1, u_2, u_3) = c_1, \qquad \Omega(x_1, x_2, x_3, u, u_1, u_2, u_3) = c_2, \tag{10}$$

whose any non-special solution is a solution of equation (5).

Equation system (5), (10) is overdetermined. Its compatibility should be investigated by a general algorithm with use of Cartan criterion. A part of the analysis can be carried out before using CAS. Let us differentiate equations (10) by all independent variables. The obtained equations can be written in the form

$$AY = t, \tag{11}$$

where $Y = (u_{11}, u_{12}, u_{13}, u_{22}, u_{23}, u_{33},)'$ and $t = (t_1, t_2, ..., t_6)'$ are column vectors. From calculations we obtain that $det(A) \equiv 0$, therefore matrix A rank is less than six. It can be proved that A rank equals five. At that for compatibility of (10) it is necessary that rank of the extended matrix of system (11) also equals five. If the last condition is fulfilled in the space of variables $(x_1, x_2, x_3, u, u_1, u_2, u_3,)$ on variety (10) identically, then according to Cartan criterion it can be shown that system (10) is in involution with arbitrariness in one function of one argument. The last condition can be reduced to the form

$$(t_1\Psi_{u_1} + t_2\Psi_{u_2} + t_3\Psi_{u_3} - t_4\Omega_{u_1} - t_5\Omega_{u_2} - t_6\Omega_{u_6}). \tag{12}$$

In the non-linear case it is difficult to obtain a definite answer, whether equations (10) are a system of intermediate integrals for (5). Such an investigation is easier in the

case of quasi-linear and linear equations. With help of the program it is possible to carry out complete investigation in more simple cases, because the further investigations are reduced to substitution of some expressions into others, obtaining coefficients at certain variables in the problem, and to some other operations which can be easily programmed in MATHEMATICA. For example it is found out that there is a wide class of weakly nonlinear equations $\sum_{i,j=1}^{3} a_{ij}u_{ij} + \sum_{i=1}^{3} a_i u_i + l = 0, \quad a_{ij} = a_{ji}$, which have weakly nonlinear intermediate integrals $\Psi \equiv u_1 - du_3 - f = 0, \quad \Omega \equiv u_2 - gu_3 - h = 0$, where a_{ij}, a_i, d, g are functions of x, l, f, h are functions of x, u.

For wave equation

$$u_{11} - a^2(u_{22} + u_{33}) = 0, \tag{13}$$

where $a = const$, we have managed with the help of CAS to complete all sequence of the described algorithm's steps and obtain solutions, which are known functionally invariant solutions in a particular case. It is shown that among solutions of the intermediate integrals there are those which are not functionally invariant.

As another example, CAS MATHEMATICA was applied to investigation of involution of overdetermined systems, which appear in result of the differential constraints method (non-classical symmetries) application to one-dimensional gas dynamics equations [8]. Compared to the old version in REDUCE the new program version has new features: deduction of characteristic equations and of proportions along the characteristics for overdetermined systems under consideration, transition to different notations of initial equations and results in order to obtain them in a convenient form for subsequent processing.

We will mention here some peculiarities that we have found while solving this problem in CAS.

1) There are two essentially different ways in MATHEMATICA to differentiate complicated expressions. The first one is to apply differentiation functions available in the language. The second one is to reduce differentiation operation to rules of replacement of one expression by another. The first way is easier for new users. Programming the second way is similar in both CASs and are more labour-intensive, because it is easy to make a mistake in complicated cases, so a thorough program testing is needed.

In problems of compatibility analysis the first way leads to overgrowth of intermediate operations, because there can be many function arguments. For example, intermediate integral (6) has seven of them. In space of second derivatives indefinite elements will have thirteen arguments. The number of arguments essentially increases if we consider equation systems. In MATHEMATICA a particular function is written with all arguments to allow the system differentiate by a given argument. Of course, in end of solving problem, when differentiation is not needed anymore, we can set transformation rules, which rewrite the obtained expressions without function arguments. However, it leads to overgrowth of the program itself.

In using the second way, one can go without ascribing arguments to the function descriptor. Moreover, in CAS REDUCE function arguments can be mentioned only one time. The second way allows one to obtain shorter expressions, thus making process of preliminary analysis and integration of differential equations in the dialogue mode easier.

2) The transformation mentioned above, pertaining to removing arguments of complicated functions, demand description of different transformation rules in the program.

There is an opportunity in MATHEMATICA to write all necessary consequence of transformation as one compound operator with help of embeded simple operators. However, in this case such a compound operator often works much slower than in the case when separate transformations are realised by consequence of commands with simple operators of needed quantity. In the latter case the program is bigger. It has to be mentioned also that one compound transformation operator may be unable to make all transformations written in simple transformations, that is in the first case additional result transformations may be needed. This makes it of little use in solving the problems under consideration.

3) In application of operators (functions) with similar purposes to operations with fractional rational functions REDUCE often provides more simple expressions than MATHEMATICA. The latter for similar simplifications demands use of other additional functions, in particular function Simplify[].

4) Application of CAS to problems, whose solution is not guaranteed without realisation in the program of complete algorithm, demands that the user knows the complete algorithm of its solution and understands the results he can obtain, that is that he knows the problem and ways of its solving. Simple example. Characteristic matrix $Ch(\lambda)$ for two-dimensional equations of gas dynamics in plain ($\nu = 0$) and axisymmetric ($\nu = 1$) cases:

$$\begin{array}{c} u\tau_x + v\tau_y - \tau u_x - \tau v_y = \nu\frac{\tau v}{y}, \quad uu_x + v_u y + \tau p_x = 0, \\ uv_x + vv_y + \tau p_y = 0, \quad up_x + vp_y + A(u_x + v_y) = -\nu\frac{Av}{y}. \end{array} \tag{14}$$

has four proper values $\lambda_i, (i = 1, ..., 4)$. Here, $\lambda \equiv y' = dy/dx$, (u, v) are coordinates of the velocity vector along axes x, y; p is pressure, τ is specific volume, c is sound speed, $A = c^2/\tau$.

MATHEMATICA could not solve the characteristic equation in symbolic form. Two characteristic directions $y'_{2,3} = v/u$ are known to coincide with direction of gas particles velocity. If we eliminate consequently these two roots with help of command

$$Chp3[\lambda] = Together[Ch[\lambda]/(x - v/u)]//Simplify,$$
$$Chp2[\lambda] = Together[Chp3[\lambda]/(x - v/u)]//Simplify,$$

then in result command $Solve[Chp2[x] == 0, \lambda]$ produces two other (of a more complicated form) roots of the characteristic equation.

In more difficult cases CAS user can act in a similar way: to apply it only to a part of operations in a complicated algorithm of symbolic computations. The rest part of the operations can be realised in the dialogue mode, using knowledge of the problem specifics. Thus, the technique used here of applying CAS only to a part of calculations needed for complete involution analysis can be justified.

References

[1] Cartan E. Les systèmes différentiels extérieurs et leurs applications geometriques. Paris: Hermann, 1945. 215 p.

[2] Finikov S.P. Cartan exterior forms method in differential geometry. M.; L.: Gostekhizdat, 1948. 432 p, (Russian).

[3] Kuranishi M. Lectures on involutive systems of partial differential equations. Sao Paulo: Publ. Soc. Math., 1967. 77 p.

[4] Shurygin V.A., Yanenko N.N., On computer realization of algebraic/differential algorithms, in: *Problemy Kibernetiki,* 6, 1961, (Russian).

[5] Arais E.A., Shapeev V.P. and Yanenko N.N. Computer realization of Cartan's external forms, Dokl. AN SSSR. 1974, **214** (4), 737–738, (Russian).

[6] Peter A. Clarkson and Elizabeth L. Mansfield, Shwarz F. Symmetries and involution systems: Some experiments in Computer algebra // Topics in Soliton theory and Exactly solvable Nonlinear equations. Proceeding of Conf. on Nonlinear Evolution Equations, Solitons and Inverse scattering Transform. - 1978. - p. 290-299.

[7] Sidorov A.F., Shapeev V.P., and Yanenko N.N. The Method of Differential Constraints and its Application in Gas Dynamics. Nauka, Novosibirsk. 1984 – 272 p., (Russian).

[8] Meleshko S.V., Shapeev V.P. Application of computer algebra to analysis of differential equations systems compatibility // 6th International IMACS Conference on Applications of Computer Algebra IMACS ACA - June, 25-28, 2000, St. Petersburg, Russia - Abstracts, pp. 37-38.

[9] Ganzha V.G., Meleshko S.V., Murzin F.A., Shapeev V.P., Yanenko N.N. Computer realization of the algorithm for the investigation of compatibility of systems of partial differential equations, Doklady Akademii Nauk SSSR, 261(5), 1044-1046, 1981, (Russian).

[10] Ganzha V.G., Meleshko S.V., Shapeev V.P.. Automation of Compatibility Analysis of Quasilinear PDE Systems with the Aid of the REDUSE System. Computer Algebra and Its Applications to Mechanics. Nova Science Publishers, New York, 1993, p. 89-96.

[11] Meleshko S.V., Shapeev V.P.. An application of the differential constraints method for the two-dimensional equations of gas dynamics. Prikladnaya Matematika i Mekhanika, 63(6), 909-916, 1999, (Russian).

Moveable Singularities of Polynomial Differential Equations

S. Sobolevsky

Minsk, Belarus

Solutions of ordinary differential equations in complex domain can admit singularities of two types: fixed (depending on the coefficients singularities) and movable (which location varies from one solution to another). The investigation of ordinary differential equations solutions movable singularities has been started by the remarkable series of papers by Fuchs [1] (concerning the first-order equations), Painlevé [2,3] and Gambier [4] (concerning the second-order equations). The main results are summarized in Ince [5]. Further, Chazy [6], Garnier [7], and Bereau [8] have obtained partial results concerning the rational third-order differential equations.

In the present paper we consider the arbitrary-order polynomial differential equations of the form

$$w^{(n)} = P(w^{(n-1)}, w^{(n-2)}, \dots, w, z), \tag{1}$$

where n is an entire positive number and P is a nonlinear polynomial on w and its derivatives with coefficients, analytic on z in a certain domain $U \subset \mathbb{C}$.

We shall prove that every equation of class (1) admits movable singular points, i.e. not all of its solutions are analytic in U. Further we'll present an algorithm and the corresponding Mathematica program for investigation of these singularities.

Let us rewrite the equation (1) in the following form

$$w^{(n)} = \sum_{\chi \in S} a_\chi(z) \prod_{j=0}^{n-1} (w^{(j)})^{\chi_j}, \tag{2}$$

where S is a certain set of n-vectors with nonnegative integer coordinates $\chi = (\chi_0, \chi_1, \dots, \chi_{n-1})$ and a_χ are functions, analytic in domain U. For each $\chi \in S$ let us assume

$$|\chi| = \sum_{j=0}^{n-1} \chi_j, \ \nu(\chi) = \sum_{j=1}^{n-1} j\chi_j.$$

Consider two cases: I. $\max\limits_{\chi \in S} \nu(\chi) < n$ and II. $\max\limits_{\chi \in S} \nu(\chi) \geq n$.

I. First, consider a case $\max\limits_{\chi \in S} \nu(\chi) < n$. For all real t let us assume

$$\varphi(t) = \min_{\chi \in S} \{t(|\chi| - 1) - \nu(\chi) + n\}, \ S(t) = \{\chi : \chi \in S, t(|\chi| - 1) - \nu(\chi) + n = \varphi(t)\}.$$

It's easy to see that function φ has at least one negative real root $t = t_0$ because φ is a continuous on the real axe function, $\lim\limits_{t\to-\infty} = -\infty$, $\varphi(0) = n - \max\limits_{\chi\in S}\nu(\chi) > 0$. This root is rational because $t_0 = (n - \nu(\chi))/(|\chi| - 1)$ for all $\chi \in S_{t_0}$.

Then after introducing the parametric variable replacement $w = \alpha^{t_0}v, z = z_0 + \alpha x$ for $z_0 \in U$ we'll transform the equation (2) to the following form

$$v^{(n)} = \sum_{\chi\in S_0} a_\chi(z_0)\prod_{j=0}^{n-1}(v^{(j)})^{\chi_j} + \tilde{o}(\alpha), \tag{3}$$

where $S_0 = S(t_0)$ and $\tilde{o}(\alpha)$ denotes the sum of terms with positive rational degrees of α. For the equation (1) to be free of movable critical singular points it is necessary that the equation (3) for $\alpha \neq 0$ would admit no solutions with critical singularities in zero as well as the reduced equation

$$v^{(n)} = \sum_{\chi\in S_0} a_\chi(z_0)\prod_{j=0}^{n-1}(v^{(j)})^{\chi_j}. \tag{4}$$

Let us call the value t_0 the Bureau number for the equation (4).

The equation (4) admits solutions of the form

$$v = \lambda x^{t_0},$$

where λ satisfies an equation

$$t_0(t_0-1)\dots(t_0-n+1)\lambda = \sum_{\chi\in S_0} a_\chi(z_0)\left[\prod_{j=0}^{n-1} t_0(t_0-1)\dots(t_0-j+1)\right]\lambda^{|\chi|}. \tag{5}$$

Let us call the equation (4) regular if the equation (5) admits nonzero roots and irregular in the opposite case. The regular equations (4) with noninteger Bureau symbol obviously admit solutions with critical algebraic singular point in zero. The regular equations (4) with integer Bureau symbol admit solutions with pole in zero.

Let us show that the irregular equations (4) always admit solutions with critical singularities in zero.

Consider the irregular equation (4). In this case the right side of the equation (4) admits parametric family of solutions $v = \lambda x^{t_0}$ for arbitrary λ. Variable replacement $x = e^{s/\alpha^{1/M}}$, $v = x^{t_0}u$, $\alpha = \varepsilon^M$ for sufficiently great positive integer M reduces the equation (4) to the form

$$\varepsilon^m u^{(n)} = F(u^{(n-1)}, u^{(n-2)}, \dots, u, \varepsilon), \tag{6}$$

where m is a natural number; F is a polynomial on all variables, and $F(0_{\mathbb{C}^{n-1}}, u, 0) \equiv 0$ and $F(0_{\mathbb{C}^{n-1}}, u, \epsilon) \not\equiv 0$.

We'll need the following

Theorem 1. *Consider an equation*

$$\alpha^m w^{(n)} = F(w^{(n-1)}, w^{(n-2)}, \dots, w, z, \alpha) \tag{7}$$

where m and n are natural numbers and F is function, analytic in a neighborhood of some point $(w_0^{n-1}, w_0^{n-2}, \dots, w_0, z_0, 0) \in \mathbb{C}^{n+2}$. Let $w = \varphi_0(z)$ be a solution of the equation (7) for $\alpha = 0$ satisfying the initial conditions $\varphi^{(j)}(z_0) = w_0^j$, $j = \overline{0, n-1}$.

Then in every neighborhood of z_0 there exists such a point z_1 and a sequence $w = \varphi_\alpha(z)$ of solutions of the equation (7) for real positive α which, for $\alpha \to +0$, uniformly converges to $\varphi_0(z)$ in some neighborhood of z_1.

To prove theorem 1 we'll need the following two lemma.

Lemma 1. *Consider the system*

$$\begin{cases} \alpha^{k_g} u_g' = \sum\limits_{l=1}^{s} A_{gl}(z) u_l + F_g(\alpha u, \alpha v, z, \alpha), \ g = \overline{1,s} \\ v' = B(z)u + C(z)v, \end{cases} \tag{8}$$

where $u = (u_g, \ g = \overline{1,s})$, $u_g \in \mathbb{C}^{n_g}$, $n_g \in \mathbb{N}$; $v \in \mathbb{C}^m$; $z, \alpha \in \mathbb{C}$; $k_g \in \mathbb{N}$, $k_1 > k_2 > \dots > k_s$; $A_{gl}, B(z), C(z)$ are correspondingly the $n_g \times n_l$-, $m \times n$- and $m \times m$-matrixes with coefficients, analytic on z in the certain domain $U \subset \mathbb{C}$; F is a mapping from $\mathbb{C}^{n+m+2}$ to $\mathbb{C}^n$, analytic in a neighborhood of a set $\{(0_{\mathbb{C}^{n+m}}, z, 0) : z \in U\}$, where $n = \sum\limits_{g=1}^{s} n_g$, and $F(0_{\mathbb{C}^{n+m}}, z, 0) \equiv 0$. Let us assume that in the domain U the determinants of matrixes $\begin{pmatrix} A_{gl}, \ l = \overline{1,\theta} \\ l = \overline{1,\theta} \end{pmatrix}$ for $\theta = \overline{1,s}$ are nonzero.

Than for every positive real α there exists a solution $u = \varphi_\alpha(z)$, $v = \phi_\alpha(z)$ of the system (8) such that for $\alpha \to +0$ the sequences $\varphi_\alpha(z), \phi_\alpha(z)$ uniformly converge to zero in some closed subdomain $V \subset U$.

Proof of the lemma 1. We may consider $A_{gl} \equiv 0$ for $g < l$, because in the opposite case it is sufficient to introduce the following replacements:

$$u_g^0 = u_g^0, \ j = \overline{1,s}, \ A_{gl}^0 = A_{gl}, \ j,l = \overline{1,s},$$

$$u_g^\theta = u_g^{\theta-1} - \alpha^{k_\theta - k_g} A_{g\theta}^{\theta-1}(z)(A_{\theta\theta}^{\theta-1}(z))^{-1} u_1, \ g = \overline{\theta+1,s},$$

$$A_{gl}^\theta = A_{gl}^{\theta-1} - A_{g\theta}^{\theta-1}(A_{\theta\theta}^{\theta-1})^{-1} A_{\theta l}^{\theta-1}, \ g = \overline{\theta+1,s}, \ l = \overline{1,s},$$

$$\theta = \overline{1, s-1}.$$

The correctness of these replacements implies from that fact, that for $\theta = \overline{1,s}$ takes place

$$0 \neq \det \begin{pmatrix} A_{gl}(z), \ l = \overline{1,\theta} \\ l = \overline{1,\theta} \end{pmatrix} = \det \begin{pmatrix} A_{gl}^{g-1}(z), \ l = \overline{1,\theta} \\ l = \overline{1,\theta} \end{pmatrix} = \prod_{g=1}^{s} A_{gg}^{g-1}(z), \ \forall z \in U.$$

In this case, on a lemma condition, for all $g = \overline{1,s}$ and $z \in U$ we have $\det A_{gg}(z) \neq 0$. We may also consider, that $A_{gl} \equiv 0$ for $g > l$, because in the opposite case it is sufficient to introduce the following replacements:

$$\tilde{u}_g = u_g + A_{gg}^{-1} \left(\sum_{l=g+1}^{s} A_{gl}(z) u_l \right).$$

It is easy to see, that there exists such a point $z_0 \in U$, in which neighborhood matrixes A_{gg}, $g = \overline{1,s}$ could be represented as $A_{gg}(z) = Q_g(z)J^g(z)Q_g^{-1}(z)$, where $Q_g(z)$ are matrixes with analytic in a neighborhood of z_0 coefficients and nonzero in a neighborhood of z_0 determinant, $J^g(z) = \operatorname{diag}(J_j^g(z),\ j = \overline{1,\tau_g})$, $\tau_g \in \mathbb{N}$, $J_j^g(z)$ is a Jordan cell of dimension ζ_j^g with analytic in a neighborhood of z_0 diagonal element $\mu_j^g(z)$, and $\sum\limits_{j=1}^{\tau} \zeta_j^g = n_g$.

Having made in (8) a variable replacement $u_g = Q_g(z)x_g$, $g = \overline{1,s}$, we'll obtain

$$\begin{cases} \alpha^{k_g} x_g' = J^g(z)x_g + \tilde{F}_g(\alpha x, \alpha v, z, \alpha),\ g = \overline{1,s}, \\ v' = B(z)Q(z)x + C(z)v, \end{cases} \tag{9}$$

where $\tilde{F}_g(a,b,z,\alpha) \equiv Q^{-1}(z)F_g(Q(z)a,b,z,\alpha) - \alpha^k Q(z)^{-1}Q'(z)a$, $Q(z) = \operatorname{diag}(Q_j(z), j = \overline{1,s})$.

We will need the following proposition

Proposition 1. *Consider a linear differential system*

$$\frac{du}{dz} = J(z)u + b(z), \tag{10}$$

Where u is an n-vector function, $\tau \in \mathbb{N}$, $J(z)$ is a Jordan cell of dimension n with analytic in the convex closed area $V \subset \mathbb{C}$ diagonal element $\mu(z)$, b is an analytic in V n-vector function. Let's also assume that for some point $z_0 \in V$ and every point $z \in V$ the inequalities

$$\begin{aligned} &||b(z)|| < M, \\ &\operatorname{Re}[\mu(z)(z-z_0)/|z-z_0|] < -\sigma,\ z \neq z_0, \end{aligned} \tag{11}$$

where $M, \sigma > 0$, take place. Then the system (10) admits a private solution $u = \xi_{V,J,b}(z)$ such, that for all $z \in V$, the inequality

$$|\xi_{V,a,b}(z)| < M(\sum_{l=1}^{n} \sigma^{-l})$$

takes place.

Proof of the proposition 1. Let's rewrite the system (10) in the following form

$$\begin{cases} \frac{du_j}{dz} = \mu(z)u_j + u_{j+1} + b_j(z),\ j = \overline{1,n-1}, \\ \frac{du_n}{dz} = \mu(z)u_n + b_n(z), \end{cases} \tag{12}$$

where $u = (u_1, u_2, \ldots, u_n)$, $b = (b_1, b_2, \ldots, b_n)$.

In the case $n = 1$ the required solution $\xi_{V,a,b}$ may be found in the following way:

$$u(z) = \frac{z-z_0}{|z-z_0|} \int\limits_0^{|z-z_0|} \exp\Big\{\frac{z-z_0}{|z-z_0|} \int\limits_s^{|z-z_0|} a(z_0 + \tau\frac{z-z_0}{|z-z_0|})d\tau\Big\}b(s)ds.$$

For this solution the required inequality

$$|u(z)| < M \int\limits_0^{|z-z_0|} \exp\Big\{ \int\limits_s^{|z-z_0|} \operatorname{Re}[a(z_0 + \tau\frac{z-z_0}{|z-z_0|})\frac{z-z_0}{|z-z_0|}]d\tau\Big\}ds <$$

$$M\int\limits_0^{|z-z_0|}\exp\Big\{-\int\limits_s^{|z-z_0|}\sigma d\tau\Big\}ds =< M\int\limits_0^t \exp\{\sigma(s-t)\}ds = \frac{M}{\sigma}(1-e^{-t}) < M/\sigma$$

holds.

Then the system (12) admits a solution

$$u_n = \varphi_n(z) = \xi_{V,\mu(z),b_n(z)}(z),$$

$$u_j = \varphi_j(z) = \xi_{V,\mu(z),b_j(z)+\varphi_{j+1}(z)}(z),\ j=\overline{n-1,1},$$

which satisfies inequalities

$$|u_n| \le M/\sigma, |u_j| \le (M+|u_{j+1}|)/\sigma = M(\sum_{l=1}^{j}\sigma^{-l}), j=\overline{n-1,1}$$

i.e. is the required one. The propositon 1 proof in comlete.

Let's choose a complex number r such, that $|r|=1$, and for all g,j an inequality $\mathrm{Re}[r\mu_j^g(z_0)] \neq 0$ takes place (it is obvious, that such r exists). As a required area V we shall choose the closed neighborhood of point z_0 in a form of a rhomb with the center in z_0 and vertexes z_0+pr, z_0-pr, z_0+pri, z_0-qpri, where p,q are positive real numbers. It is easy to see, that choosing p and q small enough, it is always possible to achieve, that for any point $z\in V\setminus\{z_0+pr, z_0-pr\}$ and any g,j inequalities $\mathrm{Re}[(z-(z_0+pr))\mu_j^g(z)]\neq 0$ would take place, $\mathrm{Re}[(z-(z_0-pr))\mu_j^g(z)]\neq 0$. Then there will be $\sigma>0$ such, that for any g,j either $\mathrm{Re}[(z-(z_0+pr))\mu_j^g(z)] < -\sigma$ for all $z\in V$ or $\mathrm{Re}[(z-(z_0-pr))\mu_j^g(z)] < -\sigma$ for all $z\in V$. We shall also consider q small enough, such that $\mathrm{diam}V = 2pr$.

Let's build solutions $\varphi_\alpha(z) = (\varphi_\alpha^g,\ g=\overline{1,s})$, $\phi_\alpha(z)$ in the closed area V as limits of the following sequences. Let's assume

$$\varphi_{\alpha,1}(z)\equiv 0_{\mathbb{C}^n},\ \phi_{\alpha,1}(z)\equiv 0_{\mathbb{C}^m}.$$

For any $j\in\mathbb{N}$, $j>1$ let's assume

$$\varphi_{\alpha,j}^g(z) = \xi_{V,J^g,h_j^g(z)}(z),\ z\in V,\ g=\overline{s,1}$$

where $h_j^g(z) = \sum\limits_{l=g+1}^{s}\tilde{A}_{gl}(z)\varphi_{\alpha,j}^g(z) + \tilde{F}_g(\alpha\varphi_{\alpha,j-1}(z), \alpha\phi_{\alpha,j-1}(z), z, \alpha)$, and as $\phi_{\alpha,j}$ we shall choose a solution of a linear system

$$\frac{dv}{dz} = C(z)v + B(z)Q(z)\varphi_{\alpha,j-1}(z)$$

with initial conditions $v(z_0-pr)=0$.

Let's consider $\epsilon,\delta,M,N,\sigma>0$, such that for every $n+m$-vector which component modulo it is less than n-vector δ, $0<\alpha<\epsilon$ and all $z\in V$, inequalities take place

$$||\frac{\partial\tilde{F}}{\partial a}(a,z,\alpha)||, ||\frac{\partial\tilde{F}}{\partial\alpha}(a,z,\alpha)|| < M, \Rightarrow ||\tilde{F}(a,z,\alpha)|| < (|\alpha|+||a||)M$$

$$||B(z)Q(z)|| < N,\ ||C(z)|| < T,\ ||\mathrm{diag}(J_j(z),\ j = \overline{1,\tau})|| < N.$$

Let's assume $\Re = \sum\limits_{l=1}^{\max\limits_{1\le q\le s, 1\le\kappa\le\tau_q} \zeta_\kappa^q} \sigma^{-l}$.

By induction on j in view of proposition 1 it is simple to show, that at all $j \in \mathbb{N}$, $z \in V$ and $0 < \alpha < \chi = \min\{1, \epsilon, \delta/(1+\delta)/M/\Re, \delta T/(N\Re(e^{2prT}-1), 1/(2M\Re(1+2\delta(N(e^{2prT}-1)/T))))\}$ the inequalities

$$\begin{array}{l} ||\varphi_{\alpha,j}^q(z)|| < \alpha(1+\delta)M\Re < \delta,\ q = \overline{1,s}, \\ ||\phi_{\alpha,j}(z)|| < \alpha(1+\delta)MN\Re(e^{2prT}-1)/T, \\ ||\varphi_{\alpha,j+1}^q(z) - \varphi_{\alpha,j}^q(z)|| < \alpha(1+\max\{N(e^{2prT}-1)/T||\varphi_{\alpha,j-1}(z) - \varphi_{\alpha,j-2}(z)||, \\ ||\varphi_{\alpha,j}^q(z) - \varphi_{\alpha,j-1}^q(z)||\})M\Re < ||\varphi_{\alpha,j}^q(z) - \varphi_{\alpha,j-1}^q(z)||/2,\ q = \overline{1,s}, \\ ||\phi_{\alpha,j+1}(z) - \phi_{\alpha,j}(z)|| < N(e^{2prT}-1)/T||\varphi_{\alpha,j}(z) - \varphi_{\alpha,j-1}(z)||. \end{array} \tag{13}$$

take place.

From inequalities (13), for every fixed $\alpha \in (0,\chi)$ the sequences $\{\varphi_{\alpha,j}(z),\ j \in \mathbb{N}\}$, $\{\phi_{\alpha,j}(z),\ j \in \mathbb{N}\}$ uniformly on V converge to some limiting vector-functions $\varphi_\alpha(z)$, $\phi_\alpha(z)$, and the pair $\varphi_\alpha(z), \phi_\alpha(z)$ is a private solution of the system (9), satisfying the inequalities

$$||\varphi_\alpha(z)|| \le \alpha(1+\delta)M\Re, ||\phi_\alpha(z)|| \le \alpha(1+\delta)MN\Re(e^{2prT}-1)/T.$$

Thus sequences $\varphi_\alpha(z), \phi_\alpha(z)$ are the required ones. The proof of lemma 1 is now complete.

Lemma 2. *Consider an equation*

$$\alpha^m w^{(n)} = F(\alpha^{k_i} w^{(i)}, i = \overline{0,n-1}, z, \alpha^r) \tag{14}$$

where m, r, k_i are nonnegative rational numbers such that $m > k_{n-1} \ge k_{n-2} \ge \ldots \ge k_0 = 0$ and $r > 0$; F is a function, analytic in a neighborhood of a set $\{(0_{\mathbb{C}^n}, z, 0) : z \in U\}$ for some domain $U \subset \mathbb{C}$, and $F(0_{\mathbb{C}^n}, z, 0) \equiv 0$. Let us also assume that for $\tau = \max\limits_{k_i=0} i$ the condition $\frac{\partial F}{\partial w^{(\tau)}}(0_{\mathbb{C}^n}, z, 0) \not\equiv 0$ holds.

Then for every positive real α there exists a solution $w = \varphi_\alpha(z)$ of the equation (14) such that for $\alpha \to +0$ the sequence $\varphi_\alpha(z)$ uniformly converges to zero in some closed subdomain $V \subset U$.

Proof of lemma 2. Let us take rational numbers $0 < \rho_{\tau+1} < \rho_{\tau+2} < \ldots < \rho_n = m$ such that there exist indexes $\tau = i_0 < i_1 < i_2 < \ldots < i_s = n$ and positive rational numbers $0 < d_0 < d_1 < d_2 < \ldots < d_{s-1}$ such that $\frac{\partial F}{\partial w^{(i_j)}}(0_{\mathbb{C}^n}, z, 0) \not\equiv 0$ and for $i_j < k \le i_{j+1}$ the condition $\rho_k = \rho_{i_j} + d_j(k - i_j)$ holds.

Then lemma 2 follows from lemma 1 after a replacement $w^{(i)} = \alpha^{\rho_i} u_i$, $i = \overline{\tau+1, n-1}$, $w^{(i)} = u_i$, $i = \overline{0,\tau-1}$, $w^{(i^*)} = u_{i^*} - \left(\sum\limits_{i=0}^{\tau} \frac{\partial F}{\partial w^{(i)}}(0_{\mathbb{C}^n}, z, 0)u_i\right) / \frac{\partial F}{\partial w^{(\tau)}}(0_{\mathbb{C}^n}, z, 0)$, $\alpha = \varepsilon^M$, where M is a sufficiently great natural number.

Remark 1. *For every natural N, one may represent the solutions $\varphi_\alpha(z)$ in the following form*

$$\varphi_\alpha(z) = \sum_{j=1}^{N} \alpha^{j/M} \phi_j(z) + \alpha^{N/M} \tilde{\varphi}_\alpha(z), \tag{15}$$

where M is a number from lemma 2 proof, ϕ_j are analytic in V functions and the sequence $\tilde{\varphi}_\alpha(z)$ uniformly in V converges to zero for $\alpha \to +0$.

The representation (15) can be obtained by repeated application of the replacement $v = \alpha^{1/M}(\phi(z) + \tilde{v})$ which does not change the form of the equation (14) for certain ϕ.

Now theorem 1 implies from the obvious proposition, that every equation (7) may be transformed either to the form (14) or to the form (7) with $m = 0$ with the help of the replacement

$$w = \varphi_0(z) + \sum_{j=1}^{s} \alpha^{k_j}\phi_j(z) + \alpha^k v,$$

where $k > k_j > k_{j-1} \dots k_1 > 0$ are rational numbers, ϕ_j are some function, analytic in some point z_1 from a neighborhood of z_0.

From the proposition above and the remark 1 one can obtain

Remark 2. *For certain natural M and every natural N, one may represent the solutions $\varphi_\alpha(z)$ in theorem 1 in the following form*

$$\varphi_\alpha(z) = \varphi_0(z) + \sum_{j=1}^{N} \alpha^{j/M}\phi_j(z) + \alpha^{N/M}\tilde{\varphi}_\alpha(z)$$

where ϕ_j are analytic in a certain neighborhood V of the point z_1 functions and the sequence $\tilde{\varphi}_\alpha(z)$ uniformly in V converge to zero for $\alpha \to +0$.

Now from the theorem 1 and remark 2 for arbitrary constant C and positive real α and arbitrary natural N there exist solutions $u = \varphi_\alpha(\tilde{z})$ of the equation (6) of the form

$$\varphi_\varepsilon(s) = C + \sum_{j=1}^{N} \varepsilon^{j/K}\phi_j(s) + \varepsilon^{N/K}\tilde{\varphi}_\alpha(s),$$

where $\phi_j(s)$ are analytic in a certain neighborhood V of some point z_1 which in spite of the equation (6) autonomy may be considered to be zero, and the sequence $\tilde{\varphi}_\alpha(s)$ uniformly in V converges to zero for $\varepsilon \to +0$. It is simple to see that for sufficiently great N not all ϕ_j are constant, because $F(0_{\mathbb{C}^{n-1}}, u, \epsilon) \not\equiv 0$.

Then the equation (4) admits a solution of the form

$$w = \zeta_\alpha(x) = C + \sum_{j=1}^{N} \alpha^{j/KM}\phi_j(\alpha^{1/M}\ln x) + \alpha^{N/M}\tilde{\varphi}_{\alpha^{1/M}}(\alpha^{1/M}\ln x),$$

where some ϕ_j are nonconstant and therefore admits movable critical singular points.

From the aforesaid we have the following

Property 1. *The solutions of regular equation (4) with Bureau number p always admit poles or critical poles of order p in zero. The solutions of irregular equation (4) always admit critical singularities in zero.*

II. Now let us consider the equation (2) in the case $\max\limits_{\chi \in S} \nu(\chi) \geq n$. Assume

$$|\chi|_\tau = \sum_{j=\tau}^{n-1} \chi_j, \quad \nu(\chi) = \sum_{j=\tau+1}^{n-1} (j-\tau)\chi_j.$$

Consider two cases: 1) there exists such a τ^* that $\max\limits_{\chi\in S}\nu_{\tau^*}(\chi)=n-\tau^*$, 2) there does not exist such a τ^*.

1) In this case let us apply a variable replacement

$$z=z_0+\alpha x,\ w^{(j)}=w_0^j+\alpha w^j,\ j=\overline{0,\tau^*-1},\ w^{(\tau^*)}=u$$

for arbitrary constant z_0, w_0^j. For $\alpha=0$ we'll obtain a reduced equation of the form

$$u^{(m)}=\sum_{\chi\in H}c_\chi(u)\prod_{j=1}^{m-1}(u^{(j)})^{\chi_j}, \tag{16}$$

where $m=n-\tau^*$; H is a certain set of $m-1$-vectors with nonnegative integer coordinates $\chi=(\chi_1,\dots,\chi_{m-1})$ such that $\sum\limits_{j=1}^{m-1}j\chi_j=m$; c_χ are not identically equal to zero polynomials on u with constant coefficients. One can choose z_0, w_0^j so that the set H would not be empty. Let us denote the degree and the senior coefficient of polynomial c_χ by d_χ and k_χ correspondingly, and $|\chi|=\sum\limits_{j=1}^{m-1}\chi_j$.

The replacement

$$\begin{aligned}x&=\alpha\tilde{x},\\ u&=v/\alpha,\\ u'&=\alpha^{t-1}p\end{aligned}$$

for positive rational t reduces an equation (2) to a system

$$\begin{cases}p^{(m-1)}=\sum\limits_{\chi\in H}\alpha^{t(|\chi|-1)-d_\chi}k_\chi(p^{(m-2)})^{\chi_{m-1}}(p^{(m-3)})^{\chi_{m-2}}\dots p^{\chi_1}v^{d_\chi}+o(\alpha^{1/M}),\\ v'=p\alpha^{t+1},\end{cases} \tag{17}$$

where M is a sufficiently great natural number. For $t=\max\limits_{\chi\in H}d_\chi/(|\chi|-1)>0$ and $H_0=\{\chi\in H: d_\chi=t(|\chi|-1)\}$ the reduced at $\alpha=0$ system (17) takes form

$$\begin{cases}p^{(m-1)}=\sum\limits_{\chi\in H_0}k_\chi(p^{(m-2)})^{\chi_{m-1}}(p^{(m-3)})^{\chi_{m-2}}\dots p^{\chi_{r+1}}v^{d_\chi},\\ v'=0.\end{cases} \tag{18}$$

For constant v the first equation of this system is a (4)-type equation with Bureau number -1. Thus, from the property 1, p in the system (18) admits movable simple poles or critical singular points. Let $p=p_0(\tilde{x}), v=v_0$ be a solution of the system (18) where p_0 admits a pole or a critical singularity. Then the system (17) admits a solution of the form

$$p=p_0(\tilde{x})+o(\alpha^{1/M}),\ v=v_0+\left(\int p_0(\tilde{x})d\tilde{x}+o(\alpha^{1/M})\right)\alpha^{t+1},$$

i.e. admits movable critical singular points. It means that the equation (2) in the considered case admits them.

2) In this case let us denote $\tau^* = \max\limits_{\max\limits_{\chi\in S}\nu_{\tau^*}(\chi)>n-\tau}\tau$, $m = n - \tau^*$. Consider a function $\varphi(t) = \min\limits_{\chi\in S}\{t(|\chi|_{\tau^*+1} - 1) - \nu_{\tau^*}(\chi) + m\}$. This function is obviously continuous on the segment $[0,1]$, while $\varphi(0) = s - m_\tau < 0$ and $\varphi(1) = s - 1 - m_{\tau+1} > 0$. Hence the function φ has at least one root $t = t_0$ on an interval $(0,1)$. Assume $S_0 = \{\chi : \chi \in S, t_0(|\chi|_{\tau^*+1} - 1) - \nu_{\tau^*}(\chi) + m = 0$.

Then after introducing in the equation (2) the replacement

$$z = z_0 + \alpha x,\ w^{(j)} = w_0^j + \alpha w^j,\ j = \overline{0,\tau^*-1},\ w^{(\tau^*)} = w_0^{\tau^*} + \alpha^{t_0}u,$$

with constant w_0^j, z_0 one can obtain at $\alpha = 0$ the reduced equation of the form

$$u^{(m)} = \sum_{\chi\in S_0}\left(a_\chi(z_0)\prod_{j=0}^{\tau^*}(w_0^j)^{\chi_j}\right)(u^{(m-1)})^{\chi_{n-1}}(u^{(m-2)})^{\chi_{n-2}}\dots(u')^{\chi_{\tau^*+1}}. \qquad (19)$$

One can choose constants z_0, w_0^j so that the right side of the equation (19) would be not identically equal to zero. Then the equation (19) would be the (4)-type equation with noninteger Bureau number t_0. According to the property 1 it means that the reduced equation (19) as well as the initial equation (2) admits movable critical singular points.

From the aforesaid we have the following

Property 2. *The (2) satisfying an inequality* $\max\limits_{\chi\in S}\nu(\chi) \geq n$ *always admits movable critical singular points.*

From properties 1,2 the following theorem follows.

Theorem 2. *The equation (1) always admits movable singular points. It can be free of movable critical singular points only if for equation (2) the inequality* $\max\limits_{\chi\in S}\nu(\chi) < n$ *holds and the corresponding equation (4) is regular with an integer Bureau number.*

Let us continue the investigation of the equation (1) under the necessary for the absence of movable critical singular points conditions of theorem 2. After reducing it to the form (3) its solution can be found in the form

$$v(x) = \sum_{j=0}^{\infty} v_j(x), \qquad (20)$$

where the function v_0 satisfies the equation (4); the function v_1 satisfies the equation (4)

$$v^{(n)} = \sum_{j=0}^{n-1} L_j(v_0)v^{(j)}, \qquad (21)$$

where $L_j(v_0) = \sum\limits_{\chi\in S_0} a_\chi(z_0)\chi_j\left(\prod\limits_{i=0}^{n-1}(v_0^{(i)}(x))^{\chi_i}\right)/v_0^{(j)}(x)$; and v_j for $j > 1$ satisfy linear differential equations of the form

$$v^{(n)} = \sum_{j=0}^{n-1} L_j(v_0)v^{(j)} + f(v_0, v_1, \dots, v_{j-1}),$$

which homogeneous part coincides with (21) and f is a function which polynomially depends on $v_0, v_1, \ldots, v_{j-1}$.

The equation (21) is called the Euler equation for the equation (3). It can be also directly obtained from the equation (3) as a reduced equation after introducing the replacement $v = v_0 + \alpha\tilde{v}$ and assuming $\alpha = 0$.

The necessary condition for the equation (3) to be free of solutions with critical singularities in zero is that its Euler equation for any v_0 satisfying the equation (4) must admit no solutions with critical singularity in zero. The numerical test for the absence of movable critical singular points for the equation (3) can be obtained by consideration Euler equations for $v_0 = \lambda x^{t_0}$ where λ is an arbitrary nonzero solution of the algebraic equation (5). For this v_0 Euler equation takes form

$$v^{(n)} = \sum_{j=0}^{n-1} l_j(\lambda) v^{(j)} x^{j-n}, \tag{22}$$

where l_j are polynomials on λ with constant coefficients. This equation admits solutions of the form $v = x^r$ where r is an arbitrary solution of an equation

$$\prod_{i=0}^{n-1}(r-i) = \sum_{j=0}^{n-1} l_j \prod_{i=0}^{j-1}(r-i). \tag{23}$$

Furthermore if the equation (23) admits multiple roots then the considered Euler equation (22) admits solution with logarithmic branch points. So the solution of the equation (22) admit no critical singularities in zero if and only if the equation (23) roots are integer and pairwise different.

The special case when $l_0 = l_1 = \ldots = l_{n-1} = 0$ needs special consideration. In this case the equation (4) admits solutions with critical singularities in zero. Really, the after introducing the replacement $v = v_0 + \alpha\tilde{v}$ we'll find the obtained equation's solution in the form $\tilde{v} = C + \alpha^k g(x) + o(\alpha^k)$ where C is a constant, k is a natural number and g satisfied an equation $g^{(n)} = h(x)$ where h admits a pole of order less than n, i.e. g admits logarithmic branch point in zero.

From aforesaid one can obtain the following

Property 3. *For the equation (2) satisfying an inequality* $\max_{\chi \in S} \nu(\chi) < n$ *to be free of movable critical singular points it is necessary that for any* λ *satisfying the equation (5) the equation (22) has not identically equal to zero right side and the equation (23) roots are integer and pairwise different.*

If the condition of property 3 holds the further investigation of the equation (3) consists of sequential determination of $v_2, v_3, \ldots$ for all v_0 and λ and investigation of there singularities in zero.

From property 3 one can obtain the following corollary concerning the special class of equations (1).

Theorem 3. *The equation (1) which right side does not essentially depend on* $w^{(n-1)}$ *and* $w^{(n-2)}$ *always admits movable critical singular points.*

Proof of theorem 3. Let us denote the corresponding equation (23) roots by $r_1, r_2, \ldots, r_n$. According to the property 3 for the equation (1) to be free of movable critical singular points it is necessary that $r_1, r_2, \ldots, r_n$ would be integer and pairwise different. Let it be.

Under the theorem 3 conditions the equality $l_{n-1} = l_{n-2} = 0$ for the equation (23) holds. It means that $r_1 + r_2 + \ldots + r_n = 0 + 1 + \ldots + n - 1$ and $r_1^2 + r_2^2 + \ldots + r_n^2 = 0^2 + 1^2 + \ldots + (n-1)^2$. One can easily sea that it is possible only if $r_1, r_2, \ldots, r_n$ are equal to $0, 1, \ldots, n-1$. But this is possible only in the case $l_0 = l_1 = \ldots = l_{n-1} = 0$ which does not satisfy the condition of proposition 3. The proof of theorem 3 is now complete.

From theorem 3 one can sea that for equations (1) free of movable critical points the corresponding equation (4) would essentially depend on $v^{(n-1)}$ or $v^{(n-2)}$ and so its Bureau number would be -1 or -2.

From aforesaid we obtain the following algorithm of equation (1) movable singularities investigation.

Step 1. Check the inequality $\max\limits_{\chi \in S} \nu(\chi) < n$. If it does not hold the original equation admits movable critical points.

Step 2. Compute the Bureau number and check whether it is equal to -1 or -2. If it does not hold the original equation admits movable critical points.

Step 3. For any nonzero λ satisfying the equation (5) check whether at least one of $l_j(\lambda)$ is nonzero and whether the roots of equation (23) are integer and pairwise different. If it does not hold the original equation admits movable critical points.

This program realization of this algorithm has been obtained using the Mathematica 4 system in the form of the following module

```
   BeginPackage["ADE'"]
Diffinst::usage = "evalutes substitution in ODE"
Check1::usage = "check of condition 1"
Check2::usage = "check of condition 2"
Evalute::usage= "Evaluates the investigation of a given equation eq of order n in the given
or arbitrary point z0"
Begin["'Private'"]
Diffinst[de_, u_, x_, u1_, x2_, x1_] := {Module[{},
      de[[1]] /. {de[[2]] -> Function[{z}, u1[u[x1[z]], x1[z]]],
de[[3]] -> x2[x]}], u, x}
Check1[de_, n_] := Module[{u,x,a,z0},((Expand[a^n*
      Diffinst[de, u, x, Function[{u, x}, u], Function[{x}, z0 + a*x],
      Function[{z}, (z - z0)/a]][[1]]]) /. a -> 0 )=== 0]
Check2[de_, n_,u_,x_,a_,z0_] := Module[{de0,b},
      de0:=Expand[a^(n+1)*Diffinst[de, u, x, Function[{u, x}, u/a], Function[{x}, z0 +
a*x],
      Function[{z}, (z - z0)/a]][[1]]];
      b=If[((de0/.a->0)=!=0)&&((de0/.a->0)=!=ComplexInfinity),1,
      de0:=Expand[a^(n+2)*Diffinst[de, u, x, Function[{u, x}, u/a^2], Function[{x}, z0
+ a*x],
      Function[{z}, (z - z0)/a]][[1]]];
      If[((de0/.a->0)=!=0)&&((de0/.a->0)=!=ComplexInfinity),2,0]];{b,de0}]
Evalute[de_,n_,z0_]:=Module[{de0,d0,re,a,i,j,l,k,ks,eul,reul,r,r0,dec,dec0},
      Off[Power::infy];
      If[Not[Check1[de,n]],Print["Weight-power condition doesn't hold, so the considered
```

```
equation admits movable critical singularities"],
    Print["Weight-power condition holds"]; de0=Check2[de,n,de[[2]],de[[3]],a,z0];
    If[de0[[1]]==0,Print["Bureau number is different from 1 or 2, so the considered equation admits movable critical singularities"],
    Print["Bureau number is ",-de0[[1]]]; d0=de0[[2]]/.a->0; Print["The reduced equation is ",D[de[[2]][de[[3]]],{de[[3]],n}]==d0];
    re=Expand[de[[3]]^(de0[[1]]+n)*Diffinst[{D[de[[2]][de[[3]]],{de[[3]],n}]
-d0,de[[2]],de[[3]]}, de[[2]],de[[3]],
Function[{u,x},k*x^(-de0[[1]])],Function[{x},x],Function[{x},x]][[1]]];
    ks=Solve[re==0,k]; If[Length[ks]< =1,
Print["The considered equation is irregular and admits movable critical singularities"],
    ks=NSolve[re==0,k]; dec0=True;
    For[l=1;j=1,l¡=Length[ks],l++,If[(k/.ks[[l]])=!=0.,
    Print[j,"]. ",de[[2]]==(k/.ks[[l]])*de[[3]]^(-de0[[1]]);j++;
    Print["The corresponding Euler equation"];
    eul=Sum[(D[d0,D[de[[2]][de[[3]]],{de[[3]],i}]]/.{de[[2]]->Function[{z},
(k/.ks[[l]])*z^(-de0[[1]])}])*D[de[[2]][de[[3]]],{de[[3]],i}],{i,0,n-1}];
    If[eul===0.,Print["Euler equation is degenerative"];dec0=False,
    Print[D[de[[2]][de[[3]]],{de[[3]],n}]==eul];
    Print["admits roots"];
    reul=Expand[de[[3]]^(-r+n)*Diffinst[{D[de[[2]][de[[3]]],{de[[3]],n}]
-eul,de[[2]],de[[3]]},
de[[2]],de[[3]],Function[{u,x},x^r],Function[{x},x],Function[{x},x]][[1]]];
    r0=r/.NSolve[reul==0,r];
    Print[r0];
    dec=True; For[i=1,i¡=Length[r0],i++,
dec=(dec && (Abs[FractionalPart[r0[[i]]]]<0.0005))];
    If[dec,Print["Euler numbers seem to be integer"],Print["Some of the Euler numbers are noninteger"]]; dec0=(dec0 && dec)]]];
    If[dec0,Print["All necessary conditions for the considered equation to be free of movable critical singularities hold"],
    Print["The considered equation amdits movable critical singularities"]];
    ]]]]
End[];
EndPackage[]
```

The module provides a function "Evalute" which checks the conditions necessary for the given equation to be free of movable critical singularities. The syntax is following: Evalute[rhs, w, z, n, z0], where *rhs* is the equation's right-hand side, w is a dependant variable, z is an independant variable, n is the equation's order, and $z0$ is a point z_0 in which the investigation is being carried out. Here are the examples of this function applications.

Example 1. Second order equation $w'' = w^3$:

Input: Evalute[w'[z]^3, w, z, 2, z0]

Output: Weight-power condition doesn't hold, so the considered equation admits movable

critical singularities

Example 2. Abel equation $w' = w^3$:

Input: Evalute[w[z]^3, w, z, 1, z0]

Output: Weight-power condition holds Bureau number is different from 1 or 2, so the considered equation admits movable critical singularities

Example 3. Second order equation class $w'' = 2w'w + w^3 + a(z)w^2 + b(z)w' + c(z)w + d(z)$:

Input: Evalute[2*w'[z]*w[z] + w[z]^3 + a[z]*w[z]^2 + b[z]*w'[z] + c[z]*w[z] + d[z], w, z, 2, z0]

Output: Weight-power condition holds

Bureau number is -1

The reduced equation is w"[z]=w[z]^3+2w[z]w'[z]

1]. w==-0.732051/z

The corresponding Euler equation

w"[z]==3.0718w[z]/z^2-1.4641w'[z]/z

admits roots

{-2,1.5359}

Some of the Euler numbers are noninteger

2]. w==2.73205/z

The corresponding Euler equation

w"[z]==16.9282w[z]/z^2+5.4641w'[z]/z

admits roots

{-2,8.4641}

Some of the Euler numbers are noninteger

The considered equation amdits movable critical singularities

Example 4. Chazy barrier equation [6] $w''' = ww'' - 2w'^2$:

Input: Evalute[w[z]*w"[z] - 2*w'[z]^2, w, z, 3, z0]

Output: Weight-power condition holds

Bureau number is -1

The reduced equation is w"'[z]==-2w'[z]^2+w[z]w"[z]

The considered equation is irregular and admits movable critical singularities

Example 5. First Painlevé equation $w'' = 6w^2 + z$:

Input: Evalute[6*w[z]^2 + z, w, z, 2, z0]

Output: Weight-power condition holds

Bureau number is -2

The reduced equation is w"[z]==6w[z]^2

1]. w==1/z^2

The corresponding Euler equation

w"[z]=12w[z]/z^2

admits roots

{-3,4}

Euler numbers seem to be integer

All necessary conditions for the considered equation to be free of movable critical singularities hold

Summary

In the present paper the nonlinear arbitrary-order polynomial differential equations are considered. It is shown that there solutions always admit movable singularities and the algorithm of these singularities investigation is given. The program realization of this algorithm in Mathematica 4 system is presented.

References

1. *L. Fuchs.* Über Differentialgleichungen deren Integrale feste Verzweigungspunkte besitzen, *Sitz. Akad. Wiss. Berlin* 32:699-710 (1884).

2. *P. Painlevé.* Mémoire sur les équations différentielles dont l'intégrale générale est uniforme, *Bull. Soc. Math. France* 28:201-261 (1900).

3. *P. Painlevé.* Sur les équations différentielles du second ordre et d'ordre supérieur dont l'intégrale générale est uniforme, *Acta. Math.* 25:1-85 (1902).

4. *B. Gambier.* Sur les équations différentielles du second ordre et du premier degré dont l'intégrale générale est à points critiques fixes, *Acta. Math.* 33:1-55 (1910).

5. *E.L. Ince. Ordinary differential equations*, Dover Publications, New York, 1956 (originally published by Longmans & Green, London and New York, 1926).

6. *J. Chazy.* Sur les équations différentielles du troisième ordre et d'ordre supérier dont l'intégrale générale a ses points critiques fixes, *Acta. Math.* 34:317-385 (1911).

7. *R. Garnier.* Sur des équations différentielles du troisième ordre dont l'intégrale générale est uniforme et sur une classe déquations nouvelles d'ordre supérier dont l'intégrale générale a ses points critiques fixes, *Ann. Sci. École Norm. Sup.* 29:1-126 (1912).

8. *F. Bureau.* Differential equations with fixed critical points, *Ann. di Math.* 66:1-116 (1964).

Bargmann-Darboux Transformations for Time-Dependent Quantum Equations

A.A. Suzko [1]

Joint Institute for Nuclear Research,
Dubna, Russia

A procedure is elaborated of constructing a time-dependent periodic Hamiltonian for which Schrödinger equations admit analytic solutions. The method is based on the transformation of time-independent soluble problems into time-dependent ones by a set of unitary time-dependent transformations and a proper choice of initial states. A class of 2×2 *periodic time-dependent potential matrices with cyclic matrix solutions is constructed in a closed explicit form. The expectation value of Hamiltonian, total, dynamical, and geometric phases are derived in terms of the obtained solutions.*

1 Introduction

Extensive study of time-dependent quantum oscillators [1]–[5], the study of quantum systems with a dynamical semisimple Lie group [6] have shown that an algebraic approach is a very useful tool for investigation of the evolution of dynamic quantum systems. The method of separation of variables [7], and the supersymmetry method [8] were suggested for construction of time-dependent exactly solvable potentials. In this paper, we elaborate the technique of constructing a periodic time-dependent Hamiltonian admitting exact solutions with the use of an exactly soluble time-independent Hamiltonian, unitary time-dependent transformations and a proper choice of initial states.

The technique of Bargmann-Darboux transformations and the supersymmetry method in quantum mechanics yield a variety of soluble stationary models [9, 10, 11]. Each of these models, to our mind, can be generalized to obtain the corresponding family of time-dependent exactly soluble Hamiltonians. Our main findings are as follows. First, we explicitly show how to construct exactly soluble time-dependent generalizations of any exactly soluble time-independent model for a 2×2 Hamiltonian. We do this by using unitary time-dependent transformations which change time-independent problems into time-dependent ones and by choosing initial states so that to derive cyclic solutions which require the initial states to be eigenstates of a time-independent Hamiltonian. We show how to generate a large family of time-dependent models from one time-independent model by employing a family of special time-dependent operators of transformation. Finally, as an application of our method, we present explicit expression of an expectation value of

[1]Permanent address: Radiation Physics and Chemistry Problems Institute, Academy of Sciences of Belarus, Minsk

Hamiltonian, the nonadiabatic geometric phase, total and dynamical phases which are derived in terms of the obtained cyclic solutions. In particular, we demonstrate how to construct a set of periodic time-dependent Hamiltonians whose expectation values do not depend on time. This procedure can be used for modeling quantum wells and wires.

Low-dimensional physical systems have attracted considerable attention as from theoretical and as well as from the practical point of view. Recent achievements in microfabrication technology give opportunities to construct two-dimensional quantum wells, superlattices, quantum wires and dots with properties of particle confinement. As it was shown in [12] and [13] that under definite conditions a particle can be localized in a nonuniform high frequency electromagnetic field. Recently Tralle [14] following Cook et.al. [15] considered a particle motion in a rapidly oscillating field of the form $V(r,t) = V(r)\cos\omega t$ in terms of the Shcrödinger equations with periodic and aperiodic $V(r)$, correspondingly. The main result of these authors is that for a sufficiently high frequency ω the time-dependent potential is, in a sense, equivalent to the time-independent effective potential and as a sequence a particle can be confined. This result is equivalent to the result [13] where the classical movement was considered. Here we generate more complicate time-dependent potentials which with special initial functions give the effect like time-independent potentials, in particular, the expectation values of the time-dependent Hamiltonian for cyclic solutions do not dependent on time. It means that these states behave equivalent stationary states with conserve energy.

2 Construction of a time-dependent Hamiltonian

To simplify the description below, we give a scheme of generation of a time-dependent Hamiltonian with corresponding solutions from a time-independent one [16]. Suppose that the state $|\Psi(t)>$ of a dynamical system evolves according to the matrix Schrödinger equation

$$i\frac{\partial|\Psi(r,t)>}{\partial t} = H(r,t)|\Psi(r,t)> \tag{1}$$

with $\hbar = 1$ and T periodic time-dependent Hamiltonian, $H(t) = H(t+T)$,

$$H(r,t) = -\nabla_r^2 + V(r,t), \tag{2}$$

the potential matrix $V(r,t) = \{V_{ij}(r,t)\}$ is Hermitian. Our goal is to give the procedure of obtaining a wide class of time-dependent Hamiltonians $H(t)$ for which exact solutions of (1) can be found. To this end, we use the time-independent Hamiltonian

$$\widetilde{H}(r) = -\nabla_r^2 + V(r) \tag{3}$$

with a real symmetric potential matrix $V(r)$ and $\hbar^2/2m = 1$, and a unitary time-dependent transformation $\mathcal{S}(t)$ [6, 16]

$$|\Psi(r,t)> = \mathcal{S}(t)|\Phi(r,t)>, \tag{4}$$

by means of which the known time-independent Hamiltonian (3) is changed into the time-dependent one

$$H(t) = \mathcal{S}(t)\widetilde{H}\mathcal{S}^{\dagger}(t) + i\dot{\mathcal{S}}(t)\mathcal{S}^{\dagger}(t). \tag{5}$$

Here $|\Phi(r,t)>$ satisfies the equation of motion with the time-independent Hamiltonian $\widetilde{H}(r)$

$$i\frac{\partial|\Phi(r,t)>}{\partial t} = \widetilde{H}(r)|\Phi(r,t)> \tag{6}$$

and it is of the form

$$|\Phi(r,t)> = \exp(-i\widetilde{H}(r)t)|\Phi(r,0)>. \tag{7}$$

It is evident, the solutions $|\Psi(r,t)>$ and $|\Phi(r,t)>$ can be defined, at corresponding choice of initial functions, by solutions of the time-independent problem

$$\widetilde{H}|\Phi(\widetilde{\mathcal{E}})> = \widetilde{\mathcal{E}}|\Phi(\widetilde{\mathcal{E}})>. \tag{8}$$

If the system of Schrödinger equations (8) with some known time-independent Hamiltonian $\widetilde{H}(r)$ is exactly soluble, the system of equations (1) with the time-dependent Hamiltonian (5) admits exact solutions too.

2.1 Construction of a 2×2 T-periodic Hamiltonian $H(t)$

Now consider reconstruction of the 2×2 periodic time-dependent Hamiltonian taken in the form (2). We start with the time-independent Hamiltonian (3) with the 2×2 real symmetric potential matrix $V(r)$, $V_{12}(r) = V_{21}(r)$. By means of a unitary time-dependent transformation taken in the form

$$\mathcal{S}(t) = \exp(-i\mathbf{s}\cdot\mathbf{h}(t)) = \exp(-i\sum_{i=1}^{3} s_i h_i(t)) \tag{9}$$

the time-independent Hamiltonian (3) with regard to equations (1) and (5) turns into the time-dependent Hamiltonian

$$H(r,t) = -\nabla_r^2 + \exp(-i\mathbf{s}\cdot\mathbf{h}(t))V(r)\exp(i\mathbf{s}\cdot\mathbf{h}(t)) + (\mathbf{s}\cdot\dot{\mathbf{h}}(t)). \tag{10}$$

Here $\mathbf{s} = (1/2)\boldsymbol{\sigma}$ is the spin operator, $\boldsymbol{\sigma} = (\hat{\sigma}_1, \hat{\sigma}_2, \hat{\sigma}_3)$ and $\hat{\sigma}_i$ are the Pauli matrices and a dot means a time-derivative. The solutions of (1) with Hamiltonian (10), according to (4), are represented as

$$|\Psi(r,t)> = \exp(-i\mathbf{s}\cdot\mathbf{h}(t)\exp(-i\widetilde{H}(r)t)|\Phi(r,t=0)>. \tag{11}$$

It is convenient to present the 2×2 Hamiltonian (3) by the sum of diagonal and zero trace matrices as

$$\widetilde{H}(r) = -\nabla_r^2 + \begin{pmatrix} V_{11}(r) & V_{12}(r) \\ V_{21}(r) & V_{22}(r) \end{pmatrix} = \Big(-\nabla_r^2 + q(r)\Big)\hat{I} + 2(\mathbf{s}\cdot\mathbf{B}(r)), \tag{12}$$

with the evident notations : $q(r) = (V_{11}(r) + V_{22}(r))/2$, $B_1(r) = V_{12}(r)$, $B_2(r) = 0$, $B_3(r) = (V_{11}(r) - V_{22}(r))/2$ and $\hat{I}$ is the identity matrix. Then the time-dependent Hamiltonian (10) can be represented as

$$H(r,t) = \Big(-\nabla_r^2 + q(r)\Big)\hat{I} + 2\exp(-i\mathbf{s}\cdot\mathbf{h}(t))(\mathbf{s}\cdot\mathbf{B}(r))\exp(i\mathbf{s}\cdot\mathbf{h}(t)) + (\mathbf{s}\cdot\dot{\mathbf{h}}(t)). \quad (13)$$

Obviously, the transformation (9) does not change the first term of (12) and transforms the second term. It is evident that the Hamiltonian for the two coupled system of equations corresponds to the three- or two-dimensional problem with coordinates $B_i(r)$ dependent on the extra parameter r. Such a Hamiltonian may be considered as an analog of the inhomogeneous magnetic field $\mathbf{B}(r)$ or a model Hamiltonian for describing a two-level atom.

In terms of the evolution operator $\mathcal{U}(t) = \mathcal{U}(t,0)$, the solution $|\Psi(r,t)>$ is

$$|\Psi(r,t)>= \mathcal{U}(t)|\Psi(r,0)>, \quad \mathcal{U}(0) = 1. \quad (14)$$

In the case when $|\Psi(r,0)>= |\Phi(r,0)>$, it is easy to find from (11) and (14) a very important relationship between the operators $\mathcal{U}(t)$ and $\mathcal{S}(t)$

$$\mathcal{U}(t) = \mathcal{S}(t))\exp -i\widetilde{H}t = \exp(-i\mathbf{s}\cdot\mathbf{h}(t))\exp(-i\widetilde{H}t). \quad (15)$$

The evolution operator in one period is written as

$$\mathcal{U}(T) = \exp(-i\mathbf{s}\cdot\mathbf{h}(T))\exp(-i\widetilde{H}T). \quad (16)$$

In assumption that $\mathcal{S}(T)$ and $\widetilde{H}$ commute $[\mathcal{S}(T)\widetilde{H}] = 0$, we easily obtain that $\mathcal{U}(T)$ and $\widetilde{H}$ commute. Indeed,

$$\mathcal{U}^\dagger(T)\widetilde{H}\mathcal{U}(T) = \mathcal{S}^\dagger(T)\exp(-i\widetilde{H}T)\widetilde{H}\exp(i\widetilde{H}T)\mathcal{S}(T) = \mathcal{S}^\dagger(T)\widetilde{H}\mathcal{S}(T) = \widetilde{H},$$

i.e.,

$$[\mathcal{U}(T)\widetilde{H}] = 0 \quad (17)$$

and therefore $\mathcal{U}(T)$ and $\widetilde{H}$ have common eigenvectors $|\Phi_\nu>$.

Now consider cyclic solutions that after one period T ($T = 2\pi/\omega$) are recovered up to the phase, i.e., initial states $|\Psi_\nu(0)>$ are eigenvectors of $\mathcal{U}(T)$ and because of (17) they are eigenvectors of the effective Hamiltonian $\widetilde{H}$. This leads to

$$|\Psi_\nu(r,T)>= \mathcal{U}(T)|\Psi_\nu(r,0)>= \exp(-i\beta_\nu)|\Psi_\nu(r,0)>, \quad (18)$$

$$|\Psi_\nu(0)>= |\Phi_\nu>, \quad (19)$$

where β_ν is the total phase. From this, the matrix representation of $\mathcal{U}(T)$ can be written as

$$\mathcal{U}(T) = \sum_\nu \exp(-i\beta_\nu)|\Phi_\nu><\Phi_\nu|$$

and $\mathcal{U}(T)$ has a set of linearly independent eigenvectors $|\Phi_\nu>$ with eigenvalues $\exp(-i\beta_\nu)$. With allowance for (11) the recurrent solutions at any time are written as

$$|\Psi_\nu(r,t)>=\exp(-i\mathbf{s}\cdot\mathbf{h}(t))\exp(-i\widetilde{E}_\nu t)|\Phi_\nu(r)>. \tag{20}$$

It is evident now, to determine the cyclic solutions we need time-independent solutions. Thus, whenever $\widetilde{H}(r)$ is an exactly soluble time-independent Hamiltonian, the properly generated time-dependent system of equations has cyclic exact solutions. The evolution of an arbitrary initial state $|\Psi(r,0)>=\sum_\nu \alpha_\nu|\Psi_\nu(r,0)>$ can be represented as a superposition of a basis set of recurrent linearly independent vector solutions, $|\Psi(r,t)>=\sum_\nu \alpha_\nu|\Psi_\nu(r,t)>$.

Now we can proceed to calculations of geometric phase, associated with the evolution of cyclic solutions, dynamical phase and the expectation value of Hamiltonian. To find the dynamical phase δ_ν, we need the expectation value $\epsilon_\nu(t)$ of $H(t)$,

$$\begin{aligned}\delta_\nu &= \int_0^T \epsilon_\nu(t)dt = \int_0^T <\Psi_\nu(r,t)|H(t)|\Psi_\nu(r,t)>dt \\ &= \int_0^T <\Psi_\nu(r,0)|\mathcal{U}^\dagger(t)H(t)\mathcal{U}(t)|\Psi_\nu(r,0)>dt.\end{aligned} \tag{21}$$

By using (15) and (5) we have

$$\mathcal{U}^\dagger(t)H(t)\mathcal{U}(t)=e^{i\widetilde{H}(r)t}\Big(\widetilde{H}(r)-i\mathcal{S}^\dagger(t)\dot{\mathcal{S}}(t)\Big)e^{-i\widetilde{H}(r)t}.$$

Since for cyclic solutions $|\Psi_\nu(r,0)>=|\Phi_\nu(r)>$ and $|\Phi_\nu(r)>$ is an eigenstate vector of $\widetilde{H}(r)$ with an eigenvalue $\widetilde{\mathcal{E}}_\nu$ we express the expectation value $\epsilon_\nu(t)$ and the dynamical phase δ_ν as

$$\epsilon_\nu(t)=\widetilde{\mathcal{E}}_\nu- <\Phi_\nu(r)|i\mathcal{S}^\dagger(t)\dot{\mathcal{S}}(t)|\Phi_\nu(r)>, \tag{22}$$

$$\delta_\nu=\widetilde{\mathcal{E}}_\nu T-\int_0^T <\Phi_\nu(r)|i\mathcal{S}^\dagger(t)\dot{\mathcal{S}}(t)|\Phi_\nu(r)>dt. \tag{23}$$

Hence, the geometric phase φ_ν, given by removing the dynamical phase from the total phase β_ν, is

$$\varphi_\nu=\beta_\nu-\delta_\nu=\beta_\nu-\widetilde{\mathcal{E}}_\nu T+\int_0^T <\Phi_\nu(r)|i\mathcal{S}^\dagger(t)\dot{\mathcal{S}}(t)|\Phi_\nu(r)>dt. \tag{24}$$

It should be noted that for a time-periodic and unitary $\mathcal{S}(t)$, the expression (24) for the geometric phase is simplified

$$\varphi_\nu=\int_0^T <\Phi_\nu(r)|i\mathcal{S}^\dagger(t)\dot{\mathcal{S}}(t)|\Phi_\nu(r)>dt. \tag{25}$$

Really, in this case in contrast with (16) the time evolution operator after one period is $\mathcal{U}(T)=\exp(-i\widetilde{H}(r)T)$. Since $|\Psi_\nu(0,r)>$ is an eigenstate of $\mathcal{U}(T)$ with eigenvalue

$\exp(-i\beta_\nu)$ then $|\Psi_\nu(0,r)>$ is also eigenstate of $\widetilde{H}$ with eigenvalue β_ν/T. At the same time, since eigenvalues of $\widetilde{H}$ are $\widetilde{\mathcal{E}}_\nu$, then $\beta_\nu/T = \widetilde{\mathcal{E}}_\nu$ and (25) is valid.

Examples. Consider some particular examples of constructing a time-dependent Hamiltonian (13) with the corresponding solutions by using $SU(2)$ transformation (9) in which the components of $\mathbf{h}(t)$ are linear functions of time, $h_i(t) = \omega_i t$

$$\mathcal{S}(t) = \prod_{i=1}^{3} \mathcal{S}_i(t) = \exp(-i\mathbf{s}\cdot\mathbf{w}t).$$

1 case. Let $\mathcal{S}(t) = \mathcal{S}_3(t) = \exp(-i\hat{\sigma}_3\omega t/2)$ where ω is some constant angular velocity. Then, in accordance with (5) or (13) the time-independent Hamiltonian (12) is transformed into the time-dependent one as

$$H(r,t) = \left(-\nabla_r^2 + q(r)\right)\hat{I} + \begin{pmatrix} \frac{V_{11}(r)-V_{22}(r)+\omega}{2} & V_{12}(r)\exp(-i\omega t) \\ V_{21}(r)\exp(i\omega t) & -\frac{V_{11}(r)-V_{22}(r)+\omega}{2} \end{pmatrix}. \tag{26}$$

It is evident that $H(r,t)$ is time-periodic, $H(r,t=0) = H(r,t=T)$, $T = 2\pi/\omega$. Its Routhian is $\widetilde{H}(r)$ connected with $H(r,t=0)$ by $H(r,0) = \tilde{H}(r) + \frac{\omega}{2}\hat{\sigma}_3$. By using (20) the cyclic solutions can be written as

$$|\Psi_\nu(r,t)>= \begin{pmatrix} \exp(-i(\widetilde{\mathcal{E}}_1^\nu + \omega_3/2)t) & 0 \\ 0 & \exp(-i(\widetilde{\mathcal{E}}_2^\nu - \omega_3/2)t) \end{pmatrix} |\Phi_\nu(r)>, \tag{27}$$

where $\widetilde{\mathcal{E}}_\alpha^\nu = (\widetilde{\mathcal{E}}_\nu - \Delta_\alpha)$ and Δ_α can be the energy of inner structure (threshold).

2 case. If $\mathcal{S}(t)$ is chosen as an operator of rotation around y-axis

$$\mathcal{S}_2(t) = \exp(-i\hat{\sigma}_2\omega_2 t/2) = \begin{pmatrix} \cos(\omega_2 t/2) & -\sin(\omega_2 t/2) \\ \sin(\omega_2 t/2) & \cos(\omega_2 t/2) \end{pmatrix}, \tag{28}$$

then in accordance with (13) the time-dependent Hamiltonian $H(t)$ takes another form

$$H(r,t) = (-\nabla_r^2 + q(r))\hat{I} + \tag{29}$$

$$+ \begin{pmatrix} \frac{V_{11}-V_{22}}{2}\cos\omega_2 t - V_{12}\sin\omega_2 t & V_{12}\cos\omega_2 t + \frac{V_{11}-V_{22}}{2}\sin\omega_2 t - i\omega_2/2 \\ V_{12}\cos\omega_2 t + \frac{V_{11}-V_{22}}{2}\sin\omega_2 t + i\omega_2/2 & -(\frac{V_{11}-V_{22}}{2}\cos\omega_2 t - V_{12}\sin\omega_2 t) \end{pmatrix}.$$

In accordance with (20) the recurrent solutions of equation (1) are immediately written as

$$|\Psi_\nu(r,t)>= \begin{pmatrix} \cos(\omega_2 t/2)\exp(-i\widetilde{\mathcal{E}}_1^\nu t) & -\sin(\omega_2 t/2)\exp(-i\widetilde{\mathcal{E}}_2^\nu t) \\ \sin(\omega_2 t/2)\exp(-i\widetilde{\mathcal{E}}_1^\nu t) & \cos(\omega_2 t/2)\exp(-i\widetilde{\mathcal{E}}_2^\nu t) \end{pmatrix} |\Phi_\nu(r)>. \tag{30}$$

3 case. Let $S(t)$ be chosen as an operator of rotation around x-axis

$$\mathcal{S}_1(t) = \exp(-i\hat{\sigma}_1\omega_1 t/2) = \begin{pmatrix} \cos(\omega_1 t/2) & -i\sin(\omega_1 t/2) \\ -i\sin(\omega_1 t/2) & \cos(\omega_1 t/2) \end{pmatrix}. \tag{31}$$

Then, from (13) the time-dependent Hamiltonian $H(t)$ is

$$H(r,t) = \Big(-\nabla_r^2 + q(r)\Big)\hat{I} \tag{32}$$

$$+\begin{pmatrix} \frac{V_{11}-V_{22}}{2}\cos(\omega_1 t) & V_{12} - \frac{\omega_1}{2} + i\frac{V_{11}-V_{22}}{2}\sin(\omega_1 t) \\ V_{12} - \frac{\omega_1}{2} - i\frac{V_{11}-V_{22}}{2}\sin(\omega_1 t) & -\frac{V_{11}-V_{22}}{2}\cos(\omega_1 t) \end{pmatrix}.$$

The cyclic solutions of (1) with account of (20) are written as

$$|\Psi_\nu(r,t) = \begin{pmatrix} \cos(\omega_1 t/2)\exp(-i\bar{\mathcal{E}}_1^\nu t) & -i\sin(\omega_1 t/2)\exp(-i\bar{\mathcal{E}}_2^\nu t) \\ -i\sin(\omega_1 t/2)\exp(-i\bar{\mathcal{E}}_1^\nu t) & \cos(\omega_1 t/2)\exp(-i\bar{\mathcal{E}}_2^\nu t) \end{pmatrix} |\Phi_\nu(r) > . \tag{33}$$

Now we show that for all the considered examples with $h_i(t) = \omega_i t$ the expectation values of $H(t)$ for corresponding cyclic solutions are independent on time. Really, the relation (22), with account of expressions for potentials and cyclic solutions (26)–(33), gives the expectation values of $H(t)$ as

$$\epsilon_\nu^i =< \Phi_\nu(r)|\widetilde{H}(r)|\Phi_\nu(r) > + \frac{\omega_i}{2} < \Phi_\nu(r)|\hat{\sigma}_i|\Phi_\nu(r) >= \widetilde{\mathcal{E}}_\nu + \frac{\omega_i}{2}\bar{\hat{\sigma}}_\nu^i. \tag{34}$$

It means that equation (1), for obtained family of time-dependent potential matrices, possesses solutions and properties like time-independent ones. Note, the spin-expectation value

$$\bar{\hat{\sigma}}_\nu^i =< \Psi_\nu(t,r)|\hat{\sigma}_i|\Psi_\nu(t,r) >=< \Phi_\nu(r)|\hat{\sigma}_i|\Phi_\nu(r) >$$

is none other than spin-alignment along the rotating axis and does not depend on time in this case. For the dynamical phase δ_ν from (34) we get

$$\delta_\nu^i = \int_0^T \epsilon_\nu^i(t)dt = \widetilde{\mathcal{E}}_\nu T + \pi\bar{\hat{\sigma}}_\nu^i. \tag{35}$$

Removing the dynamical phase (35) from the total phase

$$\beta_\nu = \pi + \widetilde{\mathcal{E}}_\nu T. \tag{36}$$

we get the geometric phase φ_ν^i

$$\varphi_\nu^i = (\beta_\nu - \delta_\nu^i) = \pi(1 - \bar{\hat{\sigma}}_\nu^i). \tag{37}$$

One can easily see that the geometric phase is determined by the spin-expectation value $\bar{\hat{\sigma}}_\nu^i$ along the rotating axis. It is evident, in all considered cases the Hamiltonian (26), (29), (32) is T-periodic, $T = 2\pi/\omega$, and the solutions (27), (30) and (33) are are $2T$-periodic and change sign after one period. As one can easily seen, if $\mathcal{S}(t)$ is chosen to be T-periodic then the Hamiltonian becomes $T/2$-periodic, and the solutions are T-periodic. In this case, the total phase is multiply of 2π, and the geometric phase is determined by (25).

Thus, by means of a different choice of transformation operators $\mathcal{S}_i(t), i = 1,2,3$, three families of time-dependent potential matrices with the corresponding cyclic solutions are

generated in a closed form from one family of time-independent potential matrices. It is evident that a more general transformation can be taken and can be used to change time-independent Hamiltonian $\widetilde{H}$ into time-dependent $H(t)$. For example, $\mathcal{S}(t)$ can be taken as a direct product of the $\mathcal{S}_i(t)$, $\mathcal{S}(t) = \prod_{i=1}^{3} \mathcal{S}_i(t)$, or with a more complicated dependence of $\mathcal{S}$ on time. So, whenever $V(r)$ is an exactly soluble potential matrix for the ordinary time-independent system of Schrödinger equations, a family of exactly soluble time-dependent potential matrices for the system of Schrödinger equations (1) can be generated.

Let $V(r)$ be a Bargmann potential matrix for which the system of equations (8) has exact solutions. For example, the matrix elements of the transparent potential and the relevant Jost solutions are [10]

$$V_{\alpha\alpha'}(r) = 2\frac{d}{dr}\sum_{\nu\lambda}\exp(-\kappa_\alpha^\nu r)\gamma_\alpha^\nu P_{\nu\lambda}^{-1}(r)\gamma_{\alpha'}^\lambda \exp(-\kappa_{\alpha'}^\lambda r), \tag{38}$$

$$\begin{aligned} F_{\alpha\alpha'}^{\pm}(k,r) &= \exp(\pm ik_\alpha r)\delta_{\alpha\alpha'} \\ &- \sum_{\nu\lambda}\gamma_\alpha^\nu \exp(-\kappa_\alpha^\nu r)P_{\nu\lambda}^{-1}(r)\int_r^\infty \gamma_{\alpha'}^\lambda \exp(-(\kappa_{\alpha'}^\lambda \pm ik_{\alpha'})r')dr', \end{aligned} \tag{39}$$

where

$$P_{\nu\lambda}(r) = \delta_{\nu\lambda} + \sum_{\alpha'}^{m}\frac{\gamma_{\alpha'}^\nu\gamma_{\alpha'}^\lambda}{\kappa_{\alpha'}^\nu + \kappa_{\alpha'}^\lambda}\exp(-(\kappa_{\alpha'}^\nu + \kappa_{\alpha'}^\lambda)r).$$

The normalized solution for the vector function of the bound state is immediately expressed through the matrix Jost solutions taken at the energy of the bound state, $k = i\kappa_\nu$, $(i\kappa_\nu)^2 = -\widetilde{\mathcal{E}}_\nu$, $k_\alpha = (\widetilde{\mathcal{E}} - \Delta_\alpha)^{1/2}$,

$$\Phi_\alpha(\widetilde{\mathcal{E}}_\nu, r) = \sum_{\alpha'}^{m} F_{\alpha\alpha'}(k = i\kappa^\nu, r)\gamma_{\alpha'}^\nu, \quad \alpha, \alpha' = 1, 2. \tag{40}$$

Here, the indices ν and λ correspond to the bound states characterized by the energies $\widetilde{\mathcal{E}}_\nu$ and amplitudes $|\gamma^\nu>$ determine normalizing matrices $|\gamma_\alpha^\nu >< \gamma_{\alpha'}^\nu|$ and the channel indices are labeled as α, α'. Substitution (38) into (26), (29), (32) and (40) into (27), (30), (33) allows determination of three families of potential matrices and pertinent cyclic solutions in an closed analytical form. Let us take another exactly solvable example, the potential matrix and corresponding regular solutions within the radial Gelfand-Levitan approach, when the spectral matrix for $\widetilde{H}(r)$ coincides with one for free Hamiltonian at $\mathcal{E} > 0$, can be written as

$$V_{\alpha\alpha'}(r) = 2\frac{d}{dr}\sum_{\nu\lambda}\sinh[\kappa_\alpha^\nu r]/\kappa_\alpha^\nu)c_{\alpha'}^\nu P_{\nu\lambda}^{-1}(r)c_{\alpha'}^\lambda(\sinh[\kappa_{\alpha'}^\lambda r]/\kappa_{\alpha'}^\lambda, \tag{41}$$

$$\phi_{\alpha\alpha'}(k,r) = \frac{\sin[k_\alpha r]}{k_\alpha}\delta_{\alpha\alpha'} - \sum_{\nu\lambda}c_\alpha^\nu\frac{\sinh[\kappa_\alpha^\nu r]}{\kappa_\alpha^\nu}P_{\nu\lambda}^{-1}(r)c_{\alpha'}^\lambda\int_0^r\frac{\sinh[\kappa_{\alpha'}^\lambda r]}{\kappa_{\alpha'}^\lambda}\frac{\sin[k_{\alpha'}r']}{k_{\alpha'}}dr', \tag{42}$$

where c^{ν}_{α} determine normalizing matrices $C^{\nu} = |c^{\nu}_{\alpha} >< c^{\nu}_{\alpha}|$ and

$$P_{\nu\lambda}(r) = \delta_{\nu\lambda} + \sum_{\alpha'}^{m} c^{\nu}_{\alpha'} c^{\lambda}_{\alpha'} \int_0^r (\sinh[\kappa^{\nu}_{\alpha'} r]/\kappa^{\nu}_{\alpha'})(\sinh[\kappa^{\lambda}_{j'} r']/\kappa^{\lambda}_{\alpha'}) dr'.$$

The normalized solution for the vector function of the bound state is given as

$$\Phi_{\alpha}(\widetilde{\mathcal{E}}_{\nu}, r) = \sum_{\alpha'}^{m} \phi_{\alpha\alpha'}(k = i\kappa^{\nu}, r) c^{\nu}_{\alpha'}. \tag{43}$$

Substituting (41) into (26), (29), (32) and (43) into (27), (30), (33) we readily obtain another three families potentials and corresponding periodic solutions in an explicit form. It is worth noting that the class of exactly solvable stationary problems is very wide. Algebraic Bargmann-Darboux transformations give multiple examples of problems that can be used to generate a large class of explicit time–dependent potential matrices admitting exact solutions of the Schrödinger equation (1). We consider 2×2 matrix equations as examples. In principle, this procedure can be used for the generation of an exactly soluble potential matrix for systems of N coupled time-dependent equations.

3 Conclusion

We have developed the method of solving a system of coupled time-dependent Schrödinger equations in an explicit form by using unitary time-dependent transformations which change time-independent problems into time-dependent ones. In distinction to earlier studies, the method is based on soluble time-independent equations and a special choice of initial functions. This approach yields a large class of previously unknown exactly solvable nonstationary models and can be used for investigation of two- and three–dimensional objects. The family of time-dependent soluble potential matrices, for which a system of two-coupled Schrödinger equations has exact solutions, is generated. As an application of the suggested method, we have presented explicit expressions of the expectation value of $H(t)$, the nonadiabatic geometric phase, total, and dynamical phases, which are derived in terms of obtained cyclic solutions. Time-dependent models, for which expectation values don't dependent on time, are given.

Acknowledgments

The author is grateful for useful discussions and hospitality to Professor H.V. Von Geramb. The paper was supported in part by the BRFFI, grant No.F99R-162/641.

References

[1] H.R. Levis, *J. Math. Phys.*, 1962, vol. 3, p. 806; H.R. Levis and W.B. Riesenfeld, *J. Math. Phys.*, 1969, vol. 10, p. 1458.

[2] I.A. Malkin and V.I., Man'ko *Dynamical Symmetries and Coherent States of Quantum Systems* (Moscow, Nauka, 1979)

[3] C.F., Lo, *Phys.Rev.A* 1992, vol. 45, p. 5262.

[4] M. Maamache, *Phys. Rev.A*, 1995, vol. 52, p. 936.

[5] K. Bay, W. Lay, A. Akopyan, *J. Phys.A: Math Gen.*, 1997, vol. 30, p. 3057.

[6] Wang, Shun-Jin, *Phys. Rev.A*, 1990, vol. 42, p. 5107; ibid p. 5103.

[7] C.J. Efthimiou and D. Spector, *Phys. Rev.A*, 1994, vol. 49, p. 2301.

[8] B.F. Samsonov, *J.Phys.A:Math.Gen.*, 2000, vol.33, p.591.

[9] K. Chadan and P.C. Sabatier, *Inverse Problems in Quantum Scattering Theory* (Springer-Verlag. Berlin Heidelberg/New York, 1989, 493p. 2-nd ed.)

[10] B.N. Zakhariev and A.A. Suzko, *Direct and Inverse Problems. (Potentials in Quantum Scattering)* (Springer-Verlag. Berlin Heidelberg/New York, 1990, 223p. 2-nd ed.)

[11] G. Junker, *Supersymmetric Method in Quantum and Statistical Physics*, (Springer-Verlag. Berlin Heidelberg/New York, 1996).

[12] W.Paul and M.Raether Z.Phys., 1955, vol.140, p.262.

[13] A.V. Gaponov and M.A. Miller J.Eksper.Teor.Fiz., 1958, vol.43, p. 242.

[14] I.E. Tralle, Phys.Stat.Sol. 1994, vol.181, p.97.

[15] R. Cook et.al., Phys.Rev., 1985, vol. A31, p.564.

[16] E. Velicheva, A. Suzko, *Theor. and Math.Physics* 1998, vol.115, p.687.

Analytic Modeling for Investigation of Quantum Systems in the Adiabatic Representation

A.A. Suzko, E.P. Velicheva

Joint Institute for Nuclear Research,
Laboratory of Information Technology,
JINR, Dubna, Russia
and Radiation Physics and Chemistry Problems Institute,
National Academy of Sciences of Belarus, Minsk
e-mail: suzko@jinr.ru

Complex dynamical systems with some degrees of freedom are analyzed on the basis of the inverse scattering problem in the adiabatic representation. Matrix elements of an exchange interaction determining the "slow" subsystem are calculated and studied in terms of exact solutions obtained within the parametric inverse problem for the "fast" subsystem. The main characteristics of the slow subsystem Hamiltonian depend on the statement of the parametric inverse problem: namely, the fast subsystem is given on the semi-axis or on the entire axis. It is shown that in the case of the radial parametric problem or parametric problem specified on the semi-axis, matrix elements of the induced vector and scalar potentials and eigenfunctions are singular at the degeneration points of two levels. The opposite occurs in the case of the parametric problem specified on the entire axis, they are regular at the degeneration points of two levels. The influence of parametric spectral characteristics of the fast subsystem on the behavior of the slow subsystem is studied. In particular, it is shown that the choice of normalizing functions determining the transparent symmetric potential in fast variables leads to the zero coupling between states for two-level systems, (even at the point of degeneration), while another choice of normalization factors leads to nonsymmetric potentials and nonzero coupling for the same energy levels. Our approach suggested allows the investigation of adiabatically driven quantum systems with a prescribed dependence on parametric adiabatic variables.

1 Introduction

Consider the system evolving according to the Schrödinger equation

$$i\hbar\frac{\partial|\Psi(t)>}{\partial t} = H(x(t))|\Psi(t)>, \tag{1}$$

where the Hamiltonian $H(x(t))$ is given in the form

$$H(x(t)) = -\partial^2/\partial x^2 + h(x(t)), \quad h(x(t)) = -\partial^2/\partial y^2 + V(x(t), y). \tag{2}$$

If $\psi_n(x(t);y)$ are solutions to the equation

$$h(x(t))|\psi_n(x(t);y)>=\mathcal{E}_n(x(t))|\psi_n(x(t);y)> \tag{3}$$

and form a complete orthonormal set $|\psi_n(x;y)><\psi_n(x;y')| = \delta(y-y'), <\psi_n(x;y)\mid \psi_m(x;y)>=\delta_{nm}\ \forall x$, with elements depending on $x=x(t)$ parametrically. Then Ψ can be expressed by the expansion

$$|\Psi(x(t),y)>=\sum_n |n><n|\Psi>=\sum_n \int \psi_n(x(t);y)F_n(x(t)) \tag{4}$$

over the eigenstates of the self-adjoint parametric Hamiltonian (3). Upon substituting the expansion (4) into the initial Schrödinger equation (1) and using the relations of orthonormalization, we arrive at a multichannel system of gauge equations

$$i\hbar\frac{\partial}{\partial t}F(x(t)) = [-(\nabla_x + A(x))^2 + V(x)]F(x(t)). \tag{5}$$

Here $A(x)$ and $V(x)$ are the effective vector and scalar potentials, respectively, the matrix elements of which are induced by the basis functions $\psi(x;y)$ of the parametric problem (3)

$$A_{nm}(x) =<\psi_n(x;y)|\nabla_x|\psi_m(x;y)>, \tag{6}$$

$$\begin{aligned} V_{nm}(x) &= <\psi_n(x;y)|h(x)|\psi_m(x;y)> - i\hbar <\psi_n(x;y)|\partial_t\psi_m(x;y)> \\ &\equiv \mathcal{E}_n(x)\delta_{nm} - i\hbar B_{nm}(x). \end{aligned} \tag{7}$$

In the case of a slow and smooth evolution in time of the collective coordinates $x(t)$, the second derivative is neglected in (2), $H(x(t)) \to h(x(t))$, and the solution of equation (1) is sought in the form of an expansion (see, for instance, [1],

$$|\Psi(x(t),y)>=\sum_n c_n(x(t))\exp\Big(-\frac{i}{\hbar}\int_0^t \mathcal{E}_n(x(t'))dt'\Big)|\psi_n(x(t);y)> . \tag{8}$$

With account of (8) and (3) in (1), we find that the system of equations for $c_n(t)$ can be written in the form

$$\partial_t c_n(t) = \sum_m B_{nm}(x(t))\exp\Big[-\frac{i}{\hbar}\int_0^t (\mathcal{E}_n(t') - \mathcal{E}_m(t'))dt'\Big]c_m(t) . \tag{9}$$

The matrix elements of the exchange interaction $B_{nm}(x(t))$ are generated by the basis functions $|n>$ of the "instantaneous" Hamiltonian (3)

$$B_{nm}(x(t)) =<n|\dot{m}>= A_{nm}(x(t))\cdot\dot{x}(t), \tag{10}$$

where the dot denotes the time derivative. Thus, the initial problem in the adiabatic approach is reduced to the consistent solution of two problems: a parametric one (3) and a multichannel one for the system of gauge-type equations (5) or (9). Here, we assume

that $H(x(t))$ is real, limited, and continuous in x. Because, for each x, the eigenfunctions are real valued and orthonormal then the nonadiabatic couplings $A_{nm} = -A_{mn}$ in (6) are real and antisymmetric in n and m. The matrix elements of (6) of the induced connection A can be computed in terms of the analytic eigenfunctions of parametric equations (3) for the given functional dependence of scattering data $\{S(x,k), M^2(x), \mathcal{E}(x)\}$ on the slow coordinate variable $x(t)$.

In accordance with the general definition of the inverse scattering problem [2] and [3], the parametric inverse problem [4] consist of the reconstruction of the potential and corresponding solutions from the known scattering data $\{S(x,k), M^2(x), \mathcal{E}(x)\}$ (in the Marchenko approach) or the spectral data $\{\rho(x,k), N^2(x), \mathcal{E}(x)\}$ (Gelfand-Levitan approach) parametrically depending on the coordinate variable x. This dependence reflects the peculiarity of the nonstandard parametric inverse problem. Specifying this dependence and employing the algebraic methods of the inverse scattering problem, we present a wide class of potentials for which one can construct exactly solvable models and, consequently, derive solutions in a closed analytic form. These generalized Bargmann potentials are defined by the rational Jost functions $f \equiv f_+$,

$$f(x;k) = \overset{\circ}{f}(k) \prod \frac{k - i\alpha(x)}{k + i\beta(x)} \tag{11}$$

parametrically depending on the "slow" dynamical variables x through the dependence of spectral parameters on these variables. This situation is, to a certain extent, analogous to the theory of nonlinear evolution equations. The parametric Jost function (11) has N curves $k = -i\beta_j(x)$, $j = 1, 2, ...N$ of simple poles and N curves of simple zeros $k = i\alpha_j(x)$ defined as functions of the parametric variable x.

For real potentials, the curves $i\alpha_j(x)$ and $-i\beta_j(x)$ must be situated symmetrically with respect to the imaginary axis in the complex k plane. In $\alpha(x)$, there are not only zeros on the imaginary semi-axis corresponding to the bound states $Re\ \kappa_j(x) = 0$, $Im\ \kappa_j(x) > 0$ for each value of x, but, also, zeros in the lower k half-plane with $Im\ \nu_j(x) < 0$ (the number of simple pole curves of $\beta_j(x)$ equals the total number of $\kappa_j(x)$ and $\nu_j(x)$). In this case, the scattering matrix and the spectral function assume the form

$$S(x;k) = \overset{\circ}{S}(k) \prod \frac{(k + i\alpha(x))(k + i\beta(x))}{(k - i\beta(x))(k - i\alpha(x))}, \quad \rho(x;k) = \overset{\circ}{\rho}(k) \prod \frac{(k - i\beta(x))(k + i\beta(x))}{(k + i\alpha(x))(k - i\alpha(x))}. \tag{12}$$

For such $S(x;k)$ and $\rho(x;k)$, the kernels of the integral equations of the parametric inverse problem can be represented as the sums of terms with a factorized dependence on the fast variable y: $Q(x;y,y') = \sum_i^N B_i(x;y)B_i(x;y')$. When the kernel Q is inserted into the base parametric equation of the inverse problem,

$$K(x;y,y') + Q(x;y,y') + \int_{y(0)}^{\infty(y)} K(x;y,y'')Q(x;y'',y')dy' = 0, \tag{13}$$

it is evident that the kernel of the generalized shift $K(x;y,y')$ also becomes degenerate: $K(x;y,y') = \sum_i^N K_i(x;y)B_i(x;y')$. As a consequence, the system of integral equations of the inverse problem is reduced to a system of algebraic equations. Then, the spherically

nonsymmetric potential and solutions corresponding to it can be expressed in a closed analytic form in terms of the known solutions and spectral characteristics by using the generalized equations of the parametric inverse problem,

$$V(x,y) = \overset{\circ}{V}(y) \mp 2\frac{d}{dy}K(x;y,y), \tag{14}$$

$$\phi(x;k,y) = \overset{\circ}{\phi}(k,y) + \int_{y(0)}^{\infty(y)} K(x;y,y')\,\overset{\circ}{\phi}(k,y')dy'. \tag{15}$$

Integration limits in (13), (15), and the signs in (14) depend on the particular approach to the inverse problem. The limits from y to ∞ and the minus sign correspond to the Marchenko approach. Limits $[0,y]$ and plus sign represent the Gelfand–Levitan approach.

Within the generalized Marchenko approach [5], the integral kernels $Q(x;y,y')$, dependent on x as a parameter

$$Q(x;y,y') = \frac{1}{2\pi}\int_{-\infty}^{\infty}[\overset{\circ}{S}(k) - S(x;k)]\,\overset{\circ}{f}(k,y)\,\overset{\circ}{f}(k,y')dk \tag{16}$$

$$+\sum_{n}^{m} M_n^2(x)\,\overset{\circ}{f}(i\kappa_n(x),y)\,\overset{\circ}{f}(i\kappa_n(x),y') - \sum_{n}^{\overset{\circ}{m}} \overset{\circ}{M}{}_n^2\,\overset{\circ}{f}(i\,\overset{\circ}{\kappa}_n,y)\,\overset{\circ}{f}(i\,\overset{\circ}{\kappa}_n,y'),$$

are constructed by using two sets of the scattering data. These are the set $\{S(x;k)$, $\mathcal{E}_n(x)$, $M_n^2(x)\}$, corresponding to equation (2) for every value of parameter x, and the ordinary scattering data $\{\overset{\circ}{S}(k), \overset{\circ}{\mathcal{E}}_n, \overset{\circ}{M}{}_n^2\}$, corresponding to (2) with $V(x;y) = 0$ and $\overset{\circ}{V}(y) \neq 0$. The functions $\overset{\circ}{f}(k,y)$ are standard Jost solutions with the known potential $\overset{\circ}{V}(y)$. Potentials (14) and Jost solutions (15) are determined from $K(x;y,y')$, with respect to which, the linear integral equation (13) is solved for every fixed x.

2 Exactly solvable models within the parametric inverse problem on the semi-axis

For the parametric inverse problem, radial or on a semi-axis, when $\overset{\circ}{V}(y) = 0$ and, correspondingly, $\overset{\circ}{S}(k) = 1$, the kernel of the basic integral equation (13) in the Marchenko approach,

$$Q(x;(y+y')) = \frac{1}{2\pi}\int_{-\infty}^{\infty}[1 - S(x;k)]\exp[ik(y+y')]\,dk \tag{17}$$

$$+ \sum_{n}^{N} M_n^2(x)\exp[-\kappa_n(x)(y+y')],$$

with the scattering matrix (12), can be rewritten as

$$Q(x;(y+y')) = -i\sum_{n}^{N} Res\ \mathcal{S}(k=i\beta_n(x))\exp[-\beta_n(x)(y+y')] \tag{18}$$

$$+\sum_{n}^{N}\{-iRes\ \mathcal{S}(k=i\kappa_n(x))\exp[-\kappa_n(x)(y+y')] + M_n^2(x)\exp[-\kappa_n(x)(y+y')]\}\ .$$

Following the procedure of constructing phase-equivalent potentials suggested in [6] for the one-dimensional problem and in [4] for the parametric problem, we shall reduce solution of the inverse problem to the successive applications of two procedures. First, one can cancel out the second summation in the right-hand side of (18) if the normalization functions $M_n^2(x) = \overset{\circ}{M}{}_n^2(x)$ are chosen to be equal to $i\ Res\ \mathcal{S}(k)$ at $k = i\kappa_n(x)$,

$$\overset{\circ}{M}{}_n^2(x) = i\ Res\ \mathcal{S}(k)|_{k=i\kappa_n(x)} \tag{19}$$

$$= -\frac{2\kappa_n(x)(\kappa_n(x)+\beta_n(x))}{(\kappa_n(x)-\beta_n(x))}\prod_{n'\neq n}^{N}\frac{(\kappa_n(x)+\beta_{n'}(x))(\kappa_n(x)+\kappa_{n'}(x))}{(\kappa_n(x)-\beta_{n'}(x))(\kappa_n(x)-\kappa_{n'}(x))}.$$

As a result, we obtain a simpler expression for the kernel $Q = \overset{\circ}{Q}$,

$$\overset{\circ}{Q}(x;y,y') = \sum_{n}^{N} A_n(x)\exp[-\beta_n(x)(y+y')], \tag{20}$$

where

$$A_n(x) = \frac{2\beta_n(x)(\beta_n(x)+\kappa_n(x))}{(\beta_n(x)-\kappa_n(x))}\prod_{n'\neq n}^{N}\frac{(\beta_n(x)+\kappa_{n'}(x))(\beta_n(x)+\beta_{n'}(x))}{(\beta_n(x)-\beta_{n'}(x))(\beta_n(x)-\kappa_{n'}(x))}. \tag{21}$$

Inserting the kernel $\overset{\circ}{Q}(x;y,y')$ (20) into the parametric Marchenko equation (13), we obtain

$$\overset{\circ}{K}(x;y,y') + \sum_{n}^{N} A_n(x)\Big\{e^{-\beta_n(x)y} + \int_{y}^{\infty}\overset{\circ}{K}(x;y,y'')e^{-\beta_n(x)y''}dy''\Big\}e^{-\beta_n(x)y'} = 0,$$

where the expression in braces is the Jost solution $\overset{\circ}{f}(k = i\beta_n(x), y)$ for the sought potential $\overset{\circ}{V}(x;y)$. Such that $\overset{\circ}{K}(x;y,y')$ has a form similar to that of $\overset{\circ}{Q}(x;y,y')$ from (20) with a separabilized dependence on y and y' and with a parametric dependence on x,

$$\overset{\circ}{K}(x;y,y') = -\sum_{n}^{N} A_n(x)\ \overset{\circ}{f}(i\beta_n(x), y)\exp[-\beta_n(x)y']. \tag{22}$$

Substituting this kernel of the generalized shift from the free wave to the Jost solution into the triangular integral equation

$$\overset{\circ}{f}(x;k,y) = \exp(iky) + \int_y^\infty \overset{\circ}{K}(x;y,y')\exp(iky')dy' \tag{23}$$

we get, at $k = i\beta_n(x)$, a set of equations for $\overset{\circ}{f}(i\beta_n(x),y)$,

$$\overset{\circ}{f}(i\beta_n(x),y) = \sum_{n'}^{N} \exp[-(\beta_{n'}(x)y]P^{-1}_{n'n}(x;y),$$

where $P_{nn'}(x;y)$ is defined as follows:

$$P_{nn'}(x;y) = \delta_{nn'} + A_n(x)\frac{\exp[-(\beta_n(x)+\beta_{n'}(x))y]}{\beta_n(x)+\beta_{n'}(x)}.$$

Then, by substituting (22) into the parametric equations of the inverse problem (13) - (15), we obtain

$$\overset{\circ}{V}(x;y) = -2\frac{d^2}{dy^2}\ln\det||P(x;y)||, \tag{24}$$

$$\overset{\circ}{f}_{\pm}(x;k,y) = \exp(\pm iky) + \sum_{nn'}^{N} A_n(x)P^{-1}_{nn'}(x;y)\frac{\exp[-(\beta_n(x)+\beta_{n'}(x)\mp ik)y]}{(\beta_n(x)\mp ik)}. \tag{25}$$

A similar situation is obtained in the Gelfand–Levitan approach, only, there, the normalizing functions $N_n^2(x)$ are expressed through $Res\,[f_+(x;k)f_-(x;k)]^{-1}$ at the points $k = i\kappa_n(x)$:

$$N_n^2(x) = 4i\kappa_n^2(x)[(df_+(x;k)/dk)|_{k=i\kappa_n(x)}f_-(i\kappa_n(x))]^{-1}.$$

The corresponding algebraic formulae for the one-dimensional Bargmann potentials and their solutions [6] can be obtained directly if we set $\kappa_n(x) \equiv \kappa_n$ and $\beta_n(x) \equiv \beta_n$.

At the second stage, by using the corresponding $\overset{\circ}{f}(x;k,y)$ as the initial solutions, we obtain a family of potentials and solutions for arbitrary normalizing functions $M_n^2(x)$ that do not obey the condition (19): $\overset{\circ}{M}{}_n^2(x) < M_n^2(x)$. Since the scattering $\mathcal{S}(x;k)$-function is independent of the choice of the normalization functions $M_n^2(x)$, we have $\mathcal{S}(x;k) = \overset{\circ}{\mathcal{S}}(x;k)$. As a result, the integral term in a generalized expression such as (16) for $Q(x;y,y')$ vanishes. Since, on the other hand, both $V(x;y)$ and $\overset{\circ}{V}(x;y)$ possess the same potential curves (curves of bound states) $\mathcal{E}_n(x) = \overset{\circ}{\mathcal{E}}_n(x)$ but different normalization factors $M_n^2(x)$ and $\overset{\circ}{M}{}_n^2(x)$, respectively, we find that

$$Q(x;y,y') = \sum_{n}^{N}(M_n^2(x) - \overset{\circ}{M}{}_n^2(x))\,\overset{\circ}{f}(i\kappa_n(x),y)\,\overset{\circ}{f}(i\kappa_n(x),y'). \tag{26}$$

And similarly, the kernel of the generalized shift $K(x;y,y')$ is written as

$$K(x;y,y') = -\sum_{n}^{N}(M_n^2(x) - \overset{\circ}{M}{}_n^2(x))f(i\kappa_n(x),y)\overset{\circ}{f}(i\kappa_n(x),y'). \tag{27}$$

Inserting $K(x;y,y')$ and $Q(x;y,y')$ into the basic parametric Marchenko equations (13)–(15), we derive the following relations for the potential and Jost solutions:

$$V(x;y) = \overset{\circ}{V}(x;y) + 2\frac{d^2}{dy^2}\ln\det||P(x;y)||\,, \tag{28}$$

$$f_{\pm}(x;k,y) = \overset{\circ}{f}_{\pm}(x;k,y) -$$

$$-\sum_{nm}^{N}(M_n^2(x) - \overset{\circ}{M}{}_n^2(x))\overset{\circ}{f}(i\kappa_n(x),y)P_{nm}^{-1}(x;y)\int_y^{\infty}\overset{\circ}{f}(i\kappa_m(x),y')\overset{\circ}{f}_{\pm}(k,y')dy'.$$

The explicit dependence on the fast variables is defined by the Jost solutions (25) determined at $k = i\kappa_n(x)$, i.e., on the level-energy curves depending on the parametric variable x. Here we employed the notation

$$P_{nm}(x;y) = \delta_{nm} + (M_n^2(x) - \overset{\circ}{M}{}_n^2(x))\int_y^{\infty}\overset{\circ}{f}(i\kappa_n(x),y')\overset{\circ}{f}(i\kappa_m(x),y')dy'.$$

Since $S(x;k)$, corresponding to the two-dimensional potentials $V(x;y)$ (28), is independent of the normalizations $M_n^2(x)$, $S(x;k) = \overset{\circ}{S}(x;k)$, the formula (28) represents a parametric family of potentials depending on N parametric functions $M_n^2(x)$. A change of the normalization functions $M_n^2(x)$ of the parametric eigenstates leads to a change in the potentials, the Jost, regular, and basis solutions of the parametric Hamiltonian, and the matrix elements of the exchange interaction, and, therefore, strongly influences the behavior of the dynamical quantum systems.

2.1 Bargmann potentials with two potential curves

Let us now present the case with two potential curves in the problem on the semi-axis. The Jost function (11) can be written in the form

$$f(x;k) = \frac{(k - i\kappa_1(x))(k - i\kappa_2(x))}{(k + i\beta_1(x))(k + i\beta_2(x))};\quad Im\,\{\kappa_j(x)\} = Im\,\{\beta_j(x)\} = 0,\quad j = 1,2 \tag{29}$$

and

$$S(x;k) = \frac{(k + i\kappa_1(x))(k + i\beta_1(x))}{(k - i\beta_1(x))(k - i\kappa_1(x))}\frac{(k + i\kappa_2(x))(k + i\beta_2(x))}{(k - i\beta_2(x))(k - i\kappa_2(x))}. \tag{30}$$

Two pole curves correspond to the zero curves $k = i\kappa_j(x)$ of the parametric Jost function $f_+(x;k))$, the other pole curves correspond to the pole trajectories $k = i\beta_j(x)$ of the parametric Jost function $f_-(x;k)$.

We have, necessarily, $\beta_j(x) > 0$ (for the Jost function $f_+(x;k)$ to be analytic in the upper half-plane k for all x). When $\kappa_j(x) > 0$, we have the bound state curves (the potential curves of bound states $\mathcal{E}_j(x) = -\kappa_j^2(x)$); when $\kappa_j(x) < 0$, we do not have any bound states. In principle, the functions $\alpha(x)$ in (11), (12) can change from negative values $\alpha(x) = -\nu(x)$ (antibound state trajectory) to positive values $\alpha(x) = \kappa(x)$. If $\alpha_j(x) = -\nu_j(x) < 0 \ \ \forall x$, the potential is not deep and wide enough to produce bound states and the potential corresponds to $\mathcal{S}(x,k)$ with pole curves at $k = i\beta_j(x)$. Here we assume that $\kappa_j(x) > 0$, to provide for the existence of two bound state curves. The ordering of the potential curves $\mathcal{E}_i < \mathcal{E}_{i+1}$ is assumed. To simplify the investigation of the problem, we take the normalizations of the bound state wave functions in the form (19)

$$\begin{aligned}
\overset{\circ}{M}{}_1^2(x) &= -\frac{2\kappa_1(x)(\kappa_1(x)+\beta_1(x))}{(\kappa_1(x)-\beta_1(x))}\frac{(\kappa_1(x)+\beta_2(x))(\kappa_1(x)+\kappa_2(x))}{(\kappa_1(x)-\beta_2(x))(\kappa_1(x)-\kappa_2(x))}, \\
\overset{\circ}{M}{}_2^2(x) &= -\frac{2\kappa_2(x)(\kappa_2(x)+\beta_2(x))}{(\kappa_2(x)-\beta_2(x))}\frac{(\kappa_2(x)+\beta_1(x))(\kappa_2(x)+\kappa_1(x))}{(\kappa_2(x)-\beta_1(x))(\kappa_2(x)-\kappa_1(x))}.
\end{aligned} \tag{31}$$

Thus, the potential $V(x,y)$ is determined only by the spectral data $\kappa_j(x)$ and $\beta_j(x)$, $j = 1,2$, and corresponds to one of the family of potentials characterized by the same energy levels $\mathcal{E}_j(x)$ and the same parametric $\mathcal{S}(x;k)$ (30) with four pole curves. From the relations (24), (25), we obtain the two–dimensional potential $V(x,y)$ and the corresponding normalized wave functions $\psi_j(x,y) = \overset{\circ}{M}_j(x)\ \overset{\circ}{f}(i\kappa_j(x),y), j = 1,2$ (Fig.1b,c) of the self-energy curves $\mathcal{E}_{1,2}(x)$. The matrix element $A_{12}(x) = \int \psi_1(x;y)\partial_x\psi_2(x;y)dy$ of the nonadiabatic connection (6) is computed in terms of the analytic functions $\psi_{1,2}(x;y)$.

From the normalizations $M_j^2(x)$ being positive definite, the conditions $\beta_2(x) \geq \kappa_2(x)$ and $\beta_1(x) \geq \kappa_1(x) \geq \beta_2(x)$ follow. This means that $\mathcal{E}_2(x) \geq \beta_2^2(x) \geq \mathcal{E}_1(x)$. If the levels $\mathcal{E}_1(x)$ and $\mathcal{E}_2(x)$ move towards each other, one or both of them would be equal to $\beta_2^2(x)$ at any point of $x = x'$. It can be easily seen from the relations (31) that the corresponding normalizing function $M_1^2(x)$ or $M_2^2(x)$ becomes singular when $\kappa_1(x') = \beta_2(x')$ or $\kappa_2(x') = \beta_2(x')$. If $\kappa_1(x') = \beta_2(x') = \kappa_2(x')$, both normalization functions $M_1^2(x)$ and $M_2^2(x)$ have double poles at the point of the degeneracy $x = x'$. It can be seen from the relations (21) and (24) that the potential $V(x;y)$ with the pertinent normalized functions $\psi_{1,2}(x,y)$ has a double pole at this point $x = x'$ somewhere on the positive y-axis, but it has no physical meaning from the point of standard scattering theory. One can see from (19) that the same investigations with singular behaviors of normalizing functions are valid for potentials with an arbitrary number of levels when two levels move closer together. Remind of the investigations of normalizations in [7] at avoided level crossings.

3 Transparent Potentials

The one-dimensional inverse problem on the entire axis $-\infty < y < \infty$ with the zero-th reflection coefficient, $\mathcal{S}^{ref}(k) = 0$, describes transparent (reflectionless) potentials along the variable y. If the reflection function $\mathcal{S}^{ref}(x;k)$ is chosen to be equal to zero at all energies and at all values of the parametric variable x, then the integral in the relation for $Q(x;y,y')$ vanishes and only the sum over the bound states remains. The transmission

coefficient $\mathcal{S}^{tr}$, with an absolute value equal to unity, is a rational function,

$$\mathcal{S}^{tr}(x;k) = \prod \frac{k + i\kappa(x)}{k - i\kappa(x)}, \tag{32}$$

depending on the dynamical parametric variable $x = x(t)$.

The relations for the potentials and solutions can be expressed in terms of normalized eigenfunctions and represented in a most symmetric and convenient form. Following [8], introduce the function

$$\lambda_n(x;y) = \gamma_n(x)\exp(-\kappa_n(x)y).$$

Then the formula for $K(x;y,y')$ can be written as

$$K(x;y,y') = -\sum_n^N \gamma_n(x)\psi_n(x;y)\exp(-\kappa_n(x)y') = -\sum_n^N \psi_n(x;y)\lambda_n(x;y'). \tag{33}$$

For the normalized eigenfunctions $\psi_n(x;y)$ from (15), we obtain

$$\psi_n(x;y) = \sum_j^N \lambda_j(x;y)A^{-1}_{jn}(x;y), \tag{34}$$

with the matrix $A_{jn}(x;y)$ given by

$$A_{jn}(x;y) = \delta_{jn} + \frac{\lambda_j(x;y)\lambda_n(x;y)}{\kappa_n(x) + \kappa_j(x)}. \tag{35}$$

Finally, the kernel $K(x;y,y')$ and the potential can be represented as

$$\begin{aligned} K(x;y,y') &= -\sum_n^N \sum_j^N \lambda_j(x;y)A^{-1}_{jn}(x;y)\lambda_n(x;y'), \\ V(x;y) &= -4\sum_n^m \kappa_n(x)\psi_n^2(x;y). \end{aligned} \tag{36}$$

Recall that these relations are obtained for the specific case of the zero reflection function $\mathcal{S}^{ref}(x;k) = 0 \ \ \forall x$.

Exactly solvable models with time-dependent symmetric in y potentials. Note that the symmetric transparent potentials for each fixed value of x and the pertinent wave functions are completely defined by the energy levels, since in this case the normalizations of the bound state functions are expressed as

$$\gamma_n^2(x) = 2\kappa_n(x)\prod_{m\neq n}\left|\frac{\kappa_m(x) + \kappa_n(x)}{\kappa_m(x) - \kappa_n(x)}\right|. \tag{37}$$

Consider the simple example of two-dimensional exactly solvable models for two-level systems with symmetric in y potentials. Taking the equation (34) for the normalized

eigenfunctions $\psi_1(x;y)$ and $\psi_2(x;y)$ and carrying out some simplifications, we obtain

$$\psi_1(x;y) = \frac{\sqrt{2\kappa_1(x)b(x)}\cosh(\kappa_1(x)y)}{\cosh[(\kappa_1(x)+\kappa_2(x))y] + b(x)\cosh[(\kappa_1(x)-\kappa_2(x))y]},$$
$$\psi_2(x;y) = \frac{\sqrt{2\kappa_2(x)b(x)}\sinh(\kappa_2(x)y)}{\cosh[(\kappa_1(x)+\kappa_2(x))y] + b(x)\cosh[(\kappa_1(x)-\kappa_2(x))y]} \tag{38}$$

with $b(x) = |(\kappa_1(x)+\kappa_2(x))/(\kappa_1(x)-\kappa_2(x))|$. Taking an account of (38) for $V(x;y)$ in (36), we get the potential in an explicit form convenient for analysis:

$$V(x;y) == -8b(x)\frac{\kappa_1^2(x)\cosh^2(\kappa_1(x)y) + \kappa_2^2(x)\sinh^2(\kappa_2(x)y)}{\{\cosh[(\kappa_1(x)+\kappa_2(x))y] + b(x)\cosh[(\kappa_1(x)-\kappa_2(x))y]\}^2}. \tag{39}$$

Obviously, the potential $V(x;y)$ is symmetric in y for each value of the parametric variable x. It is also easily seen from (38) that the eigenfunction of the ground state $\psi_1(x;y)$ is symmetric and $\psi_2(x;y)$ is antisymmetric in y for each fixed value of x, as it is required for the problem on the entire axis $-\infty < y < \infty$ with a potential $V(x;y)$ symmetric in y for each fixed x.

As a limiting case, we can consider $\kappa_2(x) \to \kappa_1(x)$. It can be seen from (39) and (38) that if $\kappa_2(x) = \kappa_1(x)$ at any point of $x = x'$, the limiting values of the potential and the wave functions are equal to zero, $V(x';y) = \psi_1(x';y) = \psi_2(x';y) = 0 \ \ \forall y$, while in the above considered problem on the half axis $0 \le y < \infty$, the potential, together with its eigenfunctions, are singular at $x = x'$.

Matrix elements of the exchange interaction $A_{12}(x)$, induced by basis functions of the parametric instantaneous Hamiltonian, can be written in the form

$$A_{12}(x) = 2b(x)\sqrt{\kappa_1(x)\kappa_2(x)}\Big[\int_{-\infty}^{\infty}\frac{y(\partial\kappa_2(x)/\partial_x)\cosh(\kappa_1(x)y)\cosh(\kappa_2(x)y)}{G^2(x;y)},$$
$$- \frac{\cosh(\kappa_1(x)y)\sinh(\kappa_2(x)y)(\partial G(x;y)/\partial_x)}{G^3(x;y)}dy\Big] \tag{40}$$

with $G(x;y) = \cosh[(\kappa_1(x)+\kappa_2(x))y] + b(x)\cosh[(\kappa_1(x)-\kappa_2(x))y]$. Since $G(x;y)$ and $\partial_x G(x;y)$ are even functions in y, it can be easily shown that this integral vanishes, since both integrands are odd functions of y, i.e., $A_{12}(x) = 0 \ \forall x$. This means that there is no coupling between the eigenstates $\psi_1(x;y)$ and $\psi_2(x;y)$ for any moment t and any point of the adiabatic variable x for transparent symmetric in y potentials. The transition amplitude between the bound state functions $\psi_1(x;y)$ and $\psi_2(x;y)$ is equal to zero because it is defined by the zero matrix elements $A_{12}(x) = 0 \ \ \forall x$.

The choice of normalizing functions $\gamma_n^2(x)$ of the energy-level states, which do not obey the condition (37), leads to loss of symmetry in y of potentials, and to another behavior of the parametric basis functions. As a result, the matrix elements of the exchange interaction $A12(x) \neq 0$, that is the coupling between neighboring states takes place, $A_{1,2}(x) \neq 0$, and the transitions take place, as well.

As a consequence of our analysis, we can conclude that the choice of normalization functions of the parametric Hamiltonian eigenstates strongly influences the behavior of the dynamical quantum systems.

4 Conclusions

The method presented permits one to construct a wide class of potentials and corresponding solutions of the parametric equation (3) in a closed analytical form and, after that, to calculate the matrix elements of the exchange interaction. The first procedure is an algebraic one, but the second step is, in general, numerical. Therefore, the method is semi-analytical. This approach allows one to investigate the influence of the parametric spectral data on the behavior of the potentials, of the basis functions of the parametric Hamiltonian, and of the matrix elements of the exchange interaction. It was shown that the main features of the exchange interaction determining the slow subsystem Hamiltonian essentially depend on the character of the parametric (fast) Hamiltonian: namely, the fast subsystem Hamiltonian is given on the semi–axis $0 \leq y < \infty$ or on the entire axis $-\infty < y < \infty$. As a consequence, the problems of level crossing are different in both cases. The matrix elements $A_{nm}(x)$ have no singularities at the degeneracy points of two-levels if the consideration is made within the parametric problem on the entire line and $A_{nm}(x)$ are singular at the degeneracy points if the consideration is made within the parametric problem on the half-axis. It was shown that in the case of the parametric problem on the entire axis the potentials, the basis functions and the matrix elements of the coupling are not singular at the degeneracy points of the two states, while in the parametric problem on the half-axis, the potential, together with its eigenfunctions and matrix elements of the exchange interaction, are singular at these points. We studied the effect of the normalizing functions on the properties of dynamical systems. In particular, we have found that in the parametric problem on the entire axis for a special choice of the normalization functions (37), the exchange interaction between the bound states for two-level systems are equal to zero for all values of the adiabatic variables, even at the point of the degeneracy. In fact, we can trace the behavior of the matrix elements of the exchange interaction (6) at any moment of time. We can recommend our approach for the investigation of the Landau–Zener transitions and level crossing problems.

Acknowledgments

The authors are grateful to Professor H. von Geramb for valuable discussions. The paper was supported in part by the BRFFI, grant No.F99R-162/641.

References

[1] J.T. Hwang, J. T. and Ph. Pechukas, *J. Chem. Phys.*, 1977, vol. 67, p. 4640.

[2] L.M. Gelfand and B.M. Levitan, *On the determination of a differential equation by its spectral function,* Sov.Izv.AN USSR, ser.Math., 1951, vol.15, No.4, 309-360.

[3] V.M. Marchenko *Sturm-Liouville Operators and their Applications* (Naukova Dumka, Kiev) 1977;
Z.S. Agranovich and V.A. Marchenko *The inverse Problem of Scattering Theory* (Gordon and Breach, New York) 1963.

[4] A.A. Suzko, *"Multidimensional and three-particle scattering problems in the adiabatic representation"*, Phys. Part. Nucl. 1993, vol.24, 485-516; Proc. Int.Conf. *" Quantum Inversion Theory and Applications", in Lecture Notes in Physics*, ed. H.V.von Geramb Vol. 427, (Springer-Verlag, Heidelberg, 1993), 67-106.

[5] H.E. Moses, J.Math.Phys., 1979, vol.20, 2047-2053.
V.N. Pivovarchik V.N and A.A. Suzko, In Proc. Heat-and Mass Transfer Inst. AN BSSR "Evolutional Problems of Energy Transfer in Inhomogeneous Mediums" Minsk, (1982) 168-177.

[6] B.N. Zakhariev, A.A. Suzko, *Potentials and Quantum Scattering: Direct and Inverse Problems* (Springer-Verlag, New York, Berlin, Heidelberg) 1990.

[7] E.A. Solov'ev, Usp. Fiz. Nauk. 1989, vol.157, 437.

[8] E.P. Velicheva and A.A. Suzko, *"Exactly Solvable Models and Dynamical Quantum Systems"*, *Teor. Math. Phys.* 1998, vol. 115, No.1, p.106-131.

Application of the Noncommutative Gröbner Bases Method for Proving Geometrical Statements in Coordinate-free Form

Irina J. Tchoupaeva

Moscow, MSU, mech-math
e-mail: ladyirina@shade.msu.ru

In this paper we consider the application of the noncommutative Gröbner bases method for proving theorems in algebraic geometry. Geometrical statements of constructive type should be given in the coordinate-free form.

1 Coordinate-free representation of points and statements

We consider theorems of elementary geometry (two-dimensional and three-dimensional). Let $A_1, A_2, A_3, \ldots, A_n$ be points in a finite-dimensional space. We treat these points as vectors drawn from the origin 0. Then, geometrically, the outer product of two vectors A and B is a bivector corresponding to the parallelogram obtained by sweeping the vector A along the vector B. The parallelogram obtained by sweeping B along A differs from the parallelogram obtained by sweeping A along B only in the orientation.

Consider the Grassman algebra generated by points $A_1, A_2, A_3, \ldots, A_n$, i.e., the free algebra with an external product $A \wedge B$, which is associative and anticommutative: $A \wedge B = -B \wedge A$.

Consider a finite-dimensional space and task-space embedded in this space. For example, in the case of a two-dimensional task we consider a plane in the enveloping space.

2 Grassman algebra

It is known that the Grassman algebra is an associative free algebra with a finite set of relations corresponding to the anticommutativity of multiplication on the generators:

$$Gr = \langle A_1, \ldots, A_n \| A_i \wedge A_j = -A_j \wedge A_i \forall i, j \in \{1, \ldots, n\} \rangle .$$

The conditions of anticommutativity $A_i \wedge A_j = -A_j \wedge A_i \forall i, j \in \{1 \ldots n\}$ on the generators allow us to permute the neighboring factors in any product $A_{i_1} \wedge \cdots \wedge A_{i_k}$. As a result, the product changes the sign.

So, any of these products in the Grassman algebra is equal up to the sign either to zero or to a product of generators with strictly increasing indices. It follows that any element of the Grassman algebra can be represented as a linear combination of such products

$$X = \sum_{(i_1,\ldots,i_k),\quad i_1<i_2<\cdots<i_k} \alpha^{(i_1,\ldots,i_k)} A_{i_1} \wedge \cdots \wedge A_{i_k}.$$

Then, the dimension of the Grassman algebra is equal to 2^n, where n is the number of the generators. Thus, all ideals of this algebra are finite dimensional and have finite Gröbner bases.

3 Statements of the constructive type

In Chou's collection of examples [2] of two-dimensional geometrical tasks, there are a number of statements of the constructive type, which can be written in the coordinate-free form.

The following assertions can be written as some relations in the Grassman algebra:

1) three points A_1, A_2, A_3 are collinear iff $(A_1 - A_2) \wedge (A_1 - A_3) = 0$;

2) the lines A_1A_2 and A_3A_4 are parallel iff $(A_1 - A_2) \wedge (A_3 - A_4) = 0$;

3) a point A_3 divides an interval $[A_1; A_2]$ in the ratio $n : m$ iff $m(A_3 - A_1) = n(A_2 - A_3) = 0$.

Moreover, some additional relations can be added to this list.

1) two points A_1 and A_2 are equal iff $A_1 - A_2 = 0$;

2) two bivectors $A_1 \wedge A_2$ and $B_1 \wedge B_2$ are collinear iff $\alpha A_1 \wedge A_2 = \beta B_1 \wedge B_2$;

3) a point P lies on a plane $\{A_1, A_2, A_3\}$ iff $(A_1 - P) \wedge (A_2 - P) \wedge (A_3 - P) = 0$,

etc.

4 Noncommutative Gröbner bases

Let us consider an associative noncommutative free algebra with unit 1 over the field F with generators $a_1, \ldots, a_k$. Each element of this algebra can be represented in the form $\sum_{(j_1,\ldots,j_l)} k^{(j_1,\ldots,j_l)} a_{j_1} \ldots a_{j_l}$. Introducing an order on the generators and an admissible order on the monomials, we can define the leading monomials $\mathrm{lm}(u) = k_u^{(j_1,\ldots,j_{l_u})} a_{j_1} \ldots a_{j_{l_u}}$ and $\mathrm{lm}(v) = k_v^{(i_1,\ldots,i_{l_v})} a_{i_1} \ldots a_{i_{l_v}}$ for any polynomials $u = \sum_{(j_1,\ldots,j_{l_u})} k_u^{(j_1,\ldots,j_{l_u})} a_{j_1} \ldots a_{j_{l_u}}$ and $v = \sum_{(i_1,\ldots,i_{l_v})} k_v^{(i_1,\ldots,i_{l_v})} a_{i_1} \ldots a_{i_{l_v}}$.

Since we consider a free algebra over a field, we can normalize these polynomials so that the leading coefficients become equal to unit. So, we assume that $\mathrm{lm}(u) = a_{j_1} \ldots a_{j_{l_u}}$ and $\mathrm{lm}(v) = a_{i_1} \ldots a_{i_{l_v}}$.

Now, we can consider all *compositions* of two monomials $\mathrm{lm}(u)$ and $\mathrm{lm}(v)$.

Two monomials have a composition $f(u,v)$, iff the end of the first monomial is equal to the beginning of the second one, namely, in our case there exists an integer $m > 0$ and a set of indices $p_1, \ldots, p_m$ such that $\mathrm{lm}(u) = a_{j_1} \ldots a_{j_{i_u - m}} a_{p_1} \ldots a_{p_m}$ and $\mathrm{lm}(v) = a_{p_1} \ldots a_{p_m} a_{i_{m+1}} \ldots a_{i_{i_v}}$.

Then, the composition is equal to

$$f(u,v) = a_{j_1} \ldots a_{j_{i_u - m}} a_{p_1} \ldots a_{p_m} a_{i_{m+1}} \ldots a_{i_{i_v}} = \mathrm{lm}(u) v_1 = u_1 \mathrm{lm}(v),$$

where $v_1 = a_{i_{m+1}} \ldots a_{i_{i_v}}$ and $u_1 = a_{j_1} \ldots a_{j_{i_u - m}}$.

Since any monomial can be represented as a finite noncommutative product of the generators, there exist at most a finite set of compositions for each pair of monomials. Having obtained all compositions of leading monomials of two polynomials, one can write a finite number of noncommutative S-polynomials, which can be constructed as $S(u,v) = u_1 v - u v_1$.

A monomial $x = a_{s_1} \ldots a_{s_n}$ is divisible by a monomial $y = a_{t_1} \ldots a_{t_m}$ iff the monomial y is the substring of the monomial x so that $x = y_{left} y y_{right}$.

A polynomial p_1 with the unit leading coefficient is reducible with respect to a polynomial p_2 with the unit leading coefficient iff the leading monomial $\mathrm{lm}(p_1)$ is divided by the leading monomial $\mathrm{lm}(p_2)$ so that $\mathrm{lm}(p_1) = \alpha \mathrm{lm}(p_2) \beta$, where α and β are some monomials. The result of reduction is the polynomial $p_1' = p_1 - \alpha p_2 \beta$.

Noncommutative Gröbner bases of an ideal I are determined by analogy with the commutative case, as a complete system of relations, which generate this ideal.

The Buchberger algorithm is the same, however, the definitions of division, reduction and S-polynomial are changed.

5 Example of a theorem

Example 1. (Gauss' line).
Let A_1, A_2, B_1, B_2 be arbitrary points. Construct the complete quadrilateral $A_1 A_2 B_1 B_2$ and diagonals $A_1 A_2$, $B_1 B_2$, $A_1 B_2$ and $B_1 A_2$. Let $A_1 A_2$ intersect $B_1 B_2$ at A_3, $A_1 B_2$ intersect $B_1 A_2$ at B_3. Let M_1 be the midpoint of $A_1 B_1$, M_2 be the midpoint of $A_2 B_2$ and M_3 be the midpoint of $A_3 B_3$. Then, the points M_1, M_2 and M_3 lie on one straight line.

Now, we can formulate the following statements of the constructive type for this theorem:

1) $col(A_1, A_2, A_3)$: $(A_1 - A_2) \wedge (A_1 - A_3) = 0$

2) $col(B_1, B_2, A_3)$: $(B_1 - B_2) \wedge (B_1 - A_3) = 0$

3) $col(B_1, A_2, B_3)$: $(B_1 - A_2) \wedge (B_1 - B_3) = 0$

4) $col(A_1, B_2, B_3)$: $(A_1 - B_2) \wedge (A_1 - B_3) = 0$

5) $midp(A_1, B_1, M_1)$: $(M_1 - A_1) = (B_1 - M_1) = 0$

6) $midp(A_2, B_2, M_2)$: $(M_2 - A_2) = (B_2 - M_2) = 0$

7) $midp(A_1, B_1, M_1)$: $(M_3 - A_3) = (B_3 - M_3) = 0$

The conclusion is the following: $col(M_1, M_2, M_3)$: $(M_1 - M_2) \wedge (M_1 - M_3) = 0$.

6 Description of Gröbner Bases method for the proof of the theorems which are true universally (commutative and anticommutative case)

Some geometrical theorems can be formulated in these terms. To prove these theorems, the theory of noncommutative Gröbner bases can be applied. The system of polynomials corresponding to the hypotheses of the theorem and anticommutativity relations for the generators of the Grassman algebra are considered as generators of an ideal in a free associative algebra. The assertion of the theorem (written as a polynomial in this algebra) is valid if it belongs to this ideal. This is equivalent to zero reducibility of this polynomial.

Implementation of the algorithm for obtaining noncommutative Gröbner bases in the ring of noncommutative polynomials with integer coefficients are being developed.

In the commutative case, we have the same idea, but for the description of hypotheses and conclusions we use the equations for the coordinates of the points.

7 Calculation of Gröbner Bases on a systems with identities. Equivalence of identities on a whole set and relations on bases elements

In the general case, a polynomial identity is not equivalent to a system of equation on generators. However, in our case, the anticommutativity of homogeneous linear polynomials is equivalent to anticommutativity relations on the generators. There is a finite number of such relations.

For example, the property $A \wedge B = -B \wedge A$ for all homogeneous linear polynomials $A, B \in Gr$ is equivalent to the finite set of relations on the generators $A_i \wedge A_j = -A_j \wedge A_i \forall i, j \in \{1, \ldots, n\}$.

The conditions of anticommutativity can also be written as $A \wedge A = 0$ for all homogeneous linear polynomials A from the Grassman algebra. However, in this case the finite system of relations on the generators $A_i \wedge A_i = 0, \quad i \in \{1, \ldots, n\}$ is not equivalent to the previous statement.

8 Description of the implemented algorithm

The algorithm for computing the anticommutative Gröbner bases with integer coefficients has been implemented.

The main algorithm in the program is the following:

1) Input the number of the variables and hypotheses.

2) Input the name of the variables in the increasing order.

3) Input the hypotheses and the conclusion.

4) Convert all the statements into the internal format.

5) Add the conditions of anticommutativity to the system.

6) Calculate the noncommutative Gröbner basis of the system.

7) Calculate the normal form of the conclusion of the theorem with respect to the Gröbner basis.

8) If the result is equal to zero, then the theorem is true universally, otherwise the theorem is not true universally.

The kernel of the program processes the general case of noncommutative Gröbner bases; however, the current interface is oriented to proving a specific class of geometrical theorems which can be formulated in terms of statements (1)–(3) in the Grassman algebra.

In CAS Maple V, functions which are able to make similar calculations are not revealed.

The program is written in Php 4.0 and uses possibilities of the web interface. Php 4.0 is a platform-free programming language. On the one hand, there is a possibility to include HTML-code into the texts of the programs for simple testing; on the other hand, this language enables us to use its resources as an object-oriented language.

9 An example of calculation of anticommutative Gröbner bases by the program

As an example, consider the operation of the program on the theorem on the Gauss line formulated above.

Determine the noncommutative Gröbner basis and the normal form of the conclusion of the theorem for the reverse lexicographical order on the monomials under the condition $A_1 < A_2 < A_3 < B_1 < B_2 < B_3 < M_1 < M_2 < M_3$.

The hypotheses of the theorem are

$$
\begin{aligned}
P[0]: &\quad 1 * A_2 \wedge A_3 - 1 * A_1 \wedge A_3 - 1 * A_2 \wedge A_1 + 1 * A_1 \wedge A_1 \\
P[1]: &\quad 1 * B_2 \wedge B_1 - 1 * B_1 \wedge B_1 - 1 * B_2 \wedge A_3 + 1 * B_1 \wedge A_3 \\
P[2]: &\quad 1 * B_2 \wedge B_3 - 1 * A_1 \wedge B_3 - 1 * B_2 \wedge A_1 + 1 * A_1 \wedge A_3 \\
P[3]: &\quad 1 * B_1 \wedge B_3 - 1 * A_2 \wedge B_3 - 1 * B_1 \wedge B_1 + 1 * A_2 \wedge B_1 \\
P[4]: &\quad 2 * M_1 - 1 * B_1 - 1 * A_1 \\
P[5]: &\quad 2 * M_2 - 1 * B_2 - 1 * A_2 \\
P[6]: &\quad 2 * M_3 - 1 * B_3 - 1 * A_3
\end{aligned}
$$

The conclusion of the theorem is

$$1 * M_1 \wedge M_1 - 1 * M_1 \wedge M_3 - 1 * M_2 \wedge M_1 + 1 * M_2 \wedge M_3$$

After adding the relations of anticommutativity on the generators of the Grassman algebra to the hypotheses of the theorem, the system of polynomials becomes

$$
\begin{aligned}
P[0]: &\quad 1*A_2\wedge A_3 - 1*A_1\wedge A_3 - 1*A_2\wedge A_1 + 1*A_1\wedge A_1\\
P[1]: &\quad 1*B_2\wedge B_1 - 1*B_1\wedge B_1 - 1*B_2\wedge A_3 + 1*B_1\wedge A_3\\
P[2]: &\quad 1*B_2\wedge B_3 - 1*A_1\wedge B_3 - 1*B_2\wedge A_1 + 1*A_1\wedge A_3\\
P[3]: &\quad 1*B_1\wedge B_3 - 1*A_2\wedge B_3 - 1*B_1\wedge B_1 + 1*A_2\wedge B_1\\
P[4]: &\quad 2*M_1 - 1*B_1 - 1*A_1\\
P[5]: &\quad 2*M_2 - 1*B_2 - 1*A_2\\
P[6]: &\quad 2*M_3 - 1*B_3 - 1*A_3\\
P[7]: &\quad 1*A_1\wedge A_1\\
P[8]: &\quad 1*A_1\wedge A_2 + 1*A_2\wedge A_1\\
P[9]: &\quad 1*A_1\wedge A_3 + 1*A_3\wedge A_1\\
P[10]: &\quad 1*A_1\wedge B_1 + 1*B_1\wedge A_1\\
P[11]: &\quad 1*A_1\wedge B_2 + 1*B_2\wedge A_1\\
P[12]: &\quad 1*A_1\wedge B_3 + 1*B_3\wedge A_1\\
P[13]: &\quad 1*A_1\wedge M_1 + 1*M_1\wedge A_1\\
P[14]: &\quad 1*A_1\wedge M_2 + 1*M_2\wedge A_1\\
P[15]: &\quad 1*A_1\wedge M_3 + 1*M_3\wedge A_1\\
P[16]: &\quad 1*A_2\wedge A_2\\
P[17]: &\quad 1*A_2\wedge A_3 + 1*A_3\wedge A_2\\
P[18]: &\quad 1*A_2\wedge B_1 + 1*B_1\wedge A_2\\
P[19]: &\quad 1*A_2\wedge B_2 + 1*B_2\wedge A_2\\
P[20]: &\quad 1*A_2\wedge B_3 + 1*B_3\wedge A_2\\
P[21]: &\quad 1*A_2\wedge M_1 + 1*M_1\wedge A_2\\
P[22]: &\quad 1*A_2\wedge M_2 + 1*M_2\wedge A_2\\
P[23]: &\quad 1*A_2\wedge M_3 + 1*M_3\wedge A_2\\
P[24]: &\quad 1*A_3\wedge A_3\\
P[25]: &\quad 1*A_3\wedge B_1 + 1*B_1\wedge A_3\\
P[26]: &\quad 1*A_3\wedge B_2 + 1*B_2\wedge A_3\\
P[27]: &\quad 1*A_3\wedge B_3 + 1*B_3\wedge A_3\\
P[28]: &\quad 1*A_3\wedge M_1 + 1*M_1\wedge A_3\\
P[29]: &\quad 1*A_3\wedge M_2 + 1*M_2\wedge A_3\\
P[30]: &\quad 1*A_3\wedge M_3 + 1*M_3\wedge A_3\\
P[31]: &\quad 1*B_1\wedge B_1\\
P[32]: &\quad 1*B_1\wedge B_2 + 1*B_2\wedge B_1\\
P[33]: &\quad 1*B_1\wedge B_3 + 1*B_3\wedge B_1
\end{aligned}
$$

$$
\begin{aligned}
P[34]: &\quad 1*B_1\wedge M_1+1*M_1\wedge B_1\\
P[35]: &\quad 1*B_1\wedge M_2+1*M_2\wedge B_1\\
P[36]: &\quad 1*B_1\wedge M_3+1*M_3\wedge B_1\\
P[37]: &\quad 1*B_2\wedge B_2\\
P[38]: &\quad 1*B_2\wedge B_3+1*B_3\wedge B_2\\
P[39]: &\quad 1*B_2\wedge M_1+1*M_1\wedge B_2\\
P[40]: &\quad 1*B_2\wedge M_2+1*M_2\wedge B_2\\
P[41]: &\quad 1*B_2\wedge M_3+1*M_3\wedge B_2\\
P[42]: &\quad 1*B_3\wedge B_3\\
P[43]: &\quad 1*B_3\wedge M_1+1*M_1\wedge B_3\\
P[44]: &\quad 1*B_3\wedge M_2+1*M_2\wedge B_3\\
P[45]: &\quad 1*B_3\wedge M_3+1*M_3\wedge B_3\\
P[46]: &\quad 1*M_1\wedge M_1\\
P[47]: &\quad 1*M_1\wedge M_2+1*M_2\wedge M_1\\
P[48]: &\quad 1*M_1\wedge M_3+1*M_3\wedge M_1\\
P[49]: &\quad 1*M_2\wedge M_2\\
P[50]: &\quad 1*M_2\wedge M_3+1*M_3\wedge M_2\\
P[51]: &\quad 1*M_3\wedge M_3
\end{aligned}
$$

The noncommutative Gröbner basis of the ideal generated by relations $P[0]-P[51]$ is

$$
\begin{aligned}
P[0]: &\quad 1*A_3\wedge A_2-1*A_3\wedge A_1+1*A_2\wedge A_1\\
P[1]: &\quad 1*B_2\wedge B_1-1*B_2\wedge A_3+1*B_1\wedge A_3\\
P[2]: &\quad 1*B_3\wedge B_2-1*B_3\wedge A_1+1*B_2\wedge A_1\\
P[3]: &\quad 1*B_3\wedge B_1-1*B_3\wedge A_2-1*B_1\wedge A_2\\
P[4]: &\quad 2*M_1-1*B_1-1*A_1\\
P[5]: &\quad 2*M_2-1*B_2-1*A_2\\
P[6]: &\quad 2*M_3-1*B_3-1*A_3\\
P[7]: &\quad 1*A_1\wedge A_1\\
P[8]: &\quad 1*A_1\wedge A_2+1*A_2\wedge A_1\\
P[9]: &\quad 1*A_1\wedge A_3+1*A_3\wedge A_1\\
P[10]: &\quad 1*A_1\wedge B_1+1*B_1\wedge A_1\\
P[11]: &\quad 1*A_1\wedge B_2+1*B_2\wedge A_1\\
P[12]: &\quad 1*A_1\wedge B_3+1*B_3\wedge A_1\\
P[13]: &\quad 1*A_2\wedge A_2\\
P[14]: &\quad 1*A_2\wedge A_3+1*A_3\wedge A_2\\
P[15]: &\quad 1*A_2\wedge B_1+1*B_1\wedge A_2
\end{aligned}
$$

$$
\begin{aligned}
P[16]: &\quad 1 * A_2 \wedge B_2 + 1 * B_2 \wedge A_2 \\
P[17]: &\quad 1 * A_2 \wedge B_3 + 1 * B_3 \wedge A_2 \\
P[18]: &\quad 1 * A_3 \wedge A_3 \\
P[19]: &\quad 1 * A_3 \wedge B_1 + 1 * B_1 \wedge A_3 \\
P[20]: &\quad 1 * A_3 \wedge B_2 + 1 * B_2 \wedge A_3 \\
P[21]: &\quad 1 * A_3 \wedge B_3 + 1 * B_3 \wedge A_3 \\
P[22]: &\quad 1 * B_1 \wedge B_1 \\
P[23]: &\quad 1 * B_1 \wedge B_2 + 1 * B_2 \wedge B_1 \\
P[24]: &\quad 1 * B_1 \wedge B_3 + 1 * B_3 \wedge B_1 \\
P[25]: &\quad 1 * B_2 \wedge B_2 \\
P[26]: &\quad 1 * B_2 \wedge B_3 + 1 * B_3 \wedge B_2 \\
P[27]: &\quad 1 * B_3 \wedge B_3
\end{aligned}
$$

The normal form of the conclusion of the theorem with respect to the Gröbner basis is equal to zero.

Thus, the theorem is true universally.

In the paper [1] Wang considers the same theorem as an example of the use of the coordinate-free technique for automatic proving of theorems. Considering the same order on the variables and monomials as a hypotheses of the theorem, he takes the same relations $P[0] - P[6]$ of the first system, but from his paper it is not clear which system of relations he considers as relations responsible for the anticommutativity of multiplication.

10 Equations describing the dimension of the task space

It seems that we have to add equations describing the dimension of the task (whether the points are on the same line in the one-dimensional case, or whether the points are on the same plane in the two-dimensional case, etc) to our systems. It is related to the fact that we consider the enveloping space of the task.

Consider the condition that all points of the task lie on the same plane.

If the number n of points is equal to 1, 2 or 3, then all these points are on the same plane, and we need no additional relations.

If the number of points is equal to $n \geq 4$, then the condition that the points belong to the same plane is equivalent to the condition that any four points are on the same plane.

The relation $(A_1 - A_0) \wedge (A_2 - A_0) \wedge (A_3 - A_0) = 0$ formally means that the vectors $(A_3 - A_0)$, $(A_2 - A_0)$ and $(A_1 - A_0)$ are linearly dependent. This condition can be expressed by the formula $0 = (A_1 - A_0) \wedge (A_2 - A_0) \wedge (A_3 - A_0) = A_1 \wedge A_2 \wedge A_3 - A_0 \wedge A_2 \wedge A_3 + A_0 \wedge A_1 \wedge A_3 - A_0 \wedge A_1 \wedge A_2$. Here we take A_0 as a marked point. Actually, as the marked point, we can take any one of these four points.

So, if the number of points $n \geq 4$, the condition all these points belong to one and the same plane if and only if the system of C_n^4 relations $(A_{i_1} - A_{i_0}) \wedge (A_{i_2} - A_{i_0}) \wedge (A_{i_3} - A_{i_0}) = 0$ hold, where $i_0, i_1, i_2, i_3 \in \{1, \ldots, n\}$ are any distinct four points.

For example, in the theorem about the Gauss line, we can add the condition that all nine points are on the same plane to the first system. The numbers of the additional relations such as $(X_1 - X_0) \wedge (X_2 - X_0) \wedge (X_3 - X_0) = 0$, where $X_0, X_1, X_2, X_3 \in \{A_1, A_2, A_3, B_1, B_2, B_3, M_1, M_2, M_3\}$ is equal to $C_9^4 = \frac{9!}{4!5!} = 126$. All these relations can be reduced to zero with respect to the Gröbner basis. This means that these polynomials lie in the ideal, generated by hypotheses of the theorem and the anticommutativity relations. Thus, this condition follows from the hypotheses of the theorem and the anticommutativity relations. Using our program, we can verify this fact automatically.

If the polynomials, describing the dimension of the task are not reducible to zero, but the conclusion of the theorem is reducible to zero, then our task is a particular case of another task of a higher dimension.

In the general case, we should consider the condition that all points of the task belong to an m -dimensional space embedded in the enveloping space of a higher dimension. So, we have to consider $C_n^{(m+2)}$ polynomials such as $(A_{i_1} - A_{i_0}) \wedge \cdots \wedge (A_{i_{m+1}} - A_{i_0}) = 0$, where A_{i_0} is one of generators of the Grassman algebra and $i_0, i_1, \ldots, i_{m+1} \in \{1, \ldots, n\}$ is an arbitrary set of $m+2$ distinct indices.

11 Advantages of the coordinate-free method

a) If the dimension of the task is m, then max degree of all hypotheses and conclusions will be less than or equal to m

b) if the conditions of the task are satisfied, then an equation, whose degree is higher than m, cannot be presented in the Gröbner basis (but this is possible in the coordinate case).

Acknowledgments

I am grateful to my advisor E.V. Pankratiev for assistance while preparing this paper and for many valuable remarks.

References

[1] D. Wang, Gröbner Bases Applied to Geometric Theorem Proving and Discovering, *Gröbner Bases and Applications, Eds. B. Buchberger and F. Winkler*, Cambridge Univ. Press, 1998, pp. 281–302.

[2] S.C. Chou, Mechanical Geometry Theorem Proving *Mathematics and Its Applications*, D. Reidel, Dordrecht, 1987.

Program Package Intended for *Internet*-Based Access to Computer Algebra Resources

S.I. Tertychniy

VNIIFTRI, Mendeleevo,
e-mail: bpt97@mail.ru

General structure and major features of the package GRGnetwork intended for remote access to software based on the universal computer algebra system Reduce *making use of Internet communication media are discussed.*

Introduction

Objectives

The work discussed below constitutes a part of the long-term programme intended for the developing of program computational tools incorporating the methods of Computer Algebra (CA) and the modern communication technologies based on the Internet infrastructure. In the most general setting, the corresponding software is intended to render assistance in the solving of certain classes of problems in theoretical physics — first of all, the theory of gravity — with the help of modern computational tools. It is designed to provide Internet-based access to the consolidated computational resources of Computer Algebra Systems (CAS), universal or/and specialized, in conjunction with a number of additional services enhancing the efficiency and convenience of this form of their usage.

Assuming more narrow frameworks, we outline here the basic points and main features of the package *GRGnetwork*, constituting one of the components of the software intended to address the problem declared above but being also of a notable interest in its own right.

Before the discussing the very content of the work an outline of its starting points has to be given.

Approach

The approach implemented on *GRGnetwork* is based on the client-server method and makes use of the object-oriented (OO) programming technique. The latter point implies the wide application of the platform independent OO Java technology, playing here the role of the universal environment and basic programming tool.

Advantages *versus* shortcomings

The choice indicated above is *a priori* distinguished by some potential advantages apparently enabling one to efficiently settle a number of subsidiary problems which constitute in fact the core of the problem to be solved. These include in particular the following noteworthy points:

- Inheriting specific feature of the Java technology, the software developed on its base automatically possesses the important property of independence on the computer platform (often called *portability* property).

- The incorporation of a *Java Virtual Machine* (JVM) in majority of the most popular and widely used WWW-browsers releases a potential user of Java-based software of concern about the providing and installing any specific background software. This circumstance makes access to Java-based software fairly straightforward for everybody interested in its capabilities. In particular, this concerns non-experienced beginners as well, that is evidently the point of significant importance.

- The constituent of Java environment called *Abstract Window Toolkit* (AWT) enables one to realize a *portable graphical multi-window user/system interface* (GUI) on a uniform, platform independent base.

- The facility of *Java applets* enables one to achieve a properly *secure* implementation of the *remote access* to common integrated computational resources through the Internet communication media.

This list of attractive features of Java-based technique of program development should seem sufficient for some substantiating of the choice adopted. However, at the same time, it might be perhaps argued that all the matter above looks rather as a fine theory and cannot really play role of practical guidelines.

Indeed, dealing closer with the issue of a brute reality, the approach outlined turns out be also associated, simultaneously with evident advantages, with a number of drawbacks, many of the latter being, in a sense, close companions of the former, though. Pursuing to maintain a reasonable balance of pluses and minuses in these speculations, the following remarks somewhat contravening with the ones just made are now in order:

- The existing realizations of Java environment and their implementations into WWW-browsers prove not be free of some apparently buggy behavior or specific improper peculiarities. This is currently true, to a great extent, for all the widely practiced software of this class, essentially. Additionally, there are still no corresponding sufficiently detailed documented or, say, practical standards which one could refer to in the developing of Java-based software on a stable ground. Most of the 'pathologies' specific Java realizations reveal, either apparent or actual, can only be captured by means of laborious testing, making use of the notorious 'trial and error method'. Moreover, in these studies, some vague deviations from the *available standards* are observed, unfortunately. Such violations of a standard usually prove computer platform and Java version dependent. Hence they claim in each concrete case a specific

treatment, sometimes fairly laborious, that does not promote the final software quality. Sometimes, in spite of all the efforts of a programmer, Java-based software may not manifest all the capabilities promised by the relevant Java specifications.

- The realization of graphical interface tools in Java' graphical engine AWT is currently difficult to qualify advantageous in many respects. In particular, it does not support some graphical facilities which are, in principle, ensured by graphics engines of the operating systems (OS) of interest. Additionally, in practice, the execution of some dynamical operations seems to be realized in a way which is far of perfectness and manifests some apparently improper visual effects. As a result, in certain cases, it turns out problematic for AWT-based interfaces to compete with the system-dependent GUI realizations.

It may be supposed that such sort shortcomings are partially connected with the portability property lying in the base of the Java technology.

Nevertheless, the above does not mean, of course, that Java-based software cannot turn out an efficient and useful tool. Besides, fortunately, it can be noted that some observations give evidence that the general situation concerning the shortcomings mentioned above tends to gradually improve.

Basic ingredients of the package background

Let us now characterize the basic ingredients of the software complex *GRGnetwork* is based upon. These include in particular the following elements:

Graphical interface tools

The source of constructive elements used for the development of *GRGnetwork* involves the collection of platform independent graphical interface tools implemented in the form of Java's *Abstract Window Toolkit* (AWT). It provides a variety of graphical primitives allowing to construct a platform-independent friendly graphical user interface (GUI).

Basic networking tools

A substantial advantage of the Java concept is the integrating into Java environment all the necessary network communication capabilities. Making use of this feature, the communication between the 'client' and 'server' components of *GRGnetwork* is realized on the base of TCP/IP protocol and employs Internet communication channels, the local networking based on TCP/IP (*intranets*) being supported on the identical ground. The two sub-components of the 'server' component of the package (called *GRGmediator* and *GRGslave*, see below) also communicate making use of TCP/IP protocol, the specific realization of this data transport bus being irrelevant.

Computer algebra server

The next point of significant importance is the implementation of Computer Algebra (CA) facilities in *GRGnetwork*. Introducing no own low level CA primitives and algorithms, *GRGnetwork* leans on the general purposes CA system Reduce [1]. The latter is regarded here as the primary tool for the performing manipulations with symbolic mathematical data of a general kind. The routines realizing the processing of the specifically structured mathematical data (the subject going beyond the scope of the present notes) are also based on the Reduce capabilities, mostly those intended for the handling of low level symbolic data.

It is worth noting the following features of Reduce which are crucial for the selection of this system as CA server in *GRGnetwork*:

- Reduce is an *open* system allowing easy and efficiently realized extensions and/or modifications on the both 'top level' (handling of mathematical structured data) and 'bottom level' (handling of symbolic data primitives of a general kind).
- Reduce performs calculations in a maximally automatic way, assuming minimum of intervenience of a user at intermediate stages of a calculation.
- Reduce is a portable system which has been implemented in all computer platforms of practical interest, essentially.
- Reduce system includes the portable *compiler* allowing to build fairly compact efficient executable codes of high performance.

Although CAS Reduce is not free of some own (fairly essential) shortcomings, currently one sees no satisfactory alternative to it among the other universal CASs (Mathematica, Maple, Macsima, Maxima, muPad), at least, as far as the class of problems considered in the present work is concerned.

Security managing

The next point of high importance, the security issue, is one of the most substantial problems concerning any software making use of the contemporary Internet communication media. Considering its relation to *GRGnetwork*, a system based on the 'client-server' method, it has to be noted that there are obviously two its branches. Namely, these are the problem of secure run of the client component on the user' computer and the security of the functioning of the server component.

Concerning the latter side of the security problem, the important element of its solution in *GRGnetwork* is the assignment of all the communication functions to a special server component called *GRGmediator*. One of its main functions is the filtering out any possible detrimental intervention outwards.

Further, the loading/startup of client components of *GRGnetwork* and further their communication with network server (*i.e.*, 'physically', with *GRGmediator*) is realized making use of the *Java applet* facility. The latter is known as basically safe way of the data exchange and the handling of the codes transported through the Internet communication media. The point is that the code of a Java applet is not immediately executed by computer hardware but is interpreted by the so called *Java Virtual Machine* (JVM) integrated with (or 'plugged in') the web-browser which runs on the user' computer. Various measures ensuring safe execution of downloaded applet codes are implemented in JVM. Hence, a necessary security level is automatically ensured as long as the appropriate integrity and authenticity of web-browser itself is being kept. This strategy constitutes the main ingredient of the service responsible for the secure functioning of the client component of *GRGnetwork*.

Finally, the communication of a user with the CA package Reduce, playing role of CA server, is managed by the so called *PSL Security Guard* (*PSLguard*). The point is that PSL (*Portable Standard Lisp*), the lower layer Reduce is based upon, takes no care about the issue of a secure functioning in an open network environment. *PSLguard* is intended to compensate for this shortcoming, shielding the server against possible hostile attacks through the security holes lurked in the CAS background.

Issue of a practical realization

The concretizing and practical realization of the approaches sketched above prove to require simultaneous solution of a variety of interrelated problems of quite different kinds. This, in turn, requires fulfillment of a number of conditions which are outlined below.

Run-time environment

The solution implemented in *GRGnetwork* is based on the following decisions.

- The server part of *GRGnetwork*, *i.e.* the sub-components named *GRGmediator* and *GRGslave*, are realized as *Java applications* which run on sufficiently powerful (preferable distinct) computers within Java Run-time Environment (JRE) (or in framework of more recent equivalent packages abbreviated by their developers as JDK and J2SKD which contain JRE as their subsets).
- The computers where *GRGmediator* and *GRGslave* reside at have to be connected by a fast IP channel of a sufficient carrying capacity.
- *GRGslave* has to be able to run the processes of CA Reduce (the more ones simultaneously, the better).
- The client component startup procedure implies the making use of `http` server.
- The realization of the functioning of the client component of the package assumes

– the making use of WWW-browser with graphical interface and Java support.

Currently, such browsers are available for all the computer platforms, essentially. Their incomplete list includes various releases of *Netscape Communicator/Navigator* and *Microsoft Internet Explorer* (excluding the latest ones), *Opera, Mozilla, Konqueror, etc.* Some of them consume comparatively moderate computer resources to be able to run on personal computers of fairly moderate capabilities and performance rate.

– The access to Internet is the second (and last) condition necessary for the functioning of the client component; the making use of a local network instead is also feasible.

The existence and availability of the software mentioned above is a crucial point making the goal of the project potentially achievable.

Developer workplace

Another point of crucial importance is the structure and, accordingly, feasibility of realization of the hard/software complex which is natural to name a *developer workplace.* Essentially, the capabilities and efficiency of this necessary tool affects all the route of the subsequent work and, essentially, determines the total feasibility of the project.

It has to be emphasized that the problem to be solved implies the development of a *real time multi-thread assemblage of 'spatially distributed' asynchronously run programs communicated via TCP/IP Internet channels*, the latter maying reveal variable carrying capability (including low one) and limited overall reliability. It is also worth also noting that the work on the project was being carried out under rather unfavorable conditions with minimal support in the country with crisis economy. In reality, the last circumstance proved the next point strictly conditioned the way of the realization of the project. The latter can be summarized as follows:

Hardware: The work of the 'spatially distributed' multi-component package communicating with the help of TCP/IP protocol mentioned above is emulated on a single personal ('IBM PC-compatible') computer.

Operating system: The most important 'constructive element' consolidating all the software utilized is OS *Linux.* A number of releases of *Slackware, RedHat, Mandrake* (currently *ALTLinux*) distributions were tested and utilized.

This choice of OS immediately implies a number of advantages. The most important ones are the following:

TCP/IP support: This function is incorporated in the OS kernel. The corresponding utilities necessary for the maintaining TCP/IP connections are ensured as OS constituents as well.

Network emulation: Inter-computer TCP/IP communication is emulated on a single computer by OS facilities available defaults.

Java developer tools: Java programs are coded, tested, and run making use of Java Development Kit (JDK, further transformed to J2SKD package) of several subsequent releases supported by *Sun Microsystems*; another clone of similar software of comparable capabilities utilized was developed and distributed by *IBM Corporation*.

Http server: The well known http-server *Apache* (included in all the recent Linux distributions) is used for the emulating of the startup procedure of the client component of *GRGnetwork*. The *help* subsystem and support of online documenting are also maintained with its assistance.

WWW-browsers: Several WWW-browsers with graphical interface and Java support are used as hosts for the client component, most notable packages are Netscape Communicator clone (currently developed under the name Mozilla Communicator) and Microsoft Internet Explorer (in the case of client component running over OS Windows).

Algebraic processor: General purpose CA Reduce system of releases 3.6 and 3.7 running upon PSL package is used as the algebraic processor.

etc... A lot of auxiliary software some of which is, essentially, indispensable for the package development (various editors, documenting tools, graphics handling tools, etc.) are utilized.

Remark: A crucial circumstance allowing the practical development of *GRGnetwork* is the existence, availability, and ability to be consistently integrated of all the functional components of developer workplace listed above within the common shell running on a single computer of a moderate performance rate assembled from standard components.

After the formal characteristic of the package given above, more informative description of its very structure and the functioning is now in order.

GRGnetwork structure

In active state, *GRGnetwork* consists of a single server component and some (one or more) client components functioning asynchronously and running, generally speaking, on different computers. It is assumed that each client component lives on an own computer (in most typical case, on a personal user' computer) connected by TCP/IP channel with Internet. Through the latter, the client component can (and does) communicate with server computer of *GRGnetwork*.

Concerning the server component of *GRGnetwork* software, it has to be reminded that it actually consists of the two relatively independent sub-components called *GRGmediator* and *GRGslave*. This is *GRGmediator* which runs on the 'server computer' mentioned above and plays the role of a common interface communicating via Internet with client components, if there are active any.

The second server component, *GRGslave*, runs, preferably, on a computer distinct from the one where *GRGmediator* resides at (for the sake of performance decoupling). It is assumed that *GRGslave* is also connected with *GRGmediator* by TCP/IP channel, being at the same time, as far as one concerns the functioning of *GRGnetwork*, isolated of the other connections. This channel is assumed to be a sufficiently fast and 'absolutely reliable' because no special provisions is made for the data losses or significant transmitting delays in it. Note that no such limitations on the channels connecting *GRGmediator* and *GRGclients* are assumed.

Therefore, the graph of TCP/IP 'links' connecting the components of *GRGnetwork* constitute a star-like figure with *GRGmediator* situated in its vertex.

More about *GRGnetwork* components

Let us consider the functions of *GRGnetwork* components in more details.

GRGmediator: communication center

The main function of *GRGmediator* is the maintaining TCP/IP connection with *GRGclient* and *GRGslave*, receiving and dispatching the data and command streams, simple specific top-layer protocols being implemented for such a purpose. *GRGmediator* occupies a separate port and effectively filters outward calls coming in through it. The latter point is substantial for the ensuring the secure functioning of the server component.

GRGslave: application server

In a sense, *GRGslave* is the 'genuine' server in *GRGnetwork*. In particular, it manages the logical connections with client components, carries out the registration of new clients, starts and controls the CA server (CAS Reduce), maintains the remote file systems of clients (which point will be discussed below), performing all the operations over them 'on the physical level', etc. *GRGslave* also handles the necessary information concerning the registered clients, updates and stores the statistics characterizing their activity and consumption of the shared resources *etc.*

GRGclient: user interface

First of all, *GRGclient* plays the role of a friendly interface with a 'final user'. It also performs some useful service tricks specific for the work with CA systems. Hence, as far as one concerns the apparent convenience of the package use, this it is perhaps the most important component. *GRGclient* is realized in the graphical multi-window format and makes use of the major active control facilities typical for window-based GUI (buttons, various menus, *etc.*).

Another function of *GRGclient*, less apparent but definitely not less important, is the maintaining stable connection with the server component ('physically', with *GRGmediator*) in accordance with the corresponding protocol. The latter includes some facilities

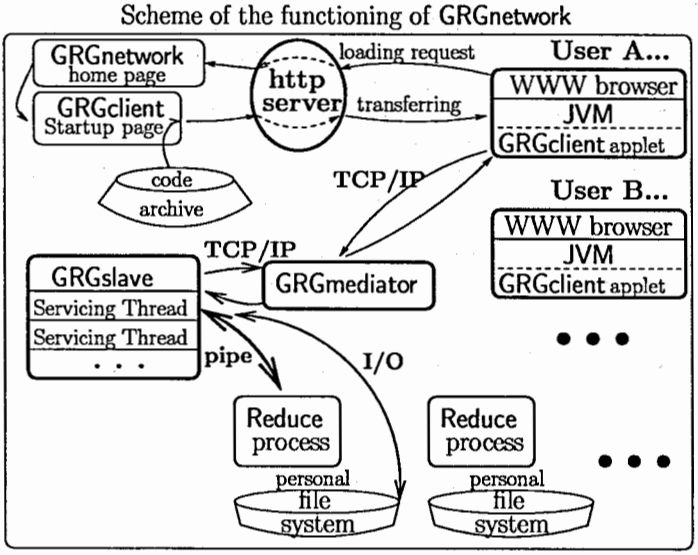


Figure 1: Package components and the major connection links for one of users (**A...**) are schematically displayed

allowing to restore the exchange by the commands and data in case of faults due to *time-outs* (comparatively large delays in data transfer) and even, up to some extent, a lost of data in communication channels (which may arise, *e.g.*, due to some technical problems).

GRGnetwork skeleton

Now let us consider the functioning of *GRGnetwork* from a more practical point of view. Its skeleton is schematically depicted in Fig. 1 where, in particular, the most important links including data and command streams, are schematically displayed.

What is necessary for the resorting to *GRGnetwork* service

To resort to the service of *GRGnetwork*, a user needs a 'computing unit' (*e.g.* a personal computer, access to a workstation, *etc.*) connected to Internet (or incorporated in the local network, provided the local installation of *GRGnetwork* server within the latter is used). It has to be able to run WWW-browser with graphical interface and Java support (for example, *Netscape Communicator* or *Microsoft Internet Explorer*). Generally speaking, no additional software (apart from a browser) need be installed.

Work session: start up and run

Having the above things at his/her disposal, a user begins the work with *GRGnetwork* with the opening the *homepage of GRGnetwork* (see [2]), that implicitly assumes the contact with the corresponding `http`-server.

This page contains, apart from some appropriate introductory elucidations and the references (links) to the auxiliary documenting matter *etc.*, the link leading to *startup* page containing the 'button' which enables one to issue the command to start the Java applet executing *GRGclient* code.

It has to be mentioned that in response to the command above, the code of *GRGclient* is automatically transferred from the same website by the browser. Hence, as it has been noted, no specific actions by user or, say, preliminary installation of additional software (apart from the browser itself) are required.

Having been started, after the initialization, *GRGclient* opens the window of the so called ControlPanel. Making use of the latter, the user may adjust some working parameters (in particular, to specify the 'name' under which he or she will be registered in *GRGnetwork*, provided the 'personalized' access is implied), to consult with the *help* notes, etc. However, at this stage, the main action ControlPanel enables one to carry out, is the establishing connection with *GRGmediator*.

It is also worth noting here that before the establishing connection with *GRGmediator* (which is carried out by means of a special procedure whose details are unseen for the user), *GRGclient* lives quite autonomously 'in the user's computer' ('inside the browser') and, essentially, is of no relation to *GRGnetwork*. Only after the establishing connection with *GRGmediator*, *GRGclient* turns out incorporated in *GRGnetwork*. Note also that the connection between *GRGclient* and *GRGmediator* is independent on the one established by the browser during the loading the content of *GRGnetwork* homepage. In particular, it is not serviced by the `http`-server (transferring the content of homepage to the browser). *GRGmediator* plays here the role of a communication server itself.

GRGmediator runs on the *GRGnetwork* server computer and maintains, apart from temporary connections with one or more *GRGclients*, the permanent connection with *GRGslave* which performs the major part of actual 'service work' in *GRGnetwork*. These actions are usually explicitly requested by client components. Making use of the graphical interface (GUI) ensured by *GRGclient*, the user directs appropriate commands to *GRGmediator* which dispatches them to *GRGslave* for subsequent execution. In response to these commands, *GRGslave* performs the actions requested. In particular, it starts (or stops, or restarts) CA Reduce processes, catching and linking (again resorting to the assistance of *GRGmediator*) the output data stream to the corresponding client, transports the Reduce input commands (constituting the program code written in the Reduce input language) to Reduce process, performs the file handling operations. The output of the application processes (*e.g.* CA Reduce), other information and control commands are received by the *GRGclient* applet through the reverse data transport bus ensured by *GRGmediator*. Finally, having get data generated by application, *GRGclient* usually put them into the corresponding window displayed on the computer monitor, making available for the user.

Some noteworthy features

Now we would like to outline some selected features of *GRGnetwork*. This matter should make the subject of our discussion more clear and definite.

Balancing rates of data generating and data use

It has to be noted that there is a substantial distinction in the remote interactive run of applied programs (including CAS) from the same work carried out with the help of a 'local' computer. This is the issue of the balancing of the rates of the 'generating' of output data by applied program (running on the remote server, in our case by *GRGslave*) and the 'consumption' of these data by the client component — finally, their utilizing, in one way or another, by the user.

A priori, the rates of these two processes may be very different. Especially often one encounters a situation where the data stream produced by a 'loose' applied program is, during some period of time, much more intensive than the maximal rate of the data processing the agent situated on the opposite side of the communication channel is able to ensure. In practice, the amount of data generated by an application can prove enormous in quite innocent working situations. For example, this can occur in the case of a logical mistake in the applied program leading to interminable data generating loop. One easily imagines also circumstances when the rate of the 'utilizing' of data by the network client practically vanishes becoming incompatible with fairly moderate rate of their generating.

A natural claim to any tool of remote application control including *GRGnetwork* is the ability to stably handle such situations. The essence of the problem is obvious: it is necessary to do something with data of volume permanently enhancing with time which had been produced by application but, in view of some reasons, still cannot be 'taken away' by the client component.

The method of the settling the problem indicated above which is adopted in *GRGnetwork* makes use of the 'quantization' of the both output (concerning the server component) and input (concerning the client component, respectively) data streams. Its idea is quite straightforward. Specifically, when the output data quantum (*e.g.* certain amount of text lines) received from application process by *GRGslave* is exceeded, it stops ('freezes') the application (provided the latter is generating further output data). Simultaneously, the client component is notified on such an event. Another reason of the 'freezing' of application may be the corresponding explicit command issued by the client component. It is generated, for example, if the user needs to suspend the reception of new output. Application run is continued (application output 'is melted') in response to the 'opposite' command of the client component (which can be either automatic or initiated by the user).

It turn, the client component stops to fill in the screen (*i.e.* some *TextArea* in the window of ApplicationControlPanel) with the new data either in the case of the filling up of the input buffer of certain volume or in response to the explicit user command (when the 'freeze' item in the appropriate menu is clicked). Then *GRGclient* directs to the *GRGslave* the directive to also 'freeze'. Since this action needs some time to be performed, client component is further collecting the messages such as those which have already been issued

and are being transported through the network, if any, without displaying them on the screen. This 'frozen' state of *GRGclient* continues until the user allows *GRGclient* to run further (*i.e.* to proceed to display the new data on the screen, removing the 'oldest' ones) by means of the corresponding command.

It is worth noting that the user may also *disable* this 'freezing/melting' mechanism. Then there will be no apparent pauses initiated by the client and the new data automatically ousts the 'most early' portions of data from the screen. At the same time, the quantization of the application output by *GRGslave* is still carried out, enabling the server to synchronize the the process of data transfer between *GRGnetwork* components.

The user is therefore provided with necessary tools (controlled with the help of GUI of the client component) enabling him or her to efficiently control the intensity of the stream of output data produced by application and its displaying on the monitor screen.

Remote libraries

A useful service worth noting here which is provided by *GRGnetwork* is an easy access to some 'standard' data libraries situated on the server. Currently, the library of the standard tests (samples of usage in fact) of numerous Reduce applied packages is supported.

On the screen, the structure of data libraries is represented in the form of tree-like collection of specifically linked menus. Having chosen some 'leave' (representing a concrete file), a user can either elicit its content, putting it 'to the screen', or submit to CAS Reduce as the input code for the processing. In the latter case the output generated by Reduce is returned to the screen. The interactive programs, such as *Reduce Lessons* by D.R. Stoutemyer, for example, are executed in a usual way, just like on a 'local' computer (if one ignores the facility of the freezing/melting the output data stream discussed above). Additionally, the system inspects the library of compiled codes of the standard packages (another 'standard library' with a distinct form of access) searching for the one corresponding to the test file (*i.e.* the package) chosen and, if there is any, enables the user to *load* this package by means of a 'single mouse click'.

Anonymous and personalized access

Generally speaking, the main service of *GRGnetwork* is the ensuring of the remote access from an 'arbitrary computer' to the capabilities of CA system which is, currently, the CAS Reduce; the interfaces to some specialized packages running over Reduce (see, in particular, [3],[4]) are planned to be implemented.

It has to be noted that there are two modes of such an access. It is convenient to call them *anonymous* and *personalized*, respectively. Formally, these modes are distinct by the necessity, in the case of personalized access, to submit to *GRGnetwork* server the 'personal' name of the user and the associated password. The procedure of the establishing 'personalized' connection claims these data from the user (while connecting as *anonymous client*, a user introduces no identifying – or any other – information).

Commenting this point, it has to be noted that, technically, in the case of personalized access, there appears feasibility for the server to properly attribute subsequent sessions of

the same user. This ability enables it to ensure such 'named' users with some 'long-term' (multi-session) resources.

On the contrary, various 'anonymous' work sessions of the *same* user cannot be attributed to their actual 'author'. All the anonymous users (there may be many of them simultaneously connected to *GRGnetwork* at each moment of time while the named user may simultaneously lead the only session alone) are treated on the equal ground being considered, on the one hand, as unrelated ones, while, on the other hand, they are provided with the common long-term resources alone. This distinction is substantial in the handling of the file systems which are supplied by *GRGnetwork* to any registered user, both named and anonymous, which point will be outlined below.

Remote file systems

GRGnetwork supplies users with remote file systems constituted of directories (folders) and data files. These are realized in the standard way as multiple-branching trees with nodes-folders and leaves-files which are created in response to user commands. All these data 'physically' reside on *GRGnetwork* server and are maintained by *GRGslave*.

All the *anonymous* clients are endowed with the common file system and possess identical rights on all its resources. In this case, in view of impossibility of the 'personal' attributing of concrete data sets to any 'specific anonymous user', the natural restrictions on the set of allowable file operations over the data files and folders are imposed. In particular, any operation which potentially could deteriorate for anybody the access to some data (for example, the altering in some way the path to it) is not allowed. These restrictions include in particular any removing or renaming files or directories, etc.

On the contrary, every *named* client (*i.e.* the one making use of personalized access mode) is supplied with own, 'individual', file system. The access to this system is protected by a password. In such a case, the full set of operations with the files and directories is allowed (and the corresponding executive tools are provided). The only substantial limitation worth mentioning is the restriction imposed on the overall data volume. The quota-based rule (regulated by system administrator) to the providing of the data storage resources to named users is followed.

The file operations with remote personal file systems provided to named users allows the following:

- to store textual data get from *Input TextArea* of ApplicationControlPanel to a named file;
- to reproduce the content of the file in the *Output TextArea* of the same window;
- to create named folders;
- to copy, to remove, to rename, and to move separate files and folders with all their content;
- to process the content of files by CAS Reduce, sending output to the monitor screen (to *Output TextArea*);

- to search for files and folders in the filesystem tree by their names;
- and more...

Conclusion

The outcome of the work reported is currently (June 2001) the software package consisting of the three main components called *GRGclient*, *GRGmediator*, *GRGslave* with associated documentation (currently not accomplished, will be enhanced). *GRGclient* component with the corresponding startup facilities and the draft of *User' guide* are available (in the demonstrative mode) on URL [2].

It has to be noted that this software should not be considered as a finalized product. The further enhancing of its capabilities is planned, some additional debugging still retains to be performed. Nevertheless in its recent state of development *GRGnetwork* already becomes an efficient tool able to provide valuable assistance in solution of problems by the methods of Computer Algebra.

Acknowledgments

This work was partially supported by Intas in frames of the grant 961-842. The author is grateful to Yu.N. Obukhov for the valuable collaboration in the test exploitation of the package under OS Solaris and useful discussions.

References

[1] A.C. Hearn, *REDUCE User's Manual. Version 3.6.* Rand publication CP78 (Rev. 7/95) (RAND, Santa Monica, CA 90407-2138, USA, 1995).

[2] S.I. Tertychniy, *GRGnetwork* homesite, `http://www.thp.uni-koeln.de/~yo/GR_ALGEBRA/html/index.html`, 2001 (in progress).

[3] S.I. Tertychniy, I.G. Obukhova, *GRGEC: Computer Algebra System for Applications to Gravity Theory, SIGSAM Bulletin* **31** (1997) 6

[4] S.I. Tertychniy, *Searching for electrovac solutions to Einstein–Maxwell equations with the help of computer algebra system GRG-EC* (LANL archive, gr-qc/9810057, 1998).

Geometry of Pairs of Keplerian Elliptic Orbits

N.N. Vassiliev, K.V. Kholshevnikov

Steklov Mathematical Institute,
St.Petersburg State University

We study using computer algebra methods a class of problems connected with mutual configaration of two Keplerian elliptic orbits. The first one is the problem of finding critical points of the distance function between two keplerian elliptic orbits. We reduce it to determination of all real roots of a trigonometric polynomial of degree eight. The coefficients of the polynomial are rational functions of orbital parameters. Using Groebner bases technigue we show that a polynomial of smaller degree with such properties does not exist. This fact shows that our result cannot be improved and allows us to construct an optimal algorithm to find the minimal distance between two keplerian orbits. We define also a function on the set of pairs of Keplerian ellipses so that sign of the function would be a topological invariant of their configuration. The sign is negative if and only if the related ellipses are linked. Explicit formulae representing the linking coefficients as functions of orbital elements are deduced. We suggest different ways to use these coefficients for determining exact intersections of pairs of osculating Keplerian orbits. Also we study different metrizations for the space of pairs of Keplerian elliptic orbits.

Exact and Asymptotic Solutions for the General Hénon–Heiles System

S.Yu. Vernov

Institute of Nuclear Physics,
Moscow State University,
Vorobievy Gory, Moscow, 119899, Russia;
e-mail: svernov@theory.sinp.msu.ru

1 The Hénon–Heiles system

Let us consider a three-dimensional galaxy with an axial symmetric and time-independent potential function. The equations of galactic motion admit of two well-known integrals: energy and angular momentum. If we know also the third integral of motion then we can solve the motion equations by the method of quadratures. However, the third integral as polynomial function does not exist in general case.

In the 1960s, numerical [1] and asymptotic methods [2, 3] have been developed to show existence or absence of the third integral for some polynomial potentials.

In [1] (1964) Hénon and Heiles wrote: "In order to have more freedom of experimentation, we forgot momentarily the astronomical origin of the problem and consider its general form: does an axisymmetrical potential admit a third isolating integral of motion ?". They chose Hamiltonian

$$H = \frac{1}{2}\left(x_t^2 + y_t^2 + x^2 + y^2\right) + x^2 y - \frac{1}{3}y^3,$$

because: **(a)** it is analytically simple; this makes the numerical computations of trajectories easy; **(b)** at the same time, it is sufficiently complicated to give trajectories which are far from trivial. Indeed, Hénon and Heiles found that for low energies this system appeared to be integrable, in so much as trajectories (numerically integrated) always lay on well defined two-dimensional surfaces. On the other hand, they also obtained that for high energies many of these integral surfaces were destroyed and that phase space acquired large ergodic regions. The Hénon–Heiles system became a paradigm of chaotic Hamiltonian dynamics.

Let us consider the **Hénon–Heiles** Hamiltonian in the general form:

$$H = \frac{1}{2}\left(x_t^2 + y_t^2 + \lambda x^2 + y^2\right) + x^2 y - \frac{C}{3}y^3, \tag{1}$$

where C and λ are numerical parameters.

Investigations of the general Hénon–Heiles system:

$$\begin{cases} x_{tt} &= -\lambda x - 2xy, \\ y_{tt} &= -y - x^2 + Cy^2, \end{cases} \tag{2}$$

can be separated on the following ways:

1. The solutions of (2) have been investigated in the complex time plane using numerical integration techniques [4, 5]. Numerical investigations of the Hénon–Heiles system are continued up to now [6, 7].

2. The second integral as formal power series in phase variables x, x_t, y and y_t (Gustavson integral) has been constructed for the Hénon–Heiles system both in the original ($\lambda = 1$, $C = 1$) [3] (see also [8]) and in the general forms [9]. To obtain the second integral Gustavson has constructed the normal form of Hamiltonian (1). Using the Bruno algorithm [10] V.F. Edneral has constructed the Poincaré–Dulac normal form and found (provided the energy is small) local families of periodic solutions of Hénon–Heiles system both in the original [11] and in the general [12] forms.

3. The singularities at the fixed points in phase space are locally analyzed via normal form theory, whereas the singularities in the complex (time) plane are studied by the Painlevé analysis. Using this analysis three integrable cases of the general Hénon–Heiles system have been found:

$$\begin{array}{lll} \text{(i)} & C = -1, & \lambda = 1, \\ \text{(ii)} & C = -6, & \lambda \textbf{ is an arbitrary number}, \\ \text{(iii)} & C = -16, & \lambda = \frac{1}{16}. \end{array}$$

The nontrivial cases are (ii) and (iii), case (i) is separable. In the nontrivial integrable cases four parameter solutions [13] and the second integrals [14–17] have been found[1]. Further development in the Painlevé approach has led to finding of Bäcklund transformation [13] and Lax pairs [19] for the case (ii). The three integrable cases of the Hénon–Heiles system correspond precisely to the stationary flows of the only three integrable cases of *fifth-order polynomial nonlinear evolution* equations of scale weight 7 (respectively the Sawada–Kotega, the fifth-order Korteweg–de Vries and the Kaup–Kupershmidt equations) [20].

4. It is known that the variable y as a solution of the Hénon–Heiles system has to satisfy the following fourth-order equation:

$$y_{tttt} = (2C - 8)y_{tt}y - (4\lambda + 1)y_{tt} + 2(C + 1)y_t^2 + \frac{20C}{3}y^3 + (4C\lambda - 6)y^2 - \lambda y - 4H. \quad (3)$$

To find a partial solution of this equation one can assume that y satisfies some more simple equation. For example, it is well-known that the Hénon–Heiles system and, hence, equation (3) have solutions in terms of Weierstrass elliptic functions satisfying the first-order differential equation:

$$y_t^2 = \mathcal{A}y^3 + \mathcal{B}y^2 + \mathcal{C}y + \mathcal{D}, \quad (4)$$

where $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ are some constants.

E.I. Timoshkova [21] considered the following equation:

$$y_t^2 = \tilde{\mathcal{A}}y^3 + \tilde{\mathcal{B}}y^2 + \tilde{\mathcal{C}}y + \tilde{\mathcal{D}} + \tilde{\mathcal{G}}y^{5/2} + \tilde{\mathcal{E}}y^{3/2} \quad (5)$$

[1] An geometrical interpretation of the second integral in the case (ii) is presented in [18]

instead of equation (4) and found new one parameter sets of solutions of the Hénon–Heiles system in nonintegrable cases ($C = \frac{-4}{3}$ or $C = \frac{-16}{5}$, λ is arbitrary number).

In present paper I use the Painlevé method to find an asymptotic solutions of Hénon–Heiles system in the case $C = \frac{-16}{5}$.

2 The Painlevé property

The Painlevé property for ordinary differential equations (ODE's) is defined as follows. The solutions of a system of ODE's are regarded as analytic functions of a complex (time) variable [22, 23]. **The movable singularity** of the solution is the singularity, location of which depends on initial conditions.

Definition. A system of ODE's is said to have **the Painlevé property** if all movable singularities of its general solution are single-valued (simple poles) [24].

All solutions of such systems (systems of P-type) are expressible as simple Laurent series. As known a Hamiltonian system in a $2s$–dimensional phase space is called *completely integrable* if it possesses s independent integrals which commute with respect to the associated Poisson bracket. When this is the case, the equations of motion are (in principal at least) separable and solutions can be obtained by the method of quadratures. Since the work of S.V. Kovalevskaya [25] (see also [26]) on the motion of a rigid body about fixed point, the Painlevé property has been proposed as a criterion for complete integrability [27, 28]. If the system misses the Painlevé property (has complex or irrational "resonances"), then the system cannot be "algebraically integrable" [29] (see also [30] and references there in). N. Ercolani and E.D. Siggia [31, 32] advance arguments as to why the Painlevé test works, *i.e.* they showed how to exploit the singular analysis to yield the integrals. They proved a theorem which demonstrates that the singularity analysis provides bounds on the degrees of polynomial integrals for a large class of separable systems.

3 The Painlevé test

3.1 Various algorithms of the Painlevé test

The Painlevé test is any algorithm designed to determine necessary conditions for a differential equation to have the Painlevé property. The original algorithm developed by Painlevé [24] is known as the α-method. The method of S.V. Kovalevskaya [25] is not as general as the α-method but is more simple than it is. In 1980, motivated by the work of S.V. Kovalevskaya, M.J. Ablowitz, A. Ramani and H. Segur [33] developed a new algorithm of the Painlevé test for ODE's[2]. Using this algorithm one can determine whether

[2]In [33] Ablowitz *et al.* have proven that if a partial differential equation (PDE) is solvable by the inverse scattering transform then a system of ODE's is obtained from this PDE by the exact similarity reduction the solutions (of this system of ODE's) associated with Gel'fand–Levitan–Marchenko equation will possess the Painlevé property. Furthermore, they conjecture that, when all the ODE's obtained by exact similarity transforms from a given PDE have the Painlevé property, perhaps after a change of variables, then PDE will be "integrable". A powerful generalization of the Painlevé test for both nonlinear ordinary and partial differential equations was developed by J.Weiss, M.Tabor and G.Carnevale [34, 35]

a system of ODE's admits movable branch points, either algebraic or logarithmic. This algorithm can be used not only to isolate values of parameters of integrable cases, but also to find asymptotic partial solutions in nonintegrable cases. The test consists of three levels, if the system passes some level then it means that the system may be a system of P-type and we have to check the system on the following level.

3.2 The first level: find the dominant behavior

We assume that the dominant behavior of solutions in a sufficiently small neighborhood of the singularity is algebraic. To find the dominant behavior we look for solutions in the form

$$x = d_1(t-t_0)^\alpha \qquad \text{and} \qquad y = d_2(t-t_0)^\beta, \tag{6}$$

where t_0 is arbitrary. Substitution (6) into (2) shows that for certain values of α and β, two or more terms in the equations of (2) may balance (these terms have the same powers), and the rest can be ignored as $t \longrightarrow t_0$. For each choice of α and β the terms which can balance are called *the leading terms.* For the Hénon–Heiles system in the general form there exist two possible dominant behaviors [5, 37]:

Case 1:		*Case 2*: ($\beta < \Re e(\alpha) < 0$)	
$\alpha = -2,$	$\beta = -2$	$\alpha = \frac{1 \pm \sqrt{1-48/C}}{2},$	$\beta = -2$
$d_1 = \pm 3\sqrt{2+C},$	$d_2 = -3$	$d_1 = arbitrary,$	$d_2 = -\frac{6}{C}$

It is possible that an original system is not of P-type, but after change of variables the obtained system is of P-type[3]. For $C = -\frac{16}{5}$ in *Case 2* we obtain $\alpha = -\frac{3}{2}$ and to continue the Painlevé test we have to introduce new variables $z = x^2$ and to consider the following system instead of system (2):

$$\begin{cases} z_{tt}z &= \frac{1}{2}z_t^2 - 2\lambda z^2 - 4z^2 y, \\ y_{tt} &= -y - z - \frac{16}{5}y^2. \end{cases} \tag{7}$$

3.3 The second level: find the resonances

For each obtained pair of values of $\tilde{\alpha} \equiv 2\alpha$ and β we construct the simplified system that retains only the leadings terms of equations of the original system (7).

For $\tilde{\alpha} = -4,\ \beta = -2$ the simplified system is

$$\begin{cases} z_{tt}z &= \frac{1}{2}z_t^2 - 4z^2 y, \\ y_{tt} &= -z - \frac{16}{5}y^2. \end{cases} \tag{8.1}$$

Substituting

$$z = -\frac{54}{5}(t-t_0)^{-4} + b_1(t-t_0)^{-4+r} \quad \text{and} \quad y = -3(t-t_0)^{-2} + b_2(t-t_0)^{-2+r}$$

(see also [19, 36, 37]). The Painlevé test for both ODE's and PDE's which is based on perturbation theory is presented in [38].

[3]In this case the original system is said to have the weak Painlevé property. For example, in the integrable case (iii) system (2) has the weak Painlevé property.

in system (8.1), we obtain that to leading order in $(t-t_0)$ $\{(t-t_0)^{2r-8}$ for the first equation and $(t-t_0)^{r-4}$ for the second equation$\}$ this system reduces to a system of two linear algebraic equations:

$$\hat{Q}(r)\bar{b} = 0, \tag{9}$$

where $\hat{Q}(r)$ is 2×2 matrix, which elements depend on r, and $\bar{b} \equiv (b_1, b_2)$. Determinant $det(\hat{Q}(r))$ is a polynomial of order 4. Equation (9) has nonzero solution only if

$$det(\hat{Q}(r)) = 0. \tag{10}$$

The roots of equation (10):

$$r_1 = -1, \qquad r_2 = 6, \qquad r_3 = \frac{5+\sqrt{53.8}}{2}, \qquad r_4 = \frac{5-\sqrt{53.8}}{2}$$

determine resonances (one root is always (-1), it represents the arbitrariness of t_0). Some roots of (10) are not integer. This result means, that the Hénon–Heiles system with $C = \frac{-16}{5}$ is a nonintegrable system. There is no algorithm to find the general solution for a nonintegrable system.

To find a partial asymptotic solutions let us consider the dominant behavior in the case of $\tilde{\alpha} = -3$ and $\beta = -2$. The simplified system is[4]

$$\begin{cases} z_{tt}z &= \frac{1}{2}z_t^2 - 4z^2y, \\ y_{tt} &= -\frac{16}{5}y^2. \end{cases} \tag{8.2}$$

Substituting

$$z = d_1(t-t_0)^{-3} + b_1(t-t_0)^{-3+r} \quad \text{and} \quad y = -\frac{15}{8}(t-t_0)^{-2} + b_2(t-t_0)^{-2+r}$$

into (8.2) we repeat calculations and obtain that resonances and corresponding arbitrary parameters can arise in terms proportional to $(t-t_0)^{-2+r}$, where $r = -1,\ 0,\ 4,\ 6$. Root $r = -1$ corresponds to arbitrary parameter t_0, root $r = 0$ corresponds to arbitrary parameter d_1, other roots correspond to new arbitrary parameters, *i.e.* new constants of integration[5].

3.4 The third level: find the constants of integration

The third (and the last) level of the Painlevé test is a substitution into the original (not simplified) system (7) the following series:

$$z = d_1(t-t_0)^{-3} + \sum_{j=1}^{r_{max}} \tilde{cz}_{j-3}(t-t_0)^{j-3} \quad \text{and} \quad y = -\frac{15}{8}(t-t_0)^{-2} + \sum_{j=1}^{r_{max}} \tilde{cy}_{j-2}(t-t_0)^{j-2},$$

where $r_{max} = 6$, $\tilde{cy}_j$ and $\tilde{cz}_j$ are unknown constants.

[4]The simplified systems (8.1) and (8.2) are different.

[5]Values of roots r in the case of arbitrary C see in [5, 37].

After this substitution system (2) is transformed to sequence of systems of linear algebraic equations. Solving these systems we find $\widetilde{cy}_j$ and $\widetilde{cz}_j$. Determinants of systems, which correspond to resonances, have to be zero.

For example, to determine $\widetilde{cy}_2$ and $\widetilde{cz}_1$ we have obtained the following system:

$$\begin{cases} 557056d_1^6 + d_1^4\Big(15552000\lambda - 4860000\Big) + \\ d_1^2\Big(864000000\widetilde{cy}_2 + 108000000\lambda^2 - 67500000\lambda + 10546875\Big) = 0, \\ 818176d_1^4 + d_1^2\Big(15660000\lambda - 4893750\Big) - 810000000\widetilde{cy}_2 - 6328125 = 0. \end{cases} \tag{11}$$

As one can see this system does not include $\widetilde{cz}_1$, so $\widetilde{cz}_1$ is an arbitrary parameter (a constant of integration). For each of values of λ this system can be solved as system in $\widetilde{cy}_2$ and d_1. We obtain new constant of integration $\widetilde{cz}_1$, but we must fix d_1, so number of constants of integration is equal to 2. It is easy to verify that $\widetilde{cy}_4$ is an arbitrary parameter, because the corresponding system is equivalent to one linear equation. So, we obtain an asymptotic solution which depends on three parameters: t_0, $\widetilde{cz}_1$ and $\widetilde{cy}_4$.

3.5 New asymptotic solutions

Now it is easy to obtain asymptotic solutions with arbitrary accuracy. For given values of λ one has to choose d_1 as one of the roots of system (11). After this the coefficients $\widetilde{cz}_j$ and $\widetilde{cy}_j$ as functions of $\widetilde{cz}_1$ and $\widetilde{cy}_4$ can be found automatically due to a computer algebra system, for example, REDUCE [39, 40]. For some values of λ asymptotic solutions have been found as the following series (without loss the generality we can put $t_0 = 0$):

$$z = d_1 t^{-3} + \sum_{j=1}^{50} \widetilde{cz}_{j-3} t^{j-3}, \qquad y = -\frac{15}{8} t^{-2} + \sum_{j=1}^{50} \widetilde{cy}_{j-2} t^{j-2}.$$

For example, if $\lambda = \frac{1}{9}$ then system (11) has the following solutions ($d_1 \neq 0$):

$$\left\{ d_1 = \pm\frac{25\sqrt{2}}{16}, \quad \widetilde{cy}_2 = -\frac{1819}{663552} \right\}, \qquad \left\{ d_1 = \pm\frac{25i\sqrt{13}}{8\sqrt{374}}, \quad \widetilde{cy}_2 = -\frac{8700683}{1364926464} \right\}.$$

Let us consider in detail the case $d_1 = \frac{25\sqrt{2}}{16}$. The solution is:

$$\begin{aligned} z = {} & \frac{25\sqrt{2}}{16} t^{-3} + \frac{125}{192} t^{-2} + \frac{25\sqrt{2}}{768} t^{-1} + \frac{1625}{82944} + \widetilde{cz}_1 t + \\ & + \left(\frac{21845}{47775744} - \frac{\sqrt{2}}{6} \widetilde{cz}_1 \right) t^2 + \left(\frac{437425\sqrt{2}}{9172942848} - \frac{25\sqrt{2}}{48} \widetilde{cy}_4 - \frac{191}{3456} \widetilde{cz}_1 \right) t^3 + \ldots, \\ y = {} & -\frac{15}{8} t^{-2} + \frac{5\sqrt{2}}{32} t^{-1} - \frac{205}{2304} + \frac{115\sqrt{2}}{13824} t - \frac{1819}{663552} t^2 + \\ & + \left(\frac{1673\sqrt{2}}{11943936} + \frac{1}{6} \widetilde{cz}_1 \right) t^3 + \widetilde{cy}_4 t^4 + \left(\frac{1044461\sqrt{2}}{220150628350} - \frac{19}{9216} \widetilde{cz}_1 - \frac{1}{2} \widetilde{cy}_4 \right) t^5 + \ldots. \end{aligned} \tag{12}$$

In the following table I present how $\widetilde{cz}_{50}$ and $\widetilde{cy}_{50}$ (coefficients of terms proportional to t^{50}) depend on the arbitrary parameters $\widetilde{cz}_1$ and $\widetilde{cy}_4$:

$\widetilde{cz}_1$	$\widetilde{cy}_4$	$\widetilde{cz}_{50}$	$\widetilde{cy}_{50}$	$\widetilde{cz}_1$	$\widetilde{cy}_4$	$\widetilde{cz}_{50}$	$\widetilde{cy}_{50}$
-1	-1	4×10^{-12}	-1×10^{-13}	0	0	-1×10^{-44}	2×10^{-45}
-1	-0.6	4×10^{-12}	5×10^{-14}	0	0.2	-8×10^{-20}	-3×10^{-20}
-1	-0.2	-1×10^{-17}	-2×10^{-18}	0	0.4	-2×10^{-17}	-9×10^{-18}
-1	0	-1×10^{-20}	3×10^{-22}	0	0.6	-5×10^{-16}	-5×10^{-16}
-1	0.4	6×10^{-15}	1×10^{-16}	0	0.8	-5×10^{-15}	-2×10^{-15}
-1	1	6×10^{-12}	8×10^{-14}	0	1	-3×10^{-14}	-1×10^{-14}
-0.6	-1	3×10^{-12}	-5×10^{-14}	0.4	0	-4×10^{-25}	-1×10^{-26}
-0.6	-0.6	4×10^{-14}	-1×10^{-15}	0.4	0.4	-1×10^{-15}	2×10^{-17}
-0.6	0	-5×10^{-23}	9×10^{-25}	0.4	0.8	-3×10^{-13}	2×10^{-15}
-0.6	0.4	3×10^{-15}	4×10^{-17}	0.8	0	-1×10^{-21}	-3×10^{-23}
-0.6	1	3×10^{-12}	2×10^{-14}	0.8	0.4	-2×10^{-15}	2×10^{-16}
-0.2	-1	1×10^{-12}	-2×10^{-14}	0.8	0.8	-6×10^{-13}	1×10^{-14}
0	-1	-3×10^{-14}	-1×10^{-14}	1	1	-4×10^{-12}	1×10^{-13}
20	20	-2.2	0.051	20	40	-603	6.88
40	20	-11.1	0.01	40	40	-1128	24.5

One can see that coefficients tend to zero very rapidly when the absolute values of the parameters are less than unit.

4 The connection between asymptotic solutions and exact solutions

E.I. Timoshkova [21] found that solutions of the following equation:

$$y_t^2 = \tilde{A}y^3 + \tilde{G}y^{5/2} + \tilde{B}y^2 + \tilde{E}y^{3/2} + \tilde{C}y + \tilde{D}, \tag{13}$$

where $\tilde{D} = 0$, $\tilde{A} = -\frac{32}{15}$, $\tilde{B}$, $\tilde{C}$, $\tilde{G}$ and $\tilde{E}$ are some depending on λ constants, satisfy (3) in the case of $C = \frac{-16}{5}$. After change of variables: $y = \varrho^2$, we obtain the following equation:

$$\varrho_t^2 = \frac{1}{4}\Big(\tilde{A}\varrho^4 + \tilde{G}\varrho^3 + \tilde{B}\varrho^2 + \tilde{E}\varrho + \tilde{C}\Big). \tag{14}$$

Equation (14) is of P-type. The general solution of equation (14) has only one arbitrary parameter t_0 and can be expressed by elliptic functions [41, 42].

For all values of λ the trajectories of motion are given by the following equation:

$$x^2 \equiv z = -\frac{1}{2}\left(\frac{5}{2}\tilde{G}y^{3/2} + 2(\tilde{B}+1)y + \frac{3}{2}\tilde{E}y^{1/2} + \tilde{C}\right), \tag{15}$$

where constants $\tilde{B}$, $\tilde{C}$, $\tilde{G}$ and $\tilde{E}$ depend on λ.

Let us compare the asymptotic solutions (12) with these exact solutions. It is easy to verify, for example, with help of the Painlevé test of equation (14), that at points of singularities these solutions and our asymptotic solutions have the same asymptotic behavior. It means that for some values of parameters $\widetilde{cz}_1$ and $\widetilde{cy}_4$ our asymptotic series give exact solutions and, hence, converge.

Let us consider in detail the case of $\lambda = \frac{1}{9}$. In this case the following solution has been found [21]:

$$x^2 \equiv z = -\frac{y}{3}\left\{\frac{5}{3} - 2i\sqrt{\frac{5y}{3}}\right\},$$

where y is a solution of the following equation:

$$y_t^2 = -\frac{32}{15}y^3 - \frac{4}{9}y^2 - \frac{8i}{\sqrt{135}}y^{5/2}.$$

Without loss the generality we assume that a solution has singularity at point $t = 0$, then the Laurent series for y and z are[6]:

$$z = \frac{25\sqrt{2}}{16}t^{-3} + \frac{125}{192}t^{-2} + \frac{25\sqrt{2}}{768}t^{-1} + \frac{1625}{82944} + \frac{3205\sqrt{2}}{3981312}t + \frac{9025}{47775744}t^2 + \frac{728395\sqrt{2}}{18059231232}t^3 + \ldots$$

$$y = -\frac{15}{8}t^{-2} + \frac{5\sqrt{2}}{32}t^{-1} - \frac{205}{2304} + \frac{115\sqrt{2}}{13824}t - \frac{1819}{663552}t^2 + \frac{6551\sqrt{2}}{23887872}t^3 - \frac{858455}{12039487488}t^4 + \ldots$$

You can see that this solution is identical with solution (12), with the following values of parameters:

$$\widetilde{Cy} \equiv \widetilde{cy_4} = -\frac{858455}{12039487488} \approx -7 \times 10^{-5}, \qquad \widetilde{Cz} \equiv \widetilde{cz_1} = \frac{3205\sqrt{2}}{3981312} \approx 10^{-3},$$

so, our asymptotic series converge for these values of parameters. It is possible that our asymptotic series converge also for another values of parameters, which are close to $\widetilde{Cy}$ and $\widetilde{Cz}$. I plan to investigate this question in future publications.

Conclusions

Using the Painlevé analysis one can not only define is the system integrable or not, but also find partial asymptotic solution even in nonintegrable case as well.

We have found the partial solution of the Hénon–Heiles system with $C = \frac{-16}{5}$. This partial solution is an asymptotic solution and depends on three parameters. Our asymptotic solutions with some values of two parameters coincide with one parameter set of exact solutions.

The author is grateful to R. I. Bogdanov and V. F. Edneral for valuable discussions and E. I. Timoshkova for comprehensive commentary of [21]. This work has been supported by the Russian Foundation for Basic Research under grant №№ 00-15-96577.

[6]It is easy to verify that

$$y = -\frac{5}{3\left(1 - 3\sin\left(\frac{t-t_0}{3}\right)\right)^2}, \qquad z = \frac{25}{27\left(1 - 3\sin\left(\frac{t-t_0}{3}\right)\right)^2}\left\{1 + \frac{2}{1 - 3\sin\left(\frac{t-t_0}{3}\right)}\right\}.$$

The energy of these solutions is equal to zero: $H = 0$.

References

[1] M. Hénon, C. Heiles, The Applicability of the Third Integral of Motion: Some Numerical Experiments, *Astronomical J.*, 1964, vol. 69, pp. 73–79.

[2] G. Contopoulos, A Third Integral of Motion in a Galaxy, *Zeitschrift für Asrtophysik*, 1960, vol. 49, pp. 273–291; On the Existence of a Third Integral of Motion, *Astronomical J.*, 1963, vol. 68, pp. 1–14; Resonance Cases and a Small Divisors in a Third Integral of Motion, *Astronomical J.*, 1963, vol. 68, pp. 763–779.

[3] A.G. Gustavson, On Constructing Formal Integrals of a Hamiltonian System Near an Equilibrium Point, *Astronomical J.*, 1966, vol. 71, pp. 670–686.

[4] Y.F. Chang, M. Tabor, J. Weiss, G. Corliss, On the Analytic Structure of the Hénon–Heiles System, *Phys. Lett. A*, 1981, vol. 85, pp. 211–213.

[5] Y.F. Chang, M. Tabor, J. Weiss, Analytic Structure of the Hénon–Heiles Hamiltonian in integrable and nonintegrable regimes, *J. Math. Phys.*, 1982, vol. 23, pp. 531–538.

[6] J.C.B. de Figueiredo, C.G. Ragazzo, C.P. Malta, Two Important Numbers in the Hénon–Heiles System, *Phys. Lett. A*, 1998, vol. 241, pp. 35–40.

[7] A.P.S. de Moura, P.S. Letelier, Fractal basins in Hénon–Heiles and other Polynomial Potentials, *Phys. Lett. A*, 1999, vol. 256, pp. 362–368.

[8] J.K. Moser, Lecture on Hamiltonian Systems, *Memoirs of the AMS*, 1968, vol. 81, pp. 1–60, {in English}, 1973, "Mir", Moscow, {in Russian}.

[9] M. Braun, On the Applicability of the Third Integral of Motion, *J. of Differential Equations*, 1973, vol. 13, pp. 300–318.

[10] A.D. Bruno, Local method in Nonlinear Differential Equations, Springer Series in Soviet Mathematics, ISBN-3-540-18926-2, 1989; Power Geometry in Algebraic and Differential Equations, Moscow, Nayka, 1998, {in Russian}.

[11] V.F. Edneral, A Symbolic Approximation of Periodic Solutions of the Hénon–Heiles System by the Normal Form Method, *Mathematics and Computers in Simulations*, 1998, vol. 45, pp. 445–463.

[12] V.F. Edneral, Bifurcation Analysis of Low Resonant Case of the Generalized Hénon–Heiles System, *Proc. of CASC 2001 (Konstanz, Germany, 2001)*, ed. by Ganzha et al., 2001, Springer, pp. 167–176.

[13] J. Weiss, Bäcklund Transformation and Linearizations of the Hénon–Heiles System, *Phys. Lett. A*, 1984, vol. 102, pp. 329–331; Bäcklund Transformation and the Hénon–Heiles System, *Phys. Lett. A*, 1984, vol. 105, pp. 387–389.

[14] J. Greene, Preprint La Jolla Institute LJI-TN-81-122, 1981.

[15] B. Grammaticos, B. Dorizzi, R. Padjen, Painlevé Property and Integrals of Motion for the Hénon–Heiles System, *Phys. Lett. A*, 1982, vol. 89, pp. 111–113.

[16] L.S. Hall, A Theory of Exact and Approximate Configurational Invariants, *Physica D*, 1983, vol. 8, pp. 90–106.

[17] A.P. Fordy, Hamiltonian Symmetries of the Hénon–Heiles System, *Phys. Lett. A*, 1983, vol. 97, pp. 21–23.

[18] C. Clementi, M. Pettini, A Geometric Interpretation of Integrable Motions, *arXiv:astro-ph/*0105364, 2001.

[19] A.C. Newell, M. Tabor, Y.B. Zeng, A Unified Approach to Painlevé Expansions, *Physica D*, 1987, vol. 29, pp. 1–68.

[20] A.P. Fordy, The Hénon–Heiles System Revisited, *Physica D*, 1991, vol. 52, pp. 204–210.

[21] E.I. Timoshkova, A New Class of Trajectories of Motion in the Hénon–Heiles Potential Field, *Rus. Astron. J.*, 1999, vol. 76, pp. 470–475, {in Russian}, *Astron. Rep.*, 1999, vol. 43, p. 406, {in English}.

[22] V.V. Golubev, Lectures on Analytical Theory of Differential Equations, Gostekhizdat, Moscow–Leningrad, 1950, {in Russian}.

[23] E. Hille, Ordinary Differential Equations on the Complex Plane, Wiley, New York, 1976.

[24] P. Painlevé, Leçons sur la théorie analytique des eqúations différentelles (Leçons de Stockholm, 1895), Paris, 1896; Mémoire sur les eqúations différentelles dont l'intégrale générale est uniforme, *Bull Soc. Math. France*, 1900, vol. 28, pp. 201–261; Sur les eqúations différentelles du second order d'ordre supérieur dont l'intégrale générale est uniforme, *Acta Mathematica*, 1902, vol. 25, pp. 1–85; reprinted in: Oeuvres de P. Painlevé, vol. 1, ed. du CNRS, Paris, 1973.

[25] S.V. Kovalevskaya, Sur le problème de la rotation d'un corps solide autour d'un point fixe, *Acta Mathematica*, 1889, vol. 12, pp. 177–232; Sur une properiété du sustème d'équations différentielles qui définit la rotation d'un corps solide autour d'un point fixe, *Acta Mathematica*, 1890, vol. 14, pp. 81–93, {in French}; Scientific Works, Moscow, 1948, {in Russian}.

[26] V.V. Golubev, Lectures on the Integration of the Equation of Motion of a Rigid Body, State Pub. House, Moscow, 1953, {in Russian}.

[27] T. Bountis, H. Segur, F. Vivaldi, Integrable Hamiltonian Systems and the Painlevé Property, *Physical Review A*, 1982, vol. 25, pp. 1257–1264.

[28] A. Ramani, B. Grammaticos, T. Bountis, The Painlevé Property and Singularity Analysis of Integrable and Nonintegrable Systems, *Physics Reports*, 1989, vol. 180, pp. 159–245.

[29] H. Yoshida, Necessary Conditions for the Existence of Algebraic First Integrals. I & II, *Celestial Mechanics*, 1983, vol. 31, pp. 363–379; pp. 381–399.

[30] H. Yoshida, A new Necessary Condition for the Integrability of Hamiltonian Systems with a Two-dimensional Homogeneous Potential, *Physica D*, 1999, vol. 128, pp. 53–69.

[31] N. Ercolani, E.D. Siggia, Painlevé Property and Integrability, *Phys. Lett. A*, 1986, vol. 119, pp. 112–116.

[32] N. Ercolani, E.D. Siggia, Painlevé Property and Geometry, *Physica D*, 1989, vol. 34, pp. 303–346.

[33] M.J. Ablowitz, A. Ramani, H. Segur, A Connection between Nonlinear Evolution Equations and Ordinary Differential Equations of P-type. I & II, *J. Math. Phys.*, 1980, vol. 21, pp. 715–721; pp. 1006–1015.

[34] J. Weiss, M. Tabor, G. Carnevale, The Painlevé Property for Partial Differential Equations, *J. Math. Phys.*, 1983, vol. 24, pp. 522–526.

[35] J Weiss, The Painlevé Property for Partial Differential Equations. II: Bäcklund Transformation, Lax Pairs and the Schwarzian derivative, *J. Math. Phys.*, 1983, vol. 24, pp. 1405–1413.

[36] R. Conte, The Painlevé Analysis of nonlinear PDE and related topics: a computer algebra program, preprint, 1988; Invariant Painlevé Analysis of Partial Differential Equations, *Phys. Lett. A*, 1989, vol. 140, pp. 383–390.

[37] M. Tabor, Chaos and Integrability in Nonlinear Dynamics, 1989, Wiles, New York, {in English}, 2001, URSS, Moscow, {in Russian}.

[38] R. Conte, A.P. Fordy, A. Pickering, A Perturbative Painlevé Approach to Nonlinear Differential Equations, *Physica D*, 1993, vol. 69, pp. 33–58.

[39] A.C. Hearn, REDUCE.*User's Manual. Vers. 3.6*, RAND Publ. CP78, 1995.

[40] V.F. Edneral, A.P. Kryukov, A.Ya. Rodionov, *The Language for Algebraic Computation REDUCE*, Moscow, MSU Publ. House, 1989, {in Russian}.

[41] Ed. by H. Bateman, A. Erdelyi, *Higher Transcendental Functions. Vol. 3*, MC Graw-Hill Book Company, New York, Toronto, London, 1955, {in English}, "Nauka", Moscow, 1967, {in Russian}.

[42] I.A. Gerasimov, Weierstrass functions and these Applications in Mechanics and Astronomy, MSU Publ. House, Moscow, 1990, {in Russian}.

Extracting a Special Class of Integrable Systems with the Birkhoff-Gustavson Normalization of Polynomial Hamiltonians

Sergue Vinitsky, Vitaly Rostovtsev, Alexander Gusev

Joint Institute for Nuclear Research,
Dubna 141980 Moscow Region, Russia

A new procedure of extracting a special class of polynomial Hamiltonians under Bertrand - Darboux integrability condition using the ordinary Birkhoff - Gustavson normalization is proposed. Typical examples corresponding to the integral of motion contained only quadratic form of momenta are examined with help of this procedure implemented as a symbolic computer algebra algorithm on REDUCE 3.6.

Thanks to RFBR for supporting by Grant No. 02-02-16716.

1 Introduction

The ordinary and inverse Birkhoff-Gustavson (BG) normalization procedures have been applied very effectively in studies of the polynomial Hamiltonian systems [1-4]. Recently, for example, the inverse BG normalization procedure applied to the perturbed harmonic oscillators with homogeneous-cubic polynomial potentials (PHOCPs) and to those with homogeneous-quartic polynomial potentials (PHOQPs) has succeeded to find a new deep relation the Bertrand-Darboux integrability condition (BDIC) for PHOCPs and that for PHOQPs [4]. Further, in [4], the validity of the same relation is conjectured as above between the BDIC of the perturbed harmonic oscillators with homogeneous-polynomial potentials of degree-k (k-PHOs) and that of $2(k-1)$-PHOs. In this paper a variant of the above conjecture [4] is posed and the proved to be true. The proof is accomplished by analyzing the vanishing condition for the BGNF of a given PHO. The condition is described as a system of algebraic equations of the coefficients in the given polynomial potential. Since both the BG normalization and the analysis of the vanish condition fit computer algebra, a procedure for this purpose is newly implemented on REDUCE 3.6. As a simple but an interesting application the relation conjectured in [4] between the BDIC for k-PHOs and that for $2(k-1)$-PHOs is proved to be true at least in the case of $k=5$ besides $k=3$. Integrals of motion are calculated for those PHOs, too.

2 The Ordinary Normalization Problem

In this section, we review the ordinary and inverse problems of the BGNF expansion very briefly following[1]. Let us consider the Hamiltonian systems with n degrees of freedom on the phase space $\Re^n \times \Re^n$, which admits a stable equilibrium point in a resonance of equal frequencies. Without a loss of generality, such an equilibrium point can be put at the origin of the phase space. So the Hamiltonian of such a system is assumed to be expanded into a formal power series, up to degree-2ρ, it takes the polynomial in (q,p) of the form

$$K(q,p) = \frac{1}{2}\sum_{j=1}^{n}(p_j^2+q_j^2) + \sum_{k=3}^{2\rho} K_k(q,p), \tag{1}$$

where $K_k(q,p)$ denotes k $(k=3,4,\cdots)$ the homogeneous part of degree-k.

The ordinary normalization problem is the conversion of a given K into a polynomial (i.e. power series) $G(\xi,\eta)$ BGNF through a local canonical transformation, $(q,p) \to (\xi,\eta)$, which is associated with generating function of second type,

$$W(q,\eta) = \sum_{j=1}^{n} q_j\eta_j + \sum_{k=3}^{2\rho} W_k(q,\eta) \tag{2}$$

on writing down $G(\xi,\eta)$ as

$$G(\xi,\eta) = \frac{1}{2}\sum_{j=1}^{n}(\eta_j^2+\xi_j^2) + \sum_{k=3}^{2\rho} G_k(\xi,\eta), \tag{3}$$

in the degree-2ρ case, $G(\xi,\eta)$ is said to be in BGNF if every homogeneous part $G_k(\xi,\eta)$ $(k=3,4,\ldots,2\rho)$ in $G(\xi,\eta)$ satisfies the Poisson-commuting relation,

$$\left\{\frac{1}{2}\sum_{j=1}^{n}(\eta_j^2+\xi_j^2), G_k(\xi,\eta)\right\} = \left(\sum_{j=1}^{n}\left(\xi_j\frac{\partial}{\partial\eta_j} - \eta_j\frac{\partial}{\partial\xi_j}\right)\right)G_k(\xi,\eta) = 0. \tag{4}$$

Indeed such as conversion $G(\xi,\eta)$ is determined uniquely through

$$G(\frac{\partial W}{\partial\eta},\eta) = K(q,\frac{\partial W}{\partial q}), \tag{5}$$

together with $W(q,\eta)$ [4].

3 A background of the algorithm

Theorem 1.(Bertrand-Darboux) The hamiltonian

$$K(q,p) = \frac{1}{2}(p_1^2+p_2^2) + V, \quad V = V_2 + V'(q_1,q_2), \quad V_2 = \frac{1}{2}(q_1^2+q_2^2) \tag{6}$$

[1]Those who are interested in the inverse problem, see[4]

has the quadratic-velocity integral of motion

$$I = (aq_2^2 + bq_2 + c)p_1^2 + (aq_1^2 + b'q_1 + c')p_2^2 \\ +(-2aq_1q_2 - bq_1 - b'q_2 + c_1)p_1p_2 + K'(q_1, q_2), \tag{7}$$

if the potential correction V' satisfy BDIC

$$(\frac{\partial^2 V'}{\partial q_2^2} - \frac{\partial^2 V'}{\partial q_1^2})(-2aq_1q_2 - b'q_2 - bq_1 + c_1) + \frac{\partial V'}{\partial q_1}(6aq_2 + 3b) \tag{8}$$

$$+2\frac{\partial^2 V'}{\partial q_1 q_2}(a(q_2^2 - q_1^2) + bq_2 - b'q_1 + c - c') + \frac{\partial V'}{\partial q_2}(-6aq_1 - 3b') = 0,$$

where a, b, b', c, c', c_1 are real-valued constants.

Non-velocity part of integral K' meet the conditions

$$\frac{\partial K'}{\partial q_1} = 2(aq_2^2 + bq_2 + c)\frac{\partial V}{\partial q_1} + (-2aq_1q_2 - bq_1 - b'q_2 + c_1)\frac{\partial V}{\partial q_2}, \tag{9}$$

$$\frac{\partial K'}{\partial q_2} = 2(aq_1^2 + b'q_1 + c')\frac{\partial V}{\partial q_2} + (-2aq_1q_2 - bq_1 - b'q_2 + c_1)\frac{\partial V}{\partial q_1}.$$

Assertion 1. Let $K_{n,M-1}(q,p)$ and $K_M(q,p)$ be the perturbed harmonic oscillator Hamiltonians of the form

$$K_{n,M-1}(q,p) = K_2(q,p) + V_n(q) + \sum_{m=1}^{M-1} V_{(2n-4)m+2}(q), \tag{10}$$

and

$$K_M(q,p) = K_2(q,p) + V_{(2n-4)M+2}(q),$$

with $n = 3, 5, 7, \cdots$ and $M = 1, 2, 3, \cdots$, where

$$K_2(q,p) = \frac{1}{2}(q_1^2 + q_2^2 + p_1^2 + p_2^2),$$

and $V_k(q)$ for every k is a homogeneous polynomial in q of degree-k. The perturbed oscillators with the Hamiltonian $K_{n,M-1}(q,p)$ and with $K_M(q,p)$ share the same Birkhoff-Gustavson normal form (BGNF) up to degree-$(2n-4)M+2$ if both oscillators satisfy the Bertrand-Darboux integrability condition. The proof can be done from the following assertion.

Assertion 2. If the hamiltonian $K_{n,M}(q,p)$ has normal form

$$G_{n,M}(\xi,\eta) = G_2(\xi,\eta) + \sum_{m=1}^{M} G_{(2n-4)m+2}(\xi,\eta), \tag{11}$$

$$G_2(\xi,\eta) = \frac{1}{2}(\xi_1^2 + \xi_2^2 + \eta_1^2 + \eta_2^2)$$

such that $G_{n,M}(\xi,\eta) - G_2(\xi,\eta) = 0$ then oscillators with polynomial hamiltonians $K_{n,M-1}(q,p)$ and $K_M(q,p)$ are satisfy the BDIC. Assertion 2 follows from the next theorem.

Theorem 2. If polynomial hamiltonian

$$K(p,q) = K_2(p,q) + \sum_m V_m(q), \qquad K_2(p,q) = \frac{1}{2}\sum_{j=1}^{n}(p_j^2 + q_j^2) \tag{12}$$

admits an integral of motion in a polynomial form

$$I(p,q) = I_2(p,q) + \sum_m I_m(q), \qquad I_2(p,q) = \sum_{j,k=1}^{n}(a_{jk}p_jp_k + a'_{jk}q_jq_k),$$

where a_{jk} are real-valued parameters, then so does the polynomial hamiltonian

$$K_*(p,q) = K_2(p,q) + \sum_m a_m V_m(q),$$

for the same parameters a_m in K the integral of motion

$$I_*(p,q) = I_2(p,q) + \sum_m a_m I_m(q).$$

Proof. Since $V_m(q)$ and $I_m(q)$ are independent of p, then Poisson bracket of K and I results in

$$[K(p,q), I(p,q)] = [K_2(p,q), I_2(p,q)] + \sum_m([V_m(q), I_2(p,q)] + [K_2(p,q), I_m(q)]).$$

Hence under condition $[K(p,q), I(p,q)] = 0$, we have

$$[K_2(p,q), I_2(p,q)] = 0, \qquad [V_m(q), I_2(p,q)] + [K_2(p,q), I_m(q)] = 0, \tag{13}$$

by equating each homogeneous part in (2ρ) in $[K, I]$. The equation (13) leads the vanishment of the Poisson bracket,

$$[K_*(p,q), I_*(p,q)] = [K_2(p,q), I_2(p,q)] + \sum_m a_m([V_m(q), I_2(p,q)] + [K_2(p,q), I_m(q)])$$

which proves own assertion.

4 Algorithm

The procedure of extracting the integrable class of the polynomial Hamiltonians under consideration is realized by the following algorithm implemented by REDUCE 3.6:

1. For polynomial Hamiltonian $K_{n,M}(q,p)$ (10), where $V_k = \sum_{j=0}^{k} c_{kj} q_1^j q_2^{k-j}$ we find BGNF $G_{n,M}(\xi,\eta)$ (11) with help of GITA program [6].

2. We put $G_{(2k-4)m+2}(\xi,\eta)$ for $m = 1$ term of normal form (11) equal zero.

3. Using standard procedure **coeff** we have the list of equations for real-value parameters $c_{(2k-4)m+2j'}$ (each equation contains one of these parameters in first order) and c_{kj} $(0 \le j \le k,\ 0 \le j' \le (2k-4)m+2)$. After that we exclude the identical equations.

4.For each derived equations with help of the standard procedure **part** we extract such coefficient $c_{(2k-4)m+2j'}$, which is contained in this(the) equation (under consideration) and express $c_{(2k-4)m+2j'}$ via coefficients c_{kj}.

If any coefficient $c_{(2k-4)m+2j'}$ is kept in few equations, then we chose one of these equations and by subtraction of the chosen equation from the above equations we arrive to requirements, i.e., equations which pose some constrain conditions on these coefficients c_{kj}(in other words equations keeping only such set of coefficients c_{kj} which satisfies to these equations).

5. We make substitution of $c_{(2k-4)m+2j'}$ expressed via c_{kj} which has been received on previous step in the normal form and in the initial Hamiltonian.

6. We realize steps 2.-5. at $m = 2...M$.

7. We solve a system of equations, composed from the requirement equations between the set of coefficients c_{kj} and for each of the founded solutions of such system, we calculate a set coefficients $c_{(2k-4)m+2j'}$ $(m = 2...M)$versus the above mentioned set of coefficients c_{kj}.

Further we calculate such real-valued parametersa, b, b', c, c', c_1 for which BDIC (8) keeps for each of potentials $V' = V_k$ and $V' = V_{(2k-4)m+2}$, separately .

Then we calculate non-velocity part of integrals $K' = I_k$ and $K' = I_{(2k-4)m+2}$ from (9) and corresponding integrals of motion I from (7).

At that we have, that quadratic part of integrals of the motion for different potentials (but definite solution of the system of equations described on this step 7.) can be reduced to the same form. Therefore, Hamiltonian (10)coefficients of which satisfy to the all above derived conditions has a integral in the form $I = I_2 + I_k + \sum_{m=1}^{M} I_{(2k-4)m+2}$.

5 Examples

5.1 $(n, M) = (3, 2)$

For hamiltonian

$$K_{3,2}(q,p) = \frac{1}{2}\sum_{j=1}^{2}\left(p_j^2 + q_j^2\right) + V_3 + V_4 + V_6, \quad V_k = \sum_{j=0}^{k} c_{kj} q_1^j q_2^{k-j}, \quad k \geq 3 \tag{14}$$

where c_{kj} are parameters chosen arbitrary, we find the normal form

$$G_{3,2}(\xi,\eta) = \frac{1}{2}\sum_{j=1}^{2}\left(\eta_j^2 + \xi_j^2\right) + G_3(\xi,\eta) + G_4(\xi,\eta) + G_6(\xi,\eta), \tag{15}$$

where $G_k(\xi,\eta)$ are homogeneous polynomial of degree k

$$
\begin{aligned}
&G_3(\xi,\eta)=0,\\
&G_4(\xi,\eta)=(\xi_2^2+\eta_2^2)^2(-\frac{15}{16}c_{30}^2-\frac{5}{48}c_{31}^2+\frac{3}{8}c_{40})\\
&+(\xi_2^2+\eta_2^2)(\xi_1\xi_2+\eta_1\eta_2)(-\frac{5}{4}c_{30}c_{31}-\frac{5}{12}c_{31}c_{32}+\frac{3}{8}c_{41})\\
&+(\xi_1^2+\eta_1^2)(\xi_2^2+\eta_2^2)(-\frac{7}{8}c_{30}c_{32}+\frac{1}{12}c_{31}^2-\frac{7}{8}c_{31}c_{33}+\frac{1}{12}c_{32}^2+\frac{1}{8}c_{42})\\
&+(\xi_1\xi_2+\eta_1\eta_2)^2(\frac{1}{4}c_{30}c_{32}-\frac{1}{2}c_{31}^2+\frac{1}{4}c_{31}c_{33}-\frac{1}{2}c_{32}^2+\frac{1}{4}c_{42})\\
&+(\xi_2^2+\eta_2^2)(\xi_1\xi_2+\eta_1\eta_2)(-\frac{5}{12}c_{31}c_{32}-\frac{5}{4}c_{32}c_{33}+\frac{3}{8}c_{43})\\
&+(\xi_1^2+\eta_1^2)^2(-\frac{5}{48}c_{32}^2-\frac{15}{16}c_{33}^2+\frac{3}{8}c_{44}),
\end{aligned}
\tag{16}
$$

correction $G_6(\xi,\eta)$ is presented by 6-order polynomial of coordinates ξ, η with the coefficients depended on the parameters c_{3j}, c_{4j}, c_{6j}. Under condition $G_4(\xi,\eta)=0$, we find, that the parameters c_{4k} must be depend on the parameters c_{3j}

$$
\begin{aligned}
&c_{44}=\frac{5}{18}c_{32}^2+\frac{5}{2}c_{33}^2,\quad c_{43}=\frac{10}{9}c_{31}c_{32}+\frac{10}{3}c_{32}c_{33},\\
&c_{42}=-c_{30}c_{32}+2c_{31}^2-c_{31}c_{33}+2c_{32}^2\\
&c_{41}=\frac{10}{3}c_{30}c_{31}+\frac{10}{9}c_{31}c_{32},\quad c_{40}=\frac{5}{2}c_{30}^2+\frac{5}{18}c_{31}^2.
\end{aligned}
\tag{17}
$$

Here, the parameters c_{3k} are satisfy by next relation

$$
c_{31}^2+c_{32}^2-3(c_{30}c_{32}+c_{31}c_{33})=0. \tag{18}
$$

Under condition $G_6(\xi,\eta)=0$, after substitution (17), we find no new relations between parameters c_{3j} and parameters c_{6j} dependent from c_{3j}

$$
\begin{aligned}
c_{66}=\ &(306c_{30}c_{32}^3-242c_{31}^2c_{32}^2-534c_{31}c_{32}^2c_{33}-242c_{32}^4-3780c_{32}^2c_{33}^2-11340c_{33}^4)/405,\\
c_{65}=\ &(96c_{30}c_{31}c_{32}^2+324c_{30}c_{32}^2c_{33}-312c_{31}^3c_{32}-852c_{31}^2c_{32}c_{33}\\
&-592c_{31}c_{32}^3-2196c_{31}c_{32}c_{33}^2-1788c_{32}^3c_{33}-7560c_{32}c_{33}^3)/135,\\
c_{64}=\ &(72c_{30}^2c_{32}^2-228c_{30}c_{31}^2c_{32}+468c_{30}c_{31}c_{32}c_{33}+156c_{30}c_{32}^3\\
&+270c_{30}c_{32}c_{33}^2-72c_{31}^4-336c_{31}^3c_{33}-412c_{31}^2c_{32}^2+306c_{31}^2c_{33}^2\\
&-792c_{31}c_{32}^2c_{33}+270c_{31}c_{33}^3-200c_{32}^4-1350c_{32}^2c_{33}^2)/27,\\
c_{63}=\ &(792c_{30}^2c_{31}c_{32}-540c_{30}^2c_{32}c_{33}-1944c_{30}c_{31}^3+972c_{30}c_{31}^2c_{33}\\
&+2892c_{30}c_{31}c_{32}^2-540c_{30}c_{31}c_{33}^2+972c_{30}c_{32}^2c_{33}-2732c_{31}^3c_{32}\\
&+2892c_{31}^2c_{32}c_{33}-2732c_{31}c_{32}^3+792c_{31}c_{32}c_{33}^2-1944c_{32}^3c_{33})/81,\\
c_{62}=\ &(270c_{30}^3c_{32}-1350c_{30}^2c_{31}^2+270c_{30}^2c_{31}c_{33}+306c_{30}^2c_{32}^2\\
&-792c_{30}c_{31}^2c_{32}+468c_{30}c_{31}c_{32}c_{33}-336c_{30}c_{32}^3-200c_{31}^4+156c_{31}^3c_{33}\\
&-412c_{31}^2c_{32}^2+72c_{31}^2c_{33}^2-228c_{31}c_{32}^2c_{33}-72c_{32}^4)/27,\\
c_{61}=\ &(-7560c_{30}^3c_{31}-2196c_{30}^2c_{31}c_{32}-1788c_{30}c_{31}^3+324c_{30}c_{31}^2c_{33}\\
&-852c_{30}c_{31}c_{32}^2-592c_{31}^3c_{32}+96c_{31}^2c_{32}c_{33}-312c_{31}c_{32}^3)/135,\\
c_{60}=\ &(-11340c_{30}^4-3780c_{30}^2c_{31}^2-534c_{30}c_{31}^2c_{32}-242c_{31}^4+306c_{31}^3c_{33}-242c_{31}^2c_{32}^2)/405.
\end{aligned}
\tag{19}
$$

We go back to the equation (18) to solve it. Three types of solution can be obtained

$$
1)\quad c_{31}=0,\quad c_{32}=0;\quad 2)\quad c_{31}\neq 0,\quad c_{32}=0,\quad c_{33}=c_{31}/3;
$$

$$3)\quad c_{32}\neq 0,\quad c_{30}=\frac{(c_{31}^2+c_{32}^2-3c_{31}c_{33})}{3c_{32}}. \tag{20}$$

We remark here that, in the case 1, we find that the hamiltonian (14) takes the form separated in the variables. Further the case 2 reduces to the case 3 by changing the pairs of variables $(q_1,p_1)\leftrightarrow(q_2,p_2)$ in the hamiltonian (14).

In fact, that the hamiltonian (14) in which the parameters c_{kj} are satisfied to conditions (17), (19) and 3) from (20) has the integral of motion in form

$$I_{3,2}(q,p)=(c_{31}-3c_{33})({p_1}^2+q_1^2)-2c_{32}(p_1p_2+q_1q_2)+I_3+I_4+I_6, \tag{21}$$

where $I_k=I_k(q_1,q_2)$ are k-degree homogeneous polynomial functions of coordinates.

5.2 $(n,M)=(5,1)$

Let us consider the hamiltonian $K_{5,1}(q,p)$ and finding normal form $G_{5,1}(\xi,\eta)$

$$K_{5,1}(q,p)=\frac{1}{2}\sum_{j=1}^{2}\left(p_j^2+q_j^2\right)+V_5+V_8,\quad G_{5,1}(\xi,\eta)=\frac{1}{2}\sum_{j=1}^{2}\left(\eta_j^2+\xi_j^2\right)+G_8(\xi,\eta).$$

Under condition $G_8(\xi,\eta)$=0, we find that the parameters c_{8j} are depended on c_{5j}

$$\begin{aligned}
c_{88}&=(9c_{54}^2+225c_{55}^2)/50,\quad c_{87}=(18c_{53}c_{54}+180c_{54}c_{55})/25,\\
c_{86}&=(49c_{52}c_{54}+70c_{53}^2+245c_{53}c_{55}+280c_{54}^2)/75,\\
c_{85}&=(-28c_{51}c_{54}+70c_{52}c_{53}-70c_{52}c_{55}+140c_{53}c_{54})/25,\\
c_{84}&=(-2c_{50}c_{54}-16c_{51}c_{53}-2c_{51}c_{55}+27c_{52}^2-16c_{52}c_{54}+27c_{53}^2)/6,\\
c_{83}&=(-70c_{50}c_{53}+140c_{51}c_{52}-28c_{51}c_{54}+70c_{52}c_{53})/25,\\
c_{82}&=(245c_{50}c_{52}+280c_{51}^2+49c_{51}c_{53}+70c_{52}^2)/75,\\
c_{81}&=(180c_{50}c_{51}+18c_{51}c_{52})/25,\quad c_{80}=(225c_{50}^2+9c_{51}^2)/50.
\end{aligned} \tag{22}$$

Here the parameters c_{5j} are satisfy by next conditions

$$\begin{aligned}
&-2c_{52}c_{54}+c_{53}^2-10c_{53}c_{55}+4c_{54}^2=0,\\
&-4c_{51}c_{54}+c_{52}c_{53}-10c_{52}c_{55}+2c_{53}c_{54}=0,\\
&20c_{50}c_{54}-92c_{51}c_{53}+20c_{51}c_{55}+45c_{52}^2-92c_{52}c_{54}+45c_{53}^2=0,\\
&-20c_{50}c_{54}-20c_{51}c_{53}-20c_{51}c_{55}+11c_{52}^2-20c_{52}c_{54}+11c_{53}^2=0,\\
&-10c_{50}c_{53}+2c_{51}c_{52}-4c_{51}c_{54}+c_{52}c_{53}=0,\\
&-10c_{50}c_{52}+4c_{51}^2-2c_{51}c_{53}+c_{52}^2=0.
\end{aligned} \tag{23}$$

If parameter $c_{53}\neq 0$ then the system of equations (23) has next solution

$$\begin{aligned}
c_{50}&=(c_{52}^3-4c_{52}^2c_{54}+2c_{52}c_{53}^2+4c_{52}c_{54}^2-2c_{53}^2c_{54})/(10c_{53}^2),\\
c_{51}&=(c_{52}^2-2c_{52}c_{54}+c_{53}^2)/(2c_{53}),\quad c_{55}=(-2c_{52}c_{54}+c_{53}^2+4c_{54}^2)/(10c_{53})
\end{aligned}$$

and the integral of motion takes form

$$I_{5,1}(q,p)=(c_{52}-2c_{54})({p_1}^2+{q_1}^2)-2c_{53}(q_2q_1+p_2p_1)+I_5+I_8.$$

References

[1] Y.Uwano, N.A.Chekanov, V.A.Rostovtsev and S.I.Vinitsky, *Computer Algebra in Scientific Computing*, V.Ganzha *et.al.* eds., 441–61 (Springer-Verlag, 1999).

[2] N.A.Chekanov, M.Hongo, V.A.Rostovtsev, Y.Uwano and S.I.Vinitsky, Phys. Atom. Nucl. **61**, 1918-22 (1998).

[3] N.Chekanov, V.Rostovtsev, Y.Uwano and S.Vinitsky, Comp. Phys. Commun. **126**, 47–50 (2000).

[4] Y.Uwano, J. Phys. A **33**, 6635-6653 (2000)

[5] J.K.Moser. Lectures of hamiltonian system. Memories of AMS No.81,1-60 AMS, Providence (1968).

[6] V.Basios, N.A.Chekanov, B.L.Markovski, V.A.Rostovtsev, Y.Uwano and S.I.Vinitsky, Comp. Phys. Commun. **90**, 355–368 (1995).

On Numerical Stability of Polynomial Evaluation

Eugene V. Zima
Department of Computer Science
University of Waterloo, Waterloo, Canada
e-mail: ezima@scg.uwaterloo.ca

The section "Tabulating polynomial values" in [4] gives a caution on the accumulation of rounding errors when tables of finite differences are used for the evaluation of polynomial values. However it does not indicate the accumulative error effect quantitatively. On the other hand, very pessimistic bounds for the accumulated error of this kind of polynomial evaluation are given in [1] and repeated in [2]. These bounds involve, for example, the degree of the evaluated polynomial. We will analyze different schemes for polynomial evaluation (forward and backward differencing [5]) and will show that the accumulated error does not depend on the degree of a polynomial (as in the case of Horner method [3]). This means that the numeric stability of finite differences based polynomial evaluation is much better than it was previously thought.

We will also consider examples of applications of differencing technique to such problems as 2D and 3D plotting (in particular, algebraic curves plotting) and discuss our implementation.

References

[1] O. Bachmann, Chains of recurrences, PhD thesis, Kent State Univ., Kent, OH - 44240, USA, December 1996.

[2] R. van Engelen, Symbolic Evaluation of Chains of Recurrences for Loop Optimization, Technical Report TR-000102, Department of Computer Science, Florida State University, 2000.

[3] N.J. Higham, Accuracy and stability of numerical algorithms, SIAM, 1996.

[4] D.E. Knuth, The art of computer programming. V.2. Seminumerical Algorithms, Second edition. Addison-Wesley, 1981.

[5] E.V. Zima, On computational properties of chains of recurrences. In *International Symposium on Symbolic and Algebraic Computation* (2001), London, Canada, ACM Press (to appear).

S. Vinitsky, W. Kallies, V. Gerdt, A. Prokopenya

V. Gerdt, N. Vassiliev, N. Kalinina, A. Bogolubskaya

V. Gerdt, E.Grebenikov, A. Prokopenya

S. Mechveliani, N. Vassiliev

N.Vassiliev, Yu.Blinkov, N.Glazunov

V. Severyanov, A. Bogolubskaya, A. Prokopenya, V. Edneral

G. Czichowski, V. Gerdt, K. Kölbig

R. Fedorova, M. Spiridonova

At the boat trip down Volga

V. Edneral, I. Tchoupaeva

On the bank of the Dubna river

Index

List of Participants

Sergei Abramov	CC RAS Moscow	abramov@ccas.ru
Mikhail Altaisky	JINR Dubna	altaisky@mx.iki.rssi.ru
Dmitry Bardin	JINR Dubna	bardin@nusun.jinr.ru
Matthias Berth	Univ. Greifswald	berth@mail.uni-greifswald.de
Juri Blinkov	SSU Saratov	blinkovua@info.sgu.ru
Tengiz Bochorishvili	Akad. Podlaska, Siedlce	tengiz@ap.siedlce.pl
Guenter Czichowski	Univ. Greifswald	czicho@mail.uni-greifswald.de
Owen L de Lange	Univ.of Natal, Pietermaritzburg	delange@unpsun1.cc.unp.ac.za
Victor Edneral	NPI MSU Moscow	edneral@theory.sinp.msu.ru
Georgy Efimov	IAM RAS Moscow	efimov@applmat.msk.su
Arcady Galperin	JINR Dubna	rqvtsk@cv.jinr.ru
Fangil Gareev	JINR Dubna	gareev@thsun1.jinr.ru
Vladimir Gerdt	JINR Dubna	gerdt@jinr.ru
Nikolaj Glazunov	GIC NAS Kiev	glanm@d105.icyb.kiev.ua
Oleg Golubitsky	MSU Moscow	oleg@shade.msu.ru
Vasiliy Govorukhin	RSU Rostov-on-Don	vgov@math.rsu.ru
Evgenij Grebenikov	CC RAS Moscow	greben@ccas.ru imif@ap.siedlce.pl
Andrey Grozin	BINP Novosibirsk	A.G.Grozin@inp.nsk.su
Alexander Gusev	JINR Dubna	vinitsky@thsun1.jinr.ru
Marcus Hausdorf	Univ. Mannheim	hausdorf@math.uni-mannheim.de
Valery Ivanov	JINR Dubna	vivanov@linserv.jinr.ru
Nina Kalinina	NSU Novosibirsk	kalinina@ccmath.nsu.ru
Arsen Khvedelidze	JINR Dubna	khved@thsun1.jinr.ru
Vladimir Kislenkov	MSU Moscow	kvv@cs.msu.su
Kurt Kolbig	CERN	ksk@mail.cern.ch
Ekaterina Komarova	PFUR Moscow	evkomarova@mail.ru
Marina Kondratieva	MSU Moscow	kondra_m@shade.msu.ru
Vladimir Kornyak	JINR Dubna	kornyak@jinr.ru
Natalia Makarevich	MSU Moscow	makarevich_n@mail.ru
Serge Mechveliani	PSI Pereslavl-Zalessky	mechvel@botik.ru
Alla Mitichkina	MSU Moscow	allam@dis.muh.ru
Vasiliy Mitrofanov	MSU Moscow	mitrofan@cmc.msu.ru
Vladimir Mityunin	MSU Moscow	vmit@metric.ru
Dimitar Mladenov	JINR Dubna	mladim@thsun1.jinr.ru
Arthur Niukkanen	GEOKHI RAS Moscow	niukkanen@tula.net, elkor@geokhi.msk.su
Alexey Ovchinnikov	MSU Moscow	leha@shade.msu.ru
Evgueny Pankratiev	MSU Moscow	pankrat@shade.msu.ru
Olga Paramonova	GEOKHI RAS Moscow	oparamonova@mail.ru
Marko Petkovšek	Univ. Ljubljana	marko.petkovsek@fmf.uni-lj.si
Victor Pervushin	JINR Dubna	pervush@thsun1.jinr.ru
Alexander Prokopenya	BTU Brest Belarus	prokopenya@belpak.brest.by
Denis Proskurin	JINR Dubna	proskur@thsun1.jinr.ru

Victor Richvitsky	JINR Dubna	rqvtsk@cv.jinr.ru
Vitaly Rostovtsev	JINR Dubna	rost@jinr.ru
Anna Ryabenko	CC RAS Moscow	ryabenko@cs.msu.su
Alexander Sapoznikov	JINR Dubna	sap@cv.jinr.ru
Alexander Semyonov	MSU Moscow	semyonov@mccme.ru
Svetlana Serdyukova	JINR Dubna	sis@cv.jinr.ru
Vasily Shapeev	ITAM SB RAS Novosibirsk	shapeev@itam.nsc.ru
Polina Sheveleva	MSU Moscow	psheveleva@rambler.ru
Igor Shevtsov	MSOU Moscow	voyage@mail.ru
Stanislav Sobolevsky	ICE Minsk	sobol@im.bas-net.by
Margarita Spiridonova	IMI BAS Sofia	mspirid@math.bas.bg
Alina Suzko	JINR Dubna	suzko@thsun1.jinr.ru
Sergey Tertychniy	VNIIFTRI Mendeleevo	bpt97@mail.ru
Irina Tchoupaeva	MSU Moscow	ladyirina@shade.msu.ru
Nikolay Vassiliev	MI RAS St.Petersburg	vasiliev@pdmiras.ru
Sergey Vernov	NPI MSU Moscow	svernov@theory.npi.msu.su
Sergue Vinitsky	JINR Dubna	vinitsky@thsun1.jinr.ru
Denis Yanovich	JINR Dubna	yan@jinr.ru
Eugene Zima	SCG Waterloo	ezima@scg.uwaterloo.ca
Alexey Zobnin	MSU Moscow	zobnin@shade.msu.ru

Научное издание

Компьютерная алгебра и ее приложения в физике

Международное рабочее совещание

Труды совещания

Computer Algebra and its Application to Physics

International Workshop

Proceedings

E5,11-2001-279

Сборник отпечатан методом прямого репродуцирования
с оригиналов, предоставленных оргкомитетом.

Ответственные за подготовку сборника к печати *А. А. Боголюбская, Ю. Г. Палий.*

Фотографии *Ю. А. Туманова, В. Ф. Еднерала.*

ЛР № 020579 от 23.06.97.
Подписано в печать 28.02.2002.
Формат 60×90/16. Бумага офсетная. Печать офсетная.
Усл. печ. л. 22,5. Уч.-изд. л. 37,26. Тираж 130 экз. Заказ № 53150.

Издательский отдел Объединенного института ядерных исследований
141980, г. Дубна, Московская обл., ул. Жолио-Кюри, 6.