

СОДЕРЖАНИЕ

Номер 2, 2021

КОМПЬЮТЕРНАЯ АЛГЕБРА

Семинар по компьютерной алгебре в 2019–2020 гг.

С. А. Абрамов, А. А. Боголюбская 3

Инвариантные координатные подпространства нормальной формы системы обыкновенных дифференциальных уравнений

А. Б. Батхин 5

Факторизация булевых полиномов: параллельные алгоритмы и экспериментальная оценка

*П. Г. Емельянов, М. Кришна, В. Кулкарни,
С. К. Нанди, Д. К. Пономарев, С. Раха* 15

Символьно-аналитические методы исследования положений равновесия спутника на круговой орбите

С. А. Гутник, В. А. Сарычев 28

Моделирование динамики квантовой запутанности в конечной квантовой механике: компьютерно-алгебраический подход

В. В. Корняк 34

Компьютерная алгебра на Julia

Д. С. Кулябов, А. В. Королькова 44

Алгоритм построения результата двух целых функций

В. И. Кузоватов, А. А. Кытманов, Е. К. Мышкина 51

Процедуры поиска усеченных решений линейных дифференциальных уравнений с бесконечными и усеченными степенными рядами в роли коэффициентов

С. А. Абрамов, А. А. Рябенко, Д. Е. Хмельнов 56

Редакционные правила для авторов журнала “Программирование”

66

Авторский указатель статей, опубликованных в 2020 году

70

CONTENTS

No. 2, 2021

Seminar on Computer Algebra in 2019–2020 <i>S. A. Abramov, A. A. Bogolyubskaya</i>	3
Invariant Coordinate Subspaces of Normal Form of a System of Ordinary Differential Equations <i>A. B. Batkhin</i>	5
Factorization of Boolean Polynomials: Parallel Algorithms and Experimental Estimation <i>P. G. Emelyanov, M. Krishna, V. Kulkarny, S. K. Nandy, S. Raha</i>	15
Symbolic Analytic Methods for Studying of Equilibrium Orientations of a Satellite in a Circular Orbit <i>S. A. Gutnik, V. A. Sarychev</i>	27
Modeling the Dynamics of Quantum Entanglement in Finite Quantum Mechanics: a Computer-Algebraic Approach <i>V. V. Korniyak</i>	32
Computer Algebra on Julia <i>D. S. Kulyabov, A. V. Korolkova</i>	42
Algorithm for Constructing the Resultant of Two Entire Functions <i>V. I. Kuzovatov, A. A. Kytmanov, E. V. Myshkina</i>	49
Search Procedures for Truncated Solutions of Linear Differential Equations with Infinite and Truncated Power Series as Coefficients <i>S. A. Abramov, A. A. Ryabenko, D. E. Khmel'nov</i>	54
Author's index of articles published in 2020	66
Editorial guidelines for authors	70

КОМПЬЮТЕРНАЯ АЛГЕБРА

УДК 004.421.6

СЕМИНАР ПО КОМПЬЮТЕРНОЙ АЛГЕБРЕ В 2019–2020 гг.

© 2021 г. С. А. Абрамов^{а,*}, А. А. Боголюбская^{б,**}

^а ВЦ ФИЦ ИУ РАН, Москва, ул. Вавилова, 40, Россия

^б Объединенный институт ядерных исследований, Дубна Московской области, Россия

* E-mail: sergeyabramov@mail.ru

** E-mail: abogol@jinr.ru

Поступила в редакцию 01.07.2020 г.

После доработки 01.07.2020 г.

Принята к публикации 02.09.2020 г.

Годовой отчет о работе научно-исследовательского семинара по компьютерной алгебре.

DOI: 10.31857/S0132347421020023

1. О СЕМИНАРЕ

В семинаре рассматриваются новые результаты в области компьютерной алгебры — символьные алгоритмы и их реализация, соответствующие вопросы системного программирования.

В 2019–2020 учебном году семинар собирался раз в месяц по третьим средам на факультете вычислительной математики и кибернетики МГУ, а майское заседание семинара прошло в режиме видеоконференции.

2. РЕГУЛЯРНЫЕ СОБРАНИЯ СЕМИНАРА

С сентября по май были прочитаны следующие доклады¹. Аннотации доступны на странице семинара <http://www.ccas.ru/sabramov/seminar/doku.php>, где также содержится информация о состоявшихся ранее докладах.

А.В. Селиверстов (Институт проблем передачи информации им. А.А. Харкевича Российской академии наук; slvstv@iitp.ru) *Матрицы Гессе приводимых многочленов третьей степени.*

Заседание семинара было посвящено памяти С.Н. Перепечко.

Д. Штефанеску (Бухарестский университет, Румыния; doru.stefanescu@gmail.com) *Вычислительные аспекты теории полиномов.*

С.А. Гутник (Московский физико-технический институт (национальный исследовательский университет); sergey.gutnik@gmail.com) *Динамика движения спутника относительно центра масс с пассивными системами ориентации.*

С.Ф. Адлай (SemjonAdlaj@gmail.com) *Механическая интерпретация и эффективное вычисление эллиптических интегралов третьего рода.*

А.И. Овчинников (Городской университет Нью-Йорка; alexey_ov@yahoo.com) *Определенность параметров в ОДУ при помощи дифференциальной алгебры.*

А.А. Михалёв (Механико-математический факультет МГУ имени М.В. Ломоносова; mikhailev@shade.msu.ru) *Комбинаторика слов и стандартные базисы идеалов свободных алгебр.*

А.И. Зобнин (факультет компьютерных наук НИУ ВШЭ; azobnin@hse.ru) *Линейная алгебра в задачах векторного представления слов.*

А.Е. Панкратьев (Механико-математический факультет МГУ имени М.В. Ломоносова; ap-ton.pankratiev@gmail.com) *О сложности проверки полиномиальной полноты конечных квазигрупп.*

А.П. Крюков, Г.Б. Шпиз (Научно-исследовательский институт ядерной физики им. Д.В. Скобельцына МГУ; kryukov@theory.sinp.msu.ru) *Проблема упрощения алгебраических выражений с индексами в компьютерной алгебре.*

СПИСОК ЛИТЕРАТУРЫ

1. Абрамов С.А., Зима Е.В. Семинар по компьютерной алгебре на факультете вычислительной математики и кибернетики МГУ в 1995–1996 гг. // Программирование. 1997. № 1. С. 75–77.
2. Абрамов С.А., Зима Е.В. Научно-исследовательский семинар “Компьютерная алгебра” в 1996–1997 гг. // Программирование. 1998. № 1. С. 69–72.
3. Абрамов С.А., Ростовцев В.А. Семинар по компьютерной алгебре в 1997–1998 гг. // Программирование. 1998. № 6. С. 3–7.

¹ Перечень докладов, прочитанных в 1995–2019 гг., опубликован в [1]–[25].

4. *Абрамов С.А., Крюков А.П., Ростовцев В.А.* Семинар по компьютерной алгебре в 1998–1999 гг. // Программирование. 2000. № 1. С. 8–12.
5. *Абрамов С.А., Крюков А.П., Ростовцев В.А.* Семинар по компьютерной алгебре в 1999–2000 гг. // Программирование. 2001. № 1. С. 3–7.
6. *Абрамов С.А., Крюков А.П., Ростовцев В.А.* Семинар по компьютерной алгебре в 2000–2001 г. // Программирование. 2002. № 2. С. 6–9.
7. *Абрамов С.А., Крюков А.П., Ростовцев В.А.* Семинар по компьютерной алгебре в 2001–2002 гг. // Программирование. 2003. № 2. С. 3–7.
8. *Абрамов С.А., Еднерал В.Ф., Ростовцев В.А.* Семинар по компьютерной алгебре в 2002–2003 гг. // Программирование. 2004. № 2. С. 3–7.
9. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А., Еднерал В.Ф.* Семинар по компьютерной алгебре в 2003–2004 гг. // Программирование. 2005. № 2. С. 3–9.
10. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А., Еднерал В.Ф.* Семинар по компьютерной алгебре в 2004–2005 гг. // Программирование. 2006. № 2. С. 3–7.
11. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А., Еднерал В.Ф.* Семинар по компьютерной алгебре в 2005–2006 гг. // Программирование. 2007. № 2. С. 3–8.
12. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А., Еднерал В.Ф.* Семинар по компьютерной алгебре в 2006–2007 гг. // Программирование. 2008. № 2. С. 3–8.
13. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А., Еднерал В.Ф.* Семинар по компьютерной алгебре в 2007–2008 гг. // Программирование. 2009. № 2. С. 3–9.
14. “Mathematical Modeling and Computational Physics (CAAP’2009)”. Book of abstracts of the international conference. Dubna, July 7–11, 2009. Dubna, 2009.
15. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А., Еднерал В.Ф.* Семинар по компьютерной алгебре в 2008–2009 гг. // Программирование. 2010. № 2. С. 3–8.
16. *Абрамов С.А., Боголюбская А.А., Еднерал В.Ф., Ростовцев В.А.* Семинар по компьютерной алгебре в 2009–2010 гг. // Программирование. 2011. № 1. С. 3–8.
17. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А.* Семинар по компьютерной алгебре в 2010–2011 гг. // Программирование. 2012. № 2. С. 3–8.
18. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А.* Семинар по компьютерной алгебре в 2011–2012 гг. // Программирование. 2013. № 2. С. 3–10.
19. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А.* Семинар по компьютерной алгебре в 2012–2013 гг. // Программирование. 2014. № 2. С. 3–11.
20. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А.* Семинар по компьютерной алгебре в 2013–2014 гг. // Программирование. 2015. № 2. С. 3–6.
21. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А.* Семинар по компьютерной алгебре в 2014–2015 гг. // Программирование. 2016. № 2. С. 4–7.
22. *Абрамов С.А., Боголюбская А.А.* Семинар по компьютерной алгебре в 2015–2016 гг. // Программирование. 2017. № 2. С. 3–6.
23. *Абрамов С.А., Боголюбская А.А.* Семинар по компьютерной алгебре в 2016–2017 гг. // Программирование. 2018. № 2. С. 3–4.
24. *Абрамов С.А., Боголюбская А.А.* Семинар по компьютерной алгебре в 2017–2018 гг. // Программирование. 2019. № 2. С. 3–5.
25. *Абрамов С.А., Боголюбская А.А.* Семинар по компьютерной алгебре в 2018–2019 гг. // Программирование. 2020. № 2. С. 3–5.

УДК 517.925+004.421.6

ИНВАРИАНТНЫЕ КООРДИНАТНЫЕ ПОДПРОСТРАНСТВА НОРМАЛЬНОЙ ФОРМЫ СИСТЕМЫ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

© 2021 г. А. Б. Батхин^{а,*}^а Институт прикладной математики им. М.В. Келдыша РАН,
125047 Москва, Миусская пл., д. 4, Россия

*E-mail: batkhin@gmail.com

Поступила в редакцию 15.07.2020 г.

После доработки 20.08.2020 г.

Принята к публикации 02.09.2020 г.

Для систем обыкновенных дифференциальных уравнений (ОДУ) с невырожденной линейной частью в общем и гамильтоновом случаях ставится задача отыскания инвариантных координатных подпространств в координатах ее нормальной формы, вычисленной вблизи положения равновесия. Приведены условия существования таких инвариантных подпространств в терминах резонансных соотношений между собственными числами линейной части системы. Дан алгоритм поиска резонансных соотношений между собственными числами без их явного вычисления, который существенно использует методы компьютерной алгебры и q -аналог субрезультантов многочлена. Обсуждается его реализация в трех распространенных системах компьютерной алгебры – Mathematica, Maple и SymPy. Приведены содержательные модельные примеры.

DOI: 10.31857/S0132347421020047

1. ВВЕДЕНИЕ

Подход А. Пуанкаре к исследованию систем нелинейных обыкновенных дифференциальных уравнений (ОДУ) заключался в том, чтобы посредством аналитических обратимых преобразований максимально упростить правые части системы. Этот подход привел к теории нормальных форм систем ОДУ, разработанной в работах А. Дюляка и А.Д. Брюно для общих систем (подробнее см. [1]) и в работах Дж. Биркгофа, Т.М. Черри, Ф.Г. Густавсона, К.Л. Зигеля, Ю. Мозера, А.Д. Брюно и других для систем Гамильтона (подробнее см. [2, Гл. I, II]).

Хотя нормальная форма системы ОДУ в окрестности инвариантного многообразия (положения равновесия, периодического решения, k -мерного тора) является формальным объектом, т.е. переход к нормальной форме обычно является расходящимся преобразованием, она может быть эффективно использована для исследования устойчивости соответствующего инвариантного многообразия [3], [4, Part II], локальной интегрируемости системы в его окрестности [5], поиска периодических решений, первых интегралов [6, 7], а также асимптотического интегрирования исходной системы [8].

Цель настоящей работы состоит в изучении инвариантных координатных подпространств в нормальной форме общей системы ОДУ в случае,

когда линейная часть этой системы невырождена, т.е. имеет хотя бы одно ненулевое собственное число. Существование координатного инвариантного подпространства позволяет изучать динамику фазового потока исходной системы на пространстве меньшей размерности и в некоторых случаях дает информацию о периодических решениях полной системы.

Работа состоит из введения, четырех разделов и двух списков – литературы и условных обозначений. В разделе 2 дается определение нормальной формы общей системы ОДУ и формулируется условие существования инвариантного координатного подпространства. В разделе 3 рассматривается нормальная форма системы Гамильтона и формулируется условие существования инвариантного координатного подпространства в этом случае. Раздел 4 содержит три модельных примера, демонстрирующих типичные ситуации с собственными числами линейной части системы ОДУ. Наконец, в заключительном разделе 5 приведена методика применения q -аналогов таких классических объектов, как субдискриминант многочлена и его производная, для анализа резонансных соотношений между собственными числами линейной части системы ОДУ в ситуации, когда эти собственные числа попарно рационально соизмеримы. Описанные алгоритмы могут

быть реализованы в различных системах компьютерной алгебры, таких как Wolfram Mathematica [9], Maplesoft Maple [10], SymPy [11]. Приведен пример определения резонансных соотношений для модельной колебательной системы с шестью степенями свободы.

Замечания об обозначениях

- Символы, набранные полужирным шрифтом $\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v}$, обозначают векторы-столбцы в n -мерном вещественном $\mathbb{R}^n$ или комплексном $\mathbb{C}^n$ пространствах.

- Символы, набранные полужирным шрифтом $\mathbf{p}, \mathbf{q}$, обозначают векторы в n -мерной целочисленной решетке $\mathbb{Z}^n$.

- $|\mathbf{p}| = \sum_{j=1}^n |p_j|$.

- Для $\mathbf{x} = (x_1, \dots, x_n)^T$ и $\mathbf{p} = (p_1, \dots, p_n)^T \in \mathbb{Z}^n$ величина $\mathbf{x}^{\mathbf{p}} \equiv \prod_{j=1}^n x_j^{p_j}$, а величина $\langle \mathbf{p}, \mathbf{x} \rangle \equiv \sum_{j=1}^n p_j x_j$ — скалярное произведение.

2. ИНВАРИАНТНЫЕ КООРДИНАТНЫЕ ПОДПРОСТРАНСТВА ОБЩЕЙ СИСТЕМЫ ОДУ

Рассмотрим аналитическую систему ОДУ

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \quad (2.1)$$

вблизи ее положения равновесия

$$\mathbf{x} = 0,$$

совпадающего с началом координат.

Пусть линейная часть

$$\dot{\mathbf{x}} = A\mathbf{x}, \quad A = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=0}, \quad (2.2)$$

системы (2.1) невырождена. Тогда матрица A имеет n собственных чисел

$$\lambda = (\lambda_1, \dots, \lambda_n),$$

из которых по крайней мере одно ненулевое.

Согласно теореме I из [1] существует формальное обратимое преобразование $\mathbf{g} : \mathbf{x} \rightarrow \mathbf{y}$

$$\mathbf{x} = \mathbf{g}(\mathbf{y}),$$

представленное в виде степенных рядов, которое приводит исходную систему (2.1) к ее *нормальной форме*.

Определение 1 ([1]). *Нормальная форма* исходной системы (2.1) это система ОДУ в виде

$$\dot{y}_j = y_j h_j(\mathbf{y}), \quad j = 1, \dots, n, \quad (2.3)$$

правые части $y_j h_j(\mathbf{y})$ которой суть степенные ряды

$$y_j h_j(\mathbf{y}) = y_j \sum_q h_{jq} \mathbf{y}^q, \quad h_{j0} = \lambda_j, \quad j = 1, \dots, n, \quad (2.4)$$

состоящие только из *резонансных членов*, показатели степени которых удовлетворяют резонансному уравнению

$$\langle \mathbf{q}, \lambda \rangle = 0. \quad (2.5)$$

Здесь h_{jq} суть постоянные коэффициенты, при этом в слагаемых $y_j h_j(\mathbf{y})$ целочисленные координаты $q_j \geq -1$, но другие координаты $q_k \geq 0$.

Введем кортеж $I = \{i_1, \dots, i_k\}$ возрастающих индексов $1 \leq i_1, i_k \leq n$, $k \leq n$. Обозначим через K_I *координатное подпространство*

$$K_I = \{\mathbf{y} : y_j = 0 \text{ для всех } j \notin I\}.$$

Все ненулевые координаты $y_j, j \in I$, подпространства K_I назовем *внутренними координатами* и обозначим их для краткости $\mathbf{y}_I$. Остальные координаты назовем *внешними*.

Аналогично, собственные числа $\lambda_j, j \in I$, соответствующие внутренним координатам $\mathbf{y}_I$, назовем *внутренними собственными числами* и обозначим их λ_I . Остальные собственные числа $\lambda_j, j \notin I$, назовем *внешними*.

Рассмотрим следующую задачу.

Задача 1. Какие координатные подпространства K_I инвариантны в нормальной форме (2.3), (2.4), (2.5)?

Решение задачи 1 дает следующая теорема.

Теорема 1. *Координатное подпространство K_I размерности k инвариантно в нормальной форме (2.3)–(2.5), если каждое внешнее собственное число $\lambda_j \notin \lambda_I$ удовлетворяет условию*

$$\lambda_j \neq \langle \mathbf{p}, \lambda_I \rangle, \quad (2.6)$$

для всех неотрицательных целочисленных векторов $\mathbf{p} \geq 0$, $\mathbf{p} \in \mathbb{Z}^k$.

Доказательство. Из условия (2.6) следует, что каждый ряд $h_j(\mathbf{y})$ для $j \notin I$ не содержит ни одного слагаемого $h_{jq} \mathbf{y}^q$, чьи индексы $q_j = -1$, $q_i \geq 0$, $i \neq j \notin I$. Поскольку внешние переменные $y_j, j \notin I$, равны нулю на подпространстве K_I , то из этого следует, что для $\mathbf{y} \in K_I$

$$y_j h_j(\mathbf{y}) = 0, \text{ для всех индексов } j \notin I.$$

Таким образом, подпространство K_I инвариантно в нормальной форме (2.3)–(2.5).

3. ИНВАРИАНТНЫЕ КООРДИНАТНЫЕ ПОДПРОСТРАНСТВА СИСТЕМЫ ГАМИЛЬТОНА

Рассмотрим аналитическую систему Гамильтона

$$\dot{\mathbf{x}} = \frac{\partial H}{\partial \mathbf{y}}, \quad \dot{\mathbf{y}} = -\frac{\partial H}{\partial \mathbf{x}} \quad (3.1)$$

с n степенями свободы вблизи положения равновесия

$$\mathbf{x} = \mathbf{y} = 0.$$

Функция Гамильтона $H(\mathbf{x}, \mathbf{y})$ раскладывается в сходящийся степенной ряд

$$H(\mathbf{x}, \mathbf{y}) = \sum H_{\mathbf{pq}} \mathbf{x}^{\mathbf{p}} \mathbf{y}^{\mathbf{q}} \quad (3.2)$$

с постоянными коэффициентами $H_{\mathbf{pq}}$, $\mathbf{p}, \mathbf{q} \geq 0$, $|\mathbf{p}| + |\mathbf{q}| \geq 2$. Каноническое преобразование координат $\mathbf{x}, \mathbf{y}$

$$\mathbf{x} = \mathbf{f}(\mathbf{u}, \mathbf{v}), \quad \mathbf{y} = \mathbf{g}(\mathbf{u}, \mathbf{v}), \quad (3.3)$$

сохраняет гамильтонову структуру исходной системы (3.1).

Введем фазовый вектор $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{2n}(\mathbb{C}^{2n})$. Тогда линейная часть системы (3.1) может быть записана в виде

$$\dot{\mathbf{z}} = B\mathbf{z}, \quad B = \frac{1}{2} \begin{pmatrix} \frac{\partial^2 H}{\partial \mathbf{y} \partial \mathbf{x}} & \frac{\partial^2 H}{\partial \mathbf{y} \partial \mathbf{y}} \\ -\frac{\partial^2 H}{\partial \mathbf{x} \partial \mathbf{x}} & -\frac{\partial^2 H}{\partial \mathbf{x} \partial \mathbf{y}} \end{pmatrix}_{\mathbf{x}=\mathbf{y}=0}. \quad (3.4)$$

Пусть $\lambda_1, \dots, \lambda_{2n}$ — собственные числа матрицы B , которые могут быть переупорядочены следующим образом $\lambda_{j+n} = -\lambda_j$, $j = 1, \dots, n$. Обозначим через вектор $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n)$ набор базовых собственных чисел системы (3.4).

Согласно теореме 12 из [12, § 12] существует каноническое формальное преобразование (3.3), где все функции $\mathbf{f}$ и $\mathbf{g}$ суть степенные ряды, которое приводит гамильтонову систему (3.1) к ее *нормальной форме*

$$\dot{\mathbf{u}} = \partial h / \partial \mathbf{v}, \quad \dot{\mathbf{v}} = -\partial h / \partial \mathbf{u}, \quad (3.5)$$

задаваемой нормализованным Гамильтонианом $h(\mathbf{u}, \mathbf{v})$

$$h(\mathbf{u}, \mathbf{v}) = \sum_{j=1}^n \lambda_j u_j v_j + \sum h_{\mathbf{pq}} \mathbf{u}^{\mathbf{p}} \mathbf{v}^{\mathbf{q}}, \quad (3.6)$$

содержащим только резонансные члены $h_{\mathbf{pq}} \mathbf{u}^{\mathbf{p}} \mathbf{v}^{\mathbf{q}}$, удовлетворяющие резонансному уравнению

$$\langle \mathbf{p} - \mathbf{q}, \boldsymbol{\lambda} \rangle = 0. \quad (3.7)$$

Здесь $0 \leq \mathbf{p}, \mathbf{q} \in \mathbb{Z}^n$, $|\mathbf{p}| + |\mathbf{q}| \geq 2$, а коэффициенты $h_{\mathbf{pq}}$ — постоянные величины.

Резонансное уравнение (3.7) имеет два вида решений, которым соответствуют два вида резонансных членов в нормальной форме (3.6):

1) *вековые члены* вида $h_{\mathbf{pp}} \mathbf{u}^{\mathbf{p}} \mathbf{v}^{\mathbf{p}}$, которые всегда присутствуют в гамильтоновой нормальной форме из-за особой структуры матрицы B линеаризованной системы (3.8);

2) *чисто резонансные члены*, которые соответствуют нетривиальным целочисленным решениям уравнения

$$\langle \mathbf{p}, \boldsymbol{\lambda} \rangle = 0. \quad (3.8)$$

Следуя [2, Гл. I, § 3] определим *кратность резонанса* $\mathbf{k}$ как число линейно независимых решений $\mathbf{p} \in \mathbb{Z}^n$ уравнения (3.8), а порядок резонанса $q = \min |\mathbf{p}|$ по $\mathbf{p} \in \mathbb{Z}^n$, $\mathbf{p} \neq 0$, $\langle \mathbf{p}, \boldsymbol{\lambda} \rangle = 0$.

Главным отличием нормальной формы системы Гамильтона (3.1) от нормальной формы общей системы ОДУ (2.1) является то, что первая всегда содержит вековые члены, даже при отсутствии нетривиальных решений уравнения (3.8). В этом случае имеем так называемую *нормальную форму Биркгофа* [13].

Пусть, как и в разделе 2, $I = \{i_1, \dots, i_k\}$ — кортеж возрастающих индексов $1 \leq i_1, i_k \leq n$, $k \leq n$. Обозначим через L_I *координатное подпространство*

$$L_I = \{\mathbf{u}, \mathbf{v} : u_j = v_j = 0 \text{ для всех } j \notin I\}.$$

Все ненулевые координаты $w_j = (u_j, v_j)$, $j \in I$, подпространства L_I назовем *внутренними координатами* и обозначим их для краткости $\mathbf{w}_I$, остальные назовем *внешними*. Собственные числа λ_j , $j \in I$, соответствующие внутренним координатам $\mathbf{w}_I$ назовем *внутренними собственными числами* и обозначим их λ_I . Остальные собственные числа λ_j , $j \notin I$, назовем *внешними собственными числами*.

Задача 2. Какие координатные подпространства L_I инвариантны в нормальной форме (3.5), (3.6), (3.7)?

Теорема 2. Координатное подпространство L_I размерности $2k$ инвариантно в нормальной форме (3.5)–(3.7), если каждое внешнее собственное число $\lambda_j \notin \lambda_I$ удовлетворяет следующему условию

$$\lambda_j \neq \langle \mathbf{p}, \lambda_I \rangle, \quad (3.9)$$

для любого ненулевого целочисленного вектора $\mathbf{p} \neq 0$ из решетки $\mathbf{p} \in \mathbb{Z}^k$.

Доказательство. Непосредственно следует из теоремы 1.

Замечание 1. Принципиальной разницей между условием (2.6) теоремы 1 и условием (3.9) теоремы 2 является то, что в гамильтоновом случае выбирается ненулевой вектор $\mathbf{p}$ из решетки $\mathbb{Z}^k$, а в общем случае выбирается уже неотрицательный вектор $0 \leq \mathbf{p} \in \mathbb{Z}^k$.

4. ПРИМЕРЫ

Рассмотрим некоторые модельные примеры существования инвариантных координатных подпространств в общем и гамильтоновом случаях.

Пример 1. Пусть собственные числа линейной части общей системы ОДУ равны $\lambda_1 = 1$, $\lambda_2 = \sqrt{2}$, $\lambda_3 = 1 + \sqrt{2}$.

Имеется три одномерных инвариантных координатных подпространства K_j , $j = 1, 2, 3$, соответствующих каждому из собственных чисел λ_j , ибо ни одно из соотношений λ_i/λ_j при $i \neq j$ не является натуральным числом.

Из трех двумерных координатных подпространств инвариантными являются подпространства K_{13} и K_{23} , но K_{12} неинвариантно, поскольку между собственными числами имеется соотношение $\lambda_3 = \lambda_1 + \lambda_2$, но при этом ни для каких векторов $\mathbf{p} \geq 0$, $\mathbf{p} \in \mathbb{Z}^2$ не выполняются соотношения $\lambda_1 = p_1\lambda_2 + p_2\lambda_3$ и $\lambda_2 = p_1\lambda_1 + p_2\lambda_3$.

Пример 2. Пусть собственные числа λ_j , $j = 1, \dots, n$, таковы, что отношения $\lambda_j/\lambda_1 \notin \mathbb{Z}$, $j = 2, \dots, n$. Тогда нормальная форма имеет двумерное инвариантное координатное подпространство $L_1 = \{u_j = v_j = 0, j \notin I_1\}$, где кортеж $I_1 = \{1\}$. На подпространстве L_1 нормальная форма (3.6) индуцирует гамильтонову нормальную форму с одной степенью свободы. В этом случае условие А выполнено (см [12, § 12]) и нормализующее преобразование сходится.

Если $\lambda_1 \neq 0$ и чисто мнимое, тогда для вещественной системы Гамильтона (3.1) вещественное подпространство L_1 является семейством периодических решений. Этот факт впервые был обнаружен А.М. Ляпуновым [14] и описан К. Зигелем в книге [15, §§16, 17] с применением гамильтонова формализма.

Пример 3. Пусть имеется единственная пара собственных чисел λ_1, λ_2 , удовлетворяющая условию

$$\frac{\lambda_2}{\lambda_1} = \frac{r}{s}, \quad (4.1)$$

где $r, s \in \mathbb{Z}$, $s \neq 0$, $(r, s) = 1$, т.е. кратность $\mathfrak{f}$ резонанса равна 1, а порядок резонанса равен $q = |r| + |s|$.

Тогда уравнение (3.8) имеет однопараметрическое семейство решений $\mathbf{p} = (lr, -ls, 0, \dots)$, $l \in \mathbb{Z} \setminus \{0\}$, а нормальная форма помимо вековых членов содержит чисто резонансные члены. Эти члены наименьшего порядка при $l = 1$ имеют вид $h_{12}u_1^r v_2^s$ и $h_{21}v_1^r u_2^s$.

Пусть, например, знаменатель s в (4.1) равен единице. Тогда нормальная форма содержит чисто резонансные члены $u_1^r v_2$ и $v_1^r u_2$. Это значит, что координатное подпространство L_1 не может быть инвариантным, поскольку правые части уравнений для внешних переменных u_2, v_2 содержат слабые, зависящие от внутренних переменных u_1, v_1 и, тем самым, эти правые части не могут быть всегда равны нулю когда $u_2 = v_2 = 0$. В то же самое время координатное подпространство L_2 является инвариантным.

Пример 4. Пусть матрица B линейной части системы Гамильтона с четырьмя степенями свободы имеет четыре пары собственных чисел: одна пара вещественных ± 1 , вторая пара чисто мнимых $\pm i$, третья и четвертая пары — комплексных $\pm 1 \pm i$. Переупорядочим эти собственные числа так, что вектор λ имеет следующий вид:

$$\lambda = (1, i, 1 + i, 1 - i).$$

Очевидно, что имеется четыре двумерных инвариантных координатных подпространства L_1, L_2, L_3, L_4 , соответствующих каждому из собственных чисел λ_j , $j = 1, 2, 3, 4$, ибо все отношения λ_i/λ_j , $i \neq j$, не являются целыми числами.

Из шести координатных подпространств размерности четыре имеем:

- L_{12} не является инвариантным, поскольку $\lambda_3 = \lambda_1 + \lambda_2$.
- L_{13} не является инвариантным, поскольку $\lambda_2 = \lambda_3 - \lambda_1$.
- L_{14} не является инвариантным, поскольку $\lambda_2 = \lambda_1 - \lambda_4$.
- L_{23} не является инвариантным, поскольку $\lambda_1 = \lambda_3 - \lambda_2$.
- L_{24} не является инвариантным, поскольку $\lambda_1 = \lambda_2 + \lambda_4$.
- L_{34} инвариантно, поскольку ни λ_1 ни λ_2 не могут быть получены как линейная комбинация λ_3 и λ_4 с целыми коэффициентами. Тем самым условие (3.9) теоремы 2 выполнено.

Наконец, не существует ни одного инвариантного координатного подпространства размерности шесть.

5. ПРИМЕНЕНИЕ q -СУБДИСКРИМИНАНТОВ ДЛЯ ПОИСКА РЕЗОНАНСОВ

Как следует из рассуждений пп. 2 и 3, вид резонансных членов нормальной формы общей (2.1) или гамильтоновой (3.1) систем определяется исключительно собственными значениями λ матрицы A или B соответственно. Это значит, что для выяснения вопроса существования инвариантных подпространств в нормальной форме (2.3)–(2.5) или (3.5)–(3.7) достаточно определить структуру собственных чисел матриц A или B , а именно, выполняются или нет соответствующие условия (2.6) или (3.9). Частично ответ на этот вопрос может быть получен с использованием техники q -субдискриминантов характеристического многочлена $f(\lambda)$ матрицы A или B .

В дальнейшем ограничимся важным с точки зрения приложений случаем, когда вектор собственных чисел λ матриц A или B состоит только из вещественных и/или чисто мнимых величин. Дополнительно предположим, что если собственные числа λ удовлетворяют резонансному уравнению (2.5) или (3.8), то все λ_j , входящие в эти уравнения попарно рационально соизмеримы. Другими словами, нет такой ситуации с собственными числами как в примерах 1 и 4.

5.1. q -Субдискриминанты многочлена и их свойства

Применение q -субдискриминантов $D_q^{(k)}(f)$ характеристического многочлена позволяет не только выяснить наличие у него соизмеримых корней, но и, при определенных условиях, найти эти корни, не прибегая к вычислению всех собственных чисел. Если правые части систем (2.1) или (3.1) зависят от параметров, то и q -субдискриминанты являются функциями этих параметров, что позволяет определить, при каких значениях последних имеет место быть резонанс соответствующей кратности и порядка.

Вначале кратко напомним определение субдискриминанта k -го порядка произвольного приведенного многочлена степени n (подробнее, см. [16, 17]).

Определение 2. Пусть

$$f_n(x) = x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n \quad (5.1)$$

некоторый приведенный многочлен от переменной x . Тогда его k -й субдискриминант $D^{(k)}(f_n)$, $k = 0, \dots, n-2$ задается формулой

$$D^{(k)}(f_n) = \sum_{\substack{I \subset \{1, \dots, n\} \\ \#(I) = n-k}} \prod_{\substack{(j,l) \in I \\ l > j}} (x_j - x_l)^2,$$

где x_j – корни многочлена (5.1), $\#(I)$ – мощность множества I . Для $k = n-1$ положим $D^{(n-1)}(f_n) = n$, а для $k = n$ положим $D^{(n)}(f_n) = 1$. Для $k = 0$ получаем $D^{(0)}(f_n) = D(f_n)$.

Составим последовательность субдискриминантов многочлена $f_n(x)$

$$S(f_n) \stackrel{\text{def}}{=} (D^{(0)}(f_n), D^{(1)}(f_n), \dots, D^{(n-1)}(f_n), 1).$$

Пусть $P(S)$ есть число знаков постоянств в последовательности S , а $V(S)$ – число знаков перемен этой последовательности.

Утверждение 1 ([17, Theorem 4.33]). Число вещественных корней многочлена $f_n(x)$ равно величине

$$\text{PmV}(f_n) \stackrel{\text{def}}{=} P(S) - V(S).$$

Для проверки условия, что все корни многочлена вещественные и/или чисто мнимые, следует перейти к вспомогательному многочлену $\check{f}_n(x)$, все корни которого суть квадраты корней исходного многочлена (2). Это легко сделать применив преобразование Чирнгауза [18, § 136]: $\check{f}_n(x) = \text{Res}_z(f_n(z), x - z^2)$. Используя последовательность субдискриминантов $S(\check{f}_n)$ и утверждение 1, легко получить следующее.

Утверждение 2. Многочлен (5.1) имеет только вещественные и/или чисто мнимые корни, если все субдискриминанты вспомогательного многочлена $\check{f}_n(x)$ неотрицательны.

Вычисление субдискриминанта многочлена осуществляется либо с помощью одного из матричных методов с использованием матрицы Безу, Сильвестра, ганкелевой матрицы ньютоновых сумм, либо с использованием алгоритма псевдоделения Якоби (см., например, [16, 19, 20]).

Отметим, что многие указанные выше методы реализованы в различных системах компьютерной алгебры. Например, в системе Wolfram Mathematica [9] функции `Subresultants[f1, f2, x]` и `SubresultantPolynomials[f1, f2, x]` позволяют находить соответственно скалярные и полиномиальные субрезультанты пары многочленов f_1, f_2 относительно переменной x . Фрагмент соответствующего кода приведен в листинге 1.

Листинг 1

```

coefflst = Reverse [Join [{1}, Array [a, 3]]];
f3 = Total [coefflst x^(Range [4] = 1)]
Df3 = D[f3, x]
x^3 + x^2 a [1] + x a [2] + a [3]
3 x^2 + 2 x a [1] + a [2]

Subresultants [f3, Df3, x]
{-a [1]^2 a [2]^2 + 4 a [2]^3 + 4 a [1]^3 a [3]
- 18 a [1] a [2] a [3] + 27 a [3]^2,
-2 a [1]^2 + 6 a [2], 3}

```

Система компьютерной алгебры Maple [10] предлагает набор процедур (SubresultantChain, SubresultantOfIndex, LastSubresultant) из пакета RegularChains[ChainTools], позволяющих вычислять структуру данных, хранящую полиномиальные субрезультанты пары многочленов от нескольких переменных. Непосредственные вычисления также могут быть произведены с использованием процедур BezoutMatrix, Syl-

vesterMatrix и HankelMatrix из пакета LinearAlgebra (см. [20]).

Наконец, пакет SymPy [11] для языка программирования Python также содержит ряд функций, позволяющих вычислять полиномиальные субрезультанты пары многочленов в символьном виде. Пример вычисления дискриминанта кватерники $f_4(x)$ дан в листинге 2.

Листинг 2

```

from sympy import *
nord = 4
x = var ("x")
c f l s t = symbols ("a1 : { }".format (nord))
f 4 = Poly (x** nord+sum([c f l s t [nord=1=i] * x** i\
for i in range (nord)]), x)
Df4 = f 4.d i f f (x)
SubDf4 = s u b r e s u l t a n t s (f4, Df4)
print (SubDf4 [=1])

```

$$\begin{aligned}
& -27a_1^4a_4^2 + 18a_1^3a_2a_3a_4 - 4a_1^3a_3^3 - 4a_1^2a_2^3a_4 + a_1^2a_2^2a_3^2 + 144a_1^2a_2a_4^2 - 6a_1^2a_3^2a_4 - 80a_1a_2^2a_3a_4 + 18a_1a_2a_3^3 - \\
& 192a_1a_3a_4^2 + 16a_2^4a_4 - 4a_2^3a_3^2 - 128a_2^2a_4^2 + 144a_2a_3^2a_4 - 27a_3^4 + 256a_4^3
\end{aligned}$$

Для определения рациональной соизмеримости корней многочлена воспользуемся q -аналогам классических производной и субдискриминанта.

Напомним основные q -объекты (см., например, [21–23]), используемые далее.

Определение 3. Определим следующие объекты:

- q -скобка числа a : $[a]_q = \frac{q^a - 1}{q - 1}$, $a \in \mathbb{R} \setminus \{0\}$,
- сдвинутый q -факториал (q -символ Похгаммера): $(a; q)_n = \prod_{k=0}^{n-1} (1 - aq^k)$, $(a; q)_0 = 1$,
- q -факториал $[n]_q! = \prod_{k=1}^n [k]_q = \frac{(q; q)_n}{(1 - q)^n}$, $q \neq 1$,
- q -биномиальные (гауссовы) коэффициенты

$$\binom{n}{k}_q = \frac{[n]_q!}{[n-k]_q! [k]_q!} = \prod_{i=1}^k \frac{q^{n-i+1} - 1}{q^i - 1},$$

• q -бином:

$$\{x; a\}_{n; q} \stackrel{\text{def}}{=} \prod_{i=0}^{n-1} (x - aq^i), \quad \{x; t\}_{0; q} = 1.$$

• При $q \rightarrow 1$ все определенные выше объекты становятся классическими.

Отметим, что коэффициенты q -бинома суть гауссовы коэффициенты.

Производная Джексона (q -производная, q -дифференциальный оператор Джексона) есть

$$(\mathcal{A}_q f)(x) \stackrel{\text{def}}{=} \begin{cases} \frac{f(qx) - f(x)}{(q-1)x}, & x \neq 0, \\ f'(0), & x = 0, \end{cases} \quad q \notin \{0, 1\}.$$

Она обладает всеми свойствами обычной производной. Исключением является правило дифференцирования сложной функции, которое отсутствует в случае q -производной.

Утверждение 3. Пусть f и h — два многочлена, тогда

1. $(\mathcal{A}_q f)(x) \equiv 0 \Leftrightarrow f(x) \equiv \text{const};$
2. $(\mathcal{A}_q(fh))(x) = (\mathcal{A}_q f)(x)h(x) + f(qx)(\mathcal{A}_q h)(x);$
3. $\left(\mathcal{A}_q \frac{f}{h}\right)(x) = \frac{(\mathcal{A}_q f)(x)h(x) - f(x)(\mathcal{A}_q h)(x)}{h(x)h(qx)};$
4. $f(qx) = f(x) + (q-1)x(\mathcal{A}_q f)(x).$

Помимо указанных выше свойств отметим, что q -производная функции x^n равна $[n]_q x^{n-1}$, а применение q -производной к q -биному степени n снова дает q -бином степени $n-1$, умноженный на $[n]_q$.

С помощью q -производной теперь определяется q -аналог классического дискриминанта многочлена.

Определение 4. Определим q -дискриминант $D_q(f_n)$ многочлена $f_n(x)$ как результат пары многочленов $f_n(x)$ и $(\mathcal{A}_q f_n)(x)$:

$$D_q(f_n) = (-1)^{n(n-1)/2} \text{Res}(f_n(x), (\mathcal{A}_q f_n)(x)).$$

Равенство нулю q -дискриминанта многочлена $f_n(x)$ при фиксированном q является признаком существования по крайней мере одной пары q -соизмеримых корней, однако подробную структуру всех соизмеримых корней можно получить с использованием последовательности q -субдискриминантов различных порядков многочлена $f_n(x)$

$$S_q(f_n) \stackrel{\text{def}}{=} (D_q^{(0)}(f_n), D_q^{(1)}(f_n), \dots, D_q^{(n-1)}(f_n)). \quad (5.2)$$

Частным случаем результата, доказанного для оператора Хана, обобщающего q -производную, в [22, 24], является следующая теорема.

Теорема 3. Многочлен $f_n(x)$ имеет ровно $n-d$ различных последовательностей q -соизмеримых корней, тогда и только тогда, когда в последовательности (5.2) $k-x$ q -субдискриминантов $D_q^{(k)}(f_n)$, критерий многочлена f_n , $k = 0, \dots, n-2$, первым отличным от нуля является q -субдискриминант $D_q^{(d)}(f_n)$ с номером d .

Все соизмеримые корни многочлена $f_n(x)$ являются корнями наибольшего общего делителя многочлена $f_n(x)$ и его q -производной $(\mathcal{A}_q f_n)(x)$:

$$\tilde{f}_q(x) = \text{НОД}(f_n(x), (\mathcal{A}_q f_n)(x)).$$

Теорема 3 утверждает, что степень многочлена $\tilde{f}_q(x)$ равна номеру d первого ненулевого q -субдискриминанта в последовательности $S_q(f_n)$.

Отметим, что q -субдискриминанты вычисляются с помощью любого из матричных методов вычисления классических субрезультантов пары многочленов $f_n(x)$ и $(\mathcal{A}_q f_n)(x)$. Например, если составить из коэффициентов указанных выше многочленов матрицу Сильвестра $\text{Sylv}_q(f_n)$ размера $(2n-1) \times (2n-1)$ в форме Сильвестра–Хабихта, то k -й q -субдискриминант $D_q^{(k)}(f_n)$ равен определителю k -го иннора [25, Гл. I] этой матрицы.

Пусть в условиях теоремы 3 первый отличный от нуля q -дискриминант имеет номер d , $0 < d < n-1$. Обозначим через $\mathbf{M}_d^{(i)}$, $i = 1, \dots, d$, d -й иннор модифицированной q -матрицы Сильвестра $\text{Sylv}_q(f_n)$, в которой столбец с номером $2n-1-d$ заменен ее столбцом с номером $2n-1-d+i$, а через $M_d^{(i)}$ — определитель этого иннора. Тогда, как показано в [22] имеет место быть утверждение.

Утверждение 4. Если в последовательности q -субдискриминантов $D_q^{(i)}(f_n)$, $i = 0, \dots, n-2$, первым отличным от нуля является q -субдискриминант с номером d , то

$$\tilde{f}_q(x) \equiv D_q^{(d)} x^d + M_d^{(1)} x^{d-1} + \dots + M_d^{(d)}. \quad (5.3)$$

Укажем здесь некоторые важные свойства q -субдискриминантов приведенного многочлена (5.1).

I. k -й q -субдискриминант $D_q^{(k)}(f_n)$ представляет собой квазиоднородный многочлен от переменных $a_1, \dots, a_n$, такой, что

$$\sum_{j=1}^m j a_j \frac{\partial D_q^{(k)}(f_n)}{\partial a_j} = (m-k)(m-k-1) D_q^{(k)}(f_n).$$

При этом k -й субдискриминант зависит не более чем от m первых коэффициентов многочлена (5.1), где $m = \min(2(n-k-1), n)$.

II. k -й q -субдискриминант $D_q^{(k)}(f_n)$ представляет собой возвратный многочлен четного порядка от переменной q . Возвратность следует из того, что соизмеримость корней $\lambda_i/\lambda_j = q$, $i \neq j$, влечет соизмеримость $\lambda_j/\lambda_i = 1/q$. Это свойство позволяет, с одной стороны, понизить вдвое степень q -субдискриминанта, как многочлена от q с помощью подстановки

$$q + \frac{1}{q} = Q, \quad (5.4)$$

а с другой стороны избежать двойной проверки соизмеримости пары корней q и $1/q$. В силу вышесказанного, степень $D_q(f_n)$ как многочлена от q равна $n(n-1)$.

5.2. Алгоритм поиска резонансов

Рассмотрим алгоритм исследования линейной части (2.2) общей системы ОДУ для поиска резонансов. Алгоритм представлен в виде последовательности шагов.

Шаг 1. Для матрицы A системы (3) вычисляем характеристический многочлен $f_n(\lambda)$ и проверяем наличие у него только вещественных и/или чисто мнимых корней с использованием утверждения 2.

Шаг 2. По многочлену $f_n(\lambda)$ вычисляем последовательность $S_q(f)$ (5.2) q -субдискриминантов порядков от 0 до $n - 2$. На этом этапе их степени как многочленов от q можно понизить вдвое, как указано в свойстве II.

Дальнейшие действия зависят от того, насколько легко удастся найти рациональные корни q -дискриминанта $D_q(f_n)$ как многочлена от q .

Шаг 3а. Пусть $D_q(f_n)$ является многочленом из кольца $\mathbb{Z}[q]$. Тогда его следует попытаться факторизовать с использованием различных алгоритмов (например, Кантора–Цассенхауза или Берлекемпа, см. [26, Гл. 6]). В случае успеха, получаем все (или часть) значений q , для которых имеет место соизмеримость корней многочлена (5.1).

Шаг 3б. Если менение предыдущего шага невозможно, то применим некоторый вариант перебора. Ограничимся определенным максимальным порядком m резонанса. Составим упорядоченный кортеж $\mathcal{P}_m$ всевозможных натуральных пар (r, s) таких, что $r, s \in \mathbb{N}$, НОД(r, s) = 1, $r \geq s$, $r + s \leq m$. Теперь, последовательно перебирая элементы кортежа $\mathcal{P}_m$, для каждой пары (r, s) проверяем равенство нулю q -дискриминанта $D_q(f)$ для значения $q = r/s$.

Шаг 4. Пусть для некоторого рационального q_1^* выполнено условие $D_{q_1^*}(f_n) = 0$. Тогда вычисляем по последовательности (5.2) степень многочлена $\tilde{f}_{q_1^*}(x)$, по формуле (5.3) сам этот многочлен и определяем структуру его корней. Если степень этого многочлена мала, то соизмеримые корни могут быть найдены по соответствующим формулам, а исходный многочлен $f_n(\lambda)$ можно представить в виде $f_n(\lambda) = u(\lambda)v(\lambda)$, где $u(\lambda)$ — множитель с уже известными корнями, а $v(\lambda)$ — с еще неизвестными, но среди которых нет q_1^* -соизмеримых корней.

Шаг 5. Последующие действия сводятся к исследованию, имеют ли множители $u(x)$ и $v(x)$ q -соизмеримые корни или нет, а также, имеет ли множитель $v(x)$ q -соизмеримые корни для $q \neq q_1^*$.

Замечание 2. При исследовании линейной части (3.4) системы Гамильтона следует учитывать, что характеристический многочлен $f_{2n}(\lambda)$ матрицы B является многочленом только четных степеней λ . Следовательно, можно понизить его порядок вдвое

и исследовать многочлен, названный в [27] *полухарактеристическим*, $\hat{f}_n(u) = f_{2n}(\lambda)$, где $\mu = \lambda^2$. В этом случае кортеж $\mathcal{P}_m$ составляется из пар (r^2, s^2) натуральных взаимно простых чисел r и s .

Отметим, что в случае когда удастся явно выразить собственные числа матрицы B , квадратичную часть гамильтониана (3.2) можно привести к нормальной форме одним из методов, описанным в книгах [8, 28].

Замечание 3. Описанный выше метод обладает двумя недостатками. Во-первых, при наличии резонанса кратности $k > 1$ может возникнуть ситуация, когда попарная соизмеримость собственных чисел задается большими значениями q , но сам резонанс может иметь малый порядок. Например, если три собственных числа $\lambda_1, \lambda_2, \lambda_3$ относятся как натуральные числа $2k - 1, 2k, 2k + 1$, $k \in \mathbb{N}$, т.е. $\lambda_1 : \lambda_2 : \lambda_3 = (2k - 1) : 2k : (2k + 1)$, то наименьший порядок попарной соизмеримости равен $4k - 1$. Однако, в то же самое время, имеет место быть резонансное соотношение $\lambda_3 = 2\lambda_2 - \lambda_1$, имеющее порядок 4. Во-вторых, метод неспособен находить такие резонансы, которые задействуют три и более собственных чисел, но при этом эти числа не являются попарно соизмеримыми.

Замечание 4. Описанные в этом разделе методы исследования характеристического многочлена $f(\lambda)$ линейной части ОДУ легко алгоритмизируются и могут быть реализованы во многих системах компьютерной алгебры. Автором были написаны библиотеки процедур в системах компьютерной алгебры Maple и SymPy, предназначенные для вычисления q -субдискриминантов многочлена и исследования резонансных соотношений между корнями последнего без их явных вычислений. Ниже дан пример таких вычислений.

5.3. Модельный пример

Рассмотрим шесть симпатических математических маятников (т.е. одинаковой массы m и длины l), точки подвеса которых расположены на равных расстояниях d на горизонтальной прямой. Пусть маятники соединены между собой невесомыми линейно упругими пружинами жесткости k длиной d в недеформированном состоянии. Точки прикрепления пружин расположены на расстоянии $b \leq d$ от точек подвеса маятников. Следуя работе [29], в которой рассмотрена пара таких маятников, несложно получить квадратичную часть Π_2 потенциальной энергии в окрестности положения равновесия, когда все углы φ_i , $i = 1, \dots, 6$, отклонения маятников от вертикали равны нулю. Переходя к каноническим переменным и масштабируя время

$\tau = \sqrt{\frac{g}{l}}$ можно получить полухарактеристический многочлен (вековое уравнение) в виде

$$\begin{aligned} \hat{f}_6(\mu) = & \mu^6 - 2(5\beta + 3)\mu^5 + (36\beta^2 + 50\beta + 15)\mu^4 - \\ & - 2(28\beta^3 + 72\beta^2 + 50\beta + 10)\mu^3 + \\ & + (35\beta^4 + 168\beta^3 + 216\beta^2 + 100\beta + 15)\mu^2 - \\ & - 2(3\beta^5 + 35\beta^4 + 84\beta^3 + 72\beta^2 + 25\beta + 3)\mu + \\ & + (2\beta + 1)(3\beta + 1)(\beta + 1)(\beta^2 + 4\beta + 1), \end{aligned} \quad (5.5)$$

где $\mu = \lambda^2$, а единственный параметр $\beta = \frac{kb^2}{mgl} > 0$.

В силу устойчивости положения равновесия, многочлен (5.5) имеет шесть положительных вещественных корней — по числу степеней свободы исходной механической системы.

Следуя шагу 2 *Алгоритма* вычислим первый q -субдискриминант $D_q(\hat{f}_6)$ и используя подстановку (5.4) упростим его. В результате получим многочлен относительно переменной Q степени 15. Здесь его явное выражение не приводится в силу его громоздкости. В системе Maplesoft Maple этот многочлен удалось разложить на линейные и квадратные множители, что позволило легко найти все соизмеримости между корнями многочлена (5.5). Таким образом, выполнен шаг 3а *Алгоритма* и имеется возможность найти все корни многочлена, а следовательно проверить выполнение резонансных соотношений между ними.

В системе SymPy разложить q -дискриминант $D_q(\hat{f}_6)$ на множители не удалось, поэтому применим шаги 3б и далее *Алгоритма* для некоторого фиксированного значения параметра $\beta = 48/25$. Выберем максимальный порядок резонанса $m = 20$ и, согласно замечанию 2, составим кортеж $\mathcal{P}_{20}$ всевозможных пар (r^2, s^2) таких, что $\text{НОД}(r, s) = 1$ и $r + s \leq 20$, $r, s \in \mathbb{N}$. Перебирая все такие пары из $\mathcal{P}_{20}$ найдем, что при $q_1^* = 121/25$ и $q_2^* = 169/25$ q -дискриминант обращается в нуль, а остальные q -субдискриминанты нет. При этом многочлены $\tilde{f}_{q_1}(\mu)$ и $\tilde{f}_{q_2}(\mu)$, вычисленные по формуле (5.3) имеют один и тот же корень $\mu_1 = 1$. Следовательно, должно быть еще одно значение $q_3^* = q_2^*/q_1^* = 169/121$, при котором обнуляется q -дискриминант. Таким образом, найдены три корня многочлена (5.5): $\mu_1 = 1$, $\mu_2 = q_1^*$, $\mu_3 = q_2^*$, и, следовательно, $f_6(\mu) = (\mu - 1)(\mu - q_1^*)(\mu - q_2^*)v_3(\mu)$. Остальные иррациональные корни μ_k , $k = 4, 5, 6$, кубического многочлена $v_3(\mu)$ легко находятся с помощью тригонометрической формулы Виета [18, § 119].

Итак, имеется резонансное соотношение (3.8)

$$5p_1 + 11p_2 + 13p_3 = 0 \quad (5.6)$$

между частотами $\lambda_1 = 1$, $\lambda_2 = 11/5$, $\lambda_3 = 13/5$. Соотношение (5.6) имеет три целочисленных решения $\mathbf{p}_1 = (1, -4, 3)$, $\mathbf{p}_2 = (3, 1, -2)$ и $\mathbf{p}_3 = (4, -3, 1)$, из которых следует, что условие (3.9) теоремы 2 не выполнено. Таким образом, имеется инвариантное координатное подпространство L_I , $I = \{1, 2, 3\}$, размерности шесть в нормальной форме, соответствующее трем собственным числам $\lambda_1, \lambda_2, \lambda_3$. Каждому из оставшихся собственных чисел λ_i , $i = 4, 5, 6$, соответствует двумерное инвариантное координатное подпространство L_i . В силу того, что порядки целочисленных векторов $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ равны соответственно, 4, 3, 4, то исследование динамики исходной системы на подпространстве L_I возможно лишь выполнив нелинейную нормализацию исходной системы по крайней мере до 6-го порядка включительно.

6. БЛАГОДАРНОСТИ

Автор выражает глубокую благодарность профессору А.Д. Брюно за помощь, плодотворное обсуждение и поддержку при написании данной статьи.

СПИСОК ЛИТЕРАТУРЫ

1. Брюно А.Д. Аналитическая форма дифференциальных уравнений (I) // Тр. ММО. 1971. Т. 25. С. 119–262.
2. Брюно А.Д. Ограниченная задача трех тел: Плоские периодические орбиты. М.: Наука, 1990. 296 с.
3. Маркеев А.П. Точки либрации в небесной механике и космодинамике. М.: Гл. ред. физ.-мат. литер. изд-ва “Наука”, 1978. 352 с.
4. Bruno A.D. Local Methods in Nonlinear Differential Equations. Berlin–Heidelberg–New York–London–Paris–Tokyo: Springer–Verlag, 1989. 350 p.
5. Bruno A.D., Enderal V.F., Romanovski V.G. On new integrals of the Algaba-Gamero-Garcia system // Computer Algebra in Scientific Computing / Ed. by V.P. Gerdt, W. Koepf, E.W. Mayr, E.V. Vorozhtsov. Berlin Heidelberg: Springer, 2017. V. 10490 of Lecture Notes in Computer Science. P. 40–50.
6. Брюно А.Д. Нормальная форма системы Гамильтона с периодическим возмущением // ЖВММФ. 2020. Т. 60. № 1. С. 36–52.
7. Брюно А.Д. Нормализация периодической системы Гамильтона // Программирование. 2020. Т. 46. № 2. С. 76–83.
8. Журавлев В.Ф., Петров А.Г., Шундерюк М.М. Избранные задачи гамильтоновой механики. М.: ЛЕ-НАНД, 2015. 304 с.
9. Wolfram S. The Mathematica Book. Wolfram Media, Inc., 2003. 1488 p.
10. Thompson I. Understanding Maple. Cambridge University Press, 2016. 228 p.

11. SymPy: symbolic computing in Python / Aaron Meurer, Christopher P. Smith, Mateusz Paprocki et al. // Peer J. Computer Science. 2017. V. 3. P. e103. <https://doi.org/10.7717/peerjcs.103>
12. Брюно А.Д. Аналитическая форма дифференциальных уравнений (II) // Тр. ММО. 1972. Т. 26. С. 199–239.
13. Биркгоф Д.Д. Динамические системы. Ижевск: Изд. дом “Удмуртский университет”, 1999. 408 с.
14. Lyapunov A.M. Problme gnral de la stabilit du mouvement // Ann. Fac. Sci. Toulouse Math. 1892. V. 9 (1907). № 2. P. 204474. Ann Math. Studies (1947). Stability of Motion. New York, London: Academic Press, 1966 (English).
15. Зигель К., Мозер Ю. Лекции по небесной механике. Ижевск: НИЦ “Регулярная и хаотическая динамика”, 2001. 384 с.
16. Калинина Е.А., Утешев А.Ю. Теория исключения: Учеб. пособие. СПб.: Изд-во НИИ химии СПбГУ, 2002. 72 с.
17. Basu S., Pollack R., Roy M.-F. Algorithms in Real Algebraic Geometry. Algorithms and Computations in Mathematics 10. Berlin Heidelberg New York: Springer-Verlag, 2006. ix+662 p.
18. Сушкевич А.К. Основы высшей алгебры. 4-е изд. М.-Л.: ОГИЗ ГИТТЛ, 1941. 460 с.
19. Gathen J. von zur, Leking T. Subresultants revisited // Theoretical Computer Science. 2003. V. 297. P. 199–239.
20. Батхин А.Б. Параметризация дискриминантного множества вещественного многочлена // Программирование. 2016. Т. 42. № 2. С. 8–21.
21. Кац В.Г., Чен П. Квантовый анализ. М.: МЦНМО, 2005. 128 с.
22. Батхин А.Б. Параметризация множества, определяемого обобщенным дискриминантом многочлена // Программирование. 2018. № 2. С. 5–17.
23. Батхин А.Б. Вычисление резонансного множества многочлена при ограничениях на коэффициенты // Программирование. 2019. № 2. С. 6–15.
24. Батхин А.Б. Вычисление обобщенного дискриминанта вещественного многочлена // Препринты ИПМ им. М.В. Келдыша. 2017. № 88. 40 с. http://www.keldysh.ru/papers/2016/prep2017_88.pdf.
25. Джурри Э. Инноры и устойчивость динамических систем. М., 1979. 304 с.
26. Акритас А.Г. Основы компьютерной алгебры с приложениями. М.: Мир, 1994. 544 с.
27. Батхин А.Б., Брюно А.Д., Варин В.П. Множества устойчивости многопараметрических гамильтоновых систем // Прикл. мат. мех. 2012. Т. 76. № 1. С. 80–133.
28. Маркеев А.П. Линейные гамильтоновы системы и некоторые задачи об устойчивости движения спутника относительно центра масс. М.-Ижевск: НИЦ “Регулярная и хаотическая динамика”, 2009. 396 с.
29. Маркеев А.П. Нелинейные колебания симпатических маятников // Нелинейная динамика. 2010. Т. 6. № 3. С. 605–621.

ФАКТОРИЗАЦИЯ БУЛЕВЫХ ПОЛИНОМОВ: ПАРАЛЛЕЛЬНЫЕ
АЛГОРИТМЫ И ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА© 2021 г. П. Г. Емельянов^{а,*}, М. Кришна^{с,**}, В. Кулкарни^{б,***}, С. К. Нанди^{б,****},
Д. К. Пономарев^{а,*****}, С. Раха^{б,*****}^а Институт систем информатики им. А.П. Ершова СО РАН; Новосибирский государственный университет,
Новосибирск, Россия^б Department of Computational and Data Sciences, Indian Institute of Science, Бангалор, Индия^с Morphing Machines Pvt Ltd, Бангалор, Индия

*E-mail: emelyanov@iis.nsk.su

**E-mail: madhava@iisc.ac.in

***E-mail: vadirajk@iisc.ac.in

****E-mail: nandy@iisc.ac.in;

*****E-mail: ponom@iis.nsk.su

*****E-mail: raha@iisc.ac.in

Поступила в редакцию 02.04.2020 г.

После доработки 06.05.2020 г.

Принята к публикации 06.05.2020 г.

Факторизация полиномов — классическая алгоритмическая проблема алгебры, имеющая широкий спектр приложений. Особый интерес представляет факторизация над конечными полями, среди которых поле порядка два является, вероятно, наиболее важным в связи с представлением булевых функций полиномами Жегалкина. В частности, факторизация булевых полиномов соответствует конъюнктивной декомпозиции булевых функций, заданных в алгебраической нормальной форме. Кроме того, факторизация дает решение проблемы декомпозиции функций, заданных в СДНФ и позитивных ДНФ, а также декартовой декомпозиции реляционных данных. Эти приложения демонстрируют важность разработки быстрых алгоритмов факторизации. В статье мы рассматриваем некоторые недавно предложенные алгоритмы факторизации полиномиальной сложности и описываем параллельную MIMD-реализацию, которая использует как параллелизм уровня задачи, так и параллелизм уровня данных. Мы представляем эксперименты, выполненные на бенчмарках логического синтеза и на синтетических (случайных) полиномах, которые показывают значительное ускорение факторизации. В заключение представлены результаты тестирования параллельной реализации алгоритма на массивнопараллельной многоядерной архитектуре (Redefine).

DOI: 10.31857/S0132347421020059

1. ВВЕДЕНИЕ

Факторизация полиномов — классическая алгоритмическая проблема алгебры [1], которая имеет широкий спектр приложений. Один вариант проблемы, привлекающий особое внимание — факторизация булевых полиномов, т.е. мультилинейных полиномов над конечным полем порядка 2. Булев многочлен — это одно из широко известных представлений булевых функций, также известных в математической логике как полиномы Жегалкина [2] или каноническая форма Риди–Маллера [3] в синтезе логических схем и теории помехоустойчивого кодирования. В последнее время это представление становится все более популярным в связи со следующими преимуществами.

Во-первых, это естественное и компактное представление некоторых важных классов булевых функций (например, арифметические функции, кодеры, и т.д.). Во-вторых, оно допускает естественное отображение в некоторых схемных технологиях (например, основанных на FPGA или в наноструктурной электронике), а также имеет хорошие свойства для тестирования.

Факторизация булевых полиномов является частным случаем конъюнктивной декомпозиции булевых функций (декомпозиция в конъюнкцию функций с непересекающимися множествами переменных или AND-декомпозиция). Действительно, для булева полинома, в котором всякая переменная имеет степень 1, факторизация дает

факторы с непересекающимися наборами переменных: $F(X, Y) = F_1(X) \times F_2(Y)$, $X \cap Y = \emptyset$.

Недавно было показано [4, 5], что факторизация булевых полиномов решает задачу конъюнктивной декомпозиции функций, заданных в СДНФ и позитивных ДНФ. Кроме того, это дает метод декартовой декомпозиции таблиц реляционных баз данных [6, 7], т.е. отыскание таблиц таких, что их неупорядоченное декартово произведение совпадает с исходной таблицей (обращение оператора CROSS JOIN). Несколько примеров, иллюстрирующих указанные применения, приведены ниже.

Рассмотрим следующую ДНФ

$$\varphi = (x \wedge u) \vee (x \wedge v) \vee (y \wedge u) \vee (y \wedge v) \vee (x \wedge u \wedge v).$$

Она эквивалентна

$$\psi = (x \wedge u) \vee (x \wedge v) \vee (y \wedge u) \vee (y \wedge v),$$

так как последняя конъюнкция избыточна. Можно видеть, что

$$\psi \equiv (x \vee y) \wedge (u \vee v)$$

и компоненты декомпозиции $x \vee y$ и $u \vee v$ могут быть восстановлены из факторов полинома

$$F_\psi = xu + xv + yu + yv = (x + y) \times (u + v),$$

построенного для ψ .

Следующая СДНФ

$$\varphi = (x \wedge \neg y \wedge u \wedge \neg v) \vee (x \wedge \neg y \wedge \neg u \wedge v) \vee (\neg x \wedge y \wedge u \wedge \neg v) \vee (\neg x \wedge y \wedge \neg u \wedge v)$$

эквивалентна

$$(x \wedge \neg y) \vee (\neg x \wedge y) \vee (u \wedge \neg v) \vee (\neg u \wedge v)$$

и компоненты декомпозиции φ могут быть восстановлены из факторов полинома

$$F_\varphi = x\bar{y}u\bar{v} + x\bar{y}u\bar{v} + \bar{x}yu\bar{v} + \bar{x}yu\bar{v} = (\bar{x}y + \bar{x}y) \times (u\bar{v} + \bar{u}v), \quad (1.1)$$

построенного для φ .

Наконец, декартова декомпозиция следующей таблицы

B	E	D	A	C
z	q	u	x	y
y	q	u	x	y
y	r	v	x	z
z	r	v	x	z
y	p	u	x	x
z	p	u	x	x

 $=$

A	B
x	y
x	z

 $\times$

C	D	E
x	u	p
y	u	q
z	v	r

может быть найдена факторизацией полинома

$$\begin{aligned} & z_B \times q \times u \times x_A \times y_C + y_B \times q \times u \times x_A \times y_C + \\ & + y_B \times r \times v \times x_A \times z_C + z_B \times r \times v \times x_A \times z_C + \\ & + y_B \times p \times u \times x_A \times x_C + z_B \times p \times u \times x_A \times x_C = \\ & = (x_A \times y_B + x_A \times z_B) \times \\ & \times (q \times u \times y_C + r \times v \times z_C + p \times u \times x_C), \end{aligned}$$

построенного на основе содержимого таблицы.

Декомпозиция позволяет находить более компактное представление булевых функций и таблиц данных, которые применяются в области синтеза логических схем, самоорганизующихся баз данных и поиска зависимостей в данных. Так как большие объемы входов – типичная ситуация в перечисленных задачах, чрезвычайно важно иметь эффективные на практике алгоритмы факторизации булевых полиномов.

В [8], изучая проблему факторизации полиномов над произвольными конечными полями, представленных в виде арифметических схем, Шпилька и Волкович показали связь между факторизацией полиномов и проверкой тождества полиномов. В частности, из их результата следует, что булев полином может быть факторизован

за время $O(l^3)$, где l – размер полинома заданного как символьная последовательность. Подход использует умножение полиномов, полученных из исходного, и эта операция является дорогостоящей для больших входов. В [4] был предложен альтернативный подход к факторизации, где явного умножения можно избежать. В дальнейшем этот подход обсуждался в [9].

В данной статье мы рассматриваем параллельную версию алгоритма декомпозиции из [4, 9] и сравниваем его с параллельными реализациями аналогичных алгоритмов. В разделе 2 мы сначала приводим обзор последовательного алгоритма факторизации и его вариантов. В разделе 3 описывается параллельная MIMD реализация алгоритма и далее в разделе 4 представлен количественный анализ параллельного и последовательного алгоритмов. Наконец, в разделе 5 приведены результаты тестирования нашего алгоритма на массивно-параллельной многоядерной архитектуре (Redefine).

2. ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ

Введем базовые определения и обозначения и рассмотрим последовательный алгоритм факторизации из [4, 9].

Полином $F \in \mathbb{F}_2[x_1, \dots, x_n]$ называется *фактуризуемым* если $F = F_1 \times \dots \times F_k$, где $k \geq 2$ и $F_1, \dots, F_k$ неконстантные полиномы. В противном случае F называется *неприводимым*. Полиномы $F_1, \dots, F_k$ на-

зываются *факторами* F . Важно отметить, что так как рассматриваются мультилинейные полиномы (каждая переменная может входить только в степени не более 1), факторы являются полиномами *над непересекающимися множествами переменных*. В дальнейшем мы полагаем, что полином F не имеет *тривиальных факторов* (делителей), т.е., ни x , ни $x + 1$ не делят F . Очевидно, тривиальные делители легко распознаваемы.

Пусть F — полином. Для переменной x из множества переменных $Var(F)$ полинома F и значения $a \in \{0, 1\}$, обозначим через $F_{x=a}$ полином, полученный из F посредством подстановки для x значения a . $\frac{\partial F}{\partial x}$ обозначает *формальную производную* F по переменной x . Если дана переменная z , то $z \mid F$ означает, что z делит F , т.е., z входит в каждый моном F (заметим, что это эквивалентно условию $\frac{\partial F}{\partial z} = F_{z=1}$). Если дано множество переменных Σ и моном m , то определим *проекцию* m на Σ следующим образом. Она равна 1, если m не содержит переменных из Σ , и в противном случае равна моному, полученному из m удалением всех переменных, которые не входят в Σ . *Проекция* полинома F на Σ , обозначаемая $F|_{\Sigma}$ — есть полином, полученный как сумма проекций мономов F на Σ , в которой мономы-дубликаты удалены.

2.1. Алгоритм факторизации

Алгоритм 1 описывает последовательную версию процедуры факторизации. Как уже было упомянуто, факторы булева полинома имеют непересекающиеся множества переменных. Это свойство используется в алгоритме, который пытается вычислить разбиение переменных по компонентам. Если разбиение построено, то соответствующие факторы могут быть найдены с помощью проекций исходного полинома на компоненты полученного разбиения. Далее мы полагаем, что входной полином F имеет, по крайней мере, две переменные и не содержит пары одинаковых мономов.

Algorithm 1. Последовательный алгоритм факторизации

Input Факторизуемый булев полином F

Output F_{same} и F_{other} , которые являются факторами входного полинома F

1: Выбрать произвольную переменную x , входящую в F

2: Пусть $A = \frac{\partial F}{\partial x}$, $B = F_{x=0}$

3: Пусть $\Sigma_{same} = x$, $\Sigma_{other} = \emptyset$, $F_{same} = 0$, $F_{other} = 0$

4: **for each** $y \in var(F) \setminus \{x\}$ **do**

5: Пусть $C = \frac{\partial A}{\partial y}$, $D = \frac{\partial B}{\partial y}$

6: **if** $IsEqual(A, D, B, C)$ **then**

7: $\Sigma_{other} = \Sigma_{other} \cup \{y\}$

8: **else**

9: $\Sigma_{same} = \Sigma_{same} \cup \{y\}$

10: **end if**

11: **end for**

12: Если $\Sigma_{other} = \emptyset$, тогда F неприводим.

13: Вернуть полиномы F_{same} и F_{other} как проекции F на Σ_{same} и Σ_{other} , соответственно.

Таким образом, алгоритм выбирает произвольным образом переменную x из F и разбивает множество переменных F на два множества относительно x :

- первое множество Σ_{same} содержит выбранную переменную x и соответствует фактору, представляющему неприводимый полином;
- второе множество Σ_{other} соответствует второму фактору, который допускает дальнейшую факторизацию.

Факторы F_{same} и F_{other} получаются проекциями исходного полинома на Σ_{same} и Σ_{other} , соответственно.

В строках 1–3 выбирается произвольная x из F и вычисляются полиномы A и B . A — это производная F по переменной x , а B — полином, полученный из F означиванием x нулем. Строки 4–11 занимают цикл над множеством переменных F , откуда исключена переменная x . Для переменной на очередной итерации вычисляются полиномы C и D , где C — производная A и D — производная B . Затем проверяется равенство $AD = BC$. Для проверки этого равенства вызывается процедура **IsEqual** (строка 6). Она описывается в следующем разделе.

2.2. Процедура *IsEqual*

В Алгоритме 2 представлена последовательная версия процедуры *IsEqual*.

• Процедура принимает на вход полиномы A , B , C , D и проверяет равенство $AD = BC$ посредством рекурсии.

• Строки 1–16 реализуют базовые случаи, когда равенство $AD = BC$ может быть определено простым образом.

• В строках 3–6 проверяется, делит ли переменная z полиномы A, B, C, D так, что условие в строке 4 верно. Если это не верно, то переменная z из A, B, C, D удаляется (строка 7) и выполняется проверка на равенство произведений полученных полиномов.

• В строках 17–26 процедура **IsEqual** вызывается рекурсивно на полиномах меньших размеров.

2.3. Возможность параллелизма

Суть Алгоритма 1 заключена в цикле, представленном строками 4–11. Легко заметить, что различные итерации цикла независимы. Следовательно цикл обеспечивает параллелизм уровня потока, который может быть использован для улучшения производительности. Условный блок внутри цикла в строках 6–10 может быть использован для организации параллелизма уровня задачи между многими потоками.

Algorithm 2. Последовательная процедура IsEqual

Input Булевы полиномы A, B, C, D
Output TRUE если AD равно BC и FALSE иначе.

```

1: If  $A = 0$  or  $D = 0$  then return ( $B = 0$  or  $C = 0$ )
2: If  $B = 0$  or  $C = 0$  then return FALSE
3: for each  $z$ , встречающуюся, по крайней мере, в
   одним из  $A, B, C, D$  do
4:   if  $z \mid A$  or  $z \mid D$  or  $z \mid B$  or  $z \mid C$  then
5:     return FALSE
6:   end if
7: Заменить каждый  $X \in \{A, B, C, D\}$  на  $\frac{\partial X}{\partial z}$  при
   условии  $z \mid X$ 
8: end for
9: if  $A = 1$  and  $D = 1$  then return ( $B = 1$  and  $C = 1$ )
10: end if
11: if  $B = 1$  and  $C = 1$  then return FALSE
12: end if
13: if  $A = 1$  and  $B = 1$  then return ( $D = C$ )
14: end if
15: if  $D = 1$  and  $C = 1$  then return ( $A = B$ )
16: end if
17: Выбрать переменную  $z$ 
18: if not(IsEqual( $A_{z=0}, D_{z=0}, B_{z=0}, C_{z=0}$ )) then return
   FALSE
19: end if
20: if not(IsEqual( $\frac{\partial A}{\partial z}, \frac{\partial D}{\partial z}, \frac{\partial B}{\partial z}, \frac{\partial C}{\partial z}$ )) then return FALSE
21: end if
```

```

22: if IsEqual( $\frac{\partial A}{\partial z}, B_{z=0}, A_{z=0}, \frac{\partial B}{\partial z}$ ) then return TRUE
23: end if
24: if IsEqual( $\frac{\partial A}{\partial z}, C_{z=0}, A_{z=0}, \frac{\partial C}{\partial z}$ ) then return TRUE
25: else return FALSE
26: end if
```

Множество частей Алгоритма 2 поддаются параллелизму. Проверка делимости полиномов A, B, C, D в строках 3–6 процедуры **IsEqual** может быть выполнено независимо. В строках 17–26 рекурсивные вызовы **IsEqual** независимы и представляют параллелизм уровня задачи. В следующем разделе рассматривается параллельный алгоритм, использующий эти наблюдения.

3. ПРЕДЛАГАЕМЫЙ ПОДХОД

3.1. Параллельный алгоритм факторизации

Алгоритм 3 описывает параллельную версию алгоритма факторизации. В строках 1–3 выбирается произвольная переменная x из множества переменных полинома F и вычисляются полиномы A и B . В строках 4–11 порождением нескольких потоков несколько итераций цикла выполняются параллельно. Каждый поток возвращает два множества Σ_{same}^{tid} и Σ_{other}^{tid} , специфичные для потока и обозначаемые идентификатором потока tid . В строках 12–14, множества переменных Σ_{same} и Σ_{other} вычисляются как объединение данных соответствующих разным потокам. Отметим синхронизационный барьер для всех параллельных потоков.

Algorithm 3. Параллельный алгоритм факторизации

Input Факторизуемый булев полином F
Output F_{same} и F_{other} , которые являются факторами входного полинома F

```

1: Выбрать произвольную переменную  $x$  из  $F$ 
2: Пусть  $A = \frac{\partial F}{\partial x}, B = F_{x=0}$ 
3: Пусть  $\Sigma_{same} = x, \Sigma_{other} = \emptyset, F_{same} = 0, F_{other} = 0$ 
4: for each  $y \in \text{var}(F) \setminus \{x\}$  do in parallel
5:   Пусть  $C = \frac{\partial A}{\partial y}, D = \frac{\partial B}{\partial y}$ 
6:   if IsEqual( $A, D, B, C$ ) then
7:      $\Sigma_{other}^{tid} = \Sigma_{other}^{tid} \cup \{y\}$ 
8:   else
9:      $\Sigma_{same}^{tid} = \Sigma_{same}^{tid} \cup \{y\}$ 
```

```

10: end if
11: end for
12: Ждать пока все параллельные потоки не закончатся
13:  $\Sigma_{other} = \bigcup_{tid} \Sigma_{other}^{tid}$ 
14:  $\Sigma_{same} = \bigcup_{tid} \Sigma_{same}^{tid}$ 
15: Если  $\Sigma_{other} = \emptyset$ , тогда F неприводим; остановиться.
16: Вернуть полиномы  $F_{same}$  и  $F_{other}$ , полученные как проекции F на  $\Sigma_{same}$  и  $\Sigma_{other}$ , соответственно.

```

3.2. Параллельная версия функции IsEqual

Алгоритм 4 описывает параллельную версию процедуры IsEqual. Алгоритм принимает на вход четыре полинома A, D, B, C и проверяет равенство $AD = BC$. Строки 1–21 описывают случаи, когда определение равенства $AD = BC$ тривиально.

Algorithm 4. Параллельная версия функции IsEqual

```

Input Булевы полиномы A, B, C, D
Output TRUE если  $AD = BC$  и FALSE, иначе.
1: If A = 0 or D = 0 then return (B = 0 or C = 0)
2: If B = 0 or C = 0 then return FALSE
3: for each z, входящая в хотя бы один из A, B, C, D do in parallel
4:   Положить  $flag^{tid} = \text{True}$ 
5:   if  $z \mid A$  or  $z \mid D$  or  $z \mid B$  or  $z \mid C$  then
6:     Положить  $flag^{tid} = \text{FALSE}$ 
7:   end if
8:   Заменить каждый  $X^{tid} \in \{A, B, C, D\}$  на  $\frac{\partial X^{tid}}{\partial z}$ ,
при условии  $z \mid X^{tid}$ 
9: end for
10: Ждать пока все параллельные потоки не закончатся
11: if  $\bigwedge_{tid} flag^{tid} = \text{FALSE}$  then return FALSE
12: end if
13:  $X = \bigcap_{tid} X^{tid}$ , for  $X \in \{A, B, C, D\}$ 
14: if A = 1 and D = 1 then return (B = 1 and C = 1)
15: end if
16: if B = 1 and C = 1 then return FALSE
17: end if
18: if A = 1 and B = 1 then return (D = C)
19: end if

```

```

20: if D = 1 and C = 1 return (A = B)
21: end if
22: Выбрать переменную z
23: Выполнять следующие 4 строки параллельно
24:  $x = \text{not}(\text{IsEqual}(A_{z=0}, D_{z=0}, B_{z=0}, C_{z=0}))$ 
25:  $y = \text{not}(\text{IsEqual}(\frac{\partial A}{\partial z}, \frac{\partial D}{\partial z}, \frac{\partial B}{\partial z}, \frac{\partial C}{\partial z}))$ 
26:  $z = \text{IsEqual}(\frac{\partial A}{\partial z}, B_{z=0}, A_{z=0}, \frac{\partial B}{\partial z})$ 
27:  $w = \text{IsEqual}(\frac{\partial A}{\partial z}, C_{z=0}, A_{z=0}, \frac{\partial C}{\partial z})$ 
28: Ждать пока все параллельные потоки не закончатся
29: if not(x) then return FALSE
30: end if
31: if not(y) then return FALSE
32: end if
33: if z then return TRUE
34: end if
35: if w then return TRUE
36: else return FALSE
37: end if

```

В строках 3–9 проверяется, делит ли переменная z входные полиномы A, D, B, C так, что условие в строке 5 выполнено. Если это не так, все они делятся на z для того, чтобы получить приведенные полиномы (не содержащие тривиальных делителей). Вышеупомянутые операции выполняются для каждой переменной независимо в параллельно работающих потоках. В строке 8 в каждом потоке проверяется, является ли переменная z^{tid} (tid – идентификатор потока) делителем какого-то полинома A, B, C, D . Если z^{tid} делит какой-то из полиномов A, B, C, D , то вычисляются соответствующие приведенные полиномы $A^{tid}, D^{tid}, B^{tid}, C^{tid}$. В строке 13 берется попарное пересечение соответствующих мономов полиномов из потоков $A^{tid}, D^{tid}, B^{tid}, C^{tid}$ для того, чтобы сформировать полиномы, свободные от тривиальных делителей. Пересечение двух мономов m_1, m_2 равно 1, если m_1, m_2 не имеют общих переменных, в противном случае это моном, состоящий из переменных, которые входят в оба монома m_1 и m_2 . В строках 23–27 выполняются четыре независимых рекурсивных вызова функции IsEqual в параллельных потоках. В строках 28–37 ожидается завершение всех потоков и сравнение их результатов для формирования окончательного результата. Отметим

в строках 10 и 28 синхронизирующий барьер для всех параллельных потоков.

3.3. Оценка ускорения параллельного алгоритма, основанного на распределении переменных по параллельным потокам исполнения

Реализация параллелизма в Алгоритмах 3 и 4 исходит из предположения о неограниченных возможностях использования ресурсов — параллельно запускаются все достаточно объемные секции кода и коммуникация между ними не представляет никаких сложностей. Это интересно с теоретической точки зрения для исследования глубины параллелизма. Однако, с точки зрения оценки глубины “практического” параллелизма, следует рассмотреть некоторые ограничения. В частности, фиксированное количество процессоров/ядер, доступных для исполнения программы. В этой связи можно рассмотреть вариант Алгоритма 3, в котором распараллеливание в цикле по всем переменным $u \in \text{var}(F) \setminus \{x\}$ заменяется на параллелизм по блокам разбиения множества переменных $\text{var}(F) \setminus \{x\}$. Количество блоков определяется количеством доступных процессоров/ядер. Следует отметить, что параллельные потоки исполнения могут не взаимодействовать друг с другом (иметь изолированную память) что, например, может быть привлекательным при реализации алгоритма факторизации на входящих в широкую практику NUMA-архитектурах. Вопрос повышения эффективности вычислений за счет доступа к переиспользуемому общим данным на данный момент остается открытым.

Изучая поведение алгоритма факторизации полиномов, представленных в виде списка мономов (ESOP), было обнаружено, что IsEqual работает в разы больше времени для переменных из компоненты, которая противоположна компоненте, содержащей выбранную на первом шаге переменную: доказательство, что равенство не выполняется, заканчивается быстрее, чем доказательство равенства, когда нужно довести разбиение полинома до “мельчайшего”, чтобы проверка равенства была тривиальной. Один из выводов, который из этого следует, — неприводимые полиномы будут обрабатываться быстрее, чем разложимые.

Оценим возможное ускорение в этом случае. Рассмотрим следующие параметры:

- N — количество переменных в полиноме;
- $\mu \in (0 \dots 1)$ — доля переменных из Σ_{same} и $1 - \mu$ — доля переменных из Σ_{other} среди всех N переменных;

- t — среднее время обработки переменной из Σ_{same} ;

- $tv, v > 1$, — среднее время обработки переменной из Σ_{other} ;

- P — количество доступных в параллельной версии процессоров (отметим, задачи, запущенные на них, никак не синхронизируются).

Время решения задачи факторизации (той части, где ищется разбиение на Σ_{same} и Σ_{other}) на одном процессоре

$$T_1 = \mu Nt + (1 - \mu)Ntv = Nt(\mu + (1 - \mu)v).$$

Легко видеть, что ускорение вычислений зависит от того, насколько “удачно” распределились по потокам переменные (разумным предположением является, что каждый поток обрабатывает примерно одинаковое количество переменных). В данном случае самый удачный вариант — это когда в каждом потоке воспроизводится исходная пропорция переменных из Σ_{same} и Σ_{other} :

$$T_p^{\text{best}} = \frac{T_1}{P}.$$

Таким образом, в идеале возможно ускорение в P раз (это максимально возможное ускорение).

Худших варианта два:

1. Существует поток, который обрабатывает переменные только из Σ_{other} . Тогда

$$T_p^{\text{worst}} = \frac{N}{P}tv.$$

$$\frac{T_1}{T_p^{\text{worst}}} = P \frac{(\mu + (1 - \mu)v)}{v} = P \left(1 - \mu \frac{v - 1}{v} \right).$$

Аналитическая зависимость для v пока не установлена, но на достаточно больших полиномах стабильно проявляется, что $v \gg 1$, а значит ускорение составляет примерно $P - \mu P$.

2. Все потоки кроме одного обрабатывают переменные только из Σ_{same} , а этот выделенный поток обрабатывает как переменные из Σ_{same} , т.е. $\mu > \frac{P-1}{P}$, так и все из Σ_{other} .

$$T_p^{\text{worst}} = \left(\mu - \frac{P-1}{P} \right) Nt + (1 - \mu)Ntv$$

$$\frac{T_1}{T_p^{\text{worst}}} = \frac{\mu + (1 - \mu)v}{\mu + (1 - \mu)v - \frac{P-1}{P}} \leq 1 + \frac{\mu}{(1 - \mu)v}.$$

Таблица 1. Статистика об ускорении при 2, 3 и 4 потоках

Потоков	Мин.	Макс.	Сред.	Ст. откл.	Медиана
2	1.528	1.947	1.715	0.062	1.704
3	1.836	2.700	2.198	0.093	2.196
4	2.150	3.673	2.630	0.155	2.613

Таблица 2. Результаты запуска на процессоре Xeon 2.8 GHz с 4-мя потоками

Тест	Послед.	Многопот.	Ускорен.
ITC'99	4324(s)	1441(s)	3.01
Iscas'85	7181(s)	2633(s)	2.73
EPFL Adder	1381(s)	374(s)	3.69

Видно, что при таком распределении переменных по потокам и при $v \gg 1$ рост количества процессоров не будет оказывать значительное влияние на ускорение.

После анализа ускорения при таком распараллеливании в Maple-реализации алгоритма было сделано следующее. Перед разбиением множества переменных по k потокам — разрезания массива переменных на k частей — выполняется несколько случайных перестановок этого массива с целью балансировки распределения переменных в разных частях.

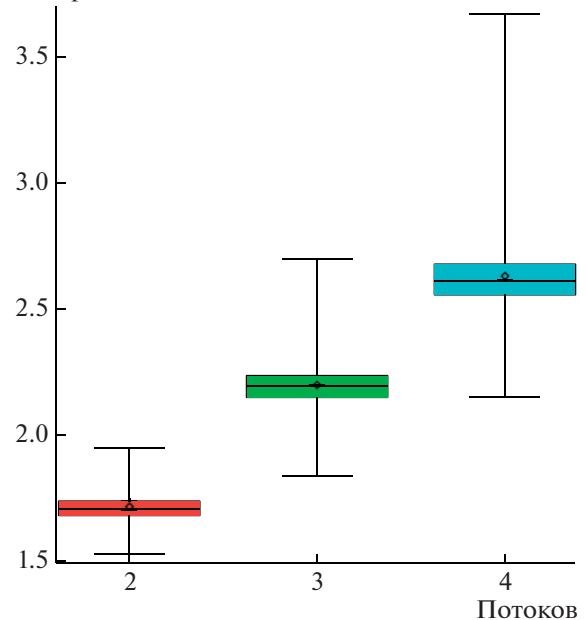
4. ЭКСПЕРИМЕНТЫ ПО РАСПАРАЛЛЕЛИВАНИЮ ОБРАБОТКИ ПЕРЕМЕННЫХ ПО ПОТОКАМ

Для экспериментальной оценки последовательного и параллельного алгоритмов, основанного на распределении проверки переменных по потокам, использовались известные бенчмарки для логического синтеза и искусственно сгенерированные (случайные) булевы полиномы.

4.1. Реализация полиномов с использованием Zero-suppressed Decision Diagrams

Для представления булевых полиномов используется ZDD из библиотеки CUDD версии 3.0.0 [10]. Его можно трактовать как представление в виде множества (сам полином) подмножеств (его мономы) некоторого конечного множества (переменные полинома). Реализация использует стандартный механизм потоков в C++. Для тестирования использовались 500 случайных полиномов со 100 переменными и 10000 монома-

Ускорение

**Рис. 1.** Ускорение при распараллеливании обработки переменных.

ми (с различными пропорциями переменных и мономов в би-компонентах разложения). В табл. 1 приведены результаты тестирования ускорения алгоритма с распараллеливанием обработки переменных на 2, 3 и 4 потока. На рис. 1 основная статистическая информация приведена в графическом виде.

4.2. Реализация полиномов в виде списков мономов

Мы использовали бенчмарки логического синтеза ITC'99 [11], Iscas'85 [12] и бенчмарк n -битного

Таблица 3. Время факторизации случайных полиномов на процессоре Xeon 2.8 GHz с 4-мя потоками

Мономов	Послед.	Многопот.	Ускорен.
10	0.023(s)	0.0074(s)	3.12
50	16.29(s)	5.07(s)	3.21
100	103.5(s)	30.44(s)	3.4
500	483.6(s)	178.1(s)	2.7
1000	1165(s)	520.9(s)	2.2
5000	1430(s)	735.11(s)	1.91
10000	12614(s)	8034(s)	1.57

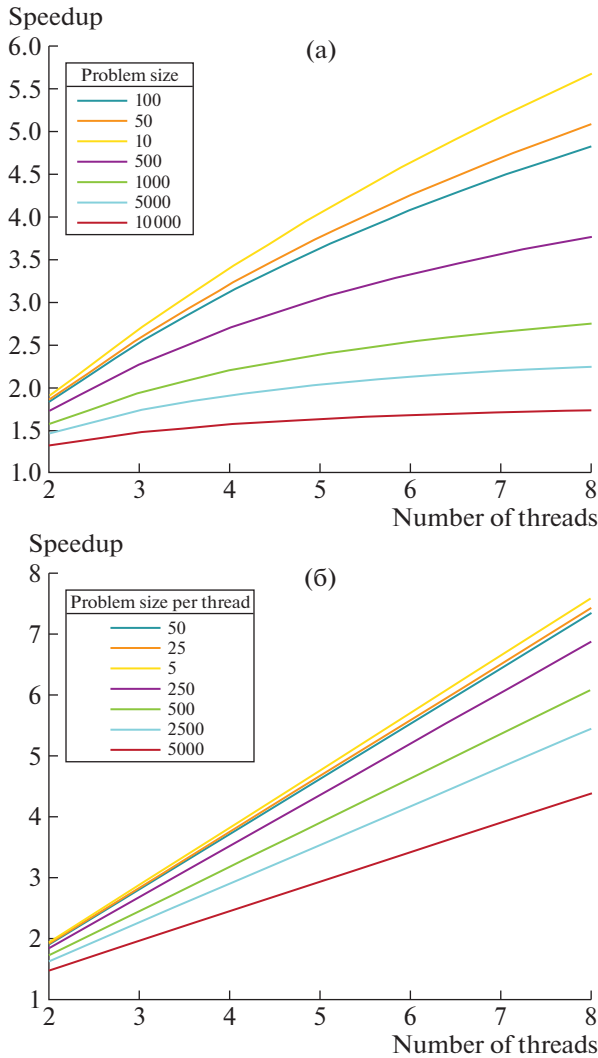


Рис. 2. (а) Ускорение за счет параллелизации по отн. к числу потоков для одного входа. (б) Ускорение за счет параллелизации по отн. к числу потоков и фикс. входа для каждого потока.

сумматора с переносом [13]. Описания схем были конвертированы из формата Verilog в СДНФ, чтобы получить соответствующие булевы полиномы. Последовательный и параллельный алгоритмы тестировались на полученных полиномах. В табл. 2 показано усредненное по 5 запускам время работы последовательного и параллельного алгоритмов на процессоре Xeon 2.8 Ghz с четырьмя потоками.

Для последовательного и параллельного алгоритмов также было проведено тестирование на случайных полиномах различной сложности (экспериментах число переменных полинома варьируется от десятков до пары сотен). В табл. 3 показано усред-

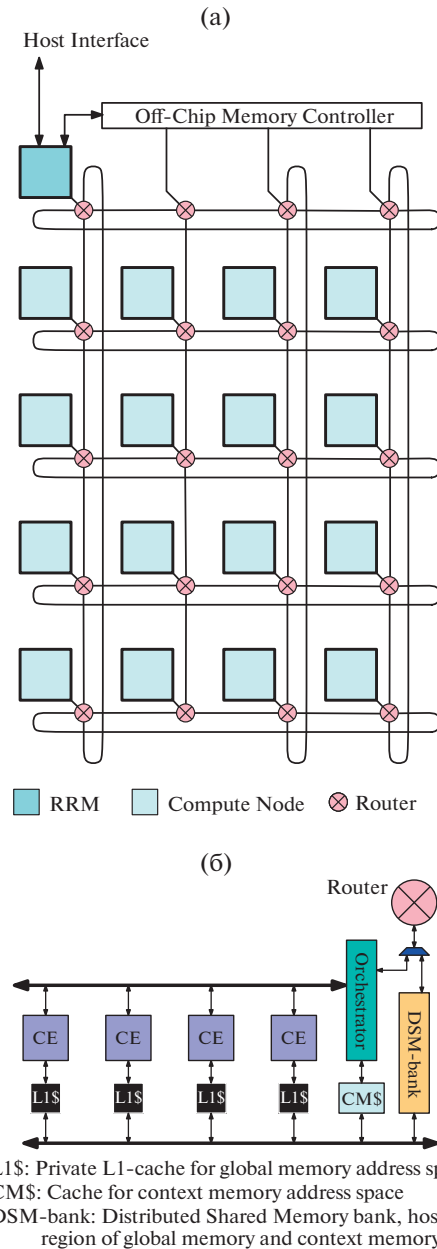


Рис. 3. (а) Архитектура REDEFINE, состоящая из 16 выч. узлов, расположенных в сеть 4 × 4 тороидальной конфигурации, включающая маршрутизаторы и менеджер ресурсов (RRM) для взаимодействия с хост-контроллером. (б) Составляющие вычислительного узла.

ненное по 5 запускам время работы последовательного и параллельного алгоритмов на процессоре Xeon 2.8 Ghz с четырьмя потоками.

График 2а показывает ускорение, достигаемое параллельным алгоритмом по отношению к числу потоков. Размер входа фиксирован для того, чтобы

проанализировать свойства сильно масштабируемости параллельного алгоритма. Можно заметить, что ускорение снижается в увеличении сложности входного полинома. С увеличением размера входа увеличивается число вызовов к узкому месту алгоритма, каковым является упрощение полиномов, что, в свою очередь, влечет снижение быстродействия.

Для результатов экспериментов, показанных на рис. 26, размер входа фиксирован для каждого из потоков, чтобы проанализировать свойства слабой масштабируемости параллельного алгоритма. Прирост скорости с увеличением числа потоков оказывается меньше максимально возможного линейного ускорения. Это обусловлено узкими местами алгоритма, требующими последовательных вычислений (упрощение полиномов), а также затратами на коммуникацию между потоками.

5. РЕАЛИЗАЦИЯ НА REDEFINE

В основе архитектуры REDEFINE [14] лежит набор вычислительных модулей, соединенных в сеть на чипе (см. иллюстрацию 3а).

REDEFINE — это ускоритель, который может настраиваться на выполнение конкретной вычислительной задачи посредством реконфигурации. Настройка конфигурации REDEFINE возможна на двух уровнях: посредством агрегации узлов в вычислительные ядра для выполнения высокоуровневых операций с множественными входами и выходами, а также на уровне специализированных функциональных элементов, представленных на слое аппаратных абстракций в виде расширенных наборов инструкций. В отличие от традиционных архитектур, расширенные наборы инструкций могут быть определены на пост-производственном этапе. Эта специфика выделяет REDEFINE на фоне широко распространенных многоядерных решений, предоставляя возможность настройки на приложения различных типов.

REDEFINE реализует исполнение в духе модели макропотоков данных. В ней исполнение представлено в виде иерархического графа потока данных (рис. 4), в котором вершинами являются гипероперации, а дуги представляют явное направление данных или порядок выполнения связанных гиперопераций. Гипероперация представляет собой макрооперацию с множественными входами и выходами. Гипероперация подлежит исполнению как только доступны все ее операнды, и все зависимости порядка исполнения или синхронизации выполнены. Помимо арифметических, связанных с чтением/записью в память и управляющих инструкций, модель исполнения REDEFINE включает

примитивы для явного обмена данными и синхронизации между гипероперациями, а также примитивы для добавления новых узлов (гиперопераций) и дуг графа в процессе исполнения. Таким образом, модель исполнения поддерживает динамический (зависимый от данных) параллелизм. В модели реализовано невытесняющее планирование гиперопераций, поэтому циклические зависимости между ними запрещены. Управляющий модуль планирует готовые гипероперации к выполнению на вычислительных узлах. Каждый узел содержит четыре вычислительных элемента, каждый из которых может выполнить ту или иную гипероперацию (рис. 26). Обмен данными между гипероперациями является односторонним, т.е. только одна и та же гипероперация может инициировать и завершить обмен. Таким образом, в условиях достаточного параллелизма обмен данными может накладываться на вычисления. По сравнению с другими гибридными моделями потока данных и управления, модель REDEFINE упрощает управление ресурсами и моделью памяти, необходимыми для поддержки параллелизма любого типа.

5.1. Алгоритм факторизации на основе гиперопераций

Algorithm 5. Алгоритм декомпозиции на основе гиперопераций

Input Факторизуемый булев полином F

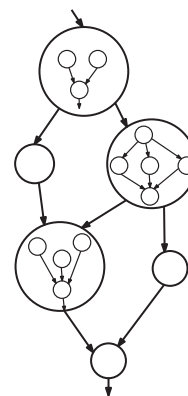


Рис. 4. Модель исполнения на уровне макропотоков данных. Исполнение представлено в виде иерархического графа потока данных, в котором вершинами являются гипероперации, а дуги представляют явное направление данных или порядок выполнения связанных гиперопераций.

Таблица 4. Параллельная факторизация случайных булевских полиномов на эмуляторе REDEFINE с использованием процессора Intel Xeon 2.8 GHz

Кол-во мономов	Последов. циклов ЦП	Многопот. циклов ЦП	Redefine циклов ЦП
30	17192×10^3	7896×10^3	6837×10^3
50	45612×10^3	14196×10^3	12320×10^3

Output F_{same} и F_{other} , которые являются факторами входного полинома F **Глобальные переменные** $\Sigma_{same}, \Sigma_{other}$

```

1: Begin HyperOp
2: Входы:  $A, B$ , переменная  $y$ 
3: Вычислить  $C = \frac{\partial A}{\partial y}, D = \frac{\partial B}{\partial y}$ 
4: if IsEqual( $A, D, B, C$ ) then
5:    $\Sigma_{other} = \Sigma_{other} \cup \{y\}$ 
6: else
7:    $\Sigma_{same} = \Sigma_{same} \cup \{y\}$ 
8: end if
9: Call Sync HyperOp
10: End HyperOp
11: Выбрать произвольную переменную  $x$  из  $F$ 
12: Let  $A = \frac{\partial F}{\partial x}, B = F_{z=0}$ 
13: Let  $\Sigma_{same} = x, \Sigma_{other} = \emptyset, F_{same} = 0, F_{other} = 0$ 
14: for each  $y \in var(F) \setminus \{x\}$  do in parallel
15:   Породить HyperOp со входами  $A, B, y$ 
16: end for
17: Ждать исполнения Sync HyperOp
18: Если  $\Sigma_{other} = \emptyset$  то  $F$  неприводим; остановиться.
19: Вернуть полиномы  $F_{same}$  и  $F_{other}$ , являющиеся проекциями  $F$  на  $\Sigma_{same}$  и  $\Sigma_{other}$ , соответственно.

```

В Алгоритме 5 приведен псевдокод алгоритма факторизации в стиле реализации на языке C с гипероперациями. Фрагмент кода, соответствующий Алгоритму 5, представлен в распечатке ниже. В ней выражения вида `__CMAAddr`, `__Sync`, `__kernel`, `__WriteCM`, `CMADDR` являются аннотациями, используемыми для REDEFINE. Строки 2–8 и 12–13 Алгоритма 5 соответствуют строкам 5–10 и 1–3 Алгоритма 1, соответственно.

В строках 14–16 Алгоритма 5 для каждой переменной y из F (отличной от x) параллельно порождаются гипероперации, которые устанавливают, какому из множеств Σ_{same} или Σ_{other} принад-

лежит y . В строках 1–10 дано определение гипероперации, которая берет на вход булевы полиномы A, B и переменную y и добавляет y в Σ_{same} или Σ_{other} . В строках 17–19 указано ожидание завершения всех гиперопераций и выдача полиномов F_{same} и F_{other} .

В ПРИЛОЖЕНИИ 1 приведена реализация алгоритма факторизации на языке C с гипероперациями. Алгоритм был протестирован на эмуляторе REDEFINE, запущенном на процессоре Intel Xeon. В табл. 4 приведено время работы алгоритма факторизации на эмуляторе REDEFINE для случайных полиномов. Реализация на REDEFINE требует наименьшее число циклов ЦП.

6. ЗАКЛЮЧЕНИЕ И ДАЛЬНЕЙШАЯ РАБОТА

В статье рассмотрена проблема факторизации булевых полиномов, лежащая в основе декомпозиции булевых функций в ДНФ и реляционных данных. В связи с этой проблемой важным является построение эффективных алгоритмов факторизации. Мы рассмотрели алгоритм факторизации булевых полиномов, предложенный в [4], и предложили параллельную MIMD реализацию этого алгоритма, который использует параллелизм уровня задачи и параллелизм уровня данных для повышения быстродействия. Эксперименты с последовательным и параллельным алгоритмами факторизации на бенчмарках логического синтеза и случайных полиномах показали существенное ускорение вследствие параллелизации. Реализация параллельного алгоритма на эмуляторе REDEFINE продемонстрировало преимущество в быстродействии благодаря массивнопараллельной многоядерной архитектуре. Модель исполнения, реализуемая архитектурой REDEFINE, основана на принципах потока данных, что позволяет избежать необходимости барьерной синхронизации. Это дает повышение скорости MIMD задач (в частности, факторизации) на архитектуре REDEFINE. Указанные реализации

будут применены для вычисления недизъюнктивной декомпозиции ДНФ [15] и табличных данных

[6], которая опирается на решение массовой задачи факторизации булевых полиномов.

7. ПРИЛОЖЕНИЕ

Листинг1: Snippet of decomposition algorithm using Hyperops

```
__hyperOp__ void
decomp ( __CMAAddr selfId ,
         __Op32 a , __Op32 b ,
         __Op32 p_s , __Op32 p_o ,
         __Op32 m, __Op32 n ,
         __Op32 i , __Op32 consumFrId )
{
    int *A=a.ptr ;
    int *B=b.ptr ;
    int * partition_same =p_s.ptr ;
    int * partition_o the r=p_o.ptr ;
    int I=i.i32;
    int n=n.i32;
    int m=m.i32;
    int J=0;
    int *C, *D;
    __CMAAddr c onf r Id=consumFrId.cmAddr ;
    for ( I=0; I<n ; I++) {
        *( partition_same+J )=0;
        *( partition_o the r+J )=0;
    }
    for ( I=0; I<m; I++) {
        for ( J=0; J<n ; J++) {
            *(C+I * columns+J )=0;
            *(D+I * columns+J )=0;
        }
    }
    derivative (A,B,C, i ) ;
    derviative (A,B,D, i ) ;
    if ( I sEqual (A,D,B,C) )
        *( partition_o the r+i ) =1;
    else
        *( partition_same+i ) =1;
    __Sync( confrId , =1);
}

__kernel int
decomp_start ( int *A, int *B,
               int * partition_same ,
```

```

        int * partition_othe r ,
        int N)
{
    int i =0;
    static int counter=0;
    __CMAddr decompFr ;
    __CMAddr syncFr=__CreateInst(&smd_Sync ) ;
    __WriteCM(CMADDR( syncFr , 1 5 ) ,N=1);

    for ( i = 1 ; i<N; i++) {
        decompFr=__CreateInst(&smd_decomp ) ;
        __WriteCM(CMADDR( decompFr , 0 ) ,
            ( void * ) (A) ) ;
        __WriteCM(CMADDR( decompFr , 1 ) ,
            ( void * ) (B) ) ;
        __WriteCM(CMADDR( decompFr , 2 ) ,
            ( void * ) ( partition_same ) ) ;
        __WriteCM(CMADDR( decompFr , 3 ) ,
            ( void * ) ( partition_o the r ) ) ;
        __WriteCM(CMADDR( decompFr , 4 ) , M) ;
        __WriteCM(CMADDR( decompFr , 5 ) , N) ;
        __WriteCM(CMADDR( decompFr , 6 ) , i ) ;
        __WriteCM(CMADDR( decompFr , 7 ) ,
CMADDR( syncFr , 1 5 ) ) ;
    }
    return 0 ;
}

```

СПИСОК ЛИТЕРАТУРЫ

1. von zur Gathen J., Gerhard J. Modern Computer Algebra. Third edition. New York, NY, USA: Cambridge University Press, 2013. P. 812.
2. Жегалкин И.И. Арифметизация символической логики // Математический сборник. 1928. Т. 35. № 1. С. 311–377.
3. Muller D.E. Application of Boolean algebra to switching circuit design and to error detection // IRE Transactions on Electronic Computers. 1954. V. EC-3. P. 6–12.
4. Емельянов П.Г., Пономарев Д.К. Алгоритмические вопросы конъюнктивной декомпозиции булевых формул // Программирование. 2015. Т. 41. № 3. С. 162–169.
5. Emelyanov P., Ponomaryov D. On tractability of disjoint AND–decomposition of boolean formulas // Proceedings of the PSI 2014: 9th Ershov Informatics Conference. V. 8974 of *Lecture Notes in Computer Science*. Springer, 2015. P. 92–101.
6. Emelyanov P. On two kinds of dataset decomposition // Proceedings of the 18th International Conference on Computational Science (ICCS 2018), Part II. V. 10861 of *Lecture Notes in Computer Science*. Springer, 2018. P. 171–183.
7. Emelyanov P., Ponomaryov D. Cartesian decomposition in data analysis // Proceedings of the Siberian Symposium on Data Science and Engineering (SSDSE 2017). 2017. P. 55–60.
8. Shpilka A., Volkovich I. On the relation between polynomial identity testing and finding variable disjoint factors // Proceedings of the 37th International Colloquium on Automata, Languages and Programming. Part 1 (ICALP 2010). V. 6198 of *Lecture Notes in Computer Science*. Springer, 2010. P. 408–419.
9. Emelyanov P., Ponomaryov D. On a polytime factorization algorithm for multilinear polynomials over F_2 // Proceedings of the 20th International Workshop on Computer Algebra in Scientific Computing (CASC

- 2018). V. 11077 of *Lecture Notes in Computer Science*. Springer, 2018. P. 164–176.
10. *Somenzi F.* CUDD: CU Decision Diagram package // <https://github.com/ivmai/cudd>. Accessed 2019-12-11.
11. *Corno F., Reorda M. S., Squillero G.* RT-level ITC'99 benchmarks and first ATPG results // *IEEE Design & Test of Computers*. 2000. V. 17. № 3. P. 44–53.
12. *Hansen M.C., Yalcin H., Hayes J.P.* Unveiling the IS-CAS-85 benchmarks: A case study in reverse engineering // *IEEE Design Test of Computers*. 1999. V. 16. № 3. P. 72–80.
13. *Fišer P., Schmidt J.* A prudent approach to benchmark collection // *Proceedings of the 12th International Workshop on Boolean Problems (IWSBP'2016)*. 2016. P. 129–136.
14. *Redefine* — reconfigurable silicon core description // <http://morphing.in/redefine>. Accessed 2018-12-07.
15. *Ponomaryov D.* A polynomial time delta-decomposition algorithm for positive DNFs // *Proceedings of the 14th International Computer Science Symposium in Russia (CSR'2019)*. V. 11532 of *Lecture Notes in Computer Science*. Springer, 2019. P. 325–336.

СИМВОЛЬНО-АНАЛИТИЧЕСКИЕ МЕТОДЫ ИССЛЕДОВАНИЯ
ПОЛОЖЕНИЙ РАВНОВЕСИЯ
СПУТНИКА НА КРУГОВОЙ ОРБИТЕ© 2021 г. С. А. Гутник^{a,b,*}, В. А. Сарычев^{c,**}^aМосковский государственный институт международных отношений,
119454 Москва, Проспект Вернадского, 76, Россия^bМосковский физико-технический институт,
141701 Долгопрудный, Институтский переулок, 9, Россия^cИнститут прикладной математики им. М.В. Келдыша РАН,
125047 Москва, Миусская пл., 4, Россия

*E-mail: s.gutnik@inno.mgimo.ru

**E-mail: vas31@rambler.ru

Поступила в редакцию 27.06.2020 г.

После доработки 21.07.2020 г.

Принята к публикации 02.09.2020 г.

В статье предложены два наиболее простых метода определения положений равновесия спутника, движущегося в центральном ньютоновом силовом поле по круговой орбите под действием гравитационного момента. В первом методе применялись подходы линейной алгебры, во втором алгоритмы компьютерной алгебры. Положения равновесия спутника в орбитальной системе координат при заданных значениях главных центральных моментов инерции определяются корнями системы нелинейных алгебраических уравнений. Для определения равновесных решений проводилась декомпозиция системы алгебраических уравнений с применением методов линейной алгебры и алгоритмов построения базисов Гребнера. Положения равновесия спутника определялись путем исследования числа действительных корней алгебраических уравнений из полученных базисов Гребнера. С использованием предложенного подхода показано, что спутник с неравными главными центральными моментами инерции имеет на круговой орбите 24 положения равновесия.

DOI: 10.31857/S0132347421020060

1. ВВЕДЕНИЕ

В работе представлены результаты применения методов линейной и компьютерной алгебры для исследования положений равновесия спутника — твердого тела, движущегося в центральном ньютоновом силовом поле по круговой орбите.

Данная задача решалась довольно сложными методами в 60-х годах прошлого века. Благодаря решению задачи о положениях равновесия спутника большое распространение получили гравитационные системы ориентации, принцип работы которых основан на том, что в центральном ньютоновом поле сил спутник с неравными главными центральными моментами инерции имеет на круговой орбите 24 положения равновесия, четыре из которых являются устойчивыми [1–3]. Подробное рассмотрение динамики спутников с гравитационными системами ориентации представлено в [4].

Исследование движения спутника в центральном ньютоновом силовом поле по круговой орби-

те под действием гравитационного момента представляет значительный практический интерес для создания систем управления ориентацией искусственных спутников Земли. Важным свойством гравитационных систем ориентации является возможность функционировать на орбите продолжительное время без расходования энергии и (или) рабочего тела.

В настоящей статье проводится подробное исследование положений равновесия спутника с применением алгебраических методов, которые ранее успешно использовались для изучения положений равновесия спутника-гиростата в работах [5–7], динамики спутника с аэродинамической системой ориентации [8], динамики движения связки двух тел [9]. Положения равновесия спутника определяются действительными корнями системы алгебраических уравнений. Для определения равновесных решений проводилась декомпозиция системы алгебраических уравнений с применением методов линейной алгебры и

алгоритмов построения базисов Гребнера [10, 11]. Для исследования применялась система компьютерной алгебры Maple [12]. Найдены все положения равновесия спутника на круговой орбите.

Системы компьютерной алгебры широко используются при решении задач небесной механики, в частности, при исследовании задачи трех тел [13]. Так, с применением символьных вычислений были получены новые результаты при рассмотрении классической задачи трех тел с переменными массами в общем случае и поиске положений равновесия в ограниченной задаче четырех тел в работах [14–17].

2. УРАВНЕНИЯ ДВИЖЕНИЯ

Рассмотрим движение спутника — твердого тела относительно центра масс на круговой орбите под действием гравитационного момента. Для записи уравнений движения введем две правые прямоугольные системы координат с началом в центре масс O спутника. В орбитальной системе координат $OXYZ$ ось OZ направлена вдоль радиуса-вектора, соединяющего центры масс Земли и спутника, ось OX направлена вдоль вектора линейной скорости центра масс O спутника. Тогда ось OY будет направлена вдоль нормали к плоскости орбиты. В связанной со спутником системе координат $Oxyz - Ox, Oy, Oz$ — главные центральные оси инерции спутника.

Определим ориентацию системы координат $Oxyz$ относительно орбитальной системы координат с использованием самолетных углов тангажа α , рыскания β и крена γ . Направляющие косинусы осей Ox, Oy, Oz в орбитальной системе координат выражаются через самолетные углы с помощью соотношений [4]:

$$\begin{aligned} a_{11} &= \cos(x, X) = \cos \alpha \cos \beta, \\ a_{12} &= \cos(y, X) = \sin \alpha \sin \gamma - \cos \alpha \sin \beta \cos \gamma, \\ a_{13} &= \cos(z, X) = \sin \alpha \cos \gamma + \cos \alpha \sin \beta \sin \gamma, \\ a_{21} &= \cos(x, Y) = \sin \beta, \\ a_{22} &= \cos(y, Y) = \cos \beta \cos \gamma, \\ a_{23} &= \cos(z, Y) = -\cos \beta \sin \gamma, \\ a_{31} &= \cos(x, Z) = -\sin \alpha \cos \beta, \\ a_{32} &= \cos(y, Z) = \cos \alpha \sin \gamma + \sin \alpha \sin \beta \cos \gamma, \\ a_{33} &= \cos(z, Z) = \cos \alpha \cos \gamma - \sin \alpha \sin \beta \sin \gamma. \end{aligned} \quad (1)$$

При малых колебаниях спутника углу тангажа соответствует поворот вокруг оси OY , углу рыскания — поворот вокруг оси OZ , углу крена — поворот вокруг оси OX .

Тогда уравнения движения спутника относительно центра масс запишутся в виде [18]:

$$\begin{aligned} A\dot{p} + (C - B)qr - 3\omega_0^2(C - B)a_{32}a_{33} &= 0, \\ B\dot{q} + (A - C)rp - 3\omega_0^2(A - C)a_{33}a_{31} &= 0, \\ C\dot{r} + (B - A)pq - 3\omega_0^2(B - A)a_{31}a_{32} &= 0, \\ p &= \bar{p} + \omega_0 a_{21}, \quad \bar{p} = \dot{\alpha} a_{21} + \dot{\gamma}, \\ q &= \bar{q} + \omega_0 a_{22}, \quad \bar{q} = \dot{\alpha} a_{22} + \dot{\beta} \sin \gamma, \\ r &= \bar{r} + \omega_0 a_{23}, \quad \bar{r} = \dot{\alpha} a_{23} + \dot{\beta} \cos \gamma. \end{aligned} \quad (2)$$

В уравнениях (2), (3) A, B, C — главные центральные моменты инерции спутника; p, q, r — проекции абсолютной угловой скорости спутника на оси Ox, Oy, Oz ; ω_0 — угловая скорость движения центра масс спутника по круговой орбите. Точкой обозначено дифференцирование по времени t .

Для системы уравнений движения (2), (3) справедлив обобщенный интеграл энергии [19, 20]

$$\begin{aligned} &\frac{1}{2}(A\bar{p}^2 + B\bar{q}^2 + C\bar{r}^2) + \\ &+ \frac{3}{2}\omega_0^2((A - C)a_{31}^2 + (B - C)a_{32}^2) + \\ &+ \frac{1}{2}\omega_0^2((B - A)a_{21}^2 + (B - C)a_{23}^2) = \text{const}. \end{aligned} \quad (4)$$

3. ПОЛОЖЕНИЯ РАВНОВЕСИЯ

Пусть все три главных центральные моменты инерции спутника различны ($A \neq B \neq C$), тогда, положив в уравнениях (2) и (3) $\alpha = \alpha_0 = \text{const}$, $\beta = \beta_0 = \text{const}$, $\gamma = \gamma_0 = \text{const}$, получим уравнения

$$\begin{aligned} a_{22}a_{23} - 3a_{32}a_{33} &= 0, \\ a_{23}a_{21} - 3a_{33}a_{31} &= 0, \\ a_{21}a_{22} - 3a_{31}a_{32} &= 0, \end{aligned} \quad (5)$$

позволяющие определить положения равновесия спутника в орбитальной системе координат. С учетом (1) систему (5) можно рассматривать как систему трех уравнений с неизвестными α_0, β_0 и γ_0 .

Другой более удобный для исследования способ замыкания уравнений (5) заключается в добавлении шести условий ортогональности направляющих косинусов

$$\begin{aligned} a_{21}^2 + a_{22}^2 + a_{23}^2 - 1 &= 0, \\ a_{31}^2 + a_{32}^2 + a_{33}^2 - 1 &= 0, \\ a_{21}a_{31} + a_{22}a_{32} + a_{23}a_{33} &= 0. \end{aligned} \quad (6)$$

Уравнения (5) и (6) образуют замкнутую алгебраическую систему уравнений относительно шести направляющих косинусов, определяющих положения равновесия спутника. Для этой системы уравнений ставится следующая задача: определить все девять направляющих косинусов, т.е. все положения равновесия спутника в орбитальной системе координат. После нахождения шести направляющих косинусов a_{21} , a_{22} , a_{23} , a_{31} , a_{32} , a_{33} оставшиеся направляющие косинусы a_{11} , a_{12} , a_{13} , с учетом того, что каждый элемент направляющих косинусов равен своему алгебраическому дополнению, можно определить по формулам

$$\begin{aligned} a_{11} &= a_{22}a_{33} - a_{23}a_{32}, \\ a_{12} &= a_{23}a_{31} - a_{21}a_{33}, \\ a_{13} &= a_{21}a_{32} - a_{22}a_{31}. \end{aligned} \quad (7)$$

Величины a_{11} , a_{12} , a_{13} не обращаются в нуль одновременно.

Решения системы уравнений (5), (6) неоднократно исследовались в научной литературе. Так в работах [1, 2] было показано существование 24 положений равновесия спутника в орбитальной системе координат. В работах [1, 2] решения системы (5) были определены с использованием выражений направляющих косинусов через эйлеровы углы. В явном и довольно сложном виде эти положения равновесия были получены в работе [3].

4. ИССЛЕДОВАНИЕ ПОЛОЖЕНИЙ РАВНОВЕСИЯ СПУТНИКА

Уравнения (5), (6) позволяют определить все положения равновесия спутника. Для получения равновесных решений используем вначале подход, основанный на методах линейной алгебры.

Рассмотрим второе и третье уравнения системы (5) относительно переменных a_{21} , a_{31} . Эти уравнения образуют однородную подсистему, определитель которой равен $\Delta_1 = 3(a_{22}a_{33} - a_{23}a_{32}) = 3a_{11}$. Если $a_{11} \neq 0$, то $a_{21} = 0$, $a_{31} = 0$ и из уравнений (7) следует $a_{12} = a_{13} = 0$, $a_{11}^2 = 1$.

Уравнения (5), (6) с учетом выражений $a_{22} = a_{11}a_{33} - a_{13}a_{31} = a_{11}a_{33}$, $a_{23} = a_{12}a_{31} - a_{11}a_{32} = -a_{11}a_{32}$, полученных из условий ортогональности, могут быть записаны в более простом виде

$$\begin{aligned} a_{32}a_{33} &= 0, \\ a_{32}^2 + a_{33}^2 &= 1. \end{aligned} \quad (8)$$

Из (8) следует, что либо $a_{32} = 0$, $a_{33}^2 = 1$, либо $a_{32}^2 = 1$, $a_{33} = 0$. Таким образом, в случае $a_{11}^2 = 1$ получим 8

равновесных решений системы (5), (6), входящих в следующие две группы:

$$a_{11}^2 = a_{33}^2 = 1, \quad a_{22} = a_{11}a_{33}, \quad (9)$$

$$a_{12} = a_{13} = a_{21} = a_{23} = a_{31} = a_{32} = 0;$$

$$a_{11}^2 = a_{32}^2 = 1, \quad a_{23} = -a_{11}a_{32}, \quad (10)$$

$$a_{12} = a_{13} = a_{21} = a_{22} = a_{31} = a_{33} = 0.$$

Аналогично проведенному выше исследованию для случая $a_{11}^2 = 1$ рассмотрим первое и третье уравнения системы (5) относительно переменных a_{22} , a_{32} . Эти уравнения образуют однородную подсистему, определитель которой равен $\Delta_2 = -3(a_{23}a_{31} - a_{21}a_{33}) = -3a_{12}$. Если $a_{12} \neq 0$, то $a_{22} = 0$, $a_{32} = 0$ и из уравнений (7) следует $a_{11} = a_{13} = 0$, $a_{12}^2 = 1$. При этих условиях получим из уравнений (5), (6) следующие 8 равновесных решений, входящих в две группы:

$$a_{12}^2 = a_{33}^2 = 1, \quad a_{21} = -a_{12}a_{33}, \quad (11)$$

$$a_{11} = a_{13} = a_{22} = a_{23} = a_{31} = a_{32} = 0;$$

$$a_{12}^2 = a_{31}^2 = 1, \quad a_{23} = a_{12}a_{31}, \quad (12)$$

$$a_{11} = a_{13} = a_{22} = a_{23} = a_{31} = a_{32} = 0.$$

Рассматривая далее первое и второе уравнения системы (5) относительно переменных a_{23} , a_{33} как однородную подсистему, определитель которой равен $\Delta_3 = -3(a_{21}a_{32} - a_{22}a_{31}) = 3a_{13}$, получим при условии $a_{13} \neq 0$ равенства $a_{23} = 0$, $a_{33} = 0$. Тогда из уравнений (7) следует, что $a_{11} = a_{12} = 0$, $a_{13}^2 = 1$. При этих условиях получим из уравнений (5), (6) следующие 8 равновесных решений, входящих в две группы:

$$a_{13}^2 = a_{32}^2 = 1, \quad a_{21} = a_{13}a_{32}, \quad (13)$$

$$a_{11} = a_{12} = a_{22} = a_{23} = a_{31} = a_{33} = 0;$$

$$a_{13}^2 = a_{31}^2 = 1, \quad a_{22} = -a_{13}a_{31}, \quad (14)$$

$$a_{11} = a_{12} = a_{21} = a_{23} = a_{32} = a_{33} = 0.$$

Отметим, что каждая из 6 групп решений (9)–(14) определяет 4 равновесных решения (4 положения равновесия в орбитальной системе координат). Все эти 24 решения соответствуют различным случаям совпадения главных центральных осей инерции спутника с осями орбитальной системы координат. Таким образом, получим, что спутник с различными главными центральными моментами инерции имеет на круговой орбите 24 положения равновесия.

5. ИССЛЕДОВАНИЕ ПОЛОЖЕНИЙ РАВНОВЕСИЯ СПУТНИКА С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ КОМПЬЮТЕРНОЙ АЛГЕБРЫ

Для нахождения решений системы алгебраических уравнений (5), (6) рассматривалась возможность применения алгоритмов построения базисов Гребнера [10, 11]. Метод построения базиса Гребнера представляет собой алгоритмическую процедуру для полного приведения задачи в случае системы полиномов от многих переменных к рассмотрению полинома от одной переменной. Алгоритмы построения базиса Гребнера реализованы в большинстве современных систем компьютерной алгебры. Исследование проводилось с использованием реализованного в системе компьютерной алгебры Maple 17 [21] пакета построения базисов Гребнера Groebner[Basis]. Построим базис Гребнера для 6 полиномов f_i , которые представляют собой левые части уравнений системы (5), (6) с 6 направляющими косинусами a_{ij} ($i = 2, 3, j = 1, 2, 3$), используя опцию лексикографического упорядочения по переменным plex:

`G:=map(factor, Groebner[Basis]([f1,...,f6], plex(a21,...,a33)))`.

При использовании данной команды система выполняет алгоритм построения базиса Гребнера FGLM, разработанный Фожером, Джани, Лазардом и Морой (Faugere, Gianni, Lazard, Mora) [22]. Построенный базис Гребнера состоит из 18 полиномов следующего вида:

$$\begin{aligned}
 &1. a_{33}(a_{33}^2 - 1) = 0, \quad 2. a_{32}(a_{32}^2 - 1) = 0, \\
 &3. a_{23}(a_{23}^2 - 1) = 0, \quad 4. a_{22}(a_{22}^2 - 1) = 0, \\
 &5. a_{21}^2 + a_{22}^2 + a_{23}^2 - 1 = 0, \\
 &6. a_{31}^2 + a_{32}^2 + a_{33}^2 - 1 = 0, \\
 &7. a_{31}(a_{22}^2 + a_{23}^2 - 1) = 0, \\
 &8. a_{21}(a_{32}^2 + a_{33}^2 - 1) = 0, \\
 &9. 1 - a_{23}^2(1 - a_{32}^2) - \\
 &\quad - a_{22}^2(1 - a_{33}^2) - a_{33}^2 - a_{32}^2 = 0, \\
 &10. a_{32}a_{33} = 0, \quad 11. a_{31}a_{33} = 0, \quad 12. a_{31}a_{32} = 0, \\
 &13. a_{23}a_{33} = 0, \quad 14. a_{22}a_{32} = 0, \quad 15. a_{22}a_{23} = 0, \\
 &16. a_{22}a_{21} = 0, \quad 17. a_{21}a_{23} = 0, \quad 18. a_{21}a_{31} = 0.
 \end{aligned} \tag{15}$$

Рассмотрим первое уравнение (15) по переменной a_{33} : $a_{33}(a_{33}^2 - 1) = 0$. При условии $a_{33}^2 = 1$ ($a_{31} = a_{32} = a_{23} = 0$) из уравнений (5) следуют равенства $a_{22}a_{23} = 0$, $a_{22}a_{21} = 0$, $a_{21}a_{23} = 0$. Эти равенства также входят в соотношения 15, 16 и 17 базиса Гребнера (15). В этом случае возможны 8 реше-

ний, когда либо $a_{21}^2 = 1$, либо $a_{22}^2 = 1$, которые в точности совпадают с рассмотренными выше группами решений (9) и (11).

Из второго уравнения (15) также можно получить две группы по 4 решения при условии $a_{32}^2 = 1$ ($a_{31} = a_{33} = a_{22} = 0$), когда либо $a_{21}^2 = 1$, либо $a_{23}^2 = 1$, которые также следуют из третьего равенства (15) и совпадают с рассмотренными выше группами решений (10) и (13).

Построим далее базис Гребнера, используя другой лексикографический порядок, чтобы получить полином по переменной a_{31} –

`plex(a21, a22, a23, a32, a33, a31)` :

$$\begin{aligned}
 &1. a_{31}(a_{31}^2 - 1) = 0, \quad 2. a_{32}(a_{32}^2 - 1) = 0, \\
 &3. a_{23}(a_{23}^2 - 1) = 0, \quad 4. a_{22}(a_{22}^2 - 1) = 0, \\
 &5. a_{21}^2 + a_{22}^2 + a_{23}^2 - 1 = 0, \\
 &6. a_{31}^2 + a_{32}^2 + a_{33}^2 - 1 = 0, \\
 &7. a_{31}(a_{22}^2 + a_{23}^2 - 1) = 0, \\
 &8. a_{23}(a_{31}^2 + a_{32}^2 - 1) = 0, \quad 9. a_{21}a_{31} = 0, \\
 &10. a_{32}a_{33} = 0, \quad 11. a_{31}a_{33} = 0, \quad 12. a_{31}a_{32} = 0, \\
 &13. a_{23}a_{33} = 0, \quad 14. a_{22}a_{32} = 0, \\
 &15. a_{22}a_{23} = 0, \quad 16. a_{22}a_{21} = 0, \quad 17. a_{21}a_{23} = 0.
 \end{aligned} \tag{16}$$

Вариант, когда $a_{31} = 0$ был рассмотрен в предыдущем случае. Рассмотрим далее случай $a_{31}^2 = 1$ ($a_{32} = a_{33} = a_{21} = 0$). Из уравнений (5) следуют равенства $a_{22}a_{23} = 0$, $a_{22}a_{21} = 0$, $a_{21}a_{23} = 0$. Эти равенства также входят в соотношения 15, 16 и 17 базиса Гребнера (16). В этом случае возможны 8 решений, когда либо $a_{22}^2 = 1$, либо $a_{23}^2 = 1$, которые следуют из третьего и четвертого равенства (16). Эти решения в точности совпадают с рассмотренными выше группами решений (12) и (14).

Из построенных базисов Гребнера следует, что в каждый из 3-х случаев, когда $a_{31}^2 = 1$, $a_{32}^2 = 1$, $a_{33}^2 = 1$, входит 8 положений равновесия спутника. Всего существуют 24 положения равновесия спутника на круговой орбите (9)–(14). Такой же результат можно получить, построив базис Гребнера, применяя лексикографический порядок по переменным a_{21} , a_{22} , a_{23} и рассмотрев случаи, когда $a_{21}^2 = 1$, $a_{22}^2 = 1$, $a_{23}^2 = 1$.

Используя левую часть обобщенного интеграла энергии (4) в качестве функции Ляпунова, можно показать, что при условии

$$B > A > C \quad (17)$$

одно из решений (9) ($a_{11} = a_{22} = a_{33} = 1$), $a_{12} = a_{13} = a_{21} = a_{23} = a_{31} = a_{32} = 0$ будет устойчивым [19, 20]. Легко убедиться в том, что достаточным условиям устойчивости (17) удовлетворяют и оставшиеся три решения (9). Таким образом, из 24 положений равновесия существуют четыре устойчивых равновесия, для которых ось максимального момента инерции спутника совпадает с нормалью к плоскости орбиты, а ось минимального момента инерции спутника совпадает с радиусом-вектором орбиты. Решения (10)–(14) условиям (17) не удовлетворяют.

6. ЗАКЛЮЧЕНИЕ

В работе проведено исследование положений равновесия спутника, движущегося по круговой орбите, с применением методов линейной и компьютерной алгебры.

Метод построения базиса Гребнера является универсальным методом, применяемым для решения систем алгебраических уравнений. Существенным ограничением для применения данного метода является экспоненциальная сложность алгоритмов вычисления базиса Гребнера. В нашем случае время построения базиса Гребнера на персональном компьютере с 8 Гигабайт оперативной памяти и процессором Intel Core i7 2,8 ГГц составило всего около 0.01 сек. Анализ полиномов, входящих в базис Гребнера, позволяет только на основе их вида сделать вывод, что исходная система алгебраических уравнений (5), (6) имеет лишь такие решения, когда направляющие косинусы, определяющие ориентацию спутника, могут принимать значения либо 0, либо 1, то есть возможны только такие решения, когда оси орбитальной и связанной систем совпадают.

В первом методе задача была решена на основе выделения в исходной системе линейных подсистем. При этом нам приходилось делать предположение об отличии от нуля определителей этих линейных подсистем. Случаи равенства нулю данных определителей, когда возникает возможность параметрических решений нами не рассматривалась. При использовании метода построения базиса Гребнера такая задача не возникает.

Применение методов компьютерной алгебры позволяет в достаточно простой форме получить решение классической задачи механики космического полета.

СПИСОК ЛИТЕРАТУРЫ

1. *Sarychev V.A.* Asymptotically Stable Stationary Rotational Motions of a Satellite. Proc. 1st IFAC Symp. on Automatic Control in Space. N. Y. Plenum Press, 1966. P. 277–286.
2. *Likins P.W., Roberson R.E.* Uniqueness of Equilibrium Attitudes for Earth-Pointing Satellites // J. Astronaut Sci. 1966. V. 13. № 2. P. 87–88.
3. *Белецкий В.В.* Движение спутника относительно центра масс в гравитационном поле. М.: изд. Московского университета, 1975. 308 с.
4. *Сарычев В.А.* Вопросы ориентации искусственных спутников. Итоги науки и техники. Серия Исследование космического пространства. М.: ВИНТИ, 1978. Т. 11. 224 с.
5. *Гутник С.А., Сантуш Л., Сарычев В.А., Силва А.* Динамика спутника-гиростата, подверженного действию гравитационного момента. Положения равновесия и их устойчивость // Изв. РАН. ТИСУ. 2015. № 3. С. 142–155.
6. *Гутник С.А., Сарычев В.А.* Символьно-численные методы исследования положений равновесия спутника-гиростата // Программирование. 2014. № 3. С. 49–58.
7. *Гутник С.А., Сарычев В.А.* Применение методов компьютерной алгебры для исследования стационарных движений спутника-гиростата // Программирование. 2017. № 2. С. 35–44.
8. *Gutnik S.A., Sarychev V.A.* Symbolic-numeric simulation of satellite dynamics with aerodynamic attitude control system. Lect. Notes Comput. Sci., Springer, Cham. 2018. V. 11077. P. 214–229.
9. *Гутник С.А., Сарычев В.А.* Применение методов компьютерной алгебры для исследования динамики системы двух связанных тел на круговой орбите // Программирование. 2019. № 2. С. 32–40.
10. *Buchberger B.* Theoretical basis for the reduction of polynomials to canonical forms // SIGSAM Bull. 1976. V. 10. № 3. P. 19–29.
11. *Бухбергер Б.* Базисы Грёбнера. Алгоритмический метод в теории полиномиальных идеалов. Компьютерная алгебра. Символьные и алгебраические вычисления. М.: Мир, 1986. С. 331–372.
12. *Char B.W., Geddes K.O., Gonnet G.H., Monagan M.B., Watt S.M.* Maple Reference Manual. Watcom Publications Limited, Waterloo, Canada, 1992.
13. *Брюно А.Д.* Ограниченная задача трех тел. Плоские периодические орбиты. М.: Наука, 1990. 295 с.
14. *Прокопеня А.Н., Минглибаев М.Дж., Маемерова Г.М.* Символьные вычисления в исследованиях проблемы трех тел с переменными массами // Программирование. 2014. № 2. С. 51–59.
15. *Прокопеня А.Н., Минглибаев М.Дж., Маемерова Г.М., Иманова Ж.У.* Исследование ограниченной задачи трех тел с переменными массами методами компьютерной алгебры // Программирование. 2017. № 5. С. 18–23.

16. *Прокопеня А.Н., Минглибаев М.Дж., Шомиекова С.А.* Применение компьютерной алгебры в исследованиях двухпланетной задачи трех тел с переменными массами // Программирование. 2019. № 2. С. 58–65.
17. *Будько Д.А., Прокопеня А.Н.* Символьно-численные методы поиска положений равновесия в ограниченной задаче четырех тел // Программирование. 2013. № 2. С. 30–37.
18. *Белецкий В.В.* Движение искусственного спутника относительно центра масс. М.: Наука, 1965. 416 с.
19. *Белецкий В.В.* Движение искусственного спутника Земли относительно центра масс // Искусственные спутники Земли. 1958. Вып. 1. С. 25–43.
20. *Белецкий В.В.* О либрации спутника // Искусственные спутники Земли. 1959. Вып. 3. С. 13–31.
21. Maple online help: <http://www.maplesoft.com/support/help/>
22. *Faugere J., Gianni P., Lazard P., Mora T.* Efficient computation of zero-dimensional Grobner bases by change of ordering // J. Symbolic Comput. 1993. V. 16. P. 329–344.

МОДЕЛИРОВАНИЕ ДИНАМИКИ КВАНТОВОЙ ЗАПУТАННОСТИ В КОНЕЧНОЙ КВАНТОВОЙ МЕХАНИКЕ: КОМПЬЮТЕРНО-АЛГЕБРАИЧЕСКИЙ ПОДХОД

© 2021 г. В. В. Корняк^{а,*}

^а Объединенный институт ядерных исследований,
141980 Дубна Московской области, Россия

*E-mail: vkornyak@gmail.com

Поступила в редакцию 01.07.2020 г.

После доработки 20.07.2020 г.

Принята к публикации 12.09.2020 г.

Рассматривается поведение квантовой запутанности в процессе унитарной эволюции в конструктивных моделях многокомпонентных квантовых систем. Описываются группы симметрий квантовых систем, допускающих возникновение геометрических структур, ассоциированных с квантовой запутанностью. Алгоритмы моделирования динамики квантовой запутанности основаны на методах компьютерной алгебры и вычислительной теории групп. Приводятся примеры конкретных вычислений.

DOI: 10.31857/S0132347421020072

1. ВВЕДЕНИЕ

Составные квантовые системы представляют особый интерес из-за наличия запутанных состояний. Понятие запутанности лежит в основе квантовых вычислений и квантовой информатики. Запутанность ответственна за такие, противоречащие обыденной интуиции, но экспериментально наблюдаемые явления, как нелокальные квантовые корреляции и квантовая телепортация. В настоящее время популярна идея о том, что геометрическое пространство физических теорий не фундаментально, а возникает за счет квантовой запутанности как приближенная феноменологическая (“эмерджентная”) структура. При этом предполагается, что метрику физического пространства можно вывести из числовых характеристик квантовых корреляций, исходя из представления о том, что чем выше квантовые корреляции между подсистемами составной системы, тем меньше геометрическое расстояние между ними.

Компьютерное моделирование представляет естественным методом исследования проблем, связанных с квантовой запутанностью. Любое компьютерное моделирование возможно только если проблема сформулирована в конечных терминах. Более того, физика, как эмпирическая наука, нечувствительна к замене бесконечностей в физических теориях подходящими конечными конструкциями. В данной работе мы используем подход, называемый для краткости “конечной

квантовой механикой”. Также мы старались избегать введения далеко идущих феноменологических упрощений и приближений, по возможности придерживаясь твердо установленных законов и правил квантовой механики.

2. КОНЕЧНАЯ КВАНТОВАЯ МЕХАНИКА

Конечной квантовой механикой мы называем версию квантовой теории, в которой унитарные группы операторов эволюции заменяются унитарными представлениями конечных групп¹, а поля комплексных и вещественных чисел заменяются их плотными конструктивными подполями, естественно возникающими из структур конкретных групп. Этот подход подробно описан в [1, 2]. Недавно Т. Бэнкс [3] проанализировал более раннее изложение [4] подхода с точки зрения реальной физики и космологии и пришел к заключению, что в его рамках можно воспроизвести результаты всех возможных экспериментов, объясняемых любой традиционной квантовой теорией.

В основе конечной квантовой механики лежит тот факт, что любое линейное представление конечной группы является подпредставлением подходящего перестановочного представления. Из этого следует, что формализм квантовой механики

¹ Любое линейное представление конечной группы над полем характеристики нуль является унитарным.

ки можно полностью² воспроизвести исходя из перестановок некоторого множества элементов

$$\Omega = \{e_0, \dots, e_{N-1}\} \cong \{0, \dots, N-1\}. \quad (2.1)$$

Здесь “изоморфизм” $\cong$ означает взаимно однозначное соответствие множеств. Для упрощения обозначений будем считать, что $\Omega = \{0, \dots, N-1\}$.

Г. ’т Хоофт в своей интерпретации квантовой механики [5] выделяет особые множества взаимно ортогональных состояний в гильбертовых пространствах, переходящие друг в друга под действием перестановок. Такие состояния ’т Хоофт наделяет статусом быть “элементами реальности” и называет “онтологическими” или, для краткости, “онтическими”. Мы будем придерживаться аналогичной терминологии, называя элементы множества Ω и связанные с ними конструкции *онтическими*. Интерпретируя Ω как множество базисных элементов, введем полумодуль H_N над полукольцом натуральных чисел $\mathbb{N} = \{0, 1, 2, \dots\}$. Элементы полумодуля, $|n\rangle = (n_0, \dots, n_{N-1})^T \in H_N$, $n_i \in \mathbb{N}$, будем называть *натуральными векторами*.

Пусть на множестве Ω действует группа перестановок $G \leq S_N$. Это действие определяет *перестановочное представление* группы G в полумодуле H_N . Матрицы перестановочного представления имеют вид

$$\mathcal{P}(g)_{i,j} = \delta_{ig,j},$$

где ig обозначает действие (справа) элемента группы $g \in G$ на элемент множества $i \in \Omega$.

Для каждой конечной группы G однозначно строится список всех ее неприводимых представлений из которых можно составить любое представление как прямую сумму. Поле, над которым любое линейное представление группы G можно разложить на неприводимые компоненты, называется *расщепляющим* полем для G . Для получения расщепляющего поля для данной группы G достаточно к натуральным числам $\mathbb{N}$ добавить ℓ -й примитивный корень из единицы r_ℓ , где ℓ — некоторый делитель наименьшего общего кратного порядков элементов группы. При этом получается кольцо, полем частных которого является абелево расширение поля рациональных чисел $\mathbb{Q}(r_\ell)$, называемое ℓ -м круговым полем. Расщепляющим полем будет некоторое подполе кругового поля: $\mathcal{F} \leq \mathbb{Q}(r_\ell)$. Расширение полукольца $\mathbb{N}$ до поля $\mathcal{F}$ индуцирует расширение полумодуля H_N до N -мерного гильбертова пространства $\mathcal{H}_N$, в котором

элементы полумодуля H_N представлены целочисленными векторами в неотрицательном ортанте. Поле $\mathcal{F}$ является плотным подполем либо поля вещественных чисел $\mathbb{R}$, либо поля комплексных чисел $\mathbb{C}$, в зависимости от структуры группы G , и поэтому эмпирически неотличимо от одного из этих континуальных полей.

Любое перестановочное представление имеет *тривиальное* подпредставление в одномерном подпространстве, порождаемом вектором $\underbrace{(1, 1, \dots, 1)}_N^T$.

$(N-1)$ -мерное подпредставление, дополненное к тривиальному, называется *стандартным* представлением. Для симметрической группы S_N (в отличие от ее собственных подгрупп) стандартное представление всегда является неприводимым.

Квантово-механическое поведение (интерференции и т.д.) проявляется именно в стандартном представлении, а квантовые состояния можно описать с помощью проекций натуральных векторов в $(N-1)$ -мерное стандартное пространство. Скалярное произведение для нормированных проекций в стандартное пространство натуральных векторов $|n\rangle$ и $|m\rangle$ имеет вид

$$S(n, m) = \frac{N \langle n | m \rangle - \langle n \rangle \langle m \rangle}{\sqrt{N \langle n | n \rangle - \langle n \rangle^2} \sqrt{N \langle m | m \rangle - \langle m \rangle^2}},$$

$$\text{где } \langle n \rangle = \sum_{k=0}^{N-1} n_k, \quad \langle n | m \rangle = \sum_{k=0}^{N-1} n_k m_k.$$

В статье [3] используется естественное предположение о том, что для описания реальных физических систем число N должно быть порядка числа фундаментальных (планковских) степеней свободы рассматриваемой системы (приводятся оценки $N \sim \text{Exp}(10^{20})$ для 1 см^3 материи и $N \sim \text{Exp}(10^{123})$ для всей Вселенной). Там же для установления соответствия между конечной квантовой механикой и традиционной теорией, основанной на непрерывных унитарных группах, используется утверждение о том, что для достаточно больших N (согласно [6] $N \geq 72$) наиболее общая конечная подгруппа группы $SU(N-1)$ имеет структуру полупрямого произведения конечной абелевой A и симметрической групп:

$$G = A \rtimes S_N \leq SU(N-1). \quad (2.2)$$

С помощью проекций натуральных векторов в стандартное пространство можно с произвольной точностью воспроизвести любое квантовое состояние. Однако, квантовые состояния, соответствующие коллинеарным векторам, эквивалентны. Чтобы устранить эту избыточность, можно предвари-

² Если пренебречь эмпирически несущественными элементами традиционного формализма, такими как разного рода бесконечности.

тельно спроектировать натуральные векторы на пересечение единичной сферы $\mathbb{S}^{N-1} \subset \mathcal{H}_N$ с неотрицательным ортантом. Это пересечение мы обозначим символом $\mathbb{S}_+^{N-1}$. Проекции натуральных векторов составляют плотное подмножество в $\mathbb{S}_+^{N-1}$. Для моделирования физических систем можно, отказавшись от требования плотности, воспользоваться конечными подмножествами натуральных векторов.

Назовем векторами порядка $m \geq 2$ натуральные векторы, координаты которых принадлежат множеству $\{0, 1, \dots, m-1\}$. Полное число таких векторов равно m^N , тогда как площадь $\mathbb{S}_+^{N-1}$ равна $\frac{N\pi^{N/2}}{2^N \Gamma(N/2 + 1)} \sim \sqrt{\frac{N}{\pi}} \left(\frac{e\pi}{2N}\right)^{N/2}$. Мы видим, что с ростом N число векторов порядка m быстро растет при любом m , тогда как площадь $\mathbb{S}_+^{N-1}$ быстро убывает, т.е. векторы порядка m хорошо представляют значительную часть квантовых состояний.

2.1. Онтические векторы

Натуральные векторы порядка 2 мы будем называть *онтическими*. Такие векторы обладают наиболее естественными и удобными для вычислений свойствами. Онтический вектор $|q\rangle$ можно записать в виде битовой строки длины N . Интерпретируя эту строку, как характеристическую функцию, можно отождествить онтический вектор с соответствующим нетривиальным подмножеством множества онтических элементов (2.1): $q \subset \Omega$. Полное множество онтических векторов имеет вид $\mathcal{Q} = 2^\Omega \setminus \{\emptyset, \Omega\}$, $|\mathcal{Q}| = 2^N - 2$. Для нормализованных проекций онтических векторов $|q\rangle$ и $|r\rangle$ в стандартное пространство имеем скалярное произведение

$$S(q, r) = \frac{N \langle q \& r \rangle - \langle q \rangle \langle r \rangle}{\sqrt{\langle q \rangle \langle \sim q \rangle \langle r \rangle \langle \sim r \rangle}}$$

и соответствующее правило Борна для квантовых вероятностей

$$\mathbf{B}(q, r) = \frac{(N \langle q \& r \rangle - \langle q \rangle \langle r \rangle)^2}{\langle q \rangle \langle \sim q \rangle \langle r \rangle \langle \sim r \rangle}.$$

Здесь $a \& b$ означает побитовое “И” для битовых строк, $\sim a$ – побитовую инверсию, $\langle a \rangle$ – число единиц в битовой строке (вес Хэмминга).

Из очевидных тождеств $\langle \sim a \rangle = N - \langle a \rangle$ и $\langle a \& b \rangle + \langle a \& \sim b \rangle = \langle a \rangle$ следуют симметрии для скалярного произведения и вероятности Борна относительно замены подмножеств онтических элементов их дополнениями:

$$S(q, r) = -S(\sim q, r) = -S(q, \sim r) = S(\sim q, \sim r), \\ \mathbf{B}(q, r) = \mathbf{B}(\sim q, r) = \mathbf{B}(q, \sim r) = \mathbf{B}(\sim q, \sim r).$$

3. СОСТАВНЫЕ КВАНТОВЫЕ СИСТЕМЫ

Пусть компоненты M -компонентной квантовой системы индексируются элементами множества

$$X \equiv \{0, \dots, M-1\}. \quad (3.1)$$

Гильбертово пространство такой составной системы имеет вид

$$\tilde{\mathcal{H}} = \bigotimes_{x \in X} \mathcal{H}_x. \quad (3.2)$$

Мы будем рассматривать “однородные” системы, для которых все пространства $\mathcal{H}_x$ изоморфны, т.е. $\mathcal{H}_x \cong \mathcal{H}$, где $\mathcal{H}$ – представитель класса изоморфизма, который мы будем называть *локальным* гильбертовым пространством. Если $\dim \mathcal{H} = n$, то $\dim \tilde{\mathcal{H}} = n^M \equiv N$. Как принято в физике, ортонормированные базисные элементы гильбертовых про-

странств записываются в виде $|0\rangle = \begin{pmatrix} 1 \\ 0 \\ \vdots \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 0 \\ 1 \\ \vdots \end{pmatrix}$ и

т.д. Таким образом, базисные элементы локального $\mathcal{H}$ и полного $\tilde{\mathcal{H}}$ пространств имеют, соответственно, вид $|i\rangle$, $0 \leq i \leq n-1$ и $|\tilde{i}\rangle$, $0 \leq \tilde{i} \leq N-1$. Связь между базисными элементами полного и локального пространств выражается взаимно однозначным соотношением

$$|\tilde{i}\rangle = |i_0\rangle \otimes |i_1\rangle \otimes \dots \otimes |i_{M-1}\rangle \equiv \bigotimes_{x \in X} |i_x\rangle, \quad (3.3)$$

где индекс $\tilde{i}$ вычисляется по формуле

$$\tilde{i} = i_0 n^{M-1} + \dots + i_{M-1} \equiv \sum_{m=0}^{M-1} i_m n^{M-m-1}.$$

Из взаимной однозначности соответствия (3.3) видно, что любое гильбертово пространство, размерность которого равна степени натурального числа, можно разложить в тензорное произведение изоморфных “локальных” гильбертовых пространств.

Состояние квантовой системы, описываемое одним вектором $|v\rangle$, называется *чистым*. В общем случае квантовое состояние можно описать матрицей плотности ρ , которая представляет собой положительно полуопределенную эрмитову матрицу с единичным следом. Матрица плотности чистого состояния имеет вид

$$\rho = \frac{|v\rangle\langle v|}{\langle v | v \rangle} \quad (3.4)$$

и является оператором проектирования ранга один. Матрица плотности общего вида описывает состояния, называемые *смешанными*, и представляет собой взвешенную сумму матриц плотностей чистых состояний.

Статистика поведения выделенной подсистемы составной системы полностью описывается *редуцированной матрицей плотности*, получаемой из матрицы плотности всей системы взятием следов по всем компонентам, не входящим в рассматриваемую подсистему. В общем случае редуцированная матрица плотности описывает смешанное состояние, даже если состояние системы в целом является чистым. С другой стороны, любое смешанное состояние можно получить редукцией некоторого чистого состояния в гильбертовом пространстве большей размерности³.

Некоторые состояния составной системы можно представить как тензорные произведения состояний подсистем. Такие состояния называются *сепарабельными*. Состояния, не являющиеся сепарабельными, называются *запутанными*.

Для иллюстрации рассмотрим пример четырехмерного гильбертова пространства, представленного как тензорное произведение двух двумерных: $\tilde{\mathcal{H}}_4 = \mathcal{H}_2 \otimes \mathcal{H}_2$. Базисные элементы пространства $\tilde{\mathcal{H}}_4$ выражаются через базисные элементы пространства $\mathcal{H}_2$ следующим образом:

$$\begin{aligned} |\tilde{0}\rangle &= |0\rangle \otimes |0\rangle, & |\tilde{1}\rangle &= |0\rangle \otimes |1\rangle, \\ |\tilde{2}\rangle &= |1\rangle \otimes |0\rangle, & |\tilde{3}\rangle &= |1\rangle \otimes |1\rangle. \end{aligned}$$

Для примера рассмотрим только онтические векторы. Из четырнадцати онтических векторов в четырехмерном пространстве восемь являются сепарабельными:

$$\begin{aligned} |\tilde{0}\rangle &= |0\rangle \otimes |0\rangle, \\ |\tilde{1}\rangle &= |0\rangle \otimes |1\rangle, \\ |\tilde{2}\rangle &= |1\rangle \otimes |0\rangle, \\ |\tilde{3}\rangle &= |1\rangle \otimes |1\rangle, \\ |\tilde{0}\rangle + |\tilde{1}\rangle &= |0\rangle \otimes (|0\rangle + |1\rangle), \\ |\tilde{0}\rangle + |\tilde{2}\rangle &= (|0\rangle + |1\rangle) \otimes |0\rangle, \\ |\tilde{1}\rangle + |\tilde{3}\rangle &= (|0\rangle + |1\rangle) \otimes |1\rangle, \\ |\tilde{2}\rangle + |\tilde{3}\rangle &= |1\rangle \otimes (|0\rangle + |1\rangle) \end{aligned}$$

и шесть запутанными:

$$\begin{aligned} |\tilde{1}\rangle + |\tilde{2}\rangle &= |0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle, \\ |\tilde{0}\rangle + |\tilde{3}\rangle &= |0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle, \\ |\tilde{0}\rangle + |\tilde{1}\rangle + |\tilde{2}\rangle &= |0\rangle \otimes |0\rangle + |0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle, \\ |\tilde{0}\rangle + |\tilde{1}\rangle + |\tilde{3}\rangle &= |0\rangle \otimes |0\rangle + |0\rangle \otimes |1\rangle + |1\rangle \otimes |1\rangle, \\ |\tilde{0}\rangle + |\tilde{2}\rangle + |\tilde{3}\rangle &= |0\rangle \otimes |0\rangle + |1\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle, \\ |\tilde{1}\rangle + |\tilde{2}\rangle + |\tilde{3}\rangle &= |0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle. \end{aligned}$$

4. СИММЕТРИИ СИСТЕМ С ГЕОМЕТРИЧЕСКИМ ПРОСТРАНСТВОМ

Любая подгруппа симметрической группы онтического множества имеет перестановочное представление в гильбертовом пространстве составной квантовой системы и может служить группой симметрий этой системы. Здесь мы рассмотрим специальный тип групп симметрий составных систем — подгруппы симметрической группы, сконструированные из заданных групп “геометрических” и “локальных” симметрий.

Для однородной составной системы изоморфные локальные гильбертовы пространства $\mathcal{H}_x$ лежат на орбите некоторой группы G , действующей на множестве X , которое можно интерпретировать как “геометрическое пространство” с группой “пространственных симметрий” G .

Пусть локальное гильбертово пространство $\mathcal{H}$ является пространством перестановочного представления группы “локальных симметрий” $F \leq S_n$, действующей перестановками на “локальном” множестве

$$V \cong \{0, \dots, n-1\}, \quad (4.1)$$

являющемся базисом локального пространства.

Исходя из групп F и G и естественных свойств, которыми должно обладать геометрическое пространство, можно конкретизировать структуру группы симметрий $\tilde{W}$, унитарное представление которой в полном гильбертовом пространстве (3.2) описывает квантовые свойства системы в целом. Прежде всего, группа $\tilde{W}$ переставляет компоненты составной системы, ассоциированные с точками пространства X (3.1), в соответствии с действием пространственной группы G , поэтому должен существовать гомоморфизм $h: \tilde{W} \rightarrow G$. Образ этого гомоморфизма — вся группа G , т.е., $\text{im}(h) = h(\tilde{W}) = G$. Естественно предположить, что, если все точки пространства неподвижны (т.е., на X действует единица $1_G \in G$ пространственной группы), то группа $\tilde{W}$ должна действовать как прямое произведение M экземпляров локальной группы

³ Метафора “Church of the Larger Hilbert Space” (J.A. Smolin) характеризует убеждение, что любое смешанное состояние фактически происходит из более фундаментального чистого состояния большей системы.

F (которое мы будем обозначать символом F^X) покомпонентно на декартово произведение M экземпляров множества (4.1) (которое мы обозначим символом V^X). Это означает, что $\ker(h) \cong F^X$, т.е., ядро гомоморфизма h изоморфно группе функций на точках пространства со значениями в локальной группе. Ядро любого гомоморфизма является нормальной подгруппой, поэтому $\tilde{W}$ содержит нормальную подгруппу, изоморфную F^X . Таким образом, группа $\tilde{W}$ представляет собой расширение группы функций F^X с помощью группы G пространственных симметрий. Это означает, что $\tilde{W}/F^X \cong G$ или, эквивалентно, что существует короткая точная последовательность

$$1 \rightarrow F^X \rightarrow \tilde{W} \rightarrow G \rightarrow 1.$$

Множество элементов группы $\tilde{W}$ можно отождествить (как множество) с декартовым произведением множеств F^X и G , т.е., элементы группы $\tilde{W}$ можно представить в виде пар $(f(x), g)$, где $f(x) \in F^X$, $g \in G$.

При построении расширения следует учесть, что расширения групп определяются с точностью до эквивалентности расширений [7]. В нашем случае эквивалентность описывается групповым изоморфизмом $\Phi: \tilde{W} \rightarrow \tilde{W}'$, обеспечивающим коммутативность диаграммы

$$\begin{array}{ccccc} & & \tilde{W} & & \\ & \nearrow & \downarrow \Phi & \searrow & \\ 1 & \rightarrow & F^X & & G \rightarrow 1. \\ & \searrow & \downarrow & \nearrow & \\ & & \tilde{W}' & & \end{array} \quad (4.2)$$

Проведя вычисления для рассматриваемого случая, получаем следующее

Предложение.

• *Классы эквивалентности расширений (4.2) параметризуются антигомоморфизмами⁴ пространственной группы: $\mu: G \rightarrow G$.*

• *Изоморфизм эквивалентных расширений имеет вид*

$$\Phi: (f(x), g) \mapsto (f(x\varphi(g)), g),$$

• *где $\varphi: G \rightarrow G$ — произвольная функция.*

⁴ Антигомоморфизм — это отображение групп, обращающее порядок группового умножения: $\mu(ab) = \mu(b)\mu(a)$.

• *Основные групповые операции имеют следующий явный вид:*

$$v(x)(f(x), g) = v(x\mu(g))f(x\varphi(g)), \quad (4.3)$$

$$(f(x), g)(f'(x), g') =$$

$$= (f(x\varphi(gg')^{-1}\mu(g')\varphi(g))$$

$$f'(x\varphi(gg')^{-1}\varphi(g')), gg'), \quad (4.4)$$

$$(f(x), g)^{-1} =$$

$$= (f(x\varphi(g^{-1})^{-1}\mu(g)^{-1}\varphi(g))^{-1}, g^{-1}), \quad (4.5)$$

где (4.3) — действие элемента группы $(f(x), g) \in \tilde{W}$ на функцию $v(x) \in V^X$, (4.4) — групповое умножение в $\tilde{W}$ и (4.5) — взятие обратного элемента.

Известны два универсальных, т.е. существующих для любых групп вне зависимости от их конкретных свойств, антигомоморфизма: $\mu(g) = 1$ и $\mu(g) = g^{-1}$.

Случай $\mu(g) = 1$ неинтересен, так как приводит к тривиальному расширению, т.е. прямому произведению групп: $\tilde{W} \cong F^X \times G$.

Антигомоморфизм $\mu(g) = g^{-1}$ приводит к полупрямому произведению групп F^X и G , называемому *сплетением групп F и G* :

$$\tilde{W} = F \wr G \cong F^X \rtimes G. \quad (4.6)$$

Что касается произвольной функции φ , в математической литературе при описании сплетений обычно используется выбор $\varphi(g) = g^{-1}$. В реализации наших алгоритмов мы используем два варианта: $\varphi(g) = g^{-1}$ и $\varphi(g) = 1$. В этих случаях выражения (4.3)–(4.5) для групповых операций получаются относительно компактными:

$$\underline{\varphi(g) = g^{-1}}:$$

$$v(x)(f(x), g) = v(xg^{-1})f(xg^{-1}),$$

$$(f(x), g)(f'(x), g') = (f(x)f'(xg), gg'),$$

$$(f(x), g)^{-1} = (f(xg^{-1})^{-1}, g^{-1}).$$

$$\underline{\varphi(g) = 1}:$$

$$v(x)(f(x), g) = v(xg^{-1})f(x),$$

$$(f(x), g)(f'(x), g') = (f(xg^{-1})f'(x), gg'),$$

$$(f(x), g)^{-1} = (f(xg)^{-1}, g^{-1}).$$

Сплетения играют важную роль в теории групп и имеют многочисленные приложения в других разделах математики [8, 9]. В частности, согласно универсальной теореме вложения Калужнина–Краснера [10], сплетение $A \wr B$ содержит в качестве подгруппы любое расширение группы A с помощью группы B , т.е. сплетение является универсальным объектом, содержащим все расширения.

Сплетение локальной F и пространственной G групп является естественной группой симметрий составных квантовых систем, гильбертово пространство которых подразумевает наличие геометрической структуры.

5. КВАНТОВЫЕ ЭНТРОПИИ

Энтропией вероятностного распределения $P = (p_1, \dots, p_n)$ называется вещественная функция $H(P)$, обладающая некоторыми естественными свойствами, в частности, аддитивностью на комбинации независимых распределений:

$$H(PQ) = H(P) + H(Q). \quad (5.1)$$

А. Реньи показал [11], что требуемым свойствам удовлетворяет семейство функций

$$H_\alpha(P) = \frac{1}{1-\alpha} \log \sum_{i=1}^n p_i^\alpha, \quad \alpha \geq 0, \quad \alpha \neq 1, \quad (5.2)$$

называемых энтропиями Реньи порядка α .

Квантовая версия семейства энтропий Реньи имеет вид

$$S_\alpha(\rho) = \frac{1}{1-\alpha} \log \text{tr}(\rho^\alpha), \quad (5.3)$$

где ρ – матрица плотности квантового состояния.

Если условие (5.1), отказавшись от предположения о независимости P и Q , заменить более сильным условием $H(PQ) = H(P) + H(Q|P)$, то получится единственная функция, называемая энтропией Шэннона

$$H_1(P) = -\sum_{i=1}^n p_i \log p_i,$$

которую также можно включить в семейство (5.2) с помощью предельного перехода $\alpha \rightarrow 1$. Квантовая версия энтропии Шэннона

$$S_1(\rho) = -\text{tr}(\rho \log \rho) \quad (5.4)$$

называется энтропией фон Неймана.

Семейство энтропий Реньи обладает свойством монотонности: $H_\alpha \geq H_\beta$, если $\alpha < \beta$.

Энтропия H_0 , равная логарифму числа элементов распределения, имеющих ненулевую вероятность: $H_0(P) = \log |\text{supp}(P)|$, называется *энтропией Хартли* или *max-энтропией* (поскольку $H_0 \geq H_\alpha$, для любых α).

Соответственно, *min-энтропией* называется энтропия $H_\infty(P) = -\log \max_i p_i$. Квантовая версия min-энтропии представляет интерес в физических задачах, в которых выражено доминирует одно из собственных значений матрицы плотности (состояния “близкие” к чистым).

Наиболее употребительной в физике является, обладающая “хорошими” статистическими свойствами, энтропия фон Неймана (5.4).

Заметим, что для чистого состояния, матрица плотности которого (3.4) имеет ровно одно, равное единице, собственное значение, любая энтропия из семейства (5.3) равна нулю.

В данной работе мы используем квантовую энтропию Реньи

$$S_2(\rho) = -\log \text{tr}(\rho^2),$$

которая называется *квадратичной энтропией* или *энтропией столкновений*. Вычисление этой энтропии, в отличие от вычислительно проблемных энтропии фон Неймана и min-энтропии, сводится к простой формуле

$$S_2(\rho) = -\log \left(\sum_{i=1}^n \rho_{ii}^2 + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^n |\rho_{ij}|^2 \right),$$

где $\rho = (\rho_{ij})$ – $n \times n$ матрица плотности.

Кроме того, 2-я энтропия Реньи привлекательна, с нашей точки зрения, по следующей причине. Теорема Глисона [12] устанавливает взаимно однозначное соответствие между вероятностными мерами на подпространствах гильбертова пространства и квантовыми состояниями в этом пространстве⁵. Согласно этой теореме, наиболее общее выражение для борновской вероятности имеет вид

$$\mathbf{P} = \text{tr}(\rho_O \rho_S), \quad (5.5)$$

где ρ_O и ρ_S – квантовые состояния “наблюдателя” и “наблюдаемой системы”, соответственно. Поскольку правило Борна это единственный фундаментальный источник вероятности в квантовой теории, представляется естественным связать состояние ρ с некоторой борновской вероятностью. С этой точки зрения 2-я энтропия Реньи это просто логарифм борновской вероятности $\mathbf{P} = \text{tr}(\rho^2)$ – “система наблюдает саму себя”. Это единственная возможность составить формулу глисоновского ти-

⁵ За исключением гильбертовых пространств размерности два.

па (5.5), имея в наличии только состояние ρ и не вводя дополнительные элементы в описание (типа “влияния среды” или наличия других систем).

6. КВАНТОВАЯ ЗАПУТАННОСТЬ В ОДНОРОДНЫХ СИСТЕМАХ

В настоящее время активно разрабатывается идея о том, что геометрическое (физическое) пространство не фундаментально, а возникает за счет запутанности как приближенная феноменологическая структура внутри гильбертова пространства составной квантовой системы [13]–[15]. При этом метрика и топология определяется мерами запутанности между компонентами составной системы.

Для однородной составной системы матрица плотности наиболее общего квантового состояния в обозначениях раздела 3 имеет вид

$$\tilde{\rho}_X = \sum_{i,j=0}^{N-1} \tilde{\rho}_{ij} |\tilde{i}\rangle\langle\tilde{j}| \equiv \sum_{i,j=0}^{N-1} \tilde{\rho}_{ij} \otimes_{x \in X} |i_x\rangle\langle j_x|,$$

где связь между “локальными” $|i_x\rangle, |j_x\rangle$ и “глобальными” $|\tilde{i}\rangle, |\tilde{j}\rangle$ базисными векторами задается соотношением (3.3). Из правил квантовой механики следует, что статистическое поведение подсистемы, относящейся к подмножеству точек $A \subseteq X$, правильно описывается редуцированной матрицей плотности, получаемой взятием следа по всем компонентам, относящимся к точкам, не входящим в A :

$$\tilde{\rho}_A = \text{tr}_{X \setminus A} \tilde{\rho}_X.$$

Для краткости редуцированные матрицы плотности одноточечных и двухточечных подсистем мы будем обозначать символами типа $\tilde{\rho}_x$ и $\tilde{\rho}_{xy}$.

Представление о геометрии, как о феномене квантовой запутанности, основано на идее определять расстояния между точками пространства X через квантовые корреляции: чем выше корреляции, тем расстояния меньше. Существует множество мер квантовой запутанности и множество способов построения метрических пространств исходя из этих мер. Эти способы используют различные модельные упрощения и аппроксимации [13]–[15]. Чтобы не углубляться в эти упрощения, мы рассмотрим только наиболее часто используемую меру квантовых корреляций.

Типичной мерой запутанности для пары точек $\{x, y\} \subseteq X$ является *взаимная информация*

$$I(x, y) = S(\rho_x) + S(\rho_y) - S(\rho_{xy}). \quad (6.1)$$

Строго говоря, взаимная информация (6.1) имеет “достаточно хороший” смысл только для энтро-

пии фон Неймана $S_1(\rho)$. Для энтропий Реньи с $\alpha \neq 1$ могут возникать различные проблемы, например, выражение (6.1) может оказаться отрицательным. Однако это выражение всегда обращается в нуль на сепарабельных и отлично от нуля на запутанных парах подсистем и поэтому, в любом случае, может служить мерой запутанности.

Мы будем изучать эволюции величин $I(x, y)$ на ребрах $\{x, y\}$ полного графа с вершинами X для проекций онтических векторов в стандартное пространство и его подпространства.

7. ПРИМЕР КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ДИНАМИКИ ЗАПУТАННОСТИ

В качестве примера небольшой размерности построим однородную составную квантовую систему, исходя из следующих элементов

- $X \equiv \{0, 1, 2\}$ – геометрическое пространство из $M = 3$ точкам;
- $G = S_3$ – группа симметрий пространства X ;
- $V \equiv \{0, 1, 2\}$ – базис локального гильбертова пространства $\mathcal{H}$ размерности $n = 3$;
- $F = Z_3$ – группа локальных симметрий.

Эти данные являются входными для компьютерной программы, написанной на языке Си.

Программа конструирует гильбертово пространство составной системы, которое, в соответствии с (3.2), имеет структуру

$$\tilde{\mathcal{H}} = \mathcal{H}_0 \otimes \mathcal{H}_1 \otimes \mathcal{H}_2.$$

Базис этого пространства мы будем отождествлять с онтическим множеством (2.1), поэтому $N = \dim \tilde{\mathcal{H}} = n^M = 27$.

Также конструируется сплетение (4.6), являющееся группой симметрий составной системы:

$$\tilde{W} = Z_3 \wr S_3 \cong (Z_3 \times Z_3 \times Z_3) \rtimes S_3. \quad (7.1)$$

Заметим, что группа (7.1) имеет, важную для описания конечных подгрупп унитарных групп, структуру (2.2).

Далее, с помощью алгоритма, описанного в [16]–[18], строится набор локальных проекторов, обеспечивающих разложение перестановочного представления локальной группы F на неприводимые подпредставления:

$$B_1 = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix},$$

$$B_2 = \begin{pmatrix} \frac{1}{3} & -\frac{1}{6} - i\frac{\sqrt{3}}{6} & -\frac{1}{6} + i\frac{\sqrt{3}}{6} \\ -\frac{1}{6} + i\frac{\sqrt{3}}{6} & \frac{1}{3} & -\frac{1}{6} - i\frac{\sqrt{3}}{6} \\ -\frac{1}{6} - i\frac{\sqrt{3}}{6} & -\frac{1}{6} + i\frac{\sqrt{3}}{6} & \frac{1}{3} \end{pmatrix},$$

$$B_3 = \begin{pmatrix} \frac{1}{3} & -\frac{1}{6} + i\frac{\sqrt{3}}{6} & -\frac{1}{6} - i\frac{\sqrt{3}}{6} \\ -\frac{1}{6} - i\frac{\sqrt{3}}{6} & \frac{1}{3} & -\frac{1}{6} + i\frac{\sqrt{3}}{6} \\ -\frac{1}{6} + i\frac{\sqrt{3}}{6} & -\frac{1}{6} - i\frac{\sqrt{3}}{6} & \frac{1}{3} \end{pmatrix},$$

$$\text{rank} B_1 = \text{rank} B_2 = \text{rank} B_3 = 1.$$

Следующим шагом, программа, используя алгоритм, описанный в [19, 20], вычисляет неприводимые проекторы сплетения (7.1) в виде тензорных полиномов от локальных проекторов:

$$\tilde{B}_1 = B_1^{\otimes 3},$$

$$\tilde{B}_2 = B_2^{\otimes 3},$$

$$\tilde{B}_3 = B_3^{\otimes 3},$$

$$\tilde{B}_4 = B_1 \otimes B_2^{\otimes 2} + B_2 \otimes B_1 \otimes B_2 + B_2^{\otimes 2} \otimes B_1,$$

$$\tilde{B}_5 = B_1^{\otimes 2} \otimes B_3 + B_1 \otimes B_3 \otimes B_1 + B_3 \otimes B_1^{\otimes 2},$$

$$\tilde{B}_6 = B_1 \otimes B_3^{\otimes 2} + B_3 \otimes B_1 \otimes B_3 + B_3^{\otimes 2} \otimes B_1,$$

$$\tilde{B}_7 = B_1^{\otimes 2} \otimes B_2 + B_1 \otimes B_2 \otimes B_1 + B_2 \otimes B_1^{\otimes 2},$$

$$\tilde{B}_8 = B_2^{\otimes 2} \otimes B_3 + B_2 \otimes B_3 \otimes B_2 + B_3 \otimes B_2^{\otimes 2},$$

$$\tilde{B}_9 = B_2 \otimes B_3^{\otimes 2} + B_3 \otimes B_2 \otimes B_3 + B_3^{\otimes 2} \otimes B_2,$$

$$\begin{aligned} \tilde{B}_{10} = & B_1 \otimes B_2 \otimes B_3 + B_2 \otimes B_1 \otimes B_3 + \\ & + B_1 \otimes B_3 \otimes B_2 + B_3 \otimes B_1 \otimes B_2 + \\ & + B_2 \otimes B_3 \otimes B_1 + B_3 \otimes B_2 \otimes B_1, \end{aligned}$$

$$\text{rank} \tilde{B}_1 = \text{rank} \tilde{B}_2 = \text{rank} \tilde{B}_3 = 1,$$

$$\text{rank} \tilde{B}_4 = \text{rank} \tilde{B}_5 = \dots = \text{rank} \tilde{B}_9 = 3,$$

$$\text{rank} \tilde{B}_{10} = 6.$$

Эти проекторы определяют неприводимые инвариантные подпространства $\tilde{\mathcal{H}}_i = \tilde{B}_i \mathcal{H}$. Размерности этих подпространств равны рангам соответ-

ствующих проекторов: $\dim \tilde{\mathcal{H}}_i = \text{rank} \tilde{B}_i$. Таким образом, 27-мерное перестановочное представление сплетения (7.1) расщепляется на 3 одномерных, 6 трехмерных и одно 6-мерное неприводимые подпредставления.

Как отмечалось ранее, для построения квантовых моделей пригодны только подпространства, принадлежащие стандартному $(N-1)$ -мерному подпространству, т.е. в нашем случае, пространства $\tilde{\mathcal{H}}_2, \tilde{\mathcal{H}}_3, \dots, \tilde{\mathcal{H}}_{10}$ либо их прямые суммы.

С помощью формулы Глисона для борновской вероятности (5.5) можно показать, что для разбиения гильбертова пространства на взаимно ортогональные подпространства, вероятность обнаружить квантовое состояние в одном из подпространств пропорциональна его размерности. Поэтому наиболее предпочтительным является полное 26-мерное стандартное подпространство. Оператор проектирования в это подпространство в терминах тензорных полиномов от локальных проекторов имеет вид

$$\tilde{B}_{std} = \tilde{B}_2 + \tilde{B}_3 + \dots + \tilde{B}_{10} \equiv 1_3^{\otimes 3} - B_1^{\otimes 3}. \quad (7.2)$$

Представление в виде тензорных полиномов позволяет свести вычисления с матрицами больших размерностей к последовательности операций с небольшими матрицами.

Для управления вычислениями создаются три множества:

1. Множество проекторов в подпространства, вычисления в которых представляют интерес. По умолчанию это множество состоит из одного элемента $\tilde{B}_{std}$.

2. Множество онтических векторов, эволюция которых будет вычисляться. Это множество задается либо вручную, либо строится с помощью генератора случайных чисел.

3. Множество элементов группы $\tilde{W}$, порождающих эволюции. Это множество также либо задается вручную, либо генерируется случайно. Элементы, порождающие эволюцию, являются дискретными аналогами гамильтонианов⁶.

Выбрав по одному элементу из каждого множества мы строим полную эволюцию. Вследствие конечности группы $\tilde{W}$ эволюция циклична. Длина цикла не превышает порядок группового элемента $\tilde{w} \in \tilde{W}$, порождающего эволюцию. При этом для каждого ребра $\{x, y\} \subseteq X$ полного графа

⁶ Любую унитарную матрицу можно представить в виде экспоненты эрмитовой, а соответствующую формулу $U = e^{iH}$ интерпретировать как унитарную эволюцию за один шаг по времени, порождаемую гамильтонианом H .

вычисляется выражение (6.1) (аналог взаимной информации для энтропии Реньи порядка 2).

Приведем примеры результатов вычислений для эволюций проекции запутанного онтического вектора

$$|000\rangle + |210\rangle + |120\rangle + |201\rangle + |011\rangle + |021\rangle + \\ + |221\rangle + |002\rangle + |102\rangle + |012\rangle + |112\rangle + |212\rangle$$

в стандартное подпространство. Здесь $|ijk\rangle$ – сокращение для $|i\rangle \otimes |j\rangle \otimes |k\rangle$.

Эволюция длины 3

t	g^t	$I(0, 1)$	$I(0, 2)$	$I(1, 2)$
0	()	1.34	0.80	0.66
1	()	1.34	0.80	0.66
2	()	1.34	0.80	0.66

Эволюция длины 6

t	g^t	$I(0, 1)$	$I(0, 2)$	$I(1, 2)$
0	()	1.34	0.80	0.66
1	(0, 2)	0.66	0.80	1.34
2	()	1.34	0.80	0.66
3	(0, 2)	0.66	0.80	1.34
4	()	1.34	0.80	0.66
5	(0, 2)	0.66	0.80	1.34

Эволюция длины 9

t	g^t	$I(0, 1)$	$I(0, 2)$	$I(1, 2)$
0	()	1.34	0.80	0.66
1	(0, 2, 1)	0.66	1.34	0.80
2	(0, 1, 2)	0.80	0.66	1.34
3	()	1.34	0.80	0.66
4	(0, 2, 1)	0.66	1.34	0.80
5	(0, 1, 2)	0.80	0.66	1.34
6	()	1.34	0.80	0.66
7	(0, 2, 1)	0.66	1.34	0.80
8	(0, 1, 2)	0.80	0.66	1.34

В этих таблицах эволюции в момент времени t вычисляются как степень $\tilde{w}^t$ некоторого элемента сплетения $\tilde{w} = (f(x), g) \in Z_3^X \rtimes S_3 \cong Z_3 \wr S_3$. Элемент пространственной группы g^t описывает “движение” точек пространства $X = \{0, 1, 2\}$ в процессе эволюции: () означает отсутствие движения, (0, 2) – перестановку двух точек, (0, 1, 2) – поворот “по часовой стрелке” и (0, 2, 1) – поворот “против часовой стрелки”.

Как видно из таблиц, эволюции взаимной информации полностью определяются элементами пространственной группы. Аналогичным образом ведут себя эволюции любых энтропий Реньи для любых подсистем составной системы. Это обусловлено тем, что рассматриваемая нами группа операторов эволюции имеет структуру сплетения.

8. ЗАКЛЮЧЕНИЕ

Методы компьютерной алгебры и вычислительной теории групп использовались в [21] для исследования классических дискретных динамических систем и мезоскопических решеточных моделей, обладающих конечными группами симметрий. Геометрические структуры моделей, изучаемых в [21], являются заранее заданными и не зависящими от состояний системы и ее динамики дискретными пространствами, т.е., графами фиксированной структуры, обладающими подходящими симметриями. Квантовая теория предоставляет возможность “вывести геометрию” из более фундаментальных правил, связанных с квантовой запутанностью. Предлагаемый нами подход позволяет для заданных “начальных квантовых состояний” и “гамильтонианов” получить эволюцию во времени набора весов на ребрах полного графа, характеризующих запутанность между вершинами этого графа. Следующим естественным шагом является разработка алгоритмов выделения геометрических структур в этих наборах данных. В основе таких алгоритмов должно быть выделение подграфов полного графа с доминирующими весами ребер с последующим использованием методов типа *многомерного шкалирования* [22] для приближенного воспроизведения (в тех случаях, когда это возможно и имеет смысл⁷) непрерывных метрических пространств (по возможности невысокой размерности).

СПИСОК ЛИТЕРАТУРЫ

1. *Kornyak V.V.* Quantum models based on finite groups // IOP Conf. Series: Journal of Physics: Conf. Series 2018. V. 965. 012023. arXiv:1803.00408 [physics.gen-ph]
2. *Kornyak V.V.* Modeling Quantum Behavior in the Framework of Permutation Groups // EPJ Web of Conferences. 2018. V. 173. 01007. arXiv:1709.01831 [quant-ph]
3. *Banks T.* Finite Deformations of Quantum Mechanics. 2020. 20. p. arXiv:2001.07662 [hep-th].

⁷ Геометрические структуры, возникающие внутри гильбертова пространства, существенно зависят от квантового состояния. Поэтому, например, в [15] рассматриваются лишь квантовые состояния, на которые наложены ограничения, обеспечивающие выполнение *голографического принципа* [23], согласно которому информация (энтропия) о физической системе внутри трехмерной области сосредоточена на двумерной границе этой области.

4. *Kornyak V.V.* Mathematical Modeling of Finite Quantum Systems. In MMCP 2011, Adam G. et al., Eds., LNCS, V. 7125. Springer, 2012. arXiv:1107.5675 [quant-ph]
5. *'t Hooft G.* The Cellular Automaton Interpretation of Quantum Mechanics. *Fundamental Theories of Physics 185*. Springer, 2016. 296 p.
6. *Collins T.* On Jordan's theorem for complex linear groups // J. Group Theory. 2007. V. 10. № 4. P. 411–423.
7. *Кириллов А.А.* Элементы теории представлений. М.: Наука, 1978. 344 с.
8. *Meldrum J.D.P.* Wreath products of groups and semi-groups. New York: John Wiley & Sons Inc., 1995.
9. *James G.D., Kerber A.* The representation theory of the symmetric group. Encyclopedia of Math. and its Appl., vol. 16. Reading: Addison-Wesley, 1981.
10. *Kaloujnine L., Krasner M.* Produit complet des groupes de permutations et problème d'extension de groupes I. // Acta Sci. Math. Szeged. 1950. V. 13. P. 208–230. II. // Acta Sci. Math. Szeged. 1951. V. 14. P. 39–66. III. // Acta Sci. Math. Szeged. 1951. V. 14. P. 69–82.
11. *Rényi A.* On measures of information and entropy / Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability. 1960. P. 547–561.
12. *Gleason A.M.* Measures on the closed subspaces of a Hilbert space // Indiana Univ. Math. J. 1957. V. 6. P. 885–893.
13. *Van Raamsdonk M.* Building up spacetime from quantum entanglement // Gen. Relativ. Grav. 2010. V. 42. P. 2323–2329.
14. *Maldacena J., Susskind L.* Cool horizons for entangled black holes // Fortsch. Phys. 2013. V. 61. № 9. P. 781–811.
15. *Cao C., Carroll S.M., Michalakis S.* Space from Hilbert space: Recovering geometry from bulk entanglement // Phys. Rev. D. 2017. V. 95. 024031.
16. *Kornyak V.V.* Splitting permutation representations of finite groups by polynomial algebra methods, Gerdt V.P. et al., Eds. CASC 2018, Cham: Springer, 2018. LNCS. V. 11077. P. 304–318.
17. *Корняк В.В.* Алгоритм разложения представлений конечных групп с помощью инвариантных проекторов // Записки научных семинаров ПОМИ. 2018. V. 468. P. 228–248.
18. *Kornyak V.V.* A new algorithm for irreducible decomposition of representations of finite groups // J. Phys.: Conf. Ser. 2019. V. 1194. 012060.
19. *Kornyak V.V.* An Algorithm for Computing Invariant Projectors in Representations of Wreath Products. In CASC 2019. England M. et al., Eds. Cham: Springer, 2019. LNCS, vol. 11661. P. 300–314.
20. *Корняк В.В.* Вычисление неприводимых разложений перестановочных представлений сплетений конечных групп // ЖВМиМФ. 2020. v. 60. № 1. С. 96–108.
21. *Корняк В.В.* Дискретные динамические системы с симметриями: компьютерный анализ // Программирование. 2008. V. 34. № 2. С. 33–47.
22. *Borg I., Groenen P.J.F.* Modern Multidimensional Scaling: theory and applications. 2nd ed. New York: Springer-Verlag, 2005.
23. *'t Hooft G.* Dimensional Reduction in Quantum Gravity. 1993. arXiv:gr-qc/9310026

КОМПЬЮТЕРНАЯ АЛГЕБРА

УДК 004.422.8

КОМПЬЮТЕРНАЯ АЛГЕБРА НА JULIA

© 2021 г. Д. С. Кулябов^{a,b,*}, А. В. Королькова^{a,**}

^a Кафедра прикладной информатики и теории вероятностей, Российский университет дружбы народов,
117198 Россия, Москва, ул. Миклухо-Маклая, д. 6

^b Лаборатория информационных технологий, Объединенный институт ядерных исследований,
141980 Россия, Московская область, Дубна, ул. Жолио-Кюри 6

*E-mail: kulyabov-ds@rudn.ru

**E-mail: korolkova-av@rudn.ru

Поступила в редакцию 03.08.2020 г.

После доработки 05.09.2020 г.

Принята к публикации 05.09.2020 г.

В последнее время на место основного языка научных и инженерных расчетов выдвигается язык Julia. У ряда пользователей возникает желание работать полностью внутри “экосистемы” Julia, подобно тому, как происходит работа в “экосистеме” Python. Для Julia существуют библиотеки, покрывающие большинство потребностей научно-инженерных расчетов. Перед авторами возникла необходимость использовать символьные вычисления для задач математического моделирования. Поскольку основным языком реализации численных алгоритмов мы выбрали язык Julia, то и задачи компьютерной алгебры хотелось бы решать на этом же языке. Авторы выделили основные функциональные области, задающие разные варианты применения систем компьютерной алгебры. В каждой из областей нами выделены наиболее характерные представители систем компьютерной алгебры на Julia. В результате авторы делают вывод, что в рамках “экосистемы” Julia возможно (и даже удобно) использовать системы компьютерной алгебры.

DOI: 10.31857/S0132347421020084

1. ВВЕДЕНИЕ

Для задач математического моделирования мы ориентированы на использование языка Julia [1–4].

Язык Julia является проблемно-ориентированным языком для научных и инженерных расчетов. Язык имеет простой синтаксис. Для облегчения перехода научно-инженерных работников, Julia использует удачные языковые конструкции, заимствованные из языков MATLAB, R, Python, FORTRAN [5, 6]. Для поддержки интерактивной разработки и оптимизации времени исполнения Julia использует JIT-компиляцию. Внутри Julia является LISP-подобным языком, поэтому она органично использует функциональные конструкции. Julia поддерживает библиотеки, написанные на других языках программирования.

Одной из причин выбора Julia является последовательная реализация принципа одного языка. Дело в том, что в своем развитии языки программирования для научных и инженерных задач разошлись по двум направлениям: языки для создания высокопроизводительных программ (FORTRAN, C, C++) и языки для быстрого прототипирования и использования непрограммистами (Matlab, R, Python). Julia старается решить проблему двух язы-

ков¹. Однако задачи компьютерной алгебры выпадают из основного направления развития языка Julia.

С учетом различного характера возникающих при этом задач, выделим следующие области применения систем компьютерной алгебры:

- Пользовательские системы компьютерной алгебры. Данные системы предлагают пользователю широкий спектр возможностей, не требуя от него глубоких знаний в программировании. Примеры: SymPy, Maxima, Axiom, Reduce.
- Языки компьютерной алгебры. С помощью них можно писать нетривиальные программы символьных вычислений. Примеры: Wolfram, R-Lisp.
- Предметно-ориентированные системы компьютерной алгебры.

¹ Проблему двух языков можно представить следующим образом. Прототип программного комплекса пишется на языке программирования, дружественном к разработчику. Приоритет — скорость написания программного кода. Для создания финального программного комплекса код переписывается на языке, позволяющем эффективно использовать ресурсы вычислительной техники. Приоритет — скорость выполнения. Язык Julia поддерживает обе эти парадигмы. То есть для него нет необходимости использовать два разных языка программирования.

В работе рассмотрены системы компьютерной алгебры, реализованные на языке Julia, с точки зрения этих областей применимости.

1.1. Структура статьи

В разделе 2 рассматривается пользовательская система компьютерной алгебры, использующая библиотеку SymPy. В разделе 3 рассматривается язык компьютерной алгебры, реализованный на основе идеологии языка Wolfram. В разделе 4 рассмотрен проблемно-ориентированный язык символьных вычислений для задач непрерывного моделирования динамических систем.

2. SymPy.lj – ИНТЕРФЕЙС К СИСТЕМЕ КОМПЬЮТЕРНОЙ АЛГЕБРЫ НА ЯЗЫКЕ PYTHON

Одной из ключевых основ идеологии языка Julia (как и у языка Python) является возможность использования сторонних библиотек. Например, Julia может подключать библиотеки, написанные на языках C, FORTRAN. Сходной идеологией обладает и язык Python (что и послужило основой его популярности). Поэтому было бы непредусмотрительно не использовать это богатство. Julia использует для вызова библиотек языка Python пакет PyCall.

Пакет SymPy (<http://sympy.org/>) является библиотекой Python для символьной математики. Фактически, это один из наиболее мощных свободных пакетов символьных вычислений. Естественно, что с ним можно работать из Julia.

Рассмотрим для примера, как мы можем использовать функционал SymPy из среды Julia. Установим пакет PyCall, загрузим его, а затем импортируем библиотеку SymPy.

```
import Pkg
Pkg.add("PyCall")
using PyCall
sympy = pyimport("sympy")
```

Дальнейшие операции похожи на работу в SymPy. Определим символьную переменную x и возьмем синус от этой переменной.

```
x = sympy.Symbol("x")
y = sympy.sin(x)
```

Вычислим теперь $\sin(\pi)$:

```
y.subs(x, pi)
```

Результат будет иметь тип PyObject. Чтобы его использовать в дальнейшем, следует привести его к какому-либо числовому типу. Например, следующим образом:

```
y.subs(x, pi) |> float
```

Это пример, кроме всего прочего, демонстрирует функциональную природу Julia. В данном выражении оператор `|>` задает действие функции

справа (по аналогии с конвейером в оболочке Unix). То есть записи $f(x)$ и $x \mid> f$ эквивалентны.

Однако, вышеприведенная запись выглядит несколько тяжеловесной. Для простейших операций требуется писать много дополнительного кода, не несущего никакого содержания. К счастью, можно использовать стандартные возможности Julia для расширений языка. Для того, чтобы операции SymPy не выделялись на фоне остальных вычислений Julia, вызов питоновской библиотеки производится с помощью пакета SymPy.lj [7]:

```
import Pkg
Pkg.add("SymPy")
using SymPy
```

Тогда выше рассмотренный пример примет следующий вид:

```
x = Sym("x")
y = sin(x)
subs(y, x, pi) |> float
```

Для объектов SymPy используется тип Sym.

В принципе, на этом можно и завершить. Если читатель имел опыт работы с SymPy в Python, он сможет работать и с SymPy.lj в Julia.

Язык Julia создавался таким образом, чтобы на нем легко могли писать программы люди, перешедшие с других языков программирования с научной и инженерной спецификой (таких, как Matlab, R, FORTRAN). По этой причине Julia содержит много синтаксического сахара, в частности одну и ту же операцию можно выполнить несколькими путями. Так, символьную переменную можно задать с помощью конструктора:

```
x = Sym("x")
```

Также можно задать с помощью макроса:

```
syms x y z
```

Также можно задать с помощью функции, имитирующей работу в SymPy из Python:

```
x, y = symbols("x y", commutative=false)
```

Здесь, кроме всего, кроме объявления символьных переменных указываются и их свойства.

Следует заметить, что обязательное явное задание типа несколько противоречит динамической природе Julia. Однако, в данном случае мы используем внешнюю библиотеку, поэтому играем по чужим правилам. Здесь можно провести аналогию с Python, который, при работе с внешними библиотеками, использует соглашения этой библиотеки, а не подменяет ее синтаксис своим.

SymPy.lj дает пользователю Julia возможность манипулировать символьными выражениями, предлагая для этого удобный интерфейс. Правда (учитывая производительность современных компьютеров) никакой оптимизации по скорости выполнения не производится.

Приведем несколько примеров работы с SymPy.lj.

При операциях с алгебраическими выражениями весьма полезны операции факторизации (factor) и расширения (expand).

```
x, y = symbols("x, y")
factor(x^2 - 2x + 1)
```

$$(x-1)^2$$

```
expand((x-1)*(x+1))
```

$$x^2 - 1$$

В символьных выражениях можно применять конструкции Julia, например включения²:

```
expand(prod([x-i for i in 1:5]))
```

$$x^5 - 15x^4 + 85x^3 - 225x^2 + 274x - 120$$

Это происходит от того, что SymPy.lj представляет символьные матрицы как массивы с элементами типа Sym, например, как Array{Sym}³.

Зададим символьную матрицу.

```
x, y = symbols("x y")
m = [x 1; 1 x]
```

$$\begin{bmatrix} x & 1 \\ 1 & x \end{bmatrix}$$

Тогда с матрицами можно производить стандартные операции, например умножать:

```
m * m
```

$$\begin{bmatrix} x^2 + 1 & 2x \\ 2x & x^2 + 1 \end{bmatrix}$$

```
m .* m
```

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

Также большинство стандартных операций с матрицами расширены для работы с символьными значениями. Это, кстати, один из примеров реализации множественной диспетчеризации.

Упомянутые выше функции factor и expand производят некоторое (необходимое нам) упрощение исходного выражения. В SymPy есть много функций для выполнения различного рода упрощений. Например, существует общая функция называемая simplify, которая пытается получить простейшую форму выражения, используя для этого разнообразные эвристики: simplify:

```
a = (x + x^2) / (x*sin(y)^2 + x*cos(y)^2)
simplify(a)
```

² Включения (comprehensions) задают общий и эффективный способ построения массивов. Их синтаксис похож на обозначения конструктора множеств в аксиоматике Цермело–Френкеля.

³ Естественно, это влечет за собой дополнительные накладные расходы.

$$x + 1$$

Для решения уравнений и систем уравнений можно использовать функцию solve. Например, решим уравнение

$$x^2 + 3x + 2 = 0$$

относительно x:

```
x = symbols("x")
solve(x^2 + 3*x + 2, x)
```

$$\begin{bmatrix} -2 \\ -1 \end{bmatrix}$$

Для систем уравнений можно использовать векторную нотацию:

```
x, y = symbols("x y")
eq1 = x + y - 1
eq2 = x - y - 2
solve([eq1, eq2], [x, y])
```

DictAny, Any with 2 entries:

```
y => -1/2
x => 3/2
```

И наконец, дифференцирование и интегрирование.

Функция diff используется для вычисления производной символьных выражений. Можно вычислить частные производные и производные высшего порядка. Зададим, например, функции:

```
x, y = symbols("x y")
f(x) = exp(-x) * sin(x)
g(x, y) = x^2 + 17*x*y^2
```

Вычислим производную $\frac{df(x)}{dx}$:

```
diff(f(x))
```

$$-e^{-x} \sin(x) + e^{-x} \cos(x)$$

Мы здесь опустили аргумент, по которому ведется дифференцирование. Впрочем, его можно и обозначать:

```
diff(f(x), x)
```

Также можем получить производную более высокого порядка $\frac{d^3 f(x)}{dx^3}$

```
diff(f(x), x, 3)
```

$$2(\sin(x) + \cos(x))e^{-x}$$

Можно взять и частную производную $\frac{\partial g(x, y)}{\partial x}$

```
diff(g(x, y), x)
```

$$2x + 17y^2$$

Аналогично можно проводить символьное интегрирование. Вычислим $\int (x^2 + x + 2)dx$:

```
x, y = symbols("x y")
integrate(x^2 + x + 2)
```

$$\frac{x^3}{3} + \frac{x^2}{2} + 2x$$

Или двойной интеграл $\int dy \int dx xy$:

```
integrate(x*y, (x, y))
```

$$\frac{y^3}{2}$$

Можно вычислить и определенный интеграл

```
integrate(x^2, (x, 0, 1))
```

$$\frac{1}{3}$$

Таким образом мы имеем очень богатую систему компьютерной алгебры. И работать с ней можно прямо из среды Julia.

3. SYMATA – ЯЗЫК КОМПЬЮТЕРНОЙ АЛГЕБРЫ ДЛЯ JULIA

При том, что SymPy является великолепной системой компьютерной алгебры, у нее есть небольшой недостаток. Это готовая система, “работающая из коробки”. Она предоставляет “пользовательский слой” для символьных вычислений, но не “слой разработчика”, позволяющий писать программный код. Для разработки можно использовать язык программирования для задач символьной математики Symata [8]⁴. Язык Symata реализован на языке Julia. В качестве основы для дизайна языка Symata принят язык Wolfram [9].

Для использования данной системы необходимо установить пакет Symata:

```
import Pkg
Pkg.add("Symata")
```

Загружается Symata стандартным образом: `using Symata`;

Следует обратить внимание на следующее свойство Symata. Если SymPy.lj проектировался таким образом, чтобы его можно было использовать напрямую из среды программирования Julia, бесшовно, то Symata является предметно-ориентированным языком, написанным на Julia. Он достаточно сильно отличается от языка Julia. Поэтому необходимо явно переключаться между

этими языками. Переключение в режим языка Julia осуществляется с помощью следующей функции

```
Julia()
```

Обратно в режим Symata можно переключиться с помощью функции

```
isymata()
```

Вычисления в Symata похожи на работу в системе Mathematica. Приведем пример. Пусть нам надо вычислить значение косинуса:

```
expr = cos(pi * x)
```

$$\text{Cos}(\pi x)$$

Зададим значение переменной x :

```
x = 1/3
```

$$\frac{1}{3}$$

Теперь собственно можно получить значение выражения `expr`:

```
expr
```

$$\frac{1}{2}$$

Изменим значение переменной x :

```
x = 1/6
```

$$\frac{1}{6}$$

Тогда изменится и значение вычисляемого выражения:

```
expr
```

$$\frac{3^{\frac{1}{2}}}{2}$$

Сбросим значение переменной x . Тогда искомое выражение перестанет вычисляться:

```
Clear(x)
```

```
expr
```

$$\text{Cos}(\pi x)$$

Фактически средствами Julia реализован язык Wolfram. Так что, наверное, нет необходимости более подробно освещать здесь синтаксис системы Symata. Впрочем, стоит остановиться на некоторых отличиях языка Symata от языка Wolfram. Большинство этих отличий проистекает из того желания, чтобы Symata наследовала синтаксис Julia. Например, комментарий задается символом `#`, в то время как в языке Wolfram комментарий выглядит следующим образом:

```
(* comment *)
```

Для задания списка используются не фигурные скобки `{ }`, а квадратные:

```
[a, b, c]
```

$$[a, b, c]$$

⁴ В случае Symata язык “слоя разработчика” и язык “пользовательского слоя” совпадают. Поэтому Symata является одновременно и языком программирования, и системой компьютерной алгебры.

Элементы списка могут разделяться как запятыми, так и переходом на новую строку:

```
[
  a
  c + d
  Expand((x+y)^2)
]
```

$$[a, c + d, x^2 + 2xy + y^2]$$

Аргумент функции задается не в квадратных скобках, как в Wolfram, а в круглых, как в Julia:

$f(x)$

$$f(x)$$

Подобно функциям в Julia, некоторые функции приобрели инфиксную нотацию.

Функция `Map(f, list)` приобретает инфиксную форму

`f % list`

Функция `Apply(x, y):`

`x .% y`

Более подробно можно посмотреть в документации к Symata.

Можно сделать вывод, что на основе Julia (с использованием ее мощных возможностей по созданию предметно-ориентированных языков) реализован исключительно удобный язык для компьютерной алгебры. При этом интересно отметить, что для многих аналитических вычислений в языке Symata используется библиотека SymPy.

4. КОМПЬЮТЕРНАЯ АЛГЕБРА В JULIA ДЛЯ ЗАДАЧ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

В предыдущих разделах мы рассмотрели реализацию систем компьютерной алгебры общего назначения на языке Julia. Но также большой интерес вызывают специализированные, предметно-ориентированные применения компьютерной алгебры.

Пакет `ModelingToolkit.jl` [10] предлагает специализированный язык компьютерной алгебры для задач математического моделирования. Для этого он использует возможности метапрограммирования [11] языка Julia. Отметим, что основным режимом работы для `ModelingToolkit.jl` является не интерактивный режим, а пакетный.

Для задания символьных переменных служит макрос `@variables`:

```
@variables x y
```

После этого данные символьные переменные можно использовать в символьных выражениях (сохраняя синтаксис Julia):

$$z = x^2 + y$$

Для моделирования непрерывных динамических систем необходимо использовать производные. В пакете `ModelingToolkit.jl` дифференциальные операторы строятся с помощью макроса `@derivatives`:

```
@variables t
```

```
@derivatives D' ~t
```

Здесь задан дифференциальный оператор $D = \frac{d}{dt}$. Количество штрихов ' указывает на порядок дифференциального оператора. Можно записать выражение:

$$z = t + t^2$$

$D(z)$

Оператор дифференцирования не вычисляет ничего непосредственно в этот момент, поскольку является "ленивым" оператором. Впрочем, мы можем получить результат сразу, применив функцию `expand_derivatives`:

```
expand_derivatives(D(z))
```

$$1 + 2t$$

Поскольку в Julia функции являются объектами первого порядка, то объявленные символьные переменные являются, фактически, функциями. Можно при объявлении переменных явно указать зависимости:

```
@variables t x(t) y(t)
```

Эта зависимость учитывается при дифференцировании:

$$z = x + y * t$$

```
expand_derivatives(D(z))
```

Последнее выражение будет раскрыто как

$$\text{derivative}(x(t), t) + y(t) +$$

$$\hookrightarrow \text{derivative}(y(t), t) * t$$

В качестве примера посмотрим, как решается классическая задача типа "хищник—жертва" [12, 13]:

$$\begin{cases} \frac{dx}{dt} = ax - bxy, \\ \frac{dy}{dt} = cxy - dy, \end{cases} \quad (1)$$

где x — количество жертв, y — количество хищников, t — время, a, b, c, d — коэффициенты взаимодействия между видами.

Сначала загрузим необходимые пакеты (при необходимости):

```
import Pkg
```

```
Pkg.add("ModelingToolkit")
```

```
Pkg.add("OrdinaryDiffEq")
```

Также загрузим пакет для поддержки построения графиков:

```
Pkg.add("Plots")
```

Теперь подключим необходимые пакеты:

```
using ModelingToolkit
```

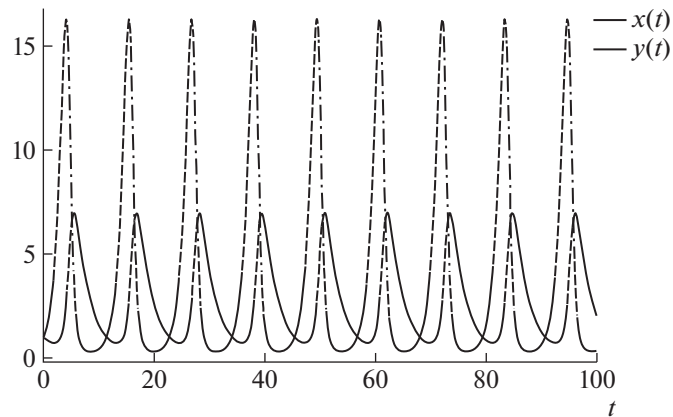



Рис. 1. Решение системы (1)

```
using OrdinaryDiffEq
using Plots
```

Собственно, теперь можно решать нашу задачу. Определим переменные и операторы дифференцирования. Часть переменных определим как параметры.

```
@parameters t a b c d
@variables x(t) y(t)
@derivatives D' ~t
```

Теперь просто перепишем исследуемую систему (1), используя синтаксис Julia:

```
eqs = [D(x) ~ a * x - b * x * y,
        D(y) ~ c * x * y - d * y]
```

Теперь применим символьные преобразования и приведем нашу систему к виду, необходимому для пакета OrdinaryDiffEq.jl [14]:

```
sys = ODESystem(eqs)
```

Дальнейшие манипуляции проводятся в рамках пакета OrdinaryDiffEq.jl. Зададим начальные значения переменных:

```
u0 = [x => 1.0
       y => 1.0]
```

Также зададим параметры задачи:

```
p = [a => 1.1
      b => 0.4
      c => 0.1
      d => 0.4]
```

Будем решать задачу на следующем временном промежутке:

```
tspan = (0.0, 100.0)
```

Создадим структуру, содержащую всю нашу задачу:

```
prob = ODEProblem(sys, u0, tspan, p;
  ↪ jac=true)
```

Дополнительный параметр `jac=true` указывает системе символически сгенерировать опти-

мизированную функцию Якоби для улучшения работы решателей дифференциальных уравнений.

И, наконец, применим решатель из пакета OrdinaryDiffEq.jl:

```
sol = solve(prob)
```

Нарисуем график, представленный на рис. 1:

```
plot(sol)
```

Рассмотренный подход очень хорошо вписывается в идеологию специализированного языка для научных и инженерных расчетов. При данном подходе для символьных вычислений используется не специальный универсальный язык, а для каждого направления исследований создается свой проблемно-ориентированный язык символьных расчетов. Этот язык должен бесшовно стыковаться с базовым языком, в нашем случае, с Julia.

5. ЗАКЛЮЧЕНИЕ

Наш обзор (далеко не полный) систем компьютерной алгебры для языка Julia показал, что и сам язык, и его инфраструктура являются достаточно зрелыми. При этом системы компьютерной алгебры языка Julia охватывают широкий спектр применения: пользовательские системы компьютерной алгебры (SymPy.jl), мощные языки компьютерной алгебры (Symata.jl), проблемно-ориентированные языки компьютерной алгебры (ModelingToolkit.jl). Все это позволяет надеяться, что популярность языка Julia будет возрастать не только в области численных, но и символьных расчетов.

6. БЛАГОДАРНОСТИ

Публикация подготовлена при поддержке Программы РУДН “5-100” и при финансовой поддержке РФФИ в рамках научного проекта № 19-01-00645.

СПИСОК ЛИТЕРАТУРЫ

1. *Bezanson J., Edelman A., Karpinski S., Shah V.B.* Julia: A fresh approach to numerical computing // *SIAM Review*. 2017. 1. V. 59. № 1. P. 65–98. arXiv: 1411.1607.
2. *Joshi A., Lakhanpal R.* Learning Julia. Packt Publishing, 2017. 316 p. ISBN: 9781785883279.
3. *Tate B.A., Daoud F., Dees I., Moffitt J.* Seven More Languages in Seven Weeks. 2015. 318 p. ISBN: 9781941222157.
4. *Bezanson J., Karpinski S., Shah V.B., Edelman A.* Julia: A Fast Dynamic Language for Technical Computing. 2012. P. 1–27. arXiv: 1209.5145.
5. *Bezanson J., Chen J., Chung B. et al.* Julia: dynamism and performance reconciled by design // *Proceedings of the ACM on Programming Languages*. 2018. 10. V. 2. № OOPSLA. P. 1–23. URL: <https://doi.org>. <https://doi.org/10.1145/3276490><https://dl.acm.org/doi/10.1145/3276490>.
6. A Comparison of Programming Languages in Economics Research; Executor: S. Boraan Aruoba, Jesús Fernández-Villaverde. Cambridge, MA: 2014. 6 p. URL: <http://www.nber.org/papers/w20263.pdf>.
7. Julia interface to SymPy via PyCall. URL: <https://github.com/JuliaPy/SymPy.jl>.
8. Symata.jl. Symbolic mathematics language. URL: <https://github.com/jlapeyre/Symata.jl>.
9. *Wolfram S.* An Elementary Introduction to the Wolfram Language. 2015. 313 p. ISBN: 978.1-1944183-01-1. URL: <http://www.wolfram.com/language/elementary-introduction/>.
10. ModelingToolkit.jl. URL: <https://github.com/SciML/ModelingToolkit.jl>.
11. *Lämmel R.* Software Languages. Syntax, Semantics, and Metaprogramming. Cham: Springer International Publishing, 2018. XXX, 424 p. ISBN: 978-3-319-90798-7. URL: <http://link.springer.com/10.1007/978-3-319-90800-7>.
12. *Lotka A.J.* Elements of Physical Biology. Baltimore: Williams and Wilking Company, 1925. 435 p. URL: <https://archive.org/details/elementsofphysic017171mbp>.
13. *Kulyabov D.S., Korolkova A.V., Sevastianov L.A.* Two Formalism of Stochastification of One-Step Models // *Physics of Atomic Nuclei*. 2018. V. 81. № 6. P. 916–922. arXiv: 1908.04294.
14. *Rackauckas C., Nie Q.* DifferentialEquations.jl — A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia // *Journal of Open Research Software*. 2017. V. 5.

АЛГОРИТМ ПОСТРОЕНИЯ РЕЗУЛЬТАНТА
ДВУХ ЦЕЛЫХ ФУНКЦИЙ© 2021 г. В. И. Кузоватов^{а,*}, А. А. Кытманов^{а,**}, Е. К. Мышкина^{б,***}^а Сибирский федеральный университет,
660041 Красноярск, пр. Свободный, 79, Россия^б Институт вычислительного моделирования СО РАН,
660036 Красноярск, Академгородок, 50/44, Россия

*E-mail: kuzovатов@yandex.ru

**E-mail: aakytм@gmail.com

***E-mail: elfifenok@mail.ru

Поступила в редакцию 15.07.2020 г.

После доработки 03.09.2020 г.

Принята к публикации 04.09.2020 г.

На основе рекуррентных формул Ньютона приведен алгоритм построения результата двух целых функций, что является одним из методов исключения неизвестных для систем неалгебраических уравнений. Алгоритм реализован в системе компьютерной алгебры Maple. Приведены примеры, демонстрирующие работу данного алгоритма.

DOI: 10.31857/S0132347421020096

1. ВВЕДЕНИЕ

Одним из основных инструментов, открывших путь к созданию алгоритмических методов исследования и решения систем алгебраических уравнений, явилось понятие базиса Гребнера идеала кольца многочленов, занимающее одно из центральных мест в современной компьютерной алгебре (см., например, [1]). Одно из дальнейших развитий понятия базиса Гребнера приведено в [2]. Классические схемы исключения неизвестных из систем алгебраических уравнений, основанные на методе базисов Гребнера, реализованы во многих существующих системах компьютерной алгебры. Однако, такие методы неприменимы для исследования систем существенно неалгебраических уравнений (т.е. уравнений, не сводящихся к алгебраическим заменам переменных).

Вместе с тем, неалгебраические системы уравнений возникают в различных областях знания. В частности, в процессах, описываемых системами дифференциальных уравнений с правыми частями, разложимыми в ряд Тейлора, актуален вопрос об определении числа стационарных состояний в множествах определенного вида (и их локализации). Эта проблема приводит к задачам построения алгоритмов для определения числа корней заданной системы уравнений в различных множествах, определения самих корней, исключения части неизвестных из системы. В частности, в монографии [3] приведены многочислен-

ные примеры из химической кинетики, где требуются алгоритмы исключения неизвестных. Здесь важно применение разработанных методов для качественного и численного анализа математических моделей термокинетики процессов горения и катализа с целью получения условий воспламенения, взрыва и критических явлений в химически реагирующих системах. Для приложений, в том числе, например, для уравнений химической кинетики, важной задачей является исследование зависимостей решений систем нелинейных, в том числе и неалгебраических, уравнений от параметров. В вычислительном плане эта задача является достаточно трудоемкой. Ее степень сложности сильно зависит от размерности пространства неизвестных. Поэтому снижение этой размерности за счет исключения переменных может привести к упрощению исходной задачи.

Метод исключения неизвестных из систем нелинейных алгебраических уравнений, основанный на теории многомерных вычетов, был предложен Л.А. Айзенбергом в [4] в 1977 г. Дальнейшие модификации метода были предложены А.П. Южаковым, А.К. Цихом, В.И. Быковым, А.М. Кытмановым, М.З. Лазманом в конце прошлого века [3]. Эти идеи были в последствии развиты в работах [5–7]. Алгоритмический метод (разработанный на основе идей Л.А. Айзенберга и А.П. Южакова) был предложен М. Елкади и А. Ижером в работе [8]. Идея метода заключалась в нахожде-

нии определенных вычетов интегралов, связанных со степенными суммами корней (в положительных степенях) заданной системы уравнений, избегая нахождения самих корней и применяя затем к ним рекуррентные формулы Ньютона. По сравнению с классическим методом, данный метод сокращал время работы алгоритма, не повышая при этом кратность корней.

Еще одним методом исключения неизвестных служит построение результата двух целых функций. Хорошо известен классический результат Сильвестра для двух многочленов и метод исключения неизвестных, на нем основанный. Для неалгебраических функций такое понятие не было изучено ранее. Лишь в последние годы в работах [9, 10] обсуждается один подход к нахождению результата двух целых функций, основанный на рекуррентных формулах Ньютона.

Разработанный в данной работе алгоритм может быть также использован при исследовании дзета-функции корней некоторых классов целых функций, которые являются важным инструментом в создании методов исключения неизвестных из систем нелинейных уравнений, как было показано в работах [11–13].

2. КЛАССИЧЕСКИЙ РЕЗУЛЬТАНТ СИЛЬВЕСТРА И ЕГО ОБОБЩЕНИЯ

Напомним, что для данных многочленов f и g классический результат $R(f, g)$ может быть определен различными способами с использованием

а) определителя Сильвестра (см., например, [14–16]);

б) формулы для произведения $R(f, g) = \prod_{\{x: f(x)=0\}} g(x)$ (см., например, [14–16]);

в) способа Безу–Кэли (см., например, [17]).

В данной работе приведем конструктивное построение результата двух целых функций на комплексной плоскости. В качестве основного определения результата возьмем формулу произведения. Такой выбор объясняется тем, что целые функции являются естественным обобщением многочленов в комплексном анализе.

В ряде работ (см., например, [18]–[22]) различными авторами были предложены обобщения понятия результата для аналитических функций в кольце матричнозначных функций, мероморфных функций на римановой поверхности, для систем алгебраических уравнений. Во всех этих исследованиях предполагалось, что число нулей или полюсов конечно. Наш случай существенно отличается тем, что целые функции могут иметь бесконечное число нулей. Поэтому для нахождения результата необходим предельный переход.

Актуальность данной задачи определяется тем, что, например, в уравнениях химической кинетики возникают функции и системы уравнений, состоящие из экспоненциальных многочленов [23].

Первым шагом в нахождении результата двух целых функций явилась работа [9], в которой рассмотрен случай, когда одна из функций — целая, а вторая является многочленом (или целой функцией с конечным числом нулей). Условие конечности числа нулей целой функции изучено в [24]. В работе [25] обобщены результаты из [9] в случае, когда одна из целых функций удовлетворяет некоторым жестким ограничениям, но все-таки может иметь бесконечное число нулей.

Преимущество нашего подхода состоит в том, что он позволяет ответить на вопрос, имеют ли целые функции общие нули или нет, не вычисляя самих нулей. Итоговая формула для результата содержит степенные суммы корней, которые можно вычислить по формулам Ньютона, не прибегая к нахождению самих нулей.

3. РЕЗУЛЬТАНТ ДВУХ ЦЕЛЫХ ФУНКЦИЙ

Рассмотрим систему уравнений, состоящую из квадратного многочлена $f(z)$ и многочлена n -й степени $g(z)$

$$\begin{cases} f(z) = a_0 + a_1 z + z^2, \\ g(z) = b_0 + b_1 z + b_2 z^2 + \dots + b_n z^n. \end{cases} \quad (1)$$

Обозначим нули многочлена $f(z)$ через z_1, z_2 . Тогда (см., например, [10])

$$\begin{aligned} \prod_{i=1}^2 g(z_i) &= g(z_1)g(z_2) = b_0^2 + b_1^2 z_1 z_2 + \\ &+ b_2^2 z_1^2 z_2^2 + \dots + b_n^2 z_1^n z_2^n + b_0 b_1 (z_1 + z_2) + \\ &+ b_0 b_2 (z_1^2 + z_2^2) + \dots + b_0 b_n (z_1^n + z_2^n) + \\ &+ b_1 b_2 z_1 z_2 (z_1 + z_2) + b_1 b_3 z_1 z_2 (z_1^2 + z_2^2) + \dots + \\ &+ b_1 b_n z_1 z_2 (z_1^n + z_2^n) + b_2 b_3 z_1^2 z_2^2 (z_1 + z_2) + \\ &+ b_2 b_4 z_1^2 z_2^2 (z_1^2 + z_2^2) + \dots + b_2 b_n z_1^2 z_2^2 (z_1^n + z_2^n) + \\ &+ \dots + b_{n-1} b_n z_1^{n-1} z_2^{n-1} (z_1^n + z_2^n). \end{aligned}$$

Таким образом, мы получим

$$\begin{aligned} \prod_{i=1}^2 g(z_i) &= g(z_1)g(z_2) = \\ &= \sum_{k=0}^n b_k^2 z_1^k z_2^k + \sum_{t=0}^{n-1} \sum_{s=t+1}^n b_t b_s (z_1^t z_2^s + z_1^s z_2^t) = \\ &= \sum_{k=0}^n b_k^2 z_1^k z_2^k + \sum_{t=0}^{n-1} \sum_{s=t+1}^n b_t b_s z_1^t z_2^t (z_1^{s-t} + z_2^{s-t}). \end{aligned}$$

Напомним классические рекуррентные формулы Ньютона для многочленов. Они связывают между

собой коэффициенты многочлена и степенные суммы его корней.

Пусть

$$P(z) = z^m + c_1 z^{m-1} + \dots + c_{m-1} z + c_m.$$

Обозначим через $z_1, z_2, \dots, z_m$ его корни (среди них могут быть и кратные). Определим степенную сумму корней

$$S_k = z_1^k + \dots + z_m^k, \quad k \in \mathbb{N}, \quad S_0 = m.$$

Степенные суммы S_k и коэффициенты c_j связаны между собой классическими рекуррентными формулами Ньютона:

$$\begin{cases} S_j + \sum_{i=1}^{j-1} S_{j-i} c_i + j c_j = 0, & 1 \leq j \leq m, \\ S_j + \sum_{i=1}^m S_{j-i} c_i = 0, & j > m. \end{cases} \quad (2)$$

Используя классические формулы Виета

$$z_1 + z_2 = -a_1, \quad z_1 z_2 = a_0$$

и определение результата в виде формулы произведения, получаем следующий результат.

Теорема 1 ([10]). *Результант $R(f, g)$ системы многочленов вида (1) вычисляется по формуле*

$$R(f, g) = \sum_{k=0}^n b_k^2 a_0^k + \sum_{t=0}^{n-1} \sum_{s=t+1}^n b_t b_s a_0^t S_{s-t}, \quad (3)$$

где степенные суммы S_j корней многочлена $f(z)$ определяются рекуррентными формулами Ньютона.

Осуществив в (3) предельный переход по n , получаем утверждение о результате относительно целой функции и многочлена (или целой функции с конечным числом нулей).

Теорема 2. ([10]). *Пусть $g(z)$ — целая функция на комплексной плоскости $\mathbb{C}$ вида*

$$g(z) = b_0 + b_1 z + b_2 z^2 + \dots + b_n z^n + \dots,$$

а $f(z)$ — многочлен вида (1). Тогда результатом $R(f, g)$ является выражение

$$R(f, g) = \sum_{k=0}^{\infty} b_k^2 a_0^k + \sum_{t=0}^{\infty} \sum_{s=t+1}^{\infty} b_t b_s a_0^t S_{s-t},$$

где ряды в правой части абсолютно сходятся.

4. ОПИСАНИЕ АЛГОРИТМА

Алгоритм состоит из функции, реализующей вычисление степенных сумм с помощью классических формул Ньютона (2) и основной функции, вычисляющей результат системы (1), вычисленный по формуле (3).

5. ПРИМЕРЫ

Алгоритм 1: Алгоритм вычисления классических рекуррентных формул Ньютона.

Function *RecurrNewton* ($\{c_1, \dots, c_m\}, k$):

Input: Список коэффициентов $c_1, \dots, c_m$; натуральное число k .

Output: Список $S_1, \dots, S_k$ степенных сумм, вычисляющихся по формулам (2)

begin

$S_1 := -c_1$

for $j = 2 \dots \min(m, k)$ **do**

$S_j := -\sum_{i=1}^{j-1} S_{j-i} c_i - j c_j$

if $k > m$ **then**

for $j = m + 1 \dots k$ **do**

$S_j := -\sum_{i=1}^m S_{j-i} c_i$

return $\{S_1, \dots, S_k\}$

Алгоритм 2: Алгоритм построения результата двух целых функций.

Input: Список коэффициентов a_0, a_1 , список коэффициентов $b_0, b_1, \dots, b_n$.

Output: Результат системы (1), вычисленный по формуле (3).

begin

$S := \text{RecurrNewton}(\{a_0, a_1\}, n)$

$R := \sum_{k=0}^n b_k^2 a_0^k$

for $t = 0 \dots n - 1$ **do**

$R := R + \sum_{s=t+1}^n b_t b_s a_0^t S[s - t]$

return R

Алгоритм был реализован в среде Maple 2016 64bit. Полный код программы доступен по адресу <https://github.com/aakytmanov/Resultant2n>. Вычисления производились на машине Intel Core i7-4790 (3.6 GHz) с 32 Gb RAM под управлением Windows 7 Enterprise x64SP1. Время счета для приведенных примеров составило менее 0.2 секунды.

Пример 1. Рассмотрим систему уравнений ($n = 2$)

$$\begin{cases} f(z) = z(z-1) = z^2 - z, \\ g(z) = (z-2)(z+2) = z^2 - 4. \end{cases}$$

В данном случае

$$a_0 = 0, \quad a_1 = -1, \quad b_0 = -4, \quad b_1 = 0, \quad b_2 = 1.$$

Тогда формула (3) примет вид

$$R(f, g) = \sum_{k=0}^2 b_k^2 a_0^k + \sum_{t=0}^1 \sum_{s=t+1}^2 b_t b_s a_0^t S_{s-t}.$$

Поскольку $a_0 = 0$, то в этих суммах останутся лишь слагаемые, соответствующие $k = 0$ и $t = 0$, то есть

$$R(f, g) = b_0^2 + \sum_{s=1}^2 b_0 b_s S_s.$$

Поскольку $b_1 = 0$, то в одномерной сумме останутся лишь слагаемые при $s = 2$. Таким образом,

$$R(f, g) = b_0^2 + b_0 b_2 S_2.$$

Найдем степенную сумму корней S_2 , не используя значения самих корней, а именно по рекуррентным формулам Ньютона. Так, для нашего случая

$$S_2 = 1.$$

Таким образом, $R(f, g) = 12$.

Входными данными алгоритма в этом случае будут списки коэффициентов a_i и b_j :

$$> \text{Res}([0, -1], [-4, 0, 1]);$$

Пример 2. Данные методы приводят к вычислению сумм некоторых типов кратных числовых рядов, ранее отсутствовавших в известных справочниках. Эти результаты представляют самостоятельный интерес. Рассмотрим систему уравнений

$$\begin{cases} f(z) = z^2 - a^2, \\ g(z) = e^{bz} = 1 + bz + \frac{(bz)^2}{2!} + \dots + \frac{(bz)^n}{n!} + \dots \end{cases}$$

Вычисляя результат $R(f, g)$, используя теорему 2 с одной стороны, и используя определение результата в виде формулы для произведения с другой стороны, получим соотношение (см. [10])

$$\sum_{k=0}^{\infty} (-1)^k \frac{(ab)^{2k}}{(k!)^2} + 2 \sum_{t=0}^{\infty} \sum_{j=1}^{\infty} \frac{(-1)^t (ab)^{2t+2j}}{t!(t+2j)!} = 1.$$

При этом первая сумма является известной специальной функцией. Это функция Бесселя 1-го рода, а именно $J_0(2ab)$.

Частичные суммы выражения в левой части могут быть получены с помощью предложенного алгоритма, например, используя входные данные

$> \text{Res}([-a^2, 0], [1, b, b^2/2!, b^3/3!, b^4/4!, b^5/5!, b^6/6!]);$
получим

$$\begin{aligned} R(f, g) &= 1 + \frac{a^4 b^4}{3} - \frac{a^6 b^6}{40} - \frac{a^8 b^8}{960} + \frac{a^{10} b^{10}}{21600} + \\ &+ \frac{a^{12} b^{12}}{518400} - \frac{a^4 b^4}{3} + \frac{a^6 b^6}{40} + \frac{a^8 b^8}{720} = \\ &= 1 + \frac{a^8 b^8}{2880} + \frac{a^{10} b^{10}}{21600} + \frac{a^{12} b^{12}}{518400}. \end{aligned}$$

6. БЛАГОДАРНОСТИ

Исследования, представленные в работе, были выполнены при поддержке следующих фондов: второй автор использовал поддержку гранта РНФ 18-71-10007 (разработка и программная реализация алгоритма); третий автор поддержан грантом РФФИ 19-31-60012 (постановка технического задания для разработки алгоритма, тестирование, вычисление примеров).

СПИСОК ЛИТЕРАТУРЫ

1. von zur Gathen J., Gerhard J. Modern Computer Algebra (3rd edition), Cambridge University Press, 2013.
2. Канцов О.В. Системы образующих идеалов алгебры сходящихся дифференциальных рядов // Программирование. 2014. № 2. С. 32–40.
3. Vykov V.I., Kytmanov A.M., Lazman M.Z. Elimination methods in polynomial computer algebra. Dordrecht–Boston–Basel: Kluwer Academic Publishers, 1998.
4. Айзенберг Л.А. Об одной формуле обобщенного многомерного логарифмического вычета и решении систем нелинейных уравнений // Докл. АН СССР. 1977. Т. 234. № 3. С. 505–508.
5. Кйтманов А.А. Об аналогах рекуррентных формул Ньютона // Изв. вузов. Математика, 2009. № 10. С. 40–50.
6. Кйтманов А.А. Алгоритм вычисления степенных сумм корней для класса систем нелинейных уравнений // Программирование. 2010. Т. 36. № 2. С. 55–63.
7. Kytmanov A.A., Kytmanov A.M., Myshkina E.K. Finding Residue Integrals for Systems of Non-algebraic Equations in $\mathbb{C}^n$ // Journal of Symbolic Computation. 2015. V. 66. P. 98–110.
8. Elkadi M., Yger A. Residue calculus and applications // Publ. Res. Inst. Math. Sci. 2007. V. 43. № 1. P. 55–73.
9. Kytmanov A.M., Naprienko Ya.M. An approach to define the resultant of two entire functions // Journal Complex Variables and Elliptic Equations. 2017. V. 62. № 2. P. 269–286.
10. Kytmanov A.M., Myshkina E.K. On Some Approach for Finding the Resultant of Two Entire Functions // J. Siberian Federal Univ. Math. Phys. 2019. V. 12. № 4. P. 434–438.
11. Kytmanov A.M., Myslivets S.G. On the Zeta-Function of Systems of Nonlinear Equations // Siberian Math. J. 2007. V. 48. № 5. P. 863–870.

12. *Kuzovатов V.I., Kytmanov A.A.* On the Zeta-Function of Zeros of Some Class of Entire Functions // J. Siberian Federal Univ. Math. Phys. 2014. V. 7. № 4. P. 489–499.
13. *Кузоватов В.И., Кытманов А.А., Кузоватова О.И.* Алгоритм построения аналога формулы Бине // Программирование. 2020. № 2. С. 38–42.
14. *Бурбаки Н.* Алгебра. Многочлены и поля, упорядоченные группы. М.: Наука, 1965.
15. *Курош А.Г.* Курс высшей алгебры. М.: Наука, 1968.
16. *van der Waerden B.L.* Algebra. Berlin–Heidelberg–New York: Springer-Verlag, 1966.
17. *Krein M.G., Naimark M.A.* The Method of Symmetric and Hermitian Forms in the Theory of the Separation of the Roots of Algebraic Equation // Linear and Multilinear Algebra. 1981. V. 10. № 4. P. 265–308.
18. *Gohberg I.C., Heinig G.* Resultant Matrix and its Generalization. I. Resultant Operator of Matrix Polynomial // Acta Sci. Math. 1975. V. 72. P. 41–61.
19. *Gohberg I.C., Heinig G.* Resultant Matrix and its Generalization. II. Continual Analog of Resultant Matrix // Acta Math. Acad. Sci. Hungar. 1976. V. 28. P. 189–209.
20. *Gohberg I.C., Lerer L.E.* Resultant Operators of a Pair of Analytic Functions // Proc. Amer. Math. Soc. 1978. V. 72. № 1. P. 65–73.
21. *Gustafsson B., Tkachev V.G.* The Resultant on Compact Riemann Surfaces // Comm. Math. Phys. 2009. V. 10. P. 265–308.
22. *Morozov A.Yu., Shakirov Sh.R.* New and Old Results in Resultant Theory // Theor. Math. Phys. 2010. V. 163. № 2. P. 587–617.
23. *Быков В.И., Цыбенова С.Б.* Нелинейные модели химической кинетики. М.: КРАСАНД, 2011.
24. *Kytmanov A.M., Khodos O.V.* On Localization of Zeros of an Entire Function of Finite Order of Growth // Complex variables and operator theory. 2017. V. 11. № 2. P. 393–416.
25. *Kytmanov A.M., Khodos O.V.* An Approach to the Determination of the Resultant of Two Entire Functions // Russian Mathematics. 2018. V. 62. № 4. P. 42–51.

ПРОЦЕДУРЫ ПОИСКА УСЕЧЕННЫХ РЕШЕНИЙ ЛИНЕЙНЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ С БЕСКОНЕЧНЫМИ И УСЕЧЕННЫМИ СТЕПЕННЫМИ РЯДАМИ В РОЛИ КОЭФФИЦИЕНТОВ¹

© 2021 г. С. А. Абрамов^{a,*}, А. А. Рябенко^{a,**}, Д. Е. Хмельнов^{a,***}

^a Федеральный исследовательский центр “Информатика и управление” РАН,
Вычислительный центр РАН им. А.А. Дородницына, Москва, ул. Вавилова 40, Россия

* E-mail: sergeyabramov@mail.ru

** E-mail: anna.ryabenko@gmail.com

*** E-mail: dennis_khmelnov@mail.ru

Поступила в редакцию 01.08.2020 г.

После доработки 01.09.2020 г.

Принята к публикации 02.09.2020 г.

Предлагается пакет символьного построения экспоненциально-логарифмических решений таких линейных обыкновенных дифференциальных уравнений, которые могут иметь неполностью заданные (усеченные) коэффициенты — степенные ряды, для которых известны только начальные члены. Входящие в решения ряды также представляются конечным числом начальных членов. Для каждого такого ряда вычисляется максимально возможное число начальных членов, полностью определенных известными начальными фрагментами коэффициентов уравнения. При этом степень усечения каждого из этих рядов не может превосходить заданной пользователем величины. Последнее обеспечивает окончание вычисления и тогда, когда по известным фрагментам коэффициентов уравнения может быть определено любое число членов рядов, входящих в решения.

DOI: 10.31857/S0132347421020035

1. ВВЕДЕНИЕ

В [1] был предложен алгоритм построения формальных экспоненциально-логарифмических решений линейных обыкновенных дифференциальных уравнений, в роли коэффициентов которых выступают усеченные степенные ряды вида $a(x) + O(x^{t+1})$, где $a(x)$ — полином, $t \geq \deg a(x)$. Вместе с алгоритмом была описана и его предварительная реализация, выполненная с целью проверки работоспособности предложенного алгоритма. Статья [1] была продолжением статей [2, 3], где также рассматривалось построение решений уравнений с усеченными коэффициентами.

Как уже говорилось в [3], А.Д. Брюно в [4] предложил метод, также основанный на многоугольнике Ньютона, который для рядов, входящих в решения, позволяет найти любое число членов. Уравнения, в общем случае, — нелинейные, заданные полностью с помощью полностью (явно) заданных аналитических функций одной

или нескольких переменных. Очевидно, это другая задача.

Ниже речь идет о реализации предложенных в [1]–[3] алгоритмов для более общего случая: коэффициенты уравнения могут быть вперемешку усеченными и полностью заданными рядами. Предлагается пакет процедур с унифицированным интерфейсом, ориентированным на пользователя, применяющего в своей работе компьютерно-алгебраическое программное обеспечение. Сами процедуры дают, в сравнении с их предварительными вариантами, больше информации о решениях. Это будет продемонстрировано в разд. 6.

2. ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ

Пусть K — алгебраически замкнутое поле характеристики 0. Для кольца полиномов от x над K мы в дальнейшем используем обычное обозначение $K[x]$. Кольцо формальных степенных рядов от x над K обозначается через $K[[x]]$, поле формальных лорановых рядов — через $K((x))$.

¹ Частичная поддержка РФФИ, грант 19-01-00032.

Положим $\theta = x \frac{d}{dx}$. Пусть дано линейное обыкновенное дифференциальное уравнение

$$a_r(x)\theta^r y + a_{r-1}(x)\theta^{r-1} y + \dots + a_0(x)y = 0, \quad (1)$$

где y — неизвестная функция. Уравнение (1) может быть записано как $L(y) = 0$, где оператор L имеет вид

$$a_r(x)\theta^r + a_{r-1}(x)\theta^{r-1} + \dots + a_0(x). \quad (2)$$

К настоящему времени достаточно полно изучен случай, когда в роли коэффициентов уравнения выступают формальные степенные ряды от x над K : $a_i(x) \in K[[x]]$, $i = 0, \dots, r$.

Определение 1. Формальными экспоненциально-логарифмическими решениями уравнения (1) называются решения вида

$$e^{Q(x)} x^\lambda w(x^{1/q}), \quad (3)$$

где $Q(x) \in K[x^{-1/q}]$, $q \in \mathbb{Z}_{>0}$, $\lambda \in K$,

$$w(x) = \sum_{s=0}^m w_s(x) \ln^s x,$$

$w_s(x) \in K((x))$, $s = 0, \dots, m$, и $w_m(x) \neq 0$. При этом $x^\lambda w(x^{1/q})$ называется *регулярной частью*, $Q(x)$ — *показателем экспоненциальной части*, q — *индексом ветвления* решения (3).

При $q = 1$ и $Q(x) \in K$ решение (3) называется *формальным регулярным*, в противном случае — *нерегулярным*. При $q = 1$, $Q(x) \in K$, $\lambda \in \mathbb{Z}$ и $w(x) \in K((x))$ формальное регулярное решение (3) называется *лорановым*. В дальнейшем слово “формальный” мы опускаем, но подразумеваем.

Пусть $a_r(x) \neq 0$, т.е. порядок уравнения равен r . Известно (см., например, [5, гл. V, 6–8]), что существует r линейно независимых решений вида (3) для уравнения (1).

В [6]–[9] предложены алгоритмы нахождения для r линейно независимых решений вида (3) их индекса ветвления q и показателя экспоненциальной части $Q(x)$.

Определение 2. Для ненулевого лоранова ряда $a(x) = \sum a_j x^j \in K((x))$ его *валюация* $\text{val}a(x)$ определяется как $\min\{j \mid a_j \neq 0\}$, при этом $\text{val}0 = \infty$. Валюация оператора (2) считается равной минимуму валюаций всех его коэффициентов:

$$\text{val}L = \min_{0 \leq i \leq r} \{\text{val}a_i(x)\}.$$

Пусть $\text{val}L = 0$. Известно (см., например, [10]), что для построения индекса ветвления q и показателя экспоненциальной части $Q(x)$ для всех r линейно независимых решений уравнения $L(y) = 0$ достаточно знать значения r $\text{val}a_r(x)$ начальных коэффициентов всех $a_i(x)$, $i = 0, 1, \dots, r$.

Определение 3. Пусть $a(x) = \sum a_j x^j \in K((x))$ и пусть $t \in \mathbb{Z}$. Если $t \geq \text{val}a(x)$, выражение

$$a^{(t)}(x) = \sum_{j=\text{val}a(x)}^t a_j x^j + O(x^{t+1})$$

будем называть *t -усечением* ряда $a(x)$, число t — *степенью усечения ряда*, $\text{val}a(x)$ — *валюацией усечения*. Если же $t < \text{val}a(x)$ (в частности, если $a(x) = 0$), то будем называть его *t -усечением* выражение $a^{(t)}(x) = O(x^{t+1})$, число t — *степенью усечения*, $t + 1$ — *валюацией усечения*.

Алгоритмы, предложенные в [5, гл. IV, 11, 12, гл. II, VIII], можно применять для построения регулярной части решения с заданной степенью усечения входящих в $w(x)$ рядов. Для этого построения также достаточно знать некоторое конечное число начальных коэффициентов всех $a_i(x)$ ([13, Prop. 1]).

Пусть в качестве коэффициентов уравнения (1) выступают формальные степенные ряды, заданные алгоритмически, т.е. для любого коэффициента $a_i(x) = \sum_{j=0}^{\infty} a_{ij} x^j$ известен некоторый алгоритм Ξ_i , вычисляющий для $j \in \mathbb{Z}_{\geq 0}$ значение $\Xi_i(j) = a_{ij}$. Такие ряды мы будем называть *полностью заданными*. В этом случае для любого неотрицательного целого k с помощью соответствующих алгоритмов из [4–9, 11, 12] можно построить для каждого из r линейно независимых решений вида (3) его *k -усечение*:

$$e^{Q(x)} x^\lambda w^{(k)}(x^{1/q}),$$

где

$$w^{(k)}(x) = \sum_{s=0}^m w_s^{(k)}(x) \ln^s x$$

и $\text{val}w_0(x) = 0$. Система компьютерной алгебры Maple предлагает процедуру `DEtools:-formal_sol` для решения этой задачи применительно к уравнению с полиномиальными коэффициентами. Для уравнения (а также для систем уравнений) с коэффициентами, в роли которых выступают полностью заданные ряды, можно ис-

пользовать процедуры Maple-пакета, представленного в [14].

3. УРАВНЕНИЕ С УСЕЧЕННЫМИ КОЭФФИЦИЕНТАМИ

Пусть часть коэффициентов уравнения (1) — полностью заданные ряды, часть — усеченные. Каждый усеченный коэффициент $a_i(x)$ будем записывать как

$$a_i(x) = \sum_{j=0}^{t_i} a_{ij}x^j + O(x^{t_i+1}), \quad (4)$$

где $t_i \geq -1$ (при $t_i = -1$ сумма в (4) равна 0).

Определение 4. Продолжением уравнения (1) будем называть уравнение $\mathcal{L}(y) = 0$ с оператором

$$\mathcal{L} = \sum_{i=0}^r b_i(x)\theta^i \in K[[x]][\theta],$$

где

- $b_i(x) = a_i(x)$ для полностью заданного $a_i(x)$,
 - $b_i^{(t_i)}(x) = a_i(x)$ для усеченного $a_i(x)$,
- $i = 0, \dots, r$.

Определение 5. Пусть уравнение $L(y) = 0$ с полностью заданными и усеченными коэффициентами имеет вид (1). Усеченный лоранов ряд

$$w^{(k)}(x) = \sum_{j=j_0}^k w_j x^j + O(x^{k+1}), \quad (5)$$

в котором $j_0, k \in \mathbb{Z}$, $k \geq j_0$ и $w_{j_0} \neq 0$, мы называем *усеченным лорановым решением* уравнения $L(y) = 0$, коль скоро любое уравнение $\mathcal{L}(y) = 0$, являющееся продолжением $L(y) = 0$, имеет решение $\tilde{w}(x) \in K((x))$, усечением которого является (5):

$$\tilde{w}^{(k)}(x) = w^{(k)}(x).$$

Выражение

$$x^\lambda \sum_{s=0}^m w_s^{(k_s)}(x) \ln^s x, \quad (6)$$

в котором

$$w_s^{(k_s)}(x) = \sum_{j=j_s}^{k_s} w_{s,j} x^j + O(x^{k_s+1}), \quad (7)$$

$j_s, k_s \in \mathbb{Z}$, $k_s \geq j_s$, $s = 0, \dots, m$, и $w_{m,j_m} \neq 0$, мы называем *усеченным регулярным решением* уравнения $L(y) = 0$, коль скоро любое уравнение $\mathcal{L}(y) = 0$, являющееся продолжением $L(y) = 0$, имеет регу-

лярное решение $x^\lambda \tilde{w}(x)$, усечением которого является (6), т.е.

$$\tilde{w}(x) = \sum_{s=0}^m \tilde{w}_s(x) \ln^s x, \quad (8)$$

где $\tilde{w}_s^{(k_s)}(x) = w_s^{(k_s)}(x)$, $s = 0, 1, \dots, m$.

Пусть $Q(x) \in K[x^{-1/q}]$, $q \in \mathbb{Z}_{>0}$, $\lambda \in K$, усеченные ряды $w^{(k_s)}(x)$ имеют вид (7), $s = 0, 1, \dots, m$, и $w_{m,j_m} \neq 0$. Для уравнения $L(y) = 0$ выражение

$$e^{Q(x)} x^\lambda \sum_{s=0}^m w_s^{(k_s)}(x^{1/q}) \ln^s x \quad (9)$$

мы называем *решением с усеченной регулярной частью*, коль скоро любое уравнение $\mathcal{L}(y) = 0$, являющееся продолжением $L(y) = 0$, имеет решение $e^{Q(x)} x^\lambda \tilde{w}(x^{1/q})$, усечением которого является (9), т.е. $\tilde{w}(x)$ имеет вид (8) и выполняется $\tilde{w}_s^{(k_s)}(x) = w_s^{(k_s)}(x)$, $s = 0, 1, \dots, m$.

В случае, если все коэффициенты уравнения суть полностью заданные ряды, то, как было сказано выше, существуют алгоритмы и их реализация в системах компьютерной алгебры построения решений любого заданного усечения.

В случае уравнения, все коэффициенты которого являются усеченными рядами, усеченные решения содержат ряды, степень усечения которых не может быть произвольным числом. При этом, в [1–3, 15] показано, что возможно построение усеченных решений с максимальной степенью усечения входящих рядов. В названных статьях представлены алгоритмы решения этой задачи и их реализация в Maple.

Для “смешанного” случая в [16, Prop. 1] показано, что задача построения усеченных решений с максимальным числом входящих в решение рядов алгоритмически неразрешима, но и усеченное решение любой степени не всегда существует.

4. ОПРЕДЕЛЯЮЩИЙ ПОЛИНОМ

Пусть оператор L вида (2) имеет усеченные и полностью заданные коэффициенты и пусть выполняется

$$\min_{0 \leq i \leq r} \{\text{val} a_i(x)\} \leq \min_{0 \leq i \leq r} t_i, \quad (10)$$

где для каждого полностью заданного коэффициента $a_i(x)$ полагаем $t_i = \infty$. Обозначим через $\gamma = \min_{0 \leq i \leq r} \{\text{val} a_i(x)\}$. Выражение

$$I_L(n) = \sum_{i=0}^r a_{i,\gamma} n^i$$

называется *определяющим* полиномом уравнения $L(y) = 0$. Очевидно, что $I_L(n) \in K[n]$. В [2] показано, что если $I_L(n)$ имеет целые корни, то для $L(y) = 0$ существуют усеченные лорановы решения, в частности решения вида (5), где $j_0 = k$ и j_0 — максимальный целый корень $I_L(n)$. Иначе, никакое продолжение уравнения $L(y) = 0$ не имеет лорановых решений.

В [3] показано, что если $\deg I_L(n) > 0$, то для $L(y) = 0$ существуют усеченные регулярные решения, в частности решения вида (6), где $m = 0$, $j_0 = 0$, $k_0 = 0$ и λ — каждый из таких корней $I_L(n)$, что $I_L(\lambda + n) \neq 0$ для всех $n \in \mathbb{Z}_{>0}$. При отсутствии корней, т.е. $I_L(n) \in K \setminus \{0\}$, никакое продолжение уравнения $L(y) = 0$ не имеет регулярных решений.

Если условие (10) не выполняется, то будем говорить, что определяющий полином для $L(y) = 0$ не существует. В [1, предл. 2] показано, что в этом случае для $L(y) = 0$ не существует усеченных лорановых и регулярных решений. Но если для некоторого i ($0 < i \leq r$) выполняется $\text{val} a_i(x) = \gamma$, то каждое продолжение имеет по крайней мере i линейно независимых регулярных решений.

5. ПОРОГ УРАВНЕНИЯ

В [16] в связи с построением усеченных лорановых решений введено понятие порога уравнения. Это понятие перенесем на случаи усеченных регулярных решений и решений с усеченной регулярной частью.

Определение 6. Пусть уравнение $L(y) = 0$ с полностью заданными и усеченными коэффициентами имеет вид (1). Рассмотрим множество N всех таких целых значений k , что $L(y) = 0$ имеет решение с усеченной регулярной частью (9) с $j_m = 0$ и при этом максимальная степень усечения входящих в решение рядов равна $k = \max \{k_0, k_1, \dots, k_m\}$. Пусть множество N не пусто и имеет максимальный элемент. Этот элемент мы назовем *порогом* исходного уравнения для экспоненциально-логарифмических решений. Если в множестве N содержатся сколь угодно большие целые числа, то полагаем порог уравнения равным ∞ . Если же это множество пусто и выполняется условие (10), то считаем, что порог равен $-\infty$. Для пустого N при невыполнении (10) будем считать, что значение порога не определено.

Условие $j_m = 0$ в этом определении обеспечивает однозначность значения порога уравнения.

Аналогично вводятся понятия порога уравнения для регулярных решений и порога уравнения для лорановых решений.

Пусть для $L(y) = 0$ существует определяющий полином $I_L(n)$. Если $I_L(n)$ имеет целые корни, то порог уравнения для лорановых решений определен и его значение (целое число или ∞) не меньше максимального целого корня $I_L(n)$. При этом же условии порог уравнения для регулярных решений и решений с усеченной регулярной частью — неотрицательное целое или ∞ .

Предложение 1. *Невозможен алгоритм, позволяющий по произвольному уравнению $L(y) = 0$ вида (1) с полностью заданными и усеченными коэффициентами, выяснять, будет ли конечным или же бесконечным его порог как для лорановых, так и для регулярных, экспоненциально-логарифмических решений.*

Доказательство. В [16, предл. 1] это утверждение доказано для порога лорановых решений. Из того, что лораново решение является частным случаем регулярного и экспоненциально-логарифмического решений, следует общее утверждение.

Предложение 2. *Пусть $L(y) = 0$ — уравнение вида (1) и пусть $d \in \mathbb{Z}$. Алгоритмически может быть проверено, верно ли, что значение порога определено для данного уравнения и это значение меньше, чем d ; при положительном ответе порог h данного уравнения может быть найден. Дополнительно могут быть построены все такие решения с усеченной регулярной частью (в частности, усеченные лорановы, регулярные решения), степени усечения которых не превосходят h .*

Доказательство. Для лорановых усеченных решений в [16, предл. 1] это утверждение доказано для случая, когда выполняется условие (10). Если (10) не выполняется, то для $L(y) = 0$ существуют как продолжения, имеющие лорановы решения, так и продолжения, их не имеющие, т.е. значение порога не определено.

Для регулярных усеченных решений при выполнении условия (10) доказательство аналогично случаю лорановых усеченных решений. Если (10) не выполняется, то не существует усеченных регулярных решений (см. [1, предл. 2]), более того, если некоторое продолжение уравнения $L(y) = 0$ не имеет регулярных решений, то найдется такое продолжение, для которого имеются регулярные решения.

В [1] приведен алгоритм построения всех таких показателей $Q(x)$ экспоненциальной части и

соответствующих индексов ветвления q , для которых существуют решения с усеченной регулярной частью. Для каждого построенного $Q(x)$, q получим новое уравнение $L_1(z) = 0$ с помощью подстановки

$$y(x) = e^{Q(t^q)} z(t), \quad x = t^q$$

в уравнение $L(y) = 0$. Каждый коэффициент нового уравнения может быть как полностью заданным, так и усеченным рядом. Для $L_1(z) = 0$ решаем задачу поиска порога для усеченных регулярных решений при заданном d . Если для нового уравнения не существует определяющего полинома (т.е. значение порога не определено), то полагаем $h(Q, q) = FAIL$. Если d превосходит порог, то получено значение порога, полагаем $h(Q, q)$ ему равным, иначе полагаем $h(Q, q) = d$.

Для исходного уравнения решим так же задачу поиска порога регулярных решений. Обозначим результат через $h(0, 1)$. Тогда $h(0, 1)$ равен либо $FAIL$, либо d , либо значению порога для регулярных решений (целое число или $-\infty$).

Если для каких-либо $Q(x)$, q (включая $Q(x) = 0$, $q = 1$) получим $h(Q, q) = d$, то порог для решений с усеченной регулярной частью не найден. Если для всех $Q(x)$, q (включая $Q(x) = 0$, $q = 1$) получено $h(Q, q) = FAIL$, то понятие порога исходного уравнения для решений с усеченной регулярной частью не определено. Иначе порогом является максимальное значение из всех тех $h(Q, q)$, значение которых не равно $FAIL$.

6. РЕШЕНИЕ С УСЕЧЕННЫМ ПОКАЗАТЕЛЕМ ЭКСПОНЕНЦИАЛЬНОЙ ЧАСТИ

Определение 7. Пусть уравнение $L(y) = 0$ с полностью заданными и усеченными коэффициентами имеет вид (1). Пусть

$$Q(x) = \frac{\varepsilon_1}{x^{\kappa_1}} + \dots + \frac{\varepsilon_\sigma}{x^{\kappa_\sigma}},$$

где $\kappa_1 > \dots > \kappa_\sigma > 0$, $\kappa_i \in \mathbb{Q}$ и $\varepsilon_i \in K \setminus \{0\}$, $i = 1, \dots, \sigma$. Выражение

$$e^{Q(x)} Y(x) \quad (11)$$

мы называем *решением с усеченным показателем экспоненциальной части* уравнения $L(y) = 0$, коль скоро любое уравнение $\mathcal{L}(y) = 0$, являющееся продолжением $L(y) = 0$, имеет нерегулярное решение

$$e^{Q(x) + \tilde{Q}(x)} x^\lambda \sum_{s=0}^m w_s(x^{1/q}), \quad (12)$$

такое, что

$$\tilde{Q}(x) = \frac{\varepsilon_{\sigma+1}}{x^{\kappa_{\sigma+1}}} + \dots + \frac{\varepsilon_{\sigma+\zeta}}{x^{\kappa_{\sigma+\zeta}}} \quad (13)$$

где $\kappa_\sigma > \kappa_{\sigma+1} > \dots > \kappa_{\sigma+\zeta} > 0$, $\kappa_i \in \mathbb{Q}$ и $\varepsilon_i \in K \setminus \{0\}$, $i = \sigma + 1, \dots, \sigma + \zeta$.

В [1] представлен алгоритм и его реализация в Maple построения для уравнения вида (1) решения вида (11) с максимальным числом слагаемых в $Q(x)$. Кроме этого может быть получена дополнительная информация, инвариантная относительно продолжений уравнения:

- для всех продолжений усеченного уравнения существует решение (12) с $\zeta = 0$, т.е. показатель экспоненциальной части $Q(x)$ построен полностью, индекс ветвления q также инвариантен относительно продолжений уравнения, но не существует инвариантного показателя λ в регулярной части; такое усеченное решение будем обозначать

$$e^{Q(x)} y_{reg}(x^{1/q}),$$

- для всех продолжений усеченного уравнения существует решение (12) с $\zeta \geq 1$, и инвариантным является значение $\kappa_{\sigma+1}$, тогда как $\varepsilon_{\sigma+1}$ неинвариантно относительно продолжений уравнения; такое усеченное решение будем обозначать

$$e^{Q(x)} y_{irr(\kappa_{\sigma+1})}(x),$$

- для всех продолжений усеченного уравнения существует решение (12) с $\zeta \geq 1$, и значение $\kappa_{\sigma+1}$ неинвариантно относительно продолжений уравнения; такое усеченное решение будем обозначать

$$e^{Q(x)} y_{irr}(x).$$

Если же существует продолжение уравнения $L(y) = 0$, имеющее решение (12) с показателем экспоненциальной части $Q(x)$ ($\zeta = 0$), и продолжение уравнения, такого решения не имеющее ($\zeta \geq 1$), то для такого усеченного решения оставим обозначение (11).

7. ПРОЦЕДУРЫ ПОСТРОЕНИЯ РЕШЕНИЙ

Алгоритмы построения усеченных решений реализованы в системе компьютерной алгебры Maple 2020 [17] в виде процедур пакета TruncatedSeries². В [15] представлены процедуры LaurentSolution (построение лорановых решений) и RegularSolution (построение регулярных решений) для случая уравнений, все коэффициенты которого являются усеченными рядами. Для того же случая в [1] пред-

² Пакет и сессии Maple с примерами использования пакета доступны по адресу <http://www.ccas.ru/ca/TruncatedSeries>

ставлена предварительная реализация процедуры `FormalSolution` (построение экспоненциально-логарифмических решений в виде (11)). Расширение возможностей процедуры `LaurentSolution` на “смешанный” случай представлено в [16].

Основные аргументы процедур:

1) однородное линейное обыкновенное дифференциальное уравнение с усеченными или полностью заданными рядами в роли коэффициентов;

2) неизвестная функция, например, $y(x)$.

Уравнение должно быть записано через оператор $\theta = x \frac{d}{dx}$, либо через оператор $\frac{d}{dx}$. Применение θ^i к $y(x)$ записывается в пакете `TruncatedSeries` как `theta(y(x), x, i)`. Для обозначения i -й производной функции $y(x)$ используется стандартное в Maple `diff(y(x), x$ i)`.

Усеченные коэффициенты уравнения задаются в виде выражений $a(x) + O(x^{t+1})$, где $a(x)$ — полином степени не выше t над $\mathbb{A}$ (т.е. над полем алгебраических чисел).

Полностью заданный ряд $\sum_{i=0}^{\infty} a_i x^i$ может быть записан либо как полином (т.е. $a_i = 0$, начиная с некоторого значения i), либо как сумма полинома (начальные члены ряда) и бесконечной степенной суммы, для представления которой должна быть использована Maple-конструкция

`Sum(f(i)*x^i, i = i0..infinity)`,

где $f(i)$ — выражение для вычисления коэффициента ряда при x^i , i — имя (индекс суммирования), $i0$ — неотрицательное целое.

Для корректной работы процедур нашего пакета, нерациональные алгебраические числа, входящие в значение $f(i)$ при всех целых значениях i , больших либо равных $i0$, должны быть представлены с помощью Maple-конструкции

`RootOf(p(_Z), index = k)`,

где $p(_Z)$ — неприводимый полином, k -м корнем которого и является данное алгебраическое число.

Например, `RootOf(_Z^2 - 2, index = 2)` представляет число $-\sqrt{2}$. То же касается и нерациональных чисел, входящих в полиномы, задающие коэффициенты уравнения.

Могут быть также указаны необязательные аргументы (опции):

- `'top' = d`, где d — целое число, для которого нужно определить превосходит ли оно порог данного уравнения (по умолчанию, если все коэффициенты данного уравнения являются усеченными рядами, d равно порогу уравнения, иначе d полагается

равным нижней границе порога уравнения, его значение указано отдельно для каждой процедуры);

- `'threshold' = 'h'`, где h — имя переменной, которой будет присвоено значение порога, если его удалось вычислить, либо Maple-константа `FAIL`, если понятие порога не определено или значение порога превышает значение d .

7.1. Лорановы решения

Результат работы процедуры `LaurentSolution`:

- список усеченных лорановых решений с различными валуациями — список содержит все усеченные лорановы решения, инвариантные относительно продолжений уравнения, количество вычисленных коэффициентов в каждом усечении является максимально возможным с учетом значения d ;

- пустой список, если лорановых решений не существует для всех продолжений уравнения;

- Maple-константа `FAIL`, иначе.

Каждое из усеченных решений из списка, возвращаемого `LaurentSolution`, представлено в виде (5), где w_j — вычисленные коэффициенты усеченного лоранова решения — являются линейными комбинациями над $\mathbb{A}$ произвольных постоянных $_c1, _c2, \dots$. Валюации всех элементов списка различны — произвольная постоянная, являющаяся коэффициентом при степени, равной валуации некоторого элемента списка, не может быть взята равной 0 в соответствующем усеченном решении.

Обозначим через h^* максимальный целый корень определяющего полинома, либо $-\infty$, если целых корней нет. Порог уравнения для лорановых решений, как указано в разд. 4, не меньше h^* . Если опция `'top'` отсутствует, либо с ее помощью задано значение d , меньшее h^* , то d полагается равным h^* .

Степени усечения разных элементов списка могут быть различны, поскольку для разных валуаций может отличаться максимальное количество коэффициентов, инвариантных относительно продолжений уравнения. Степень усечения каждого элемента не превосходит d и порог уравнения для лорановых решений, но не меньше h^* .

7.2. Регулярные решения

Результат работы процедуры `RegularSolution`:

- список усеченных регулярных решений с различными наборами валуаций входящих в решения усеченных лорановых рядов — список содержит все усеченные регулярные решения, инвариантные относительно продолжений уравнения, количество вычисленных коэффициентов в каждом лорановом ряду является максимально возможным с учетом значения d ;

- пустой список, если усеченных регулярных решений не существует для всех продолжений уравнения;

- Maple-константа FAIL, иначе.

Каждое усеченное решение y_i из списка, возвращаемого RegularSolution, является конечной суммой $y_{i1} + \dots + y_{ig}$ выражений вида

$$y_{ij} = x^{\lambda_j} \sum_{s=0}^{m_{i,j}} w_{i,j,s}(x) \ln^s x,$$

где $\lambda_1, \dots, \lambda_g$ — все такие различные корни $I_L(n)$, что $I_L(\lambda_j + n) \neq 0$ для всех $n \in \mathbb{Z}_{>0}$, $w_{i,j,s}(x)$ — усеченные лорановы ряды, $j = 1, \dots, g$, $s = 0, \dots, m_{i,j}$. В каждой паре различных усеченных решений y_{i_1} и y_{i_2} из списка, для каждого λ_j либо $m_{i_1,j} \neq m_{i_2,j}$, либо для некоторого s ($0 \leq s \leq m_{i_1,j}$)

$$\text{val} w_{i_1,j,s}(x) \neq \text{val} w_{i_2,j,s}(x).$$

Коэффициенты всех усеченных рядов являются линейными комбинациями над $\mathbb{A}$ произвольных постоянных $_{c_1}, _{c_2}, \dots$. Произвольная постоянная, являющаяся коэффициентом при степени, равной валуации некоторого ряда, входящего в элемент списка, не может быть взята равной 0 в соответствующем усеченном решении.

Если существуют корни определяющего полинома, положим $h^* = 0$, иначе $h^* = -\infty$. Порог уравнения для регулярных решений, как указано в разд. 4, не меньше h^* . Если опция 'top' отсутствует, либо с ее помощью задано значение d , меньшее h^* , то d полагается равным h^* .

Степени усечения рядов в разных элементах списка могут быть различны, поскольку для разных валуаций может отличаться максимальное количество коэффициентов, инвариантных относительно продолжений уравнения. Степень усечения каждого ряда в решении не превосходит d и порог уравнения для регулярных решений, но не меньше h^* .

7.3. Формальные решения

Результат работы процедуры FormalSolution:

- список усеченных решений уравнения;
- Maple-константа FAIL, если усеченных решений не существует.

Каждое усеченное решение из списка, возвращаемого FormalSolution, является конечной суммой выражений вида $e^Q Y$, где $Q \in \mathbb{A}[x^{-1/q}]$, $q \in \mathbb{Z}_{>0}$, при этом возможно, что $Q = 0$, $q = 1$. Для Y возможны следующие варианты:

- усеченная регулярная часть решения, содержит произвольные постоянные $_{c_1}, _{c_2}, \dots$;

- $y_{reg,i}(x^{1/q})$ (где $q \in \mathbb{Z}_{>0}$), $y_{irr(\kappa),i}(x)$ (где $\kappa \in \mathbb{Q}$), $y_{irr,i}(x)$ либо $y_i(x)$, где $i \in \mathbb{Z}_{>0}$, для решений с усеченным показателем экспоненциальной части тех видов, которые перечислены в разд. 6.

Если для уравнения существует решение с показателем экспоненциальной степени Q и индексом ветвления q и усеченной регулярной частью Y , то для построения этой регулярной части применяется процедура RegularSolution к уравнению, полученному с помощью подстановки $y(x) = e^{Q(x^{1/q})} z(t)$, $x = t^q$ в исходное уравнение. Для каждого нового уравнения, в зависимости от того, все ли его коэффициенты являются усеченными или нет, определяется значение d . Все элементы списка, построенного RegularSolution, входят в результат FormalSolution.

7.4. Примеры

Приведем несколько примеров применения процедур пакета.

Уравнение

$$(-1 + x + x^2 + O(x^3))\theta^2 y - (2 + O(x^3))\theta y + \left(\sum_{i=4}^{\infty} (i^2 + 2i + 1 - (i+1)^2)x^i \right) y = 0$$

в системе Maple записывается следующим образом:

```
> eq1 := (-1+x+x^2+O(x^3))
      *theta(y(x), x, 2) -
      (2+O(x^3))*theta(y(x), x, 1)+
      Sum((i^2+2*i+1-(i+1)^2)*x^i,
      i=4..infinity)*y(x):
```

В этом уравнении присутствуют как усеченные коэффициенты, так и коэффициент, заданный алгоритмически с помощью бесконечной суммы. Применим процедуру LaurentSolution к eq1 с указанием опций:

```
> LaurentSolution(eq1, y(x),
                  'top'=7, 'threshold'='h');
```

$$\left[\frac{-c_1}{x^2} - \frac{4-c_1}{x} + -c_2 + O(x), -c_2 + O(x^8) \right]$$

Полученный ответ означает, что для всех продолжений уравнения существуют лорановы решения с валюациями, равными -2 и 0 . Степень усечения для разных валюаций различна. Отметим, что применение к этому уравнению процедуры `RegularSolution` дает тот же результат, то есть в данном случае все усеченные регулярные решения являются усеченными лорановыми решениями.

Проверим значение порога

$>h;$

FAIL

Это означает, что заданное с помощью опции `'top'` значение не превосходит порога. Аналогичный результат получается для любого значения этой опции, поскольку в данном случае порог уравнения равен ∞ , и любое конечное значение опции не превосходит значение этого порога. При этом, видно, что для валюации -2 уже достигнута максимально возможная степень усечения, поскольку она меньше значения, заданного с помощью опции `'top'`, и она не будет меняться при увеличении значения этой опции.

Уравнение

$$(x^7 + x^5) \frac{d^4 y}{dx^4} + \left(18x^4 + \sum_{k=7}^{\infty} \frac{(-1)^k x^k}{k!} \right) \frac{d^3 y}{dx^3} + (96x^3 + O(x^6)) \frac{d^2 y}{dx^2} + (168x^2 + O(x^5)) \frac{dy}{dx} + 72xy = 0$$

в системе `Maple` записывается следующим образом:

```
> eq2 := (x^7+x^5)*diff(y(x), x$4) +
(18*x^4 + Sum((-1)^k*x^k/k!,
k=7..infinity))*diff(y(x), x$3) +
(96*x^3 + O(x^6))*diff(y(x), x$2) +
(168*x^2 + O(x^5))*diff(y(x), x) +
72*x*y(x) :
```

В этом уравнении присутствуют как усеченные коэффициенты, так и коэффициенты, заданные алгоритмически (и в виде полинома, и с помощью бесконечной суммы). Отметим, что уравнение записано с помощью обычного дифференцирования. Также отметим, что если наименьшая валюация коэффициентов уравнения равна $\gamma > 0$ (для `eq2` $\gamma = 1$), и если при этом $t_i \geq \gamma$ для каждого усеченного коэффициента $a_i(x)$, то процедуры пакета выполняют деление уравнения на x^γ и за-

меняют каждое t_i на $t_i - \gamma$. Применим процедуру `RegularSolution` к `eq2` с указанием опций:

```
> RegularSolution(eq2, y(x),
'top'=3, 'threshold'='h');
```

$$\left[\frac{1}{x^2} \left(-\frac{1}{420} \frac{c_2}{x^2} + \frac{c_3}{x} + -c_4 + O(x) + \ln(x) \left(\frac{-c_1}{x} + -c_2 - 30x_{-c_1} + O(x^2) \right) \right), \right.$$

$$\left. \frac{1}{x^2} \left(-\frac{1}{420} \frac{c_2}{x^2} + \frac{c_3}{x} + -c_4 + O(x) + \ln(x) \left(-c_2 - \frac{5}{3} x^2_{-c_2} + O(x^3) \right) \right), \right.$$

$$\left. \frac{1}{x^2} \left(\frac{c_3}{x} + -c_4 + x \left(-30_{-c_3} + \frac{197}{2}_{-c_1} \right) + O(x^2) + \ln(x) \left(\frac{-c_1}{x} - 30x_{-c_1} + O(x^2) \right) \right), \right.$$

$$\left. \frac{1}{x^2} \left(\frac{c_3}{x} + -c_4 - 30x_{-c_3} + O(x^2) \right), \right.$$

$$\left. \frac{1}{x^2} \left(-c_4 + \frac{197}{2} x_{-c_1} + O(x^2) + \ln(x) \left(\frac{-c_1}{x} - 30x_{-c_1} + O(x^2) \right) \right), \right.$$

$$\left. \frac{1}{x^2} \left(-c_4 - \frac{5}{3} x^2_{-c_4} + O(x^3) \right), \right.$$

$$\left. \frac{1}{x^2} \left(\frac{197}{2} x_{-c_1} + O(x^2) + \ln(x) \left(\frac{-c_1}{x} - 30x_{-c_1} + O(x^2) \right) \right) \right]$$

Полученный ответ означает, что для всех продолжений уравнения существуют регулярные решения. Пять элементов списка соответствуют решениям с $\ln x$ и различными валюациями входящих в решения лорановых рядов. Два других элемента списка не содержат $\ln x$ и представляют собой лорановы решения с различными валюациями (эти решения могут быть найдены с помощью процедуры `LaurentSolution`). Видно, что для разных валюаций входящие в решения усеченные ряды могут иметь разную степень усечения.

Проверим значение порога

$> h;$

2

Найденный порог не превышает, как и должно быть, заданного с помощью опции `'top'` значения. Легко видеть, что он достигается в тех реше-

ниях, которые содержат ряды с валюацией, равной 0.

Уравнение

$$(x^4 + O(x^7))\theta^3 y + (3x + O(x))\theta^2 y + \left(1 + \sum_{i=1}^{\infty} ix^i\right)\theta y = 0$$

в системе Maple записывается следующим образом:

```
> eq3 := (x^4 + O(x^7))*theta(y(x), x, 3) +
(3*x + O(x^5))*theta(y(x), x, 2) +
(1 + Sum(i*x^i, i = 1 .. infinity))*
theta(y(x), x, 1):
```

Применим процедуру FormalSolution к eq3 без опции 'top':

```
> FormalSolution(eq3, y(x), 'threshold'='h');
```

$$\left[\begin{aligned} & -c_1 + O(x) + e^{\frac{1}{3x}} x^{2/3} \left(-c_2 + \frac{35}{27} c_2 x + \right. \\ & \left. \frac{8947}{1458} c_2 x^2 + O(x^3) \right) + e^{\frac{1}{x^3} - \frac{1}{3x}} y_{reg}(x) \end{aligned} \right]$$

Полученный ответ означает, что для всех продолжений уравнения существует одномерное пространство лорановых решений вида $(-c_1 + O(x))$, где $-c_1 \in \mathbb{A}$. Проверка значения порога уравнения $> h$;

FAIL

показывает, что для лорановых решений порог не достигнут, их степень усечения определяется по умолчанию: $d = 0$.

Так же eq3 имеет решение с усеченной регулярной частью, причем для регулярной части вычислено больше слагаемых, поскольку при подстановке $y(x) = e^{1/(3x)} z(x)$ в eq3 получится уравнение, все коэффициенты которого являются усеченными рядами. Для нового уравнения значение d по умолчанию равно порогу, который является конечным числом.

Еще одно усеченное решение не содержит рядов. Все продолжения eq3 имеют решения вида (12), где

$$Q(x) = \frac{1}{x^3} - \frac{1}{3x},$$

$\tilde{Q}(x) = 0$, $q = 1$, но значения λ различны для разных продолжений, поэтому регулярная часть в ответе обозначена $y_{reg}(x)$.

В [1–3, 14–16, 18] можно найти другие примеры работы процедур пакета TruncatedSeries.

Авторы благодарят компанию Maplesoft (Ватерлоо, Канада) за консультации и дискуссии.

СПИСОК ЛИТЕРАТУРЫ

1. Абрамов С.А., Рябенко А.А., Хмельнов Д.Е. Усеченные ряды и формальные экспоненциально-логарифмические решения линейных обыкновенных дифференциальных уравнений // Ж. вычисл. матем. и матем. физ. 2020. Т. 60. № 10. С. 1664–1675.
2. Абрамов С.А., Рябенко А.А., Хмельнов Д.Е. Линейные обыкновенные дифференциальные уравнения и усеченные ряды // Ж. вычисл. матем. и матем. физ. 2019. Т. 59. № 10. С. 66–77.
3. Абрамов С.А., Рябенко А.А., Хмельнов Д.Е. Регулярные решения линейных обыкновенных дифференциальных уравнений и усеченные ряды // Ж. вычисл. матем. и матем. физ. 2020. Т. 60. № 1. С. 4–17.
4. Брюно А.Д. Асимптотики и разложения решений обыкновенного дифференциального уравнения // Успехи математических наук. 2004. Т. 59. Вып. 3 (357). С. 31–80.
5. Коддингтон Э.А., Левинсон Н. Теория обыкновенных дифференциальных уравнений. М.: Изд-во иностр. лит., 1958.
6. Malgrange B. Sur la réduction formelle des équations différentielles à singularités irrégulières. Université Scientifique et Médicale de Grenoble. 1979.
7. Tournier E. Solutions formelles d'équations différentielles. Le logiciel de calcul formel DESIR Étude théorique et réalisation // Thèse d'État. Université de Grenoble. 1987.
8. Barkatou M. Rational Newton algorithm for computing formal solutions of linear differential equations // Lecture Notes in Computer Science. 1989. V. 358. P. 183–195.
9. Баркату М., Ришар-Жюне Ф. Формальные решения линейных дифференциальных и разностных уравнений // Программирование. 1997. № 2. С. 24–42.
10. Lutz D.A., Schäfke R. On the identification and stability of formal invariants for singular differential equations // Linear Algebra And Its Applications. 1985. V. 72. P. 1–46.
11. Frobenius G. Integration der linearen Differentialgleichungen mit veränder Koeffizienten // J. für die reine und angewandte Mathematik. 1873. V. 76. P. 214–235.
12. Heffter L. Einleitung in die Theorie der linearen Differentialgleichungen. Leipzig: Teubner, 1894.
13. Abramov S.A., Barkatou M.A., Pfluegel E. Higher-order linear differential systems with truncated coefficients // In Proc. of CASC'2011. 2011. P. 10–24.

14. *Абрамов С.А., Рябенко А.А., Хмельнов Д.Е.* Процедуры поиска локальных решений линейных дифференциальных систем с бесконечными степенными рядами в роли коэффициентов // Программирование. 2016. № 2. С. 75–86.
15. *Абрамов С.А., Рябенко А.А., Хмельнов Д.Е.* Процедуры поиска лорановых и регулярных решений линейных дифференциальных уравнений с усеченными степенными рядами в роли коэффициентов // Труды ИСП РАН. 2019. Т. 31. № 5. С. 233–248.
16. *Abramov S., Khmelnov D., Ryabenko A.* Truncated and infinite power series in the role of coefficients of linear ordinary differential equations // In Proc. of CASC'2020 Lecture Notes in Computer Science. 2020. V. 12291. P. 63–76..
17. Maple online help // <http://www.maplesoft.com/support/help/>
18. Страница пакета TruncatedSeries // <http://www.ccas.ru/ca/TruncatedSeries>

РЕДАКЦИОННЫЕ ПРАВИЛА ДЛЯ АВТОРОВ ЖУРНАЛА “ПРОГРАММИРОВАНИЕ”

DOI: 10.31857/S0132347421100010

Журнал “Программирование” выходит 6 раз в год, каждые два месяца. В журнале публикуются только оригинальные статьи, ранее не публиковавшиеся в других изданиях. Объем публикуемых статей, как правило, не должен превышать 20 страниц. Публикации в журнале бесплатны для авторов. Прием статей для публикации осуществляется непрерывно. Все статьи проходят обязательное предварительное рецензирование. Рецензенты назначаются редакционной коллегией в соответствии с тематикой представленной работы. Журнал применяет практику двойной анонимности рецензирования, при которой авторы не знают, кто рецензирует их статьи, а рецензенты, не знают, чьи статьи они рецензируют. Принятая и получившая положительные рецензии статья печатается в очередном номере в порядке общей очереди.

1. Общие требования к подаче рукописи:

1. Подачу рукописей осуществляется через портал <https://sciencejournals.ru> после прохождения процедуры регистрации автора.

2. Отправляя рукопись в журнал, автор гарантирует, что соответствующий материал (в оригинале или в переводе на другие языки или с других языков) ранее нигде не публиковался и не находится на рассмотрении для публикации в других издательствах.

3. Для принятия редколлегией решения о публикации статьи в журнале авторам необходимо представить в редакцию рукопись статьи и авторский договор с издателем журнала, заполненный и подписанный автором и всеми соавторами. Авторский договор вступает в силу в случае и с момента принятия статьи к публикации.

4. В состав электронной версии рукописи должны входить файлы, содержащие все элементы статьи (текст, таблицы) в форматах DOC (или DOCX, RTF) и исходные файлы иллюстраций, соответствующие требованиям (см. ниже требования к иллюстрациям). Также принимаются файлы рукописи в форматах Tex и pdf.

5. Файлы рукописи и сопроводительных документов должны быть собраны в один архив (желательно ZIP). Файлы рукописи должны учитывать замечания рецензента и правку редактора от редакции.

6. Рукописи для рецензирования принимают только на русском языке.

7. Все материалы, поступившие для публикации, проходят анонимное рецензирование. Рукописи после рецензирования не возвращаются. Редакция вправе не вступать в переписку с авторами относительно причин (оснований) отказа в публикации статьи.

8. После принятия рукописи к публикации автор не может вносить в нее серьезные изменения (добавлять и изменять абзацы, менять иллюстрации и список литературы).

2. Структура рукописи

Обязательными являются следующие элементы статьи:

1. Название статьи.

2. Полный список авторов (инициалы и фамилии). Необходимо указать, кто из авторов ответственный за переписку.

3. Место работы авторов. Полное (без сокращений) название организации, почтовый адрес с указанием города, страны и почтового индекса. Если авторы работают в разных организациях, то должно быть понятно, кто и в какой именно организации работает.

4. Электронный адрес автора, ответственного за переписку. Так как статьи для проверки авторам рассылаются только по электронной почте, то в случае, когда у статьи только один автор, желательно указать альтернативный адрес электронной почты на случай возможных технических проблем. В качестве альтернативного рекомендуется указывать почтовый ящик, который проверяется во время отпуска или командировки. Если у статьи несколько авторов, желательно указать адреса электронной почты двух или трех авторов, которые регулярно проверяют поступающие сообщения.

5. Аннотация статьи (Abstract). Аннотация не должна быть слишком краткой и не должна содержать ссылок на другие работы. Аннотация статьи должна быть информативной и подробной, описывать методы и главные результаты исследования. Из аннотации должно быть ясно, какие вопросы поставлены для исследования и какие ответы на них получены.

6. Желательно, но не обязательно, подготовить также аннотацию на английском. Это поможет повысить качество перевода статьи.

7. Собственно рукопись (основной текст). При подготовке рукописи следует соблюдать единообразие терминов. Не стоит называть одно и то же разными именами. Следует соблюдать единообразие в обозначениях, системах единиц измерения, номенклатуре. Следует по мере возможности избегать сокращений, кроме общеупотребительных. Если все-таки используются сокращения, то они должны быть расшифрованы в тексте при их первом упоминании.

8. Список литературы. Список литературы должен в достаточной мере отражать современное состояние дел в исследуемой области и не быть избыточным. Он должен содержать ссылки на доступные источники.

9. При наличии иллюстраций или таблиц располагать их следует в конце статьи на отдельных листах. К каждой иллюстрации должна быть указана подписанная подпись. При наличии нескольких частей в одной иллюстрации они должны располагаться последовательно и иметь общую подпись.

3. Требования к иллюстрациям

1. При сдаче рукописи все иллюстрации с подписанными подписями должны быть размещены в конце файла статьи. При наличии нескольких частей одной иллюстрации они должны располагаться последовательно и иметь общую подпись. При этом с рукописью обязательно должны быть сданы исходные файлы иллюстраций.

2. Исходные файлы иллюстраций принимают в след. форматах — jpg, tiff, gif, eps.

3. При подготовке иллюстраций необходимо учитывать, что у журнале они будут опубликованы в черно-белом варианте.

4. Каждый файл должен содержать одну иллюстрацию. Если иллюстрация состоит из нескольких частей, все они должны быть сгруппированы в один файл с правильным расположением частей.

5. Иллюстрации должны иметь размеры, соответствующие их информативности: 8–8.5 см (на

одну колонку), либо 17–17.5 см (на две колонки), и иметь разрешение не ниже 300 dpi.

4. Правила оформления библиографических ссылок

Особое внимание следует уделять правильному оформлению списка используемой литературы и соответствующих ссылок на источники внутри статьи. Список литературы должен строиться из библиографических ссылок на используемые публикации, список нумеруется в последовательности использования ссылки на соответствующий источник в тексте статьи (т.е. первой в этом списке должна содержаться библиографическая ссылка, первой упоминаемая в тексте статьи). Для ссылки на источник в тексте используется номер соответствующей библиографической ссылки в квадратных скобках.

Библиографические ссылки должны включать следующие данные:

- для книги — фамилии и инициалы всех авторов; полное название книги; наименование издательства и город, в котором оно находится; год издания; количество страниц книги, например: *Базыкин А.Д. Математическая биофизика взаимодействующих популяций. М.: Наука, 1985. 181 с.* *Berryman A.A. Population systems: a general introduction. N.Y.: Plenum Press, 1981. 222 p.*
- для статей — фамилии и инициалы всех авторов; полное название статьи; название журнала, газеты или сборника, в котором (которой) опубликована статья; год издания, идентификатор времени публикации (для газеты-номер выпуска или дата выхода, для журнала-год, том или номер выпуска, серия), номера страниц, занятых статьей (начальная и конечная), например:

Абрамов С. А., Хмельнов Д. Е. Линейные дифференциальные и разностные системы // Программирование. 2013. № 2. с. 51–74.

Schaffer W. M. Order and chaos in ecological systems // Ecology, 1985, v. 11, № 1, pp. 93–106.

Мандрыкин М. У., Мутилин В. С., Хорошилов А. В. Введение в метод CEGAR уточнение абстракции по контрпримерам // Труды Института системного программирования РАН. 2013. т. 24. с. 219–292.

Lawall J. L., Brunel J., Palix N., Rydhof H. R., Stuart H., Muller G. WYSIWIB: A declarative approach to finding API protocols and bugs in Linux code // Proceedings of the 39th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). 2009, pp. 43–52.

- для стандартов — название стандарта, номер стандарта, место и год издания, страницы;

- для патентных документов — название изобретения; номер патента; страна, номер и дата заявки на изобретение, дата опубликования патента; номер бюллетеня изобретений, страницы;

- для депонированных научных работ — фамилии и инициалы всех авторов; полное название работы; название депонирующего информационного центра; номер и дата депонирования; количество страниц работы;

- для диссертаций — фамилии и инициалы автора, полное название диссертации; на соискание какой ученой степени представлена диссертация; место и год защиты диссертации; количество страниц диссертации;

- для электронных ресурсов в Интернет — фамилии и инициалы всех авторов (если они известны), полное название материала, полный электронный адрес, дата публикации или создания, дата обращения (если невозможно установить дату публикации или создания), например: *http://www.ispras.ru/proceedings/authors.php Froese R., Pauly D., (eds.), 2006. FishBase. http://www.fish-base.org.*

Названия книг, статей, иных материалов и документов, опубликованных на иностранном языке, а также фамилии их авторов должны быть приведены в оригинальной транскрипции. В список используемой литературы не должны включаться неопубликованные материалы или материалы, не находящиеся в общественном доступе.

5. Этические нормы публикаций

Редколлегия журнала «Программирование» стремится поддерживать высокие этические стандарты, относящиеся к публикации статей в научных журналах. Плагиат недопустим. При выявлении в статье плагиата ее публикация не допускается. Рукописи или их части, представленные к публикации, не должны находиться на рассмотрении в других изданиях. Рукопись не должна быть ранее опубликована в любом другом журнале или других изданиях. Авторы должны следить за тем, чтобы материал, содержащийся в рукописи, не нарушал никакие авторские права и права третьих лиц. Наше положение об этике публикаций основано на COPE's Best Practice Guidelines for Journal Editors.

Обязанности редакционной коллегии журнала

- **Решение о публикации.** После процедуры рецензирования редколлегия журнала принимает решение о том, будет ли рукопись опубликована, и извещает о нем автора.

- **Непредвзятость.** Редколлегия всегда оценивает рукописи исключительно по их содержанию

без учета расовой принадлежности, пола, сексуальной ориентации, религиозных убеждений, этнического происхождения, гражданства или политических убеждений автора.

- **Конфиденциальность.** Члены редколлегии не должны разглашать информацию о присланной рукописи никому, кроме автора, потенциальных рецензентов, редакционных консультантов и издателя, в зависимости от обстоятельств.

- **Раскрытие информации и конфликт интересов.** Информация, содержащаяся в неопубликованных рукописях, не должна быть использована в работах членов редколлегии без письменного согласия автора.

Обязанности авторов

- **Соблюдение правил подготовки статьи.** Статья должна быть подготовлена в соответствии с правилами, принятыми в редакции (см. раздел «Правила для авторов»).

- **Оригинальность и плагиат.** Авторы обязуются гарантировать оригинальность собственных работ и, в случае если автор приводит чужие утверждения или результаты исследований, на эти работы приведены ссылки.

- **Множественные или параллельные публикации.** Автор не должен публиковать результат работы более чем в одном журнале. Одновременная подача рукописи в два и более журнала рассматривается как неэтичное поведение и является неприемлемой.

- **Обязательность ссылок.** Все заимствованные материалы в рукописи должны содержать ссылки на авторов. Авторы обязуются приводить ссылки на исследования, которые определили направление и характер их работы.

- **Авторство статей.** Список авторов должен быть ограничен теми, кто внес существенный вклад в концепцию, реализацию и содержимое работы. Участники, внесшие меньший вклад, должны быть указаны в качестве соавторов. Если существуют лица, внесшие существенный вклад в отдельные этапы работы, они должны быть упомянуты в качестве лиц, внесших вклад. Представитель авторского коллектива обязуется продемонстрировать финальную версию статьи всем участникам её создания и получить их согласие с оценкой их вклада в работу.

Обязанности рецензентов

- **Конфиденциальность.** Любые рукописи, полученные для рассмотрения, должны рассматриваться как конфиденциальные документы. Они

должны демонстрироваться или обсуждаться с посторонними только с разрешения редактора.

- **Оперативность.** Любой рецензент, который чувствует недостаточность собственной квалификации для рецензирования исследования, представленного в рукописи, или знает, что не сможет в краткие сроки написать рецензию, должен уведо-

мить об этом редактора и отказаться от рецензирования.

- **Стандарты объективности.** Рецензирование должно быть объективным. Не допускается персональная критика автора. Рецензенты должны четко выражать свое мнение и приводить аргументацию.

АВТОРСКИЙ УКАЗАТЕЛЬ СТАТЕЙ, ОПУБЛИКОВАННЫХ В 2020 ГОДУ

DOI: 10.31857/S0132347421100022

- Абламейко С. см. Чен Х.*
- Абрамов С.А., Боголюбская А.А.* Семинар по компьютерной алгебре в 2018–2019 гг. № 2, 3–5.
- Александров И.В. см. Савченко В.В.*
- Анохин Д.А. см. Кий К.И.*
- Ануреев И.С., Гаранина Н.О., Лях Т.В., Розов А.С., Зюбин В.Е., Горлач С.П.* Дедуктивная верификация Reflex-программ. № 4, 14–27.
- Айдинян А.Р. см. Чуйкова Е.Н.*
- Баар Т.* Некоторые недостатки входного синтаксиса KeYmaera. № 4, 37–46.
- Батхин А.Б.* Бифуркации периодических решений системы Гамильтона с дискретной группой симметрий. № 2, 14–28.
- Бобков В.А., Кудряшов А.П.* Идентификация и расчет траекторий динамических объектов по стереоизображениям. № 1, 3–14.
- Богданов Н.Н. см. Сорокин М.И.*
- Боголюбская А.А. см. Абрамов С.А.*
- Богущ Р. см. Чен Х.*
- Боженкова Е.Н., Вирбицкайте И.Б.* Тестовые эквивалентности временных сетей Петри. № 4, 3–13.
- Бондарев А.Е., Галактионов В.А., Кувшинников А.Е.* Параллельные решения параметрических задач газовой динамики с помощью технологии DVM/DVMH. № 3, 16–23.
- Боровикова О.И. см. Загоруйко Ю.А.*
- Бронштейн И.Е. см. Савченко В.В.*
- Брюно А.Д.* Нормализация периодической системы Гамильтона. № 2, 6–13.
- Васенин В.А., Кривчиков М.А.* Промежуточное представление программ для описания типов в терминах сопоставления значений с образцом. № 1, 63–72.
- Вирбицкайте И.Б. см. Боженкова Е.Н.*
- Волков А.С. см. Савченко В.В.*
- Вьюкова Н.И., Галатенко В.А., Павлов А.Н., Самборский С.В.* Отображение параллельных вычислений на распределенные системы, использующие технологию RapidIO. № 6, 55–66.
- Вьюкова Н.И., Галатенко В.А., Самборский С.В.* Средства динамического анализа программ в компиляторах GCC и CLANG. № 4, 47–65.
- Галактионов В.А. см. Бондарев А.Е.*
- Галактионов В.А. см. Санжаров В.В.*
- Галатенко В.А. см. Вьюкова Н.И.*
- Гаранина Н.О. см. Ануреев И.С.*
- Геворкян М.Н., Королькова А.В., Кулябов Д.С., Севастьянов Л.А.* Пример модульного расширения системы компьютерной алгебры. № 2, 29–35.
- Гергет О.М. см. Данилов В.В.*
- Гонахчан В.И. см. Семенов В.А.*
- Горлач С.П. см. Ануреев И.С.*
- Данилов В.В., Гергет О.М., Скурневский И.П., Манаков Р.А., Колтащиков Д.Ю.* Сегментация на основе распространения динамически изменяемых суперпикселей. № 3, 3–15.
- Дзюбенко А.Л. см. Лукин В.Н.*
- Е Ш. см. Чен Х.*
- Ермаков М.К. см. Савченко В.В.*
- Жданов А.Д. см. Сорокин М.И.*
- Жданов Д.Д. см. Сорокин М.И.*
- Загоруйко Ю.А., Боровикова О.И.* Использование системы разнородных паттернов онтологического проектирования для разработки онтологий научных предметных областей. № 4, 28–36.
- Замараев Р.Ю. см. Попов С.Е.*
- Золотов В.А. см. Семенов В.А.*
- Зюбин В.Е. см. Ануреев И.С.*
- Карпов Л.Е. см. Карпов Ю.Л.*
- Карпов Ю.Л., Карпов Л.Е., Сметанин Ю.Г.* Некоторые аспекты построения ассоциативной памяти на нейронной сети Хопфилда. № 5, 3–10.
- Касимов Д.Р. см. Кучуганов В.Н.*
- Качанов В.В. см. Савченко В.В.*

Кий К.И., Анохин Д.А., Подопросветов А.В. Программная система обработки изображений с параллельными вычислениями. № 6, 41–54.

Колпащиков Д.Ю. см. *Данилов В.В.*

Королькова А.В. см. *Геворкян М.Н.*

Костенко В.А., Чупахин А.А. Схемы организации “живой” миграции в центрах обработки данных. № 5, 11–15.

Кочегурова Е.А., Мартынова Ю.А. Особенности непрерывной идентификации пользователей на основе свободных текстов в режиме скрытого мониторинга. № 1, 15–28.

Краснов М.М. Библиотека функционального программирования для языка C++. № 5, 45–56.

Кривчиков М.А. см. *Васенин В.А.*

Крылов А.С. см. *Люков Д.А.*

Кувшинников А.Е. см. *Бондарев А.Е.*

Кудряшов А.П. см. *Бобков В.А.*

Кузоватов В.И., Кытманов А.А., Кузоватова О.И. Алгоритм построения аналога формулы Бине. № 2, 36–40.

Кузоватова О.И. см. *Кузоватов В.И.*

Кулябов Д.С. см. *Геворкян М.Н.*

Курилович А. см. *Чен Х.*

Кучуганов А.В. см. *Кучуганов В.Н.*

Кучуганов В.Н., Кучуганов А.В., Касимов Д.Р. Алгоритм кластеризации множества деталей по чертежам. № 1, 29–38.

Кытманов А.А. см. *Кузоватов В.И.*

Лукин В.Н., Дзюбенко А.Л., Чечиков Ю.Б. Подходы к разработке пользовательского интерфейса. № 5, 16–24.

Лукишин В.А. см. *Люков Д.А.*

Люков Д.А., Крылов А.С., Лукишин В.А., Усачев Д.Ю. Проекционный метод анализа перфузионных изображений мозга. № 3, 35–41.

Лях Т.В. см. *Ануреев И.С.*

Макеев С.Д. см. *Марченков С.С.*

Манакон Р.А. см. *Данилов В.В.*

Марков С.И. см. *Савченко В.В.*

Мартынова Ю.А. см. *Кочегурова Е.А.*

Марченков С.С., Макеев С.Д. Моделирование молекул машин Минского и Тьюринга трехленточными машинами Минского. № 6, 67–72.

Мешвелиани С.Д. О машинном доказательстве для арифметики дробей над кольцом с НОД. № 2, 41–50.

Михайлюк М.В. см. *Тимохин П.Ю.*

Морозов А.В. см. *Раздьяконов Д.С.*

Морозов С.В. см. *Семенов В.А.*

Муромцев Д.И. см. *Раздьяконов Д.С.*

Насибуллин А.Р., Новиков Б.А. Репликация в распределенных системах: модели, методы и протоколы. № 5, 57–72.

Новиков Б.А. см. *Насибуллин А.Р.*

Павлов А.Н. см. *Вьюкова Н.И.*

Павлов Д.С. см. *Раздьяконов Д.С.*

Панкратенко Г.А. см. *Савченко В.В.*

Подопросветов А.В. см. *Кий К.И.*

Попов С.Е., Замараев Р.Ю. Быстрый алгоритм классификации сейсмических событий на базе распределенных вычислений Apache Spark. № 1, 39–53.

Потемин И.С. см. *Сорокин М.И.*

Прокопья А.Н. Построение периодического решения уравнений движения обобщенной машины Атвуда с применением компьютерной алгебры. № 2, 51–56.

Раздьяконов Д.С., Морозов А.В., Павлов Д.С., Муромцев Д.И. Подход к обработке пустых узлов при порционной визуализации данных на примере инструмента Ontodia. № 6, 16–29.

Розов А.С. см. *Ануреев И.С.*

Савченко В.В., Сорокин К.С., Бронштейн И.Е., Волков А.С., Качанов В.В., Панкратенко Г.А., Ермаков М.К., Марков С.И., Спиридонов А.В., Александров И.В. Nobrainer: инструмент преобразования C/C++ кода на основе примеров. № 5, 32–44.

Сальников В.Н., Хамдуни А. Дифференциальная геометрия и механика — источник задач для компьютерной алгебры. № 2, 57–63.

Самборский С.В. см. *Вьюкова Н.И.*

Санжаров В.В., Фролов В.А., Галактионов В.А. Исследование Технологии Nvidia RTX. № 4, 66–72.

Севастьянов Л.А. см. *Геворкян М.Н.*

Семенов В.А., Шуткин В.Н., Золотов В.А., Морозов С.В., Гонахьян В.И. Визуализация больших сцен с детерминированной динамикой. № 3, 42–52.

Скирневский И.П. см. *Данилов В.В.*

Сляднев С.Е., Турлапов В.Е. Упрощение САД-моделей путем автоматического распознавания и подавления цепочек скруглений. № 3, 53–64.

Сметанин Ю.Г. см. *Карпов Ю.Л.*

Сорокин К.С. см. *Савченко В.В.*

Сорокин М.И., Жданов Д.Д., Жданов А.Д., Потемин И.С., Богданов Н.Н. Восстановление параметров освещения в системах смешанной реальности с помощью технологии сверточных нейронных сетей по RGBD-изображениям. № 3, 24–34.

Спиридонов А.В. см. *Савченко В.В.*

Степанян И.В. Методология и инструментальные средства проектирования бинарных нейронных сетей. № 1, 54–62.

Тимохин П.Ю., Михайлюк М.В. Метод извлечения поверхностей уровня на GPU с помощью программируемой тесселяции. № 3, 65–72.

Турлапов В.Е. см. *Сляднев С.Е.*

Усачев Д.Ю. см. *Люков Д.А.*

Фролов В.А. см. *Санжаров В.В.*

Хамдуни А. см. *Сальников В.Н.*

Цветкова О.Л. см. *Чуйкова Е.Н.*

Чен Х., Е Ш., Курилович А., Богуш Р., Абламейко С. Распознавание реквизитов банковских карт в мо-

бильных устройствах по видеопоследовательностям. № 6, 3–15.

Чечиков Ю.Б. см. *Лукин В.Н.*

Чупахин А.А. см. *Костенко В.А.*

Чуйкова Е.Н., Айдинян А.Р., Цветкова О.Л. Алгоритм адаптивного изменения меню программного приложения. № 6, 30–40.

Шуткин В.Н. см. *Семенов В.А.*

Щетинин Е.Ю. Разработка компьютерных технологий энергосбережения умных зданий с применением компьютерной алгебры. № 5, 25–31.

Янович Д.А. Вычисление инволютивных базисов и базисов Грёбнера используя табличное представление полиномов. № 2, 64–72.