

ISSN 0132-3474

Номер 2

Март - Апрель 2023



ПРОГРАММИРОВАНИЕ

www.sciencejournals.ru



СОДЕРЖАНИЕ

Номер 2, 2023

КОМПЬЮТЕРНАЯ АЛГЕБРА

Семинар по компьютерной алгебре в 2021–2022 г. <i>С. А. Абрамов, А. А. Боголюбская</i>	3
Решение задачи Коши для трехмерного разностного уравнения в параллелепипеде <i>М. С. Апанович, А. П. Ляпин, К. В. Шадрин</i>	5
О вычислении линий уровня многочлена на плоскости <i>А. Д. Брюно, А. Б. Батхин, З. Х. Хайдаров</i>	13
Символьный алгоритм для задачи нахождения нулей системы голоморфных функций <i>В. И. Кузоватов, А. А. Кытманов, Е. К. Мышкина</i>	31
Вычисление связанных компонент дополнения к амбеде многочлена от нескольких комплексных переменных <i>Т. А. Жуков, Т. М. Садыков</i>	36
Символьно-численная реализации метода Галеркина для приближенного решения задачи волноводной дифракции <i>Д. В. Диваков, А. А. Тютюнник</i>	46
Разработка алгоритмического и программного обеспечения моделирования управляемых динамических систем с применением символьных вычислений и стохастических методов <i>А. В. Демидова, О. В. Дружинина, О. Н. Масина, А. А. Петров</i>	54
Компьютерно-алгебраические вычисления в суперсимметричной электродинамике <i>И. Е. Широков</i>	69

CONTENTS

No. 2, 2023

COMPUTER ALGEBRA

Seminar on Computer Algebra in 2021–2022 <i>S. A. Abramov, A. A. Bogolyubskaya</i>	3
Solving the Cauchy Problem for a Three-dimensional Difference Equation in a Parallelepiped <i>M. S. Apanovich, A. P. Lyapin, and K. V. Shadrin</i>	5
Computing Level Lines of a Polynomial on the Plane <i>A. D. Bruno, A. B. Batkhin, and Z. Kh. Khaidarov</i>	13
Symbolic Algorithm for Finding Zeros of a System of Holomorphic Functions <i>V. I. Kuzovatov, A. A. Kytmanov, and E. K. Myshkina</i>	31
Computing the Connected Components of the Complement to the Amoeba of a Polynomial in Several Complex Variables <i>T. A. Zhukov and T. M. Sadykov</i>	36
Symbolic-Numerical Implementation of the Galerkin Method for Approximate Solution of the Waveguide Diffraction Problem <i>D. V. Divakov and A. A. Tyutyunnik</i>	46
Development of Algorithms and Software for Modeling Controlled Dynamic Systems Using Symbolic Computations and Stochastic Methods <i>A. V. Demidova, O. V. Druzhinina, O. N. Masina, and A. A. Petrov</i>	54
Computer Algebra Calculations in Supersymmetric Electrodynamics <i>I. E. Shirokov</i>	69

СЕМИНАР ПО КОМПЬЮТЕРНОЙ АЛГЕБРЕ В 2021–2022 г.

© 2023 г. С. А. Абрамов^{a,*}, А. А. Боголюбская^{b,*}

^aВЦ ФИЦ ИУ РАН 119333 Москва, ул. Вавилова, 40, Россия

^bОбъединенный институт ядерных исследований 141980 Дубна Московской области, Россия

*E-mail: sergeyabramov@mail.ru

**E-mail: abogol@jinr.ru

Поступила в редакцию 22.08.2022 г.

После доработки 20.10.2022 г.

Принята к публикации 20.10.2022 г.

Годовой отчет о работе научно-исследовательского семинара по компьютерной алгебре

DOI: 10.31857/S0132347423020024, EDN: GYJUPJ

1. О СЕМИНАРЕ

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

В 2021–2022 учебном году семинар собирался раз в месяц по третьим средам (онлайн).

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](http://www.ccas.ru/sabramov/seminar/doku.php), где также содержится информация о состоявшихся ранее докладах.

А.Н. Прокопеня (Варшавский университет естественных наук — SGGW, Варшава, Польша; alexander_prokopenya@sggw.edu.pl), М.Дж. Минглибаев, А.Б. Кошербаева (Казахский национальный университет им. аль-Фараби, Алматы, Казахстан; minglibayev@gmail.com; kosherbaevaayken@gmail.com)

Символьные вычисления в исследованиях эволюционных уравнений в задаче многих тел с переменными массами.

М. Баркату, Т. Клюзо, А. Эль Хадж (Лиможский университет; CNRS; XLIM UMR 7252; MATHIS; moulay.barkatou@unilim.fr, thomas.cluzeau@unilim.fr, ali.el-hajj@unilim.fr)

Символьные алгоритмы исследования и решения псевдolineйных систем.

Ю.А. Блинков (Научный центр вычислительных методов в прикладной математике института

прикладной математики и телекоммуникаций РУДН, Москва; BlinkovUA@info.sgu.ru)

Объектно-ориентированное динамическое перераспределение памяти в C++.

Б. Тегия Табугия (Университет Касселя, Германия; bteguia@mathematik.uni-kassel.de)

Ряды и последовательности, определяемые квадратными дифференциальными уравнениями.

В.Г. Захаров (Институт механики сплошных сред УрО РАН, Пермь; victor@icmm.ru)

Матричный метод нахождения полиномиальных решений системы линейных дифференциальных уравнений в частных производных.

С.Д. Мешвелиани (Институт программных систем им. А.К. Айламазяна РАН, Переславль-Залесский; mechvel@botik.ru)

О конструктивных доказательствах завершаемости алгоритма нормальной формы многочлена нескольких переменных относительно набора многочленов.

М.Н. Геворкян (Российский университет дружбы народов, Москва; gevorgyan_mn@rudn.university)

Описание аппарата геометрической алгебры и его реализации в системе компьютерной алгебры SymPy.

А. Хименес-Пастор (Эколь Политекник, Париж, Франция; jimenezpastor@lix.polytechnique.fr), М. Петковшек (Люблянский Университет, факультет математики и физики, Любляна, Словения; Marko.Petkovsek@fmf.uni-lj.si)

Линейные рекуррентные уравнения с полиномиальными коэффициентами: метод факториального базиса для нахождения решений, имеющих вид определенных сумм.

¹ Перечень докладов, прочитанных в 1995–2021 гг., опубликован в [1]– [27].

3. Ли (Главная лаборатория математики и механизации, AMSS Китайская академия наук, Пекин; zmli@mmrc.iss.ac.cn)

Вычисление элементарных интегралов с помощью аддитивных разложений и гомоморфного оценивания.

Ш. Чен (Академия математики и системных наук, Китайская академия наук, Пекин; schen@amss.ac.cn)

Задачи устойчивости в символьном интегрировании.

Доклад посвящен Зимингу Ли в честь его шестидесятилетия.

СПИСОК ЛИТЕРАТУРЫ

1. Абрамов С.А., Зима Е.В. Семинар по компьютерной алгебре на факультете вычислительной математики и кибернетики МГУ в 1995–1996 г. // Программирование. 1997. № 1. С. 75–77.
2. Абрамов С.А., Зима Е.В. Научно-исследовательский семинар “Компьютерная алгебра” в 1996–1997 г. // Программирование. 1998. № 1. С. 69–72.
3. Абрамов С.А., Ростовцев В.А. Семинар по компьютерной алгебре в 1997–1998 г. // Программирование. 1998. № 6. С. 3–7.
4. Абрамов С.А., Крюков А.П., Ростовцев В.А. Семинар по компьютерной алгебре в 1998–1999 г. // Программирование. 2000. № 1. С. 8–12.
5. Абрамов С.А., Крюков А.П., Ростовцев В.А. Семинар по компьютерной алгебре в 1999–2000 г. // 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 г. // Программирование. 2003. № 2. С. 3–7.
8. Абрамов С.А., Еднерал В.Ф., Ростовцев В.А. Семинар по компьютерной алгебре в 2002–2003 г. // 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 г. // 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 г. // 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 г. // 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 г. // 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 г. // 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 г. // 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 г. // Программирование. 2011. № 1. С. 3–8.
17. Абрамов С.А., Боголюбская А.А., Ростовцев В.А. Семинар по компьютерной алгебре в 2010–2011 г. // Программирование. 2012. № 2. С. 3–8.
18. Абрамов С.А., Боголюбская А.А., Ростовцев В.А. Семинар по компьютерной алгебре в 2011–2012 г. // Программирование. 2013. № 2. С. 3–10.
19. Абрамов С.А., Боголюбская А.А., Ростовцев В.А. Семинар по компьютерной алгебре в 2012–2013 г. // Программирование. 2014. № 2. С. 3–11.
20. Абрамов С.А., Боголюбская А.А., Ростовцев В.А. Семинар по компьютерной алгебре в 2013–2014 г. // Программирование. 2015. № 2. С. 3–6.
21. Абрамов С.А., Боголюбская А.А., Ростовцев В.А. Семинар по компьютерной алгебре в 2014–2015 г. // Программирование. 2016. № 2. С. 4–7.
22. Абрамов С.А., Боголюбская А.А. Семинар по компьютерной алгебре в 2015–2016 г. // Программирование. 2017. № 2. С. 3–6.
23. Абрамов С.А., Боголюбская А.А. Семинар по компьютерной алгебре в 2016–2017 г. // Программирование. 2018. № 2. С. 3–4.
24. Абрамов С.А., Боголюбская А.А. Семинар по компьютерной алгебре в 2017–2018 г. // Программирование. 2019. № 2. С. 3–5.
25. Абрамов С.А., Боголюбская А.А. Семинар по компьютерной алгебре в 2018–2019 г. // Программирование. 2020. № 2. С. 3–5.
26. Абрамов С.А., Боголюбская А.А. Семинар по компьютерной алгебре в 2019–2020 г. // Программирование. 2021. № 2. С. 3–4.
27. Абрамов С.А., Боголюбская А.А. Семинар по компьютерной алгебре в 2020–2021 г. // Программирование. 2022. № 2. С. 3–6.

УДК 519.85

РЕШЕНИЕ ЗАДАЧИ КОШИ ДЛЯ ТРЕХМЕРНОГО РАЗНОСТНОГО УРАВНЕНИЯ В ПАРАЛЛЕЛЕПИПЕДЕ

© 2023 г. М. С. Апанович^{а,*}, А. П. Ляпин^{б,с,**}, К. В. Шадрин^{а,***}^аКрасноярский государственный медицинский университет имени профессора В.Ф. Войно-Ясенецкого
660022 Красноярск, ул. Партизана Железняка, 1, Россия^бСибирский федеральный университет
660041 Красноярск, пр. Свободный, 79, Россия^сФэрмонтский государственный университет
660041 Фэрмонт, Западная Вирджиния, США, Локуст стрит, 1201

*E-mail: marina.apanovich@list.ru

**E-mail: aplyapin@sfu-kras.ru

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

Поступила в редакцию 17.07.2022 г.

После доработки 10.08.2022 г.

Принята к публикации 21.10.2022 г.

Представлен алгоритм вычисления решения задачи Коши для трехмерного разностного уравнения с постоянными коэффициентами в точке по коэффициентам разностного уравнения и начальным данным Коши, заданным в параллелепипеде. Наш алгоритм использует теоремы Апанович М.С. и Лейнартаса Е.К., характеризующие корректность задачи Коши, и методы компьютерной алгебры для достижения вычислительной эффективности. Алгоритм реализован в среде Matlab.

DOI: 10.31857/S0132347423020048, EDN: GYQZFW

1. ВВЕДЕНИЕ

Метод конечно-разностной дискретизации, примененный к дифференциальным и интегральным уравнениям, приводит к разностным уравнениям (см. [1, 2]). Комбинация линейных разностных уравнений и метода производящих функций эффективно применяется к многочисленным задачам перечислительного комбинаторного анализа (см., например, [3–5]). Для уравнений Коши–Римана это дает дискретную теорию аналитических функций (см. [6, 7]), используемую для изучения Римановых поверхностей (см., например, [8, 9]). Также разностные уравнения используются в динамических моделях с дискретным временем (см. [10, 11]).

Дополнительные условия (“начальные”, “граничные”, “данные Коши”) позволяют выбрать единственное решение из бесконечного множества решений, а возникающая при этом задача называется задачей Коши для многомерного разностного уравнения. Одномерная задача Коши проста, так как ее исходные данные конечны. Многомерный случай отличается тем, что исходные данные заданы на бесконечном множестве и имеют сложную структуру. Алгоритмы решения многомерного разностного уравнения с постоянными, полиномиальными или рациональными

коэффициентами функции изучались в [12–15]. Связь производящей функции решения линейной задачи Коши и производящей функции ее данных Коши исследована в работах [16–19]. Алгоритм вычисления решения двумерной задачи Коши в точке через коэффициенты разностного уравнения и данные Коши были разработаны и реализованы в [20, 21].

Отметим, что многомерные случаи должны образом не исследованы, однако они имеют большое значение, например, трехмерный случай имеет большое значение в задачах термодинамики (теплопроводность, см. [2]), анизотропная диффузия, см. [22]), математической биологии (распределение морфогена в крыльях насекомых [22], распределение плотности популяции видов [23]). В данной статье рассматривается решение трехмерной задачи Коши для линейного разностного уравнения с постоянными коэффициентами в параллелепипеде в точке через коэффициенты линейного разностного уравнения и данные Коши. Мы определяем задачу Коши в параллелепипеде и описываем функцию начальных данных таким образом, который позволяет нам разработать и реализовать компьютерный алгоритм.

2. ЗАДАЧА КОШИ ДЛЯ ПОЛИНОМИАЛЬНОГО РАЗНОСТНОГО ОПЕРАТОРА В “ПАРАЛЛЕЛЕПИПЕДЕ”

Пусть функция f целочисленных аргументов $f: \mathbb{Z}_+^n \rightarrow \mathbb{C}$ и δ_j – оператор сдвига по j -й переменной, т.е. $\delta_j f(t_1, \dots, t_n) = f(t_1, \dots, t_j - 1, t_j + 1, t_{j+1}, \dots, t_n)$, $j = 1, 2, \dots, n$. Некоторые полезные свойства оператора сдвига были исследованы в [24]. Если $\alpha = (\alpha_1, \dots, \alpha_n)$ – мультииндекс, то $|\alpha| = \alpha_1 + \dots + \alpha_n$, $\delta^\alpha = \delta_1^{\alpha_1} \dots \delta_n^{\alpha_n}$. Для двух мультииндексов $\alpha = (\alpha_1, \dots, \alpha_n)$, $\beta = (\beta_1, \dots, \beta_n)$ неравенство $\alpha \leq \beta$ означает, что $\alpha_j \leq \beta_j$, $j = 1, 2, \dots, n$.

Рассмотрим разностный полиномиальный оператор вида

$$P(\delta) = \sum_{0 \leq \alpha \leq m} c_\alpha \delta^\alpha, \quad (1)$$

где c_α – постоянные коэффициенты оператора $P(\delta)$. Если $\alpha = (\alpha_1, \dots, \alpha_n)$, то будем обозначать $'\alpha = (\alpha_1, \dots, \alpha_{n-1})$ и тогда оператор $P(\delta)$ можно записать в виде

$$P(\delta) = \sum_{\alpha_n=0}^{m_n} \left(\sum_{0 \leq \alpha' \leq m'} c_{\alpha, \alpha_n} \delta'^{\alpha'} \right) \delta_n^{\alpha_n}.$$

Многочлен

$$\begin{aligned} P(z, z_n) &= \sum_{\alpha_n=0}^{m_n} \left(\sum_{0 \leq \alpha' \leq m'} c_{\alpha, \alpha_n} z'^{\alpha'} \right) z_n^{\alpha_n} = \\ &= \sum_{\alpha_n=0}^{m_n} P_j(z) z_n^{\alpha_n} \end{aligned}$$

будем называть характеристическим многочленом для разностного оператора $P(\delta)$, а его степень по переменной z_n – порядком разностного оператора.

В целочисленной решетке $\mathbb{Z}_+^n$ выберем точку $x = (x_1, \dots, x_{n-1}, 0)$ и обозначим $\Pi_x = \{t \in \mathbb{R}_+^n : 0 \leq t \leq x\}$ $(n-1)$ -мерный параллелепипед в гиперплоскости $t_n = 0$.

Зафиксируем точку $\beta = (\beta_1, \dots, \beta_{n-1}, m_n) \in \mathbb{Z}_+^n$, $0 \leq \beta \leq m$ такую, что коэффициент $c_\beta \neq 0$, и рассмотрим множество $\Pi_{\beta, m} = \{t \in \Pi_x : \beta \leq t \leq x - m + \beta\}$. Обозначим $L = (\Pi_x \setminus \Pi_{\beta, m}) \times [0, x_n]$ множество, на котором будем задавать начальные данные и сформулируем задачу:

найти решение разностного уравнения

$$P(\delta) f(x) = g(x), \quad x \in \Pi = \Pi_x \times [0, x_n], \quad (2)$$

удовлетворяющее условию

$$f(x) = \varphi(x), \quad x \in L, \quad (3)$$

где $g(x)$ и $\varphi(x)$ – заданные функции целочисленных аргументов.

Однородная двухслойная линейная разностная схема с постоянными коэффициентами рассматривалась в [25], где исследована ее устойчивость при $m_n = 1$. Разрешимость двумерной задачи Коши (2), (3) исследована в [26]. Такие задачи известны как многослойные неявные разностные схемы. В [27] исследована корректность задачи (2), (3) и доказано достаточное условие корректности, а достаточное условие разрешимости трехмерной задачи (2), (3) доказано в [28].

Рассмотрим задачу Коши (2), (3) для $n = 3$. $\mathbb{Z}^3 = \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ – целочисленная решетка, а $\mathbb{Z}_+^3$ – подмножество этой решетки, состоящее из точек с целыми неотрицательными координатами. Пусть $\delta_1, \delta_2, \delta_3$ – операторы сдвига по переменной x, y, z соответственно, т.е. $\delta_1 f(x, y, z) = f(x + 1, y, z)$, $\delta_2 f(x, y, z) = f(x, y + 1, z)$, $\delta_3 f(x, y, z) = f(x, y, z + 1)$. Зададим параллелепипед $\Pi = \{(x, y, z) \in \mathbb{Z}_+^3, 0 \leq x \leq B_x, 0 \leq y \leq B_y, z > 0\}$ в положительном октанте целочисленной решетки, число $B_x + 1$ будем называть шириной Π , а $B_y + 1$ – длиной Π . Разностный полиномиальный оператор (1) будет иметь вид

$$\begin{aligned} P(\delta_1, \delta_2, \delta_3) &= \sum_{j=0}^m \sum_{i_y=0}^{b_y} \sum_{i_x=0}^{b_x} c_{i_x, i_y, j} \delta_1^{i_x} \delta_2^{i_y} \delta_3^j = \\ &= \sum_{j=0}^m P_j(\delta_1, \delta_2) \delta_3^j, \end{aligned} \quad (4)$$

где b_x, b_y и m определяют размер разностной схемы, а

$$P_j(\delta_1, \delta_2) = \sum_{i_x=0}^{b_x} \sum_{i_y=0}^{b_y} c_{i_x, i_y, j} \delta_1^{i_x} \delta_2^{i_y}, \quad j = 0, 1, \dots, m.$$

Характеристический многочлен будет иметь вид

$$P(s, w, v) = \sum_{j=0}^m \sum_{i_x=0}^{b_x} \sum_{i_y=0}^{b_y} c_{i_x, i_y, j} s^{i_x} w^{i_y} v^j,$$

где m – порядок разностного оператора $P(\delta_1, \delta_2, \delta_3)$, $b_x < B_x$, $b_y < B_y$.

Зафиксируем $\beta = (x_\beta, y_\beta, m)$ такое, что $c_{x_\beta, y_\beta, m} \neq 0$ и рассмотрим множество $\Pi_\beta = \{(x, y, z) \in \mathbb{Z}_+^3 : 0 \leq x - x_\beta \leq B_x - b_x, 0 \leq y - y_\beta \leq B_y - b_y, z > m - 1\}$, тогда $L = \Pi \setminus \Pi_\beta$ и задача Коши будет иметь вид:

найти решение разностного уравнения

$$P(\delta_1, \delta_2, \delta_3) f(x, y, z) = g(x, y, z), \quad (x, y, z) \in \Pi \quad (5)$$

удовлетворяющее условию

$$f(x, y, z) = \varphi(x, y, z), \quad (x, y, z) \in L, \quad (6)$$

где $g(x, y, z)$ и $\varphi(x, y, z)$ — заданные функции целочисленных аргументов.

Задачу (5), (6) назовем задачей Коши для полиномиального разностного оператора (4).

В [28] доказано, что задача (5), (6) однозначно разрешима, если выполняется условие

$$|c_{x_\beta, y_\beta, m}| > \sum_{(\alpha_1, \alpha_2) \neq (x_\beta, y_\beta), \alpha_3 = m} |c_{\alpha_1, \alpha_2, \alpha_3}|. \quad (7)$$

Поставим задачу вычислить значение функции $f(x, y, z)$ в точке A с координатами (x_1, y_1, z_1) .

3. ОПИСАНИЕ ВХОДНЫХ ДАННЫХ

Решение задачи Коши (5), (6) для трехмерного разностного уравнения с постоянными коэффициентами в точке A с координатами (x_1, y_1, z_1) представляет собой значение функции $f(x, y, z)$ в точке A . Алгоритм вычисления значения функции $f(x, y, z)$ в заданной точке с координатами (x_1, y_1, z_1) имеет рекурсивный характер и сводится к вычислению значений функции $f(x, y, z)$ на конечном подмножестве точек (x, y, z) из множества $L = \Pi \setminus \Pi_\beta$.

Начальные данные (6) задаются трехмерной матрицей F , содержащей конечное подмножество значений начальных данных задачи Коши. Коэффициенты трехмерного разностного уравнения задаются трехмерной матрицей C . Для технической реализации алгоритма матрицы коэффициентов C и начальных данных F задаются по

слоям, начиная с самого нижнего. Проиллюстрируем процедуру задания матриц F и C .

Для разностного уравнения

$$\begin{aligned} & c_{000}f(x, y, z) + c_{100}f(x+1, y, z) + \\ & + c_{200}f(x+2, y, z) + c_{010}f(x, y+1, z) + \\ & + c_{110}f(x+1, y+1, z) + c_{210}f(x+2, y+1, z) + \\ & + c_{001}f(x, y, z+1) + c_{101}f(x+1, y, z+1) + \\ & + c_{201}f(x+2, y, z+1) + c_{011}f(x, y+1, z+1) + \\ & + c_{111}f(x+1, y+1, z+1) + \\ & + c_{211}f(x+2, y+1, z+1) = 0 \end{aligned} \quad (8)$$

трехмерная матрица коэффициентов C на первом слое будет содержать матрицу C_0 вида

$$C_0 = \begin{pmatrix} c_{000} & c_{100} & c_{200} \\ c_{010} & c_{110} & c_{210} \end{pmatrix},$$

а на втором — матрицу C_1 вида

$$C_1 = \begin{pmatrix} c_{001} & c_{101} & c_{201} \\ c_{011} & c_{111} & c_{211} \end{pmatrix}.$$

Запишем трехмерную матрицу коэффициентов C следующим образом

$$C = (C_0 | C_1) = \begin{pmatrix} c_{000} & c_{100} & c_{200} & c_{001} & c_{101} & c_{201} \\ c_{010} & c_{110} & c_{210} & c_{011} & c_{111} & c_{211} \end{pmatrix}.$$

Для разностного уравнения (8) $b_x = 2$, $b_y = 1$, $m = 1$. Пусть $\beta = (1, 1, 1)$, $B_x = 4$, $B_y = 2$, тогда трехмерная матрица начальных данных F будет иметь вид $F = (F_0 | F_1)$, где

$$\begin{aligned} F_0 &= \begin{pmatrix} \varphi(0,0,0) & \varphi(1,0,0) & \varphi(2,0,0) & \varphi(3,0,0) & \varphi(4,0,0) \\ \varphi(0,1,0) & \varphi(1,1,0) & \varphi(2,1,0) & \varphi(3,1,0) & \varphi(4,1,0) \\ \varphi(0,2,0) & \varphi(1,2,0) & \varphi(2,2,0) & \varphi(3,2,0) & \varphi(4,2,0) \end{pmatrix}, \\ F_1 &= \begin{pmatrix} \varphi(0,0,1) & \varphi(1,0,1) & \varphi(2,0,1) & \varphi(3,0,1) & \varphi(4,0,1) \\ \varphi(0,1,1) & * & * & * & \varphi(4,1,1) \\ \varphi(0,2,1) & * & * & * & \varphi(4,2,1) \end{pmatrix}. \end{aligned}$$

Здесь элементы, обозначенные *, вычисляются при выполнении алгоритма. При этом, вычислить элемент $\varphi(1, 2, 1)$ не представляется возможным без вычисления элементов $\varphi(2, 2, 1)$, $\varphi(1, 1, 1)$, $\varphi(2, 1, 1)$, $\varphi(3, 1, 1)$, $\varphi(3, 2, 1)$. Таким образом, для нахождения неизвестных элементов необходимо решить систему линейных разностных уравнений вида (8) с использованием начальных данных $\varphi(i, j, k)$, где $i = 0, \dots, 4$, $j = 0, 1, 2$, $k = 0$, $\varphi(i, j, k)$,

$i = 0, \dots, 4$, $j = 0, k = 1$ и $\varphi(0, 1, 1)$, $\varphi(0, 2, 1)$, $\varphi(4, 1, 1)$, $\varphi(4, 2, 1)$.

Итак, входные данные конечны и имеют вид:

1. Трехмерная матрица коэффициентов $C = (c_{\alpha_1 \alpha_2 \alpha_3})$, $\alpha_1 = 0, \dots, b_x$, $\alpha_2 = 0, \dots, b_y$, $\alpha_3 = 0, \dots, m$ размера $(b_x + 1) \times (b_y + 1) \times (m + 1)$ из коэффициентов $c_{\alpha_1 \alpha_2 \alpha_3}$ трехмерного разностного уравнения;

2. Точка $\beta_0 = (x_\beta, y_\beta, m)$;

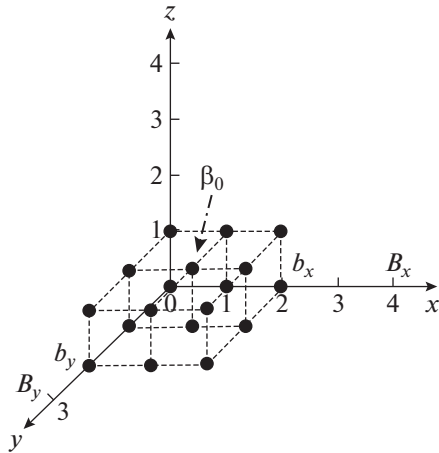


Рис. 1. Расположение элементов C в декартовой системе координат.

3. Точка A с координатами (x_1, y_1, z_1) , определяющая координаты искомого значения функции $f(x, y, z)$ и количество слоев в трехмерной матрице начальных данных F ;

4. Трехмерная матрица начальных данных $F = (\varphi(x, y, z))$, размера $(B_x + 1) \times (B_y + 1) \times (z_1 + 1)$, где $(x, y, z) \in L$, для всех остальных значений (x, y, z) значения $\varphi(x, y, z) = 0$.

Поскольку координаты элементов трехмерной матрицы коэффициентов разностного оператора и трехмерной матрицы начальных данных в декартовой системе координат (X, Y, Z) не совпадают с их координатами в матрице (строка $\times$ столбец $\times$ слой), то следует перейти из декартовой системы координат $(D(d_1, d_2, d_3))$ в “матричную” $(M(m_1, m_2, m_3))$ по правилу: $D(d_1, d_2, d_3) \rightarrow M(m_1, m_2, m_3)$, где $m_1 = d_2 + 1$, $m_2 = d_1 + 1$, $m_3 = d_3 + 1$.

Далее необходимо будет проверить задачу Коши (5), (6) на разрешимость, т.е. проверить, выполняется ли условие (7) для коэффициентов разностного оператора (4).

$$F = (F_0 | F_1 | F_2 | F_3 | F_4) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 3 & 4 & 5 & 7 & 6 & 3 & 4 & 5 & 7 & 6 & 3 & 4 & 5 & 7 & 6 & 3 & 4 & 5 & 7 & 6 \\ 4 & 5 & 6 & 1 & 3 & 6 & 0 & 0 & 0 & 8 & 5 & 0 & 0 & 0 & 2 & 6 & 0 & 0 & 0 & 1 & 3 & 0 & 0 & 0 & 8 \\ 1 & 2 & 3 & 8 & 4 & 3 & 0 & 0 & 0 & 4 & 3 & 0 & 0 & 0 & 4 & 3 & 0 & 0 & 0 & 4 & 3 & 0 & 0 & 0 & 4 \\ 1 & 2 & 3 & 4 & 6 & 6 & 5 & 3 & 2 & 3 & 6 & 8 & 1 & 4 & 3 & 6 & 6 & 7 & 1 & 4 & 6 & 2 & 6 & 7 & 3 \end{pmatrix}.$$

Расположение элементов матрицы начальных данных F в декартовой системе координат представлено на рис. 2.

4. ПРИМЕР

Будем рассматривать полиномиальный разностный оператор

$$\begin{aligned} P(\delta_1, \delta_2, \delta_3) = & c_{000} + c_{100}\delta_1 + c_{200}\delta_1^2 + c_{010}\delta_2 + \\ & + c_{110}\delta_1\delta_2 + c_{210}\delta_1^2\delta_2 + c_{020}\delta_2^2 + c_{120}\delta_1\delta_2^2 + \\ & + c_{220}\delta_1^2\delta_2^2 + c_{001}\delta_3 + c_{101}\delta_1\delta_3 + c_{201}\delta_1^2\delta_3 + \\ & + c_{011}\delta_2\delta_3 + c_{111}\delta_1\delta_2\delta_3 + c_{211}\delta_1^2\delta_2\delta_3 + c_{021}\delta_2^2\delta_3 + \\ & + c_{121}\delta_1\delta_2^2\delta_3 + c_{221}\delta_1^2\delta_2^2\delta_3 = 1 + 2\delta_1 + 3\delta_1^2 + 4\delta_2 + \\ & + 5\delta_1\delta_2 + 6\delta_1^2\delta_2 + 1\delta_2^2 + 2\delta_1\delta_2^2 + 3\delta_1^2\delta_2^2 + 3\delta_3 + \\ & + 4\delta_1\delta_3 + 5\delta_1^2\delta_3 + 6\delta_2\delta_3 + 8\delta_1\delta_2\delta_3 + 8\delta_1^2\delta_2\delta_3 + \\ & + 3\delta_2^2\delta_3 + 4\delta_1\delta_2^2\delta_3 + 5\delta_1^2\delta_2^2\delta_3, \end{aligned} \quad (9)$$

где $b_x = 2$, $b_y = 2$, $m = 1$. Фиксируем $\beta_0 = (x_\beta, y_\beta, m) = (1, 1, 1)$, $B_x = 4$, $B_y = 3$. Множество начальных данных будет иметь вид:

$$L = \Pi \setminus \Pi_{(1,1)},$$

где $\Pi = \{(x, y, z) \in \mathbb{Z}^3, 0 \leq x \leq 4, 0 \leq y \leq 3, z \geq 0\}$, $\Pi_{(1,1)} = \{(x, y, z) \in \mathbb{Z}_+^3 : 1 \leq x \leq 3, 1 \leq y \leq 2, z \geq 1\}$. Трехмерная матрица коэффициентов C полиномиального разностного оператора (9) задается по слоям, начиная с самого нижнего:

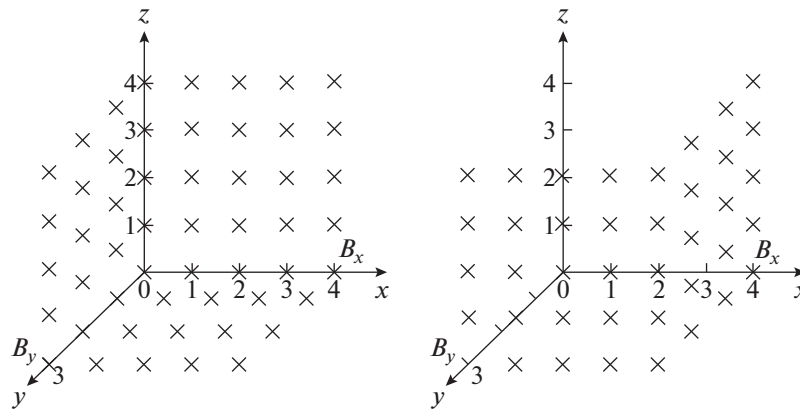
$$\begin{aligned} C = (C_0 | C_1) &= \begin{pmatrix} c_{000} & c_{100} & c_{200} & c_{001} & c_{101} & c_{201} \\ c_{010} & c_{110} & c_{210} & c_{011} & c_{111} & c_{211} \\ c_{020} & c_{120} & c_{220} & c_{021} & c_{121} & c_{221} \end{pmatrix} = \\ &= \begin{pmatrix} 1 & 2 & 3 & 3 & 4 & 5 \\ 4 & 5 & 6 & 6 & 8 & 8 \\ 1 & 2 & 3 & 3 & 4 & 5 \end{pmatrix}. \end{aligned}$$

Расположение элементов матрицы коэффициентов C в декартовой системе координат представлено на рис. 1

Поставим задачу: найти значение функции $f(x, y, z)$ в точке A с координатами $(2, 1, 4)$.

Трехмерная матрица данных Коши задается по слоям, начиная с самого нижнего:

Произведем переход из декартовой системы координат в “матричную”: $\beta_0 = [1, 1, 1] \rightarrow \beta_1 = [2, 2, 2]$, $A(2, 1, 4) \rightarrow A_1(2, 3, 5)$.

Рис. 2. Расположение элементов F в декартовой системе координат.

Проверим задачу на разрешимость. Так как

$$80 = c_{111} > \sum_{(\alpha_1, \alpha_2) \neq (1,1), \alpha_3=1} |c_{\alpha_1 \alpha_2 \alpha_3}| = 38,$$

то задача разрешима.

Для нахождения неизвестных (нулевых) значений в матрице F на втором слое необходимо решить систему разностных уравнений

$$\begin{cases} P(\delta_1, \delta_2, \delta_3)f(x, y, z) = 0, & (x, y, z) \in \Pi, \\ \text{для которых } z = 0, 1, \\ f(x, y, z) = \varphi(x, y, z), & (x, y, z) \in L, \\ \text{для которых } z = 0, 1. \end{cases} \quad (10)$$

Матрица T системы разностных уравнений (10) будет блочной теплицевой матрицей (подробнее см. [29, 30]) вида

$$T = \begin{pmatrix} T_0 & T_1 \\ T_{-1} & T_0 \end{pmatrix},$$

где

$$T_{-1} = \begin{pmatrix} 4 & 5 & 0 \\ 3 & 4 & 5 \\ 0 & 3 & 4 \end{pmatrix}, \quad T_0 = \begin{pmatrix} 80 & 8 & 0 \\ 6 & 80 & 8 \\ 0 & 6 & 80 \end{pmatrix},$$

$$T_1 = \begin{pmatrix} 4 & 5 & 0 \\ 3 & 4 & 5 \\ 0 & 3 & 4 \end{pmatrix}.$$

Решив систему уравнений (10), получим

$$F_1 = \begin{pmatrix} 3.00 & 4.00 & 5.00 & 7.00 & 6.00 \\ 6.00 & -2.13 & -1.86 & -2.64 & 8.00 \\ 3.00 & -1.40 & -1.39 & -2.48 & 4.00 \\ 6.00 & 5.00 & 3.00 & 2.00 & 3.00 \end{pmatrix}.$$

Для нахождения неизвестных (нулевых) значений в матрице F на третьем слое необходимо решить систему разностных уравнений

$$\begin{cases} P(\delta_1, \delta_2, \delta_3)f(x, y, z) = 0, & (x, y, z) \in \Pi, \\ \text{для которых } z = 1, 2, \\ f(x, y, z) = \varphi(x, y, z), & (x, y, z) \in L, \\ \text{для которых } z = 1, 2. \end{cases} \quad (11)$$

Матрица системы уравнений (11) будет равна матрице системы уравнений (10). Решив систему уравнений (11), получим

$$F_2 = \begin{pmatrix} 3.00 & 4.00 & 5.00 & 7.00 & 6.00 \\ 5.00 & -1.19 & -0.50 & -1.99 & 2.00 \\ 3.00 & -1.01 & 0.02 & -1.22 & 4.00 \\ 6.00 & 8.00 & 1.00 & 4.00 & 3.00 \end{pmatrix}.$$

Продолжая процесс, найдем

$$F_3 = \begin{pmatrix} 3.00 & 4.00 & 5.00 & 7.00 & 6.00 \\ 6.00 & -1.40 & -0.75 & -1.52 & 1.00 \\ 3.00 & -1.48 & -0.36 & -1.29 & 4.00 \\ 6.00 & 6.00 & 7.00 & 1.00 & 4.00 \end{pmatrix},$$

$$F_4 = \begin{pmatrix} 3.00 & 4.00 & 5.00 & 7.00 & 6.00 \\ 3.00 & -1.16 & -0.71 & -2.11 & 8.00 \\ 3.00 & -1.17 & -0.47 & -1.88 & 4.00 \\ 6.00 & 2.00 & 6.00 & 7.00 & 3.00 \end{pmatrix}.$$

Таким образом, искомое значение $f(2, 1, 4) = -0.71$.

5. ОПИСАНИЕ АЛГОРИТМА

Алгоритм был реализован в среде MatLab2014 32bit. Вычисления производились на машине Intel(R) Core(TM) i5-3330S CPU 2.70 GHz, 32bit, ОЗУ 4.00 Гб под управлением Windows 7 Корпоративная SP1. Время счета для приведенного примера составило менее 1 секунды.

Алгоритм 1. Схема алгоритма.**Input:** C, β_0, A, F .**Output:** Значение функции $f(x, y, z)$ в точке A .**begin**

```

 $\beta_1$  := координаты точки  $\beta_0$  в “матричной” системе координат;
if  $|C(\beta_1(1), \beta_1(2), \beta_1(3))| < \sum(\sum |C(:, :, \beta_1(3))|) - |C(\beta_1(1), \beta_1(2), \beta_1(3))|$ 
then
  return Ошибка ввода матрицы  $C$ ;
 $A_1$  := координаты точки  $A$  в “матричной” системе координат;  $Q := C(:, :, end)$ ;
 $p = size(Q, 2)$ ;  $NAD = \beta_1(1) - 1$ ;
 $POD = size(Q, 1) - \beta_1(1)$ ;
 $sTblock = size(F, 1) * size(F, 2) - size(F, 1) * (size(Q, 2) - 1) -$ 
 $-(size(F, 2) - (size(Q, 2) - 1)) * (size(Q, 1) - 1)$ ;
 $T := zeros(sTblock, sTblock)$ ;
 $a = Q(\beta_1(1), :)$ ;  $e = \beta_1(2)$ ;
 $column_c = zeros(p, 1)$ ;  $col = 1$ ;
while  $e \geq 1$  do
   $column_c(col) := a(e)$ ;  $col := col + 1$ ;  $e := e - 1$ ;
 $e := \beta_1(2)$ ;  $row_c := zeros(1, p)$ ;  $r := 1$ ;
for  $m$  from  $e$  to  $length(a)$  do
   $row_c(r) := a(e)$ ;  $r := r + 1$ ;  $e := e + 1$ ;
 $T_0 := toeplitz(column_c, row_c)$ ;
for  $i$  from 0 to  $sTblock / (size(Q, 2)) - 1$  do
   $T(((i * p) + 1) : p * (i + 1), ((i * p) + 1) : p * (i + 1)) := T_0$ ;
for  $w$  from 1 to  $NAD$  do
   $a := Q(\beta_1(1) - w, :)$ ;  $e := \beta_1(2)$ ;
   $column_c := zeros(p, 1)$ ;  $col := 1$ ;
  while  $e \geq 1$  do
     $column_c(col) := a(e)$ ;  $col := col + 1$ ;  $e := e - 1$ ;
   $e := \beta_1(2)$ ;  $row_c = zeros(1, p)$ ;  $r := 1$ ;
  for  $m$  from  $e$  to  $length(a)$  do
     $row_c(r) := a(e)$ ;  $r := r + 1$ ;  $e := e + 1$ ;
   $T_{NAD} := toeplitz(column_c, row_c)$ ;
  for  $i$  from 0 to  $sTblock / (size(Q, 2)) - 1 - w$  do
     $T((((i * p) + 1)) : p * (i + 1), ((i * p) + 1) + p * w : p * (i + 1) + p * w) := T_{NAD}$ ;
for  $w$  from 1 to  $POD$  do
   $a := Q(\beta_1(1) + w, :)$ ;  $e := \beta_1(2)$ ;
   $column_c = zeros(p, 1)$ ;  $col = 1$ ;
  while  $e \leq 1$  do
     $column_c(col) = a(e)$ ;  $col = col + 1$ ;  $e = e - 1$ ;
   $e = \beta_1(2)$ ;  $row_c = zeros(1, p)$ ;  $r = 1$ ;
  for  $m$  from  $e$  to  $length(a)$  do
     $row_c(r) = a(e)$ ;  $r = r + 1$ ;  $e = e + 1$ ;
   $T_{POD} = toeplitz(column_c, row_c)$ ;
  for  $i$  from 0 to  $sTblock / (size(Q, 2)) - 1 - w$  do
     $T((((i * p) + 1) + p * w : p * (i + 1) + p * w, ((i * p) + 1) : p * (i + 1))) := T_{POD}$ 

```

```

b := zeros (size (T,1),1);
for k from  $\beta_1(3)$  to A(3) do
    t := 1;
    for i from 0 to (size(F,2) – size(C,2)) do;
        for j from 0 to (size(F,1) – size(C,1)) do
            do
                 $f_{iter} := F(j+1 : size(C,1) + j,$ 
                     $(i+1) : size(C,2) + i,$ 
                     $k - size(C,3) + 1 : k);$ 
                 $f_{new} := f_{iter} * C;$ 
                 $b(t) = -(\sum(\sum(\sum(f_{new}))));$ 
                t := t + 1
            do
        end for j
    end for i
    elementy := linsolve(T,b); h := 1;
    for i from 0 to (size(F,2) – size(C,2)) do
        for j from 0 to (size(F,1) – size(C,1)) do
             $F(\beta_1(1) + j, \beta_1(2) + i, k) := elementy(h);$ 
            h := h + 1;
        end for j
    end for i
return f(A)

```

6. ЗАКЛЮЧЕНИЕ

Классические методы решения дифференциальных уравнений (методы Рунге–Кутты четвертого порядка, Эйлера, Ньютона и т.д.) создают определенные трудности при их применении в многомерном случае. В каждой конкретной задаче необходимо индивидуально подбирать способ решения. Это делает разработку универсального подхода решения многомерных дифференциальных уравнений с использованием указанных методов трудоемкой. В проведенном исследовании мы расширили возможность использования стандартных симметричных разностных схем для аппроксимации двумерных дифференциальных уравнений с постоянными коэффициентами, где рассматривается задача “на плоскости”, в которой, как правило, в качестве независимых переменных x, y используют временную и одну пространственную переменные. Предложенная нами схема, во-первых, имеет произвольное число точек, а во-вторых, может применяться для аппроксимации дифференциальных уравнений в трехмерном случае. Для уравнения теплопроводности это означало бы, что при описании теплопереноса появляется возможность учитывать анизотропность этого процесса, через добавление дополнительной пространственной переменной.

С математической точки зрения процесс аппроксимации дифференциальных уравнений разностной схемой в трехмерном случае дает систему разностных уравнений, матрица которых имеет блочный теплицевый вид. Процесс решения таких систем для большого количества неизвестных и начальных данных связан с большими вычислительными затратами. В нашей работе мы предложили алгоритм решения такой системы через значения коэффициентов разностного уравнения, что позволило автоматизировать решение задачи Коши с начальными данными в параллелепипеде. Таким образом, предложенный нами подход определяет единый алгоритм для решения дифференциальных уравнений при их аппроксимации разностной схемой в параллелепипеде.

7. БЛАГОДАРНОСТИ

Работа поддержана Красноярским математическим центром, финансируемым Минобрнауки РФ в рамках мероприятий по созданию и развитию региональных НОМЦ (Соглашение 075-02-2022-876).

СПИСОК ЛИТЕРАТУРЫ

1. Годунов С.К., Рябенский В.С. Разностные схемы. Введение в теорию. Изд. 2, перераб. и доп. М.: Наука. 1977. 440 с.

2. Самарский А.А. Введение в теорию разностных схем. М.: Наука. 1971. 553 с.
3. Bousquet-Melou M., Petkovsek M. Linear Recurrences with Constant Coefficients: the Multivariate Case // Discrete Mathematics. 2000. V. 225. № 1–3. P. 51–75.
4. Стенли Р. Перечислительная комбинаторика: Пер. с англ. М.: Мир. 1990. 440 с.
5. Стенли Р. Перечислительная комбинаторика. Деревья, производящие функции и симметрические функции: Пер. с англ. М.: Мир. 2005. 767 с.
6. Duffin R.J. Basic Properties of Discrete Analytic Functions // Duke Math. J. 1956. V. 23. № 2. P. 335–363.
7. Duffin R.J. Potential Theory on Rhombic Lattice // J. Combinatorial Theory. 1968. V. 5. № 3. P. 258–272.
8. Данилов О.А. Интерполяционная формула Лагранжа для дискретной аналитической функции // Вестник НГУ. Серия: Математика, механика, информатика. 2008. Т. 8. № 4. С. 33–39.
9. Данилов О.А., Медных А.Д. Дискретные аналитические функции многих переменных и формула Тейлора // Вестник НГУ. Серия: Математика, механика, информатика. 2009. Т. 9. № 2. С. 38–46.
10. Даджион Д., Мерсеро Р. Цифровая обработка многомерных сигналов. М.: Мир. 1988. 448 с.
11. Изерман Р. Цифровые системы управления. М.: Мир. 1984. 541 с.
12. Abramov S.A., Barkatou M.A., van Hoeij M., Petkovsek M. Subanalytic Solutions of Linear Difference Equations and Multidimensional Hypergeometric Sequences // Journal of Symbolic Computation. 2011. V. 46. № 11. P. 1205–1228.
13. Abramov S.A., Gheffar A., Khmelnov D.E. Rational Solutions of Linear Difference Equations: Universal denominators and denominator bounds // Programming and Computer Software. 2011. V. 37. № 2. P. 78–86.
14. Abramov S.A., Petkovsek M., Ryabenko A.A. Hypergeometric Solutions of First-order Linear Difference Systems with Rational-function Coefficients. Lecture Notes in Computer Science 9301. 2015. P. 1–14.
15. Abramov S.A. Search of Rational Solutions to Differential and Difference Systems by Means of Formal Series // Programming and Computer Software. 2015. V. 41. № 2. P. 65–73.
16. Kytmanov A.A., Lyapin A.P., Sadykov T.M. Evaluating the Rational Generating Function for the Solution of the Cauchy Problem for a Two-dimensional Difference Equation with Constant Coefficients // Programming and computer software. 2017. V. 43. № 2. P. 105–111.
17. Leinartas E.K., Lyapin A.P. On the Rationality of Multidimensional Recursive Series // Journal of Siberian Federal University. Mathematics & Physics. 2009. V. 2. № 4. P. 449–455.
18. Lyapin A.P., Chandragiri S. Generating Functions For Vector Partition Functions And A Basic Recurrence Relation // Journal of Difference Equations & Applications. 2019. V. 25. № 7. P. 1052–1061.
19. Lyapin A.P. Riordan's Arrays and Two-dimensional Difference Equations // Journal of Siberian Federal University. Mathematics & Physics. 2009. V. 2. № 2. P. 210–220.
20. Apanovich M.S., Lyapin A.P., Shadrin K.V. Solving the Cauchy Problem for a Two-Dimensional Difference Equation at a Point Using Computer Algebra Methods // Programming and Computer Software. 2021. V. 47. № 1. P. 1–5.
21. Апанович М.С., Ляпин А.П., Шадрин К.В. Алгоритм вычисления решения задачи Коши для двумерного разностного уравнения с начальными данными, заданными в “полосе”. Программирование. 2022. № 4. С. 50–56.
22. Murray J.D. Mathematical Biology. II: Spatial Models and Biomedical Applications. Berlin [Germany] : Springer-Verlag. 2003. 811 p.
23. Братусь А.С., Новожилов А.С., Платонов А.П. Динамические системы и модели биологии. М.: ФИЗМАТЛИТ. 2010. 400 с.
24. Ляпин А. П., Кучта Т. Сечения производящего ряда решения разностного уравнения в симплициальном конусе // Известия Иркутского государственного университета. Серия Математика. 2022. Т. 42. С. 75–89.
25. Федорюк М.В. Асимптотика: интегралы и ряды. М.: Наука. 1987. 544 с.
26. Рогозина М.С. Разрешимость разностной задачи Коши для многослойных неявных разностных схем // Вестник СибГАУ. Математика, механика, информатика. 2014. Т. 55. № 3. С. 126–130.
27. Apanovich M.S. Correctness of a Two-dimensional Cauchy Problem for a Polynomial Difference Operator with Constant Coefficients // Journal of Siberian Federal University. Mathematics & Physics. 2017. V. 10. № 2. P. 199–205.
28. Апанович М.С. О разрешимости трехмерной разностной задачи Коши // ИТНОУ: Информационные технологии в науке, образовании и управлении. 2018. № 5(9). С. 9–13.
29. Апанович М.С., Ляпин А.П., Шадрин К.В. Вычисление последовательности главных миноров теплоцевой ленточной матрицы // Прикладная математика & Физика. 2020. Т. 52. № 1. С. 5–10.
30. Иохвидов И.С. Ганкелевы и теплоцевы матрицы. М.: Наука. 1974. 264 с.

УДК 517.5+004.434

О ВЫЧИСЛЕНИИ ЛИНИЙ УРОВНЯ МНОГОЧЛЕНА НА ПЛОСКОСТИ

© 2023 г. А. Д. Брюно^{a,*}, А. Б. Батхин^{a,b,**}, З. Х. Хайдаров^{c,***}^aИнститут прикладной математики им. М.В. Келдыша РАН
125047 Москва, Миусская пл., д. 4, Россия^bМосковский физико-технический институт (государственный университет)
141701 г. Долгопрудный Московской области, Институтский пер., д. 9, Россия^cСамаркандский государственный университет им. Ш. Рашидова
140104 г. Самарканд, Университетский бул., д. 15, Узбекистан

*E-mail: abruno@keldysh.ru

**E-mail: batkhin@gmail.com

***E-mail: zafarxx@gmail.com

Поступила в редакцию 05.07.2022 г.

После доработки 01.08.2022 г.

Принята к публикации 21.10.2022 г.

Демонстрируется применение метода вычисления расположения всех типов линий уровня вещественного многочлена на вещественной плоскости, теория которого основана на методах локального и глобального анализа средствами степенной геометрии и компьютерной алгебры. Подробно рассмотрены 3 нетривиальных примера вычисления линий уровня вещественных многочленов на вещественной плоскости. При этом используются следующие алгоритмы компьютерной алгебры: факторизация многочлена, вычисление базиса Гребнера, построение многоугольника Ньютона, изображение алгебраической кривой на плоскости. Показано, как преодолевать вычислительные трудности.

DOI: 10.31857/S0132347423020073, EDN: MFOETX

1. ВВЕДЕНИЕ

Различные прикладные задачи предполагают построение линий уровня для двумерных скалярных полей. Соответствующее программное обеспечение использует различные вычислительные алгоритмы, которые хорошо справляются с ситуацией, когда у линии уровня нет особенностей. Наличие особенностей заставляет существенно уменьшать шаг разбиения и, соответственно, увеличивать объем вычислений. В статье подробно разбирается, как используя методы степенной геометрии и алгоритмы компьютерной алгебры можно радикально улучшить качество решения указанной задачи.

Пусть $X = (x_1, x_2) \in \mathbb{R}^2$. Рассмотрим вещественный многочлен $f(X)$. При постоянной $c \in \mathbb{R}$ кривая на плоскости $\mathbb{R}^2$

$$f(X) = c \quad (1.1)$$

является *линией уровня* многочлена $f(X)$.

Наша задача — описать все типы линий уровня многочлена $f(X)$ на вещественной плоскости $X \in \mathbb{R}^2$.

В разделе 2 кратко изложена теория построения линий уровня вещественного многочлена на вещественной плоскости согласно [1] и [2]. В разделе 3 выписан многочлен 6-й степени от трех переменных x, y, z , возникший в задаче линейной устойчивости положения равновесия некоторой механической системы [3]. Нули этого многочлена описывают границу области устойчивости. Затем в разделах 4, 5 и 6 рассматриваются сечения этой границы плоскостями $x = 0$, $y = 0$, $z = 0$ соответственно. На каждой из этих трех плоскостей вычисляются линии уровня соответствующих многочленов. При этом используются такие алгоритмы компьютерной алгебры как факторизация многочлена, вычисление базиса Гребнера, построение многоугольника Ньютона, изображение алгебраической кривой на плоскости.

2. ТЕОРИЯ МЕТОДА

2.1. Основная теорема

Пусть $C_* = \inf f(X)$ и $C^* = \sup f(X)$ по $X \in \mathbb{R}^2$.

Теорема 1. *Имеется конечное множество критических значений c :*

$$C_* < c_1^* < c_2^* < \dots < c_m^* < C^*, \quad (2.1)$$

которым соответствуют критические линии уровня

$$f(X) = c_j^*, \quad j = 1, \dots, m, \quad (2.2)$$

а для значений c из каждого из $m + 1$ интервала

$$I_0 = (C_*, c_1^*), \quad I_j = (c_j^*, c_{j+1}^*), \quad I_m = (c_m^*, C^*), \quad (2.3)$$

$$j = 1, \dots, m-1,$$

линии уровня топологически эквивалентны. Если $C_* = c_1^*$ или $C^* = c_m^*$, то интервалы I_0 или I_m отсутствуют.

Поэтому для выявления расположения всех линий уровня многочлена $f(X)$ надо найти все критические значения c_j^* , изобразить m критических линий уровня (2.2) и по одной линии уровня для произвольного значения c из $m + 1$ интервала (2.3). Способ вычисления этих линий уровня с помощью степенной геометрии описан в [1] и отчасти в [4, гл. I, § 2]. Более традиционный подход см. в [5, гл. I]. Локальное строение линий уровня многочлена рассматривалось в [4, гл. I, § 3]. Здесь некоторые результаты из [4, гл. I, § 3] дополнены.

2.2. Критические точки и критические кривые

Точка $X = X^0$ называется *простой* для многочлена $f(X)$, если в ней отлична от нуля хотя бы одна из частных производных $\partial f / \partial x_1, \partial f / \partial x_2$.

Определение 1. Точка $X = X^0$ для многочлена $f(X)$ называется *критической порядка k* , если в точке $X = X^0$ равны нулю все частные производные от $f(X)$ до порядка k , т.е. все

$$\frac{\partial^l f}{\partial x_1^i \partial x_2^j}(X^0) = 0, \quad 1 \leq i + j = l \leq k,$$

и отлична от нуля хотя бы одна частная производная порядка $k + 1$.

Определение 2. *Кривая*

$$g(X) = 0 \quad (2.4)$$

называется *критической* для многочлена $f(X)$, если

1. она лежит на какой-то линии уровня (1.1) и
2. на ней $\partial f / \partial x_1 \equiv 0$, или $\partial f / \partial x_2 \equiv 0$.

Значения постоянной $c = f(X)$ в критических точках $X = X^0$ и на критических линиях (2.4) назовем *критическими* и обозначим c_j^* согласно (2.1).

2.3. Локальный анализ линий уровня

В дальнейшем вблизи точки $X = X^0 = (x_1^0, x_2^0)$ будем рассматривать аналитические обратимые замены координат

$$y_i = x_i^0 + \varphi_i(x_1 - x_1^0, x_2 - x_2^0), \quad i = 1, 2, \quad (2.5)$$

где φ_i — аналитические функции от $X - X^0$.

Лемма 1 ([4, гл. I, § 3]). *Если точка X^0 простая и в ней $\partial f / \partial x_2 \neq 0$, то существует замена (2.5), приводящая уравнение (1.1) к виду*

$$f(X) = y_2 = c. \quad (2.6)$$

Она следует из теоремы о неявной функции.

Линии уровня (2.6) — это прямые, параллельные оси y_1 .

Рассмотрим решения уравнения (1.1) вблизи критической точки $X^0 = 0$ порядка 1. Тогда

$$f(X) = f_0 + ax_1^2 + bx_1x_2 + cx_2^2 + \dots$$

Дискриминант Δ выписанной квадратичной формы есть $\Delta = b^2 - 4ac$.

Лемма 2 ([4, гл. I, § 3]). *Если в критической точке первого порядка $X^0 = 0$ дискриминант $\Delta \neq 0$, то существует замена (2.5), приводящая уравнение (1.1.) к виду*

$$f(X) = f_0 + \sigma y_1^2 + y_2^2 = c, \quad (2.7)$$

где $\sigma = 1$ (если $\Delta < 0$) или $\sigma = -1$ (если $\Delta > 0$).

Это двумерный вариант известной леммы Морса [6, гл. I, § 2].

Лемма 3. *Если в критической точке первого порядка $X^0 = 0$ дискриминант $\Delta = 0$, то существует замена (2.5), приводящая уравнение (1.1) к виду*

$$f(X) = f_0 + y_2^2 + \tau y_1^n = c, \quad (2.8)$$

где целое $n > 2$ и число $\tau \in \{-1, 0, +1\}$.

Выражения (2.7) и (2.8) — это *нормальные формы* многочлена $f(X)$ вблизи его критической точки первого порядка $X^0 = 0$.

Пусть теперь критическая точка $X^0 = 0$ имеет порядок $k > 1$. Согласно [1, п. 5] соответствующая ей линия уровня либо не имеет ветвей, входящих в критическую точку $X^0 = 0$, либо имеет несколько таких ветвей.

В первом случае критическая линия уровня состоит из этой точки $X^0 = 0$, а остальные линии уровня являются замкнутыми кривыми вокруг нее и соответствуют одному знаку разности $c - f(X^0)$.

Во втором случае критическая линия уровня состоит из конечного числа ветвей разных крат-

ностей, входящих в критическую точку $X^0 = 0$. Они разбивают окрестность этой критической точки на криволинейные секторы. Остальные линии уровня заполняют эти секторы, оставаясь на некотором расстоянии от критической точки $X^0 = 0$. При этом в соседних секторах они соответствуют разным знакам разности $c - f(X^0)$, если разделяющая их ветвь имеет нечетную кратность, и одному знаку этой разности, если разделяющая их ветвь имеет четную кратность.

2.4. Глобальный анализ линий уровня

Используемые в этом подразделе объекты степенной геометрии — многоугольник Ньютона $\Gamma(f)$ многочлена f , ребро $\Gamma_j^{(1)}$ многоугольника, его внешняя нормаль $P_j^{(1)}$, укороченное уравнение $\hat{f}_j^{(1)}(X)$, соответствующее ребру $\Gamma_j^{(1)}$, — подробно описаны в [1].

Определение 3. Пусть $\Gamma_j^{(1)}$ — ребро многоугольника Ньютона многочлена $g(X) = f(X) - c$, которое имеет внешнюю нормаль $P_j^{(1)} = (p_1, p_2)$ с одной или двумя положительными координатами p_i и соответствует укороченному многочлену $\hat{g}_j^{(1)}(X)$. Тогда пересечение корня $x_1 = A\tau^{p_1}$, $x_2 = B\tau^{p_2}$ (где постоянные $A, B \in \mathbb{R}$) укороченного многочлена $\hat{g}_j^{(1)}(X)$ при $\tau \rightarrow \pm\infty$ с бесконечностью $x_i = \pm\infty$ назовем *бесконечной точкой пересечения*. Кратность корня — это *кратность* этой точки.

Лемма 4. Если все ребра $\Gamma_j^{(1)}$ из определения 3 не содержат точки $(q_1, q_2) = 0$ и у многочлена $f(X)$ все бесконечные точки пересечения имеют кратность один, то для всех значений постоянной c из одного из $t + 1$ интервалов (2.3) линии уровня (1.1) топологически эквивалентны.

3. БОЛЬШОЙ МНОГОЧЛЕН

3.1. Описание

В работе [3] рассматривалась устойчивость в линейном приближении стационарного решения одной многопараметрической системы Гамильтона. В процессе исследования возникла необходимость изучить нули некоторого многочлена $g(x, y, z)$ 6-го порядка от трех переменных x, y, z :

$$g(x, y, z) = 4x^5y - 4x^5z + 48x^4y^2 - 92x^4yz + 40x^4z^2 + 344x^3y^3 - 904x^3y^2z + 736x^3yz^2 - 192x^3z^3 + 1200x^2y^4 - 4376x^2y^3z +$$

$$+ 5360x^2y^2z^2 - 2752x^2yz^3 + 512x^2z^4 + 2500xy^5 - 11700xy^4z + 19168xy^3z^2 - 14656xy^2z^3 + 5376xyz^4 - 768xz^5 - 7500y^5z + 21800y^4z^2 - 25408y^3z^3 + 14848y^2z^4 - 4352yz^5 + 512z^6 + 2x^5 + 82x^4y - 64x^4z + 740x^3y^2 - 1152x^3yz + 456x^3z^2 + 3156x^2y^3 - 6784x^2y^2z + 5144x^2yz^2 - 1344x^2z^3 + 6490xy^4 - 14464xy^3z + 13816xy^2z^2 - 6720xyz^3 + 1344xz^4 + 20250y^5 - 45120y^4z + 37928y^3z^2 - 14976y^2z^3 + 2880yz^4 - 256z^5 + 13x^4 + 412x^3y - 344x^3z + 2782x^2y^2 - 5176x^2yz + 2200x^2z^2 + 828xy^3 - 12936xy^2z + 15888xyz^2 - 5088xz^3 - 37827y^4 + 67608y^3z - 35496y^2z^2 + 2016yz^3 + 1872z^4 + 64x^3 + 1728x^2y - 1152x^2z + 15552xy^2 - 20736xyz + 6912xz^2 + 46656y^3 - 93312y^2z + 62208yz^2 - 13824z^3. \quad (3.1)$$

Здесь исследуем линии уровня на трех нулевых сечениях поверхности $g(x, y, z) = 0$, т. е. для функций $f_1(y, z) = g(x, y, z)|_{x=0}$, $f_2(x, z) = g(x, y, z)|_{y=0}$ и $f_3(x, y) = g(x, y, z)|_{z=0}$.

3.2. Вычисление критических значений, точек и кривых

Учитывая полиномиальный характер задачи, вычисление критических значений c_j^* и соответствующих им критических точек и кривых удобно проводить с использованием алгоритмов компьютерной алгебры, в первую очередь, вычисляя базисы Гребнера соответствующих идеалов с подходящими мономиальными порядками (подробнее, см. [7, гл. 2, 3]).

Идеал, определяющий критические точки и кривые, состоит из многочлена $f(X) - c$ и его частных производных по X :

$$\mathcal{J} = \left\{ f(X) - c, \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \right\}. \quad (3.2)$$

Как следует из основной теоремы, число критических значений c_j^* конечно. Тогда базис Гребнера $\mathcal{GB}\mathcal{J}$ идеала (3.2) для чисто лексикографического порядка $c \succ x_1 \succ x_2$ содержит полином $h(c)$, зависящий только от переменной c . Среди веще-

ственных корней этого полинома следует искать критические значения c_j^* , для которых имеются вещественные критические линии уровня.

Базис Гребнера, задаваемый идеалом $\mathcal{GB}\mathcal{J}$, имеет размерность либо 0, либо 1. В первом случае критических кривых нет, ибо идеал $\mathcal{GB}\mathcal{J}$ нульмерен. Во втором случае некоторым критическим значениям c_j^* соответствуют критические кривые.

Вычисление и исследование идеала $\mathcal{GB}\mathcal{J}$ удобно выполнять с использованием какой-либо системы компьютерной алгебры, как свободно распространяемой (например, Sympy, Sage, Singular, Macaulya2) или коммерческой (например, Maple, Mathematica, Magma). В любой из них имеются процедуры построения базисов Гребнера для различных лексикографических порядков, а также некоторые дополнительные процедуры для проверки нульмерности идеала, вычисления его размерности и др. Все вычисления в последующих разделах работы были выполнены в СКА Maple с использованием пакета Groebner. Отметим, что имеется пакет PolynomialIdeals, в состав которого входят процедуры примарной декомпозиции или декомпозиции идеала на его нульмерные подидеалы и другие, но, во-первых, все процедуры этого пакета реализуются с помощью процедур пакета Groebner, а во-вторых, авторы предпочитают детально контролировать процесс вычислений.

Используется следующий порядок вычислений.

1. Составляется идеал $\mathcal{J}$ и для него вычисляется базис Гребнера $\mathcal{GB}\mathcal{J}$ с чисто лексикографическим порядком $c \succ x_1 \succ x_2$ с применением процедуры Basis пакета Groebner.

2. Базис $\mathcal{GB}\mathcal{J}$ позволяет найти все критические значения c_j^* с помощью полинома $h(c)$ либо точно, либо в виде алгебраических чисел. Такой полином автоматически определяется для указанного выше лексикографического порядка, или с помощью процедуры UnivariatePolynomial и, по возможности, раскладывается на множители.

3. Среди критических значений c_j^* отбираются те, которым соответствуют вещественные критические точки и критические кривые. Для этого для каждого множителя $h_j(c)$ многочлена $h(c)$ вычисляется новый базис Гребнера на основе ранее вычисленного базиса и этого множителя, но уже с другими лексикографическими порядками: $c \prec x_1 \prec x_2$ и $c \prec x_2 \prec x_1$. Это есть некоторый вариант примарной декомпозиции, который позволяет разделить многочлены только от x_1 или от x_2 , задающие координаты критических точек.

4. Теперь для каждого примарного идеала следует найти его множество вещественных нулей.

Если критическое значение c_j^* принадлежит $\mathbb{Q}$, то все вычисления выполняются точно. Если же оно есть иррациональное алгебраическое число, то можно поступить так, как описано в [1, п. 5.6], т.е. проводить все вычисления по модулю идеала, определяющего критическое значение и критическую точку. В данной работе вычисления иррациональных критических значений и иррациональных координат критических точек проводятся приближенно с повышенной точностью.

5. Вид нормальной формы (ВНФ) (2.7) или (2.8) каждой критической точки определяется с использованием дискриминанта Δ квадратичной формы разложения функции $f(X)$ вблизи нее согласно леммам 2 и 3. Разложения вещественных ветвей линий уровня в окрестности каждой критической точки и на бесконечности вычисляются с использованием алгоритмов степенной геометрии. В частности, в СКА Maple удобно выполнять такие вычисления с помощью пакета PolyhedralSets, а также пакета процедур из пакета algcurves.

6. Проведенные вычисления позволяют выполнить построение эскизов линий уровня так, как это описано в [2].

3.3. Замечания об обозначениях

Последующие разделы работы содержат большое число различных объектов, поэтому дадим здесь краткое описание принятой нотации.

- Каждое из нулевых сечений поверхности, заданной многочленом (3.1), обозначается символом f_j , где j принимает значения 1, 2, 3, соответствующие той переменной, которая полагается равной нулю. Многочлен, определяющий критические значения c_j^* обозначается $h_j(c)$, а его k -й множитель соответственно $h_{jk}(c)$.

- Для соответствующего многочлена f_j вычисляется его носитель и многоугольник Ньютона $\Gamma(f_j)$. Все ребра многоугольника нумеруются и обозначаются в тексте $\Gamma_k^{(1)}$, а соответствующий k -му ребру укороченный многочлен обозначается $\hat{f}_k^{(1)}$. Нумерация ребер и укороченных многочленов сквозная в пределах своего раздела. В силу технических трудностей на рисунках многоугольников Ньютона верхний индекс у k -го ребра $\Gamma_k^{(1)}$ опускается. Поскольку в вычислениях используются только грани размерности один, то это не должно привести к путанице.

- Если многочлен f_j преобразуется, то результат его i -го преобразования обозначается $_i f_j$. При этом все соответствующие этому многочлену объекты степенной геометрии имеют соответствующий ле-

вый нижний индекс, т.е. ${}_2\Gamma_3^{(1)}$ обозначает 3-е ребро многоугольника Ньютона многочлена, полученного на втором шаге преобразований, а ${}_2\hat{f}_3^{(1)}$ – соответствующий этому ребру многочлен.

4. ПРИМЕР 1 ПРИ $x = 0$

4.1. Критические значения

Рассмотрим вычисление линий уровня многочлена $f_1(y, z) = g(x, y, z)|_{x=0}$.

$$\begin{aligned} f_1 = & -7500y^5z + 21800y^4z^2 - 25408y^3z^3 + \\ & + 14848y^2z^4 - 4352yz^5 + 512z^6 + 20250y^5 - \\ & - 45120y^4z + 37928y^3z^2 - 14976y^2z^3 + \\ & + 2880yz^4 - 256z^5 - 37827y^4 + 67608y^3z - \\ & - 35496y^2z^2 + 2016yz^3 + 1872z^4 + 46656y^3 - \\ & - 93312y^2z + 62208yz^2 - 13824z^3. \end{aligned} \quad (4.1)$$

Для многочлена (4.1) составляем идеал Id_1 , определяющий критические точки и кривые: $Id_1 = \{f_1 - c, \partial f_1/\partial x, \partial f_1/\partial z\}$. Для него строим базис Гребнера $\mathcal{GB}\mathcal{J}_1$ с чисто лексикографическим порядком $y \prec z \prec c$, в который входят полиномы, зависящие соответственно от $\{c\}$, $\{c, z\}$, $\{c, z\}$, $\{c, y\}$, $\{c, y, z\}$, $\{c, y, z\}$. Получаем 6 полиномов, некоторые из которых имеют 15-ю степень и выглядят очень громоздко. Поэтому мы их не приводим здесь в явном виде. Пользуемся первым полиномом, определяющим критические значения полинома (4.1), и разлагаем его на множители [7, гл. 2, 3].

$$\begin{aligned} h_1(c) = & c^2(791015625 \times 10^{12}c^4 - \\ & - 4137285273243305806216 \times 10^5c^3 + \\ & + 2268208541119374290121362928c^2 - \\ & - 42844219213475982104286084531c + \\ & + 38437670697580573719552 \times 10^6). \end{aligned}$$

Первый множитель этого полинома $h_{11}(c) = c^2$ задает первое критическое значение $c_1^* = 0$. Вторым множителем h_{12} имеет два вещественных корня $c_2^* \approx 0.9355$, $c_3^* \approx 523029.1004$ и пару комплексных корней. Добавляя каждый множитель многочлена $h_1(c)$ в найденный выше базис Гребнера, составляем его примарную декомпозицию. Рассмотрим эти критические значения последовательно.

4.2. Критическая линия $h(y, z) = c_1^* = 0$

Вычисляя идеал $SP_1(h_1)$ для двух различных номинальных порядков, находим соответствующие многочлены от переменных y и z соответственно

$$\begin{aligned} & y^8(y-2)(y-14) \times \\ & \times (8000y^3 - 50832y^2 + 111321y - 84672)^2, \\ & z^8(z-3)(z-27) \times \\ & \times (2560z^3 - 22176z^2 + 63909z - 61344)^2. \end{aligned}$$

С их помощью определяем, что критическому значению $c_1^* = 0$ соответствуют четыре критические точки: $(y_1, z_1) = (0, 0)$, $(y_2, z_2) = (2, 3)$, $(y_3, z_3) \approx (2.3427, 3.2427)$ и $(y_4, z_4) = (14, 27)$. Рассмотрим ВНФ каждой из этих критических точек.

Многочлен (4.1) при $c = 0$ факторизуется на два множителя: линейный

$$3z - 2y \quad (4.2)$$

и полином пятой степени

$$\begin{aligned} h(y, z) = & 2500y^4z - 5600y^3z^2 + 4736y^2z^3 - \\ & - 1792yz^4 + 256z^5 - 6750y^4 + 10540y^3z - \\ & - 5616y^2z^2 + 1248yz^3 - 128z^4 + 12609y^3 - \\ & - 14130y^2z + 2412yz^2 + 936z^3 - 15552y^2 + \\ & + 20736yz - 6912z^2. \end{aligned} \quad (4.3)$$

Это единственный случай, когда многочлен $f_j - c_k^*$ факторизуется. Поэтому особенности находим для множителя (4.3), но в критической линии учитываем и (4.2). Далее рассматриваем только многочлен $h(y, z)$ и для него применяем систему обозначений подраздела 3.3.

а) *Разложение в точке* $(y_1, z_1) = (0, 0)$. Здесь конус задачи есть $\mathcal{H}_3 = \{P = (p_1, p_2) < 0\}$. Непосредственные вычисления показывают, что квадратичная форма h_2 ряда Тейлора многочлена h есть полный квадрат $h_2 = -1728(3y - 2z)^2$ и имеет общий множитель с кубической формой h_3 . Таким образом, теорема о неявной функции неприменима, и здесь необходимо применить следующую технику [1, п. 5.2].

Вычисляем выпуклую оболочку носителя многочлена (4.3) с использованием пакета Polyhedral-Sets. Многоугольник Ньютона и его ребра показаны на рис. 1.

Ребру $\Gamma_1^{(1)}$ соответствует укороченный многочлен $\hat{h}_1^{(1)} = -1728(3y - 2z)^2$. Выполним подстановку $z = 3y/2 + z_1$ и получим многочлен ${}_1h$ в переменных y, z_1 :

$$\begin{aligned} {}_1h = & 6y^5 - 44y^4z_1 + 160y^3z_1^2 - 256y^2z_1^3 + 128yz_1^4 + \\ & + 256z_1^5 - 12y^4 + 388y^3z_1 - 1728y^2z_1^2 + 480yz_1^3 - \\ & - 128z_1^4 - 576y^2z_1 + 6624yz_1^2 + 936z_1^3 - 6912z_1^2. \end{aligned} \quad (4.4)$$

Его многоугольник Ньютона показан на рис. 2.

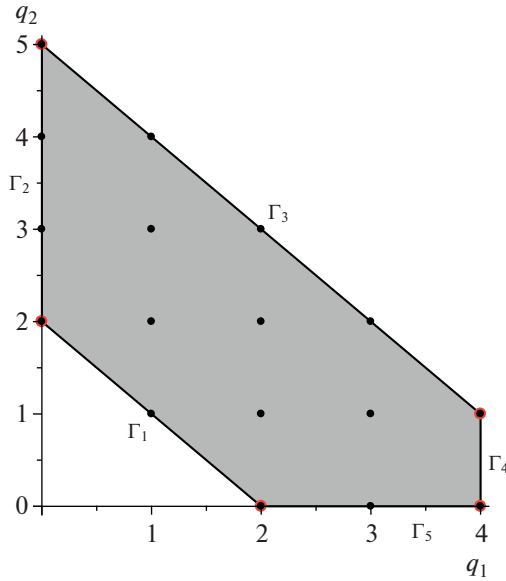


Рис. 1. Многоугольник Ньютона и его ребра для многочлена (4.3).

Единственному ребру ${}_1\Gamma_1^{(1)}$ этого многоугольника, у которого нормальный конус попадает в конус задачи $\mathcal{H}_3 = \{P < 0\}$, соответствует укороченный многочлен ${}_1\hat{h}_1^{(1)} = 12(y^2 + 24z_1)^2$. Следовательно, здесь требуется еще одна подстановка $z_1 = -y^2/24 + z_2$. Она приводит к многочлену ${}_2h$ от переменных y, z_2 :

$$\begin{aligned} {}_2h = & \frac{1}{31104}y^{10} + \frac{1}{2592}y^9 + \frac{5}{1296}y^8z_2 + \frac{47}{2592}y^8 - \\ & - \frac{1}{27}y^7z_2 - \frac{5}{27}y^6z_2^2 + \frac{35}{144}y^7 - \frac{35}{27}y^6z_2 + \frac{4}{3}y^5z_2^2 + \\ & + \frac{40}{9}y^4z_2^3 - \frac{79}{64}y^6 - \frac{65}{6}y^5z_2 + \frac{92}{3}y^4z_2^2 - \frac{64}{3}y^3z_2^3 - \\ & + \frac{40}{9}y^4z_2^3 - \frac{79}{64}y^6 - \frac{65}{6}y^5z_2 + \frac{92}{3}y^4z_2^2 - \frac{64}{3}y^3z_2^3 - \\ & - \frac{160}{3}y^2z_2^4 + \frac{4}{3}y^5 + \frac{839}{8}y^4z_2 + 100y^3z_2^2 - \\ & - \frac{704}{3}y^2z_2^3 + 128yz_2^4 + 256z_2^5 - 164y^3z_2 - 1845y^2z_2^2 + \\ & + 480yz_2^3 - 128z_2^4 + 6624yz_2^2 + 936z_2^3 - 6912z_2^2. \end{aligned} \quad (4.5)$$

Его многоугольник Ньютона (см. рис. 3) имеет единственное ребро ${}_2\Gamma_1^{(1)}$ с нормальным конусом из конуса задачи $\mathcal{H}_3 = \{P < 0\}$, которому соответствует многочлен ${}_2\hat{h}_1^{(1)} = 4y^5/3 - 6912z_2^2$. Таким образом, получаем два решения $z_2 = \pm y^{5/2}/72$. Выполняя обратную подстановку, в итоге получаем разложения двух ветвей

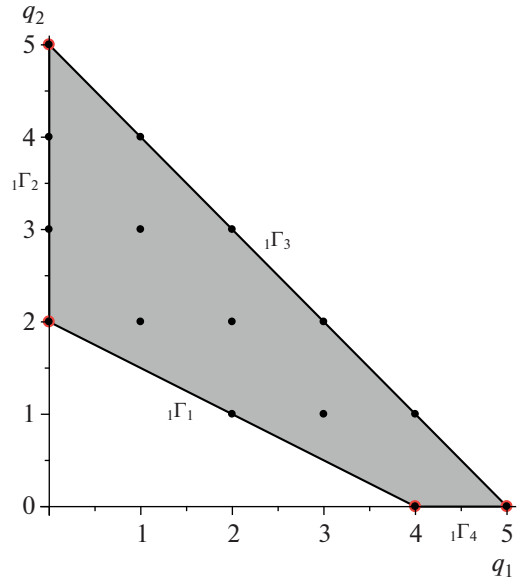


Рис. 2. Многоугольник Ньютона и его ребра для многочлена (4.4).

$$z_{1,2} = \frac{3}{2}y - \frac{1}{24}y^2 \pm \frac{1}{72}y^{5/2},$$

которые в особой точке $(0, 0)$ имеют касание второго порядка.

Вычисления с использованием функции `puil-seux` из пакета `algcures` системы Maple или с использованием функции `AsymptoticSolve` системы Mathematica приводят к таким же результатам.

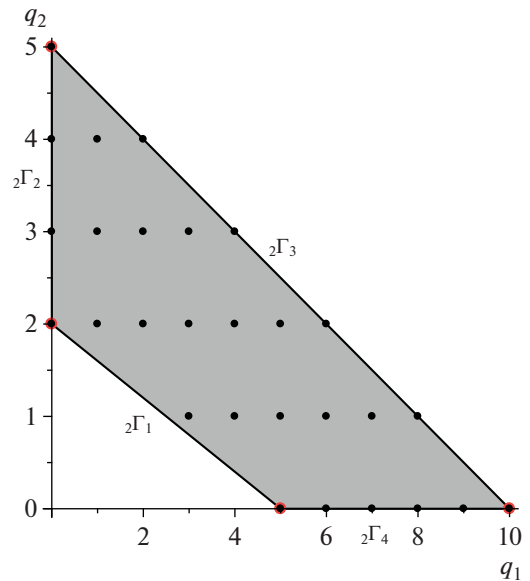


Рис. 3. Многоугольник Ньютона и его ребра для многочлена (4.5).

б) Точка $(y_2, z_2) = (2, 3)$ лежит на прямой $z = 3y/2$, являющейся корнем многочлена (4.2). Эта точка является критической точкой самопересечения.

в) Точка $(y_3, z_3) \approx (2.3427, 3.2427)$ была подробно рассмотрена в статье [1] первых двух авторов, поэтому здесь мы приведем только итоговый результат. Дискриминант квадратичной формы $h_2(y_3 + \eta, z_3 + \zeta)$, вычисленный с большой точностью, равен нулю, а кратный корень этой квадратичной формы не обнуляет кубическую форму $h_3(y_3 + \eta, z_3 + \zeta)$. Следовательно, указанная точка является точкой возврата кривой $f_1(y, z) = 0$, а ее разложение в окрестности особой точки описывается формулой

$$z = 0.823545771 + 1.03264746y + 0.05480984628(2.342744299 - y)^{3/2}$$

при $y \leq y_3 \approx 2.342744299$.

г) В точке $(y_4, z_4) = (14, 27)$ вычисляем (с точностью до постоянного множителя) квадратичную форму h_2 :

$$f_1(y_4 + \eta, z_4 + \zeta) = 243\eta^2 - 256\zeta\eta + 68\zeta^2.$$

Ее дискриминант $\Delta = -560$, следовательно, по лемме 2 критическая точка (y_4, z_4) является изолированной.

д) Ребрам $\Gamma_2^{(1)}$ и $\Gamma_5^{(1)}$ соответствуют укороченные многочлены

$$\hat{h}_2^{(1)} = 8z^2(32z^3 - 16z^2 + 117z - 864),$$

и

$$\hat{h}_5^{(1)} = -27y^2(250y^2 - 467y + 576).$$

Их нули дают координаты пересечения кривой $h = 0$ с осями Oz и Oy соответственно. Многочлен $\hat{h}_2^{(1)}$ имеет два вещественных корня: 0 и 2.7470, а многочлен $\hat{h}_5^{(1)}$ — только корень 0.

е) Ребру $\Gamma_4^{(1)}$ соответствует укороченный многочлен $\hat{h}_4^{(1)} = -250y^4(10z - 27)$. Значит, при $y \rightarrow \infty$ имеем $z \rightarrow 27/10$. Применяя технику из [1], найдем разложение ветви кривой на бесконечности:

$$z = \frac{27}{10} - \frac{243}{2500y}. \quad (4.6)$$

Итак, кривая $h = 0$ имеет горизонтальную асимптоту $z = 27/10$. Этой асимптоте соответствует бесконечная точка пересечения первой кратности.

Стремящейся к ∞ координате z в многоугольнике Γ соответствует верхняя вершина, т.е. при $z = \infty$ нет точек кривой (4.3).

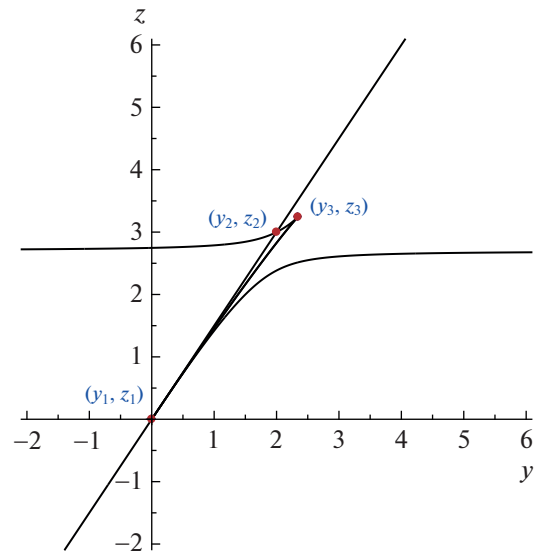


Рис. 4. Линии уровня для критического значения $c_1^* = 0$ многочлена f_1 .

Конусам задач $\mathcal{H}_1$ и $\mathcal{H}_2$ соответствуют только вершины многоугольника Ньютона, рис. 1.

жс) Ребру $\Gamma_3^{(1)}$ соответствует укороченный многочлен

$$\hat{h}_3^{(1)} = 2500y^4z - 5600y^3z^2 + 4736y^2z^3 - 1792yz^4 + 256z^5 = 4z(25y^2 - 28yz + 8z^2)^2,$$

который не имеет нетривиальных вещественных корней. Следовательно, ветвей, соответствующих конусу задачи $\mathcal{H}_1 = \{P : p_1 > 0, p_2 > 0\}$, нет.

На рис. 4 показана сплошная линия уровня $f_1 = 0$, состоящая из сплошной линии $h = 0$ и сплошной прямой (4.2) $3z - 2y = 0$, обе черного цвета. Изолированная критическая точка $(y_4, z_4) = (14, 27)$ не попала на рисунок. Прямой $y = 3z/2$ также соответствует простая бесконечная точка пересечения.

4.3. Критическая линия $f_1(y, z) = c_2^* \approx 0.9355$

Из второго идеала $SP_2(h_1)$ получаем по два действительных и комплексных значения c_j . Рассматривая только вещественные значения, найдем, что критическому значению $c_2^* \approx 0.9355$ соответствует критическая точка $(y_5, z_5) \approx (1.8205, 2.6593)$, координаты которой суть корни многочленов

$$\begin{aligned} &18750y^4 - 285290y^3 + 1217757y^2 - \\ &\quad - 2088009y + 1280664, \\ &8000z^4 - 211120z^3 + 1348356z^2 - \end{aligned}$$

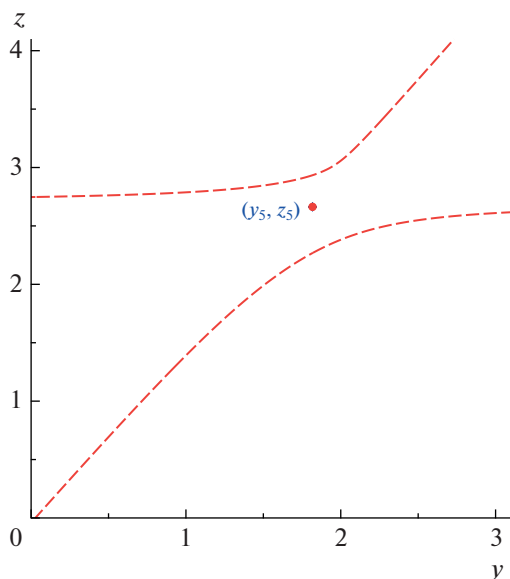


Рис. 5. Линии уровня для критического значения $c_3^* \approx 0.9355$ многочлена f_1 .

$$-3312315z + 2843316. \quad (4.7)$$

Вычисляем квадратичную форму ряда Тейлора многочлена (4.1) в этой точке:

$$f_1(y_5 + \eta, z_5 + \zeta) \approx 0.93558 - 392.0146\eta^2 + 568.6296\zeta\eta - 209.3918\zeta^2,$$

и получаем, что ее дискриминант $\Delta \approx -4998.9489$ отрицательный, следовательно, точка $(1.8205, 2.6593)$ является изолированной критической точкой.

Эта линия пересекается с осями в точках $(0, 0)$ и $(0, 2.747016946)$. Ее асимптотики суть $y = 3/2z$ и (4.6). Сама линия показана на рис. 5 красной пунктирной линией.

4.4. Критическая линия $f_1(y, z) = c_3^* \approx 523029.1004$

Также из идеала $SP_2(h)$ находим, что критическому значению $c_3^* = 523029.1004$ соответствует критическая точка $(y_6, z_6) \approx (9.5641, 18.3933)$, координаты которой суть другие вещественные корни многочленов (4.7). Вычисление квадратичной формы ряда Тейлора многочлена (4.1) в этой точке дает результат:

$$f_1(y_6 + \eta, z_6 + \zeta) \approx 523029.101 + 3.108 \times 10^6 \eta^2 - 3.109 \times 10^6 \eta \zeta + 766031.179 \zeta^2.$$

Видно, что в разложении свободный член близок к критическому значению c_3^* , коэффициенты линейных членов равны нулю, а дискриминант

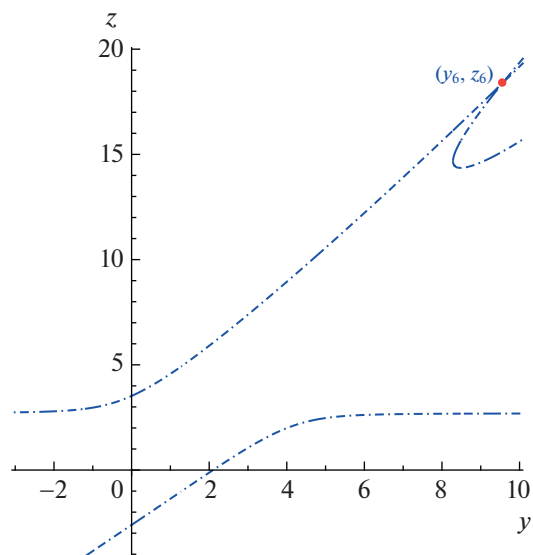


Рис. 6. Линии уровня для критического значения $c_3^* \approx 523029.1004$ многочлена f_1 .

квадратичной формы $\Delta \approx 1.4 \times 10^{11}$ положительный, следовательно, точка (y_6, z_6) является точкой самопересечения.

Эта линия пересекается с осями в точках $(2.0996, 0)$, $(0, -2.6070)$ и $(0, 3.5223)$, ее асимптотики те же, ее эскиз показан на рис. 6 в виде штрихпунктирных синих линий.

4.5. Все критические линии уровня многочлена f_1

На рис. 7 показаны все критические линии уровня многочлена f_1 для соответствующих критических значений c_j^* , а также критические точки. В силу выбранного масштаба, критические точки уровня c_1^* плохо различимы. Подробнее эти точки показаны на рис. 4. На нем в криволинейном треугольнике с вершинами (y_1, z_1) , (y_2, z_2) , (y_3, z_3) лежит точка (y_5, z_5) . Некритические линии уровня лежат между критическими линиями и критическими точками рис. 7.

5. ПРИМЕР 2 ПРИ $y = 0$

5.1. Критические значения

Рассмотрим

$$\begin{aligned} f_2 = & -4x^5z + 40x^4z^2 - 192x^3z^3 + 512x^2z^4 - \\ & - 768xz^5 + 512z^6 + 2x^5 - 64x^4z + 456x^3z^2 - \\ & - 1344x^2z^3 + 1344xz^4 - 256z^5 + 13x^4 - \end{aligned} \quad (5.1)$$

$$-344x^3z + 2200x^2z^2 - 5088xz^3 + 1872z^4 + \\ + 64x^3 - 1152x^2z + 6912xz^2 - 13824z^3.$$

Идеал, определяющий критические точки и кривые и включающий в себя функцию $f_2 - c$ и его первые производные, состоит из следующих полиномов

$$Id_2 = \{f_2 - c, \partial f_2 / \partial x, \partial f_2 / \partial z\}.$$

Для этого идеала строим базис Гребнера $\mathcal{GB}\mathcal{J}_2$ с чистым лексикографическим порядком $c \succ z \succ x$, найденный с помощью процедуры Groebner[Basis]. Первый полином этого базиса есть

$$h_2(c) = c(16c^2 - 17435c + 15925248) \times \\ \times (18225 \times 10^8 c^4 + 219593845975 \times 10^5 c^3 - \\ - 166039069316301172071c^2 + \quad (5.2) \\ + 199547881177990316949504c - \\ - 84569630503397437621665792).$$

Он зависит только от c и определяет критические значения полинома (5.1). Критическое значение $c_1^* = 0$ является корнем множителя $h_{21}(c) = c$. Множитель $h_{22}(c) = 16c^2 - 17435c + 15925248$ не имеет вещественных корней, значит, не выявляет вещественные критические значения c_j . Еще два иррациональных критических значения $c_2^* \approx -17590.60940$ и $c_3^* \approx 4162.403160$ задаются корнями третьего множителя в (5.2).

Идеал $\mathcal{GB}\mathcal{J}_2$ имеет нулевую размерность, его примарная декомпозиция дает три идеала. Можно сразу исключить из рассмотрения идеал $SP_2(h_2)$, содержащий множитель $h_{22}(c) = 16c^2 - 17435c + 15925248$, который не имеет вещественных корней.

5.2. Критическая линия $f_2(x, z) = c_1^* = 0$

Идеал $SP_1(h_2)$, являющийся примарной декомпозицией множителя $h_{21}(c) = c$, включает многочлены

c ,

$$262144z^{14} + 3178496z^{13} + 17618944z^{12} + \\ + 53712704z^{11} + 83353369z^{10} + 42910541z^9 - \\ - 34250732z^8 + 92661536z^7 + \\ + 44242816z^6 - 44658432z^5 + 36578304z^4, \\ - 2687921593523634176z^{13} - \\ - 34906091624775188480z^{12} -$$

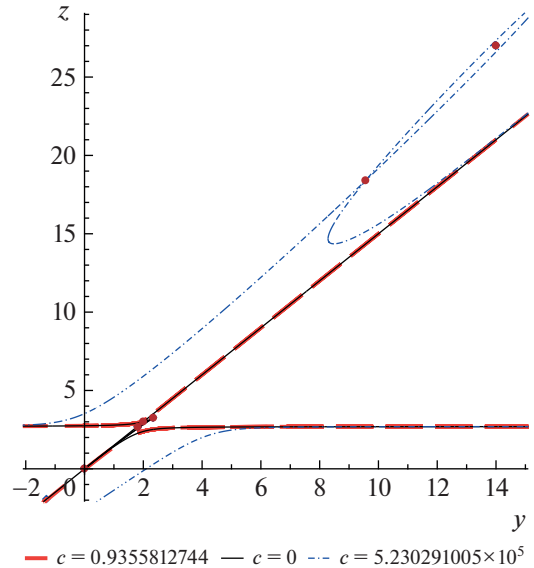


Рис. 7. Все критические линии уровня многочлена f_1 .

$$-209646836565166307328z^{11} - \\ - 717809757784574929984z^{10} - \\ - 1395096902560948574269z^9 - \\ - 1391000168810966419626z^8 - \\ - 414219249823112818193z^7 - \\ - 989978918218726674788z^6 - \\ - 1359995427547542403776z^5 + \\ + 44989667987413693824xz^3 - \\ - 427910073600905300160z^4 + \\ + 44989667987413693824xz^2 - \\ - 179958671949654775296z^3, \\ - 21973906577585552490496z^{13} - \\ - 274219691042743274733568z^{12} - \\ - 1569796991169618160588800z^{11} - \\ - 5004433077233995232707904z^{10} - \\ - 8441114957318849630660081z^9 - \\ - 5521961458811860727741337z^8 + \\ + 2910068553190244608309880z^7 - \\ - 4856586515216271331078816z^6 -$$

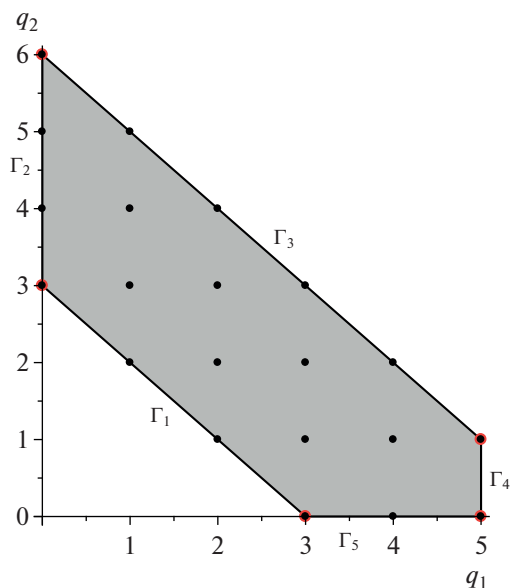


Рис. 8. Многоугольник Ньютона для многочлена f_2 .

$$\begin{aligned}
 & -4913807691571657866494208z^5 + \\
 & +3773420647566644086822656z^4 - \\
 & -70543799404264671916032xz^2 + \\
 & +293932497517769466316800z^3 + \\
 & +17635949851066167979008x^2 - \\
 & -211631398212794015748096xz + \\
 & +634894194638382047244288z^2.
 \end{aligned}$$

а) Корень его первого полинома — вещественное число $c_1^* = 0$. Среди корней полиномов идеала $SP_1(h_2)$ ищем вещественные значения. Таковыми являются две критические точки $(0, 0)$ и $(-8, -4)$.

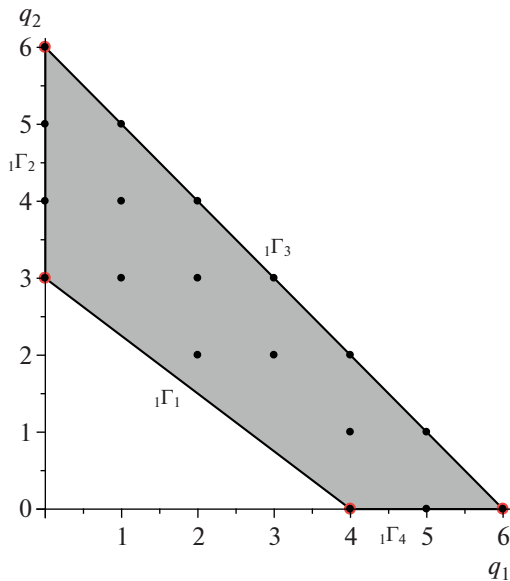
б) Рассмотрим ВНФ (2.7) или (2.8) вещественных критических точек. При $c_1 = 0$ в критической точке $(0, 0)$ квадратичная форма h_2 ряда Тейлора многочлена (5.1) обнуляется и теорема о неявной функции здесь не применима, поэтому применяем анализ особой точки согласно [1, п. 5.2]. Вычисляем выпуклую оболочку носителя многочлена (5.1) с использованием пакета **Polyhedral-Sets**. Носитель и многоугольник Ньютона показаны на рис. 8. Ребру $\Gamma_1^{(1)}$ соответствует укороченный многочлен $\hat{f}_1^{(1)} = 64(x - 6z)^3$. Выполним подстановку $z = x/6 + z_1$ и получим многочлен в переменных x и z_1 :

$$\begin{aligned}
 {}_1f_2 = & \frac{100}{729}x^6 - 13824z_1^3 + 1872z_1^4 - 256z_1^5 + 512z_1^6 + \\
 & + \frac{20}{81}x^5z_1 - \frac{8}{27}x^4z_1^2 - \frac{448}{27}x^3z_1^3 - 32x^2z_1^2 - 3840xz_1^3 + \\
 & + \frac{256}{3}x^2z_1^4 - \frac{8}{81}x^4z_1 - \frac{104}{27}x^3z_1^2 - \frac{4672}{9}x^2z_1^3 + \\
 & + \frac{3392}{3}xz_1^4 - 256xz_1^5 - \frac{16}{3}x^4 - \frac{296}{243}x^5.
 \end{aligned}$$

Его многоугольник Ньютона показан на рис. 9.

Единственному ребру $\Gamma_1^{(1)}$ многоугольника Ньютона, нормальный конус которого попадает в конус нашей задачи $K = \{P < 0\}$, соответствует укороченный многочлен $\hat{f}_1^{(1)} = 13824z_1^3 - 16x^4/3$. Следовательно, здесь требуется еще одна подстановка $z_1 = 18^{1/3}x^{4/3}/36 + z_2$. Она приводит к еще одному многочлену от переменных x , z_2 , который содержит слагаемые с дробными степенями. Дабы избежать больших объемов вычислений, сразу произведем замену вида $x = x_1^3$ и получим более удобный вид многочлена для вычислений:

$$\begin{aligned}
 {}_2f_2 = & 1152 \times 18^{1/3}x_1^4z_2^2 + \frac{26}{3} \times 18^{2/3}x_1^8z_2^2 - \\
 & - 208 \times 18^{1/3}x_1^4z_2^3 + \frac{1168}{27} \times 18^{1/3}x_1^{10}z_2^2 + \\
 & + \frac{16 \times 18^{1/3}x_1^{10}z_2}{9} - \frac{292 \times 18^{2/3}x_1^{14}z_2}{243} - \\
 & - \frac{160 \times 18^{2/3}x_1^8z_2^3}{81} - 32 \times 18^{2/3}x_1^8z_2 - \frac{256 \times 18^{1/3}x_1^4z_2^5}{3} + \\
 & + \frac{320 \times 18^{1/3}x_1^4z_2^4}{9} + \frac{160 \times 18^{2/3}x_1^8z_2^4}{27} + \frac{2 \times 18^{1/3}x_1^{22}}{2187} + \\
 & + \frac{448x_1^9z_2^3}{27} + \frac{256x_1^6z_2^4}{3} + 256x_1^3z_2^5 - \frac{242x_1^{12}z_2}{81} + \\
 & + \frac{104x_1^9z_2^2}{27} - \frac{4672x_1^6z_2^3}{9} - \frac{3392x_1^4z_2^4}{3} - 32x_1^6z_2^2 + \\
 & + 3840x_1^3z_2^3 - 256z_2^5 + 512z_2^6 - 13824z_2^3 + \\
 & + 1872z_2^4 - \frac{2 \times 18^{2/3}x_1^{14}}{81} + \frac{133 \times 18^{1/3}x_1^{16}}{5832} - \frac{18^{2/3}x_1^{20}}{6561} - \\
 & - \frac{23 \times 18^{1/3}x_1^{19}}{4374} - \frac{18^{2/3}x_1^{23}}{13122} + \frac{13 \times 18^{2/3}x_1^{17}}{4374} - \frac{32x_1^{18}z_2}{243} + \\
 & + \frac{20 \times 18^{1/3}x_1^{16}z_2^2}{243} + \frac{2 \times 18^{1/3}x_1^{16}z_2}{729} - \frac{2 \times 18^{2/3}x_1^{20}z_2}{2187} - \\
 & - \frac{256 \times 18^{1/3}x_1^{10}z_2^3}{27} + \frac{32 \times 18^{2/3}x_1^{14}z_2^2}{81} - \frac{80x_1^{15}z_2^2}{81} -
 \end{aligned}$$

Рис. 9. Многоугольник Ньютона для многочлена $1f_2$.

$$\begin{aligned}
 & -\frac{320x_1^{12}z_2^3}{81} + \frac{364x_1^{15}z_2}{243} + \frac{56x_1^{12}z_2^2}{81} + \frac{x_1^{24}}{13122} - \frac{14x_1^{21}}{2187} + \\
 & + \frac{46x_1^{18}}{729} - \frac{64x_1^{15}}{243} - 320 \times 18^{1/3} x_1^7 z_2^2 + \\
 & + \frac{3392 \times 18^{1/3} x_1^7 z_2^3}{27} + \frac{160 \times 18^{2/3} x_1^{11} z_2^3}{81} + \\
 & + \frac{28 \times 18^{2/3} x_1^{17} z_2}{729} - \frac{52 \times 18^{1/3} x_1^{13} z_2}{243} - \frac{112 \times 18^{1/3} x_1^{13} z_2^2}{81} - \\
 & - \frac{320 \times 18^{1/3} x_1^7 z_2^4}{9} + \frac{80 \times 18^{2/3} x_1^{11} z_2}{9} + \frac{10 \times 18^{1/3} x_1^{19} z_2}{729} - \\
 & - \frac{424 \times 18^{2/3} x_1^{11} z_2^2}{81}.
 \end{aligned}$$

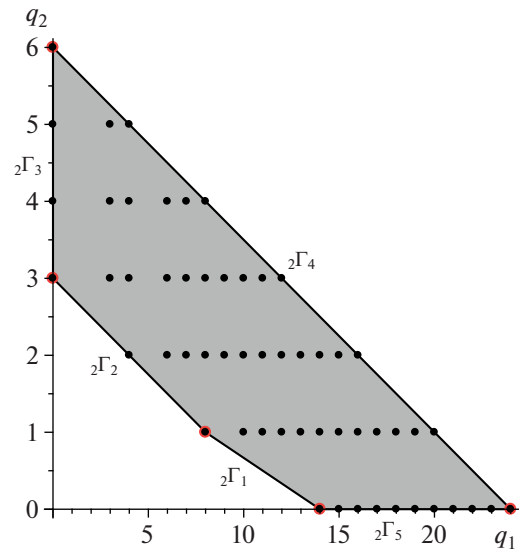
Его многоугольник Ньютона (см. рис. 10) имеет два ребра $2\Gamma_1^{(1)}$ и $2\Gamma_2^{(1)}$ из конуса нашей задачи, которым соответствуют следующие укороченные многочлены:

$$\begin{aligned}
 2\hat{f}_1^{(1)} &= \frac{2 \times 18^{2/3} (x_1^6 + 1296z_2)x_1^8}{81}, \\
 2\hat{f}_2^{(1)} &= 32 \times 18^{2/3} (-x_1^8 + 2 \times 18^{2/3} z_2 x_1^4 - 24 \times 18^{1/3} z_2^2) z_2.
 \end{aligned}$$

Решением первого укороченного уравнения будет $z_2 = -x_1^6/1296$. Выполняя обратную подстановку как для z , так и для x , получаем разложение ветви вблизи точки $(0, 0)$.

$$z = \frac{x}{6} - \frac{18^{1/3} (-x)^{4/3}}{36} - \frac{x^2}{1296}.$$

Поскольку соответствующее ребро содержит точку с ординатой 1, то здесь применима теорема

Рис. 10. Многоугольник Ньютона для многочлена $2f_2$.

о неявной функции, которая дает единственную ветвь.

Укороченный многочлен $2\hat{f}_2^{(1)}$, соответствующий второму ребру, не имеет нетривиальных вещественных корней, следовательно, ему не соответствует вещественная ветвь. Вычисления с использованием функции `puiseux` из пакета `algsurfaces` системы Maple приводят к таким же результатам.

б) Также критическому значению $c_1^* = 0$ соответствует критическая точка $(x_2, z_2) = (-8, -4)$. Вычисляем квадратичную форму ряда Тейлора многочлена (5.1) в этой точке, используя команду `mtaylor`:

$$h_2(x_2 + \xi, z_2 + \zeta) = 9216\xi^2 - 28672\xi\zeta + 36864\zeta^2.$$

Дискриминант данной квадратичной формы $\Delta = -536870912 < 0$, следовательно, точка $(-8, -4)$ является изолированной критической точкой.

Для анализа критических линий на бесконечности проанализируем многоугольник Ньютона (рис. 8) многочлена (5.1) и выделим укороченные многочлены, соответствующие ребрам, чьи нормали имеют хотя бы одну положительную координату.

Из этого многоугольника Ньютона можно видеть, что два ребра соответствуют бесконечности $\Gamma_3^{(1)}$ и $\Gamma_4^{(1)}$.

Рассмотрим ребро $\Gamma_3^{(1)}$, которому соответствует укорочение

$$\hat{f}_3^{(1)} = -4z(-2z + x)(x^2 - 4xz + 8z^2)^2.$$

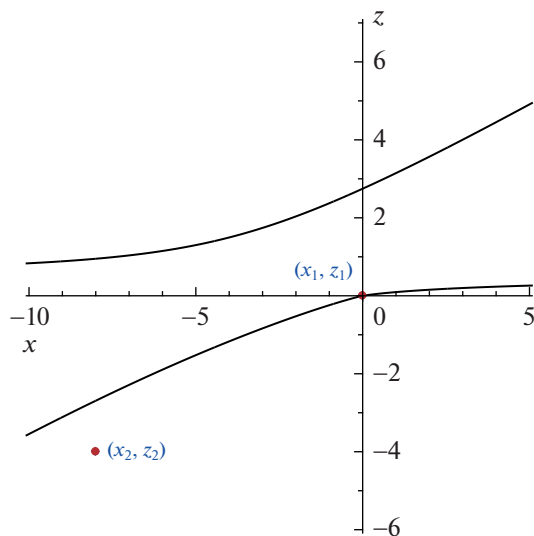


Рис. 11. Линии уровня для критического значения c_1^* многочлена f_2 .

По нему можно определить первый член разложения $z = x/2 + z_1$. Произведя такую замену в (5.1), получаем

$$\begin{aligned} {}_3f_2 = & 4x^5z_1 + 40x^4z_1^2 + 192x^3z_1^3 + 512x^2z_1^4 + \\ & + 768xz_1^5 + 512z_1^6 - 8x^5 - 24x^4z_1 + 136x^3z_1^2 + \\ & + 704x^2z_1^3 + 704xz_1^4 - 256z_1^5 - 128x^4 - 1024x^3z_1 - \\ & - 2624x^2z_1^2 - 1344xz_1^3 + 1872z_1^4 - 512x^3 - \\ & - 4608x^2z_1 - 13824xz_1^2 - 13824z_1^3. \end{aligned}$$

В выпуклой оболочке многочлена выбираем соответствующее бесконечности ребро, тогда его укороченный многочлен будет выглядеть следующим образом: ${}_1\hat{f}_3^{(1)} = 4x^5(z_1 - 2)$. Также по этому укорочению определяется второй член разложения: $z_1 = 2 + z_2$. Продолжая аналогично, выявляем первую ветвь разложения в бесконечности

$$z = \frac{x}{2} + 2 + \frac{4}{x} \quad (5.3)$$

Аналогично рассмотрим ребро $\Gamma_4^{(1)}$, ему соответствует укороченный многочлен $\hat{f}_4^{(1)} = -2x^5(2z - 1)$. Значит при $x \rightarrow \infty$ имеем $z \rightarrow 1/2$. Произведем замену $z = 1/2 + z_1$ в многочлене (5.1), получаем

$$\begin{aligned} {}_4f_2 = & -4x^5z_1 + 40x^4z_1^2 - 192x^3z_1^3 + 512x^2z_1^4 - \\ & - 768xz_1^5 + 512z_1^6 - 24x^4z_1 + 168x^3z_1^2 - \\ & - 320x^2z_1^3 - 576xz_1^4 + 1280z_1^5 - 9x^4 - 32x^3z_1 + \end{aligned}$$

$$\begin{aligned} & + 952x^2z_1^2 - 4320xz_1^3 + 3152z_1^4 - 18x^3 + 296x^2z_1 + \\ & + 336xz_1^2 - 9440z_1^3 - 162x^2 + 3528xz_1 - \\ & - 17768z_1^2 + 1152x - 9416z_1 - 1611. \end{aligned}$$

У его многоугольника Ньютона возьмем единственное ребро, с нормальным конусом, попадающим в конус нашей задачи. Соответствующее укорочение выглядит так: ${}_4\hat{f}_2^{(1)} = -x^4(4xz_1 + 9)$. Для определения 3-го члена разложения произведем еще одну замену вида: $z_1 = -9/(4x) + z_2$. После очередных вычислений соответствующий укороченный многочлен будет иметь вид ${}_5\hat{f}_2^{(1)} = -4x^5z_2 + 36x^3$. Таким образом, разложение второй ветви в бесконечности выглядит так:

$$z = \frac{1}{2} - \frac{9}{4x} + \frac{9}{x^2}. \quad (5.4)$$

Вычисленные выше разложения ветвей совпадают с результатами использования функции `puiseux` из пакета `algsurves` системы Maple.

Корни укороченных многочленов

$$\begin{aligned} \hat{f}_2^{(1)} &= 16z^3(32z^3 - 16z^2 + 117z - 864), \\ \hat{f}_5^{(1)} &= x^3(2x^2 + 13x + 64), \end{aligned}$$

соответствующих ребрам $\Gamma_2^{(1)}$ и $\Gamma_5^{(1)}$ многоугольника Ньютона многочлена (5.1), дают точки пересечения с осями координат. Пересечение с Oz при $z = 0$ и $z \approx 2.747016946$, а с осью Ox только при $x = 0$.

Для изображения этих линий уровня на плоскости (рис. 11) воспользуемся командой `plot_real_curve` пакета `algsurves`. На этом рисунке критическая линия обозначена сплошной линией черного цвета. Асимптотики (5.3) и (5.4) соответствуют простым бесконечным точкам пересечения. Поэтому здесь работает лемма 4.

5.3. Критическая линия $f_2(x, z) = c_2^* \approx -17590.6094$

Первый из двух наборов вещественных решений уравнений, входящих в идеал SP_3 , есть критическое значение $c_2^* \approx -17590.6094$ и соответствующая критическая точка с иррациональными координатами $(x_3, z_3) \approx (-4.2020, 0.9946)$. Рассмотрим эту точку, координаты которой являются корнями многочленов 4-й степени. Возьмем их значения приблизительно с большой точностью. Найдём квадратичную форму ряда Тейлора в точке (x_3, z_3) :

$$\begin{aligned} h_2(x_3 + \xi, z_3 + \zeta) = & -17590.6094 + 704.2988\xi^2 - \\ & - 21546.0609\xi\zeta + 50690.9070\zeta^2. \end{aligned}$$

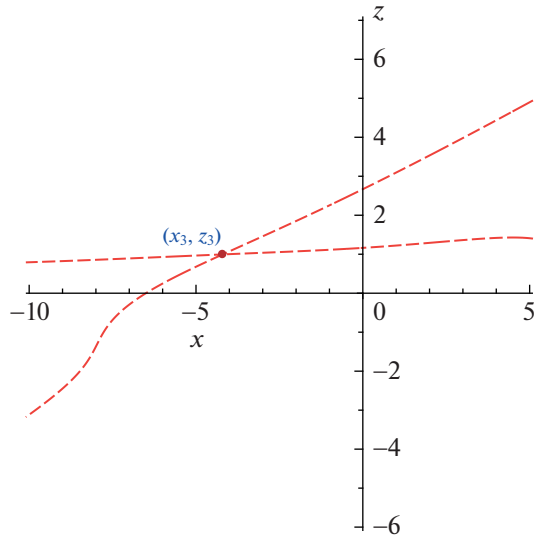


Рис. 12. Линии уровня для критического значения $c = -17590.6094$ многочлена f_2 .

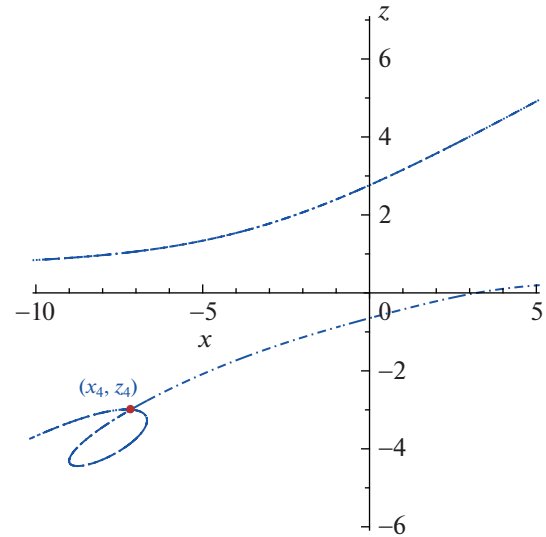


Рис. 13. Линии уровня для критического значения $c^* = 4162.4031$ многочлена f_2 .

Здесь свободный член близок к c_2^* , коэффициенты линейных членов равны нулю, а дискриминант квадратичной формы $\Delta \approx 3.2142 \times 10^8$ положительный, следовательно, эта точка по лемме 2 является точкой самопересечения (в ней пересекаются две ветви). Критические линии уровня при c_2^* пересекают ось Ox в точке $(x, z) \approx (-6.5012, 0)$, а ось Oz в точках $(x, z) \approx (0, 1.1627)$ и $(x, z) \approx (0, 2.6681)$. Здесь верны асимптотики (5.3) и (5.4). Эскиз этой критической линии показан на рис. 12 штриховой красной линией.

5.4. Критическая линия $f_2(x, z) = c_3^* \approx 4162.4031$

Второй из двух наборов вещественных решений уравнений, входящих в идеал SP_3 , есть критическое значение $c_3 \approx 4162.4031$ и соответствующая критическая точка с иррациональными координатами $(x_4, z_4) \approx (-7.1522, -2.9929)$. Рассмотрим квадратичную форму разложения ряда Тейлора исходного многочлена (5.1) в критической точке (x_4, z_4) :

$$h_2(x_4 + \xi, z_4 + \zeta) = 4162.4022 + 350.7520\xi^2 + 4785.9994\xi\zeta - 11228.6467\zeta^2.$$

Здесь свободный член близок к c_3^* , а ее дискриминант равен $\Delta \approx 3.8659 \times 10^7 > 0$. Следовательно это точка самопересечения.

Критические линии уровня пересекают ось Ox в точке $(x, z) \approx (3.206553809, 0)$, а ось Oz в точках $(x, z) \approx (0, -0.6481)$ и $(x, z) \approx (0, 2.7632)$. Эта кри-

тическая линия показана на рис. 13 штрих-пунктирной линией синего цвета.

5.5. Все критические линии уровня многочлена f_2

На рис. 14 показаны все критические линии уровня многочлена f_2 для соответствующих критических значений c_j^* , а также критические точки. Некритические линии уровня расположены между соответствующими критическими согласно (2.3).

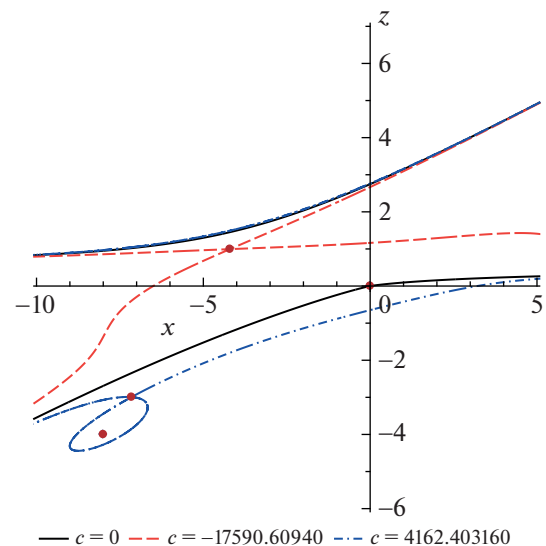


Рис. 14. Все критические линии уровня многочлена f_2 .

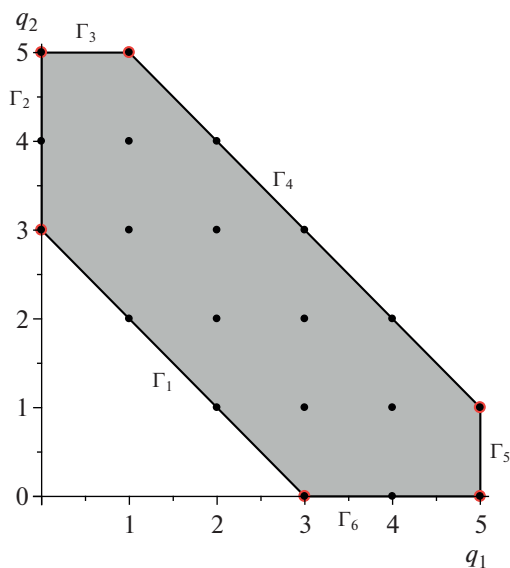


Рис. 15. Многоугольник Ньютона для многочлена f_3 .

6. ПРИМЕР 3 ПРИ $z = 0$

6.1. Критические значения

Рассмотрим последнее нулевое сечение:

$$f_3 = 4x^5y + 48x^4y^2 + 344x^3y^3 + 1200x^2y^4 + 2500xy^5 + 2x^5 + 82x^4y + 740x^3y^2 + 3156x^2y^3 + 6490xy^4 + 20250y^5 + 13x^4 + 412x^3y + 2782x^2y^2 + 828xy^3 - 37827y^4 + 64x^3 + 1728x^2y + 15552xy^2 + 46656y^3. \quad (6.1)$$

Первый полином базиса Гребнера $\mathcal{GB}f_3$ есть

$$h_3(c) = c(1738071441650390625c^6 + 8280321886721438116 \times 10^5 c^5 + 13366043451422095808794624c^4 + 226388815999145527309322158080c^3 + 33335217445009148607411951501312c^2 + 14684298419139851546639897167134720c - 876052122097662955869886609920884736). \quad (6.2)$$

Он зависит только от c , факторизуется на два множителя и определяет критические значения полинома (6.1). Критическое значение $c_1^* = 0$ является корнем множителя $h_{11}(c) = c$. Второй множитель в (6.2) имеет только два вещественных критических значения $c_2^* \approx -476393.0349992$ и $c_3^* \approx 51.5040463$.

6.2. Критическая линия $f_3(x, y) = c_1^* = 0$

Идеал $\mathcal{GB}f_3$ имеет нулевую размерность, его примарная декомпозиция дает два идеала. Рассмотрим идеал $SP_1(h_3)$, являющийся примарной декомпозицией множителя $h_{31}(c) = c$, выявляет критическое значение $c_1^* = 0$. Корни многочленов, входящих в идеал $SP_1(h_3)$, определяют критические точки $(x_1, y_1) = (0, 0)$ и $(x_2, y_2) = (-9/2, 3/2)$.

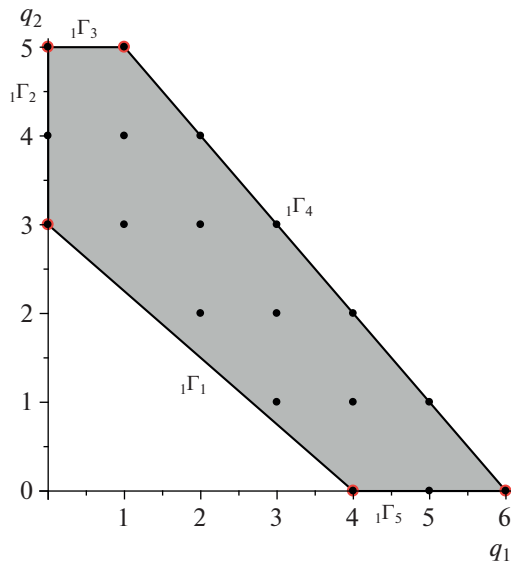
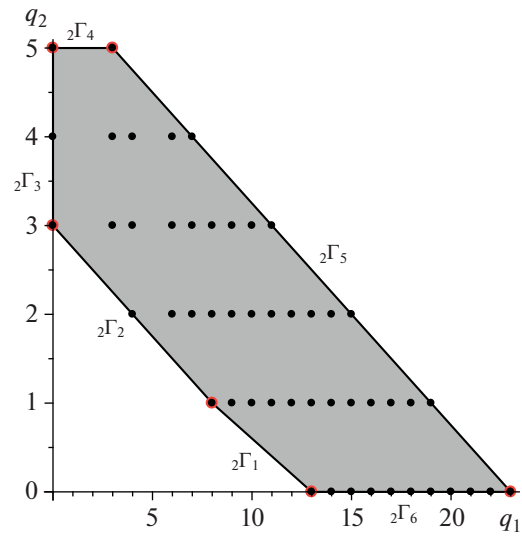
а) Рассмотрим ВНФ (2.7) или (2.8) вещественных критических точек. При $c_1^* = 0$ в критической точке $(0, 0)$ квадратичная форма h_2 ряда Тейлора многочлена (6.1) обнуляется и теорема о неявной функции здесь не применима, поэтому применяем анализ особой точки. Вычисляем выпуклую оболочку носителя многочлена (6.1). Многоугольник Ньютона показан на рис. 15.

Ребру $\Gamma_1^{(1)}$, внешняя нормаль которого входит в конус нашей задачи $K = \{P < 0\}$, соответствует укороченный многочлен $\hat{f}_1^{(1)} = 64(x + 9y)^3$. Выполним подстановку $y = -x/9 + y_1$ и получим многочлен в переменных x и y_1 :

$$\begin{aligned} {}_1f_3 = & \frac{10816}{59049}x^6 + 46656y_1^3 - 37827y_1^4 + 20250y_1^5 + \\ & + \frac{9152}{6561}x^5y_1 - \frac{8800}{729}x^4y_1^2 + \frac{9664}{81}x^3y_1^3 + 32x^3y_1 - \\ & - 296x^2y_1^2 + 17640xy_1^3 - \frac{1700}{9}x^2y_1^4 + \frac{10400}{729}x^4y_1 - \\ & - \frac{2944}{27}x^3y_1^2 + \frac{24944}{9}x^2y_1^3 - 4760xy_1^4 + 2500xy_1^5 - \\ & - \frac{16}{3}x^4 - \frac{10880}{6561}x^5. \end{aligned}$$

Его многоугольник Ньютона показан на рис. 16. Его единственному ребру $\Gamma_1^{(1)}$ с нормальным конусом в конусе нашей задачи $K = \{P < 0\}$, соответствует укороченный многочлен ${}_1\hat{f}_1^{(1)} = 16x^4/3 + 46656y_1^3$. Следовательно, здесь требуется еще одна подстановка $y_1 = 18^{1/3}x^{4/3}/54 + y_2$. Она приводит к еще одному многочлену от переменных x , y_2 . Но при подстановке выявляются дробные степени переменной x , что в свою очередь приводит к некоторым трудностям. Чтобы упростить дальнейшие действия, тут же произведем еще одну замену вида $x = x_1^3$. Так получаем многочлен

$$\begin{aligned} {}_2f_3 = & 46656y_2^3 - 37827y_2^4 + 20250y_2^5 - \frac{1700x_1^{18}y_2}{19683} + \\ & + \frac{6250x_1^{15}y_2^2}{2187} - \frac{5128x_1^{15}y_2}{6561} + \frac{8075x_1^{12}y_2^2}{729} + \frac{9664x_1^9y_2^3}{81} - \end{aligned}$$

Рис. 16. Многоугольник Ньютона для многочлена $1f_3$.Рис. 17. Многоугольник Ньютона для многочлена $2f_3$.

$$\begin{aligned}
 & -\frac{1700x_1^6y_2^4}{9} + 2500x_1^3y_2^5 - \frac{2209x_1^{12}y_2}{729} - \frac{2944x_1^9y_2^2}{27} + \\
 & + \frac{24944x_1^6y_2^3}{9} - 4760x_1^3y_2^4 + 32x_1^9y_2 - 296x_1^6y_2^2 + \\
 & + 17640x_1^3y_2^3 + \frac{2416x_1^{21}}{177147} + \frac{7892x_1^{18}}{59049} + \frac{2350x_1^{15}}{6561} + \\
 & + \frac{16 \times 18^{\frac{1}{3}}x_1^{13}}{27} - \frac{736 \times 18^{\frac{2}{3}}x_1^{17}}{19683} + \frac{28991 \times 18^{\frac{1}{3}}x_1^{16}}{157464} - \\
 & - \frac{74 \times 18^{\frac{2}{3}}x_1^{14}}{729} + \frac{2791 \times 18^{\frac{1}{3}}x_1^{19}}{177147} - \frac{425 \times 18^{\frac{1}{3}}x_1^{22}}{1062882} - \\
 & - \frac{14225 \times 18^{\frac{2}{3}}x_1^{20}}{4251528} + \frac{625 \times 18^{\frac{2}{3}}x_1^{23}}{6377292} + 2592 \times 18^{\frac{1}{3}}x_1^4y_2^2 - \\
 & - 2802 \times 18^{\frac{1}{3}}x_1^4y_2^3 + 48 \times 18^{\frac{2}{3}}x_1^8y_2 - \frac{296 \times 18^{\frac{1}{3}}x_1^{10}y_2}{27} + \\
 & + 980 \times 18^{\frac{1}{3}}x_1^7y_2^2 + \frac{490 \times 18^{\frac{2}{3}}x_1^{11}y_2}{27} - \frac{467 \times 18^{\frac{2}{3}}x_1^8y_2^2}{6} - \\
 & - \frac{9520 \times 18^{\frac{1}{3}}x_1^7y_2^3}{27} + 1875 \times 18^{\frac{1}{3}}x_1^4y_2^4 + \frac{12472 \times 18^{\frac{1}{3}}x_1^{10}y_2^2}{81} + \\
 & + \frac{625 \times 18^{\frac{2}{3}}x_1^8y_2^3}{9} - \frac{2944 \times 18^{\frac{1}{3}}x_1^{13}y_2}{729} + \frac{6236 \times 18^{\frac{2}{3}}x_1^{14}y_2}{2187} -
 \end{aligned}$$

$$\begin{aligned}
 & -\frac{2380 \times 18^{\frac{2}{3}}x_1^{11}y_2^2}{243} + \frac{6250 \times 18^{\frac{1}{3}}x_1^7y_2^4}{27} + \frac{4832 \times 18^{\frac{1}{3}}x_1^{13}y_2^2}{729} - \\
 & - \frac{3400 \times 18^{\frac{1}{3}}x_1^{10}y_2^3}{243} - \frac{18325 \times 18^{\frac{1}{3}}x_1^{16}y_2}{78732} - \frac{850 \times 18^{\frac{2}{3}}x_1^{14}y_2^2}{2187} + \\
 & + \frac{6250 \times 18^{\frac{2}{3}}x_1^{11}y_2^3}{729} + \frac{2416 \times 18^{\frac{2}{3}}x_1^{17}y_2}{19683} + \frac{3125 \times 18^{\frac{1}{3}}x_1^{19}y_2}{118098}.
 \end{aligned}$$

Его многоугольник Ньютона показан на рис. 17, и имеет два ребра $2\Gamma_1^{(1)}$ и $2\Gamma_2^{(1)}$, с отрицательными внешними нормальными, которым соответствует следующие укороченные многочлены

$$2\hat{f}_1^{(1)} = \frac{16 \times 18^{1/3}(x_1^5 + 81 \times 18^{1/3}y_2)x_1^8}{27}$$

$$2\hat{f}_2^{(1)} = 48 \times 18^{2/3}(x_1^8 + 3 \times 18^{2/3}y_2x_1^4 + 54 \times 18^{1/3}y_2^2)y_2.$$

Для $2\hat{f}_1^{(1)}$ корнем будет $y_2 = -18^{2/3}x_1^5/1458$. Выполняя обратную подстановку по x и y , получаем разложение ветви вблизи точки $(0, 0)$

$$y = -\frac{x}{9} + \frac{18^{1/3}x^{4/3}}{54} - \frac{8748^{1/3}x^{5/3}}{4374}.$$

Поскольку первое ребро имеет верхнюю точку с ординатой 1, то здесь применима теорема о неявной функции, которая дает единственную ветвь.

Второе укорочение $2\hat{f}_2^{(1)}$ не имеет нетривиальных вещественных корней, следовательно, ему не соответствует вещественная ветвь.

б) Также критическому значению $c_1^* = 0$ соответствует критическая точка $(x_2, y_2) = (-9/2, 3/2)$. Вычисляем квадратичную форму ряда Тейлора

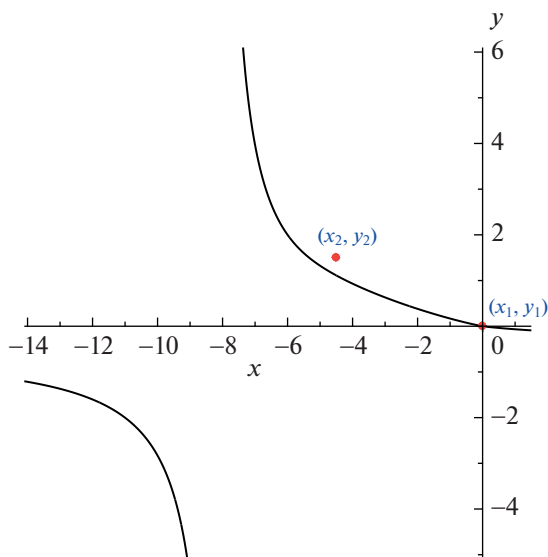


Рис. 18. Линии уровня для критического значения $c_1^* = 0$ многочлена f_3 .

многочлена (6.1) в этой точке с использованием команды `mtaylor`:

$$h_2(x_2 + \xi, y_2 + \eta) = 576\xi^2 + 1728\xi\eta + 5184\eta^2.$$

Дискриминант $\Delta = -8957952$ отрицательный, следовательно, точка (x_2, y_2) является изолированной критической точкой.

в) Для анализа критических линий на бесконечности рассмотрим многоугольник Ньютона (рис. 15) многочлена (6.1). Выделим укороченные многочлены, соответствующие ребрам, чьи нормали имеют хотя бы одну положительную координату. Из этого многоугольника Ньютона можно видеть, что таковых три: $\Gamma_3^{(1)}$, $\Gamma_4^{(1)}$ и $\Gamma_5^{(1)}$.

з) Рассмотрим ребро $\Gamma_3^{(1)}$, ему соответствует укороченный многочлен $\hat{f}_3^{(1)} = 250y^5(10x + 81)$. Значит, при $y \rightarrow \infty$ имеем $x \rightarrow -81/10$. Произведем замену $x = -81/10 + x_1$ в многочлене и возьмем соответствующее ребро, попадающее в конус нашей задачи. Соответствующее укорочение выглядит так: ${}_3\hat{f}_2^{(1)} = 4y^4(625yx_1 - 2916)$. Для определения 2-го члена разложения произведем еще одну замену вида: $x_1 = 2916/625y + x_2$. После очередных вычислений соответствующий укороченный многочлен будет иметь вид

$${}_4\hat{f}_3^{(1)} = 2500y^5x_2 + \frac{472392}{125}y^3.$$

Таким образом, разложение первой ветви в бесконечности выглядит так:

$$x \approx -\frac{81}{10} + \frac{2916}{625y} - \frac{118098}{78125y^2}.$$

Ей соответствует простая бесконечная точка пересечения.

д) Ребру $\Gamma_4^{(1)}$ соответствует укороченный многочлен $\hat{f}_4^{(1)} = 4xy(x^2 + 6xy + 25y^2)^2$, который не имеет нетривиальных вещественных корней. Следовательно, ветвей в бесконечности не имеет.

е) Аналогично рассмотрим ребро $\Gamma_5^{(1)}$, которому соответствует укорочение $\hat{f}_5^{(1)} = 2x^5(2y + 1)$. По нему можно определить, что первая замена будет $y = -1/2 + y_1$, а также, что при $x \rightarrow \infty$ имеем горизонтальную асимптоту $y = -1/2$. Произведя такую замену, в многоугольнике Ньютона полученного многочлена выбираем соответствующее бесконечности ребро, его укороченный многочлен есть: ${}_1\hat{f}_5^{(1)} = 4x^4(xy_1 - 4)$. Также по этому укорочению определяется второй член разложения: $y_1 = 4/x + y_2$. Продолжая аналогично, выявляем очередное укорочение ${}_2\hat{f}_5^{(1)} = 4x^5y_2 + 136x^3$ и, соответственно, вторую ветвь разложения в бесконечности

$$y \approx -\frac{1}{2} + \frac{4}{x} - \frac{34}{x^2}.$$

Ей также соответствует простая бесконечная точка пересечения.

Пересечение с осями координат можно определить по ребрам $\Gamma_2^{(1)}$ и $\Gamma_6^{(1)}$, таковым и единственным для осей Ox и Oy является точка $(0, 0)$. Вычисленные выше разложения ветвей совпадают с результатами использования функции `puiseux` из пакета `algsurves` системы `Maple`.

Теперь изобразим линии уровня на плоскости. Для этого воспользуемся командой `plot_real_curve` пакета `algsurves`. Критическую линию уровня и точки для $c_1^* = 0$ см. на рис. 18, критическая линия обозначена сплошной линией черного цвета.

6.3. Критическая линия

$$f_3(x, y) = c_2^* \approx -476393.0349$$

При этом c_2^* имеется только одна критическая точка $(x_3, y_3) \approx (-11.3777, -1.0391)$. Ее координаты являются первым набором корней многочленов 6-й степени:

$$\begin{aligned} &12000x^6 + 376640x^5 + 4789184x^4 + \\ &+ 32751568x^3 + 128618398x^2 + 276229080x + \\ &+ 255518145, \end{aligned} \quad (6.3)$$

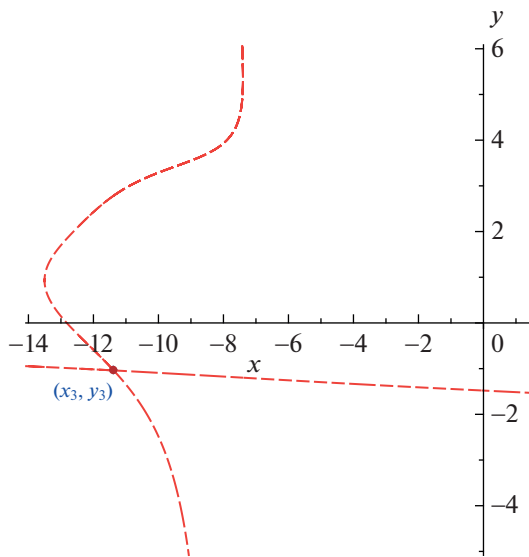


Рис. 19. Линии уровня для критического значения $c_2^* \approx -476393.035$ многочлена f_3 .

$$3 \times 10^5 y^6 - 607040 y^5 + 56704 y^4 + 505904 y^3 - 403522 y^2 + 143976 y - 26487,$$

которые получаются при примарной декомпозиции идеала $SP_2(h_3)$, включающего множитель h_{32} , при лексикографических порядках $c \succ y \succ x$ и $c \succ x \succ y$ соответственно. Корни многочленов (6.3) суть иррациональные числа, поэтому также возьмем их приближительные значения с большой точностью. Найдем квадратичную форму ряда Тейлора в точке (x_3, y_3) :

$$h_2(x_3 + \xi, y_3 + \eta) = -476393.0342 + 373296.3064\eta\xi + 13334.3035\xi^2 + 462406.3700\eta^2.$$

Здесь постоянная близка к c_2^* , коэффициенты линейных членов равны нулю, а дискриминант этой формы $\Delta \approx 1.1468 \times 10^{11}$ положительный. Следовательно, эта точка по лемме 2 является точкой самопересечения (в ней пересекаются две ветви). Также критическая линия уровня при c_2^* пересекает ось Oy при $y \approx -1.4798$ и ось Ox при $x \approx -12.82060973$. Она показана на рис. 19 штриховой линией красного цвета.

6.4. Критическая линия $f_3(x, y) = c_3^* \approx 51.5040$

При этом c_3^* критическая точка $(x_4, y_4) \approx (-4.8952, 1.4224)$. Ее координаты являются вторым набором корней многочленов (6.3). Аналогично рассмотрим квадратичную форму разложения

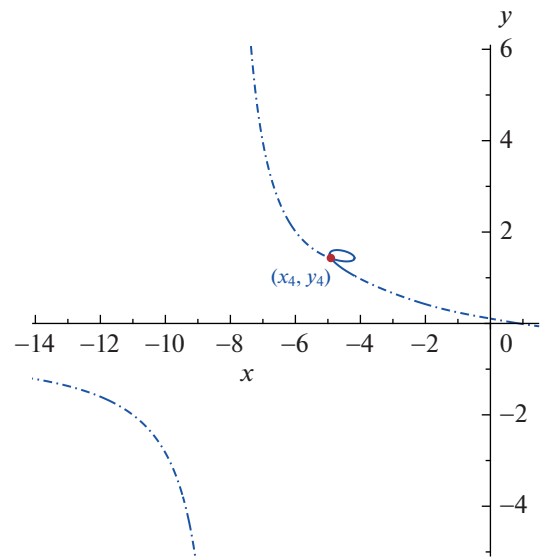


Рис. 20. Линия уровня для критического значения $c_3^* \approx 51.5040$ многочлена f_3 .

ряда Тейлора исходного многочлена в критической точке $x_4 \approx -4.8952, y_4 \approx 1.4224$ соответствующей критическому значению $c_3 = 51.5040$:

$$h_2(x_4 + \xi, y_4 + \eta) = 51.5041 - 2046.65935\eta\xi - 408.778677\xi^2 - 1602.0119\eta^2.$$

Здесь постоянная близка к c_3^* и коэффициенты линейных членов равны нулю. Дискриминант $\Delta \approx 1.5693 \times 10^6$ положительный, что по лемме 2 означает точку, в которой пересекаются две ветви

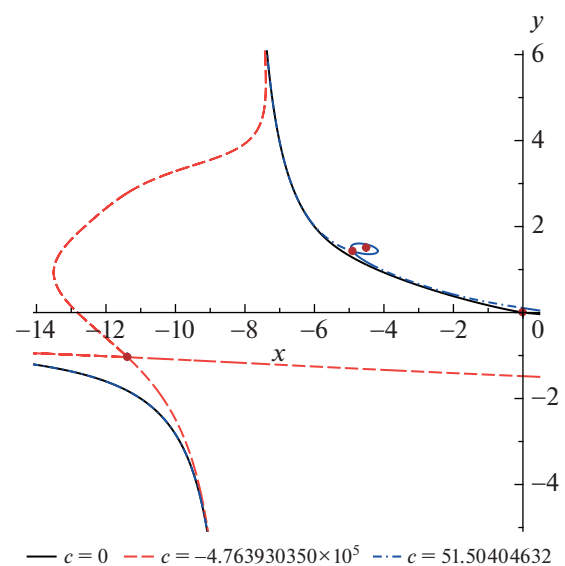


Рис. 21. Все критические линии уровня многочлена f_3 .

кривой. Критические линии уровня при c_3^* пересекают оси Ox и Oy при $x \approx 0.8749$ и $y \approx 0.1063$ соответственно. Линия уровня и критическая точка изображены на рис. 20, где критическая линия показана штрих-пунктирной линией синего цвета.

6.5. Все критические линии уровня многочлена f_3

На рис. 21 показаны все критические линии уровня многочлена f_3 для соответствующих критических значений c_j^* , а также критические точки. Некритические линии уровня расположены между соответствующими критическими линиями и критическими точками.

Отметим, что для всех трех многочленов f_1, f_2, f_3 все бесконечные точки пересечения имеют первую кратность. Поэтому применима лемма 4 и все некритические линии уровня расположены между найденными критическими и топологически эквивалентны в интервалах (2.3).

СПИСОК ЛИТЕРАТУРЫ

1. Брюно А.Д., Батхин А.Б. Алгоритмы и программы вычисления корней многочлена от одной или двух неизвестных // Программирование. 2021. № 5. С. 22–43.
2. Брюно А.Д., Батхин А.Б. Линии уровня многочлена на плоскости // Программирование. 2022. № 1. С. 22–33.
3. Батхин А.Б., Брюно А.Д., Варин В.П. Множества устойчивости многопараметрических гамильтоновых систем // Прикладная математика и механика. 2012. Т. 76. № 1. С. 80–133.
4. Брюно А.Д. Локальный метод нелинейного анализа дифференциальных уравнений. М.: Наука, 1979. 252 с.
5. Kollár J. Lectures on Resolution of Singularities. Princeton and Oxford : Princeton University Press, 2007.
6. Милнор Дж. Теория Морса: Пер. с англ. 3-е изд. М.: Издательство ЛКИ, 2011. 184 с.
7. Кокс Д., Литтл Д., О'Ши Д. Идеалы, многообразия и алгоритмы. Введение в вычислительные аспекты алгебраической геометрии и коммутативной алгебры. М.: Мир, 2000. 687 с.

УДК 519.85

СИМВОЛЬНЫЙ АЛГОРИТМ ДЛЯ ЗАДАЧИ НАХОЖДЕНИЯ НУЛЕЙ СИСТЕМЫ ГОЛОМОРФНЫХ ФУНКЦИЙ

© 2023 г. В. И. Кузоватов^{a,*}, А. А. Кытманов^{b,c,**}, Е. К. Мышкина^{d,***}

^aСибирский федеральный университет

660041 Красноярск, пр. Свободный, 79, Россия

^bМИРЭА — Российский технологический университет

119454 Москва, пр. Вернадского, 78, Россия

^cУчебно-научная лаборатория искусственного интеллекта, нейротехнологий и бизнес-аналитики, Российский экономический университет им. Г.В. Плеханова

117997 Москва, Стремянный пер., 36, Россия

^dИнститут вычислительного моделирования СО РАН

660036 Красноярск, Академгородок, 50/44, Россия

*E-mail: kuzovатов@yandex.ru

**E-mail: aakyt@gmail.com

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

Поступила в редакцию 28.08.2022 г.

После доработки 13.10.2022 г.

Принята к публикации 30.10.2022 г.

На основе интегрального представления Бохнера–Мартинелли приведен алгоритм, позволяющий определять число нулей системы голоморфных функций. Нули системы функций ищутся на множестве поликуба. Использование методов компьютерной алгебры в данной задаче обусловлено вычислительной сложностью разрабатываемых алгоритмов и получаемых результатов. Дана реализация данного алгоритма в системе компьютерной алгебры Maple, позволяющая существенно упростить необходимые вычисления.

DOI: 10.31857/S0132347423020139, EDN: MGDMMHW

1. ВВЕДЕНИЕ

Данная работа посвящена исследованию систем неалгебраических уравнений в $\mathbb{C}^n$. Такие системы возникают при описании нелинейных процессов в различных областях знания, таких как теоретическая физика, химическая кинетика (включая задачи, возникающие при описании процессов в нефтегазовой промышленности, а также при изучении кинетики химических превращений композитов на основе горных пород), математическая биология. На данный момент системы неалгебраических уравнений, как правило, поддаются решению лишь в случае, когда их можно свести к алгебраическим с помощью замены переменных. В противном случае поиск решения производится приближенно с помощью численных методов [1].

Вместе с тем, в различных 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 разных множествах, определения самих корней, исключения части неизвестных из системы.

Научная значимость данной работы связана с определением нового подхода к итерационным методам решения систем трансцендентных уравнений, состоящих из голоморфных функций многих комплексных переменных. Полученные результаты позволят расширить области применения методов и алгоритмов компьютерной алгебры на нелинейные уравнения неполиномиального типа. В связи с этим решение поставленных в данной работе задач важно для развития данного направления науки.

2. ОСНОВНОЙ РЕЗУЛЬТАТ

Пусть D — ограниченная область в $\mathbb{C}^2$ с кусочно-гладкой границей ∂D , а $f : \bar{D} \rightarrow \mathbb{C}^2$ голоморфное отображение, $f|_{\partial D} \neq 0$.

Известно обобщение теоремы о логарифмическом вычете на случай многих переменных на основе интегрального представления Бохнера–Мартинелли (см., например, [2, теорема 2.4]):

$$\int_{\partial D} w(f, \bar{f}) = N, \quad (1)$$

где N — число нулей отображения f в D с учетом кратности, а

$$w(f, \bar{f}) = \frac{1}{(2\pi i)^2} \frac{\bar{f}_1 df_2 - \bar{f}_2 df_1}{(|f_1|^2 + |f_2|^2)^2} \wedge df_1 \wedge df_2.$$

Рассмотрим случай, когда D — поликуб, то есть произведение отрезков

$$[a, b] \times [\alpha, \beta] \times [c, d] \times [\gamma, \delta] \subset \mathbb{C}^2 = \mathbb{R}^4.$$

Перейдем к вещественным координатам

$$\begin{cases} z_1 = x_1 + iy_1, \\ z_2 = x_2 + iy_2. \end{cases}$$

Тогда

$$\begin{aligned} a \leq x_1 \leq b, \quad \alpha \leq y_1 \leq \beta, \\ c \leq x_2 \leq d, \quad \gamma \leq y_2 \leq \delta. \end{aligned} \quad (2)$$

Запишем форму w в вещественных координатах. Будем иметь

$$\begin{aligned} w = & -\frac{2}{(2\pi i)^2} \frac{\left(\frac{\partial f_1}{\partial z_1} \frac{\partial f_2}{\partial z_2} - \frac{\partial f_1}{\partial z_2} \frac{\partial f_2}{\partial z_1} \right)}{(|f_1|^2 + |f_2|^2)^2} \times \\ & \times \left[\left(f_1 \frac{\partial f_2}{\partial z_1} - f_2 \frac{\partial f_1}{\partial z_1} \right) \times \right. \\ & \times (dx_1 \wedge dy_1 \wedge dy_2 + idx_1 \wedge dx_2 \wedge dy_1) + \\ & + \left(f_1 \frac{\partial f_2}{\partial z_2} - f_2 \frac{\partial f_1}{\partial z_2} \right) \times \\ & \times (dx_2 \wedge dy_1 \wedge dy_2 + idx_1 \wedge dx_2 \wedge dy_2) \Big]. \end{aligned}$$

Для удобства записи введем следующие обозначения. Обозначим через J якобиан отображения f , через $|f|^2$ — квадрат евклидовой нормы. Таким образом,

$$\begin{aligned} \frac{\partial f_1}{\partial z_1} \frac{\partial f_2}{\partial z_2} - \frac{\partial f_1}{\partial z_2} \frac{\partial f_2}{\partial z_1} &= J, \\ |f_1|^2 + |f_2|^2 &= |f|^2, \\ \frac{f_1 \frac{\partial f_2}{\partial z_1} - f_2 \frac{\partial f_1}{\partial z_1}}{f_1 \frac{\partial f_2}{\partial z_2} - f_2 \frac{\partial f_1}{\partial z_2}} &= A, \\ \frac{f_1 \frac{\partial f_2}{\partial z_2} - f_2 \frac{\partial f_1}{\partial z_2}}{f_1 \frac{\partial f_2}{\partial z_1} - f_2 \frac{\partial f_1}{\partial z_1}} &= B. \end{aligned}$$

Таким образом, с учетом введенных обозначений форму w можно записать в виде

$$\begin{aligned} w = & -\frac{2}{(2\pi i)^2} \frac{J}{|f|^4} \times \\ & \times [A(dx_1 \wedge dy_1 \wedge dy_2 + idx_1 \wedge dx_2 \wedge dy_1) + \\ & + B(dx_2 \wedge dy_1 \wedge dy_2 + idx_1 \wedge dx_2 \wedge dy_2)]. \end{aligned} \quad (3)$$

Для сведения интеграла (1) к интегралу Римана по гиперграням нужно выбрать их правильную ориентацию. Ориентация на многообразии задается (см., например, [3], глава II, п. 14) указанием дифференциальной формы максимальной степени, которая сопоставляется элементу объема dV при интегрировании. Ориентация $\mathbb{R}^4 = \mathbb{C}^2$, как и гиперкуба $D \subset \mathbb{C}^2$, традиционно задается формой

$$dx_1 \wedge dx_2 \wedge dy_1 \wedge dy_2.$$

Согласованная с ней ориентация (в соответствии с [3], глава II, п. 14) граней следующая:

$$\begin{aligned} x_1 = a & : -dx_2 \wedge dy_1 \wedge dy_2 \\ x_1 = b & : dx_2 \wedge dy_1 \wedge dy_2 \\ x_2 = c & : dx_1 \wedge dy_1 \wedge dy_2 \\ x_2 = d & : -dx_1 \wedge dy_1 \wedge dy_2 \\ y_1 = \alpha & : -dx_1 \wedge dx_2 \wedge dy_2 \\ y_1 = \beta & : dx_1 \wedge dx_2 \wedge dy_2 \\ y_2 = \gamma & : dx_1 \wedge dx_2 \wedge dy_1 \\ y_2 = \delta & : -dx_1 \wedge dx_2 \wedge dy_1. \end{aligned}$$

Интеграл (1) есть сумма интегралов по правильно ориентированным гиперграням. При этом имеется 4 пары гиперграней.

Рассмотрим для примера пару гиперграней, задаваемых условиями $x_1 = a$ и $x_1 = b$. В этом случае 3 из 4 дифференциальных форм в выражении (3) обращаются в нуль и форма w примет вид

$$w = -\frac{2}{(2\pi i)^2} \frac{J}{|f|^4} B dx_2 \wedge dy_1 \wedge dy_2.$$

Форма $dx_2 \wedge dy_1 \wedge dy_2$ на $x_1 = a$ отрицательна, а на $x_1 = b$ — положительна. Поэтому

$$\begin{aligned} I_1 &= \int_{x_1=a} w + \int_{x_1=b} w = \\ &= \frac{2}{(2\pi i)^2} \int_{\substack{x_1=a, \\ c \leq x_2 \leq d, \\ \alpha \leq y_1 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} \frac{J}{|f|^4} B dV - \frac{2}{(2\pi i)^2} \int_{\substack{x_1=b, \\ c \leq x_2 \leq d, \\ \alpha \leq y_1 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} \frac{J}{|f|^4} B dV = \end{aligned} \quad (4)$$

$$= \frac{2}{(2\pi i)^2} \int_{\substack{c \leq x_2 \leq d, \\ \alpha \leq y_1 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} \frac{J}{|f|^4} B \Big|_{x_1=b}^{x_1=a} dV.$$

Аналогично, получим следующие результаты для остальных пар гиперграней:

$$I_2 = \int_{x_2=c} w + \int_{x_2=d} w = \frac{2}{(2\pi i)^2} \int_{\substack{a \leq x_1 \leq b, \\ \alpha \leq y_1 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} \frac{J}{|f|^4} A \Big|_{x_2=c}^{x_2=d} dV, \quad (5)$$

$$I_3 = \int_{y_1=\alpha} w + \int_{y_1=\beta} w = \frac{2i}{(2\pi i)^2} \int_{\substack{a \leq x_1 \leq b, \\ c \leq x_2 \leq d, \\ \gamma \leq y_2 \leq \delta}} \frac{J}{|f|^4} B \Big|_{y_1=\beta}^{y_1=\alpha} dV, \quad (6)$$

$$I_4 = \int_{y_2=\gamma} w + \int_{y_2=\delta} w = \frac{2i}{(2\pi i)^2} \int_{\substack{a \leq x_1 \leq b, \\ c \leq x_2 \leq d, \\ \alpha \leq y_1 \leq \beta}} \frac{J}{|f|^4} A \Big|_{y_2=\gamma}^{y_2=\delta} dV. \quad (7)$$

Тем самым мы приходим к следующему утверждению.

Теорема 1. Пусть D — поликуб, то есть произведение отрезков $[a, b] \times [\alpha, \beta] \times [c, d] \times [\gamma, \delta] \subset \mathbb{C}^2 = \mathbb{R}^4$, $a, f : \bar{D} \rightarrow \mathbb{C}^2$ голоморфное отображение, $f|_{\partial D} \neq 0$. Тогда количество N нулей системы

$$\begin{cases} f_1(z_1, z_2) = 0, \\ f_2(z_1, z_2) = 0 \end{cases} \quad (8)$$

на множестве D численно равно сумме значений интегралов (4)–(7). При этом интегралы (4)–(7) являются интегралами Римана, в которых переменные интегрирования изменяются от меньшего значения к большему.

Заметим, что сумма интегралов (4)–(7) должна быть целым числом. Это означает, что для приближенного вычисления этих интегралов большая точность не требуется.

При наличии корней в поликубе D (то есть если указанная сумма интегралов не равна нулю), он разбивается на несколько поликубов меньшего размера, и вычисления повторяются в каждом из них для локализации корней. Алгоритм останавливается при достижении необходимой точности.

3. ПРИМЕР

Пусть $f_1 = z_1$, $f_2 = z_2$, то есть рассмотрим количество N нулей системы

$$\begin{cases} z_1 = 0, \\ z_2 = 0 \end{cases}$$

в области $D = [-1; 1] \times [-1; 1] \times [-1; 1] \times [-1; 1]$. Очевидно, что единственным решением данной системы является точка $(0, 0) \in \mathbb{C}^2$ и $N = 1$. Покажем это с помощью теории, изложенной выше. Такой выбор функций f_1 и f_2 объясняется необходимостью упрощения выкладок.

Вычислим необходимые величины. Будем иметь:

$$\frac{\partial f_1}{\partial z_1} = 1, \quad \frac{\partial f_1}{\partial z_2} = 0, \quad \frac{\partial f_2}{\partial z_1} = 0, \quad \frac{\partial f_2}{\partial z_2} = 1,$$

$$J = \frac{\partial f_1}{\partial z_1} \frac{\partial f_2}{\partial z_2} - \frac{\partial f_1}{\partial z_2} \frac{\partial f_2}{\partial z_1} = 1,$$

$$|f|^2 = |f_1|^2 + |f_2|^2 = |z_1|^2 + |z_2|^2 = x_1^2 + y_1^2 + x_2^2 + y_2^2,$$

$$A = f_1 \frac{\partial f_2}{\partial z_1} - f_2 \frac{\partial f_1}{\partial z_1} = -\bar{z}_2 = -x_2 + iy_2,$$

$$B = f_1 \frac{\partial f_2}{\partial z_2} - f_2 \frac{\partial f_1}{\partial z_2} = \bar{z}_1 = x_1 - iy_1.$$

Вычислим интегралы по гиперграням $x_1 = a$ и $x_1 = b$, при этом $a = -1$, $b = 1$. Получим

$$\begin{aligned} & \frac{2}{(2\pi i)^2} \int_{\substack{c \leq x_2 \leq d, \\ \alpha \leq y_1 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} \frac{J}{|f|^4} B \Big|_{x_1=b}^{x_1=a} dV = \\ &= \frac{2}{(2\pi i)^2} \int_{\substack{c \leq x_2 \leq d, \\ \alpha \leq y_1 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} \frac{(x_1 - iy_1)}{(x_1^2 + y_1^2 + x_2^2 + y_2^2)^2} \Big|_{x_1=b}^{x_1=a} dV = \\ &= \frac{2}{(2\pi i)^2} \iiint_{\substack{c \leq x_2 \leq d, \\ \alpha \leq y_1 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} \left(\frac{a - iy_1}{(a^2 + y_1^2 + x_2^2 + y_2^2)^2} - \right. \\ & \quad \left. - \frac{b - iy_1}{(b^2 + y_1^2 + x_2^2 + y_2^2)^2} \right) dV = \\ &= \frac{2}{(2\pi i)^2} \iiint_{\substack{c \leq x_2 \leq d, \\ \alpha \leq y_1 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} \frac{a - b}{(1 + y_1^2 + x_2^2 + y_2^2)^2} dV = \\ &= \frac{1}{\pi^2} \int_{-1}^1 dx_2 \int_{-1}^1 dy_1 \int_{-1}^1 \frac{dy_2}{(1 + y_1^2 + x_2^2 + y_2^2)^2}. \end{aligned}$$

Вычислим далее интегралы по гиперграням $x_2 = c$ и $x_2 = d$, при этом $c = -1$, $d = 1$. Получим

$$\begin{aligned}
& \frac{2}{(2\pi i)^2} \int_{\substack{a \leq x_1 \leq b, \\ \alpha \leq y_1 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} \frac{J}{|f|^4} A \Big|_{x_2=c}^{x_2=d} dV = \\
& = \frac{2}{(2\pi i)^2} \int_{\substack{a \leq x_1 \leq b, \\ \alpha \leq y_1 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} \frac{(-x_2 + iy_2)}{(x_1^2 + y_1^2 + x_2^2 + y_2^2)^2} \Big|_{x_2=c}^{x_2=d} dV = \\
& = \frac{2}{(2\pi i)^2} \iiint_{\substack{a \leq x_1 \leq b, \\ \alpha \leq y_1 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} \left(\frac{-d + iy_2}{(x_1^2 + y_1^2 + d^2 + y_2^2)^2} - \right. \\
& \quad \left. - \frac{(-c + iy_2)}{(x_1^2 + y_1^2 + c^2 + y_2^2)^2} \right) dV = \\
& = \frac{2}{(2\pi i)^2} \iiint_{\substack{a \leq x_1 \leq b, \\ \alpha \leq y_1 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} \frac{c - d}{(x_1^2 + y_1^2 + 1 + y_2^2)^2} dV = \\
& = \frac{1}{\pi^2} \int_{-1}^1 dx_1 \int_{-1}^1 dy_1 \int_{-1}^1 \frac{dy_2}{(1 + x_1^2 + y_1^2 + y_2^2)^2}.
\end{aligned}$$

Вычислим далее интегралы по гиперграням $y_1 = \alpha$ и $y_1 = \beta$, при этом $\alpha = -1$, $\beta = 1$. Получим

$$\begin{aligned}
& \frac{2i}{(2\pi i)^2} \int_{\substack{a \leq x_1 \leq b, \\ c \leq x_2 \leq d, \\ \gamma \leq y_2 \leq \delta}} \frac{J}{|f|^4} B \Big|_{y_1=\beta}^{y_1=\alpha} dV = \\
& = \frac{2i}{(2\pi i)^2} \int_{\substack{a \leq x_1 \leq b, \\ c \leq x_2 \leq d, \\ \gamma \leq y_2 \leq \delta}} \frac{(x_1 - iy_1)}{(x_1^2 + y_1^2 + x_2^2 + y_2^2)^2} \Big|_{y_1=\beta}^{y_1=\alpha} dV = \\
& = \frac{2i}{(2\pi i)^2} \iiint_{\substack{a \leq x_1 \leq b, \\ c \leq x_2 \leq d, \\ \gamma \leq y_2 \leq \delta}} \left(\frac{x_1 - i\alpha}{(x_1^2 + \alpha^2 + x_2^2 + y_2^2)^2} - \right. \\
& \quad \left. - \frac{x_1 - i\beta}{(x_1^2 + \beta^2 + x_2^2 + y_2^2)^2} \right) dV = \\
& = \frac{2i}{(2\pi i)^2} \iiint_{\substack{a \leq x_1 \leq b, \\ c \leq x_2 \leq d, \\ \gamma \leq y_2 \leq \delta}} \frac{i(\beta - \alpha)}{(1 + x_1^2 + x_2^2 + y_2^2)^2} dV = \\
& = \frac{1}{\pi^2} \int_{-1}^1 dx_1 \int_{-1}^1 dx_2 \int_{-1}^1 \frac{dy_2}{(1 + x_1^2 + x_2^2 + y_2^2)^2}.
\end{aligned}$$

Нам осталось вычислить последние интегралы по гиперграням $y_2 = \gamma$ и $y_2 = \delta$, при этом $\gamma = -1$, $\delta = 1$. Получим

$$\begin{aligned}
& \frac{2i}{(2\pi i)^2} \int_{\substack{a \leq x_1 \leq b, \\ c \leq x_2 \leq d, \\ \alpha \leq y_1 \leq \beta}} \frac{J}{|f|^4} A \Big|_{y_2=\gamma}^{y_2=\delta} dV = \\
& = \frac{2i}{(2\pi i)^2} \int_{\substack{a \leq x_1 \leq b, \\ c \leq x_2 \leq d, \\ \alpha \leq y_1 \leq \beta}} \frac{(-x_2 + iy_2)}{(x_1^2 + y_1^2 + x_2^2 + y_2^2)^2} \Big|_{y_2=\gamma}^{y_2=\delta} dV = \\
& = \frac{2i}{(2\pi i)^2} \iiint_{\substack{a \leq x_1 \leq b, \\ c \leq x_2 \leq d, \\ \alpha \leq y_1 \leq \beta}} \left(\frac{(-x_2 + i\delta)}{(x_1^2 + y_1^2 + x_2^2 + \delta^2)^2} - \right. \\
& \quad \left. - \frac{(-x_2 + i\gamma)}{(x_1^2 + y_1^2 + x_2^2 + \gamma^2)^2} \right) dV = \\
& = \frac{2i}{(2\pi i)^2} \iiint_{\substack{a \leq x_1 \leq b, \\ c \leq x_2 \leq d, \\ \alpha \leq y_1 \leq \beta}} \frac{i(\delta - \gamma)}{(x_1^2 + y_1^2 + x_2^2 + 1)^2} dV = \\
& = \frac{1}{\pi^2} \int_{-1}^1 dx_1 \int_{-1}^1 dy_1 \int_{-1}^1 \frac{dx_2}{(1 + x_1^2 + y_1^2 + x_2^2)^2}.
\end{aligned}$$

Таким образом, количество N нулей искомой системы равно

$$N = \frac{4}{\pi^2} \int_{-1}^1 du \int_{-1}^1 dv \int_{-1}^1 \frac{dw}{(1 + u^2 + v^2 + w^2)^2}.$$

Очевидно, что даже в таком простом случае системы, как в рассмотренном нами примере, необходима помощь вычислительных процедур.

4. ОПИСАНИЕ АЛГОРИТМА

Алгоритм был реализован в среде Maple 2016 64bit. Полный код программы доступен по адресу <https://github.com/aakytmanov/Zeros>. Вычисления первого шага алгоритма производились на машине Intel Core i7-4790 (3.6 GHz) с 32 Gb RAM под управлением Windows 10 Pro x64 21H1. Время счета для системы функций

$$f_1 = z_1, \quad f_2 = z_2$$

составило 53 секунды.

Вызов функции для данного примера приведен ниже:

```
> fZeros([(z1, z2) -> z1, (z1, z2) -> z2],
[-1, 1, -1, 1, -1, 1, -1, 1]);
```

Реализация итерационной процедуры позволит приближенно находить корни системы уравнений, содержащей голоморфные функции, в за-

данном поликубе, однако она требует оптимизации приближенного вычисления интегралов с учетом отсутствия необходимости большой точности вычислений, поскольку результатом интегрирования всегда является целое число. Это является актуальной текущей задачей.

Алгоритм 1: Алгоритм вычисления числа нулей системы в заданной области.

Input: Список функций $f_i(z)$ из левой части системы (8), список пределов интегрирования $a, b, \alpha, \beta, c, d, \gamma, \delta$ из (2), определяющих поликуб.

Output: Значение интеграла (1).

begin

$$f_1(z_1, z_2) \rightarrow u_1(x_1, y_1, x_2, y_2) + iv_1(x_1, y_1, x_2, y_2)$$

$$f_2(z_1, z_2) \rightarrow u_2(x_1, y_1, x_2, y_2) + iv_2(x_1, y_1, x_2, y_2)$$

$$|f|^4 := (f_1 \bar{f}_1 + f_2 \bar{f}_2)^2$$

$$J := \frac{\partial f_1}{\partial z_1} \frac{\partial f_2}{\partial z_2} - \frac{\partial f_1}{\partial z_2} \frac{\partial f_2}{\partial z_1}$$

$$A := f_1 \frac{\partial f_2}{\partial z_1} - f_2 \frac{\partial f_1}{\partial z_1}$$

$$B := f_1 \frac{\partial f_2}{\partial z_2} - f_2 \frac{\partial f_1}{\partial z_2}$$

$$I_{x_1} := \frac{J}{|f|^4} B \Big|_{x_1=b}^{x_1=a}$$

$$I_{x_2} := \frac{J}{|f|^4} A \Big|_{x_2=c}^{x_2=d}$$

$$I_{y_1} := \frac{J}{|f|^4} B \Big|_{y_1=\beta}^{y_1=\alpha}$$

$$I_{y_2} := \frac{J}{|f|^4} A \Big|_{y_2=\gamma}^{y_2=\delta}$$

$$I_1 := \frac{2}{(2\pi i)^2} \int_{\substack{c \leq x_2 \leq d, \\ \alpha \leq y_2 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} I_{x_1} dV$$

$$I_2 := \frac{2}{(2\pi i)^2} \int_{\substack{a \leq x_1 \leq b, \\ \alpha \leq y_1 \leq \beta, \\ \gamma \leq y_2 \leq \delta}} I_{x_2} dV$$

$$I_3 := \frac{2i}{(2\pi i)^2} \int_{\substack{a \leq x_1 \leq b, \\ c \leq x_2 \leq d, \\ \gamma \leq y_2 \leq \delta}} I_{y_1} dV$$

$$I_4 := \frac{2i}{(2\pi i)^2} \int_{\substack{a \leq x_1 \leq b, \\ c \leq x_2 \leq d, \\ \alpha \leq y_1 \leq \beta}} I_{y_2} dV$$

$$I := I_1 + I_2 + I_3 + I_4$$

Return I

5. ИСТОЧНИКИ ФИНАНСИРОВАНИЯ

Исследования, представленные в работе, были выполнены при поддержке следующих фондов: первый автор использовал финансовую поддержку РНФ, Правительства Красноярского края и Красноярского краевого фонда науки, грант 22-21-20028 (разработка примеров, тестирование программной реализации алгоритма); второй автор использовал финансовую поддержку РНФ, грант 18-71-10007-П (разработка и программная реализация алгоритма); третий автор поддержан грантом РФФИ 19-31-60012 (постановка технического задания для разработки алгоритма, вычисление примеров).

СПИСОК ЛИТЕРАТУРЫ

1. Ортега Дж., Рейнболдт В. Итерационные методы решения нелинейных систем уравнений со многими неизвестными. М.: Мир, 1975.
2. Айзенберг Л.А., Южаков А.П. Интегральные представления и вычеты в многомерном комплексном анализе. Новосибирск: Наука, 1979.
3. Шабат Б.В. Введение в комплексный анализ. Функции нескольких переменных. Ч. 2. СПб.: Лань, 2004.

УДК 519.688

ВЫЧИСЛЕНИЕ СВЯЗНЫХ КОМПОНЕНТ ДОПОЛНЕНИЯ К АМЕБЕ МНОГОЧЛЕНА ОТ НЕСКОЛЬКИХ КОМПЛЕКСНЫХ ПЕРЕМЕННЫХ

© 2023 г. Т. А. Жуков^{а,*}, Т. М. Садыков^{а,**}^аРоссийский экономический университет им. Г.В. Плеханова,
117997 Москва, Стремянный пер., 36, Россия

*E-mail: Zhukov.TA@rea.ru

**E-mail: Sadykov.TM@rea.ru

Поступила в редакцию 13.05.2022 г.

После доработки 03.08.2022 г.

Принята к публикации 30.10.2022 г.

В настоящей работе предложен метод вычисления и визуализации амобы многочлена Лорана нескольких комплексных переменных, применимый в произвольной размерности. Разработанные на основе этого метода алгоритмы реализованы в виде общедоступного сетевого сервиса <http://amoebas.ru/>, позволяющего осуществлять интерактивный расчет амоб многочленов двух переменных и содержащего набор рассчитанных амоб и их сечений в более высоких размерностях. Тестирование корректности и скорости работы предложенных алгоритмов осуществлено с использованием набора оптимальных многочленов двух, трех и четырех переменных, для генерации которых применен функционал системы компьютерной алгебры Mathematica. Разработанный программный код позволяет, в частности, осуществлять генерацию оптимального гипергеометрического многочлена от произвольного числа переменных с носителем в произвольном зонотопе, заданном набором порождающих векторов.

DOI: 10.31857/S0132347423020164, EDN: MHCDUI

1. ВВЕДЕНИЕ

Понятие амобы [2], [3, глава 1] многочлена Лорана нескольких комплексных переменных прочно вошло в современную математику и широко используется в многочисленных приложениях. В частности, оно возникает в задачах математической физики [4, 5] и при построении решений разностных уравнений и их систем [1]. Известно, что фазовая диаграмма мер Гиббса для димеров на графе может быть представлена в виде амобы ассоциированной спектральной кривой [6, Теорема 4.1]. Изучение распределения энергии в квантовых термодинамических ансамблях естественным образом приводит к рассмотрению понятия амобы и ее т.н. контура [7]. Двойственность Лежандра между энтропией и свободной энергией позволяет вычислять последнюю в рамках теории колчанов как площадь амобы, заданной многочленом Ньютона для модели димеров [8]. С понятием амобы связан ряд важных открытых вопросов, в частности, остающаяся недоказанной на протяжении длительного времени гипотеза М. Пассаре о сплошном свойстве амобы произвольного максимального разреженного многочлена. Эти и многие другие приложения теории амоб в математической физике [9] и вещественной алгебраической геометрии [10] обуславливают давно назревшую не-

обходимость в разработке алгоритмов и общедоступного программного обеспечения для вычисления амоб многочленов Лорана нескольких комплексных переменных.

Попытки создания таких алгоритмов и их программной реализации предпринимались многими исследователями на протяжении трех последних десятилетий. В их основе лежали численное решение алгебраического уравнения относительно выделенного переменного или группы переменных [11–15], аппроксимация амоб с помощью сумм квадратов [16], полиэдральная аппроксимация фиксированной компоненты дополнения к амобе [17], подход с использованием циклических результатов [18] и другие методы [19]. Существенную роль в работах [16, 18] играют методы современной компьютерной алгебры. Сетевой ресурс <http://dvbogdanov.ru/amoeba> позволяет генерировать программный код на языке среды MATLAB для расчета аффинных и компактифицированных амоб многочленов двух переменных [20, 21]. Текущие границы применимости данного подхода иллюстрируются набором данных www.researchgate.net/publication/338341129_Giant_amoeba_zoo.

Однако, несмотря на значительный прогресс по сравнению с первоначальным представлением

авторов определения амёбы о виде амёб многочленов нескольких комплексных переменных (см. [2], стр. 194, рис. 16), задача вычисления и визуализации амёбы многочлена нескольких комплексных переменных (особенно в размерности 3 и выше) в настоящий момент не может считаться удовлетворительно разрешенной. Причин этому несколько: общедоступный программный код для генерации амёб дает приемлемые результаты лишь для многочленов невысокой степени и, как правило, в размерности 2; при этом некоторые из имеющихся решений существенно опираются на функционал коммерческих программных продуктов с закрытым кодом; отсутствует общедоступный сетевой сервис для интерактивной генерации амёб достаточно сложных многочленов двух и трех переменных; не существует инструментария для визуализации деформации амёбы многочлена, зависящего от параметров. Настоящая работа призвана восполнить этот пробел.

Высокая вычислительная трудность задачи расчета и визуализации амёбы многочлена нескольких переменных обусловлена, в первую очередь, следующими обстоятельствами:

1. За исключением тривиального одномерного случая, амёба многочлена является неограниченным подмножеством вещественного линейного пространства.

2. Амёба $\mathcal{A}_f$ многочлена f двух переменных имеет конечную площадь. В размерности три и выше отношение меры пересечения $\mathcal{A}_f \cap B(a, r)$ амёбы многочлена f с шаром радиуса r и центром в точке a к объему этого шара стремится к нулю при $r \rightarrow \infty$ для любого многочлена f и любой точки $a \in \mathbb{R}^n$. Таким образом, определение местоположения амёбы многочлена в пространстве $\mathbb{R}^n$ само по себе является, как правило, нетривиальной задачей.

3. Ограниченные связные компоненты дополнения к амёбе могут быть сколь угодно малы, а получение оценок снизу на их размеры является весьма трудоемкой задачей.

4. Количество связных компонент дополнения к амёбе многочлена f с фиксированным многогранником Ньютона $\mathcal{N}$ может принимать (в зависимости от выбора коэффициентов f) произвольные значения в диапазоне от числа вершин многогранника $\mathcal{N}$ до числа целых точек в нем [13].

5. Численный расчет достаточно сложной амёбы многочлена высокой степени возможен лишь при весьма высокой точности вычислений. Например, при расчете амёбы многочлена десятой степени от двух переменных с целыми коэффициентами в квадрате $[-20, 20] \times [-20, 20]$ необходимо оперировать величинами порядка $e^{10 \cdot 20} \simeq 7.22597 \times 10^{86}$, поддерживая точность вычислений, сравнимую с

$e^{-10 \cdot 20} \simeq 1.38389 \times 10^{-87}$. Многочисленные компьютерные эксперименты показывают, что уменьшение точности расчета приводит, как правило, к быстрому накоплению вычислительной погрешности и потере связи результата расчета с подлинной геометрией амёбы.

В настоящей работе представлен новый метод вычисления и визуализации амёбы многочлена нескольких комплексных переменных, применимый в произвольной размерности. Разработанные алгоритмы вычисления амёб многочленов нескольких переменных реализованы в виде общедоступного сетевого сервиса <http://amoebas.ru/>. Данный ресурс позволяет в интерактивном режиме выполнять расчет и визуализацию амёб многочленов двух переменных, а также содержит заранее рассчитанные облака точек, аппроксимирующие амёбы некоторых оптимальных гипергеометрических многочленов [22] трех и четырех переменных. Для визуализации четырехмерных амёб на сайте выполняется отрисовка их сечений трехмерными гиперплоскостями.

2. ОСНОВНЫЕ ОБОЗНАЧЕНИЯ И ОПРЕДЕЛЕНИЯ

Всюду в дальнейшем $n \geq 2$ обозначает размерность пространства комплексных переменных, $f \in \mathbb{C}[x_1^{\pm 1}, \dots, x_n^{\pm 1}]$ — многочлен Лорана от переменных $(x_1, \dots, x_n) \in \mathbb{C}^n$. Обозначим через $\mathbb{C}^* := \mathbb{C} \setminus \{0\}$ проколотую комплексную плоскость и рассмотрим отображение $\text{Log} : (\mathbb{C}^*)^n \rightarrow \mathbb{R}^n$, определенное формулой

$$\text{Log} : \mathbf{x} = (x_1, \dots, x_n) \mapsto (\ln |x_1|, \dots, \ln |x_n|).$$

Амёбой (см. [2, глава 6, определение 1.4]) алгебраического множества $V \subset (\mathbb{C}^*)^n$ называется образ $\text{Log} V$, который обозначается $\mathcal{A}_V$. Данное название связано с характерной формой множества $\mathcal{A}_V$ в случае, когда множество V является комплексной алгебраической кривой $\{(x, y) \in \mathbb{C}^2 : f(x, y) = 0\}$. Соответствующая такому множеству амёба, как правило, ограничивает шарообразные “вакуоли” и содержит тонкие “щупальца”, уходящие в бесконечность (см. примеры амёб в работах [14, 20, 21] и на сайте <http://amoebas.ru/>).

Амёба подмножества комплексного пространства лишь логарифмической шкалой отличается от его диаграммы Рейнхардта. Помимо амёбы, были предложены и исследованы другие проекции комплексного пространства $\mathbb{C}^n$ на вещественное пространство той же размерности, на следующие ключевые аналитические и топологические свойства дополнения к алгебраической гиперповерхности. К их числу относятся, в част-

ности, компактифицированная амеба (см. [2], глава 6, теорема 1.12 и рис. 19) и “амебоподобный” полиэдральный комплекс, конструкция которого предложена в работе [21]. Следующий результат [2], глава 6, следствие 1.6 устанавливает простую связь между амебой многочлена Лорана f и разложениями рациональной функции $1/f$ в ряды Лорана с центром в нуле.

Теорема 1 (см. [2], глава 6). *Дополнение ${}^c\mathcal{A}_f := \mathbb{R}^n \setminus \mathcal{A}_f$ к амебе многочлена f есть объединение конечного числа непересекающихся связных компонент. Эти компоненты выпуклы и находятся во взаимно-однозначном соответствии с разложениями рациональной функции $1/f$ в ряды Лорана с центром в начале координат.*

Выпуклость связных компонент дополнения к амебе алгебраической гиперповерхности выгодно отличает ее от всех конкурирующих проекций комплексного пространства в вещественное и, как будет показано далее, играет существенную роль в построении алгоритмов для вычисления амеб.

Напомним, что носителем многочлена Лорана f называется конечное множество целых точек $\alpha \in \mathbb{Z}^n$, таких, что моном $x_1^{\alpha_1} \dots x_n^{\alpha_n}$ входит в многочлен f с ненулевым коэффициентом. Многогранником Ньютона $\mathcal{N}_f$ многочлена Лорана f называется выпуклая оболочка (в пространстве $\mathbb{R}^n$) носителя многочлена f . Следующий результат показывает, что многогранник Ньютона $\mathcal{N}_f$ несет в себе информацию о геометрии амебы $\mathcal{A}_f$ [23, теорема 2.8 и предложение 2.6].

Теорема 2 (см. [23]). *Пусть f — многочлен Лорана. Обозначим через $\{M\}$ семейство связных компонент дополнения к амебе ${}^c\mathcal{A}_f$. Существует инъективное отображение*

$$v_f : \{M\} \rightarrow \mathbb{Z}^n \cap \mathcal{N}_f, \quad (2.1)$$

такое, что конус, двойственный к $\mathcal{N}_f$ в точке $v_f(M)$, совпадает с конусом рецессии множества M .

Целочисленный вектор $v_f(M)$ называется порядком компоненты M дополнения к амебе многочлена f . Из приведенных теорем следует, что число связных компонент дополнения к амебе $\mathcal{A}_f$ ограничено снизу числом вершин многогранника $\mathcal{N}_f$ и сверху — числом целых точек в данном многограннике. Нижняя граница была получена в [2]. Варьируя коэффициенты многочлена Лорана f с заданным многогранником Ньютона, можно добиться достижения как верхней [10], так и нижней [13] границы для числа связных компонент множества ${}^c\mathcal{A}_f$. Более того, вершины многогранника Ньютона всегда лежат в образе отображения v_f . Из теоремы 2 следует, что конусы тех связных

компонент множества ${}^c\mathcal{A}_f$, которые соответствуют вершинам многогранника $\mathcal{N}_f$, заведомо имеют непустую внутренность. В то же время связные компоненты дополнения амебы, соответствующие внутренним точкам многогранника Ньютона, являются ограниченными.

Пусть $f(\zeta) = a_0 + a_1\zeta + a_2\zeta^2 + \dots + a_{m-1}\zeta^{m-1} + \zeta^m$ — многочлен одного переменного $\zeta \in \mathbb{C}$ с корнями $\zeta_1, \dots, \zeta_m$. Будем без ограничения общности предполагать, что $|\zeta_1| \leq |\zeta_2| \leq \dots \leq |\zeta_m|$. Амеба многочлена f есть конечное множество точек $\ln|\zeta_1|, \dots, \ln|\zeta_m|$, из которого исключены повторяющиеся элементы. Типичная компонента дополнения ${}^c\mathcal{A}_f$ к амебе данного многочлена есть (непустой) интервал $(\ln|\zeta_k|, \ln|\zeta_{k+1}|)$ для некоторого $k = 1, \dots, m-1$, порядок данной компоненты равен k . Таких компонент может, однако, не быть вовсе, в случае, если все корни многочлена $f(\zeta)$ лежат на некоторой окружности $\{|\zeta| = r\} \subset \mathbb{C}$ с центром в точке $\zeta = 0$. Кроме того, всегда присутствуют две неограниченные компоненты дополнения $(-\infty, \ln|\zeta_1|)$ и $(\ln|\zeta_m|, \infty)$, чьи порядки равны 0 и m , соответственно. Таким образом, в одномерном случае вычисление амебы многочлена сводится к нахождению его корней. Для решения данной задачи существует широкий спектр точных и приближенных методов и в настоящей работе мы не рассматриваем этот частный случай, предполагая всюду в дальнейшем, что $n \geq 2$.

Задача вычисления амебы многочлена двух или более переменных может, конечно, рассматриваться как задача разрешения алгебраического уравнения относительно одного из входящих в него переменных. Ключевое отличие от одномерного случая состоит в том, что при $n \geq 2$ алгебраическое уравнение необходимо решать для всевозможных значений прочих присутствующих в нем переменных. В общем случае осуществить это не представляется возможным, что заставляет обращаться к другим методам вычисления амеб многочленов нескольких переменных.

В некоторых простейших случаях амеба многочлена может быть вычислена точно, а ее граница явно задана трансцендентными уравнениями. Например, амеба многочлена двух переменных $f(x, y) = 1 + x + y$ имеет вид

$$\{(s, t) \in \mathbb{R}^2 : 1 \leq e^s + e^t, e^s - 1 \leq e^t \leq e^s + 1\}.$$

Аналогичное описание допускает амеба любой комплексной гиперплоскости в произвольной размерности. Амеба многочлена $f(x, y) = a + x + y + xy$ с произвольным комплексным параметром a есть множество решений неравенства (см. [13, стр. 57])

$$(e^{2s} - |a|^2)^2 + (e^{2s} - 1)^2 e^{4t} - 2e^{2t}(e^{2s}(|a|^2 - 4\Re a + e^{2s} + 1) + |a|^2) \leq 0.$$

Сложность вычисления границы амобы многочлена быстро растет с увеличением его степени и числа переменных. Несмотря на это, контуры амоб произвольных классических и обобщенных дискриминантов [2] допускают бирациональную параметризацию Горна–Капранова [24]. Однако амоба многочлена достаточно высокой степени, чьи коэффициенты не обладают многочисленными свойствами симметрии, допускает, вообще говоря, лишь приближенное описание. Построение множества, аппроксимирующего амобу с заданной точностью, является задачей высокой вычислительной сложности.

Под вычислением амобы многочлена Лорана f мы будем всюду в дальнейшем понимать построение достаточно плотного облака точек, лежащих в пересечении амобы с заданной областью $D \subset \mathbb{R}^n$, такой, что D содержит точки любой связной компоненты дополнения ${}^c\mathcal{A}_f$.

3. АЛГОРИТМ ВЫЧИСЛЕНИЯ АМЕБЫ МНОГОЧЛЕНА НА ОСНОВЕ ПОРЯДКА КОМПОНЕНТЫ ЕЕ ДОПОЛНЕНИЯ

В работе [23] показано, что каждой точке дополнения к амобе $\mathcal{A}_f$ многочлена f можно сопоставить целую точку из многогранника Ньютона $\mathcal{N}_f$ с помощью отображения порядка v_f :

$$v_f : {}^c\mathcal{A}_f \rightarrow \mathcal{N}_f \cap \mathbb{Z}^n, \quad s \mapsto (u_1, \dots, u_n),$$

где

$$u_j := \frac{1}{(2\pi i)^n} \int_{\text{Log } \mathbf{x} = \mathbf{s}} \frac{x_j \partial_j f(\mathbf{x}) dx_1 \dots dx_n}{f(\mathbf{x}) x_1 \dots x_n}, \quad (3.1)$$

для всех $j = 1, \dots, n$ и $i = \sqrt{-1}$. Всяду в дальнейшем вектор $v_f(\mathbf{s})$ называется *порядком точки* $\mathbf{s} \in {}^c\mathcal{A}_f$. Любые две точки, лежащие в одной связной компоненте дополнения к амобе многочлена f , имеют одинаковый порядок, называемый порядком этой компоненты (см. формулу (2.1) в Теореме 2); более того, порядки точек из разных компонент дополнения к амобе различны. Таким образом, порядок $v_f(\mathbf{s})$ является классификатором точек дополнения к амобе многочлена f , позволяющим устанавливать их принадлежность к компоненте заданного порядка.

В работе [23] показано, что для $\mathbf{s} \in {}^c\mathcal{A}_f$ интеграл (3.1) допускает представление в следующем существенно более простом виде:

$$u_j = \frac{1}{2\pi i} \int_{|x_j|=e^{s_j}} \frac{\partial_j f(\mathbf{x})}{f(\mathbf{x})} dx_j, \quad (3.2)$$

причем значение данного интеграла не зависит от $\arg x_k$ при $k \neq j$. Поведение интеграла (3.2) в случае, когда точка $\mathbf{s}$ лежит внутри или на границе амобы $\mathcal{A}_f$, является, вообще говоря, весьма сложным. В этом случае интеграл (3.2) может оказаться расходящимся, однако в большинстве случаев он сходится, а его значение существенным образом зависит от $\arg x_k$ для всех $k = 1, \dots, n$. Скачок интеграла (3.2) может быть выявлен с помощью символично-численных алгоритмов контурного интегрирования и использован в качестве критерия принадлежности точки $\mathbf{s}$ амобе многочлена f . Данное наблюдение является основой алгоритма вычисления амоб, представленного в настоящей работе и реализованного в виде сетевого сервиса <http://amoebas.ru/>.

Входные данные алгоритма: размерность пространства переменных $n \geq 2$; многочлен $f(\mathbf{x})$ от n комплексных переменных $x_1, \dots, x_n$; множество вершин $\text{vert}(\Pi)$ прямоугольного параллелепипеда $\Pi \subset \mathbb{R}^n$; малое число $\varepsilon > 0$, равное линейному размеру наименьшего элемента объема в $\mathbb{R}^n$, подлежащего проверке на предмет принадлежности амобе многочлена $f(\mathbf{x})$. Предполагается, что объем параллелепипеда Π не равен нулю, а длина по крайней мере одного из его ребер больше ε .

Результатом работы алгоритма является облако точек, аппроксимирующее множество ${}^c\mathcal{A}_f \cap \Pi$ с точностью ε .

Схема работы алгоритма состоит в следующем. Будем обозначать через $S(\Pi)$ конечное множество прямоугольных параллелепипедов в пространстве $\mathbb{R}^n$ и положим на первом шаге работы алгоритма $S(\Pi) := \{\Pi\}$.

Пусть Π_α — некоторый элемент множества $S(\Pi)$. Если длина всех его ребер меньше ε , то Π_α исключается из множества $S(\Pi)$ и осуществляется переход к следующему его элементу. В противном случае каждая из точек множества $\text{vert}(\Pi_\alpha)$ проверяется на принадлежность амобе многочлена $f(\mathbf{x})$ путем выявления скачка интеграла (3.2) по его параметрам $\arg x_k$, $k \neq j$.

Если все точки множества $\text{vert}(\Pi_\alpha)$ лежат в одной и той же связной компоненте дополнения к амобе $\mathcal{A}_f$, то, в силу выпуклости и открытости любой такой компоненты, делается вывод об отсутствии точек амобы $\mathcal{A}_f$ в Π_α . Координаты точек $\text{vert}(\Pi_\alpha)$ и порядок содержащей их компоненты дополнения к амобе сохраняются, и выполняется

переход к следующему элементу множества $S(\Pi)$. Если рассмотрены все элементы данного множества, то алгоритм завершает работу.

Если же множество $\text{vert}(\Pi_\alpha)$ содержит точки амобы $\mathcal{A}_f$ или точки, лежащие в разных компонентах ее дополнения, то параллелепипед Π_α разбивается на 2^n равных параллелепипедов своими n плоскостями симметрии, параллельными координатным плоскостям в $\mathbb{R}^n$. Эти параллелепипеды добавляются к множеству $S(\Pi)$ и описанный выше процесс повторяется, пока не будут рассмотрены все элементы данного множества. В двумерном случае результатом выполнения данного дихотомического (по каждой размерности) процесса является аппроксимирующее амобу квадродерево, см. рис. 1.

По завершении перебора элементов множества $S(\Pi)$ для каждого возможного значения порядка компоненты дополнения к амобе $\mathcal{A}_f$ сформирован список точек (вершин рассмотренных параллелепипедов), лежащих в компоненте с данным порядком. Для отрисовки этой компоненты берется выпуклая оболочка всех точек из данного списка.

Блок-схема изложенного выше алгоритма представлена на рис. 2. Данный алгоритм реализован на языке программирования C# и лежит в основе вычислительного функционала сетевого сервиса <http://amoebas.ru/>. Объем связной компоненты дополнения к амобе может быть сколь угодно мал, а число таких компонент, вообще говоря, весьма сложным образом зависит от коэффициентов многочлена. Эта зависимость не является непрерывной. Например, дополнение к амобе многочлена $x + y + \lambda xy + x^2y + xy^2$ при $\lambda > 0$ содержит компоненту порядка (1,1) в том и только том случае, когда $\lambda > 4$ (см. [22]). При этом площадь данной компоненты стремится к нулю при $\lambda \rightarrow 4 +$. В силу этих причин предложенный алгоритм не может, вообще говоря, гарантировать обнаружение связной компоненты, чей диаметр меньше величины ϵ . Более того, мера подмножества в пространстве аргументов комплексных переменных, на котором наблюдается скачок интеграла (3.2), может быть сколь угодно мала, если степень многочлена достаточно высока, а его коэффициенты велики. Однако в некоторых классах частных случаев скорость работы предложенного алгоритма допускает оценку снизу. Например, в случае, когда f — оптимальный гипергеометрический многочлен n комплексных переменных, алгоритм позволяет не более чем за $n2^{n-1} \text{Vol}(\Pi)(1 + 1/\epsilon)^n$ шагов, представленных на блок-схеме 2, выявить все связные компоненты дополнения к амобе $\mathcal{A}_f$, лежащие в параллелепи-

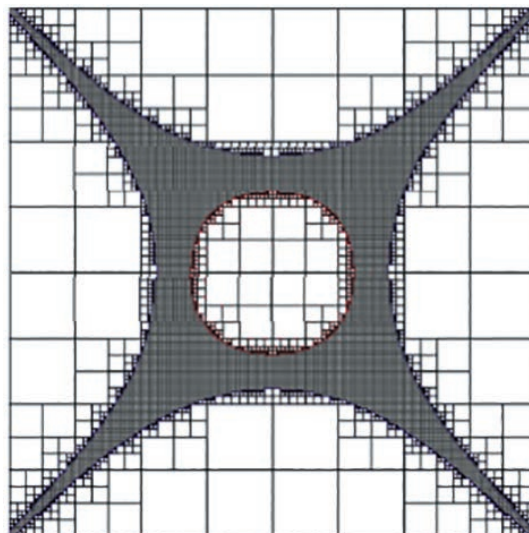


Рис. 1. Квадродерево в дополнении к амобе многочлена $x + y + x^2y + xy^2 + 7xy$.

пипе Π , и такие, что все их линейные размеры не меньше ϵ . Данное обстоятельство является следствием строгой логарифмической выпуклости функции, определяющей коэффициенты такого многочлена, благодаря которой проверку скачка интеграла (3.2) достаточно осуществлять в двух точках в пространстве аргументов комплексных переменных многочлена f .

4. СЕТЕВОЙ СЕРВИС ДЛЯ ВЫЧИСЛЕНИЯ И ИЗОБРАЖЕНИЯ АМЕБ МНОГОЧЛЕНОВ ДВУХ И ТРЕХ ПЕРЕМЕННЫХ

Визуализация амобы многочлена трех переменных достаточно высокой степени, вычисленной в виде облака точек, является в большинстве случаев концептуально и технически нетривиальной задачей ввиду сложной геометрической структуры амобы: наличия в “теле” амобы “полостей” и уходящих на бесконечность параллельных слоев, а также возможности присутствия на границе амобы подмножеств негладкости границы с труднопредсказуемой структурой. В силу этих причин в сетевом сервисе http://amoebas.ru/amoeba_3d.html визуализация амобы многочлена f трех переменных осуществляется путем отрисовки пересечений всех компонент дополнения $^c\mathcal{A}_f$ с кубом $[-a, a] \times [-a, a] \times [-a, a]$ для некоторого значения $a > 0$. По умолчанию все эти компоненты отрисованы полупрозрачными, при этом каждая из них может быть сделана непрозрачной путем ее выбора с помощью левой кнопки мыши. Сама же амоба $\mathcal{A}_f$ представлена как пустое пространство между компонентами своего

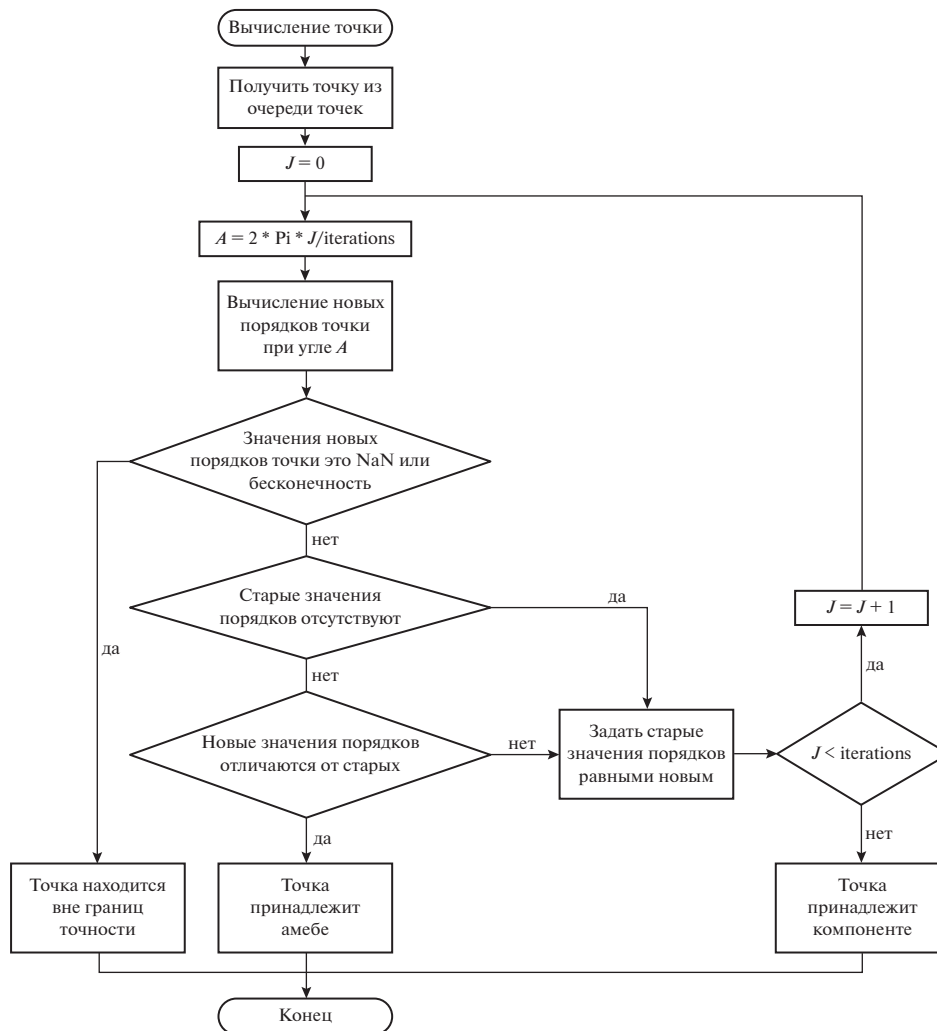


Рис. 2. Упрощенная схема работы алгоритма классификации точек пространства по принадлежности к амebe и связным компонентам ее дополнения всех возможных порядков.

дополнения. Визуализация амeb осуществляется с помощью пакета интерактивной трехмерной графики `plotly` (см. <https://plotly.com/>), обеспечивающего возможность приближения, удаления и вращения амebы, а также движения “наблюдателя” внутри нее.

Различные компоненты дополнения к амebe окрашены в различные цвета. Длина волны, соответствующая цвету связной компоненты дополнения к амebe многочлена f двух переменных, пропорциональна евклидову расстоянию от порядка компоненты до центра тяжести многоугольника $\mathcal{N}_f$ (см. рис. 3).

Тестирование алгоритмов вычисления амeb и оценку скорости их работы целесообразно проводить на многочленах, чьи амebы имеют предсказуемую, но при этом достаточно сложную структуру. Для этой цели хорошо подходят т.н. оптимальные гипергеометрические многочлены

нескольких переменных [22]: число связных компонент дополнения к амebe такого многочлена является максимально возможным, то есть, совпадает с числом целых точек в его многограннике Ньютона. Для генерации оптимальных гипергеометрических многочленов от трех переменных, амebы которых представлены на общедоступном сетевом ресурсе http://amoebas.ru/amoeba_3d.html, был использован, в частности, следующий программный код в системе компьютерной алгебры `Mathematica 11.3`:

```

OptimalPolynomialSimplex[dimension_,
  degree_, power_] :=
Block[{variables, coefficient, exponents},
variables=Table[x_j, {j, 1, dimension}];
coefficient[vector_] :=
(Gamma[1+degree-Total[vector]]

```

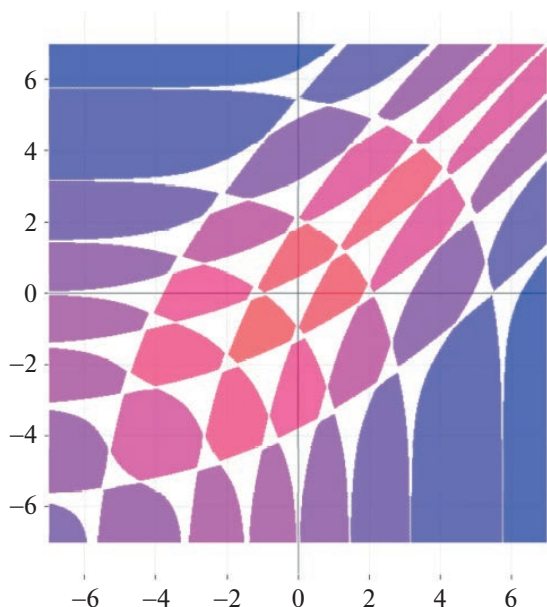


Рис. 3. Связные компоненты дополнения к амбе многочлена $x^7 + 343x^6y + 343x^6 + 9261x^5y^2 + 74088x^5y + 9261x^5 + 42875x^4y^3 + 1157625x^4y^2 + 1157625x^4y + 42875x^4 + 42875x^3y^4 + 2744000x^3y^3 + 9261000x^3y^2 + 2744000x^3y + 42875x^3 + 9261x^2y^5 + 1157625x^2y^4 + 9261000x^2y^3 + 9261000x^2y^2 + 1157625x^2y + 9261x^2 + 343xy^6 + 74088xy^5 + 1157625xy^4 + 2744000xy^3 + 1157625xy^2 + 74088xy + 343x + y^7 + 343y^6 + 9261y^5 + 42875y^4 + 42875y^3 + 9261y^2 + 343y + 1$.

```
(Times @@ Gamma[vector + 1]))^-power;
exponents=Flatten[Outer @@
(Join[List, Table[Range[0, degree],
dimension]]), dimension-1];
Factor[(coefficient /@ exponents).
(Times @@@ (variables# & /@ expo-
nents))]]]
```

Результатом работы функции `OptimalPolynomialSimplex` является оптимальный [22] многочлен степени `degree` с заданным числом переменных, равным `dimension`. Коэффициенты данного многочлена заданы сужением строго логарифмически вогнутой функции на множество целых точек симплекса с вершинами $(0,0,\dots,0)$, $(\text{degree}, 0,\dots,0)$, $\dots$, $(0,\dots,0, \text{degree})$. Технический параметр `power` позволяет управлять “степенью логарифмической вогнутости” коэффициентов генерируемого многочлена, отвечающей за геометрию его амобы. Непрерывная деформация амоб, соответствующая изменению данного параметра в пределах некоторых интервалов, представлена на сайте <http://amoe-bas.ru/>. Примерами многочленов, генерируе-

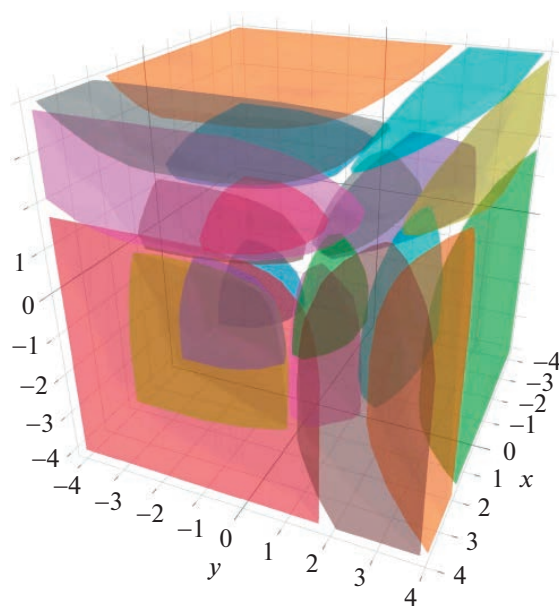


Рис. 4. Связные компоненты дополнения к амбе многочлена $x^3 + 9x^2y + 9x^2z + 9x^2 + 9xy^2 + 36xyz + 36xy + 9xz^2 + 36xz + 9x + y^3 + 9y^2z + 9y^2 + 9yz^2 + 36yz + 9y + z^3 + 9z^2 + 9z + 1$.

мых функций `OptimalPolynomialSimplex`, являются многочлены, чьи амобы представлены на рис. 4 и 5.

Обширный класс выпуклых многогранников, важных в многочисленных приложениях алгебраической и вычислительной геометрии, образуют зонотопы. Напомним, что под зонотопом понимается сумма Минковского конечного числа отрезков в вещественном пространстве. Следующий фрагмент программного кода на языке системы компьютерной алгебры `Mathematica 11.3` предназначен для генерации оптимального гипергеометрического многочлена с носителем в произвольном зонотопе с вершинами в узлах целочисленной решетки, заданном как сумма Минковского векторов из списка `vectors`.

```
OptimalPolyZonotope[vectors_, power_] :=
Block[{dimension, variables, crossProducts,
normals, sumsOfVectors, massCenter,
constants, coefficient, maxDegree, ex-
ponents},
dimension=Length[First[vectors]];
variables=Table[x_, {j, 1, dimension}];
crossProducts=Cross @@@ Subsets[vectors,
{dimension-1}];
normals=Join[crossProducts, -
crossProducts];
sumsOfVectors=
```

Таблица 1.

n	degree	power	Многочлен от n комплексных переменных	Число уровней 2^n -дерева	Время расчета, с.
2	2	2	$x^2 + 4xy + 4x + y^2 + 4y + 1$	9	10.79
2	3	2	$x^3 + 9x^2y + 9x^2 + 9xy^2 + 36xy + 9x + y^3 + 9y^2 + 9y + 1$	9	29.46
2	4	2	$x^4 + 16x^3y + 16x^3 + 36x^2y^2 + 144x^2y + 36x^2 +$ $+16xy^3 + 144xy^2 + 144xy + 16x + y^4 + 16y^3 + 36y^2 + 16y + 1$	9	61.30
2	5	2	$x^5 + 25x^4y + 25x^4 + 100x^3y^2 + 400x^3y + 100x^3 + 100x^2y^3 +$ $+900x^2y^2 + 900x^2y + 100x^2 + 25xy^4 + 400xy^3 +$ $+900xy^2 + 400xy + 25x + y^5 + 25y^4 + 100y^3 + 100y^2 + 25y + 1$	9	106.66
3	2	2	$x^2 + 4xy + 4xz + 4x + y^2 + 4yz + 4y + z^2 + 4z + 1$	7	202.52
3	6	1	$2x^2y^2z^2 + x^2y^2z + x^2yz^2 + 2x^2yz +$ $+xy^2z^2 + 2xy^2z + 2xyz^2 + 16xyz + 2xy + 2xz + x + 2yz + y + z + 2$	7	268.14
3	3	2	$x^3 + 9x^2y + 9x^2z + 9x^2 + 9xy^2 + 36xyz +$ $+36xy + 9xz^2 + 36xz + 9x + y^3 + 9y^2z + 9y^2 + 9yz^2 +$ $+36yz + 9y + z^3 + 9z^2 + 9z + 1$	7	460.32
3	4	2	$x^4 + 16x^3y + 16x^3z + 16x^3 + 36x^2y^2 + 144x^2yz + 144x^2y + 36x^2z^2 +$ $+144x^2z + 36x^2 + 16xy^3 + 144xy^2z + 144xy^2 +$ $+144xyz^2 + 576xyz + 144xy + 16xz^3 + 144xz^2 + 144xz +$ $+16x + y^4 + 16y^3z + 16y^3 + 36y^2z^2 + 144y^2z + 36y^2 + 16yz^3 +$ $+144yz^2 + 144yz + 16y + z^4 + 16z^3 + 36z^2 + 16z + 1$	7	894.35
3	5	2	$x^5 + 25x^4y + 25x^4z + 25x^4 + 100x^3y^2 + 400x^3yz + 400x^3y + 100x^3z^2 +$ $+400x^3z + 100x^3 + 100x^2y^3 + 900x^2y^2z + 900x^2y^2 + 900x^2yz^2 + 3600x^2yz +$ $+900x^2y + 100x^2z^3 + 900x^2z^2 + 900x^2z + 100x^2 + 25xy^4 + 400xy^3z +$ $+400xy^3 + 900xy^2z^2 + 3600xy^2z + 900xy^2 + 400xyz^3 + 3600xyz^2 + 3600xyz +$ $+400xy + 25xz^4 + 400xz^3 + 900xz^2 + 400xz + 25x + y^5 + 25y^4z + 25y^4 +$ $+100y^3z^2 + 400y^3z + 100y^3 + 100y^2z^3 + 900y^2z^2 + 900y^2z + 100y^2 +$ $+25yz^4 + 400yz^3 + 900yz^2 + 400yz + 25y + z^5 + 25z^4 +$ $+100z^3 + 100z^2 + 25z + 1$	7	1558.12
4	2	2	$w^2 + 4wx + 4wy + 4wz + 4w + x^2 + 4xy + 4xz +$ $+4x + y^2 + 4yz + 4y + z^2 + 4z + 1$	5	1438.40
4	3	2	$w^3 + 9w^2x + 9w^2y + 9w^2z + 9w^2 + 9wx^2 + 36wxy + 36wxz + 36wx + 9wy^2 +$ $+36wyz + 36wy + 9wz^2 + 36wz + 9w + x^3 + 9x^2y + 9x^2z + 9x^2 + 9xy^2 +$ $+36xyz + 36xy + 9xz^2 + 36xz + 9x + y^3 + 9y^2z + 9y^2 + 9yz^2 +$ $+36yz + 9y + z^3 + 9z^2 + 9z + 1$	5	4574.97

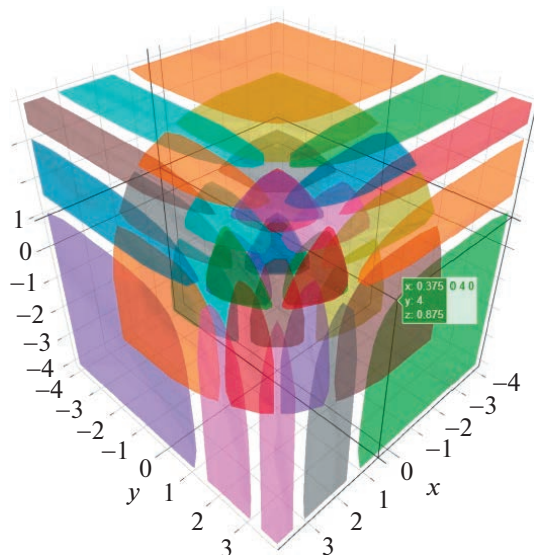


Рис. 5. Связные компоненты дополнения к амбе

многочлена $x^4 + 16x^3y + 16x^3z + 16x^3 + 36x^2y^2 + 144x^2yz + 144x^2y + 36x^2z^2 + 144x^2z + 36x^2 + 16xy^3 + 144xy^2z + 144xy^2 + 144xyz^2 + 576xyz + 144xy + 16xz^3 + 144xz^2 + 144xz + 16x + y^4 + 16y^3z + 16y^3 + 36y^2z^2 + 144y^2z + 36y^2 + 16yz^3 + 144yz^2 + 144yz + 16y + z^4 + 16z^3 + 36z^2 + 16z + 1$.

```
Join[Table[0,dimension],Union[Total /@
Drop[Subsets[vectors],1]]];
massCenter=
Total[sumsOfVectors]/Length[sumsOf
Vectors];
distanceFromSideToMassCenter[side-
Normal_]:=
Max[Norm /@ ((#.sideNormal)
sideNormal/Norm[sideNormal]^2 & /@
(# - massCenter & /@ sumsOfVec-
tors))];
constants=
distanceFromSideToMassCen-
ter[#]Norm[#] & /@ normals;
coefficient[vec_]:=
Simplify[(Times @@ Gamma /@
(1+constants +((vec+massCen-
ter).# & /@
normals)))^-power];
maxDegree=Abs[Max[Max /@ (Total /@
Subsets[vectors])]];
exponents=
Flatten[Outer@@(Join[List,
Table[Range[-maxDegree,maxDegree],
```

```
dimension]]),dimension-1];
Numerator[
Factor[
(coefficient /@ exponents).
(Times @@@ (variables^# & /@ expo-
nents))
]]]
```

Результаты расчета и визуализации амб оптимальных гипергеометрических многочленов с носителями в зонотопах представлены на сайте <http://amoebas.ru/>. Таблица 1 содержит данные о скорости расчета амб многочленов двух, трех и четырех переменных, генерируемых с помощью приведенного выше программного кода, для различных значений размерности (dimension), степени многочлена (degree), степени логарифмической вогнутости его коэффициента (power) и числа уровней квадродерева, с помощью которого аппроксимируется амба. Расчеты выполнялись на компьютере с центральным процессором Intel Xeon Gold 6146 с тактовой частотой 3.20 ГГц и 128 Гб оперативной памяти.

5. ИСТОЧНИК ФИНАНСИРОВАНИЯ

Исследование выполнено за счет средств гранта Российского научного фонда № 22-21-00556, <https://rscf.ru/project/22-21-00556/>.

СПИСОК ЛИТЕРАТУРЫ

1. Абрамов С.А., Рябенко А.А., Хмельнов Д.Е. Лорановы, рациональные и гипергеометрические решения линейных q -разностных систем произвольного порядка с полиномиальными коэффициентами // Программирование, 2018, No 2. С. 60–73.
2. Gelfand I.M., Kapranov M.M., Zelevinsky A.V. Discriminants, resultants, and multidimensional determinants. Birkhäuser, 1994.
3. Viro O. What is an amoeba? // Notices of the AMS. 2002. V. 49. Issue 8. P. 916–917.
4. Cherkis S.A., Ward R.S. Moduli of monopole walls and amoebas // J. High Energy Physics. 2012. Issue 5. 90.
5. Fujimori T., Nitta M., Ohta K., Sakai N., Yamazaki M. Intersecting solitons, amoeba, and tropical geometry // Physical Review D – Particles, Fields, Gravitation, and Cosmology. 2008. V. 78. Issue 10. 105004.
6. Kenyon R., Okounkov A., Sheffield S. Dimers and amoebae // Ann. Math. 2006. V. 163. P. 1019–1056.
7. Passare M., Pochekutov D., Tsikh A. Amoebas of complex hypersurfaces in statistical thermodynamics // Mathem. Physics, Analysis, and Geometry. 2013. V. 16. Issue 1. P. 89–108.
8. Zahabi A. Quiver asymptotics and amoeba: Instantons on toric Calabi–Yau divisors // Physical Review D. 2021. V. 103. Issue 8. 086024.
9. Maeda T., Nakatsu T. Amoebas and instantons // International Journal of Modern Physics A. 2007. V. 22. Issue 5. P. 937–983.

10. *Mikhalkin G.* Real algebraic curves, the moment map and amoebas // *Ann. Math.* 2000. V. 151. Issue 2. P. 309–326.
11. *Forsberg M.* Amoebas and Laurent series. 1998. Doctoral thesis presented at Royal Institute of Technology (KTH), Stockholm. ISBN 91-7170-259-8.
12. *Leksell M., Komorowski W.* Amoeba Program: Computing and visualizing amoebas for some complex-valued bivariate expressions // <http://qrf.servequake.com/amoeba/AmoebaProgram.pdf>
13. *Rullgård H.* Topics in geometry, analysis, and inverse problems. 2003. Doctoral thesis presented at Stockholm University. ISBN 91-7265-738-3. <http://www.diva-portal.org/smash/get/diva2:190169/FULLTEXT01.pdf>
14. *Theobald T.* Computing amoebas // *Experimental Math.* 2002. V. 11. Issue 4. P. 513–526.
15. *Timme S.* A package to compute amoebas in 2 and 3 variables // <https://github.com/saschatimme/PolynomialAmoebas.jl>
16. *Theobald T., De Wolff T.* Approximating amoebas and coamoebas by sums of squares // *Math. of Computation.* 2015. V. 84(291). P. 455–473.
17. *Purbhoo K.* A Nullstellensatz for amoebas // *Duke Math. J.* 2008. V. 141. Issue 3. P. 407–445.
18. *Forsgård J., Matusevich L.F., Mehlhop N., De Wolff T.* Lopsided approximation of amoebas // *Math. of Computation.* 2018. V. 88. P. 485–500.
19. *Anthony E., Grant S., Gritzmman P., Rojas J.M.* Polynomial-time amoeba neighborhood membership and faster localized solving // *Mathematics and Visualization.* 2015. V. 38. P. 255–277.
20. *Bogdanov D.V., Kytmanov A.A., Sadykov T.M.* Algorithmic computation of polynomial amoebas // *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* 2016. V. 9890. P. 87–100.
21. *Nisse M., Sadykov T.M.* Amoeba-shaped polyhedral complex of an algebraic hypersurface // *J. Geom. Analysis.* 2019. V. 29. Issue 2. P. 1356–1368.
22. *Bogdanov D.V., Sadykov T.M.* Hypergeometric polynomials are optimal // *Math. Z.* 2020. V. 296. Issue 1–2. P. 373–390.
23. *Forsberg M., Passare M., Tsikh A.K.* Laurent determinants and arrangements of hyperplane amoebas // *Adv. Math.* 2000. V. 151. P. 45–70.
24. *Klausen R.P.* Kinematic singularities of Feynman integrals and principal A-determinants // *J. High Energy Physics.* 2022. Issue. 2. 4.

УДК 004.421.6

СИМВОЛЬНО-ЧИСЛЕННАЯ РЕАЛИЗАЦИИ МЕТОДА ГАЛЕРКИНА ДЛЯ ПРИБЛИЖЕННОГО РЕШЕНИЯ ЗАДАЧИ ВОЛНОВОДНОЙ ДИФРАКЦИИ

© 2023 г. Д. В. Диваков^{a,b,*}, А. А. Тютюнник^{a,b,**}^aРоссийский университет дружбы народов
117198 Москва, ул. Миклухо-Маклая, д. 6, Россия^bОбъединенный институт ядерных исследований
141980 г. Дубна, Московская обл., ул. Жолио-Кюри, д. 6, Россия

*E-mail: divakov-dv@rudn.ru

**E-mail: tyutyunnik-aa@rudn.ru

Поступила в редакцию 31.08.2022 г.

После доработки 16.10.2022 г.

Принята к публикации 30.10.2022 г.

В работе построена символьно-численная реализация метода Галеркина для приближенного решения задачи волноводной дифракции на стыке двух открытых планарных трехслойных волноводов. Метод Гелеркина реализован в системе компьютерной алгебры Maple с использованием символьных манипуляций, основа программной реализации – символьно-численная процедура `scprod`, реализующая численный расчет скалярных произведений метода Галеркина на основе символьных выражений. Использование символьных манипуляций позволяет ускорить расчет интегралов в методе Галеркина благодаря однократному символьному расчету типовых для задачи интегралов вместо многократного численного интегрирования.

DOI: 10.31857/S0132347423020097, EDN: MFUGDU

1. ВВЕДЕНИЕ

В работе рассматривается построение символьно-численной реализации метода Галеркина в системе компьютерной алгебры для приближенного решения задачи волноводной дифракции на стыке двух открытых планарных трехслойных волноводов. Самая ресурсозатратная процедура в методе Галеркина – это вычисление большого числа скалярных произведений элементов функционального пространства. Каждое скалярное произведение представляет собой интеграл от произведения двух функций, причем для планарных открытых волноводов эти функции являются кусочно-гладкими собственными функциями задачи Штурма–Лиувилля, заданными в символьно-численном виде: в символьном виде задана фундаментальная система решений (ФСР) дифференциального уравнения с постоянным коэффициентом в каждом слое волновода, а в численном виде заданы только коэффициенты разложения по ФСР. Различные коэффициенты при ФСР определяют разные собственные функции. Если численные константы заменить на символьные, то интегралы от произведения собственных функций можно вычислить однократно в каждом слое волновода в виде символьного вы-

ражения, а после подставлять в полученные символьные выражения численные значения констант. В таком подходе вычисление интегралов сведено к минимуму для ускорения численных расчетов.

Описанный подход реализован в системе компьютерной алгебры Maple для решения задачи волноводной дифракции на стыке двух открытых планарных трехслойных волноводов. Собственные функции вычисляются в символьно-численном виде методом волнового сопряжения, который реализован в виде процедуры `wave_conj` в Maple. Используя собственные функции применяем процедуру `scprod` для численного расчета матрицы скалярных произведений метода Галеркина. Матрица скалярных произведений используется для формирования системы уравнений, которая решается численно в Maple методом `LinearSolve` пакета `LinearAlgebra`.

С помощью разработанного символьно-численного метода исследована сходимость метода Галеркина для задачи волноводной дифракции.

2. МОТИВАЦИЯ

Корректная постановка задачи волноводной дифракции на стыке двух планарных открытых волноводов может быть сформулирована следующим образом. Имеется стык двух планарных открытых волноводов, который для корректной постановки задачи помещаем в объемлющий закрытый волновод аналогично подходу [1].

В таком случае компонента $E_y(x, z) = u(x, z)$ удовлетворяет уравнению Гельмгольца [2, 3]

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} + k_0^2 n^2(x, z) \right) u \equiv (\Delta + k_0^2 n^2) u = 0, \quad (1)$$

условиям на границе объемлющего закрытого волновода [2, 3]

$$u|_{x=\pm L} = 0, \quad (2)$$

и условиям сопряжения на стыке волноводов [4]

$$[u]_{z=0} = 0, \quad \left[\frac{\partial u}{\partial z} \right]_{z=0} = 0, \quad (3)$$

где $[f]_{s=a} = f|_{s=a-0} - f|_{s=a+0}$. Кусочно-постоянный коэффициент $n^2(x, z)$ в уравнении Гельмгольца определяет показатель преломления в рассматриваемой области: $n^2(x, z)|_{z<0} = n_1^2(x)$, $n^2(x, z)|_{z>0} = n_2^2(x)$, где

$$n_l^2(x) = \begin{cases} n_c^2, & x > h_l \\ n_f^2, & 0 < x < h_l \\ n_s^2, & x < 0 \end{cases} \quad (4)$$

Применяя метод разделения переменных к уравнению Гельмгольца в области $z < 0$ и в области $z > 0$ получаем общий вид решения в каждой из областей в виде разложения по нормальным модам

$$\begin{aligned} u|_{z<0} &= \sum_{j=0}^{\infty} A_j \varphi_j(x) e^{ik_0 \beta_j z} + \sum_{j=0}^{\infty} R_j \varphi_j(x) e^{-ik_0 \beta_j z}, \\ u|_{z>0} &= \sum_{j=0}^{\infty} T_j \psi_j(x) e^{ik_0 \gamma_j z} + \sum_{j=0}^{\infty} S_j \psi_j(x) e^{-ik_0 \gamma_j z}, \end{aligned} \quad (5)$$

где $\varphi_j(x)$ и β_j ортонормированные собственные функции и собственные значения задачи Шурма–Лиувилля при $l = 1$

$$\begin{cases} v'' + k_0^2 n_l^2(x) v = k_0^2 \eta^2 v, \\ [v]_{x=0, h_l} = 0, \quad [v']_{x=0, h_l} = 0, \\ v(\pm L) = 0, \end{cases} \quad (6)$$

а $\psi_j(x)$ и γ_j – ортонормированные собственные функции и собственные значения приведенной выше задачи Шурма–Лиувилля при $l = 2$.

Постановка задачи дифракции формулируется следующим образом: в области $z < 0$ имеется одна падающая m -я мода с заданным амплитудным коэффициентом A и счетное количество отраженных волноводных мод, характеризующихся амплитудными коэффициентами отражения R_j [2]

$$u|_{z<0} = A \varphi_m(x) e^{ik_0 \beta_m z} + \sum_{j=0}^{\infty} R_j \varphi_j(x) e^{-ik_0 \beta_j z}, \quad (7)$$

а в области $z > 0$ имеется счетное количество прошедших волноводных мод, характеризующихся амплитудными коэффициентами прохождения T_j [2]

$$u|_{z>0} = \sum_{j=0}^{\infty} T_j \psi_j(x) e^{ik_0 \gamma_j z}. \quad (8)$$

Условие $[u]_{z=0} = 0$, учитывая вид решения (7) и (8), принимает следующий вид

$$\sum_{j=0}^{\infty} R_j \varphi_j(x) - \sum_{j=0}^{\infty} T_j \psi_j(x) = -A \varphi_m(x), \quad (9)$$

а условие $\left[\frac{\partial u}{\partial z} \right]_{z=0} = 0$ с учетом (7) и (8) принимает следующий вид

$$\sum_{j=0}^{\infty} \beta_j R_j \varphi_j(x) + \sum_{j=0}^{\infty} \gamma_j T_j \psi_j(x) = \beta_m A \varphi_m(x). \quad (10)$$

Задача волноводной дифракции на стыке двух планарных открытых волноводов, помещенных в объемлющий закрытый волновод, состоит в определении амплитудных коэффициентов отражения R_j и прохождения T_j по заданным характеристикам волноводного перехода, а также по заданной амплитуде A падающей m -й моды.

3. МЕТОДЫ

3.1. Приближенный метод решения задачи волноводной дифракции

Систему из двух уравнений (9), (10) решаем приближенно, для этого в каждой из бесконечных сумм оставляем конечное число слагаемых: в каждой сумме оставляем только те слагаемые, для которых выполняется $\beta_j > 0$ в первой сумме и $\gamma_j > 0$ во второй сумме соответственно, то есть незванесцентные моды. К усеченным суммам применяем метод Галеркина [5] и получаем систему линейных алгебраических уравнений следующего вида

$$\begin{cases} \vec{r} - \Phi \vec{t} = -\vec{e}_m, \\ D_1 \vec{r} + \Phi D_2 \vec{t} = D_1 \vec{e}_m, \end{cases} \quad (11)$$

где $\vec{r} = (R_0, R_1, \dots, R_N)^T$ и $\vec{t} = (T_0, T_1, \dots, T_M)^T$ – векторы искоемых величин, $D_1 = \text{diag}\{\beta_j\}_{j=0}^N$, $D_2 = \text{diag}\{\gamma_j\}_{j=0}^M$, $\vec{e}_m$ – вектор, m -я компонента которого равна A , а остальные компоненты нулевые. Матрица Φ определена следующим образом

$$\Phi = \begin{pmatrix} (\varphi_0, \psi_0) & (\varphi_0, \psi_1) & \dots & (\varphi_0, \psi_M) \\ (\varphi_1, \psi_0) & (\varphi_1, \psi_1) & \dots & (\varphi_1, \psi_M) \\ \vdots & \vdots & & \vdots \\ (\varphi_N, \psi_0) & (\varphi_N, \psi_1) & \dots & (\varphi_N, \psi_M) \end{pmatrix}, \quad (12)$$

где $(\varphi_p, \psi_q) = \int_{-L}^L \varphi_p(x) \psi_q(x) dx$.

Численно решаем систему линейных алгебраических уравнений (11) и определяем приближенные значения амплитудных коэффициентов отражения и прохождения. Численное решение системы не представляет трудностей, более ресурсоемкая операция в данном случае – это численный расчет матрицы Φ , каждый элемент которой представляет собой интеграл от произведения двух собственных функций. Используем возможности системы компьютерной алгебры Maple по символному расчету интегралов для символно-численной реализации метода Галеркина чтобы сократить общее время расчетов. Для этого в первую очередь отыскиваем собственные функции задачи (6) в символно-численном виде методом волнового сопряжения.

3.2. Символьно-численная реализация метода волнового сопряжения

Определяем собственные функции и собственные значения задачи Штурма–Лиувилля (6) методом волнового сопряжения: в каждой подобласти с постоянным показателем преломления записываем общее решение соответствующего обыкновенного дифференциального уравнения с постоянными коэффициентами

$$v_\alpha'' + k_0^2(n_\alpha^2 - \eta^2)v_\alpha = 0, \quad \alpha = c, f, s, \quad (13)$$

в виде разложения по фундаментальной системе решений $v_\alpha(x) = A_\alpha^1 v_\alpha^1(x) + A_\alpha^2 v_\alpha^2(x)$ с неизвестными коэффициентами $A_\alpha^{1,2}$. Общее решение $v_\alpha(x)$ для $\alpha = c, f, s$ подставляем в граничные условия и условия сопряжения из (6). Получается однородная система линейных алгебраических уравнений относительно неизвестных величин $A_\alpha^{1,2}$, $\alpha = c, f, s$, коэффициенты которой зависят от спектрального параметра η^2

$$M(\eta^2)\vec{A} = \vec{0}. \quad (14)$$

Система (14) имеет нетривиальное решение только тогда, когда $\det M(\eta^2) = 0$. Каждому корню η_j^2 этого уравнения соответствует однородная система $M_j \vec{A}_j = \vec{0}$, где $M_j = M(\eta_j^2)$. Решив систему, то есть отыскав вектор $\vec{A}_j$, мы определяем искоемые коэффициенты j -й моды $A_{\alpha,j}^{1,2}$, $\alpha = c, f, s$ тем самым полностью определяя и собственную функцию задачи, соответствующую собственному значению η_j^2 .

Численное отыскание корней нелинейного уравнения $\det M(\eta^2) = 0$ осуществляется методом деления отрезка пополам. Приближенное решение системы $M_j \vec{A}_j = \vec{0}$ с численно вырожденной матрицей M_j осуществляется с помощью численного отыскания собственного вектора матрицы M_j , отвечающего почти нулевому по модулю собственному значению. Алгоритм реализован в численном виде в системе компьютерной алгебры Maple с использованием метода Eigenvectors библиотеки LinearAlgebra [6], псевдокод алгоритма приведен ниже.

Algorithm 1: Algorithm wave_conj for numeric calculation η_j , $\vec{A}_j$

Input: $v_c^1(x)$, $v_c^2(x)$, $v_s^1(x)$, $v_s^2(x)$, $v_f^2(x)$ – symbolic expressions for each layer defined in eqs (15), (16), h – numeric value for thickness of the waveguide layer, L – numeric value for thickness of the “Dirichlet box”, n_c , n_s , n_f – numeric values for refractive indices

Output: array of η_j , array of $\vec{A}_j$

generate matrix for $\eta \in (n_s; n_f)$

GenerateMatrix $M_1(\eta)$

using $v_c^1(x)$, $v_f^2(x)$, $v_s^1(x)$ and conditions (6)

generate matrix for $\eta \in (n_c; n_s)$

GenerateMatrix $M_2(\eta)$

using $v_c^1(x)$, $v_f^2(x)$, $v_s^2(x)$ and conditions (6)

generate matrix for $\eta \in (0; n_c)$

GenerateMatrix $M_3(\eta)$

using $v_c^2(x)$, $v_f^2(x)$, $v_s^2(x)$ and conditions (6)

$\eta_j \leftarrow \text{RootOf}(\text{Determinant}(M_1(\eta)))$,

$\eta \in (n_s; n_f)$, $j = 1, \dots, N_1$

$\eta_j \leftarrow \text{RootOf}(\text{Determinant}(M_2(\eta)))$,

$\eta \in (n_c; n_s)$, $j = N_1 + 1, \dots, N_2$

```

 $\eta_j \leftarrow \text{RootOf}(\text{Determinant}(M_3(\eta))),$ 
 $\eta \in (0; n_c), j = N_2 + 1, \dots, N_3$ 
for  $j = 1, \dots, N_1$  do
     $e, v \leftarrow \text{Eigenvectors}(M_1(\eta_j));$ 
     $\text{indx\_min} \leftarrow \min[\text{index}](\text{abs} \sim (e));$ 
     $A_j \leftarrow \text{Column}(v, \text{indx\_min});$ 
for  $j = N_1 + 1, \dots, N_2$  do
     $e, v \leftarrow \text{Eigenvectors}(M_2(\eta_j));$ 
     $\text{indx\_min} \leftarrow \min[\text{index}](\text{abs} \sim (e));$ 
     $A_j \leftarrow \text{Column}(v, \text{indx\_min});$ 
for  $j = N_2 + 1, \dots, N_3$  do
     $e, v \leftarrow \text{Eigenvectors}(M_3(\eta_j));$ 
     $\text{indx\_min} \leftarrow \min[\text{index}](\text{abs} \sim (e));$ 
     $A_j \leftarrow \text{Column}(v, \text{indx\_min});$ 
return  $[A_j], [\eta_j], j = 1, \dots, N_3$ 

```

3.3. Символьно-численная реализация метода Галеркина

Общее решение уравнения (13) при $\eta > n_\alpha$ имеет следующий вид

$$v_\alpha^1(x) = A_\alpha^1 e^{-k_0 \sqrt{\eta^2 - n_\alpha^2} x} + A_\alpha^2 e^{k_0 \sqrt{\eta^2 - n_\alpha^2} x}, \quad (15)$$

а при $\eta < n_\alpha$

$$v_\alpha^2(x) = A_\alpha^1 \sin(k_0 \sqrt{n_\alpha^2 - \eta^2} x) + A_\alpha^2 \cos(k_0 \sqrt{n_\alpha^2 - \eta^2} x). \quad (16)$$

Применение метода Галеркина предполагает вычисление большого количества интегралов от произведения собственных функций вида

$$\begin{aligned} \int_{-L}^L \varphi_p \psi_q dx &= \int_{-L}^0 \varphi_{s,p} \psi_{s,q} dx + \int_0^{h_1} \varphi_{f,p} \psi_{f,q} dx + \\ &+ \int_{h_1}^{h_2} \varphi_{c,p} \psi_{f,q} dx + \int_{h_2}^L \varphi_{c,p} \psi_{c,q} dx, \end{aligned} \quad (17)$$

где зависимость от переменной x опущена для краткости записи. Отметим, что $\varphi_{\alpha,p}(x)$ и $\psi_{\alpha,q}(x)$, $\alpha = c, f, s$ принимают либо вид (15), либо вид (16) в зависимости от положительности/отрицательности выражений $\beta_p - n_\alpha$ и $\gamma_q - n_\alpha$. Поэтому для численного расчета любого интеграла вида (17) достаточно иметь только 3 символьных выражения для определенных интегралов:

$$\begin{aligned} I_{1,1}(A_1, A_2, B_1, B_2, a, b, \beta, \gamma) = \\ = \int_a^b (A_1 e^{-\beta x} + A_2 e^{\beta x}) (B_1 e^{-\gamma x} + B_2 e^{\gamma x}) dx, \end{aligned} \quad (18)$$

$$I_{1,2}(A_1, A_2, B_1, B_2, a, b, \beta, \gamma) = \int_a^b (A_1 e^{-\beta x} + A_2 e^{\beta x}) (B_1 \sin(\gamma x) + B_2 \cos(\gamma x)) dx, \quad (19)$$

$$\begin{aligned} I_{2,2}(A_1, A_2, B_1, B_2, a, b, \beta, \gamma) = \\ = \int_a^b (A_1 \sin(\beta x) + A_2 \cos(\beta x)) \times \\ \times (B_1 \sin(\gamma x) + B_2 \cos(\gamma x)) dx. \end{aligned} \quad (20)$$

Используем возможности системы компьютерной алгебры Maple, а именно функцию `int` для символьного интегрирования, а также функции `factor`, `expand` для упрощения символьных выражений [6]. Полученные в системе компьютерной алгебры символьные выражения $I_{1,1}$, $I_{1,2}$, $I_{2,2}$ используются для расчета интегралов по формуле (17).

Описанные символьно-численные методы реализованы в системе компьютерной алгебры Maple.

Algorithm 2: Algorithm `ipq` for numeric calculation of integral (18), (19) or (20)

Input: β, γ – numeric values for eigenvalues, n_1, n_2 – numeric values for refractive indices, a, b – numeric values for integration limits, $A_{1,2}, B_{1,2}$ – numeric values for amplitudes

Output: numeric value of sub-integral

```

if  $\beta > n_1$  &  $\gamma > n_2$  then
     $\text{res} \leftarrow$  subs numeric values of
     $A_{1,2}, B_{1,2}, a, b, \beta, \gamma$  in  $I_{1,1}$ 
if  $\beta < n_1$  &  $\gamma < n_2$  then
     $\text{res} \leftarrow$  subs numeric values of
     $A_{1,2}, B_{1,2}, a, b, \beta, \gamma$  in  $I_{2,2}$ 
if  $\beta < n_1$  &  $\gamma > n_2$  then
     $\text{res} \leftarrow$  subs numeric values of
     $A_{1,2}, B_{1,2}, a, b, \beta, \gamma$  in  $I_{1,2}$ 
if  $\beta > n_1$  &  $\gamma < n_2$  then
     $\text{res} \leftarrow$  subs numeric values of
     $B_{1,2}, A_{1,2}, a, b, \beta, \gamma$  in  $I_{1,2}$ 
return  $\text{res}$ 

```

3.4. Реализация символьно-численных алгоритмов в Maple

Реализация символьно-численного метода решения волноводной задачи состоит из 3-х символьно-численных алгоритмов, реализованных в системе компьютерной алгебры Maple. Метод

волнового сопряжения, описанный в разделе 3.2, реализован в системе компьютерной алгебры Maple в виде процедуры `wave_conj`, псевдокод которой приведен выше, см. алг. 1. Символьная часть метода состоит в формировании матрицы $M_{1,2,3}(\eta)$ для каждого из трех диапазонов $\eta \in (n_s, n_f)$, $\eta \in (n_c, n_s)$ и $\eta \in (0, n_c)$. Численно отыскиваем значения β_j , решая уравнение $\det M_{1,2,3}(\beta) = 0$ и значения $\tilde{A}_j$, решая систему (14) для левого волновода и, аналогично, γ_j и $\tilde{B}_j$ для правого волновода.

Далее численные значения β_j , $\tilde{A}_j$ и γ_j , $\tilde{B}_j$ в качестве аргументов передаются в процедуру `scprod` (см. алг. 3), которая вычисляет матрицу скалярных произведений (12), опираясь на заранее полученные символьные выражения для интегралов типа (18), (19) и (20) с помощью процедуры `ipq` (см. алг. 2).

После численного расчета матрицы (12) решаем систему (11) с помощью метода `LinearSolve` библиотеки `LinearAlgebra`.

Algorithm 3: Algorithm `scprod` for numeric calculation matrix defined in (12)

Input: $\beta_j, A_j, j = 1, \dots, N$ and $\gamma_j, B_j, j = 1, \dots, M$ – numeric values for eigenpairs, calculated by `wave_conj` algorithm, h_1, h_2, L – numeric values for thicknesses, n_c, n_s, n_f – numeric values for refractive indices

Output: Matrix F of scalar product defined in (12)

```

for p = 1, ..., N do
  for q = 1, ..., M do
     $I_{pq}^{(cc)} \leftarrow \text{ipq}(A_1, A_2, B_1, B_2, h_2, L, n_c, n_c, \beta_p, \gamma_q)$ 
     $I_{pq}^{(cf)} \leftarrow \text{ipq}(A_1, A_2, B_3, B_4, h_1, h_2, n_c, n_f, \beta_p, \gamma_q)$ 
     $I_{pq}^{(ff)} \leftarrow \text{ipq}(A_3, A_4, B_3, B_4, 0, h_1, n_f, n_f, \beta_p, \gamma_q)$ 
     $I_{pq}^{(ss)} \leftarrow \text{ipq}(A_5, A_6, B_5, B_6, -L, 0, n_s, n_s, \beta_p, \gamma_q)$ 
     $F_{pq} \leftarrow I_{pq}^{(cc)} + I_{pq}^{(cf)} + I_{pq}^{(ff)} + I_{pq}^{(ss)}$ 
  end for
end for
return F

```

4. РЕЗУЛЬТАТЫ

Используем разработанный символьно-численный алгоритм для приближенного решения задачи дифракции на стыке двух волноводов методом Галеркина. Исследуем для начала точность метода Галеркина в зависимости от параметра L , для этого оцениваем невязку δ , определяющую баланс энергии

$$\delta = \sqrt{\sum_{j=0}^N |R_j|^2 + \sum_{j=0}^M |T_j|^2} - 1. \quad (21)$$

В зависимости от величины L изменяется число базисных функций, а именно $N = 52$, $M = 52$ при $L = 5\lambda$, $N = 152$, $M = 151$ при $L = 15\lambda$ и $N = 248$, $M = 249$ при $L = 25\lambda$. Для дальнейших расчетов выбираем значение $L = 25\lambda$, при котором величина δ достигаем минимума. Приведем далее численные результаты решения задачи дифракции на стыке двух волноводов, где толщина $h_1 = 1.5\lambda$ фиксирована, а толщина второго волновода h_2 изменяется от 1.5λ до 3.7λ при $\lambda = 0.55 \mu\text{m}$, показатели преломления слоев определены следующим образом $n_c = 1.0$, $n_s = 1.47$, $n_f = 1.565$. Оцениваем численно изменение абсолютных значений амплитуд прошедших волноводных мод $|T_0|$ и $|T_1|$.

Величины абсолютных значений амплитуд прошедших волноводных мод $|T_0|$ и $|T_1|$ характеризуют перераспределение энергии между первыми двумя волноводными модами. Численно оценим также суммарную интенсивность E_1 , которая приходится на моды с $n_s < \gamma_q < n_f$ и суммарную интенсивность E_2 , приходящуюся на моды с $0 < \gamma_q < n_s$. Величины E_1 и E_2 определены следующим образом

$$E_1 = \sum_{n_s < \gamma_q < n_f} |T_q|^2, \quad E_2 = \sum_{0 < \gamma_q < n_s} |T_q|^2. \quad (22)$$

5. ОБСУЖДЕНИЕ

В работе построена символьно-численная реализация метода Галеркина для решения задачи волноводной дифракции на стыке двух планарных трехслойных волноводов. Благодаря тому, что собственные функции задачи (6) вычисляются в символьно-численном виде можно применить метод волнового сопряжения, который реализован в виде процедуры `wave_conj` в Maple. В системе Maple также реализована процедура `scprod` для численного расчета матрицы скалярных произведений метода Галеркина на основе вычисленных собственных функций. Процедура `scprod` использует рассчитанные и упрощенные формулы для типовых интегралов (18)–(20), возникающих в задаче дифракции. Другими словами процедура интегрирования осуществляется три раза для расчета интегралов (18)–(20), далее полученные символьные выражения упрощаются и используются в численных расчетах.

Важно отметить, что несмотря на использование символьных манипуляций, матрица скалярных произведений Φ из (12) вычисляется приближенно, из-за того, что собственные функции ϕ_p и

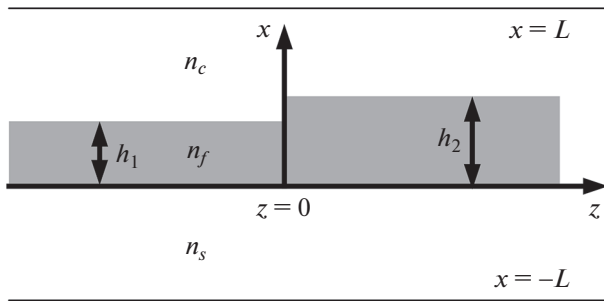


Рис. 1. Стык двух планарных открытых волноводов, помещенных в объемлющий закрытый волновод с границами $x = \pm L$.

ψ_q определяются приближенно. Каждый элемент диагональных матриц собственных значений $D_{1,2}$ системы (11) известен с точностью 10^{-15} — так как собственные значения являются корнями нелинейного уравнения, которые без труда отыскиваются методом деления отрезка пополам с точностью 10^{-15} . При такой точности вычисления собственных значений собственные функции отыскиваются с точностью $a \times 10^{-13}$, где $a < 10$ — см. например [8]. Для расчета матричных элементов Φ из (12) необходимо вычислить интеграл от произведения ортонормированных функций φ_p и ψ_q , ошибка при расчете такого интеграла будет порядка $2a \times 10^{-13}$.

Символьно-численная реализация метода Галеркина позволяет однократно вычислить однотипные интегралы, ускоряя тем самым последующие численные расчеты, что особенно актуально для векторных задач — см. [9]. В настоящей работе рассматривается скалярная задача, которую, однако, необходимо решать многократно, как для исследования сходимости метода Галеркина (см. рис. 2), так и для исследования самой задачи дифракции на различных стыках. Стоит отметить, что использование систем компьютерной алгебры характерно не только для методов Галеркина [10] и Ритца [11], но и для других приближенных методов решения волноводных задач [12].

Корректная постановка задачи дифракции основана на использовании подхода, предложенного А.Г. Свешниковым для задач волноводной дифракции [7], схожему с подходом О.И. Толстихина для задач квантовой механики [1]. Подход состоит в помещении рассматриваемого волноводного стыка открытых волноводов в объемлющий закрытый волновод (“ящик Дирихле”), границы которого $x = \pm L$ удалены от рассматриваемой структуры на “достаточно большое” расстояние. В работе величина этого расстояния L определена из численного эксперимента: исследовано поведение невязки δ , описывающей численное “нару-

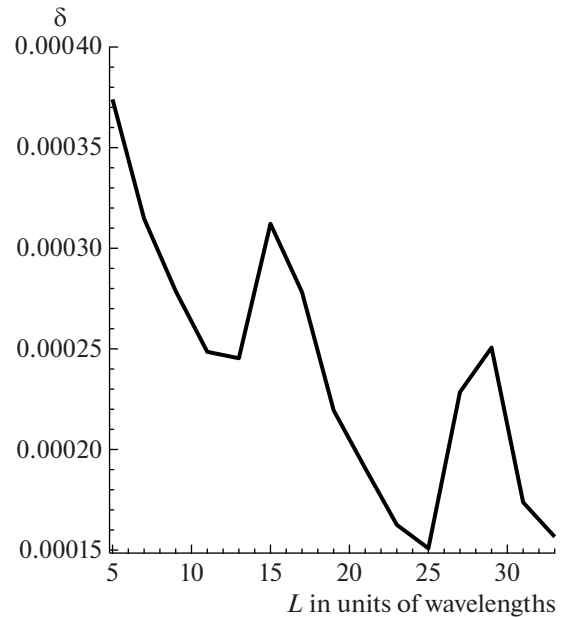


Рис. 2. График сходимости величины δ для стыка с параметрами $h_1 = 1.5w$, $h_2 = 1.8w$, $n_c = 1.0$, $n_s = 1.47$, $n_f = 1.565$, $A = 1$, $m = 0$, где длина волны $\lambda = 0.55 \mu\text{m}$.

шение” закона сохранения энергии (21). Полученные результаты (см. рис. 2) демонстрируют немонотонное поведение невязки, которое характерно для метода Галеркина в волноводных задачах — см. [9]. Получено значение $L = 25\lambda$,

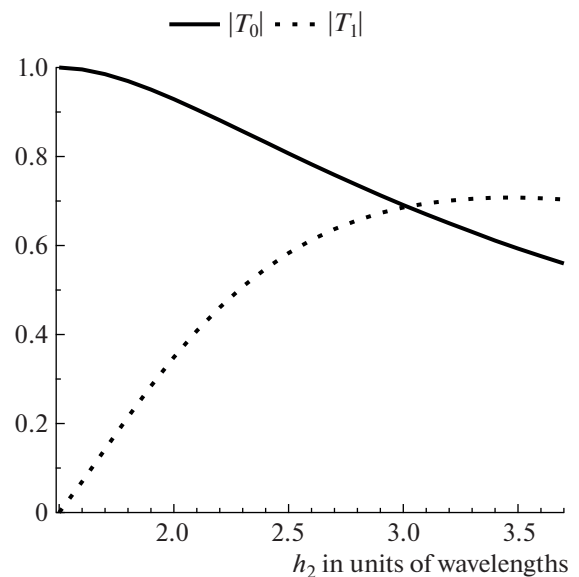


Рис. 3. График изменения величин $|T_0|$ и $|T_1|$ в зависимости от толщины h_2 для стыка с параметрами $h_1 = 1.5\lambda$, $n_c = 1.0$, $n_s = 1.47$, $n_f = 1.565$, $A = 1$, $m = 0$, где длина волны $\lambda = 0.55 \mu\text{m}$.

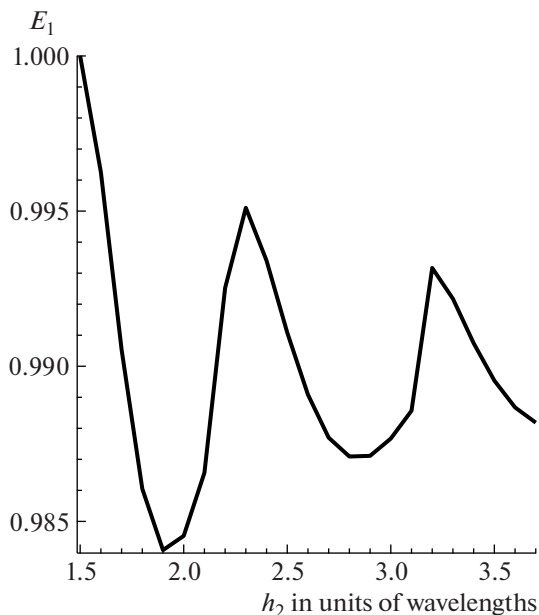


Рис. 4. График изменения величины E_1 в зависимости от толщины h_2 для стыка с параметрами $h_1 = 1.5\lambda$, $n_c = 1.0$, $n_s = 1.47$, $n_f = 1.565$, $A = 1$, $m = 0$, где длина волны $\lambda = 0.55 \mu\text{m}$.

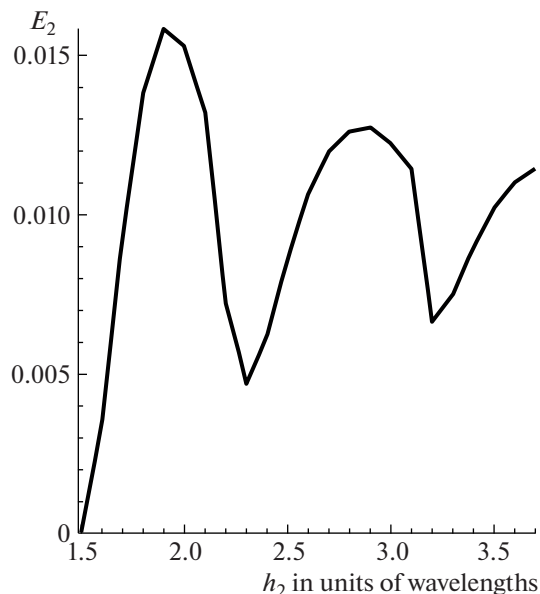


Рис. 5. График изменения величины E_2 в зависимости от толщины h_2 для стыка с параметрами $h_1 = 1.5\lambda$, $n_c = 1.0$, $n_s = 1.47$, $n_f = 1.565$, $A = 1$, $m = 0$, где длина волны $\lambda = 0.55 \mu\text{m}$.

обеспечивающее минимальное значение невязки δ , где λ — длина волны.

Использование “ящика Дирихле” позволяет аппроксимировать непрерывный спектр исходной задачи для открытых планарных волноводов [13–15] дискретным спектром благодаря введению искусственных границ $x = \pm L$. Этот подход позволяет сформулировать корректную задачу дифракции и применить для ее исследования метод Галеркина, а также приближенно оценить какое количество энергии приходится на моды непрерывного спектра исходной задачи (см. рис. 4, 5). На рис. 4 представлен график величины E_1 , которая описывает суммарную интенсивность, приходящуюся на моды, аппроксимирующие дискретный спектр исходной задачи в зависимости от толщины h_2 . Когда $h_2 = h_1$ стык отсутствует, поэтому $|T_0| = |A| = 1$ (см. рис. 3) — единственная волноводная мода, налетающая на стык из области $z < 0$, продолжает распространяться вдоль волновода с неизменной амплитудой. В данном случае перераспределение энергии отсутствует, о чем свидетельствует значение величин $E_1 = 1$ на рисунке 4. Величина E_2 на рисунке 5 описывает суммарную интенсивность, приходящуюся на моды, аппроксимирующие непрерывный спектр исходной задачи в зависимости от толщины h_2 . Когда $h_2 = h_1$ стык отсутствует, поэтому $E_2 = 0$.

Первые две моды, которым соответствуют амплитуды T_0 и T_1 , аппроксимируют направляемые моды исходной задачи — моды дискретного спектра исходной задачи. При увеличении толщины $h_2 > h_1$ происходит перераспределение энергии между модами, о чем свидетельствует рисунок 3: амплитуда T_0 уменьшается по абсолютному значению, а амплитуда T_1 — увеличивается. Перераспределение энергии происходит не только между модами дискретного спектра исходной задачи, о чем свидетельствуют графики на рис. 4 и 5: при увеличении толщины h_2 суммарная интенсивность мод, аппроксимирующих дискретный спектр исходной задачи, уменьшается от 1.0 до примерно 0.984 при увеличении h_2 от 1.5λ до 1.9λ (см. рис. 4), а суммарная интенсивность мод, аппроксимирующих непрерывный спектр исходной задачи, увеличивается от 0.0 до примерно 0.016 при увеличении h_2 от 1.5λ до 1.9λ (см. рис. 5). При дальнейшем увеличении толщины h_2 до значения 2.3λ суммарная интенсивность мод, аппроксимирующих дискретный спектр исходной задачи, увеличивается до примерно 0.995 (см. рис. 4), а суммарная интенсивность мод, аппроксимирующих непрерывный спектр исходной задачи, уменьшается до примерно 0.005 (см. рис. 5). Стоит также отметить, что суммарная интенсивность всех отраженных мод не более 0.001, поэтому графики этих значений не приводятся.

Резюмируя вышесказанное, перераспределение энергии между модами, аппроксимирующими дискретный спектр исходной задачи, существенно — при увеличении толщины h_2 абсолютное значение амплитуды T_0 уменьшается практически в 2 раза, а абсолютное значение амплитуды T_1 увеличивается от 0.0 до 0.7. Доля энергии, которая приходится на моды, аппроксимирующие непрерывный спектр исходной задачи, составляет не более 1.6% для рассматриваемой структуры (см. рис. 5).

6. ЗАКЛЮЧЕНИЕ

В работе построена символьно-численная реализация метода Галеркина, которая позволяет исследовать задачу волноводной дифракции на стыке двух планарных волноводов в “ящике Дирихле”. Использование символьных манипуляций позволяет ускорить расчет интегралов в методе Галеркина благодаря однократному символьному расчету типовых для задачи интегралов вместо многократного численного интегрирования, что позволяет проводить ресурсоемкие расчеты на персональном компьютере для задачи дифракции с количеством собственных функций, исчисляемым сотнями.

В работе численно определена высота “ящика Дирихле”, исследовано перераспределение энергии между направляемыми модами, аппроксимирующими дискретный спектр исходной задачи, а также оценена доля энергии, которая приходится на моды, аппроксимирующие непрерывный спектр исходной задачи.

7. ИСТОЧНИК ФИНАНСИРОВАНИЯ

Исследование выполнено за счет гранта Российского научного фонда (проект № 20-11-20257).

СПИСОК ЛИТЕРАТУРЫ

1. Tolstikhin O.I., Ostrovsky V.N., Nakamura H. Siegert Pseudo-States as a Universal Tool: Resonances, S Matrix, Green Function // Physical Review Letters. 1997. V. 79. № 11. P. 2026–2029.
2. Sveshnikov A.G. The basis for a method of calculating irregular waveguides // USSR Computational Mathematics and Mathematical Physics. 1963. V. 3. № 1. P. 219–232.
3. Eremin Y.A., Sveshnikov A.G. Study of scalar diffraction at a locally inhomogeneous body by a projection method // USSR Computational Mathematics and Mathematical Physics. 1976. V. 16. № 3. P. 255–260.
4. Delitsyn A.L. On the completeness of the system of eigenvectors of electromagnetic waveguides // Computational Mathematics and Mathematical Physics. 2011. V. 51. № 10. P. 1771–1776.
5. Sveshnikov A.G. A substantiation of a method for computing the propagation of electromagnetic oscillations in irregular waveguides // USSR Computational Mathematics and Mathematical Physics. 1963. V. 3. № 2. P. 413–429.
6. Mathematics-based software and services for education, engineering, and research <https://www.maple-soft.com/>
7. Свешников А.Г. Неполный метод Галеркина // ДАН СССР. 1977. Т. 236. № 5. С. 1076–1079.
8. Диваков Д.В., Тютюнник А.А. Символьное исследование спектральных характеристик направляемых мод плавно-нерегулярных волноводов // Программирование. 2022. № 2. С. 23–32.
9. Tiutiunnik A.A., Divakov D.V., Malykh M.D., Sevastianov L.A. Symbolic-Numeric Implementation of the Four Potential Method for Calculating Normal Modes: An Example of Square Electromagnetic Waveguide with Rectangular Insert // Lecture Notes in Computer Science. 2019. V. 11661. P. 412–429.
10. Виноцкий С.И., Гердт В.П., Гусев А.А., Касчиев М.С., Ростовцев В.А., Самойлов В.Н., Тюпикова Т.В., Чулуунбаатар О. Символьночисленный алгоритм вычисления матричных элементов параметрической задачи на собственные значения // Программирование. 2007. Т. 33. № 2. С. 63–76.
11. Зорин А.В., Севастьянов Л.А., Третьяков Н.П. Компьютерное моделирование водородоподобных атомов в квантовой механике с неотрицательной функцией распределения // Программирование. 2007. Т. 33. № 2. С. 50–62.
12. Диваков Д.В., Тютюнник А.А. Символьное исследование собственных векторов для построения общего решения системы ОДУ с символьной матрицей коэффициентов // Программирование. 2021. 1. С. 11–24.
13. Shevchenko V.V. Spectral decomposition in eigen- and associated functions of a nonselfadjoint problem of Sturm–Liouville type on the entire axis // Differ. Uravn. 1979. V. 15. № 11. P. 2004–2020.
14. Gevorkyan M.N., Kulyabov D.S., Lovetskiy K.P., Sevastianov A.L., Sevastyanov L.A. Waveguide modes of a planar optical waveguide // Mathematical Modelling and Geometry. 2015. V. 3. № 1. P. 43–63.
15. Sevastianov L.A., Egorov A.A., Sevastyanov A.L. Method of adiabatic modes in studying problems of smoothly irregular open waveguide structures // Physics of Atomic Nuclei. 2013. V. 76. № 2. P. 224–239.

УДК 004.421.6

РАЗРАБОТКА АЛГОРИТМИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОДЕЛИРОВАНИЯ УПРАВЛЯЕМЫХ ДИНАМИЧЕСКИХ СИСТЕМ С ПРИМЕНЕНИЕМ СИМВОЛЬНЫХ ВЫЧИСЛЕНИЙ И СТОХАСТИЧЕСКИХ МЕТОДОВ

© 2023 г. А. В. Демидова^{a,*}, О. В. Дружинина^{b,**}, О. Н. Масина^{c,***}, А. А. Петров^{c,****}

^aРоссийский университет дружбы народов
117198 Москва, ул. Миклухо-Маклая, д. 6, Россия

^bФедеральный исследовательский центр “Информатика и управление” РАН
119333 Москва, ул. Вавилова, д. 44, кор. 2, Россия

^cЕлецкий государственный университет им. И.А. Бунина
399770 Елец, Липецкая обл., ул. Коммунаров, д. 28, Россия

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

**E-mail: ovdruzh@mail.ru

***E-mail: olga121@inbox.ru

****E-mail: xeal91@yandex.ru

Поступила в редакцию 01.07.2022 г.

После доработки 29.07.2022 г.

Принята к публикации 30.10.2022 г.

Разработка нового программного обеспечения для решения задач синтеза и анализа управляемых моделей с учетом детерминированного и стохастического описания является актуальным научным направлением. В статье представлены результаты разработки программного комплекса моделирования динамических систем, поведение которых может быть описано одношаговыми процессами. В качестве примеров рассмотрены модели популяционной динамики. Программный комплекс с использованием на входе детерминированного описания модели позволяет получить соответствующую стохастическую модель в символьном виде, а также провести детальный анализ модели (расчет траекторий в детерминированном и стохастическом случаях, поиск управляющих функций, графическая визуализация результатов). Важным аспектом разработки программного комплекса является применение методов компьютерной алгебры в задачах анализа модели и синтеза управлений. В программном комплексе реализованы методы и алгоритмы, базирующиеся на детерминированных и стохастических методах Рунге–Кутты, методах теории устойчивости и теории управления, методе построения самосогласованных стохастических моделей, алгоритмах численной оптимизации и искусственного интеллекта. Предложенный программный комплекс разрабатывается на основе языков высокого уровня Python и Julia. В качестве базового инструментального программного обеспечения используются высокопроизводительные библиотеки для векторно-матричных расчетов, библиотеки символьных вычислений, библиотеки для численного решения обыкновенных дифференциальных уравнений, библиотеки алгоритмов глобальной оптимизации.

DOI: 10.31857/S0132347423020085, EDN: MFTAWU

1. ВВЕДЕНИЕ

Разработка инструментального обеспечения для моделирования динамических систем с применением символьных вычислений является актуальной проблемой [1–4]. В частности, теоретический и прикладной интерес представляет разработка средств получения формализованного представления динамических моделей популяционной динамики исходя из общего описания принципов взаимодействия популяций. Проблема формализации и изучения многомерных дина-

мических моделей популяционной динамики относится к многоплановым и многоаспектным проблемам, требующим как понимания сущности процессов взаимодействия фазовых переменных, так и привлечения современных методов теории детерминированных и стохастических дифференциальных уравнений, теории устойчивости, теории управления.

Развиваемый в настоящей работе подход к разработке инструментально-методического обеспечения базируется на реализации возможностей

построения и исследования нелинейных динамических систем, поведение которых может быть описано одношаговыми процессами. Как известно, класс одношаговых процессов охватывает многие процессы физики, химии, экологии, демографии, социологии. Для дифференциальных уравнений, описывающих одношаговые процессы, применяется ряд методов интегрирования, например, явные и неявные методы Рунге–Кутты [5]. Особый интерес для численного изучения представляют стохастические дифференциальные уравнения, моделирующие одношаговые процессы [6, 7]. Однако реализация стохастических методов Рунге–Кутты в рамках прикладных пакетов и математических библиотек развиты недостаточно. Для конкретных научных задач часто приходится разрабатывать собственную реализацию для численного решения стохастических дифференциальных уравнений [8–11]. В [10] дано описание комплекса программ, предназначенного для моделирования одношаговых стохастических процессов и реализующего модифицированные методы Рунге–Кутты. Указанный программный комплекс реализован на языке программирования Python с использованием библиотек NumPy и SymPy.

Отметим, что в [8–11] рассматривались такие задачи реализации алгоритмов моделирования одношаговых стохастических процессов, которые в качестве входных параметров используют вектор сносов и матрицу диффузии в уравнении Фоккера–Планка. В настоящей работе проблема ставится шире: требуется разработать единый инструмент для формализованного построения детерминированных моделей, поиска состояний равновесия с помощью символьных вычислений, а также усовершенствовать формализованное построение стохастических моделей. Кроме того, требуется учитывать управляющие воздействия в системе и построить управляющие функции с применением современных алгоритмов глобальной оптимизации [12]. Важность создания алгоритмов численной оптимизации в контексте данного научного направления связана с расширением требований к управлению экосистемами, в частности к управлению численностью популяций ценных промысловых видов.

В процессе стохастического моделирования возникает вопрос о механизме ввода стохастических членов в детерминированное уравнение [13, 14]. Часто стохастизация выполняется аддитивно с учетом добавления члена со стохастической переменной. Однако большей информативностью обладает подход, базирующийся на использовании стохастической части, согласованной с детерминированной. Этот подход реализуется в случае, если обе части (детерминированная и стохастическая) получаются из одного уравнения. Такую возможность предоставляет метод постро-

ения самосогласованных стохастических моделей [8, 11, 15]. В основе данного метода лежит комбинаторная методология [16, 17]. Данный метод предполагает, что эволюция во времени многомерных систем рождения-гибели может быть рассмотрена как результат индивидуальных взаимодействий между элементами этой системы. Согласно этому методу для описания системы используется основное кинетическое уравнение (Master equation). Это уравнение описывает эволюцию распределения вероятностей в цепи Маркова с непрерывным временем и представляет собой балансовое уравнение для вероятности каждого состояния системы. Кроме того, предполагается, что вероятность перехода из одного состояния в другое состояние (являющееся следствием рассматриваемого взаимодействия) пропорциональна числу возможных взаимодействий данного типа. Указанный метод позволяет выполнить переход к стохастической модели, важным этапом изучения которой является оценка влияния стохастики на качественные свойства системы.

В настоящей работе в качестве примеров реализации рассмотрены задачи моделирования систем популяционной динамики. Необходимость изучения экологических систем с различными взаимосвязями между подсистемами и изменение структуры этих взаимосвязей в процессе функционирования часто приводят к построению математических моделей, аналитическое исследование которых весьма затруднительно. Об актуальности проблем математической биологии свидетельствуют публикации последнего десятилетия (например, [18, 19]). Различные обобщения и модификации классических моделей экологических процессов [20] рассматривались в многочисленных работах исследователей (см., например, [21–23]). При аналитическом исследовании эффективно применяются методы Ляпунова [24–26]. В [27–31] предложено построение новых детерминированных моделей популяционной динамики, разработаны конкретизации метода стохастических самосогласованных моделей. Проведено численное исследование, дан сравнительный анализ траекторий. Поставлены и в ряде случаев решены задачи оптимального управления с разработкой и реализацией ряда алгоритмов численной оптимизации.

Модули разработанного в рамках настоящего исследования программного комплекса решают следующие задачи: расчет траекторий детерминированных динамических систем размерности $n \geq 2$; получение символьных выражений для перехода к стохастическим моделям; детальная формализация структуры стохастических систем; поиск управляющих функций в детерминированном случае. Программный комплекс включает в себя библиотеку алгоритмов численной оптими-

зации; библиотеку нейронных сетей и машинного обучения; графический модуль.

Предложенный программный комплекс разрабатывается на основе языков высокого уровня Python [32–36] и Julia [37]. В качестве базового инструментального программного обеспечения используются высокопроизводительные библиотеки для векторно-матричных расчетов (NumPy, LinearAlgebra.jl), библиотеки символьных вычислений (SymPy), библиотеки для численного решения обыкновенных дифференциальных уравнений (SciPy, DifferentialEquations.jl), библиотеки алгоритмов глобальной оптимизации (дифференциальная эволюция, генетический алгоритм, NES из состава BlackBoxOptim.jl). К задачам, решаемым в статье с помощью методов компьютерной алгебры, относятся нахождение состояний равновесия динамических моделей с помощью библиотеки SymPy, построение стохастических моделей в символьном виде (авторские программные модули), синтез управляющих функций с использованием полиномиальной регрессии и библиотек глобальной оптимизации.

Структура статьи следующая. В разделе 2 проведен сравнительный анализ используемого в работе программного обеспечения и средств символьных вычислений, приведена общая структура разработанного программного комплекса. В разделе 3 описано методическое обеспечение для построения детерминированной модели. В разделе 4 представлено алгоритмическое и программное обеспечение стохастического моделирования. В разделе 5 рассмотрены принципы построения модуля программного комплекса для получения управляющих функций. Раздел 6 посвящен вопросам получения и визуализации численных решений. В разделе 7 представлены результаты тестирования производительности в рамках отдельных задач, решаемых программным комплексом.

2. СРАВНИТЕЛЬНЫЙ АНАЛИЗ ИНСТРУМЕНТАЛЬНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ

В настоящем разделе мы проводим анализ инструментального обеспечения для моделирования динамических систем, поведение которых может быть описано одношаговыми процессами, при этом накладываются требования учета управляющих воздействий и перехода к стохастизации. Новые эффекты в динамических системах часто связаны с управляющими воздействиями и необходимостью перехода к стохастической структуре модели, обеспечивающей более реалистичное описание свойств. Используемое в работе инструментальное обеспечение должно предоставлять эффективные средства для решения комплекса задач в рамках изучаемой проблемы:



Рис. 1. Дерево инструментальных средств.

- построение детерминированных моделей динамических систем;
- построение стохастических моделей динамических систем;
- численное решение обыкновенных дифференциальных уравнений и стохастических дифференциальных уравнений;
- аналитический вывод в символьном виде (в частности, символьное решение систем нелинейных уравнений для поиска состояний равновесия);
- матрично-векторные расчеты;
- построение логических регуляторов (в том числе на основе искусственного интеллекта);
- глобальная параметрическая оптимизация.

Помимо возможностей инструментального обеспечения особое внимание уделяется доступности результатов. По этой причине в рамках исследования используется открытое программное обеспечение.

Для используемых в работе инструментальных программных средств можно выделить три категории, которые представлены на рис. 1.

Функциональные инструментальные средства непосредственно направлены на решение задач исследования. По внутренней структуре указанные средства представляют собой интегрированные комплексы высокоэффективных алгорит-

Таблица 1. Перечень инструментальных средств, использованных при разработке программного комплекса

Инструментальные средства		
Функциональные	Технические	Общего назначения
SymPy/SciPy (Python)	NumPy (Python)	Python
Symbolics.jl (Julia)	Numba (Python)	Julia
Maxima	ArrayFire (Python, Julia)	
BlackBoxOptim.jl (Julia)	Matplotlib (NumPy)	

мов. Технические инструментальные средства направлены на ускорение разработки, улучшение читаемости программного кода и эффективное использование современных программно-аппаратных платформ. Инструментальные средства общего назначения направлены на решение широкого спектра задач, например, на интеграцию разработанных модулей в единый программный комплекс.

Мы используем инструментальные средства, приведенные в табл. 1.

В качестве основного языка разработки используется мультипарадигменный язык Python в сочетании с библиотеками SymPy/SciPy, NumPy, Matplotlib. Пакет SymPy/SciPy представляет собой решение CAS-класса (CAS – англ. Computer Algebra Software, система компьютерной алгебры), которое позволяет решать ряд научных задач с использованием инструментария, близкого по возможностям к коммерческим решениям. Данный пакет содержит, в том числе, инструменты численного решения обыкновенных дифференциальных уравнений, а также библиотеки глобальной параметрической оптимизации (дифференциальная эволюция, имитация отжига и другие методы) [38].

Тем не менее, существенным недостатком указанного инструментального ПО является относительно невысокая производительность в ряде задач, среди которых можно отметить численное решение дифференциальных уравнений, а также аналитический вывод выражений. Указанный недостаток непосредственно влияет на время поиска состояний равновесия модели, а также на возможность применения машинного обучения с подкреплением для управления. Более подробно вопросы производительности рассмотрены в разделе 7.

Для того, чтобы компенсировать указанный недостаток, возможно использование дополнительных инструментальных средств, позволяю-

щих достичь более высокой производительности разработанного ПО в определенных задачах.

Для решения обыкновенных дифференциальных уравнений при реализации машинного обучения с подкреплением мы используем язык Julia с библиотеками bboptim.jl и Symbolics.jl. Библиотека Symbolics.jl, аналогично SymPy, представляет собой решение CAS-класса, которое может быть расширено библиотекой моделирования динамических систем ModelingToolkit.jl. Для нахождения состояний равновесия модели используются пакеты SymPy и Maxima, обеспечивающие высокую производительность в символьных вычислениях.

Отдельным вопросом является выявление условий эффективного применения технических инструментальных средств для увеличения производительности разработанного ПО при решении описанных выше задач моделирования. Совместно с языком Python используется оптимизирующий компилятор Numba, который позволяет многократно увеличить скорость выполнения некоторых участков кода благодаря использованию JIT-компиляции и автоматическому распараллеливанию. Кроме того, важной задачей является реализация гетерогенных вычислений (в частности, GPGPU) с высокой степенью переносимости в рамках разрабатываемого программного комплекса. Для интеграции указанной возможности в программный комплекс используется пакет ArrayFire, который позволяет производить “ленивые” вычисления с применением широкого спектра ускорителей вычислений. Преимуществами указанного пакета является высокий уровень абстракции по отношению к аппаратному обеспечению, что дает высокую переносимость разработанного программного обеспечения.

На основе выбранных инструментальных средств разработан авторский программный комплекс для проведения всестороннего анализа систем с взаимодействующими популяциями. Структура указанного программного комплекса представлена на рис. 2. Реализация проведена с использованием языка Python с применением компьютерной алгебры. При создании программного комплекса использован ряд результатов, полученных в [10].

Отметим, что разработанный программный комплекс комбинирует численные и символьные вычисления для увеличения степени универсальности по отношению к многообразию изучаемых систем. В последующих разделах основные принципы работы программного комплекса рассматриваются на примерах, конкретизациях и обобщениях. Листинги обсуждаемых в настоящей статье программ и примеры их использования доступны по адресу <https://github.com/demidav/one-step-to-sdu>.

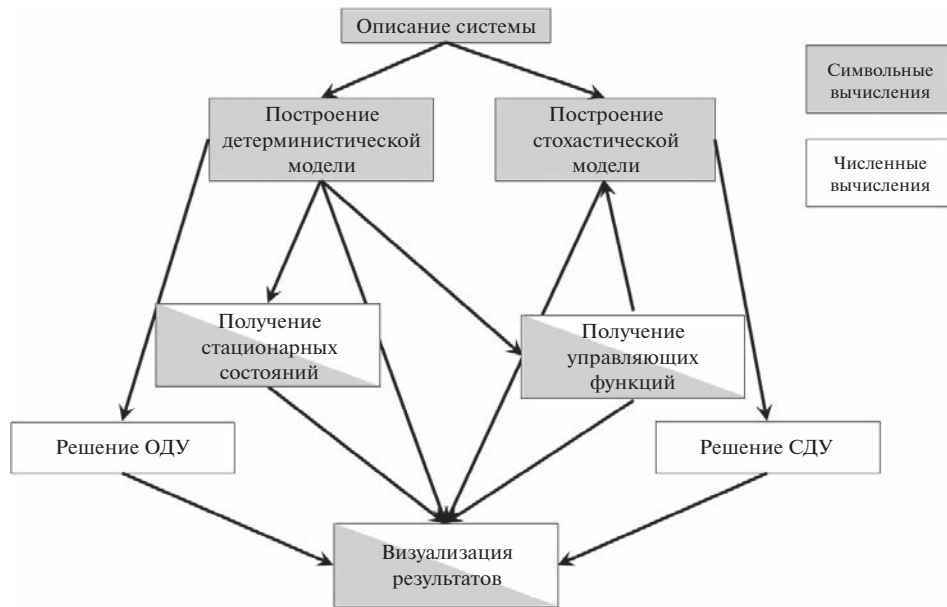


Рис. 2. Структура программного комплекса.

3. ПОСТРОЕНИЕ ДЕТЕРМИНИРОВАННОЙ МОДЕЛИ

Модели взаимодействия популяций в общем виде можно записать в форме векторного дифференциального уравнения:

$$\dot{x} = f_p(x), \quad x \in \mathbb{R}^n, \quad (3.1)$$

где $f_p(x)$ — параметрическая функция, определяющая взаимодействия в системе, x — фазовый вектор плотности популяций. Конкретный вид параметрической функции определяется существенными для модели факторами (наличием конкуренции, мутуализма, миграции, видом трофических цепей).

Кроме того, интерес представляет изучение модификаций модели (3.1) с учетом воздействия внешних факторов, в частности, управления. В таком случае уравнение (3.1) можно записать в виде:

$$\dot{x} = f_p(x) - u(x, t), \quad x \in \mathbb{R}^n, \quad (3.2)$$

где $u(x, t)$ — управление. Характерной особенностью данной модели является ограничение на управление $u(x, t) > 0$, поскольку непосредственное управление возможно, как правило, только путем изъятия особей из популяции. Пример управляемой модели рассматривается в разделе 0.

Рассмотрим модель “жертва—два хищника” без управления, рассмотренную в работе [22]. Уравнения указанной модели имеют вид

$$\begin{aligned} \dot{x} &= ax - b_1xy_1 - b_2xy_2, \\ \dot{y}_1 &= -c_1y_1 + d_1xy_1, \quad \dot{y}_2 = -c_2y_2 + d_2xy_2, \end{aligned} \quad (3.3)$$

где x — плотность популяции жертвы, y_1, y_2 — плотности популяции хищника, a — коэффициент естественного прироста популяции жертвы, c_1, c_2 — коэффициенты естественной смертности хищника, b_1, b_2 — коэффициенты убыли популяции жертв, d_1, d_2 — коэффициенты прироста популяции хищника. Модель (3.3) можно представить в виде графа взаимодействий, который имеет вид, представленный на рис. 3.

На рис. 3 кругами обозначены плотности популяций, ромбами обозначены источники естественного прироста/смертности, ребрами графа обозначены взаимодействия между популяциями.

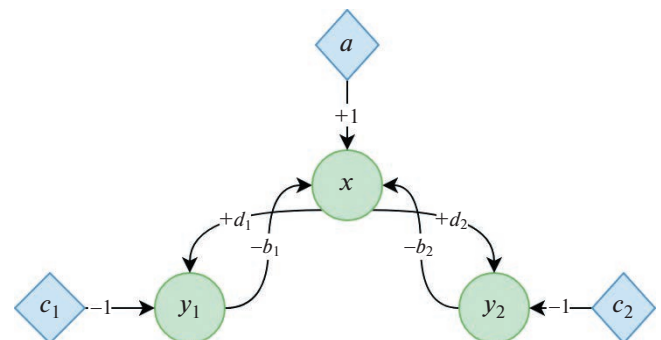


Рис. 3. Граф взаимодействий модели (3.3).

Важным вопросом при изучении моделей вида (3.3) является существование положительных состояний равновесия. Для предварительного анализа устойчивости модели (3.3) необходимо найти особые точки для уравнений системы, что приводит к необходимости решения уравнений вида

$$f_p(x) = 0, \quad x \in \mathbb{R}^n,$$

что для (3.3) эквивалентно

$$\begin{aligned} ax - b_1xy_1 - b_2xy_2 &= 0, \\ -c_1y_1 + d_1xy_1 &= 0, \quad -c_2y_2 + d_2xy_2 = 0. \end{aligned}$$

Для символьного решения уравнений используется функция `solve` из состава библиотеки SymPy. Особые точки для модели (3.3) имеют вид

$$\left[(0, 0, 0), \left(\frac{c_1}{d_1}, \frac{a}{b_1}, 0 \right), \left(\frac{c_2}{d_2}, 0, \frac{a}{b_2} \right) \right].$$

Следует отметить, что отсутствие положительных нетривиальных решений для уравнений (3.3) не является достаточным условием невозможности перманентного сосуществования популяций.

Упрощением модели (3.3) является модель, уравнения которой имеют вид:

$$\begin{aligned} \dot{x} &= ax - b_1xy_1 - b_2xy_2, \\ \dot{y}_1 &= -c_1y_1 + b_1xy_1, \\ \dot{y}_2 &= -c_2y_2 + b_2xy_2, \end{aligned} \quad (3.4)$$

где b_1, b_2 — коэффициенты трофических взаимодействий. Модель (3.4) представляет собой “уравновешенный” вариант модели (3.3), где скорость поглощения популяции жертвы соответствует скорости прироста популяции хищников. Граф указанной модели представлен на рис. 4.

Обозначения аналогичны рис. 3. Состояния равновесия модели (3.4) имеют вид

$$\left[(0, 0, 0), \left(\frac{c_1}{b_1}, \frac{a}{b_1}, 0 \right), \left(\frac{c_2}{b_2}, 0, \frac{a}{b_2} \right) \right].$$

По аналогии с моделью (3.3), модель (3.4) не имеет положительных нетривиальных решений. Следует отметить, что модели (3.3) и (3.4) оперируют рядом допущений (в частности, отсутствует внутривидовая конкуренция и конкуренция между жертвами). Однако, качественный и численный анализ указанных моделей, а также построение управлений является необходимым для апробации разработанного программного комплекса. При исследовании моделей вида (3.3) и (3.4), а также их многомерных модификаций и обобще-

ний возникают задачи, связанные с символьными вычислениями. Разработанный программный комплекс позволяет выполнять исследование устойчивости многофакторных моделей.

4. АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СТОХАСТИЧЕСКОЙ МОДЕЛИ

В основе рассматриваемого подхода к моделированию лежит метод построения самосогласованных стохастических моделей [8–11]. При использовании данного метода динамика изменения численности популяций рассматривается как случайный одношаговый процесс. На первом этапе необходимо сформулировать в терминах одношаговых процессов все взаимодействия, проходящие в системе между ее элементами. Далее требуется составить схему взаимодействия, под которой понимается символическая запись всех возможных взаимодействий между элементами системы с помощью оператора состояния системы и оператора изменения состояния системы. На основе схемы взаимодействий с помощью преобразований можно получить коэффициенты сноса и диффузии для уравнения Фоккера–Планка, что позволяет записать это уравнение и эквивалентное ему стохастическое дифференциальное уравнение в форме Ланжевена. Таким образом, можно одновременно получить как детерминированное описание системы, которому соответствует вектор сносов уравнения Фоккера–Планка, так и стохастическое описание, которое обеспечивается стохастическим дифференциальным уравнением в форме Ланжевена. Далее приведен алгоритм применения метода построения самосогласованных стохастических моделей.

Входные данные: Описание системы

Выходные данные: Детерминированная и стохастическая модели (системы ОДУ и СДУ)

begin

1. Добавление взаимодействий в модель
2. Построение схемы взаимодействия модели
3. Получение операторов состояния системы из схемы взаимодействия
4. Получение оператора изменения состояния системы
5. Получение интенсивностей переходов
6. Получение коэффициентов уравнения Фоккера–Планка

end

Алгоритм 1: Алгоритм применения метода построения самосогласованных стохастических моделей.

Опишем основные этапы этого алгоритма более подробно.

4.1. Описание системы и построение схемы взаимодействия

В [10] были формализованы шаги (3–6), в данной работе сделана формализация 1 и 2 шагов, а именно, формализация определения уравнений и построение схемы взаимодействия. Таким образом, в качестве входных данных используется не схема взаимодействия, а описание взаимодействий, имеющихся в системе, что упрощает построение.

Для этого был реализован класс PopModel с использованием библиотеки sympy и символьных вычислений. Исходными данными является описание происходящих в системе взаимодействий. Вызов конструктора PopModel(*n*) создает такой объект класса, который является формальной моделью системы, содержащей описание всех взаимодействий, происходящих в системе. Здесь *n* – это размерность системы. Для класса PopModel реализованы следующие методы:

- interaction(self, type_int);
- coef(self);
- adder(self, type_int, id_pop_1, id_pop_2, coef=0);
- matr_M(self);
- matr_N(self);
- display_infos(self, model, X, coef).

Метод adder(self, type_int, id_pop_1, id_pop_2, coef=0) является основным методом класса для добавления уравнений в схему взаимодействия. type_int – тип взаимодействия; id_pop_1 – идентификатор 1-й взаимодействующей популяции, id_pop_2 – идентификатор 2-й взаимодействующей популяции (соответствует индексу элемента в векторе состояния системы), coef – коэффициенты взаимодействия (по умолчанию принимает значение k_i).

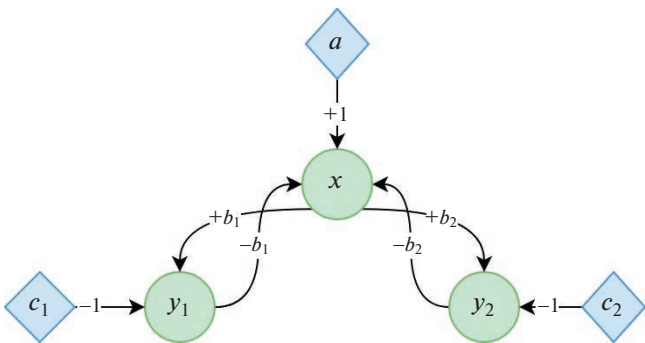


Рис. 4. Граф модели (3.4).

Метод interaction(self, type_int) – вспомогательный метод, содержащий описание различных взаимодействий. В моделировании популяционных систем принято выделять такие виды взаимодействий, как естественное размножение, естественная гибель, конкуренция, симбиоз, отношения хищник-жертва и др [22]. Типы взаимодействий, реализованных в классе PopModel представлены в табл. 2.

Метод coef(self) – вспомогательный метод, используется для определения коэффициентов взаимодействия, если они не заданы пользователем.

Методы matr_M(self) и matr_N(self) служат для получения матриц *M* и *N* в символьном виде, описывающих состояние системы до взаимодействия и после него.

Метод display_infos(self, model, X) предназначен для символьного отображения схемы взаимодействия.

Пример реализации. Для верификации полученных результатов рассмотрим трехмерную популяционную модель “два хищника–жертва” вида (3.4). В данной модели учитывается естественное размножение жертвы, естественная смертность хищников, а также убыль популяции жертвы и

Таблица 2. Описание параметров метода interaction() класса PopModel

type_int	id_pop_1	id_pop_2	Тип взаимодействия
1	<i>i</i>	0	Естественное размножение
2	<i>i</i>	0	Естественная гибель
3	<i>i</i>	<i>j</i>	Симбиоз
4	<i>i</i>	<i>j</i>	Хищник–жертва
5	<i>i</i>	0	Внутривидовая конкуренция
6	<i>i</i>	<i>j</i>	Межвидовая конкуренция
7	<i>i</i>	<i>j</i>	Миграция

```
In [11]:
model.display_infos(model,X)

x = [a] ⇒ 2x
y1 = [c1] ⇒ 0
y2 = [c2] ⇒ 0
x + y1 = [b1] ⇒ 2y1
x + y2 = [b2] ⇒ 2y2
```

Рис. 5. Результат вывода метода `display_infos`.

прирост популяции хищников происходит за счет взаимодействий типа “хищник–жертва”.

Импортируем необходимые библиотеки:

```
import sympy as sp
import numpy as np
import IS
```

Зададим вектор состояний системы $x = (x, y_1, y_2)$ в символьном виде:

```
X = sp.Matrix(['x', 'y_1', 'y_2'])
```

Создадим объект класса размерности 3:

```
model = IS.PopModel(3)
```

Добавим описание взаимодействий:

```
model.adder(type_int=1, 1, 0, "a")
model.adder(type_int = 2, 2, 0, "c_1")
model.adder(type_int = 2, 3, 0, "c_2")
model.adder(type_int = 4, 1, 2, "b_1")
model.adder(type_int = 4, 1, 3, "b_2")
```

В данном описании первая строка описывает размножение жертв (`type_int=1, id_pop_1=1`). Строки 2 и 3 соответствуют естественной убыли популяций хищников (`type_int=2, id_pop_1=2` и `id_pop_1=3` соответственно). Строки 4 и 5 описывают отношения типа хищник–жертва (`type_int=4`) жертвы с первым хищником (`id_pop_1=1, id_pop_2=2`) и со вторым хищником (`id_pop_1=1, id_pop_2=3`).

Далее с помощью метода `display_infos` можно вывести схему взаимодействия. При использовании интерактивной оболочки Jupyter вывод будет иметь вид, представленный на рис. 5.

После получения схемы взаимодействия можно перейти к этапу построения стохастических дифференциальных уравнений.

4.2. Построение коэффициентов уравнения Фоккера–Планка

Подробное описание модуля (`IStoDE.py`) получения коэффициентов уравнения Фоккера–Планка из схемы взаимодействия дано в [10]. Здесь дадим краткое описание. Алгоритм получения коэффициентов уравнения Фоккера–Планка из схемы взаимодействия (шаги 3–6 алгоритма 1) реализован как последовательность операций над векторными данными.

В качестве входных данных принимаются матрицы M и N (атрибуты объекта класса `PopModel`) состояний системы до взаимодействия и после, векторы `K_plus` и `K_minus` – коэффициенты взаимодействия и вектор X – вектор состояния системы. В качестве результата получаем символьную запись коэффициентов уравнения Фоккера–Планка. Использование библиотеки `SymPy` позволяет получить также и код этих коэффициентов в TeX, что упрощает их чтение.

Основными функциями данного модуля являются `drift_vector` для получения вектора сносов A и `diffusion_matrix` для получения матрицы диффузии B .

Пример реализации. На рис. 6 представлен результат работы функций получения коэффициентов уравнения Фоккера–Планка для модели (3.4).

Согласно рис. 6, вектор сносов полностью соответствует правой части системы уравнений для модели (3.4), что позволяет использовать этот вектор для исследования детерминированного поведения системы.

```
In [22]: k_plus=sp.Matrix(model.coef)

In [23]: IStoDE.drift_vector(X, k_plus, model.matr_N(), model.matr_M())

Out[23]: 
$$\begin{bmatrix} ax - b_1xy_1 - b_2xy_2 \\ b_1xy_1 - c_1y_1 \\ b_2xy_2 - c_2y_2 \end{bmatrix}$$


In [24]: IStoDE.diffusion_matrix(X, k_plus, model.matr_N(), model.matr_M())

Out[24]: 
$$\begin{bmatrix} ax + b_1xy_1 + b_2xy_2 & -b_1xy_1 & -b_2xy_2 \\ -b_1xy_1 & b_1xy_1 + c_1y_1 & 0 \\ -b_2xy_2 & 0 & b_2xy_2 + c_2y_2 \end{bmatrix}$$

```

Рис. 6. Результат вывода функций `drift_vector` и `diffusion_matrix`.

В результате проведенных символьных вычислений мы получаем стохастическую и детерминированную модели популяционной динамики. Далее могут быть применены остальные модули программного комплекса для исследования системы, такие как добавление и исследование управления, символьное и численное получение стационарных состояний системы, получение численных решений.

5. БАЗОВЫЙ АЛГОРИТМ МОДУЛЯ ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ ПОЛУЧЕНИЯ УПРАВЛЯЮЩИХ ФУНКЦИЙ

С учетом управляющих воздействий модель (3.4) можно записать в виде

$$\begin{aligned}\dot{x} &= ax - b_1xy_1 - b_2xy_2 - u_1(t)x, \\ \dot{y}_1 &= c_1y_1 + d_1xy_1 - u_2(t)y_1, \\ \dot{y}_2 &= -c_2y_2 + d_2xy_2 - u_3(t)y_2,\end{aligned}\quad (5.1)$$

где u_1, u_2, u_3 — функции управления. Рассматривается задача оптимального управления, состоящая в построении такого управления, которое позволяет с учетом выбранного критерия качества реализовать траекторию L при условиях:

$$\begin{aligned}x(0) &= x_0, & y_1(0) &= y_{10}, & y_2(0) &= y_{20}, \\ x(t_1) &= x_1, & y_1(t_1) &= y_{11}, & y_2(t_2) &= y_{21}.\end{aligned}$$

Будем рассматривать функции управления в виде нормированных полиномов. Примем

$$U = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}.$$

С учетом полиномиального управления имеет место равенство

$$U = \|PT\|, \quad P = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix},$$

где P — матрица коэффициентов, $T = (1, t, t^2)^*$.

Следует отметить, что размерность матрицы коэффициентов определяет порядок управляющих полиномов, а нормирование матричного произведения осуществляется построчно. С учетом того, что начальные условия для системы (5.1) известны, критерий оптимальности можно записать в виде:

$$\|X(t_1) - X_1\| \rightarrow \min,$$

где $X = (x(t), y_1(t), y_2(t))$, $X_1 = (x_1, y_{11}, y_{21})$.

Для нахождения матрицы коэффициентов разработан алгоритм на основе машинного обучения с подкреплением (см. алгоритм 2).

Входные данные: $X_1, X(0)$

Выходные данные: P

Function Loss($P, X(0)$):

prob₁ = ODEProblem(ecosystem, X_0, t_1)

sol₁ = solve(prob₁, *args)

return norm(sol₁.y[end] - X_1)

Function Main:

P = optimize(Loss, array(9), *args)

serialize(P)

return 0

return

Алгоритм 2: Алгоритм поиска коэффициентов управления.

Алгоритм 2 реализован на языке Julia с применением дифференциальной эволюции из состава пакета BlackBoxOptim.jl. Значения коэффициентов для $X_1 = (10, 10, 15)$ имеют вид

$$P = \begin{pmatrix} -5.13979 & 2.68192 & -0.233943 \\ -1.67237 & 4.09481 & -0.367989 \\ 2.71886 & 0.457653 & -0.015932 \end{pmatrix}.$$

Графическое представление плотностей популяций хищника и жертвы для управляемого случая при рассмотренных условиях приведено в разделе 6.

Разработанный модуль программного комплекса позволяет строить функции управления для многомерных систем популяционной динамики с учетом различных критериев качества. Взаимодействие указанного модуля с программным комплексом осуществляется посредством сериализации полученных данных. Отметим, что возможно расширение модуля на основе предложенного в [28] способа построения нейрорегулирования.

6. ПОЛУЧЕНИЕ И ВИЗУАЛИЗАЦИЯ ЧИСЛЕННЫХ РЕШЕНИЙ

Как известно, для численного решения систем обыкновенных дифференциальных уравнений и соответствующих им стохастических дифференциальных уравнений часто используются методы Рунге–Кутты и их модификации. Для решения обыкновенных дифференциальных уравнений в данной работе используется программная реали-

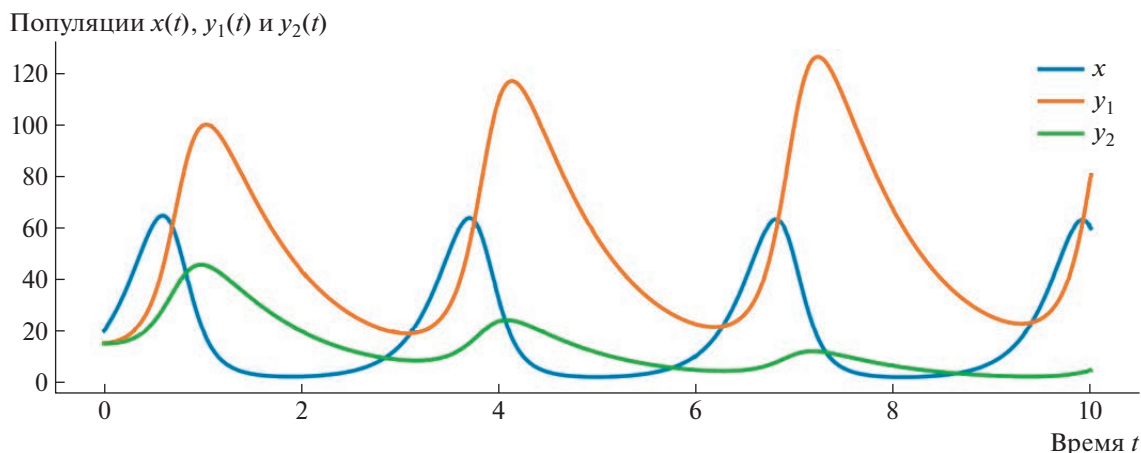


Рис. 7. Пример визуализации численного решения детерминированной модели при параметрах $(a, b_1, b_2, c_1, c_2) = [4.5, 1.2, 1.1, 0.07, 0.05]$. Начальные значения $(x(0), y_1(0), y_2(0)) = [20, 15, 15]$.

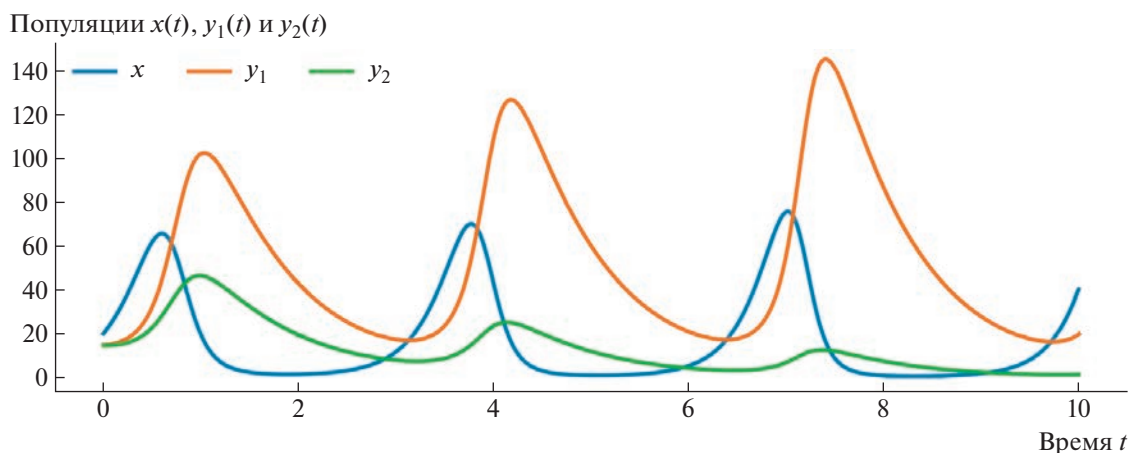


Рис. 8. Пример визуализации численного решения стохастической модели при параметрах $(a, b_1, b_2, c_1, c_2) = [4.5, 1.2, 1.1, 0.07, 0.05]$. Начальные значения $(x(0), y_1(0), y_2(0)) = [20, 15, 15]$.

зация стандартных методов Рунге–Кутты четвертого порядка. Для стохастических дифференциальных уравнений используется специально разработанная библиотека, подробное описание которой приведено в [8, 9].

В настоящей работе для получения численных решений применительно к блоку `IStODE.py` реализована функция преобразования символьных данных в численные:

```
def func_for_rk(x, p, f):
    F = sp.lambdify(args>(*x, *p),
                     expr=f,
                     modules='numpy')
    def func_f(t, x, p):
        return F(*x, *p).flatten()
    return func_f
```

```
def func_for_sdu(x, p, g):
    G = sp.lambdify(args>(*x, *p),
                    expr=g,
                    modules='numpy')
    def func_g(x, p):
        return G(*x, *p).flatten()
    return func_g
```

В качестве иллюстрации приведена визуализация численного решения для детерминированного случая (рис. 7) и для стохастического случая (рис. 8). На рис. 9 приведена визуализация численного решения для управляемой детерминированной модели.

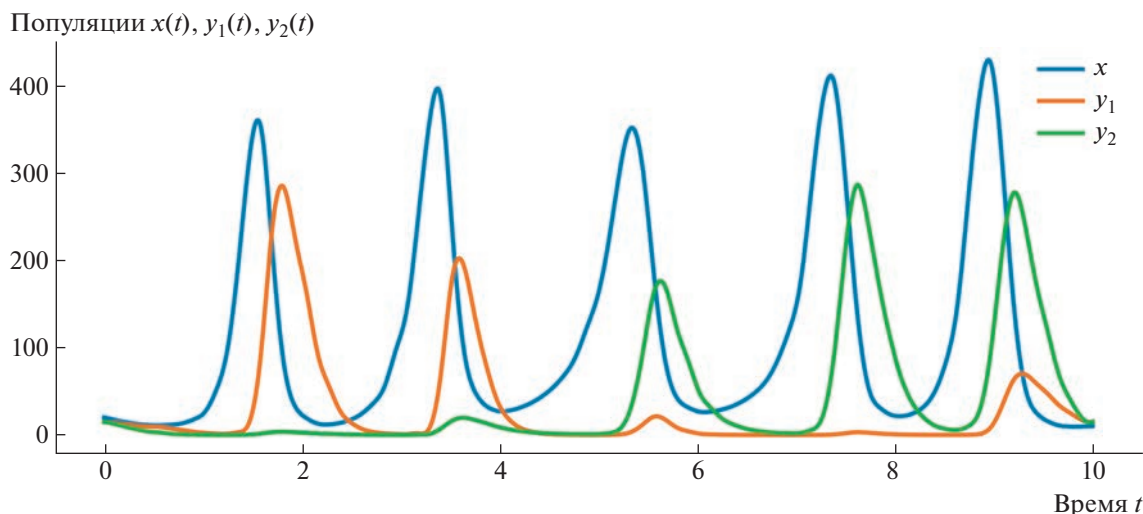


Рис. 9. Пример визуализации численного решения детерминированной модели с управлением при параметрах $(a, b_1, b_2, c_1, c_2) = [4.5, 1.2, 1.1, 0.07, 0.05]$. Начальные значения $(x(0), y_1(0), y_2(0)) = [20, 15, 15]$.

7. ТЕСТИРОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ В РАМКАХ ОТДЕЛЬНЫХ ЗАДАЧ, РЕШАЕМЫХ ПРОГРАММНЫМ КОМПЛЕКСОМ

Мы изучаем вопросы производительности инструментальных средств на примере модифицированной для управляемого случая известной модели Лотки–Вольтерры вида:

$$\begin{aligned} \dot{x}_1 &= (\alpha - \beta x_2) x_1 - v, \\ \dot{x}_2 &= (-\gamma + \delta x_1) x_2, \end{aligned} \quad (7.1)$$

где x_1, x_2 — плотности популяций жертвы и хищника соответственно, $\alpha, \beta, \gamma, \delta$ — коэффициенты взаимодействия, v — переключаемое управление, реализующее скользящий режим с законом управления:

$$\begin{aligned} x_1 < 3 : v &= 0, \\ x_1 \geq 3 : v &= 0.25. \end{aligned}$$

Экологический смысл указанного управления заключается в регулировании плотности популяции жертвы. Мы производим тестирование скорости работы кода при численном интегрировании методом Рунге–Кутты четвертого порядка для языков Python (Scipy.integrate) и Julia (DifferentialEquations.jl). Для указанных решателей используются настройки по умолчанию. Отметим, что для Julia не учитывается время компиляции программы. Конфигурация используемой ЭВМ: ЦП AMD Ryzen 5 5600X, 16 GB RAM.

Фрагмент тестовой программы на языке Python 3 приведен в листинге 1.

```
class lotki_volterra :
    def __init__ (self , alpha ,
```

```
        beta , gamma , delta ) :
    self . alpha = alpha
    self . beta = beta
    self . gamma = gamma
    self . delta = delta
    self .v = 0
```

```
def __call__ (self , t, y):
    a,b,g,d,v = ( self .alpha ,
                  self .beta ,
                  self .gamma ,
                  self .delta ,
                  self .v)
    x1, x2 = y
```

```
    dx1 = (a - b*x2)*x1 - v
    dx2 = (-g + d*x1)*x2
```

```
    if x1 < 3: self .v = 0
    else : self .v = 0.25
```

```
    return dx1 , dx2
```

```
model1 = lotki_volterra (1.5 , 1, 3, 1)
solution = solve_ivp ( model1 , (0 ,30) ,
                      (2 ,2) , t_eval =(0 , 30, 240) )
```

Листинг 1: Тестовая программа для расчета траекторий (Python 3)

Результатом работы программы являются графики плотностей популяций, представленные на рис. 10.

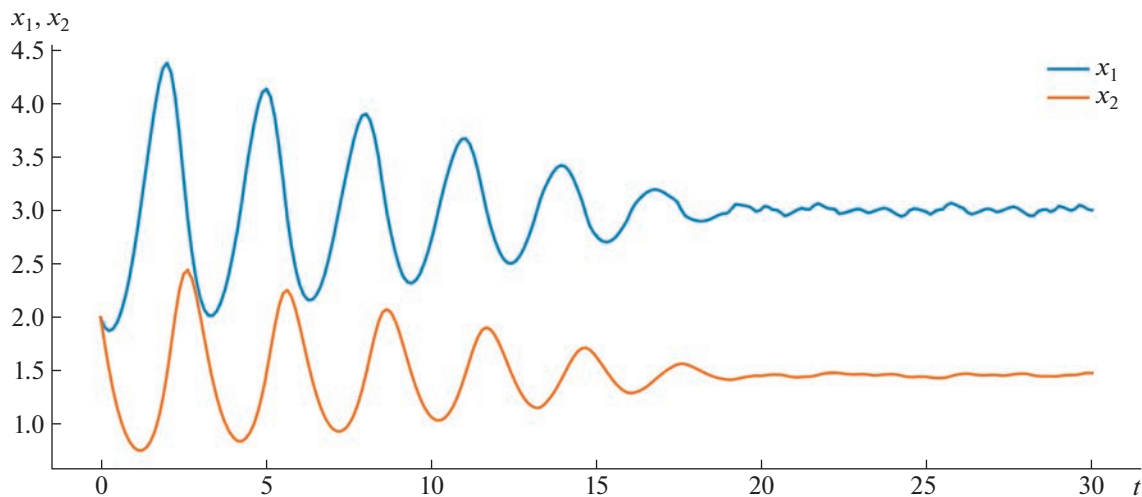


Рис. 10. Плотности популяций хищника и жертвы для модели (7.1) при выбранных начальных условиях.

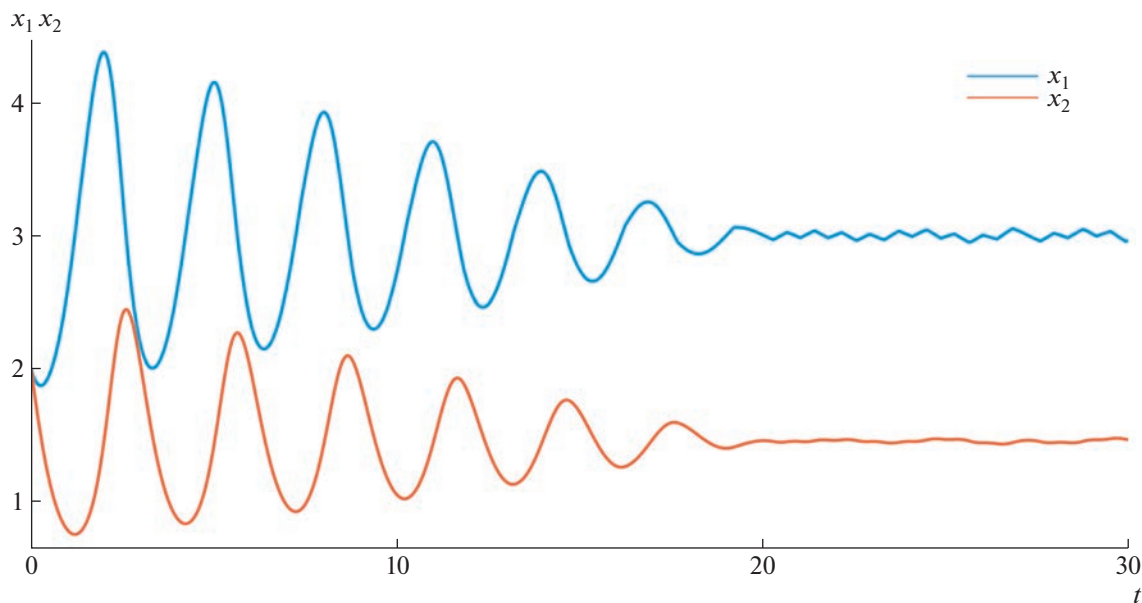


Рис. 11. Плотности популяций хищника и жертвы для модели (7.1) при выбранных начальных условиях (Julia).

Для замера производительности использована директива `%timeit` из пакета JupyterLab. Скорость выполнения функции `solve_ivp` составляет $8.43 \text{ ms} \pm 120 \text{ } \mu\text{s}$ per loop (mean $\pm$ std. dev. of 7 runs, 100 loops each). Рассмотрим аналогичную реализацию модели (7.1) на языке Julia, которая приведена на листинге 2.

```
function lotki_volterra (du , u, p, t)
    a,b,g,d,c = p
    du [1] = (a - b*u [2]) *u [1] - v
    du [2] = (-g + d*u [1]) *u [2]
end
switch_condition1 (u, t, integrator) =
```

```
    integrator .u [1] < 3
    affect1 ! ( integrator ) = integrator .p [5]
    = 0
    switch_condition2 (u, t, integrator) =
        integrator .u [1] >= 3
    affect2 ! ( integrator ) = integrator .p [5]
    = 0.25
    cb1 = DiscreteCallback (
        switch_condition1 , affect1 ! )
    cb2 = DiscreteCallback (
        switch_condition2 , affect2 ! )
    cbs = CallbackSet (cb1 , cb2 )
```

```

u0 = [2.0; 2.0]
p = [1.5 , 1., 3., 1., 0]
prob = ODEProblem (lotki_volterra , u0
, (0.0 , 30.0) , p)
sol = solve (prob , RK4 () , callback =
cbs)

```

Листинг 2: Тестовая программа для расчета траекторий (Julia)

Результат работы программы представляет собой график, аналогичный рис. 10, который приведен на рис. 11.

Для замера производительности использован пакет BenchmarkTools.jl (макрос @btime). Скорость работы функции 42.091 μ s (1021 allocations: 98.25 KiB). Таким образом, программа на языке Julia показывает в 195.53 меньшее время выполнения задачи построения траекторий управляемой системы (7.1). Несмотря на то, что программы на языке Julia требуют продолжительной предварительной компиляции, более высокая скорость работы имеет существенное значение при решении задач, требующих многократного решения обыкновенного дифференциального уравнения с разными параметрами. К таким задачам можно отнести, в частности, машинное обучение с подкреплением или анализ устойчивости модели. Далее рассмотрим производительность программного кода для поиска состояний равновесия экологических моделей. Как известно, задача нахождения особых точек уравнений вида

$$\dot{x} = f(x), \quad x \in \mathbb{R}^n,$$

требует решения уравнения

$$f(x) = 0.$$

Уравнения моделей взаимодействия популяций, как правило, имеют нелинейный характер правых частей. Мы изучаем пример, когда правые части модели задаются полиномами второго порядка:

$$\begin{aligned} \dot{y}_1 &= k_1 + k_2 y_1 + k_3 y_2^2, \\ \dot{y}_2 &= k_1 + k_2 y_2 + k_3 y_1^2. \end{aligned} \quad (7.2)$$

Программный код на языке Python 3 с применением библиотеки SymPy приведен в листинге 3.

```

from sympy import *
init_printing ( use_latex='mathjax ')
k1 ,k2 ,k3 ,k4 ,y1 ,y2 = symbols ('k1
k2 k3
k4 y1 y2 ')
eq1 = k1 + k2*y1 + k3*y2 **2
eq2 = k1 + k2*y2 + k3*y1 **2
solution = solve (( eq1 , eq2 ), (y1 , y2))
for sol in solution : display ( sol )

```

Листинг 3: Тестовая программа для поиска состояний равновесия (Python 3)

Программа, представленная в листинге 3, предназначена для поиска состояний равновесия для модели в форме дифференциальных уравнений. Для замера производительности использована директива %timeit из пакета JupyterLab. Скорость выполнения функции solve составляет 32.6 ms $\pm$ 634 μ s per loop (mean $\pm$ std. dev. of 7 runs, 10 loops each).

Далее рассмотрим решение приведенных уравнений в Maxima. Для расчета состояний равновесия используются следующие команды:

```

s:[ k1 + k2*y1 + k3*y2 ^2, k1 + k2*y2 +
k3*y1 ^2]
solve (s, [y1 , y2 ])

```

Время работы символьного решателя оценено с помощью встроенной функции time и составляет 0.002234 сек. (2.234 ms). Таким образом, время решения системы уравнений (7.2) для символьного решателя системы компьютерной алгебры Maxima составляет менее 7% от времени решения для пакета SymPy.

Полученные при тестировании производительности результаты свидетельствуют о том, что для решаемых научных задач целесообразно создавать интегрированные программные средства, базирующиеся на инструментах различных языков программирования. Одним из перспективных направлений развития работы является интеграция CAS Maxima со средствами символьного решения уравнений в рамках разрабатываемого программного комплекса.

8. ЗАКЛЮЧЕНИЕ

Таким образом, разработанный программный комплекс продемонстрировал возможности решения таких задач, как получение формализованного описания детерминированных моделей, получение символьных выражений для перехода к стохастическим моделям, детальная формализация структуры стохастических систем, расчет траекторий многомерных динамических систем, поиск управляющих функций. В программном комплексе находят реализацию методы и алгоритмы, базирующиеся на детерминированных и стохастических методах Рунге–Кутты, методах теории устойчивости и теории управления, методе построения самосогласованных стохастических моделей, алгоритмах численной оптимизации и искусственного интеллекта.

Методология, предложенная в работе, позволяет осуществлять построение моделей любой размерности, однако необходимо учитывать производительность используемого компьютера и возможности пакетов компьютерной алгебры. В настоящей работе задействованы несколько языков про-

граммирования и пакетов компьютерной алгебры, что позволило в полной мере использовать преимущества каждого из них и реализовать многоплановость и многоаспектность поставленной задачи моделирования динамических систем с учетом стохастизации и управления.

Разработанный программный комплекс допускает дальнейшее расширение и усовершенствование. В качестве перспективных направлений исследования можно выделить разработку модуля для моделирования динамики перекрывающихся систем популяционной динамики (например, с учетом сезонных изменений условий взаимодействия популяций), совершенствование модулей программного комплекса с учетом возможностей эффективного применения современных аппаратных средств, разработка модуля с учетом распространения изучаемых моделей популяционной динамики на нестационарный случай (например, для учета трофических функций, зависящих от времени). Кроме того, важным направлением работы является достижение более высокой плотности интеграции модулей с учетом реализации на различных языках программирования.

СПИСОК ЛИТЕРАТУРЫ

1. Кулябов Д.С., Кокотчикова М.Г. Аналитический обзор систем символьных вычислений // Вестник РУДН. Серия: Математика. Информатика. Физика. 2007. № 1–2. С. 38–45.
2. Алтунин К.Ю., Сениченков Ю.Б. О возможности символьных вычислений в пакетах визуального моделирования сложных динамических систем // Информатика, телекоммуникации и управление. 2009. № 3(80). С. 153–158.
3. Малашонок Г.И., Рыбаков М.А. Решение систем линейных дифференциальных уравнений и расчет динамических характеристик систем управления в веб-сервисе mathpartner // Вестник российских университетов. Математика. 2014. № 2. С. 517–529.
4. Банищikov А.В., Бурлакова Л.А., Иртегов В.Д., Титоренко Т.Н. Символьные вычисления в моделировании и качественном анализе динамических систем // Вычислительные технологии. 2014. № 6. С. 3–18.
5. Фалейчик Б.В. Одношаговые методы численного решения задачи Коши. Минск: БГУ, 2010.
6. Platen E. An introduction to numerical methods for stochastic differential equations // Acta Numerica. 1999. V. 8. P. 197–246.
7. Kulchitskiy O., Kuznetsov D. Numerical methods of modeling control systems described by stochastic differential equations // Journal of Automation and Information Sciences. 1999. 06. V. 31. P. 47–61.
8. Gevorkyan M.N., Velieva T.R., Korolkova A.V. et al. Stochastic Runge–Kutta software package for stochastic differential equations // Dependability Engineering and Complex Systems / Ed. by Wojciech Zamojski, Jacek Mazurkiewicz, Jarosław Sugier. Cham: Springer International Publishing, 2016. P. 169–179.
9. Gevorkyan M.N., Demidova A.V., Korolkova A.V., Kulyabov D.S. Issues in the software implementation of stochastic numerical Runge–Kutta // Distributed Computer and Communication Networks / Ed. by Vladimir M. Vishnevskiy, Dmitry V. Kozyrev. Cham: Springer International Publishing, 2018. V. 919 of Communications in Computer and Information Science. P. 532–546. arXiv: 1811.01719.
10. Геворкян М.Н., Демидова А.В., Велиева Т.Р. и др. Реализация метода стохастизации одношаговых процессов в системе компьютерной алгебры // Программирование. 2018. № 2. С. 18–27.
11. Демидова А.В. Уравнения динамики популяций в форме стохастических дифференциальных уравнений // Вестник РУДН. Серия: Математика. Информатика. Физика. 2013. № 1. С. 67–76. URL: <https://journals.rudn.ru/miph/article/view/8319>.
12. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. 2-е изд. изд. Москва: МГТУ им. Н.Э. Баумана, 2016.
13. Firsov A.N., Inovenkov I.N., Nefedov V.V., Tikhomirov V.V. Numerical study of the effect of stochastic disturbances on the behavior of solutions of some differential equations // Modern Information Technologies and IT-Education. 2021. V. 17. № 1. P. 37–43.
14. Mao X. Stochastic Differential Equations and Applications, 2nd ed. Cambridge: Woodhead Publ, 2008.
15. Korolkova A., Kulyabov D. Onestep stochastization methods for open systems // EPJ Web of Conferences. 2020. V. 226. P. 02014. URL: <https://doi.org/10.1051/epjconf/202022602014>
16. Gardiner C.W. Handbook of Stochastic Methods: For Physics, Chemistry and the Natural Sciences. Heidelberg: Springer, 1985.
17. Van Kampen N. Stochastic Processes in Physics and Chemistry. Amsterdam: Elsevier, 1992.
18. Bairey E., Kelsic E.D., Kishony R. High-order species interactions shape ecosystem diversity // Nature Communications. 2016. V. 7. P. 12285.
19. Голубятников В.П., Подколюдная О.А., Подколюдный Н.Л., Аюпова Н.Б., Кириллова Н.Е., Юношева Е.В. Об условиях существования циклов в двух базовых моделях циркадного осциллятора млекопитающих // Сиб. журн. индустр. матем. 2021. Т. 24. № 4. С. 39–53.
20. Вольтерра В. Математическая теория борьбы за существование. Москва: Наука, 1976.
21. Свирижев Ю.М., Логофет Д.О. Устойчивость биологических сообществ. Москва: Наука, 1978.
22. Базыкин А.Д. Нелинейная динамика взаимосвязанных популяций. Москва–Ижевск: Институт компьютерных исследований, 2003.
23. Dilao R. Mathematical Models in Population Dynamics and Ecology // In Biomathematics: Modelling and Simulation. Singapore: World Scientific, 2006. P. 399–449.
24. Четаев Н.Г. Устойчивость движения. Москва: ГИТТЛ, 1964.

25. *Пых Ю.А.* Равновесие и устойчивость в моделях популяционной динамики. Москва: Наука, 1983.
26. *Шестаков А.А.* Обобщенный прямой метод Ляпунова для систем с распределенными параметрами. Москва: Наука, 1990.
27. *Demidova A.V., Druzhinina O.V., Jacimovic M. et al.* The generalized algorithms of global parametric optimization and stochastization for dynamical models of interconnected populations // Optimization and Applications. OPTIMA 2020. Lecture Notes in Computer Science / Ed. by *Nicholas Olenov, Yuri Evtushenko, Michael Khachay, Vlasta Malkova*. V. 12422. Cham: Springer, 2020. P. 40–54.
28. *Demidova A.V., Druzhinina O.V., Masina O.N., Petrov A.A.* Synthesis and computer study of population dynamics controlled models using methods of numerical optimization, stochastization and machine learning // Mathematics. 2021. V. 9. № 24. URL: <https://www.mdpi.com/2227-7390/9/24/3303>.
29. *Demidova A.V., Druzhinina O.V., Jacimovic M., Masina O.N., Mijajlovic N.* Synthesis and analysis of multidimensional mathematical models of population dynamics // Proceedings of the Selected Papers of the 10th International Congress on Ultra Modern Telecommunications and Control Systems ICUMT (Moscow, Russia, November 5–9, 2018). New York: IEEE Xplore Digital Library, 2018. IEEE Catalog Number CFP 1863G-USB. P. 361–366. <https://doi.org/10.1109/ICUMT.2018.8631252>
30. *Demidova A., Druzhinina O., Jacimovic M. et al.* Problems of synthesis, analysis and optimization of parameters for multidimensional mathematical models of interconnected populations dynamics // Optimization and Applications. OPTIMA 2019. Communications in Computer and Information Science / Ed. by *Milojica Jacimovic, Michael Khachay, Vlasta Malkova, Mikhail Posypkin*. V. 1145. Cham: Springer, 2020. P. 56–71.
31. *Demidova A.V., Druzhinina O.V., Masina O.N., Petrov A.A.* Computer research of the controlled models with migration rows // Proceedings of the Selected Papers of the 10th International Conference “Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems” (ITTMM-2020). CEUR Workshop Proceedings. 2020. V. 2639. P. 117–129.
32. *Harris C.R., Millman K.J., van der Walt S.J. et al.* Array programming with NumPy // Nature. 2020. V. 585. № 7825. P. 357–362. URL: <https://doi.org/10.1038/s41586-020-2649-2>
33. *Fuhrer C., Solem J., Verdier O.* Scientific Computing with Python 3. Packt Publishing, 2016.
34. *Lamy R.* Instant SymPy Starter. Packt Publishing, 2013.
35. *Oliphant T.E.* Guide to NumPy. 2nd edition. North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 2015. ISBN: 151730007X.
36. *Virtanen P., Gommers R., Oliphant T.E. et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python // Nature Methods. 2020. V. 17. P. 261–272. URL: <https://doi.org/10.1038/s41592-019-0686-2>
37. *Bezanson J., Edelman A., Karpinski S., Shah V.B.* Julia: A fresh approach to numerical computing // SIAM Review. 2017. V. 59. № 1. P. 65–98.
38. *Meurer A., Smith C.P., Paprocki M. et al.* SymPy: symbolic computing in Python // PeerJ Computer Science. 2017. V. 3. P. e103. URL: <https://doi.org/10.7717/peerj-cs.103>

УДК 004.421.6

КОМПЬЮТЕРНО-АЛГЕБРАИЧЕСКИЕ ВЫЧИСЛЕНИЯ В СУПЕРСИММЕТРИЧНОЙ ЭЛЕКТРОДИНАМИКЕ

© 2023 г. И. Е. Широков^{a,*}^aФизический факультет МГУ им. М.В. Ломоносова
119991 Москва, Ленинские горы, д. 1 с. 2, Россия

*E-mail: shi95@yandex.ru

Поступила в редакцию 02.04.2022 г.

После доработки 21.04.2022 г.

Принята к публикации 20.07.2022 г.

Предлагаются новый символьный алгоритм и C++ программа для генерации и вычисления суперсимметричных диаграмм Фейнмана для $\mathcal{N} = 1$ суперсимметричной электродинамики, регуляризованной высшими производными в четырех измерениях. Программа генерирует, в соответствии со стандартными правилами, все диаграммы, необходимые для вычисления конкретного вклада в двухточечную функцию Грина суперполей материи в рассматриваемом порядке, а затем сводит ответ к сумме евклидовых импульсных интегралов. На данный момент программа применялась для вычисления аномальной размерности в $\mathcal{N} = 1$ суперсимметричной квантовой электродинамике, регуляризованной высшими производными, в трехпетлевом приближении.

DOI: 10.31857/S0132347423020152, EDN: MGXCTG

1. ВВЕДЕНИЕ

Попытки создать программное обеспечение для вычислений в рамках квантовой теории возмущений предпринимаются уже более пятидесяти лет (см. например [1], а также обзор [2]). Разумеется ранние программы были очень ограничены, как правило рассматривалась квантовая электродинамика. Во многом ограничением для развития таких программ была крайне низкая производительность машин того времени. Однако начиная с 90-х годов началось бурное развитие такого рода программного обеспечения. Некоторые программы, созданные в то время, не потеряли актуальности до сих пор (например [3, 4]).

Различные программы для вычислений в физике высоких энергий можно разделить на несколько групп. Во-первых, есть программы, созданные для вычисления различных процессов, как правило, в рамках стандартной модели. Они обычно ограничены древесным (PHEGAS [5], O'Mega [6], MadEvent [7], FDC [8], CompHEP/CalcHEP [9, 10], WHIZARD [11], Herwig++ [12], SHERPA [13]) или в самом лучшем случае однопетлевым приближением (GoSam [14], aMC@NLO [15]). Некоторые из этих программ являются также так называемыми генераторами событий [5, 7, 12, 13], которые могут симулировать результат эксперимента. Данные программы хороши тем, что, как правило, генерируют диаграммы, амплитуды и даже вычисляют фейн-

новские интегралы, однако, они очень ограничены возможными теориями, числом петель (как правило не больше одной), а также областью применения (как правило, это сечения рассеяния, ширины распада и пр.).

Вторая группа программ связана с генерацией амплитуд в более общем случае. Одна из самых известных программ это QGRAPH [3]. Эта программа известна около 30 лет, однако до сих пор активно используется в различных исследованиях. Она генерирует все диаграммы в любом порядке теории возмущений, выводит их в некоем символьном виде, а также считает комбинаторный коэффициент. При этом она не генерирует амплитуды и не рисует диаграмм. Еще одна известная программа такого плана это FeynArts [4, 16]. Она генерирует диаграммы и амплитуды для заданной теории. Также она изображает их графически с помощью специального пакета для LaTeX. В качестве недостатка можно указать, что она ограничена приближением трех петель.

Отдельно нужно отметить различные системы компьютерной алгебры. Аналитические вычисления диаграмм имеют свою специфику, поскольку помимо простых алгебраических действий (которые можно проводить с помощью хорошо известных систем, таких как Mathematica [17], Maple [18], Schoonschip [19], а также FORM [20]), необходимо производить анализ тензорных структур, вычислять следы гамма-матриц и пр. Для этого

существуют ряд пакетов и программ. Например GiNaC [21], Cadabra [22], RedBerry [23], FeynCalc [24].

Есть ряд программ, которые вычисляют фейнмановские импульсные интегралы в d -мерии, которые образуются после построения амплитуд. Как правило, реализуется следующий подход к их вычислению. Сначала тензорные интегралы стандартными способами сводятся к скалярным, а затем с помощью таких приемов, как интегрирование по частям в d -мерии [25] и применение Лоренц-инвариантности, скалярные интегралы сводятся к небольшому числу типовых мастер-интегралов. Этот процесс реализован в программах AIR [26], FIRE [27], LiteRed [28], Reduze [29] и Kira [30]. Также хорошо известны программы по вычислению мастер-интегралов, такие как AMBRE [31], FIESTA [32] и SecDec [33]. Отдельно стоит отметить пакет для Schoonschip Mincer [34], впервые примененный для вычисления четырехпетлевой бета-функции с помощью размерной регуляризации [35].

Наконец, существуют пакеты программ, сочетающие в себе генерацию диаграмм и амплитуд, операции над ними и взятие интегралов. Например aTALC [36], FeynMaster [37], HepLib [38], tapir [39]. В основном все эти пакеты основаны на генерации диаграмм программой QGRAPH, а затем на анализе результатов другими упомянутыми выше программами.

Некоторые из упомянутых программ адаптированы в том числе и к работе с суперсимметричными теориями [8, 9, 13, 40]. Все они работают в рамках минимальной суперсимметричной стандартной модели в терминах компонентных полей. Однако известны две программы для работы с суперполями в суперпространстве. Это программа SUSYCAL [41] написанная на языке PASCAL, а также пакет для Mathematica SusyMath [42]. Эти программы могут работать с выражениями в терминах суперпространства, которые генерируются суперграфами. В теории они могут их упрощать до импульсных интегралов, однако эти проекты не развиваются и на данный момент недоступны для скачивания.

Таким образом, можно отметить, что несмотря на существенный прогресс в этой области, замечен недостаток программного обеспечения для работы в рамках суперсимметричных теорий в терминах суперполей. Даже существующие программы нуждаются в интеграции с теми, которые генерируют диаграммы. К тому же сама генерация графов в суперсимметричном варианте имеет свою специфику. С этой целью автором были разработаны компьютерно-алгебраические подходы для работы с суперполями в суперпространстве. На их основе была создана программа, способная генерировать диаграммы Фейнмана в терминах

суперпространства, а также проводить с ними различные операции, после которых результат выводится в виде стандартных фейнмановских интегралов. На данный момент программу можно использовать для вычислений аномальной размерности суперполей материи в $\mathcal{N} = 1$ суперсимметричной квантовой электродинамике, регуляризованной высшими ковариантными производными в четырех измерениях.

2. ФОРМАЛИЗМ $\mathcal{N} = 1$ СУПЕРПРОСТРАНСТВА

$\mathcal{N} = 1$ суперпространство представляет собой пространство, координатами которого являются (ct, x, y, z, θ) , где θ — майорановский спинор. Спиноры в четырех измерениях преобразуются по специальному спинорному закону относительно группы Лоренца¹. Это пространство вводится, чтобы суперсимметрия была явной, в том числе на квантовом уровне. Индексы поднимаются и опускаются с помощью матриц зарядового сопряжения:

$$\theta^a \equiv \theta_b C^{ba}; \quad \theta_a = \theta^b C_{ab}. \quad (2.1)$$

Суперсимметричная ковариантная производная обычно вводится следующим образом:

$$\bar{D}_a = \frac{\partial}{\partial \bar{\theta}^a} - i(\gamma^\mu)_a{}^b \theta_b \partial_\mu. \quad (2.2)$$

При этом спинорные индексы обозначаются латинскими буквами, правые без точки, а левые с точкой, также левые спиноры имеют черту. Например $\bar{D}_a$ — левая суперсимметричная ковариантная производная, а правая обозначается как D_a . При этом для квадратов производных справедливо:

$$D^2 = D^a D_a \quad \bar{D}^2 = \bar{D}^a \bar{D}_a \quad (2.3)$$

Обычные поля в данном подходе являются компонентами суперполей. Так, калибровочное поле является компонентой вещественного суперполя $V(x^\mu, \theta)$, спинорные и скалярные поля — компонентами кирального или антикирального суперполей $\phi(x^\mu, \theta)$ и $\phi^*(x^\mu, \theta)$ соответственно, которые по определению удовлетворяют условиям:

$$\bar{D}_a \phi = 0, \quad D_a \phi^* = 0 \quad (2.4)$$

Кроме того, при построении суперсимметричных лагранжианов вводится “интегрирование” по перемещению θ . В наших обозначениях его можно определить следующим образом:

¹ Подробнее о спинорах можно прочесть например в [43].

$$\int d^2\bar{\theta} = \frac{1}{2}\bar{D}^2, \quad \int d^2\theta = -\frac{1}{2}D^2 \quad (2.5)$$

$$\int d^4\theta = \int d^2\bar{\theta}d^2\theta$$

3. ТЕОРИЯ ВОЗМУЩЕНИЙ В СУПЕРСИММЕТРИЧНЫХ ТЕОРИЯХ

Теперь рассмотрим, как работает стандартная теория возмущений в терминах суперпространства². Прежде всего необходимо рассмотреть, как записывается действие теории в терминах суперполей. Существуют 2 инварианта относительно преобразований суперсимметрии, которые можно записать следующим образом:

$$S_1 = \int d^4\theta d^4x \mathbb{V}, \quad S_2 = \int d^2\theta d^4x \Phi + \text{к.с.}, \quad (3.1)$$

где $\mathbb{V}$ – вещественное суперполе, а Φ – киральное суперполе.

Удобнее всего проводить квантование методом функционального интегрирования. Основным элементом этого подхода, из которого можно получить различные величины, является производящий функционал Z , который, как известно, получается следующим образом:

$$Z = \int \mathcal{D}(\text{суперполя}) e^{iS + iS_{\text{ист}}}, \quad (3.2)$$

где под $\mathcal{D}(\text{суперполя})$ понимается мера континуального интегрирования. Действие можно представить в следующем виде:

$$S = S^{(2)} + S_{\text{int}}(\phi, V) \quad (3.3)$$

где $S^{(2)}$ – квадратичный по суперполям вклад в действие, а S_{int} – вклад степени выше 2. Далее, к действию добавляются члены с источниками. Например, для вещественных и киральных суперполей они выглядят следующим образом:

$$S_{\text{ист}} = j\phi + JV \quad (3.4)$$

В таком случае (3.2) можно переписать следующим образом:

$$Z = e^{iS_{\text{int}}\left(\frac{1}{i\delta j} \frac{1}{i\delta J}\right)} \int \mathcal{D}(\text{суперполя}) e^{i(S^{(2)} + S_{\text{ист}})} \quad (3.5)$$

Оставшийся гауссов интеграл берется с помощью стандартных методов. Разложение в ряд членов взаимодействия интерпретируется графически с помощью диаграмм Фейнмана.

Естественно, после разложения придется вычислять следующие выражения, которые порождают пропагаторы в диаграммах. Например, для источников калибровочного суперполя:

² Про обычную теорию возмущений в КТП можно прочитать, например, в [44].

$$\frac{\delta}{\delta J_1} \frac{\delta}{\delta J_2} Z_0(J) \quad (3.6)$$

При этом неизбежно возникают следующие структуры:

$$\frac{\delta J_1}{\delta J_2} = \delta_{12}^8 = \delta^4(x_1 - x_2) \delta_{12}^4(\theta), \quad (3.7)$$

где $\delta_{12}^4(\theta)$ определяется следующим образом:

$$\delta_{12}^4(\theta) = \delta^4(\theta_1 - \theta_2) = 4(\theta_1 - \theta_2)^2 (\bar{\theta}_1 - \bar{\theta}_2)^2 \quad (3.8)$$

при таком определении получаются стандартные правила для суперпространственной дельта-функции:

$$\int d^4\theta_1 \delta_{12}^4(\theta) = 1 \int d^4\theta_1 f(\theta_1) \delta_{12}^4(\theta) = f(\theta_2) \quad (3.9)$$

Для источников киральных суперполей справедливо следующее соотношение:

$$\frac{\delta j_1}{\delta j_2} = -\frac{D^2}{2} \delta_{12}^8 \quad (3.10)$$

Кроме того, D -производные могут содержаться в вершинах и самом производящем функционале.

При этом реально обычно приходится рассматривать так называемое эффективное действие. Оно определяется следующим образом. Сначала строится производящий функционал связанных функций Грина:

$$W[\text{Источники}] = -i \ln Z[\text{Источники}], \quad (3.11)$$

а эффективное действие $\Gamma[\text{суперполя}]$ получается из $W[\text{Источники}]$ преобразованием Лежандра:

$$\Gamma = W - S_{\text{source}}|_{\text{источники} \rightarrow \text{суперполя}},$$

$$\frac{\delta W}{\delta j_i} = \phi_i \quad (3.12)$$

по сути дела функционал W выделяет из Z все несвязные диаграммы, а Γ так называемые одночастично-приводимые диаграммы. Это такие диаграммы, которые можно разделить одним разрезанием какой-либо внутренней линии. В качестве примера на рис. 1 верхняя диаграмма является одночастично-неприводимой, а нижняя – одночастично-приводимой.

4. СУПЕРСИММЕТРИЧНАЯ “D-АЛГЕБРА”

Видно, что конкретное выражение для диаграммы Фейнмана будет содержать какое-то количество суперпространственных дельта-функций δ_{12}^8 , на которые действуют суперсимметричные ковариантные производные, при этом все выражение будет интегрироваться по суперпространству. Типичное выражение имеет вид:

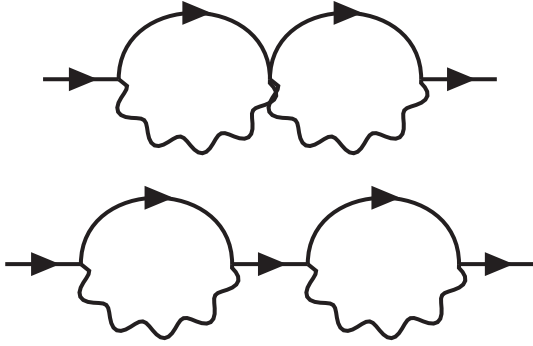


Рис. 1. Одночастично-(не)приводимые диаграммы.

$$\int d^8 x_1 \dots d^8 x_n D \bar{D} \dots D \delta_{12}^8 \times \dots \times D \bar{D} \dots D \delta_{nm}^8 \quad (4.1)$$

При этом, в соответствии с [45], можно определить следующий алгоритм работы с этим выражением:

1. Производим “интегрирование по частям”. Используя (2.5), можно переносить внешнюю D -производную с любого элемента на другой, относящийся к той же точке. Таким образом, можно очистить суперпространственную дельта-функцию от производных и проинтегрировать ее в соответствии с (3.9).

2. Также можно переставлять производные между собой в соответствии с правилами суперсимметричной “ D -алгебры” в четырех измерениях:

$$\{D_a, \bar{D}_b\} = 2i(\gamma^\mu)_{ab} \partial_\mu, \quad (4.2)$$

$$\{D_a, D_b\} = 0, \quad (4.3)$$

$$\{\bar{D}_a, \bar{D}_b\} = 0, \quad (4.4)$$

$$[D_a, \bar{D}^2] = 4i(\gamma^\mu)_a^{\bar{b}} \bar{D}_{\bar{b}} \partial_\mu, \quad (4.5)$$

$$[D_a, D^2] = 0, \quad (4.6)$$

$$[\bar{D}^2, D^2] = 4i(\gamma^\mu)^{\bar{a}a} [D_a, \bar{D}_{\bar{b}}] \partial_\mu, \quad (4.7)$$

$$\bar{D}^2 D^2 \bar{D}^2 = -16\partial^2 \bar{D}^2, \quad (4.8)$$

$$D^2 \bar{D}^2 D^2 = -16\partial^2 D^2. \quad (4.9)$$

Эти тождества помогают уменьшить количество действующих производных.

3. Если в выражении есть несколько одинаковых суперпространственных дельта-функций, то следует применять следующие тождества:

$$\delta_{12}^8 D^\alpha \delta_{12}^8 = 0, \quad \text{если } \alpha \leq 3 \quad (4.10)$$

$$\delta_{12}^8 D^2 \bar{D}^2 \delta_{12}^8 = 4\delta_{12}^8 \delta^4(x_1 - x_2) \quad (4.11)$$

В результате этих действий можно снять большую часть интегрирований и убрать суперпространственные дельта-функции, могут остаться ковариантные производные, действующие на внешние су-

перполя. Также останется некоторое количество следов гамма-матриц, которые раскрываются стандартными способами. После этого выражения приобретают вид обычных импульсных интегралов.

5. АЛГОРИТМ, РЕАЛИЗУЕМЫЙ ПРОГРАММОЙ

Кратко опишем алгоритм работы программы. Мы вычисляем вклад в эффективное действие (3.12). Для этого можно использовать (3.5), если исключить из него все одночастично-приводимые и несвязные диаграммы, а также добавить члены взаимодействия с внешними суперполями. В соответствии с вышесказанным программа действует следующим образом.

1. В формуле (3.5) разлагается экспонента с вершинами взаимодействия до нужного порядка теории возмущений. При этом предварительно в нее добавляются вклады, где производные по источникам всеми возможными способами заменяются на внешние суперполя.

2. Далее во всех получившихся вкладах мы пытаемся спарить производные по источникам в соответствии с пропагаторами теории, как, например, в формуле (3.6). Если в итоге это не удастся, то вариант отбрасывается, если получилось — создается новое выражение, в котором производные по источникам сгруппированы парами.

3. Проверяем получившиеся графы на одночастичную приводимость. Для этого используем функцию, которая убирает по очереди каждый из пропагаторов, после чего проверяет связность. После каждой проверки либо убираем граф, либо оставляем.

4. На этом этапе переходим в импульсное представление, генерируем список петлевых импульсов исходя из порядка разложения. Петлевые импульсы являются базисными импульсами, через линейную комбинацию которых выражаются остальные импульсы внутренних линий диаграммы. Каждый петлевой импульс кодируется простым числом. Далее каждому пропагатору присваиваем импульс, исходя из закона сохранения в вершинах диаграммы, и заменяем спаренные производные по источникам на конкретное выражение для пропагатора, которое содержит оператор, действующий на суперпространственную дельта-функцию (также он может содержать импульсы и константы). После этого каждый граф также проверяется на соблюдение закона сохранения в каждой точке, если что-то сгенерировалось не так, программа выдает ошибку.

5. Теперь выражения готовы для работы с D -алгеброй. Каждое выражение обрабатывается отдельно. Мы рассматриваем каждый оператор и выбираем тот, у которого меньше всего ковари-

антных производных, начинаем перебрасывать с него интегрированием по частям все операторы, пока полностью не очистим суперпространственную дельта-функцию, на которую он действует. При этом получается множество новых выражений, каждое из которых мы затем также отдельно обрабатываем. Когда суперпространственная дельта-функция очищена, мы ищем такую же и, если находим, то применяем правила (4.2)–(4.9), чтобы уменьшить число производных, а затем (4.10) и (4.11), чтобы устранить суперпространственную дельта-функцию. Если такой суперпространственной дельта-функции нет, интегрирование просто снимается. Действуем так, пока не устраним все суперпространственные дельта-функции. Затем, если внешние импульсы не равны нулю, проделываем то же самое для суперполей материи, учитывая их киральность.

6. Теперь мы имеем выражения с внешними суперполями, импульсами и прочими структурами, а также гамма-матрицами, дельта-символами и матрицами зарядового сопряжения со спинорными индексами. На этом шаге мы сворачиваем все индексы с дельта-символов и матриц зарядового сопряжения, а затем берем след гамма-матриц. Вообще в наших обозначениях это не совсем чистый след γ^μ , в нем также есть γ_5 , что может приводить к структурам, содержащим антисимметричный тензор $\epsilon_{\mu\nu\lambda\delta}$, но он всегда будет сворачиваться с петлевыми импульсами и, следовательно, не будет давать вклад, вплоть до пятиточечных функций Грина, которые мы пока не рассматриваем³. Нужно отметить, что в отличие от вычислений в d-мерии [46], мы работаем в четырех измерениях и поэтому все стандартные соотношения для γ -матриц справедливы в нашем случае.

7. После этого мы получаем вклады в виде квадратов и скалярных произведений импульсов. При этом импульс задается не строкой, а числом. Это сделано для удобства работы с ним, можно применять стандартные алгоритмы нахождения наибольшего общего делителя и пр. Каждый импульс задается простым числом, а сумма их произведением, таким образом каждое число однозначно задает сумму. Например: у нас есть импульс k , l и q . Припишем k – число 2, l – число 3, q – число 5. Тогда, например, $k + l$ будет 6, а $k + l + q$ – 30. Как правило, при построении диаграмм всем линиям можно приписать именно сумму импульсов. Если в какой-то момент это будет не так, то, например, квадрат можно представить $(k - l)^2 = -(k + l)^2 + 2k^2 + 2l^2$. При этом проще работать с полными квадратами, выделяя их

из общего выражения. В качестве примера рассмотрим следующее выражение:

$$(q + k + l)^\mu (q + k)_\mu = (q + k)^2 + l^\mu (q + k)_\mu = \\ = (q + k)^2 + \frac{1}{2}(q + k + l)^2 - \frac{1}{2}l^2 - \frac{1}{2}(q + k)^2 \quad (5.1)$$

В обозначениях программы это произведение импульса 30 ($q + k + l$) и 10 ($q + k$). Сначала ищем наибольший общий делитель, он равен 10 ($q + k$), а затем частное каждого с НОД (в данном случае это 3(l) и 1(0)) как произведение с каждым из других сомножителей (10 ($q + k$) и 30 ($q + k + l$) соответственно). Если наибольший общий делитель 1, то раскладываем в полные квадрат стандартным образом (как во второй части (5.1)). В конце для каждого отдельного графа пытаемся привести подобные.

8. На последнем этапе мы пытаемся привести подобные между разными графами. При этом также используем некоторые возможные замены переменных, поскольку все итоговые выражения предполагаются проинтегрированными по петлевым импульсам. Реализованы замены такого типа:

$$k \rightarrow k + l, \quad k \rightarrow k + l + q \quad \text{и т.д.}, \quad (5.2)$$

а также все возможные перестановки импульсов:

$$k \leftrightarrow l, \quad k \leftrightarrow q \quad \text{и т.д.} \quad (5.3)$$

При этом действуем следующим образом. Пробуем совершить все возможные замены этих двух классов, при этом каждой замене ставим в соответствие некий вес (который представляет сумму значений всех импульсов) и выбираем такую замену, у которой вес наименьший. В общем случае этот минимум не единственный, но в большинстве ситуаций это так. Потом проводим прямое сравнение всех выражений и приводим подобные, это достаточно сильно упрощает выражение.

Следует также сделать несколько общих замечаний. Почти на всех этапах проводится приведение подобных в текущих на тот момент структурах. Обычно это делается прямым почленным сравнением выражения. На первом этапе, где происходит спаривание производных по источникам, приведение подобных производится путем перестановки точек и дальнейшего сравнения, пока не перебраны все перестановки. Это ускоряет работу программы на порядки. Также на этапе генерации и непосредственной обработки каждого графа программа разделяется на потоки с помощью библиотеки OpenMP [47].

6. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ

Опишем примеры использования программы. Будем рассматривать $\mathcal{N} = 1$ суперсимметричную электродинамику [45, 48, 49] с N_f -суперполями материи различных типов (которые в дальней-

³ Подробнее об алгебре γ -матриц можно прочитать например в [43].

шем будем называть ароматами) регуляризованную высшими производными [50–53]. Регуляризация проводится в четырех измерениях. Действие исходной теории имеет вид:

$$S = \frac{1}{4e_0^2} \text{Re} \int d^4x d^2\theta W^a W_a + \frac{1}{4} \sum_{\alpha=1}^{N_f} \int d^4x d^4\theta (\phi_\alpha^* e^{2V} \phi_\alpha + \tilde{\phi}_\alpha^* e^{-2V} \tilde{\phi}_\alpha), \quad (6.1)$$

где e_0 — голая константа связи, V — калибровочное суперполе, ϕ_α — N_f киральных суперполей материи. Для того, чтобы регуляризовать теорию, необходимо добавить в действие регуляризационную функцию $R\left(\frac{\partial^2}{\Lambda^2}\right)$, на которую накладываются следующие условия:

$$R(0) = 1, \quad R(\infty) = \infty \quad (6.2)$$

Тогда регуляризованное действие теории примет вид:

$$S = \frac{1}{4e_0^2} \text{Re} \int d^4x d^2\theta W^a R\left(\frac{\partial^2}{\Lambda^2}\right) W_a + \frac{1}{4} \sum_{\alpha=1}^{N_f} \int d^4x d^4\theta (\phi_\alpha^* e^{2V} \phi_\alpha + \tilde{\phi}_\alpha^* e^{-2V} \tilde{\phi}_\alpha) \quad (6.3)$$

Также необходимо добавить член, фиксирующий калибровку:

$$S_{\text{gf}} = -\frac{1}{32e_0^2\xi_0} \int d^4x d^4\theta D^2 V K\left(\frac{\partial^2}{\Lambda^2}\right) \bar{D}^2 V, \quad (6.4)$$

тут функция $K\left(\frac{\partial^2}{\Lambda^2}\right)$ удовлетворяет таким же условиям (6.2) как и функция $R\left(\frac{\partial^2}{\Lambda^2}\right)$. Для регуляризации в однопетлевом приближении нужно добавить также детерминанты Паули–Вилларса [54]. Они могут быть представлены в виде континуальных интегралов по соответствующим киральным суперполям Φ и $\tilde{\Phi}$:

$$(\det(PV, M))^{-1} = \int D\Phi D\tilde{\Phi} \exp(iS_{PV}), \quad (6.5)$$

где действие для суперполей Паули–Вилларса определяется выражением:

$$S_{PV} = -\frac{1}{4} \int d^4x d^4\theta (\Phi^* e^{2V} \Phi + \tilde{\Phi}^* e^{-2V} \tilde{\Phi}) + \frac{1}{2} \int d^4x d^2\theta (M\Phi\tilde{\Phi} + \text{c.c.}), \quad (6.6)$$

где $M = a\Lambda$ — масса суперполей Паули–Вилларса.

Итоговое выражения для действия имеет вид:

$$S_{\text{полн}} = S + S_{\text{source}} + S_{\text{reg}} + S_{\text{gf}} \quad (6.7)$$

Производящий функционал может быть записан в следующей форме:

$$Z[J, j, \tilde{j}] = \int D V D \Phi D \tilde{\Phi} (\det(PV, M))^{N_f} \times \exp(iS + iS_{\text{reg}} + iS_{\text{source}} + iS_{\text{gf}}), \quad (6.8)$$

где $J, j, \tilde{j}$ — источники. Исходя из этого можно получить, что пропагаторы в такой теории для обычных суперполей и суперполей Паули–Вилларса (массивных) имеют вид:

$$P(\phi_{\alpha,x}, \phi_{\beta,y}^*) = P(\tilde{\phi}_{\alpha,x}, \tilde{\phi}_{\beta,y}^*) = \delta_{\alpha\beta} \frac{\bar{D}_x^2 D_y^2}{4\partial^2} \delta_{xy}^8; \\ P(\Phi_x, \tilde{\Phi}_y) = \frac{M\bar{D}^2}{\partial^2 + M^2} \delta_{xy}^8; \\ P(\Phi_x^*, \tilde{\Phi}_y^*) = \frac{MD^2}{\partial^2 + M^2} \delta_{xy}^8; \quad (6.9)$$

$$P(\Phi_x, \Phi_y^*) = P(\tilde{\Phi}_x, \tilde{\Phi}_y^*) = \frac{\bar{D}_x^2 D_y^2}{4(\partial^2 + M^2)} \delta_{xy}^8$$

Пропагатор же калибровочного суперполя выглядит следующим образом:

$$P(V_x, V_y) = 2e_0^2 \left[-\frac{1}{R\partial^2} + \frac{1}{16\partial^4} \left(\frac{\xi_0}{K} - \frac{1}{R} \right) \times \right. \\ \left. \times (\bar{D}^2 D^2 + D^2 \bar{D}^2) \right] \delta_{xy}^8 \quad (6.10)$$

Вершины в данном случае могут быть легко построены прямым разложением действия теории.

Регулятор $\frac{1}{R}$ далее в примере будет называться K4.

При этом индексы ароматов опустим, в программе пропорциональность вклада N_f определяется числом замкнутых петель материи.

Программа на данный момент представляет собой консольное приложение, данные считываются из текстового файла. На вход программы необходимо отправить:

1. Тип вычисления. То есть какой именно вклад в эффективное действие мы считаем.
2. Различные опции, например пометка, что мы работаем с суперсимметричной квантовой электродинамикой с N_f -ароматами.
3. Порядок разложения и число петель.
4. Список пропагаторов и вершин теории.

Рассмотрим пример входного файла для вычисления в одной петле в калибровке Фейнмана (это означает, что $\xi_0 = 1$ и $K = R$ в (6.10)).

Type:

F#_1^1{-1}*F_1^1{1}

Option:

SQED

Nf^0

Loops:

1

Order:

2

Propagators:

V_1^1{1}*V_2^1{1}

2*i*e^0*I{1^1-2}*K4{1}*d_{1}{12}

F#_1^1{1}*F_2^1{1}

-1/4*i*e^0*I{1^1-2}*D_1(D#_2(d_{1}{12}))

Vertexes:

1/2*i*e*F#_1^1{6}*V_1^1{-3}*F_1_1{-2}

В нашем примере мы рассматриваем вклад в эффективное действие с внешними двумя суперполями материями, при этом рассматривается случай нулевых внешних импульсов. Мы рассматриваем теорию с N_f ароматами и вычисляем только вклады, в которых нет N_f .

Теперь приведем вид вывода для этого примера:

Файл считан корректно

Идет генерация 1-петлевых диаграмм

Завершено 100%

Диаграммы созданы

Идет расстановка импульсов...

Генерация диаграмм завершена

Генерация диаграмм заняла 0.027 сек

Идет вычисление диаграмм...

0

$$-1/8*e^2*F#_1^1{-1}*F_2_2{1}*d_{-2}{12}*D_2(D#_1(d_{2}{12}))*K4{2}*I{2^1-2}*I{2^1-2}$$

Вычисление диаграмм завершено

Вычисление диаграмм заняло 0.018 сек

Идет приведение подобных...

Приведение подобных завершено

Результат:

$$-1/2*e^2*F#_1^1{-1}*F_1_2{1}*K4{2}*I{2^1-4}$$

Общее время работы 0.052 сек

Каждый шаг в принципе поясняется, в процессе генерации выводится процент от сделанного на данный момент. Затем выводится время, которое потребовалось для генерации. Когда идет вычисление диаграмм, то для каждой диаграммы выво-

дится ее номер (начиная с 0) и выражение для нее. В конце выводится результат. В данном случае это выражение переводится на аналитический язык следующим образом:

$$-\frac{1}{2} \int \frac{d^4 k}{(2\pi)^4} d^4 \theta \phi_{\alpha}^*(0, \theta) \phi_{\alpha}(0, \theta) \frac{1}{R_k k^4} \quad (6.11)$$

Вообще, все вводные выражения предполагаются проинтегрированными по полному суперпространству, а выводные по $d^4 \theta$ и по всем петлевым импульсам. Внешние импульсы в данном случае положены равными нулю, но если это не так, то интегрирование предполагается и по ним. Поскольку связь эффективного действия с двухточечной функцией Грина выражается формулой:

$$\Gamma^{(2)} \sim \frac{1}{4} \sum_{\alpha=1}^{N_f} \int \frac{d^4 q}{(2\pi)^4} d^4 \theta (\phi_{\alpha}^*(-q, \theta) \phi_{\alpha}(q, \theta) + \tilde{\phi}_{\alpha}^*(-q, \theta) \tilde{\phi}_{\alpha}(q, \theta)) G(\alpha_0, \Lambda/q), \quad (6.12)$$

то вклад в функцию Грина (в которой внешний импульс положен равным нулю) имеет вид:

$$\Delta G|_{q=0} = -2e_0^2 \int \frac{d^4 K}{(2\pi)^4} \frac{1}{K^4 R_K} \quad (6.13)$$

Этот результат согласуется с результатами статьи [55]. Кроме того, он соответствует формуле (57) из статьи [56], в которой $\xi_0 = 1$. Кроме того стоит отметить, что программа выводит выражение в виде евклидовых интегралов, если все входные выражения также писать в виде евклидовых импульсов. Изменение меры в результате поворота Вика и прочие преобразования программа делает автоматически.

7. ТЕХНИЧЕСКАЯ ИНФОРМАЦИЯ И ВРЕМЯ РАБОТЫ

Программа использовалась для вычислений вплоть до трех петель. С результатами трехпетлевого вычисления можно ознакомиться в статье [57].

В конце приведем время, которое заняли различные вычисления данной программой. Вычисления проводились на следующей конфигурации:

Операционная система: Windows 10 x64

Процессор: AMD Ryzen 5 1600 Six-Core Processor 3.20 GHz

Оперативная память: 8 Гб

Компилятор: GNU GCC Compiler

Опции компиляции: -march=native, -O3

Условно разделим их на три части в зависимости от степени числа ароматов. При этом поскольку эта степень определяется числом замкнутых петель материи, то некоторые вклады существуют

Таблица 1. Время работы, вклад N_f^0

	1 петля	2 петли	3 петли
$\xi_0 = 1$	0.052 сек	0.14 сек	2.6 сек
$\xi_0 \neq 1$	0.067 сек	0.57 сек	2 мин 27 сек

Таблица 2. Время работы, вклад N_f^1

	2 петли	3 петли
$\xi_0 = 1, m = 0$	0.16 сек	6.6 сек
$\xi_0 \neq 1, m = 0$	0.52 сек	13 мин 49 сек
$\xi_0 = 1, m \neq 0$	0.41 сек	41.5 сек
$\xi_0 \neq 1, m \neq 0$	1.23 сек	3 ч 54 мин

Таблица 3. Время работы, вклад N_f^2

	3 петли
$\xi_0 = 1, m = 0$	4.2 сек
$\xi_0 \neq 1, m = 0$	6.6 сек
$\xi_0 = 1, m \neq 0$	35 сек
$\xi_0 \neq 1, m \neq 0$	2 мин 58 сек

только в определенном порядке. Кроме того, будем рассматривать также случаи минимальной ($\xi_0 = 1$) и неминимальной ($\xi_0 \neq 1$) калибровок, а также отсутствие или наличие массивных суперполей в теории m . Время работы приведено в табл. 1, 2, 3. При этом нужно отметить, что эти вычисления проводились в минимально допустимой для каждого случая теории, это означает, что при рассмотрении, например, случая ($\xi_0 = 1$) рассматривались вершины, в которых число внешних линий калибровочного суперполя не превышает число петель, диаграммы, в которых это не так, равны нулю в такой калибровке в соответствии с (4.10). Это немного уменьшало время работы.

8. ЗАКЛЮЧЕНИЕ

Была создана программа для генерации и вычисления суперпространственной части диаграмм Фейнмана для двухточечных функций Грина суперсимметричной электродинамики. Она была проверена на вычислениях в одной и двух петлях и использована для вычисления в трех петлях, в том числе в неминимальной калибровке ($\xi_0 \neq 1$). Время работы в большинстве вычислений оказалось приемлемым (как правило —

не больше нескольких минут), что дает возможность в дальнейшем модернизировать ее для вычислений в более интересных теориях, таких как суперсимметричная теория Янга—Миллса.

9. БЛАГОДАРНОСТИ

Автор выражает глубокую благодарность К.В. Степаньянцу за помощь в изучении методов вычислений в суперпространстве, а также за внимательное чтение текста статьи и важные замечания. Кроме того автор благодарит С.В. Морозова за помощь и ценные консультации о возможностях языка C++ и пакета OpenMP, а также за внимательное чтение статьи и важные замечания. Также автор благодарит А.Л. Катаева за ценные замечания.

СПИСОК ЛИТЕРАТУРЫ

1. *Campbell J.A., Hearn A.C.* Symbolic analysis of feynman diagrams by computer // J. Comput. Phys. 1970. V. 5. P. 280.
2. *Гердт В.П., Тарасов О.В., Ширков Д.В.* Аналитические вычисления на ЭВМ в приложении к физике и математике // УФН, 1980. Т. 130. С. 113–147.
3. *Nogueira P.* Automatic Feynman graph generation // J. Comput. Phys. 1993. V. 105. P. 279–289.
4. *Kublbeck J., Bohm M., Denner A.* Feyn Arts: Computer Algebraic Generation of Feynman Graphs and Amplitudes // Comput. Phys. Commun. 1990. V. 60. P. 165–180. a8 citations counted in INSPIRE as of 04 Feb 2022
5. *Papadopoulos C.G.* PHEGAS: A Phase space generator for automatic cross-section computation // Comput. Phys. Commun., 2001. V. 137. P. 247–254.
6. *Moretti M., Ohl T., Reuter J.* O'Mega: An Optimizing matrix element generator // [arXiv:hep-ph/0102195 [hep-ph]].
7. *Maltoni F., Stelzer T.* MadEvent: Automatic event generation with MadGraph // JHEP. 2003. V. 02. P. 027.
8. *Wang J.X.* Progress in FDC project // Nucl. Instrum. Meth. A, 2004. V. 534. P. 241–245.
9. *Boos E. et al. [CompHEP]* CompHEP 4.4: Automatic computations from Lagrangians to events // Nucl. Instrum. Meth. A. 2004. V. 534. P. 250–259.
10. *Belyaev A., Christensen N.D., Pukhov A.* CalcHEP 3.4 for collider physics within and beyond the Standard Model // Comput. Phys. Commun. 2013. V. 184. P. 1729–1769.
11. *Kilian W., Ohl T., Reuter J.* WHIZARD: Simulating Multi-Particle Processes at LHC and ILC // Eur. Phys. J. C. 2011. V. 71. P. 1742.
12. *Bahr M., Gieseke S., Gigg M.A., Grellscheid D., Hamilton K., Latunde-Dada O., Platzer S., Richardson P., Seymour M.H., Sherstnev A., et al.* Herwig++ Physics and Manual // Eur. Phys. J. C. 2008. V. 58. P. 639–707.
13. *Gleisberg T., Hoeche S., Krauss F., Schonherr M., Schumann S., Siegert F., Winter J.* Event generation with SHERPA 1.1 // JHEP. 2009. V. 02. P. 007.

14. Cullen G., van Deurzen H., Greiner N., Heinrich G., Luisi G., Mastrolia P., Mirabella E., Ossola G., Peraro T., Schlenk J., et al. GOSAM-2.0: a tool for automated one-loop calculations within the Standard Model and beyond // *Eur. Phys. J. C*. 2014. V. 74. № 8. P. 3001.
15. Alwall J., Frederix R., Frixione S., Hirschi V., Maltoni F., Mattelaer O., Shao H.S., Stelzer T., Torrielli P., Zaro M. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations // *JHEP*. 2014. V. 07. P. 079.
16. Hahn T. Generating Feynman diagrams and amplitudes with FeynArts 3 // *Comput. Phys. Commun.* 2001. V. 140. P. 418–431.
17. Wolfram Wolfram Mathematica, 2022. <https://www.wolfram.com/mathematica/>
18. Maplesoft, a division of Waterloo Maple Inc. Maple, 2022. <https://www.maplesoft.com/products/maple/>
19. Veltman M.J.G., Williams D.N. Schoonschip'91 // [arXiv:hep-ph/9306228 [hep-ph]].
20. Ruijl B., Ueda T., Vermaseren J. FORM version 4.2 // [arXiv:1707.06453 [hep-ph]].
21. Vollinga J. GiNaC: Symbolic computation with C++ // *Nucl. Instrum. Meth. A*. 2006. V. 559. P. 282–284.
22. Peeters K. A Field-theory motivated approach to symbolic computer algebra // *Comput. Phys. Commun.* 2007. V. 176. P. 550–558.
23. Bolotin D.A., Poslavsky S.V. Introduction to Redberry: a computer algebra system designed for tensor manipulation // [arXiv:1302.1219 [cs.SC]].
24. Shtabovenko V., Mertig R., Orellana F. FeynCalc 9.3: New features and improvements // *Comput. Phys. Commun.* 2020. V. 256. P. 107478.
25. Chetyrkin K.G., Tkachov F.V. Integration by Parts: The Algorithm to Calculate beta Functions in 4 Loops // *Nucl. Phys. B*. 1981. V. 192. P. 159–204.
26. Anastasiou C., Lazopoulos A. Automatic integral reduction for higher order perturbative calculations // *JHEP*. 2004. V. 07. P. 046.
27. Smirnov A.V., Chuharev F.S. FIRE6: Feynman Integral REDuction with Modular Arithmetic // *Comput. Phys. Commun.* 2020. V. 247. P. 106877.
28. Lee R.N. LiteRed 1.4: a powerful tool for reduction of multiloop integrals // *J. Phys. Conf. Ser.* 2014. V. 523. P. 012059.
29. Studerus C. Reduze-Feynman Integral Reduction in C++ // *Comput. Phys. Commun.* 2010. V. 181. P. 1293–1300.
30. Maierhöfer P., Usovitsch J., Uwer P. Kira—A Feynman integral reduction program // *Comput. Phys. Commun.* 2018. V. 230. P. 99–112.
31. Dubovik I., Gluza J., Riemann T., Usovitsch J. Numerical integration of massive two-loop Mellin-Barnes integrals in Minkowskian regions // *PoS*. 2016. V. LL2016. P. 034.
32. Smirnov A.V. FIESTA4: Optimized Feynman integral calculations with GPU support // *Comput. Phys. Commun.* 2016. V. 204. P. 189–199.
33. Borowka S., Heinrich G., Jones S.P., Kerner M., Schlenk J., Zirke T. SecDec-3.0: numerical evaluation of multi-scale integrals beyond one loop // *Comput. Phys. Commun.* 2015. V. 196. P. 470–491.
34. Gorishnii S.G., Larin S.A., Surguladze L.R., Tkachov F.V. Mincer: Program for Multiloop Calculations in Quantum Field Theory for the Schoonschip System // *Comput. Phys. Commun.* 1989. V. 55. P. 381–408.
35. Gorishnii S.G., Kataev A.L., Larin S.A., Surguladze L.R. The Analytical four loop corrections to the QED Beta function in the MS scheme and to the QED psi function: Total reevaluation // *Phys. Lett. B*. 1991. V. 256. P. 81–86.
36. Lorca A., Riemann T. Automated calculations for massive fermion production with aTALC // *Nucl. Phys. B Proc. Suppl.* 2004. V. 135. P. 328–332.
37. Fontes D., Romão J.C. FeynMaster: a plethora of Feynman tools // *Comput. Phys. Commun.* 2020. V. 256. P. 107311.
38. Feng F., Xie Y.F., Zhou Q.C., Tang S.R. HepLib: A C++ library for high energy physics // *Comput. Phys. Commun.* 2021. V. 265. P. 107982.
39. Gerlach M., Herren F., Lang M. tapir: A tool for topologies, amplitudes, partial fraction decomposition and input for reductions // [arXiv:2201.05618 [hep-ph]].
40. Hahn T., Schappacher C. The Implementation of the minimal supersymmetric standard model in FeynArts and FormCalc // *Comput. Phys. Commun.* 2002. V. 143. P. 54–68.
41. Kreuzberger T., Kummer W., Schweda M. SUSYCAL: A PROGRAM FOR SYMBOLIC COMPUTATIONS IN SUPERSYMMETRIC THEORIES // *Comput. Phys. Commun.* 1990. V. 58. P. 89–104.
42. Ferrari A.F. SusyMath: A Mathematica package for quantum superfield calculations // *Comput. Phys. Commun.* 2007. V. 176. P. 334–346.
43. Степаньянц К.В. Классическая теория поля. М.: ФИЗМАТЛИТ, 2009. 540 с.
44. Боголюбов Н.Н., Ширков Д.В. Введение в теорию квантованных полей. М.: Наука, 1973. 416 с.
45. Уэст П. Введение в суперсимметрию и супергравитацию. Пер. с англ. М.: Мир, 1989. 328 с.; West P.C. Introduction to supersymmetry and supergravity. Singapore: World Scientific, 1990. 425 p.
46. Tarasov O.V., Vladimirov A.A. Three Loop Calculations in Non-Abelian Gauge Theories // *Phys. Part. Nucl.* 2013. V. 44. P. 791–802.
47. OpenMP ARB OpenMP 5.2 Reference Guide, 2021. <https://www.openmp.org/wp-content/uploads/OpenMPRefCard-5-2-web.pdf>
48. Gates S.J., Grisaru M.T., Rocek M., Siegel W. Superspace Or One Thousand and One Lessons in Supersymmetry // *Front. Phys.* 1983. V. 58. P. 1–548.
49. Buchbinder I.L., Kuzenko S.M. Ideas and Methods of Supersymmetry and Supergravity: Or a Walk Through Superspace, IOP, Bristol, UK, 1998. 656 p.
50. Slavnov A.A. Invariant regularization of nonlinear chiral theories // *Nucl. Phys. B*. 1971. V. 31. P. 301–315.
51. Славнов А.А. Инвариантная регуляризация калибровочных теорий // *ТМФ*. 1972. Т. 13:2. С. 174–177.

52. *Кривошеков В.К.* Инвариантная регуляризация для суперсимметричных калибровочных теорий // ТМФ. 1978. Т. 36. С. 291.
53. *West P.C.* Higher Derivative Regulation Of Supersymmetric Theories // Nucl. Phys. B. 1986. V. 268. P. 113.
54. *Славнов А.А.* Регуляризация Паули–Вилларса для неабелевых калибровочных групп // ТМФ. 1977. Т. 33:2. С. 210–217.
55. *Катаев А.Л., Степаньянц К.В.* β -Функция Новикова–Шифмана–Вайнштейна–Захарова в суперсимметричных теориях при различных регуляризациях и перенормировочных предписаниях // ТМФ. 2014. Т. 181:3. С. 475–486.
56. *Aleshin S.S., Durandina I.S., Kolupaev D.S., Korneev D.S., Kuzmichev M.D., Meshcheriakov N.P., Novgorodtsev S.V., Petrov I.A., Shatalova V.V., Shirokov I.E. et al.* Three-loop verification of a new algorithm for the calculation of a β -function in supersymmetric theories regularized by higher derivatives for the case of $N = 1$ SQED // Nucl. Phys. B. 2020. V. 956. P. 115020.
57. *Shirokov I.E., Stepanyantz K.V.* The three-loop anomalous dimension and the four-loop β -function for $\mathcal{N} = 1$ SQED regularized by higher derivatives // JHEP. 2022. V. 2204. P. 108.