

СОДЕРЖАНИЕ

Номер 1, 2021

Памяти Н.П. Трифонова 3

КОМПЬЮТЕРНАЯ АЛГЕБРА

Применение методов компьютерной алгебры для вычисления решения задачи Коши для двумерного разностного уравнения в точке <i>М. С. Апанович, А. П. Ляпин, К. В. Шадрин</i>	5
Символьное исследование собственных векторов для построения общего решения системы ОДУ с символьной матрицей коэффициентов <i>Д. В. Диваков, А. А. Тютюнник</i>	11
Использование шаблонизатора как инструментария компьютерной алгебры <i>М. Н. Геворкян, А. В. Королькова, Д. С. Кулябов</i>	25
Метод цветных графов для упрощения выражений с индексами <i>Г. Б. Шпиз, А. П. Крюков</i>	35
Алгоритм для решения семейства диофантовых уравнений четвертой степени, удовлетворяющих условию Рунге <i>Н. Н. Осипов, А. А. Кытманов</i>	39
Построение частичных Лорановых решений усеченных дифференциальных систем <i>А. А. Панфёров</i>	45
Поиск равновесных состояний машины Атвуда с двумя колеблющимися грузами с применением компьютерной алгебры <i>А. Н. Прокопеня</i>	56
Эвристические алгоритмы распознавания некоторых кубических гиперповерхностей <i>А. В. Селиверстов</i>	65

CONTENTS

No. 1, 2021

Obituary in Memory of N.P. Trifonova 3

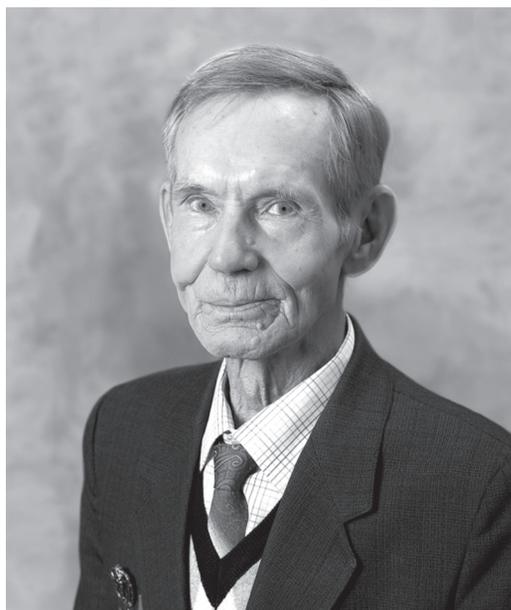
COMPUTER ALGEBRA

- Application of Computer Algebra Methods to Calculate the Solution
to a Problem Cauchy for a Two-Dimensional Difference Equation at a Point
M. S. Apanovich, A. P. Lyapin, K. V. Shadrin 5
- Symbolic Study of Eigenvectors to Construct a General Solutions
of the ODE System with a Symbolic Coefficient Matrix
D. V. Divakov, A. A. Tyutyunnik 11
- Using a Templating Engine as a Computer Algebra Toolkit
M. N. Gevorkyan, A. V. Korolkova, D. S. Kulyabov 25
- Colored Graph Method to Simplify Expressions with Indices
G.B. Shpiz, A.P. Kryukov 35
- Algorithm for Solving a Family of Diophantine Equations
of the Fourth Degree, Satisfying the Runge Condition
N. N. Osipov, A. A. Kytmanov 39
- Construction of Partial Laurent Solutions of Truncated Differential Systems
A. A. Panferov 45
- Searching for Equilibrium States of Atwood's Machine with Two Oscillating Bodies
with Application of Computer Algebra
A.N. Prokopenya 56
- Heuristic Algorithms for Recognition of Some Cubic Hypersurfaces
A. V. Seliverstov 65
-
-

ПАМЯТИ Н.П. ТРИФОНОВА

DOI: 10.31857/S0132347421010027

20 июля 2020 г. скончался Николай Павлович Трифонов — заслуженный работник высшей школы РФ, профессор, один из основоположников отечественного программирования.



Н.П. Трифонов родился 2 декабря 1925 г. в деревне Кулига Кирилловского района Вологодской области в крепкой крестьянской семье. В возрасте 4 лет лишился матери, его отец остался один с четырьмя детьми: Николаем и его тремя сестрами. После раскулачивания семья фактически осталась без средств к существованию.

Николай Павлович учился в средней школе г. Кириллова. В 1943 г., когда Н.П. Трифонову исполнилось 18 лет, со школьной скамьи он был призван в ряды Красной Армии. С июня по сентябрь 1943 г. в составе пехотного полка он сражался на передовой в районе Курска, награжден медалью “За отвагу”. В сентябре 1943 г. Н.П. Трифонов получил тяжелое ранение, до января 1944 г. находился на лечении. Затем последовала учеба в танковом училище, 7 апреля 1945 г. Н.П. Трифонову было присвоено звание младшего лейтенанта. Военскую службу он продолжил в Уральском военном округе в качестве преподавателя учебного танкового полка, расположенного в г. Верхний Уфалей Челябинской области. Здесь Николай Павлович и встретил День Победы. Вернувшись

домой, Николай Павлович закончил обучение в Кирилловской средней школе.

В 1946 г. Н.П. Трифонов поступил на механико-математический факультет МГУ. В 1951 г. Николай Павлович окончил с отличием университет и поступил в аспирантуру. Написанная под руководством акад. С.Л. Соболева диссертация была защищена в 1954 г.

Трудовая биография Николая Павловича — живая история отечественного программирования. Начало научной деятельности Н.П. Трифопова совпало с периодом возникновения первых отечественных вычислительных машин, и Николай Павлович внес вклад в основные научные направления, связанные с вычислительной техникой: применение ЭВМ для научных расчетов, создание математического обеспечения вычислительных машин и методика обучения будущих ученых-программистов.

С 1955 по 1962 год Н.П. Трифонов заведовал математической лабораторией Вычислительного Центра МГУ. В конце 50-х годов под руководством Николая Павловича там проводились космические расчеты на машине “Стрела”. Центра управления полетами тогда еще не существовало. В 1961 г. за успешное выполнение вычислительных работ по обеспечению запусков космических аппаратов Николай Павлович был награжден медалью “За трудовую доблесть”. Н.П. Трифонов принимал участие в пионерских работах по автоматизации программирования для вычислительных машин: он являлся руководителем работ по созданию универсальной системы программирования для машины “Стрела”, включающей транслятор, систему отладки программ и библиотеку стандартных подпрограмм — одной из первых систем подобного рода. Позже модификация этой системы была установлена на машинах серии М-20.

В 1962 году Н.П. Трифонов перешел на преподавательскую работу на кафедру вычислительной математики механико-математического факультета МГУ. В 1964 году вышел в свет учебник “Курс программирования” Е.А. Жоголева и Н.П. Трифопова — первый учебник, в котором обучение программированию построено на основе алгоритмического языка высокого уровня. Учебник выдержал несколько переизданий и стал основой вводных курсов программирования в университетах.

В 1970 г. был образован факультет вычислительной математики и кибернетики МГУ, Н.П. Трифонов стал ученым секретарем учебно-методического совета факультета. Николай Павлович оказал огромное влияние на формирование принципов обучения программированию на факультете ВМК МГУ и в стране в целом. В 1980 году Н.П. Трифонов (в соавторстве с Э.З. Любимским и В.В. Мартынюком) создал учебник “Программирование”, содержащий материал базового двухгодичного курса начального обучения программированию. Образец систематического подхода, учебник не потерял актуальности и сейчас. В 1975 году Н.П. Трифоновым был опубликован “Сборник упражнений по Алголу”, определивший стиль многих последующих задачников по программированию.

Н.П. Трифонов сыграл большую роль в организации научного семинара “Автоматизация программирования”, который начал работу в МГУ в 1956 г. и со временем обрел характер городского. Длительное время Н.П. Трифонов входил в со-

став редколлегии журнала “Вестник МГУ. Серия Вычислительная математика и кибернетика”.

Большое внимание Николай Павлович Трифонов уделял воспитанию преподавательских кадров. С 1971 г. по 1993 г. Н.П. Трифонов заведовал кафедрой алгоритмических языков факультета ВМК МГУ. Именно в этот период произошло становление педагогического коллектива кафедры. Николая Павловича отличали строгое внимание и забота о молодых преподавателях; он находил время для посещения лекций и семинаров своих сотрудников, скрупулезно и в то же время деликатно, с глазу на глаз, обсуждал преподавательские ошибки, давал советы. Сотрудники кафедры тепло и с благодарностью вспоминают это время.

Память о Николае Павловиче Трифонове – замечательном человеке, выдающемся ученом, преподавателе, организаторе образования, останется в сердцах коллег и учеников.

Редколлегия журнала "Программирование"

**ПРИМЕНЕНИЕ МЕТОДОВ КОМПЬЮТЕРНОЙ АЛГЕБРЫ
ДЛЯ ВЫЧИСЛЕНИЯ РЕШЕНИЯ ЗАДАЧИ КОШИ
ДЛЯ ДВУМЕРНОГО РАЗНОСТНОГО УРАВНЕНИЯ В ТОЧКЕ**

© 2021 г. М. С. Апанович^{a,b,*}, А. П. Ляпин^{b,**}, К. В. Шадрин^{a,***}

^a Красноярский государственный медицинский университет имени профессора В.Ф. Войно-Ясенецкого,
660022 Красноярск, ул. Партизана Железняка, 1, Россия

^b Сибирский федеральный университет,
660041 Красноярск, пр. Свободный, 79, Россия

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

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

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

Поступила в редакцию 15.07.2020 г.

После доработки 06.09.2020 г.

Принята к публикации 12.09.2020 г.

В данной работе представлен алгоритм вычисления решения задачи Коши для двумерного разностного уравнения с постоянными коэффициентами в точке по коэффициентам разностного уравнения и начальным данным задачи Коши методами компьютерной алгебры. В одномерном случае решение задачи Коши для разностного уравнения не представляет сложности, однако уже в двумерном случае число неизвестных растет на каждом шаге очень быстро. Для автоматизации процесса вычисления решения задачи Коши для двумерного разностного уравнения с постоянными коэффициентами в заданной точке в среде MATLAB был разработан алгоритм, где входными данными являются: матрица коэффициентов, полученная исходя из структуры двумерного полиномиального разностного уравнения; координаты точки, регламентирующей структуру матрицы начальных данных; координаты точки, регламентирующей размерность матрицы начальных данных; матрица начальных данных. Результатом работы алгоритма является решение задачи Коши для двумерного разностного уравнения, представляющее собой значение функции в искомой точке.

DOI: 10.31857/S0132347421010039

1. ВВЕДЕНИЕ

Линейные разностные уравнения имеют многочисленные приложения в различных областях науки и техники, а поиск их решений является сложной математической задачей. Например, разностные уравнения часто используются в моделях динамики с дискретным временем ([1, 2]), для приближенного (численного) решения дифференциальных уравнений ([3]), а в комбинаторном анализе разностные уравнения в сочетании с методом производящих функций (дискретным аналогом преобразования Фурье) дают мощный аппарат исследования перечислительных задач (см., например, [4–6]). Для пространства решений многомерного разностного уравнения задаются дополнительные условия (“начальные”, “граничные”, “данные Коши”), которые позволяют из бесконечного множества решений выделить един-

ственное, а возникающая при этом задача называется задачей Коши для многомерного разностного уравнения.

Разработке алгоритмов решения многомерных разностных уравнений с постоянными и полиномиальными коэффициентами посвящено значительное число работ (см., например, [7–9]), а в [10] рассмотрены разностные уравнения с коэффициентами в виде рациональных функций. Алгоритм вычисления производящей функции решения задачи Коши для двумерного разностного уравнения с постоянными коэффициентами по коэффициентам разностного уравнения и начальным данным задачи Коши представлен в работах [11] и [12]. В данной работе представлен алгоритм вычисления решения задачи Коши для двумерного разностного уравнения с постоянными коэффициентами в точке по коэффициентам

разностного уравнения и начальным данным задачи Коши.

2. ПОСТАНОВКА ЗАДАЧИ И ИЗВЕСТНЫЕ РЕЗУЛЬТАТЫ

Обозначим через \mathbb{Z} множество целых чисел, $\mathbb{Z}^n = \overbrace{\mathbb{Z} \times \dots \times \mathbb{Z}}^n$ – n -мерную целочисленную решетку и \mathbb{Z}_+^n – подмножество этой решетки, состоящее из точек с целыми неотрицательными координатами (в частности, \mathbb{Z}^2 и \mathbb{Z}_+^2 – двумерная решетка и ее подмножество). Пусть δ_1 – оператор сдвига по переменной x_1 , т.е. $\delta_1 f(x_1, x_2) = f(x_1 + 1, x_2)$, а δ_2 – оператор сдвига по переменной x_2 , т.е. $\delta_2 f(x_1, x_2) = f(x_1, x_2 + 1)$.

Рассмотрим полиномиальный разностный оператор вида

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

где $\alpha = (\alpha_1, \alpha_2)$ – мультииндекс, $|\alpha| = \alpha_1 + \alpha_2$, $\delta^\alpha = \delta_1^{\alpha_1} \cdot \delta_2^{\alpha_2}$, c_α – постоянные коэффициенты, m – порядок оператора $P(\delta)$.

Будем рассматривать разностные уравнения вида

$$P(\delta_1, \delta_2)f(x_1, x_2) = 0, \quad (x_1, x_2) \in \mathbb{Z}_+^2, \quad (1)$$

где $f(x_1, x_2)$ – неизвестная функция.

Для точек x и y решетки \mathbb{Z}^n неравенство $x \geq y$ означает, что $x_i \geq y_i$ для $i = 1, \dots, n$, а запись $x \not\geq y$ означает, что найдется $i_0 \in \{1, \dots, n\}$ такое, что $x_{i_0} < y_{i_0}$.

Фиксируем $\beta = (\beta_1, \beta_2)$ такое, что $\beta \neq (0, m)$, $\beta \neq (m, 0)$, $|\beta| = m$ и $c_\beta \neq 0$. Обозначим

$$X_{0,\beta} = \{(x_1, x_2) \in \mathbb{Z}_+^2 : (x_1, x_2) \not\geq \beta\}$$

и сформулируем задачу: *найти функцию $f(x_1, x_2)$, которая для всех $(x_1, x_2) \in X_{0,\beta}$ совпадает с заданной функцией $\varphi(x_1, x_2)$, т.е. удовлетворяет условию*

$$f(x) = \varphi(x), \quad x \in X_{0,\beta}. \quad (2)$$

Задачу (1)–(2) будем называть задачей Коши для полиномиального разностного оператора $P(\delta)$, $\varphi(x)$ – функцией начальных данных этой задачи, а $f(x)$ – решением задачи Коши.

Известно (см. [13, 14]), что задача (1)–(2) однозначно разрешима, если выполняется условие

$$|c_\beta| > \sum_{\substack{|\alpha| = |\beta| \\ \alpha \neq \beta}} |c_\alpha|. \quad (3)$$

Поставим задачу: вычислить значение функции $f(x_1, x_2)$ в точке с координатами (y_1, y_2) .

3. ОПИСАНИЕ ВХОДНЫХ ДАННЫХ

Решение задачи Коши (1)–(2) для двумерного разностного уравнения с постоянными коэффициентами в точке $y = (y_1, y_2)$ представляет собой значение функции $f(x_1, x_2)$ в заданной точке. Алгоритм вычисления значения функции $f(x_1, x_2)$ в заданной точке $y = (y_1, y_2)$ имеет рекурсивный характер и сводится к вычислению значений функции $f(x_1, x_2)$ на конечном подмножестве точек x из множества точек $X_{0,y} = \{(x_1, x_2) \in \mathbb{Z}_+^2 : (x_1, x_2) \not\geq (y_1, y_2)\}$ и удовлетворяющих условию $|x_1 + x_2| \leq |y_1 + y_2|$.

Начальные данные (2) задаются квадратной матрицей F , содержащей конечное подмножество значений начальных данных задачи Коши. Матрица начальных данных F будет иметь размерность $(y_1 + y_2 + 1) \times (y_1 + y_2 + 1)$.

Коэффициенты двумерного разностного уравнения задаются квадратной матрицей C , имеющей нижнетреугольный вид.

Проиллюстрируем процедуру задания функции начальных данных на примере.

Для разностного уравнения

$$\begin{aligned} & c_{02}f(x_1, x_2 + 2) + c_{01}f(x_1, x_2 + 1) + \\ & + c_{11}f(x_1 + 1, x_2 + 1) + c_{10}f(x_1 + 1, x_2) + \\ & + c_{20}f(x_1 + 2, x_2) + c_{00}f(x_1, x_2) = 0 \end{aligned} \quad (4)$$

и $\beta = (1, 1)$ матрица коэффициентов C имеет вид

$$C = \begin{pmatrix} c_{02} & 0 & 0 \\ c_{01} & c_{11} & 0 \\ c_{00} & c_{10} & c_{20} \end{pmatrix},$$

а матрица начальных данных F размерности, например, 4×4 будет иметь вид

$$F = \begin{pmatrix} \varphi(0, 3) & & & \\ \varphi(0, 2) & * & & \\ \varphi(0, 1) & * & * & \\ \varphi(0, 0) & \varphi(1, 0) & \varphi(2, 0) & \varphi(3, 0) \end{pmatrix}.$$

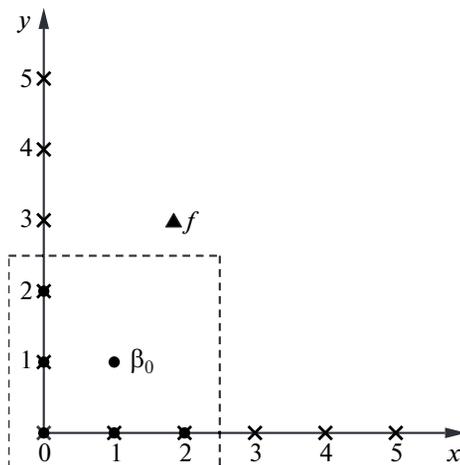


Рис. 1. Расположение элементов матрицы C в декартовой системе координат.

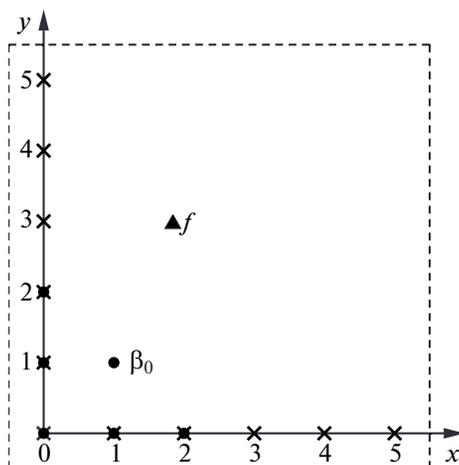


Рис. 2. Расположение элементов матрицы F в декартовой системе координат.

Здесь элементы, обозначенные *, вычисляются при выполнении алгоритма. Например $\varphi(1, 1)$ вычисляется с помощью разностного уравнения (4) и начальных данных $\varphi(0, 2)$, $\varphi(0, 1)$, $\varphi(0, 0)$, $\varphi(1, 0)$, $\varphi(2, 0)$.

Итак, входные данные конечны и имеют вид:

1. точка $\beta = (\beta_1, \beta_2)$, определяющая размерность матрицы коэффициентов C , $\beta_1 + \beta_2 + 1 = m$;
2. нижнетреугольная матрица $C = (c_{\alpha_1, \alpha_2})$, $\alpha_1 = 0, \dots, m$, $\alpha_2 = 0, \dots, m$ размера $(m + 1) \times (m + 1)$ из коэффициентов c_{α_1, α_2} двумерного разностного уравнения;

3. точка f с координатами (y_1, y_2) , определяющая координаты искомого значения функции $f(x_1, x_2)$ и размерность матрицы начальных данных F ;

4. матрица начальных данных $F = (\varphi(x_1, x_2))$, размера $(y_1 + y_2 + 1) \times (y_1 + y_2 + 1)$, где $(x_1, x_2) \in X_{(0, \beta)}$, для всех остальных значений (x_1, x_2) значения $\varphi(x_1, x_2) = 0$.

Поскольку координаты элементов матрицы коэффициентов разностного оператора и матрицы начальных данных в декартовой системе координат (X, Y) не совпадают с их координатами в матрице (строка \times столбец), то для технической реализации работы алгоритма целесообразно перейти из

декартовой системы координат $(D(d_1, d_2))$ в “матричную” $(M(m_1, m_2))$ по правилу: $D(d_1, d_2) \rightarrow M(m_1, m_2)$, где $m_1 = p - d_2$, $m_2 = d_1 + 1$, p – размерность матрицы коэффициентов или матрицы начальных данных.

Например, элемент c_{00} матрицы коэффициентов C , имеющий в декартовой системе координат координаты $(d_1, d_2) = (0, 0)$, в “матричной” системе координат будет иметь координаты $(m_1, m_2) = (3, 1)$, а, например, элемент $f(1, 0)$ матрицы начальных данных F в “матричной” системе координат будет иметь координаты $(4, 2)$.

Далее необходимо будет проверить задачу Коши (1)–(2) на разрешимость, т.е. проверить, выполняется ли условие (3) для коэффициентов разностного оператора $P(\delta)$.

4. ПРИМЕР

Алгоритм был реализован в среде Matlab 2014 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.

Пример 1. Фиксируем $\beta = \beta_0 = (1, 1)$.

Будем рассматривать полиномиальный разностный оператор

$$P(\delta_1, \delta_2) = c_{02}\delta_2^2 + c_{01}\delta_2 + c_{11}\delta_1\delta_2 + c_{10}\delta_1 + c_{20}\delta_1^2 + c_{00}.$$

Множество начальных данных будет иметь вид:

$$X_{0,(1,1)} = \{(x_1, x_2) \in \mathbb{Z}_+^2 : (x_1, x_2) \not\equiv (1, 1)\}$$

Зададим матрицу коэффициентов полиномиального разностного оператора $P(\delta_1, \delta_2)$

$$C = \begin{pmatrix} c_{02} & 0 & 0 \\ c_{01} & c_{11} & 0 \\ c_{00} & c_{10} & c_{20} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 10 & 0 \\ 4 & 2 & 4 \end{pmatrix}.$$

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

Поставим задачу найти значения функции f в точке с координатами $(2, 3)$, т.е. $f(2, 3)$.

Зададим матрицу начальных данных:

$$X_{0,(1,1)} = F = \begin{pmatrix} 7 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 \\ 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix}.$$

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

1. Переход из декартовой системы координат в “матричную”:

$p = 3$ – размерность матрицы коэффициентов
 $i = 6$ – размерность матрицы начальных данных

$\beta_1 = [2, 2]$ – координаты точки β_0 в “матричной” системе координат.

“Матричная” система координат для коэффициентов разностного оператора $P(\delta)$:

	1	2	3
1	•		
2	•	• β_0	
3	•	•	•

Здесь • – коэффициенты c_α .

$F_1 = [3, 3]$ – координаты точки f в “матричной” системе координат.

“Матричная” система координат начальных данных:

	1	2	3	4	5	6
1	×					
2	×					
3	×		f			
4	×					
5	×					
6	×	×	×	×	×	×

Здесь × – начальные данные.

2. Проверка матрицы C на разрешимость задачи Коши (1)–(2): $|10| = |c_{1,1}| > |c_{0,2}| + |c_{2,0}| = 5$, следовательно выполняется условие (3) и задача Коши разрешима.

3. Вектор диагонали матрицы коэффициентов C : $a = (1 \ 10 \ 4)^T$.

4. Получение общей теплицевой матрицы [15], размер которой зависит от искомого значения $f(y_1, y_2)$:

$$column_c = (10 \ 1 \ 0 \ 0)^T,$$

$$row_c = (10 \ 4 \ 0 \ 0),$$

$$T_1 = \begin{pmatrix} 10 & 4 & 0 & 0 \\ 1 & 10 & 4 & 0 \\ 0 & 1 & 10 & 4 \\ 0 & 0 & 1 & 10 \end{pmatrix}.$$

5. Решение системы линейных уравнений с целью нахождения значений f , которые могут потребоваться для вычисления искомого значения.

На каждом шаге размер матрицы Тейлица постоянно увеличивается с ростом числа неизвестных, а также происходит запись найденных значений функции F в матрицу начальных данных F .

$$F = \begin{pmatrix} 7 & 0 & 0 & 0 & 0 & 0 \\ 1 & -0.02 & 0 & 0 & 0 & 0 \\ 2 & -4.90 & -1.98 & 0 & 0 & 0 \\ 3 & 0.72 & 7.14 & 4.33 & 0 & 0 \\ 4 & -7.50 & -4.82 & -7.26 & -7 & 0 \\ 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix}$$

6. Переход к декартовой системе координат $f(2,3) = F(3,3)$.

Output: значение функции в искомой точке $f(2,3) = -1.98$.

Входными данными алгоритма в этом случае будут:

1. $\beta_0 = (1, 1)$;
2. матрица коэффициентов

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 10 & 0 \\ 4 & 2 & 4 \end{pmatrix};$$

3. $f_0 = (2, 3)$;
4. матрица начальных данных

$$F = \begin{pmatrix} 7 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 \\ 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix}.$$

`>Plan((1,1), (1 3 4, 0 10 2, 0 0 4), (2, 3), (7 1 2 3 4 5, 0 0 0 0 0 6, 0 0 0 0 0 7, 0 0 0 0 0 8, 0 0 0 0 0 9, 0 0 0 0 0 10));`

Алгоритм 1: Пример оформления алгоритма

Input: Точка β_0 , матрица коэффициентов C , точка f_0 , матрица начальных данных F .

Output: Значение функции $f(x_1, x_2)$ в точке с координатами (y_1, y_2) .

Procedure Plan($\mathbf{B}_0, C; f_0; F$)

begin

$p :=$ размерность матрицы C

$i :=$ размерность матрицы F

$\beta_1 :=$ координаты точки β_0 в “матричной” системе координат

$f_1 :=$ координаты точки f_0 в “матричной” системе координат

if $|C(\beta_1(1), \beta_1(2))| \leq$

$\sum \text{diag}(|C|) - |C(\beta_1(1), \beta_1(2))|$ **then**

┌ Ошибка ввода матрицы C

$a := \text{diag}(C)$ вектор диагонали матрицы C

$e := \beta_1(1)$

$column_c := \text{zeros}(i - 2, 1)$

$col := 1$

while $e \geq 1$ **do**

┌ $column_c(col) := a(e)$

$col := col + 1$

$e := e - 1$

$row_c := \text{zeros}(l, i - 2)$

$e := \beta_1(1)$

$r := 1$

$l :=$ длина вектора a

for m from $\beta_1(1)$ to l **do**

┌ $row_c(r) = a(e)$

$r := r + 1$

$e := e + 1$

$T_1 := \text{toeplitz}(column_c, row_c)$

for k from 1 to $i - p + 1$ **do**

┌ $T := T_1(1 : k, 1 : k)$

$b := \text{zeros}(k, 1)$

for s from 1 to k **do**

┌ $P := f((\text{end} - p + (s - k + 1)) :$

$(\text{end} + (s - k)), (s : (p + s - 1)))$

$P_{work} := P \circ C$

$b(s) = -(\sum (P_{work}((1 : \text{end}), 1)) +$

$\sum (P_{work}(\text{end}, (2 : \text{end}))))$

$elements := T^{-1} \cdot b$

for m from 1 to k **do**

┌ $F(\text{end} + m - (k + 1) - (p - \beta_1(1)) +$

$1, m + \beta_1(2) - 1) := elements(m)$

return $f(y_1, y_2)$

СПИСОК ЛИТЕРАТУРЫ

1. *Даджион Д., Мерсеро О.* Цифровая обработка многомерных сигналов. М.: Мир, 1988.
2. *Изерман Р.* Цифровые системы управления. М.: Мир, 1984. 541 с.
3. *Рябенский В.С.* Введение в вычислительную математику: учеб. пособие, изд. 2-е, исправл., М.: ФИЗМАТЛИТ, 2000. 296 с.
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. *Vousquet-Melou M., Petkovsek M.* Linear recurrences with constant coefficients: the multivariate case // *Discrete Mathematics*, 2000, V. 225. P. 51–75.
7. *Абрамов С.А., Геффар А., Хмельнов Д.Е.* Рациональные решения линейных разностных уравнений: универсальные знаменатели и границы знаменателей // *Программирование*. 2011. № 2. С. 28–39.
8. *Абрамов С.А.* Поиск рациональных решений дифференциальных и разностных систем с помощью формальных рядов // *Программирование*. 2015. № 2. С. 69–80.
9. *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. P. 1205–1228.
10. *Abramov S., Petkovšek M., Ryabenko A.* Hypergeometric solutions of first-order linear difference systems with rational-function coefficients // *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2015. V. 9301. P. 1–14.
11. *Leinartas E.K., Lyapin A.P.* On the Rationality of Multidimensional Recursive Series // *Journal of Siberian Federal University. Mathematics & Physics*. 2009. V. 2 (2). P. 449–455.
12. *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.
13. *Рогозина М.С.* О разрешимости задачи Коши для полиномиального разностного оператора // *Вестник Новосибирского государственного университета*. 2014. Т. 14. № 3. С. 83–94.
14. *Лейнартас Е.К., Рогозина М.С.* Разрешимость задачи Коши для полиномиального разностного оператора и мономиальные базисы факторов в кольце полиномов // *Сибирский математический журнал*. 2015. Т. 56. № 1. С. 111–121.
15. *Иохвидов И.С.* Ганкелевы и теплицевы матрицы. М.: Наука, 1974. 264 с.

СИМВОЛЬНОЕ ИССЛЕДОВАНИЕ СОБСТВЕННЫХ ВЕКТОРОВ ДЛЯ ПОСТРОЕНИЯ ОБЩЕГО РЕШЕНИЯ СИСТЕМЫ ОДУ С СИМВОЛЬНОЙ МАТРИЦЕЙ КОЭФФИЦИЕНТОВ

© 2021 г. Д. В. Диваков^{a,*}, А. А. Тютюнник^{a,**}

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

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

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

Поступила в редакцию 14.07.2020 г.

После доработки 05.08.2020 г.

Принята к публикации 13.09.2020 г.

В работе исследуется задача символьного представления общего решения системы обыкновенных дифференциальных уравнений с постоянными коэффициентами, заданными в символьном виде, при условии, что некоторые символьные константы могут обращаться в ноль. Кроме того, символьное представление собственных векторов матрицы коэффициентов системы не единственно.

В работе на примере исследуемой системы показано, что стандартные процедуры компьютерной алгебры отыскивают конкретные символьные представления собственных векторов, игнорируя существование других символьных представлений собственных векторов. В свою очередь предлагаемые системой компьютерной алгебры собственные векторы могут быть непригодны для построения численных алгоритмов на их основе, что продемонстрировано в работе.

Авторами предложен алгоритм отыскания различных символьных представлений собственных векторов символьно заданных матриц. В работе рассматривается конкретная система дифференциальных уравнений, полученная при исследовании решений уравнений Максвелла, однако предложенный алгоритм исследования применим к произвольной системе с нормальной матрицей коэффициентов.

DOI: 10.31857/S0132347421010040

1. ВВЕДЕНИЕ

Существуют различные подходы к моделированию оптических явлений, которые можно разбить на две большие группы: символьный и численный.

1.1. Символьный подход

Символьный подход состоит в ряде аналитических упрощений уравнений, выделяющих важные частные случаи, допускающие при этом исследование на символьном уровне [1, 2]. При таком подходе стараются получать символьные либо символьно-численные выражения, позволяющие качественно исследовать упрощенную модель.

1.2. Численный подход

В рамках численного подхода за постановкой задачи следует построение приближенного метода ее решения – либо разностного [3–5], либо ка-

кого-то иного, вроде методов Галеркина [6–8], Канторовича [9, 10] или метода конечных элементов [11, 12]. Численный подход требует от исследователя внимательного построения математического метода решения задачи и почти всегда исключает аналитическое упрощение представленной модели и угадывания вида решения. Однако в численном подходе можно строить методы, подходящие для исследования широкого класса задач с большим произволом параметров модели.

1.3. Цель исследования

Авторский подход относится к группе символьных методов решения уравнений Максвелла, отвечающих волноводному распространению в плавно-нерегулярных протяженных волноводных структурах [1, 2].

Основа подхода – представление электромагнитного поля в виде асимптотического ряда [13], в котором явно разделены амплитудная и фазовая ча-

сти поля. Такое представление электромагнитного поля позволяет в нулевом приближении редуцировать уравнения Максвелла к двум алгебраическим уравнениям и системе четырех обыкновенных дифференциальных уравнений относительно амплитудных частей электромагнитного поля, причем матрица коэффициентов этой системы задана в символьном виде. Общее решение такой системы обыкновенных дифференциальных уравнений можно записать с помощью собственных векторов и собственных значений матрицы коэффициентов. Так как матрица коэффициентов задана в символьном виде, собственные значения и собственные векторы нужно по этой причине отыскивать в символьном виде.

Настоящая работа посвящена исследованию различных возможных вариантов символьного представления собственных векторов матрицы коэффициентов системы, от которого зависит представление общего решения и дальнейшие численные расчеты. Особенность исследования также состоит в том, что некоторые символьные константы в матрице коэффициентов могут обратиться в ноль, что также нужно учитывать при символьных манипуляциях с целью исключения возникновения особенностей, которые подробно описаны в следующем разделе.

2. МОТИВАЦИЯ

В рамках предлагаемого авторами подхода получается система обыкновенных дифференциальных уравнений следующего вида [2]

$$\frac{\partial \vec{u}}{\partial x} + A(x, y, z) \vec{u} = \vec{0}, \quad (2.1)$$

в которую входит производная искомой вектор-функции \vec{u} только по x , однако коэффициенты системы есть кусочно-постоянные функции переменной x , а также функции от y, z , поэтому искомая величина \vec{u} зависит от x, y, z .

Если рассмотреть систему в какой-нибудь области постоянства коэффициентов по x , то она будет иметь вид

$$\frac{\partial \vec{u}_j}{\partial x} + A_j(y, z) \vec{u}_j = \vec{0}, \quad (2.2)$$

то есть это система обыкновенных дифференциальных уравнений с постоянными по x коэффициентами, так как коэффициенты системы зависят только от y, z . Также важно отметить, что среди коэффициентов системы встречаются неизвестные функции от y, z , которые определяются из условия разрешимости системы граничных уравнений [2]. Система граничных уравнений формируется из записи граничных условий [2], следующих из условий сопряжения электромагнитных

полей на всех поверхностях разрыва коэффициентов системы (2.1).

Для каждой системы вида (2.2), отвечающей каждой области постоянства коэффициентов по x , можно записать ее общее решение с помощью собственных векторов и собственных значений матрицы коэффициентов, в которые будут входить в виде символьных выражений неизвестные функции от y, z . Поэтому для записи общего решения необходимо в символьном виде решать задачу на собственные значения и собственные векторы квадратной матрицы, элементы которой заданы в символьном виде. Подобные возможности заложены в различных системах компьютерной алгебры.

Практика символьного отыскания собственных векторов в системе Maple показывает, что символьное представление собственного вектора может быть не единственно. Другими словами, для каждого собственного значения символьно заданной матрицы может существовать некоторое множество различных символьных представлений собственного вектора, поэтому важно иметь алгоритм отыскания множеств различных символьных представлений собственных векторов. Он необходим для выбора наиболее подходящих для последующих этапов исследования символьных представлений собственных векторов.

В символьное представление собственных векторов могут входить неизвестные функции от y, z , поэтому необходимо, чтобы:

1) символьное представление собственного вектора имело смысл для всей области значений неизвестных функций: например, если функция находится в знаменателе выражения и может обращаться в ноль, то это выражение может устремиться к бесконечности и использовать его в общем случае некорректно;

2) линейно независимые собственные векторы оставались таковыми для всей области значений неизвестных функций, если это возможно: например, если функция может обращаться в ноль, то она может обнулять компоненты собственных векторов и, тем самым, линейно независимые собственные векторы могут стать линейно зависимыми.

Учитывая описанные особенности рассматриваемой системы дифференциальных уравнений вида (2.1), для построения эффективного численного алгоритма ее решения необходимо исследовать символьное представление собственных векторов ее матрицы коэффициентов с учетом требований 1) и 2) выше. Исследованию этой задачи и посвящена настоящая работа.

Результаты исследования ценны как промежуточный шаг, на базе которого строится последующий численный алгоритм расчета электромагнитного поля.

3. МЕТОДЫ И АЛГОРИТМЫ

3.1. Асимптотическое разложение решения

3.1.1. Вид решения. Электромагнитное поле представляем в виде асимптотического ряда [13], в котором явно разделены амплитудная и фазовая части поля [2]:

$$\begin{aligned} \bar{E}(x, y, z, t) &= \\ &= \sum_{s=0}^{\infty} \frac{\bar{E}^s(x, y, z)}{(-i\omega)^{\gamma+s}} \exp\{i\omega t - ik_0\varphi(y, z)\}, \end{aligned} \quad (3.1)$$

$$\begin{aligned} \bar{H}(x, y, z, t) &= \\ &= \sum_{s=0}^{\infty} \frac{\bar{H}^s(x, y, z)}{(-i\omega)^{\gamma+s}} \exp\{i\omega t - ik_0\varphi(y, z)\}, \end{aligned} \quad (3.2)$$

где k_0 – волновое число, $\varphi(y, z)$ определяет фазу, а $\bar{E}^s(x, y, z)$, $\bar{H}^s(x, y, z)$ определяют амплитуду s -го порядка. В записи $\bar{E}^s(x, y, z)$, $\bar{H}^s(x, y, z)$ отделение x точкой с запятой означает следующее предположение: $\partial \bar{E}^s / \partial y$, $\partial \bar{E}^s / \partial z$, $\partial \bar{H}^s / \partial y$, $\partial \bar{H}^s / \partial z$ являются малыми величинами. Другими словами справедливы следующие выражения для производных:

$$\partial \bar{E} / \partial y = -ik_0\varphi_y \bar{E}, \quad \partial \bar{E} / \partial z = -ik_0\varphi_z \bar{E},$$

и аналогичные выражения:

$$\partial \bar{H} / \partial y = -ik_0\varphi_y \bar{H}, \quad \partial \bar{H} / \partial z = -ik_0\varphi_z \bar{H},$$

в которых φ_y и φ_z – частные производные $\varphi(y, z)$ по y и z соответственно.

3.1.2. Редукция уравнений Максвелла – общий случай. Уравнения Максвелла в нулевом приближении ($s = 0$) асимптотического разложения редуцируются к системе обыкновенных дифференциальных уравнений первого порядка [2]:

$$\frac{\partial \bar{u}}{\partial x} + A(x, y, z)\bar{u} = \bar{0}, \quad (3.3)$$

и двум дополнительным соотношениям

$$\begin{aligned} E_x^0 &= \varepsilon^{-1}(\varphi_z H_y^0 - \varphi_y H_z^0), \\ H_x^0 &= -\mu^{-1}(\varphi_z E_y^0 - \varphi_y E_z^0), \end{aligned} \quad (3.4)$$

где искома вектор-функция \bar{u} содержит величины $\bar{u}(x; y, z) = (E_y^0 \ H_z^0 \ H_y^0 \ E_z^0)^T$, которые описывают распределение по x соответствующей компоненты поля в каждой точке (y, z) . Матрица A определена следующим образом [2]

$$A(x, y, z) = ik_0 \begin{pmatrix} 0 & -\frac{\varphi_y^2}{\varepsilon} + \mu & \frac{\varphi_y \varphi_z}{\varepsilon} & 0 \\ -\frac{\varphi_z^2}{\mu} + \varepsilon & 0 & 0 & \frac{\varphi_y \varphi_z}{\mu} \\ -\frac{\varphi_y \varphi_z}{\mu} & 0 & 0 & \frac{\varphi_y^2}{\mu} - \varepsilon \\ 0 & -\frac{\varphi_y \varphi_z}{\varepsilon} & \frac{\varphi_z^2}{\varepsilon} - \mu & 0 \end{pmatrix}, \quad (3.5)$$

где $\varepsilon = \varepsilon(x, y, z)$ и $\mu = \mu(x, y, z)$ есть кусочно-постоянные функции – диэлектрическая и магнитная проницаемости соответственно. В каждом слое волновода с постоянными диэлектрической и магнитной проницаемостями ε, μ решение системы дифференциальных уравнений (3.3) можно представить в следующем виде [2], который получен с помощью функции Eigenvectors пакета LinearAlgebra в Maple [14]:

$$\begin{aligned} \bar{u}(x, y, z) &= A \begin{pmatrix} q \\ -i\varepsilon\eta \\ 0 \\ \varphi_y \varphi_z \end{pmatrix} e^{\gamma x} + B \begin{pmatrix} -i\mu\eta \\ p \\ \varphi_y \varphi_z \\ 0 \end{pmatrix} e^{\gamma x} + \\ &+ C \begin{pmatrix} q \\ i\varepsilon\eta \\ 0 \\ \varphi_y \varphi_z \end{pmatrix} e^{-\gamma x} + D \begin{pmatrix} i\mu\eta \\ p \\ \varphi_y \varphi_z \\ 0 \end{pmatrix} e^{-\gamma x}, \end{aligned} \quad (3.6)$$

где $q = \varphi_y^2 - \varepsilon\mu$, $p = \varphi_z^2 - \varepsilon\mu$, $\eta = \sqrt{\varphi_y^2 + \varphi_z^2 - \varepsilon\mu}$, $\gamma = k_0\eta$, а A, B, C, D – неопределенные константы в каждой точке (y, z) .

Замечание. Собственному значению $\gamma = k_0\sqrt{\varphi_y^2 + \varphi_z^2 - \varepsilon\mu}$ отвечают два собственных вектора

$$v_1^+ = \begin{pmatrix} q \\ -i\varepsilon\eta \\ 0 \\ \varphi_y \varphi_z \end{pmatrix}, \quad v_2^+ = \begin{pmatrix} -i\mu\eta \\ p \\ \varphi_y \varphi_z \\ 0 \end{pmatrix}, \quad (3.7)$$

а собственному значению $-\gamma$ отвечают два других собственных вектора

$$v_1^- = \begin{pmatrix} q \\ i\varepsilon\eta \\ 0 \\ \varphi_y \varphi_z \end{pmatrix}, \quad v_2^- = \begin{pmatrix} i\mu\eta \\ p \\ \varphi_y \varphi_z \\ 0 \end{pmatrix}. \quad (3.8)$$

При $\varphi_y \varphi_z \neq 0$ они линейно независимы.

3.1.3. Редукция уравнений Максвелла – частный случай. Уравнения Максвелла в нулевом приближении ($s = 0$) асимптотического разложения

при $\varphi_y \equiv 0$ редуцируются к системе обыкновенных дифференциальных уравнений первого порядка [2]:

$$\frac{\partial \bar{u}}{\partial x} + A^0(x, y, z)\bar{u} = \bar{0}, \quad (3.9)$$

и двум дополнительным соотношениям

$$E_x^0 = \varepsilon^{-1}\varphi_z H_y^0, \quad H_x^0 = -\mu^{-1}\varphi_z E_y^0, \quad (3.10)$$

где искомая вектор-функция \bar{u} содержит величины $\bar{u}(x; y, z) = (E_y^0 \ H_z^0 \ H_y^0 \ E_z^0)^T$, которые описывают распределение по x соответствующей компоненты поля в каждой точке (y, z) . Матрица A^0 определена следующим образом

$$A^0(x, y, z) = ik_0 \begin{pmatrix} 0 & \mu & 0 & 0 \\ -\frac{\varphi_z^2}{\varepsilon} + \varepsilon & 0 & 0 & 0 \\ \mu & 0 & 0 & -\varepsilon \\ 0 & 0 & \frac{\varphi_z^2}{\varepsilon} - \mu & 0 \end{pmatrix}. \quad (3.11)$$

В слое с постоянными диэлектрической и магнитной проницаемостями ε, μ решение системы дифференциальных уравнений (3.10) можно представить в следующем виде [2], который получен с помощью функции `EigenVectors` пакета `LinearAlgebra` в `Maple` [14]:

$$\begin{aligned} \bar{u}(x; y, z) = & A \begin{pmatrix} 0 \\ 0 \\ i\varepsilon\eta^0 \\ p^0 \end{pmatrix} e^{\gamma^0 x} + B \begin{pmatrix} -i\mu\eta^0 \\ p^0 \\ 0 \\ 0 \end{pmatrix} e^{\gamma^0 x} + \\ & + C \begin{pmatrix} 0 \\ 0 \\ -i\varepsilon\eta^0 \\ p^0 \end{pmatrix} e^{-\gamma^0 x} + D \begin{pmatrix} i\mu\eta^0 \\ p^0 \\ 0 \\ 0 \end{pmatrix} e^{-\gamma^0 x}, \end{aligned} \quad (3.12)$$

где $p^0 = \varphi_z^2 - \varepsilon\mu$, $\eta^0 = \sqrt{\varphi_z^2 - \varepsilon\mu}$, $\gamma^0 = k_0\eta^0$, а A, B, C, D – неопределенные константы в каждой точке (y, z) .

Замечание. Собственному значению $\gamma^0 = k_0\sqrt{\varphi_z^2 - \varepsilon\mu}$ отвечают два собственных вектора

$$v_1^+ = \begin{pmatrix} 0 \\ 0 \\ i\varepsilon\eta^0 \\ p^0 \end{pmatrix}, \quad v_2^+ = \begin{pmatrix} -i\mu\eta^0 \\ p^0 \\ 0 \\ 0 \end{pmatrix}, \quad (3.13)$$

а собственному значению $-\gamma^0$ отвечают два других собственных вектора

$$v_1^- = \begin{pmatrix} 0 \\ 0 \\ -i\varepsilon\eta^0 \\ p^0 \end{pmatrix}, \quad v_2^- = \begin{pmatrix} i\mu\eta^0 \\ p^0 \\ 0 \\ 0 \end{pmatrix}. \quad (3.14)$$

Причем, по сравнению с общим случаем

$$p^0 = \lim_{\varphi_y \rightarrow 0} p, \quad \eta^0 = \lim_{\varphi_y \rightarrow 0} \eta, \\ \gamma^0 = \lim_{\varphi_y \rightarrow 0} \gamma.$$

Главная мотивация исследования. Неудобство использования приведенных выше символьных представлений собственных векторов состоит в том, что при $\varphi_y \rightarrow 0$ собственные векторы (3.7), (3.8) не переходят в (3.13), (3.14) и становятся линейно зависимыми. С физической точки зрения φ_y может равняться нулю, причем, так как φ_y – искомая величина, заранее мы не знаем при каких y, z она обратится в ноль, и потому нам нужно универсальное представление собственных векторов. А именно такое, при котором собственные векторы в общем случае переходят в (3.13), (3.14) при $\varphi_y \rightarrow 0$ и, аналогично, при $\varphi_z \rightarrow 0$. С целью добиться такого соответствия проведем дополнительное исследование возможных символьных представлений собственных векторов, основанное на преобразованиях перестановок.

3.2. Основа алгоритма исследования – преобразование перестановок

Рассмотрим систему обыкновенных дифференциальных уравнений с постоянными коэффициентами

$$\bar{y}' - A\bar{y} = \bar{0}, \quad (3.15)$$

где $\bar{y} = (y_1 \ y_2 \ \dots \ y_N)^T$ – вектор искомых величин, A – квадратная невырожденная матрица коэффициентов. Рассмотрим эту же систему, но при другом порядке следования искомых величин $\bar{z} = P\bar{y}$, где матрица перестановок P определяет новый порядок следования искомых величин. Система (3.15) при новом порядке следования искомых величин \bar{z} преобразуется следующим образом:

$$\bar{z}' - PAP^T\bar{z} = \bar{0}. \quad (3.16)$$

При выводе (3.16) использовано свойство матрицы перестановок $P^{-1} = P^T$.

Будем рассматривать случай диагоналируемой матрицы A . В таком случае общее решение однородной системы (3.15) можно записать в следующем виде

$$\bar{y}(x) = A_1\bar{v}_1 e^{\lambda_1 x} + A_2\bar{v}_2 e^{\lambda_2 x} + \dots + A_N\bar{v}_N e^{\lambda_N x} \quad (3.17)$$

где $\{\vec{v}_j\}_{j=1}^N$ – собственные векторы матрицы A , отвечающие ее собственным значениям $\{\lambda_j\}_{j=1}^N$. Нетрудно проверить, что собственные значения матрицы PAP^T совпадают с собственными значениями матрицы A . Собственные векторы $\{\vec{w}_j\}_{j=1}^N$ матрицы PAP^T связаны с собственными векторами матрицы A следующим образом

$$\vec{w}_j = P\vec{v}_j, \quad (3.18)$$

где нумерация векторов выбрана таким образом, чтобы при фиксированном j векторы \vec{w}_j и \vec{v}_j отвечали одному и тому же собственному значению λ_j . Отсюда же следует связь между матрицами собственных векторов

$$W = PV, \quad (3.19)$$

где столбцами матриц W и V выступают векторы \vec{w}_j и \vec{v}_j соответственно.

Замечание. Существует множество различных невырожденных преобразований, фактически любая невырожденная матрица P определяет некоторое преобразование искомым величин исследуемой системы дифференциальных уравнений $\vec{z} = P\vec{u}$. В настоящей работе мы рассматриваем только преобразования перестановок.

Как показывает практика в случае, когда коэффициенты системы (3.15) заданы в символьном виде, в зависимости от преобразования порядка следования искомым величин $\vec{z} = P\vec{u}$ символьное представление собственных векторов различно. Другими словами, в зависимости от порядка следования искомым величин системы (3.15) могут получаться различные символьные представления собственных векторов.

Задача настоящего исследования состоит в том, чтобы исследовать все возможные различные символьные представления собственных векторов, получаемые при различных перестановках искомым величин, заданных различными матрицами перестановок P .

Для изучения возможных символьных представлений собственных векторов используем следующий алгоритм.

3.3. Алгоритм исследования

Ниже приведем формулировку метода, с помощью которого в рамках работы мы изучаем собственные векторы, определяющие решение системы (3.15).

1. Для изучения различных вариантов символьного представления собственных векторов сначала сгенерируем все возможные матрицы перестановок $P_j = P(s_j)$, где s_j определяет текущую перестановку из n элементов, причем $s_1 = (1 \ 2 \ \dots \ n)$ – тождественная перестановка.

Генерацию всех возможных различных перестановок $s_j, j = \overline{1, n!}$ в Maple осуществляем с помощью функций `firstperm` и `nexthperm` пакета `combinat` [14].

Для каждой перестановки s_j определяем матрицу перестановок P_j , элементы которой состоят из нулей и единиц: $P_j[k][s_{jk}] = 1$, а остальные равны нулю.

2. Далее формируем матрицу $B_j = P_jAP_j^T$ и для нее символьно определяем матрицу собственных векторов W_j .

Символьное вычисление собственных векторов осуществляем в Maple с помощью функции `Eigenvectors` пакета `LinearAlgebra` [14].

Учитывая формулу (3.19) каждая матрица собственных векторов W_j определяет матрицу собственных векторов $V_j = P_jW_j$ исходной матрицы A . Собственные значения исходной матрицы A и любой матрицы $B_j = P_jAP_j^T$ совпадают, однако символьное представление собственных векторов могут отличаться.

3. После вычисления всех матриц собственных векторов $V_j = P_jW_j$, полученных всеми возможными различными перестановками s_j , необходимо избавиться от всех линейно зависимых собственных векторов, отвечающих каждому собственному значению.

Если векторы \vec{u} и \vec{v} линейно зависимы, то $\|\alpha\vec{u} - \vec{v}\| = 0$ при $\alpha = \vec{u}^H \vec{v} / \vec{u}^H \vec{u}$ и из векторов \vec{u} и \vec{v} достаточно оставить только один любой вектор.

4. В результате получим некоторое множество различных символьных представлений собственных векторов, отвечающих каждому собственному значению. В тривиальном случае каждому собственному значению отвечает множество из единственного собственного вектора, в нетривиальном случае одному собственному значению может соответствовать несколько различных символьных представлений собственных векторов.

Применим описанный алгоритм для исследования собственных векторов, определяющих решения системы (3.9) в частном случае $\varphi_y = 0$ и системы (3.3) в общем случае.

3.4. Численный эксперимент

В качестве демонстрации применения полученных символьных результатов проведем численный эксперимент, в качестве структуры рассматриваем структуру из [2]. В рамках предположения линейной фазы из [2] $\varphi_z = \beta$, где β – коэффици-

ент фазового замедления. Вычислим коэффициенты фазового замедления и соответствующие им направляемые волноводные моды для волноводной структуры из [2].

Структура представляет собой трехслойный волновод с двумя полубесконечными слоями – подложкой с $\epsilon_s = n_s^2$, занимающей область $x < 0$ и проводящим слоем с $\epsilon_c = n_c^2$, занимающей область $x > h$. Между двумя полубесконечными слоями $0 < x < h$ расположен волноводный слой с $\epsilon_f = n_f^2$. На границах между слоями $x = 0$ и $x = h$ должны выполняться условия непрерывности компонент E_y, H_z, H_y, E_z . Поля направляемых мод характеризуются также убыванием при бесконечном удалении от волноводного слоя, то есть для направляемых мод должны выполняться асимптотические условия вида $\|\vec{E}\| \xrightarrow{x \rightarrow \pm\infty} 0, \|\vec{H}\| \xrightarrow{x \rightarrow \pm\infty} 0$.

Зная символьное представление общего решения системы (3.3) в каждом слое (причем в полубесконечных слоях общее решение должно удовлетворять асимптотическим условиям) можно записать условия непрерывности компонент поля. В итоге получится однородная система линейных алгебраических уравнений – система граничных уравнений. Из условия разрешимости системы граничных уравнений определяется фаза, а решение системы граничных уравнений есть набор коэффициентов разложения решения в каждом слое по ФСР. Численные параметры расчета следующие:

$$\begin{aligned} \lambda &= 0.55; & k_0 &= 2\pi/\lambda; & h &= 2\lambda \\ n_c &= 1.0; & n_f &= 1.565; & n_s &= 1.47; \\ \mu_c &= \mu_f = \mu_s = 1. \end{aligned} \quad (3.20)$$

Проведем численный эксперимент: для рассматриваемой структуры вычислим допустимые коэффициенты фазового замедления и поля соответствующих волноводных мод используя три различных представления общего решения в слое волновода:

1. первое представление соответствует случаю $\varphi_y = 0$;
2. второе представление соответствует случаю $\varphi_y = \delta \neq 0$, собственные векторы для которого выберем с помощью описанного в разделе 3 символьного алгоритма;
3. третье представление соответствует случаю $\varphi_y = \delta \neq 0$, собственные векторы для которого определены с помощью встроенной функции `Eigenvectors` (без использования описанного алгоритма).

Мы ожидаем, что при стремлении δ к нулю коэффициенты фазового замедления и поля соответствующих волноводных мод, полученные на

основе собственных векторов с помощью предложенного алгоритма (второе представление), сходятся к аналогичным результатам для случая $\varphi_y = 0$ (первое представление). Исследуем также сходимость коэффициентов фазового замедления и полей соответствующих волноводных мод, построенных на основе собственных векторов, определенных с помощью встроенной функции `Eigenvectors` (третье представление) к аналогичным результатам для случая $\varphi_y = 0$ (первое представление).

Поля волноводных мод для каждого из трех представлений задаются в каждой подобласти в виде отдельного символьного выражения:

$$\vec{u}_m(x; y, z) = \begin{cases} \vec{u}_m^c(x; y, z), & x > h \\ \vec{u}_m^f(x; y, z), & 0 < x < h \\ \vec{u}_m^s(x; y, z), & x < 0 \end{cases} \quad (3.21)$$

где $m = 1, 2, 3$. Для каждого из трех представлений система граничных уравнений формулируется из условий сопряжения при $x = 0$ и $x = h$:

$$\vec{u}_m^c(h; y, z) - \vec{u}_m^f(h; y, z) = \vec{0}, \quad (3.22)$$

$$\vec{u}_m^f(0; y, z) - \vec{u}_m^s(0; y, z) = \vec{0}, \quad (3.23)$$

где $m = 1, 2, 3$. В результате получаем для каждого $m = 1, 2, 3$ однородную систему линейных алгебраических уравнений относительно искомого величин A_j^α для $m = 1$, B_j^α для $m = 2$ и D_j^α для $m = 3$. Коэффициенты каждой системы зависят от $\beta = \varphi_z$ при фиксированном $\varphi_y = \delta$. Условие разрешимости однородной системы линейных алгебраических уравнений – это равенство нулю определителя матрицы коэффициентов системы $\det M_m(\beta) = 0$, $m = 1, 2, 3$.

Отыскиваем численно такие значения $\beta_{s,m}$ (индекс s определяет номер направляемой моды), при которых $\det M_m(\beta) = 0$ при $m = 1, 2, 3$, а также для каждого из таких значений отыщем набор искомого величин, которые определяют поле соответствующей волноводной моды: A_j^α для $m = 1$, B_j^α для $m = 2$ и D_j^α для $m = 3$.

Для поиска нулей воспользуемся классическим методом деления отрезка пополам [15], для поиска решений однородных систем уравнений воспользуемся командой `Eigenvectors` и выберем тот собственный вектор, который соответствует минимальному по модулю собственному значению. Для почти вырожденной матрицы минимальное по модулю собственное значение близко к нулю, поэтому собственный вектор, отвечающий минимальному по модулю собственному значению, будет приближенным решением соответствующей однородной системы.

Определив искомые величины A_j^α для $m = 1$, B_j^α для $m = 2$ и D_j^α для $m = 3$, можно построить поля волноводных мод.

Замечание. Поля волноводных мод определяются с помощью посчитанных численно компонент собственного вектора, отвечающего почти нулевому собственному значению матрицы. Собственный вектор известен с точностью до мультипликативной константы, поэтому и соответствующие поля также известны с точностью до мультипликативной константы. Поэтому, для корректного сравнения полей необходимо их нормировать в каком-либо функциональном пространстве. Учитывая только непрерывность компонент поля на границах раздела и их убывание на бесконечности, каждую компоненту поля можно рассматривать как элемент пространства $L_2(\mathbb{R})$ на всей вещественной оси. Поэтому рассматриваемые нами вектор-функции $\vec{u}_m(x; y, z)$, $m = 1, 2, 3$, состоящие из четырех компонент поля, могут рассматриваться как элементы пространства $H = \{L_2(\mathbb{R})\}^4$, состоящего из таких вектор-функций \vec{u} , для которых справедливо $u_j \in L_2(\mathbb{R})$, $j = 1..4$. Скалярное произведение в пространстве H определено как

$$(\vec{u}, \vec{v}) = \int_{-\infty}^{+\infty} \left\{ \sum_{j=1}^4 u_j(x) \cdot \overline{v_j(x)} \right\} dx, \quad (3.24)$$

где черта сверху обозначает комплексное сопряжение. Норма элемента \vec{u} определена следующим образом $\|\vec{u}\| = \sqrt{(\vec{u}, \vec{u})}$.

Вычисленные для каждого $\beta_{s,m}$ вектор-функции $\vec{u}_{s,m}(x; y, z)$

- во-первых нормируем, чтобы норма каждой функции была единичной, то есть $\vec{u}_{s,m}^{norm} = \vec{u}_{s,m} / \|\vec{u}_{s,m}\|$ для $m = 1, 2, 3$;

- во-вторых подбираем такие комплексные числа $\tau_{s,2}$ и $\tau_{s,3}$, что $|\tau_{s,2}| = 1$, $|\tau_{s,3}| = 1$, чтобы $\|\vec{u}_{s,1}^{norm} - \tau_{s,2} \cdot \vec{u}_{s,2}^{norm}\|$ и $\|\vec{u}_{s,1}^{norm} - \tau_{s,3} \cdot \vec{u}_{s,3}^{norm}\|$ были минимальны (для этого воспользуемся алгоритмом, представленным ниже в разделе 3.5).

В рамках численных экспериментов оценим для различных значений δ и для различных направляемых мод (индекс s определяет номер направляемой моды) следующие величины

$$\Delta_2 \beta_s(\delta) = \left| \frac{\beta_{s,1} - \beta_{s,2}}{\beta_{s,1}} \right|, \quad (3.25)$$

$$\Delta_3 \beta_s(\delta) = \left| \frac{\beta_{s,1} - \beta_{s,3}}{\beta_{s,1}} \right|, \quad (3.26)$$

характеризующие близость вычисленных коэффициентов фазового замедления при $\varphi_y \neq 0$ с использованием приведенного алгоритма ($m = 2$) и без его использования ($m = 3$) к соответствующим коэффициентам фазового замедления при $\varphi_y = 0$ ($m = 1$).

Также оценим для различных значений δ и для различных направляемых мод (индекс s определяет номер направляемой моды) следующие величины

$$\Delta_2 u_s(\delta) = \left\| \vec{u}_{s,1}^{norm} - \tau_{s,2} \cdot \vec{u}_{s,2}^{norm} \right\|, \quad (3.27)$$

$$\Delta_3 u_s(\delta) = \left\| \vec{u}_{s,1}^{norm} - \tau_{s,3} \cdot \vec{u}_{s,3}^{norm} \right\|, \quad (3.28)$$

характеризующие близость вычисленных (нормированных) полей направляемых волноводных мод при $\varphi_y \neq 0$ с использованием приведенного алгоритма ($m = 2$) и без его использования ($m = 3$) к соответствующим (нормированным) полям направляемых мод при $\varphi_y = 0$ ($m = 1$).

3.5. Алгоритм подбора констант для сравнения полей

Рассмотрим абстрактное гильбертово пространство H [16, 17]. Пусть имеется два элемента $u, v \in H$ такие, что $\|u\| = \|v\| = 1$. Ищем такую комплексную константу $\alpha \in \mathbb{C}$, $|\alpha| = 1$, чтобы элементы $v_\alpha = \alpha v$ и u отличались минимально друг от друга в метрике рассматриваемого пространства H . Другими словами ищем такую комплексную константу $\alpha \in \mathbb{C}$, $|\alpha| = 1$ (при таком выборе норма элемента αv остается единичной, так как $\|\alpha v\| = |\alpha| \|v\| \equiv \|v\|$) чтобы $\|\alpha v - u\|^2 \rightarrow \min$. Для простоты рассмотрения введем параметр $t \in [0, 2\pi]$: $\alpha = e^{it}$ и отыщем минимум неотрицательной функции

$$f(t) = \|e^{it} v - u\|^2 \geq 0 \quad (3.29)$$

на отрезке $t \in [0, 2\pi]$.

Правая часть $f(t)$ в развернутом виде имеет следующий вид $f(t) = (e^{it} v - u, e^{it} v - u)$:

$$f(t) = \|u\|^2 + \|v\|^2 - e^{it}(v, u) - e^{-it}(u, v). \quad (3.30)$$

Для поиска точек экстремума функции $f(t)$ пользуемся необходимым условием экстремума [18]: находим ее производную

$$f'(t) = i(e^{-it}(u, v) - e^{it}(v, u)), \quad (3.31)$$

и решаем уравнение $f'(t) = 0$. Уравнение $f'(t) = 0$ тождественно выполнено, если $e^{2it} = (u, v) / (v, u)$. Обозначим $c = (u, v)$, поэтому $\bar{c} = (v, u)$ и с учетом этих обозначений:

$$\alpha_{\pm} = e^{it_{\pm}} \equiv \pm \frac{c}{|c|}. \quad (3.32)$$

Замечание. В случае, когда элементы $u, v \in H$ ортогональны друг другу, $c = 0$. В таком случае задача не имеет смысла, так как $f(t)$ не имеет ни минимума, ни максимума, потому что $f(t) \equiv \text{const}$ при $t \in [0, 2\pi]$. Поэтому рассматриваем случай $c \neq 0$.

Одно из двух значений $\alpha_{\pm} = e^{it_{\pm}}$ обеспечивает минимум функции $f(t)$, а второе – максимум на отрезке $t \in [0, 2\pi]$. Для определения искомого значения константы α воспользуемся достаточным условием минимума [18]: находим вторую производную функции $f(t)$

$$f''(t) = e^{-it}(u, v) + e^{it}(v, u) \quad (3.33)$$

и проверяем в какой из точек t_{\pm} значение второй производной положительно.

Непосредственной подстановкой проверяется, что $f''(t_{\pm}) = 2|c| > 0$ при $c \neq 0$. Поэтому искомое значение константы α определено следующим образом

$$\alpha = e^{it_{\pm}} \equiv \frac{c}{|c|}. \quad (3.34)$$

Замечание. Для решения описанной задачи требуется вычислить только одно скалярное произведение для расчета константы c .

4. РЕЗУЛЬТАТЫ

Применим алгоритм, описанный в предыдущем разделе, сначала для исследования собственных векторов системы (3.9), описывающей важный частный случай $\varphi_y = 0$.

4.1. Символьные результаты для частного случая

Исследуя этот частный случай при любых возможных преобразованиях перестановок, мы получаем тривиальный случай – каждому собственному значению $\lambda_1^0 = \lambda_2^0 = \gamma^0$, $\lambda_3^0 = \lambda_4^0 = -\gamma^0$ соответствует единственное символьное представление собственного вектора, а именно:

$$\begin{aligned} \vec{v}_1^0 &= \begin{pmatrix} -i\mu \\ \sqrt{\varphi_z^2 - \varepsilon\mu} \\ 0 \\ 0 \end{pmatrix}, & \vec{v}_2^0 &= \begin{pmatrix} 0 \\ 0 \\ i\varepsilon \\ \sqrt{\varphi_z^2 - \varepsilon\mu} \end{pmatrix}, \\ \vec{v}_3^0 &= \begin{pmatrix} i\mu \\ \sqrt{\varphi_z^2 - \varepsilon\mu} \\ 0 \\ 0 \end{pmatrix}, & \vec{v}_4^0 &= \begin{pmatrix} 0 \\ 0 \\ -i\varepsilon \\ \sqrt{\varphi_z^2 - \varepsilon\mu} \end{pmatrix}, \end{aligned} \quad (4.1)$$

где $\gamma^0 = k_0\sqrt{\varphi_z^2 - \varepsilon\mu}$.

Замечание. В действительности мы получили по два ортогональных собственных вектора, отвечающих каждому из двух двукратных собственных значений $\pm\gamma^0$. Однако, если за первое и второе собственные значения принять $\lambda_1^0 = \lambda_2^0 = \gamma^0$ и условиться за первый собственный вектор принять тот из собственных векторов, у которого первые две компоненты отличны от нуля, а за второй – тот, у которого две последних компоненты отличны от нуля, то в таком случае представление собственных векторов, отвечающих первым двум собственным значениям $\lambda_1^0 = \lambda_2^0 = \gamma^0$, единственно. Также единственны представления третьего и четвертого собственных векторов, принимая аналогичную условность для них, отвечающих третьему и четвертому собственным значениям $-\gamma^0$.

4.2. Символьные результаты для общего случая

В общем случае ($\varphi_y \neq 0$, $\varphi_z \neq 0$) мы получаем нетривиальный случай – каждому собственному значению $\lambda_1 = \lambda_2 = \gamma$, $\lambda_3 = \lambda_4 = -\gamma$ соответствует по два различных представления каждого собственного вектора, а именно:

$$\vec{v}_{1,1} = \begin{pmatrix} -i\mu\sqrt{\varphi_y^2 + \varphi_z^2 - \varepsilon\mu} \\ \varphi_z^2 - \varepsilon\mu \\ \varphi_y\varphi_z \\ 0 \end{pmatrix}, \quad (4.2)$$

$$\vec{v}_{1,2} = \begin{pmatrix} \varphi_y^2 - \varepsilon\mu \\ -i\varepsilon\sqrt{\varphi_y^2 + \varphi_z^2 - \varepsilon\mu} \\ 0 \\ \varphi_y\varphi_z \end{pmatrix},$$

$$\vec{v}_{2,1} = \begin{pmatrix} \varphi_y\varphi_z \\ 0 \\ i\varepsilon\sqrt{\varphi_y^2 + \varphi_z^2 - \varepsilon\mu} \\ \varphi_z^2 - \varepsilon\mu \end{pmatrix}, \quad (4.3)$$

$$\vec{v}_{2,2} = \begin{pmatrix} 0 \\ \varphi_y\varphi_z \\ \varphi_y^2 - \varepsilon\mu \\ i\mu\sqrt{\varphi_y^2 + \varphi_z^2 - \varepsilon\mu} \end{pmatrix},$$

$$\vec{v}_{3,1} = \begin{pmatrix} i\mu\sqrt{\varphi_y^2 + \varphi_z^2 - \varepsilon\mu} \\ \varphi_z^2 - \varepsilon\mu \\ \varphi_y\varphi_z \\ 0 \end{pmatrix}, \quad (4.4)$$

$$\vec{v}_{3,2} = \begin{pmatrix} \varphi_y^2 - \varepsilon\mu \\ i\varepsilon\sqrt{\varphi_y^2 + \varphi_z^2 - \varepsilon\mu} \\ 0 \\ \varphi_y\varphi_z \end{pmatrix},$$

$$\vec{v}_{4,1} = \begin{pmatrix} \varphi_y\varphi_z \\ 0 \\ -i\varepsilon\sqrt{\varphi_y^2 + \varphi_z^2 - \varepsilon\mu} \\ \varphi_z^2 - \varepsilon\mu \end{pmatrix}, \quad (4.5)$$

$$\vec{v}_{4,2} = \begin{pmatrix} 0 \\ \varphi_y\varphi_z \\ \varphi_y^2 - \varepsilon\mu \\ -i\mu\sqrt{\varphi_y^2 + \varphi_z^2 - \varepsilon\mu} \end{pmatrix},$$

где $\gamma = k_0\sqrt{\varphi_y^2 + \varphi_z^2 - \varepsilon\mu}$.

Замечание 1. Нумерация собственных значений выбрана по аналогии с частным случаем, причем $\lim_{\varphi_y \rightarrow 0} \lambda_j = \lambda_j^0$, $j = \overline{1,4}$. Нумерация собственных векторов выбрана исходя из условия $\lim_{\varphi_y \rightarrow 0} \vec{v}_{j,1} = \alpha_{j,1}\vec{v}_j^0$,

$\lim_{\varphi_y \rightarrow 0} \vec{v}_{j,2} = \alpha_{j,2}\vec{v}_j^0$, $j = \overline{1,4}$. Другими словами, нумерация выбрана таким образом, чтобы каждый собственный вектор в общем случае переходил при $\varphi_y \rightarrow 0$ в соответствующий собственный вектор при $\varphi_y = 0$.

Замечание 2. Фактически любая линейная комбинация $\vec{v}_j = A\vec{v}_{j,1} + B\vec{v}_{j,2}$ — собственный вектор, отвечающий собственному значению λ_j .

Замечание 3. Используя какие-либо другие преобразования, например, преобразования поворота, возможно получится большее число различных символьных представлений собственных векторов.

На основе полученных символьных выражений запишем для каждого из трех представлений вид решения в каждом из слоев.

Первое представление $\vec{u}_1(x; y, z)$ характеризуется следующим видом решения для каждого из слоев:

$$\vec{u}_1^c(x; y, z) = (A_3^c \cdot \vec{v}_{3,c}^1 + A_4^c \cdot \vec{v}_{4,c}^1)e^{-\gamma_c^1(x-h)}, \quad (4.6)$$

$$\vec{u}_1^f(x; y, z) = (A_1^f \cdot \vec{v}_{1,f}^1 + A_2^f \cdot \vec{v}_{2,f}^1)e^{\gamma_f^1 x} + (A_3^f \cdot \vec{v}_{3,f}^1 + A_4^f \cdot \vec{v}_{4,f}^1)e^{-\gamma_f^1 x}, \quad (4.7)$$

$$\vec{u}_1^s(x; y, z) = (A_1^s \cdot \vec{v}_{1,s}^1 + A_2^s \cdot \vec{v}_{2,s}^1)e^{\gamma_s^1 x}, \quad (4.8)$$

где $\vec{v}_{j,\alpha}^1 = \vec{v}_j^1|_{\varepsilon=\varepsilon_\alpha, \mu=\mu_\alpha}$, $j = \overline{1,4}$ и $\gamma_\alpha^1 = \gamma^0|_{\varepsilon=\varepsilon_\alpha, \mu=\mu_\alpha}$ при $\alpha = c, f, s$, причем \vec{v}_j^0 определены согласно формулам (4.1).

Второе представление $\vec{u}_2(x; y, z)$ характеризуется следующим видом решения для каждого из слоев:

$$\vec{u}_2^c(x; y, z) = (B_3^c \cdot \vec{v}_{3,c}^2 + B_4^c \cdot \vec{v}_{4,c}^2)e^{-\gamma_c^2(x-h)}, \quad (4.9)$$

$$\vec{u}_2^f(x; y, z) = (B_1^f \cdot \vec{v}_{1,f}^2 + B_2^f \cdot \vec{v}_{2,f}^2)e^{\gamma_f^2 x} + (B_3^f \cdot \vec{v}_{3,f}^2 + B_4^f \cdot \vec{v}_{4,f}^2)e^{-\gamma_f^2 x}, \quad (4.10)$$

$$\vec{u}_2^s(x; y, z) = (B_1^s \cdot \vec{v}_{1,s}^2 + B_2^s \cdot \vec{v}_{2,s}^2)e^{\gamma_s^2 x}, \quad (4.11)$$

где $\vec{v}_{j,\alpha}^2 = \vec{v}_j^2|_{\varepsilon=\varepsilon_\alpha, \mu=\mu_\alpha}$, $j = \overline{1,4}$ и $\gamma_\alpha^2 = \gamma^2|_{\varepsilon=\varepsilon_\alpha, \mu=\mu_\alpha}$ при $\alpha = c, f, s$, причем \vec{v}_j^2 определены согласно формулам (4.2)–(4.5).

Третье представление $\vec{u}_3(x; y, z)$ характеризуется следующим видом решения для каждого из слоев:

$$\vec{u}_3^c(x; y, z) = (D_3^c \cdot \vec{v}_{3,c}^3 + D_4^c \cdot \vec{v}_{4,c}^3)e^{-\gamma_c^3(x-h)}, \quad (4.12)$$

$$\vec{u}_3^f(x; y, z) = (D_1^f \cdot \vec{v}_{1,f}^3 + D_2^f \cdot \vec{v}_{2,f}^3)e^{\gamma_f^3 x} + (D_3^f \cdot \vec{v}_{3,f}^3 + D_4^f \cdot \vec{v}_{4,f}^3)e^{-\gamma_f^3 x}, \quad (4.13)$$

$$\vec{u}_3^s(x; y, z) = (D_1^s \cdot \vec{v}_{1,s}^3 + D_2^s \cdot \vec{v}_{2,s}^3)e^{\gamma_s^3 x}, \quad (4.14)$$

где $\vec{v}_{1,\alpha}^3 = \vec{v}_1^+|_{\varepsilon=\varepsilon_\alpha, \mu=\mu_\alpha}$, $\vec{v}_{2,\alpha}^3 = \vec{v}_2^+|_{\varepsilon=\varepsilon_\alpha, \mu=\mu_\alpha}$, $\vec{v}_{3,\alpha}^3 = \vec{v}_1^-|_{\varepsilon=\varepsilon_\alpha, \mu=\mu_\alpha}$, $\vec{v}_{4,\alpha}^3 = \vec{v}_2^-|_{\varepsilon=\varepsilon_\alpha, \mu=\mu_\alpha}$, и $\gamma_\alpha^3 = \gamma^3|_{\varepsilon=\varepsilon_\alpha, \mu=\mu_\alpha}$ при $\alpha = c, f, s$, причем $\vec{v}_{1,2}^\pm$ определены согласно формулам (3.7), (3.8).

Для каждого из трех представлений полей согласно разделу 3.4 формируется система граничных уравнений (3.22), (3.23), после чего вычисляются коэффициенты фазового замедления и коэффициенты, определяющие поля волноводных мод. Полученные численно величины коэффициентов фазового замедления и полей соответствующих волноводных мод анализируются в разделе ниже.

4.3. Численные результаты

Ниже приведем вычисленные с точностью 10^{-14} коэффициенты фазового замедления для первого случая $\beta_{s,1}$, где $s = \overline{1,4}$:

$$\begin{aligned}\beta_{1,1} &= 1.55149273806928903, \\ \beta_{2,1} &= 1.55018111589009942, \\ \beta_{3,1} &= 1.51175061453743748, \\ \beta_{4,1} &= 1.50727495127641732.\end{aligned}\quad (4.15)$$

Ниже приведены графики величин $\Delta_{2,3}\beta_s(\delta)$, определенных формулами (3.25), (3.26) для первой моды ($s = 1$).

Для старших мод $s = 2, 3, 4$ поведение величин $\Delta_{2,3}\beta_s(\delta)$ полностью идентично их поведению при $s = 1$, поэтому ограничимся только одним графиком.

Далее приведем графики величин $\Delta_{2,3}u_s(\delta)$, определенных формулами (3.27), (3.28) для четырех направляемых мод $s = \overline{1,4}$.

5. ОБСУЖДЕНИЕ

5.1. Символьные результаты

Задача отыскания собственных векторов является одной из основных задач алгебры. Собственные значения и собственные векторы часто представляют интерес отдельного исследования, часто также используются при решении системы обыкновенных дифференциальных уравнений, если собственные значения и собственные векторы известны.

В рамках настоящей работы исследовались собственные векторы, определяющие решение системы дифференциальных уравнений для амплитудных частей электромагнитного поля в качестве искомого величин. Особенность исследуемой системы состоит в том, что коэффициенты системы заданы в символьном виде и отыскивать собственные векторы и собственные значения также необходимо в символьном виде. Некоторые символьные константы, входящие в коэффициенты системы дифференциальных уравнений, могут обращаться в ноль и потому для построения универсального вида решения важно иметь набор линейно независимых собственных векторов, переходящий также в линейно независимый набор собственных векторов при обнулении соответствующих констант.

Благодаря предложенному алгоритму символьного отыскания различных собственных векторов, удалось исследовать множество различных символьных представлений собственных векторов, отвечающих каждому собственному значению, и

выделить для каждого собственного значения подмножество различных символьных представлений собственных векторов. Кроме того, удалось также сгруппировать собственные векторы таким образом, чтобы обеспечить корректный предельный переход собственных векторов при стремлении к нулю символьных констант, которые могут обращаться в ноль.

Как показывают символьные расчеты, каждому собственному значению матрицы коэффициентов отвечают минимум два различных представления собственного вектора, что дает исследователю больший выбор представления общего решения системы дифференциальных уравнений и использовать в каждом конкретном случае наиболее удобную комбинацию из этих двух или более собственных векторов.

Важно отметить, что стандартный функционал системы компьютерной алгебры Maple при символьном отыскании собственных векторов выдает некоторый конкретный набор собственных векторов, который не обязательно обеспечивает корректный предельный переход при обнулении символьных констант, и потому для дальнейших численных расчетов такой набор собственных векторов может не подходить.

Настоящее исследование не претендует на исчерпывающее множество всех различных символьных представлений собственных векторов, а скорее дает минимальное множество, которое получается при преобразованиях перестановок. Это минимальное множество тем не менее позволило получить собственные векторы, которые обеспечивают корректный предельный переход при обнулении символьных констант модели. Однако существует значительно больше преобразований — например, преобразования поворота — которые могут дать большее количество различных символьных представлений собственных векторов и дать исследователю еще больше вариантов различных символьных представлений собственных векторов для дальнейшего численного и качественного анализа.

5.2. Численные результаты

Замечание. Все приведенные выше графики необходимо читать в направлении убывания δ — то есть справа налево.

В рамках численных расчетов демонстрируется практическая польза от разработанного символьного алгоритма (см. раздел 3). В рамках расчетов показано, что при любом выборе собственных векторов — как с использованием предложенного алгоритма, так и с использованием встроенной функции `Eigenvalues` — коэффициенты фазового замедления

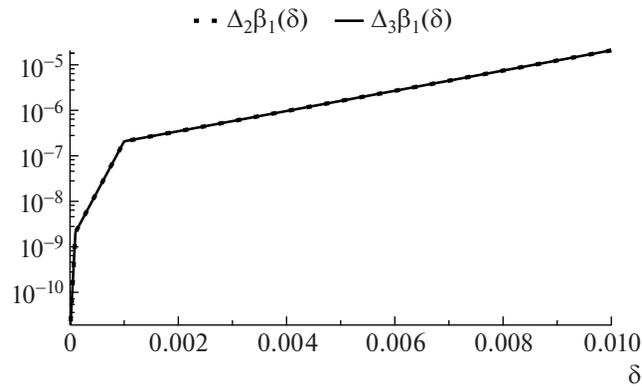


Рис. 1. Графики величин $\Delta_{2,3}\beta_s(\delta)$, определенных формулами (3.25), (3.26) для первой моды ($s = 1$).

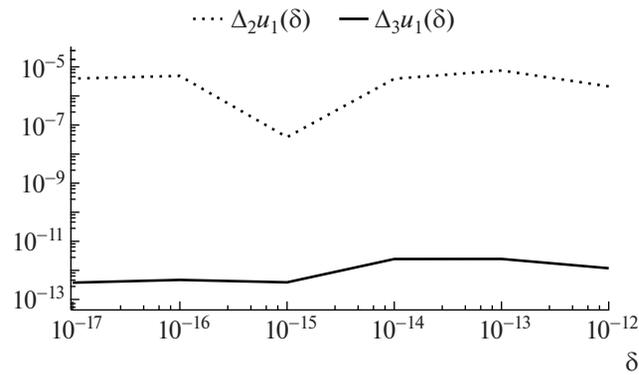


Рис. 2. Графики величин $\Delta_{2,3}u_s(\delta)$, определенных формулами (3.27), (3.28) для первой моды ($s = 1$).

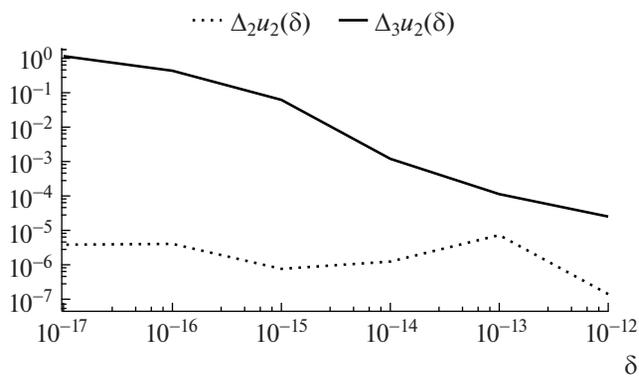


Рис. 3. Графики величин $\Delta_{2,3}u_s(\delta)$, определенных формулами (3.27), (3.28) для второй моды ($s = 2$).

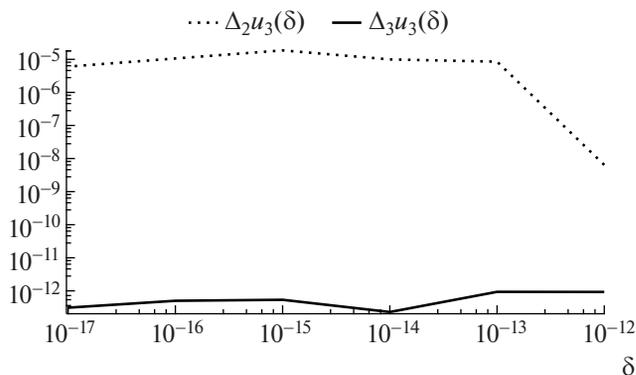


Рис. 4. Графики величин $\Delta_{2,3}u_s(\delta)$, определенных формулами (3.27), (3.28) для третьей моды ($s = 3$).

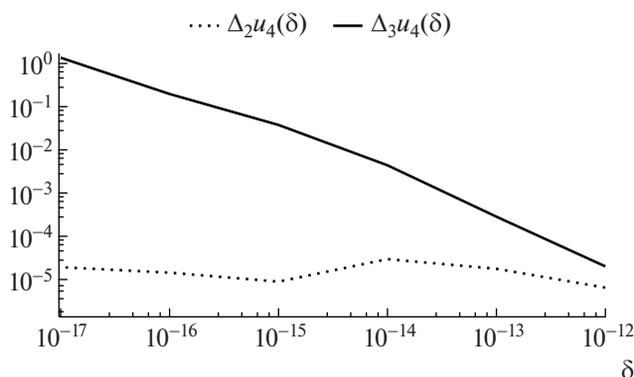


Рис. 5. Графики величин $\Delta_{2,3}u_s(\delta)$, определенных формулами (3.27), (3.28) для четвертой моды ($s = 4$).

сходятся при уменьшении δ к значениям при $\delta = 0$ (см. рис. 1).

Практическая польза от применения алгоритма видна при рассмотрении графиков сходимостей полей направляемых волноводных мод – см. рис. 2–5. Как видно из рисунка 2, погрешность вычисленных полей первой волноводной моды без использования предложенного алгоритма $\Delta_3 u_1(\delta)$ на несколько порядков меньше, чем при использовании предложенного алгоритма $\Delta_2 u_1(\delta)$, эти величины имеют порядок 10^{-5} и 10^{-12} соответственно.

Рисунок 3 демонстрирует эффективность использования предложенного алгоритма: погрешность вычисленного поля второй волноводной моды без использования предложенного алгоритма $\Delta_3 u_2(\delta)$ растет с уменьшением параметра δ и достигает значения порядка 1.1 при $\delta = 10^{-17}$, причем сравниваемые поля имеют единичную норму (то есть относительная погрешность более 100%). При этом погрешность вычисленного поля второй волноводной моды с использованием пред-

ложенного алгоритма $\Delta_2 u_2(\delta)$ имеет порядок 10^{-5} при $\delta = 10^{-17}$.

Рисунок 4 показывает, что обе величины $\Delta_3 u_3(\delta)$ и $\Delta_2 u_3(\delta)$ достаточно малы для численных расчетов и имеют порядок 10^{-12} и 10^{-5} соответственно.

Рисунок 5 демонстрирует эффективность использования предложенного алгоритма: величина $\Delta_3 u_4(\delta)$ растет с уменьшением параметра δ и достигает значения порядка 1.3, при том что сравниваемые поля имеют единичную норму (то есть относительная погрешность более 100%) при $\delta = 10^{-17}$. При этом относительная погрешность вычисленного поля четвертой волноводной моды с использованием предложенного алгоритма $\Delta_2 u_4(\delta)$ имеет порядок 10^{-5} .

Другими словами, использование собственных векторов, полученных в символьном виде без использования предложенного в разделе 3 алгоритма, позволяет корректно вычислять коэффициенты фазового замедления направляемых вол-

новодных мод при малых δ . Однако поля соответствующих волноводных мод могут вычисляться с недопустимо большими погрешностями, достигающими порядка 100% и более, и потому предложенный алгоритм необходим для проведения корректных практических расчетов.

Расчеты с использованием разработанного символьного алгоритма показывают невысокую точность: относительная погрешность вычисленных полей направляемых волноводных мод составляет 10^{-5} . Однако, эта величина относительной погрешности сохраняется для всех вычисленных полей направляемых волноводных мод, в то время как вычисления без использования разработанного алгоритма могут характеризоваться относительными погрешностями более единицы.

Отдельного внимания также заслуживает вопрос неравномерной сходимости вычисленных полей. По мнению авторов численные артефакты, которые наблюдаются на рис. 2–5, связаны с тем, что при расчетах были использованы символьные представления собственных векторов без дополнительной численной нормировки. Однако вопрос разработки устойчивого численного алгоритма вычисления полей выходит за рамки настоящего исследования и будет отдельно рассмотрен в рамках будущих исследований.

6. ЗАКЛЮЧЕНИЕ

В работе построен символьный алгоритм отыскания различных символьных представлений собственных векторов символьно заданных нормальных матриц, основанный на преобразованиях перестановок. С помощью представленного алгоритма можно строить различные символьные представления общего решения системы обыкновенных дифференциальных уравнений с постоянными коэффициентами в случае, когда матрица коэффициентов задана в символьном виде, что позволит исследователю использовать наиболее универсальное представление (либо наиболее компактное, либо универсальное с точки зрения существования при различных численных значениях символьных констант) для дальнейших численных расчетов.

Численные расчеты с применением предложенного алгоритма демонстрируют его эффективность на фоне аналогичных результатов, полученных без его использования. На примере задачи вычисления полей направляемых волноводных мод планарного трехслойного волновода показано, что поля волноводных мод, вычисленные без применения предложенного алгоритма, могут вычисляться с недопустимо большими погрешностями, достигающими порядка 100% и более.

Приведенный алгоритм расширяет функционал системы компьютерной алгебры и позволяет рассматривать различные символьные представления собственных векторов, а не одно фиксированное.

7. БЛАГОДАРНОСТИ

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

СПИСОК ЛИТЕРАТУРЫ

1. *Sevastyanov L.A., Sevastyanov A.L., Tyutyunnik A.A.* Analytical Calculations in Maple to Implement the Method of Adiabatic Modes for Modelling Smoothly Irregular Integrated Optical Waveguide Structures // Lecture Notes in Computer Science. 2014. V. 8660. P. 419–431. https://doi.org/10.1007/978-3-319-10515-4_30
2. *Divakov D.V., Sevastianov A.L.* The Implementation of the Symbolic-Numerical Method for Finding the Adiabatic Waveguide Modes of Integrated Optical Waveguides in CAS Maple. // Lecture Notes in Computer Science. 2019. V. 11661. P. 107–121. https://doi.org/10.1007/978-3-030-26831-2_8
3. *Yee K.* Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media // IEEE Transactions on Antennas and Propagation. 1966. V. 14 (3.1). P. 302–307. <https://doi.org/10.1109/TAP.1966.1138693>
4. *Taflove A.* Application of the Finite-Difference Time-Domain Method to Sinusoidal Steady-State Electromagnetic-Penetration Problems // IEEE Transactions on Electromagnetic Compatibility. 1980. V. EMC-22 (3.1). P. 191–202. <https://doi.org/10.1109/TEMC.1980.30387>
5. *Joseph R., Goorjian P., Taflove A.* Direct time integration of Maxwell's equations in two-dimensional dielectric waveguides for propagation and scattering of femtosecond electromagnetic solitons // Optics Letters. 1993. V. 18 (3.5). P. 491–493. <https://doi.org/10.1364/OL.18.000491>
6. *Sveshnikov A.G.* The incomplete Galerkin method. Dokl. Akad. Nauk SSSR. 1977. V. 236 (3.3). P. 1076–1079.
7. *Petukhov A.A.* Joint application of the incomplete Galerkin method and scattering matrix method for modeling multilayer diffraction gratings. Mathematical Models and Computer Simulations. 2014. V. 6. P. 92–100. <https://doi.org/10.1134/S2070048214010128>
8. *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 11661, 412–429 (2019). https://doi.org/10.1007/978-3-030-26831-2_27
9. *Kantorovich L.V., Krylov V.I.* Approximate Methods of Higher Analysis. Wiley, New York, 1964.

10. *Gusev A.A., Chuluunbaatar O., Vinitky S.I., Derbov V.L.* Solution of the Boundary-Value Problem for a Systems of ODEs of Large Dimension: Benchmark Calculations in the Framework of Kantorovich Method // *Discrete and Continuous Models and Applied Computational Science*. 2016. № 3. P. 31–37.
11. *Bathe K.J.* Finite Element Procedures in Engineering Analysis. Prentice Hall, Englewood Cliffs. 1982.
12. *Bogolyubov A.N., Mukhartova Yu.V., Gao J., Bogolyubov N.A.* Mathematical modeling of plane chiral waveguide using mixed finite elements // *Progress in Electromagnetics Research Symposium*. 2012. 1216–1219.
13. *Babich V.M., Buldyrev V.S.* Asymptotic Methods in Short-Wave Diffraction Problems. Nauka, Moscow, 1972. [English translation: Springer Series on Wave Phenomena 4. Springer, Berlin Heidelberg New York, 1991.]
14. Mathematics-based software and services for education, engineering, and research <https://www.maple-soft.com/>
15. *Hamming R.W.* Numerical Methods for Scientists and Engineers. Dover Publications; 2nd Revised ed. edition. 1987.
16. *Reed M., Simon B.* Methods of Modern Mathematical Physics. I: Functional Analysis (v. 1). Academic Press; 1st edition. 1972.
17. *Lang S.* Real and Functional Analysis. Springer-Verlag New York; 3rd edition, 1993.
18. *Hardy G.H.* A course of pure mathematics. Cambridge, At the University Press; 3rd edition, 1921.

ИСПОЛЬЗОВАНИЕ ШАБЛОНИЗАТОРА КАК ИНСТРУМЕНТАРИЯ
КОМПЬЮТЕРНОЙ АЛГЕБРЫ© 2021 г. М. Н. Геворкян^{а,*}, А. В. Королькова^{а,**}, Д. С. Кулябов^{а,б,***}^а Кафедра прикладной информатики и теории вероятностей, Российский университет дружбы народов,
ул. Миклухо-Маклая, д. 6, Москва, Россия, 117198^б Лаборатория информационных технологий, Объединенный институт ядерных исследований,
ул. Жолио-Кюри 6, Дубна, Московская область, Россия, 141980

*E-mail: gevorgyan-mn@rudn.ru

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

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

Поступила в редакцию 15.07.2020 г.

После доработки 07.08.2020 г.

Принята к публикации 12.09.2020 г.

В исследовательских задачах, требующих применения численных методов решения систем обыкновенных дифференциальных уравнений, часто возникает необходимость выбора наиболее эффективного и оптимального для конкретной задачи численного метода. В частности, для решения задачи Коши, сформулированной для системы обыкновенных дифференциальных уравнений, применяются методы Рунге–Кутты (явные или неявные, с управлением шагом сетки или без и т.д.). При этом приходится перебирать множество реализаций численного метода, подбирать коэффициенты или другие параметры численной схемы. В данной статье предложено описание разработанной авторами библиотеки и скриптов автоматизации генерации функций программного кода на языке Julia для набора численных схем методов Рунге–Кутты. При этом для символьных манипуляций использовано программное средство подстановки по шаблону. Предлагаемый подход к автоматизации генерации программного кода позволяет вносить изменения не в каждую подлежащую сравнению функцию по отдельности, а использовать для редактирования единый шаблон, что с одной стороны дает универсальность в реализации численной схемы, а с другой позволяет свести к минимуму число ошибок в процессе внесения изменений в сравниваемые реализации численного метода. Рассмотрены методы Рунге–Кутты без управления шагом, вложенные методы с управлением шагом и методы Розенброка также с управлением шагом. Полученные автоматически с помощью разработанной библиотеки программные коды численных схем протестированы при численном решении нескольких известных задач.

DOI: 10.31857/S0132347421010052

1. ВВЕДЕНИЕ

Методы Рунге–Кутты являются основными численными методами для решения нежестких систем обыкновенных дифференциальных уравнений. Известны численные схемы высоких порядков (10 и выше) с управлением шагом и плотной выдачей. Для многих языков программирования разработаны библиотеки, реализующие наиболее эффективные из построенных численных схем. Наиболее отлаженные коды были созданы для языка Fortran 77 Э. Хайгером (Ernst Hairer) и его коллегами, они описаны в книгах [1, 2] и доступны для скачивания на сайте [3]. Эти программы реализуют методы DOPRI5 [4] и DOPRI853 [5], которые обеспечивают управление шагом и плотную выдачу. Второй из этих методов также поддерживает переключение порядка ме-

тода между 8 и 5 для обеспечения большего быстродействия при решении гладких задач.

Подпрограммы DOPRI5 и DOPRI853 очень хорошо оптимизированы и доступны для использования во многих библиотеках и математических пакетах, таких как Matlab [6], Octave [7], SciPy [8], SciLab [9], Boost C++ [10] и т.д. Однако часто для учебных и исследовательских целей возникает необходимость перебора нескольких методов Рунге–Кутты для выбора оптимального для конкретной задачи [11–13]. Кроме того, эффективность той или иной численной схемы также существенно зависит от решаемой задачи.

Чтобы можно было достоверно сравнить разные численные схемы между собой, необходимо иметь универсальный код, который реализует вложенные методы для любого набора коэффи-

циентов. При этом все реализации должны быть унифицированы и отличаться лишь набором коэффициентов.

С целью решения поставленной задачи нами разработана библиотека для языка программирования Julia [14–16], описание которой и приводится в данной статье. Библиотека состоит из двух частей: вычислительной части, реализующей алгоритмы типа Рунге–Кутты, и конструирующей части, создающей с помощью символьных вычислений конкретный вариант алгоритма Рунге–Кутты. Отличительной особенностью конструирующей части является автоматическая генерация кода функций из нескольких готовых шаблонов. Первоначально рассматривался вариант реализации конструирующей части с помощью универсальной системы компьютерной алгебры, например, с использованием SymPy [17]. Однако в ходе реализации выяснилось, что нам нужно от системы компьютерной алгебры в основном операции сопоставления по образцу. В результате было принято решение использовать более легковесное средство символьных манипуляций. Фактически, для символьных манипуляций мы используем не развитую систему компьютерной алгебры, а программное средство подстановки по шаблону. Такой подход хотя и несколько экзотичен, но оправдан большей производительностью сгенерированных функций. Для генерации кода использован язык Python [18] и шаблонизатор Jinja2 [19].

1.1. Структура статьи

В разделе 2 работы дается краткое описание вложенных методов Рунге–Кутты и стратегии управления шагом. Раздел 3 посвящен описанию разработанной библиотеки для языка программирования Julia. В разделе 4 приведены результаты тестирования сгенерированных автоматически с помощью разработанной библиотеки функций для численного решения задач для систем обыкновенных дифференциальных уравнений. В частности в этом разделе подробно рассмотрено решение ограниченной задачи трех тел (задачи по вычислению орбит Аренсторфа).

2. МЕТОДЫ РУНГЕ–КУТТЫ

Методы Рунге–Кутты применяются к задаче Коши, сформулированной для системы обыкновенных дифференциальных уравнений (ОДУ). Теория методов Рунге–Кутты хорошо известна и обстоятельно изложена в книгах [1, 2, 20]. Поэтому в данном разделе мы приведем лишь основные формулы и кратко остановимся на используемой в разработанной программной библиотеке стратегии управления шагом.

2.1. Постановка задачи

Пусть даны две гладкие функции: неизвестная функция $y^\alpha(t): [t_0, T] \rightarrow \mathbb{R}^N$ и известная функция $f^\alpha(t, y^\beta(t)): \mathbb{R} \times \mathbb{R}^N \rightarrow \mathbb{R}^N$, где $[t_0, T] \in \mathbb{R}$, $\alpha = 1, \dots, N$. Пусть также известно значение функции в начальный момент времени $y_0^\alpha = y^\alpha(t_0)$. Тогда задача Коши для системы из N обыкновенных дифференциальных уравнений формулируется следующим образом:

$$\begin{cases} \dot{y}^\alpha(t) = f^\alpha(t, y^\beta(t)), \\ y^\alpha(t_0) = y_0^\alpha, \quad \alpha, \beta = 1, \dots, N. \end{cases} \quad (1)$$

Систему (1) можно записать покомпонентно:

$$\begin{cases} \dot{y}^1(t) = f^1(t, y^1(t), \dots, y^N(t)), \\ \vdots \\ \dot{y}^N(t) = f^N(t, y^1(t), \dots, y^N(t)), \\ y^\alpha(t_0) = y_0^\alpha. \end{cases}$$

Зададим на отрезке $[t_0, T]$ сетку из набора точек $t_0 < t_1 < t_2 < \dots < t_k < \dots < t_N = T$ с шагом сетки $h_{k+1} = t_{k+1} - t_k$. Каждой точке сетки по некоторому правилу, называемому *численной схемой*, ставится в соответствие некоторая величина y_k^α , которая должна с необходимой точностью аппроксимировать решение системы ОДУ в точках сетки, т.е. $y_k^\alpha \approx y^\alpha(t_k)$. Для оценки погрешности аппроксимации используют норму $\|y^\alpha(t_k) - y_k^\alpha\|$.

2.2. Явные вложенные методы Рунге–Кутты

Погрешность метода Рунге–Кутты может оцениваться и другим образом. Идея состоит в том, что кроме основного решения y_m^α в точке рассматривается также вспомогательное решение \hat{y}_m^α , выполненное с помощью метода Рунге–Кутты смежного порядка. Разность этих решений может служить оценкой локальной погрешности метода меньшего порядка. Полученная оценка локальной погрешности помогает выбрать переменный шаг интегрирования. Методы, использующие данный способ оценки локальной погрешности, называются *вложенными* методами Рунге–Кутты [1]. При этом метод y_m^α называется *основным*, а метод \hat{y}_m^α – *вложенным*.

Явный вложенный численный метод Рунге–Кутты для задачи Коши (2.1) задается следующими формулами:

$$k^{l\alpha} = f^\alpha(t_m, y_m^\beta),$$

$$k^{i\alpha} = f^\alpha\left(t_m + c^i h_m, y_m^\alpha + h \sum_{j=1}^{i-1} a_j^i k^{j\beta}\right),$$

$$i = 2, \dots, s,$$

$$y_{m+1}^\alpha = y_m^\alpha + h_m (b_1 k^{1\alpha} + b_2 k^{2\alpha} + \dots + b_{s-1} k^{s-1,\alpha} + b_s k^{s\alpha}),$$

$$\hat{y}_{m+1}^\alpha = y_m^\alpha + h_m (\hat{b}_1 k^{1\alpha} + \hat{b}_2 k^{2\alpha} + \dots + \hat{b}_{s-1} k^{s-1,\alpha} + \hat{b}_s k^{s\alpha}),$$

где N – число уравнений в системе ОДУ, s – число стадий численного метода. Латинские индексы $i, j, l = 1, \dots, s$ относятся к численной схеме, греческие α, β – к системе ОДУ, а индексом m обозначен номер шага.

Порядок аппроксимации решения сеточными функциям y_m^α и \hat{y}_m^α различен, что позволяет на каждом шаге вычислять погрешность аппроксимации, поэтому при указании порядка вложенного метода используют обозначение $p(\hat{p})$, где p и \hat{p} – порядки основного и вложенного метода Рунге–Кутты соответственно.

Величины $c^i, a_j^i, b_j, \hat{b}_j$ полностью определяют численную схему и называются *коэффициентами метода*, их принято группировать в виде *таблицы Бутчера*, названной так в честь Дж.Ч. Бутчера (John C. Butcher):

0	0	0	...	0	0
c^2	a_1^2	0	...	0	0
c^3	a_1^3	a_2^3	...	0	0
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
c^{s-1}	a_1^{s-1}	a_2^{s-1}	...	0	0
c^s	a_1^s	a_2^s	...	a_{s-1}^s	0
	b_1	b_2	...	b_{s-1}	b_s
	\hat{b}_1	\hat{b}_2	...	\hat{b}_{s-1}	\hat{b}_s

Нахождение коэффициентов метода Рунге–Кутты для порядков выше 4 является отдельной сложной задачей, которой мы здесь не будем касаться и которая подробно раскрыта в книге [1].

Заметим только, что на коэффициенты c^i и a_j^i также принято налагать дополнительные условия следующего вида: $c^i = a_1^i + \dots + a_s^i$.

Явный метод Рунге–Кутты наиболее прост для реализации в программном виде, так как на каждом шаге метода необходимо последовательно вычислять элементы двухмерного массива $k[1:s, 1:N]$ в цикле по первому измерению $i=1:s$. Элемент $k[1, 1:N]$ требует только значений t_m, y_m , вычисленных на предыдущем шаге, элемент $k[2, 1:N]$ требует $k[1, 1:N]$, элемент $k[3, 1:N]$ требует $k[1, 1:N]$ и $k[2, 1:N]$ и т.д. Заметим, что ин-

дексация элементов массива в языке Julia начинается с 1, также как и в языке Fortran.

К настоящему времени найдены коэффициенты для множества вложенных методов Рунге–Кутты вплоть до 10-го порядка точности. Наиболее часто используются методы, найденные Дж.Р. Дорманом в соавторстве с П.Дж. Принцем (J.R. Dormand, P.J. Prince) [4, 5, 21], Э. Фельбергом (E. Fehlberg) [22, 23] и Дж.Р. Кэшем в соавторстве с А.Х. Карпом (J.R. Cash, A.H. Karp) [24].

2.3. Стратегии управления длиной шага

Существуют различные стратегии управления величиной шага h в зависимости от локальной погрешности. Выбор той или иной стратегии чаще всего диктуется характером решаемой задачи. Здесь мы изложим классический алгоритм, предложенный в книге Э. Хайрера [2], который основан на идеях из теории управления и хорошо работает для большинства нежестких задач.

Пусть y_m^α и \hat{y}_m^α – два численных решения разного порядка аппроксимации, вычисленные на шаге m работы алгоритма. Потребуем, чтобы $|y_m^\alpha - \hat{y}_m^\alpha| \leq sc^\alpha$, где величина желаемой локальной погрешности sc^α (scale) вычисляется по формуле

$$sc^\alpha = A_{tol} + \max(|y_m^\alpha|, |\hat{y}_m^\alpha|) R_{tol},$$

где A_{tol} и R_{tol} – желаемая величина абсолютной и относительной погрешности соответственно. Допустимая ошибка на шаге m находится как среднее квадратичное:

$$E_m = \sqrt{\frac{1}{N} \sum_{\alpha=1}^N \left(\frac{y_m^\alpha - \hat{y}_m^\alpha}{sc^\alpha} \right)^2}.$$

Новая величина шага вычисляется по формуле

$$h_{m+1} = h_m / \max(f_{\min}, \min(f_{\max}, E_m^a E_{m-1}^{-b} / f_s)),$$

где E_{m-1} – величина ошибки, вычисленная на предыдущем шаге. Фактор f_s обычно равен 0.9 или 0.8 и предназначен для предотвращения слишком резкого увеличения величины шага. Показатели степени a и b подбираются исходя из задачи. В книге [2] рекомендуются следующие универсальные значения $a = 0.7/p - 0.75b$ и $b = 0.4/p$, где p – порядок аппроксимации используемого метода. Факторы f_{\min} и f_{\max} позволяют ограничить границы изменения размера шага и также зависят от решаемой задачи. На практике хорошо зарекомендовали себя значения $f_{\min} = 0.1$ и $f_{\max} = 5.0$.

В случае $E_m < 1$ вычисленные значения y_m^α считаются удовлетворительными и метод переходит к следующему шагу вычислений. Если же $E_m > 1$, то вычисления считаются неудовлетворительными

ми и текущий шаг повторяется с новым значением величины h_m

$$h_m \leftarrow h_m / \min(f_{\max}, E_m^a / f_s).$$

В качестве начального значения y_m^α на следующем шаге $m + 1$ можно использовать как непосредственно y_m^α , так и значение вложенного метода \hat{y}_m^α .

3. ОПИСАНИЕ ПРОГРАММЫ

3.1. Мотивация использования генерации кода

Универсальная реализация явного вложенного метода Рунге–Кутты подразумевает, что программе в качестве параметров должны передаваться массивы коэффициентов из таблиц Бутчера, которые она будет использовать для проведения расчетов. Очевидным способом хранения данных коэффициентов является использование массивов. Однако матрица a_j^i является ниже-диагональной и хранение ее в виде массива $s \times s$ приводит к тому, что более чем половина выделенной для массива памяти тратится на хранение нулей. Коэффициенты c^i , b_j и \hat{b}_j также часто имеют в своем составе нули, хранение которых неразумно.

В связи с этим большинство кодов, реализующих явные вложенные методы Рунге–Кутты, используют для хранения коэффициентов набор именованных констант, а не массивы. Это вызвано также и тем, что операции со скалярными величинами в большинстве языков проводятся быстрее, чем операции с массивами.

В языке Julia стандартные массивы являются динамическими, поэтому накладные расходы на хранение двумерного массива $s \times s$ больше, чем на расходы на хранение $s \times s$ именованных констант.

При сохранении требования универсальности создаваемого кода и вместе с тем желание увеличить скорости вычислений и уменьшить расход памяти, привели нас к решению использовать автоматическую генерацию кода по одному шаблону для каждого отдельного метода.

Кроме выигрыша в производительности, автоматическая генерация кода позволяет добавлять или изменять не каждую функцию по отдельности, а все функции совокупно путем редактирования одного лишь шаблона. Это позволяет как уменьшить количество ошибок, так и генерировать различные варианты функций для разных целей.

3.2. Описание реализации генератора методов Рунге–Кутты

В качестве языка для генерации кода нами был выбран язык Python, так как он поддерживает разнообразные средства для работы с текстом.

Также в стандартной библиотеке Python присутствует тип данных Fraction, который позволяет задать коэффициенты Рунге–Кутты в виде рациональных дробей, а потом уже преобразовывать их в вещественный вид с нужным порядком точности.

Кроме самого языка Python была использована библиотека для обработки шаблонов Jinja2 [19]. Данный шаблонизатор (template engine) разрабатывался изначально для генерации HTML-страниц, однако он обладает очень гибким синтаксисом и может использоваться как универсальное средство для генерации текстовых файлов любого вида, в том числе и исходных кодов на любых языках программирования. Кроме Jinja2 мы использовали библиотеку numpy [8] для работы с массивами коэффициентов.

Шаблоны для генерации функций хранятся в файлах rk_method.jl и erk_method.jl, которые представляют собой исходный код на языке Julia с инструкциями Jinja2. Шаблонизатор позволяет поместить всю логику генерации кода в шаблон и передавать извне только данные о методе.

Информация о численных схемах хранится в виде JSON-файла, где каждый метод представляет собой JSON-объект примерно следующего вида:

```
{ "name": "Название (одним словом)",
  "description": "Краткое описание",
  "stage": [s],
  "order": [p],
  "extrapolation_order": [p_hat],
  "a": [[ "0", "0", "0" ], [ "1", "0",
    ↪ "0" ], [ "1/4", "1/4", "0" ] ],
  "b": [ "1/2", "1/2", "0" ],
  "b_hat": [ "1/6", "1/6", "2/3" ],
  "c": [ "0", "1", "1/2" ] }
```

Название JSON-объекта используется в качестве имени сгенерированной в последствии функции. Массивы a, b, b_hat, c могут быть как числового, так и строкового типа. Если коэффициенты метода заданы в виде рациональных дробей, то можно указать их в виде "m/n", затем они будут преобразованы в объект Fraction стандартной библиотеки Python, а в теле сгенерированной функции будут представлены десятичной дробью двойной точности с 17 значащими знаками.

Список объектов указанного выше вида последовательно обрабатывается скриптами erk_generator.py и rk_generator.py. Скрипты, используя переданную им информацию о мето-

дах, генерируют для каждого метода код нескольких функций языка Julia.

Скрипты позволяют сгенерировать функции для 16 вложенных методов Рунге–Кутты. Методы низкого порядка взяты из книги Хайрера [1]. Из методов пятого порядка и выше заданы коэффициенты из работ [4, 5, 21] и [22, 23]. Метод из работы [24] реализован отдельно, так как предусматривает переключение порядка точности и сложный алгоритм управления шагом. Для добавления своих методов следует задать JSON-объект указанного выше формата и запустить скрипты вновь. Для добавленных методов будут сгенерированы соответствующие функции.

3.3. Описание сгенерированных функций

Разработанная нами библиотека имеет стандартную для модулей языка Julia структуру. Весь исходный код находится в каталоге `src`. В подкаталоге `src/generated` располагаются файлы с автоматически генерируемыми функциями, которые, в свою очередь, с помощью директивы `include` включаются в главный файл модуля `src/RungeKutta.jl`. Код, ответственный за управление шагом, находится в файле `StepControl.jl` и является общим для всех методов.

Для каждого из вложенных методов Рунге–Кутты генерируются три функции, две из которых имеют следующий вид:

```
ERK(func, A_tol, R_tol, x_0, t_start,
    ⇨ t_stop) -> (T, X)
```

```
ERK(func, A_tol, R_tol, x_0, t_start,
    ⇨ t_stop, last) -> (t, x)
```

Функции имеют одинаковое имя, а список их аргументов различается лишь последним аргументом. Благодаря поддержке языком Julia множественной диспетчеризации (перегрузка функций), компилятор сам определяет в каком случае какую из реализаций вызывать.

В указанных выше функциях:

- `func::Function` – правая часть системы ОДУ $\dot{x}^\alpha(t) = f^\alpha(t, x^\beta)$: `func(t, x)`, где `t::Float64` – время, `x::Vector{Float64}` – значение функции $x(t)$; аргумент `t` должен всегда присутствовать в вызове функции, даже если система автономная и явно от времени не зависит;

- аргументы `A_tol` и `R_tol` должны иметь тип `Float64` и обозначают абсолютную и относительную точности методы соответственно;

- `x_0::Vector{Float64}` – начальное значение функции $x_0^\alpha = x^\alpha(t_0)$;

- `t_start` и `t_stop` имеют тип `Float64` и обозначают начальную и конечную точки промежутка интегрирования;

- если указан аргумент `last::Bool`, то будут возвращены последняя точка промежутка интегрирования t_n и вектор $x_n^\alpha \approx x^\alpha(t_n)$, в противном случае возвращаются массивы `T::Vector{Float64}` и `Matrix{Float64}`.

Для методов Рунге–Кутты без управления шагом генерируются аналогичные функции, единственное отличие которых заключается в отсутствии аргументов `A_tol` и `R_tol`. Вместо них следует передавать аргумент `h`, который задает размер шага, используемого для вычислений.

Для реализации алгоритма управления шагом генерируется еще одна функция:

```
ERK_info(func, A_tol, R_tol, x_0,
    ⇨ t_start, t_stop) -> (accepted_t,
    ⇨ accepted_h, rejected_t, rejected_h)
```

Эта функция возвращает следующие массивы:

- `accepted_t`: все точки сетки, в которых погрешность вычислений была признана удовлетворительной;

- `accepted_h`: все принятые размеры шагов;

- `rejected_t`: все точки сетки, в которых погрешность вычислений была признана неудовлетворительной;

- `rejected_h`: все отбракованные размеры шага;

- `errors`: значения локальной ошибки E_n .

Все возвращаемые значения имеют тип `Vector{Float64}`.

4. ТЕСТИРОВАНИЕ СГЕНЕРИРОВАННЫХ ПРОГРАММНЫХ КОДОВ ЧИСЛЕННЫХ СХЕМ

Для тестирования созданных методов были использованы три системы дифференциальных уравнений, рассмотренные в [2]. Первая система – уравнения ван дер Поля:

$$\begin{cases} \frac{d^2x}{dt^2} - \mu(1-x^2)\frac{dx}{dt} + x = 0, \\ \dot{x}_1(t) = x_2(t), \\ \dot{x}_2(t) = \mu(1-x_1^2(t))x_2(t) - x_1(t), \end{cases}$$

заданные со следующими начальными значениями:

$$\mu = 1, \quad \mathbf{x} = (0, \sqrt{3})^T, \quad 0 \leq t \leq 12,$$

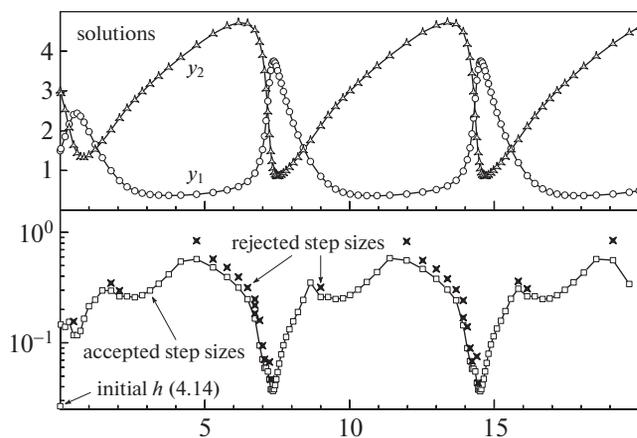


Рис. 1. Решение системы (2) и отброшенные и принятые шаги (рисунок из [1, с. 170, рис. 4.1]).

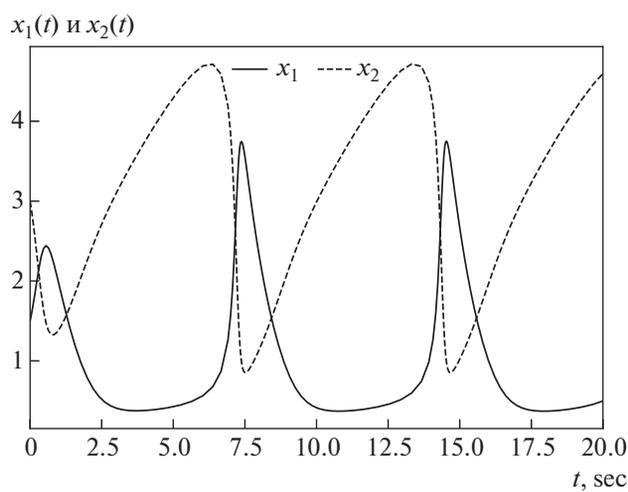


Рис. 2. Решение системы (2).

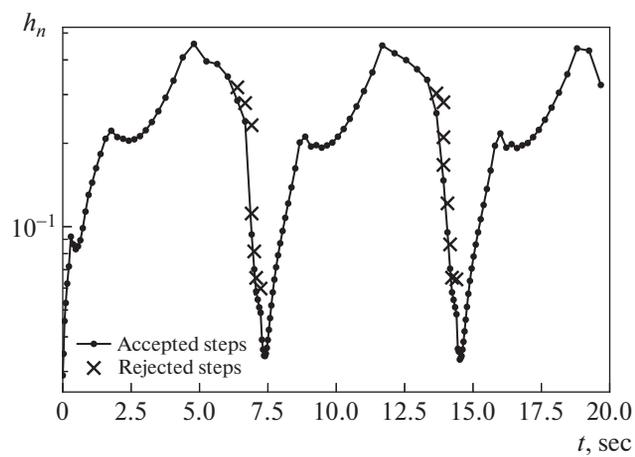


Рис. 3. Отброшенные и принятые шаги (2).

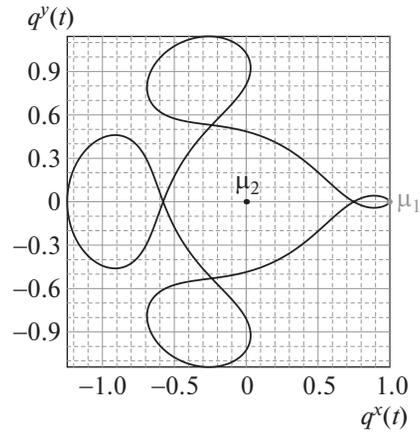


Рис. 4. Орбита для первой группы начальных значений (3).

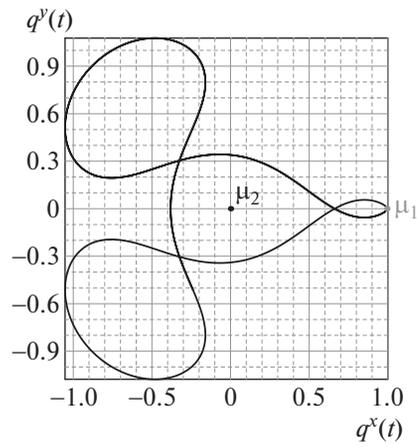


Рис. 5. Орбита для второй группы начальных значений (4).

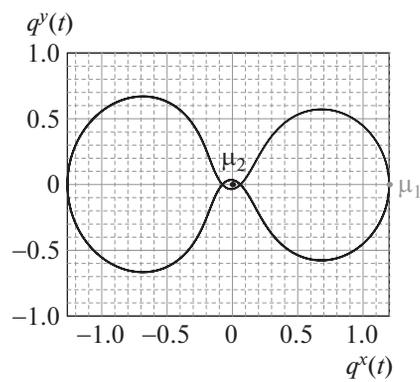


Рис. 6. Орбита для третьей группы начальных значений (5).

Таблица 1. Глобальные погрешности численных методов за один оборот по орбите для первой группы начальных значений

Метод	Погрешность
DPRK546S	1.84566×10^{-12}
DPRK547S	2.93634×10^{-12}
DPRK658M	4.22771×10^{-12}
Fehlberg45	7.42775×10^{-12}
DOPRI5	1.95463×10^{-13}
DVERK65	1.67304×10^{-12}
Fehlberg78B	5.45348×10^{-12}
DOPRI8	1.06343×10^{-11}

где коэффициент μ характеризует нелинейность и силу затухания колебаний.

Вторая система – уравнения твердого тела без внешних сил, или иначе, уравнения Эйлера твердого тела:

$$\begin{cases} \dot{x}_1(t) = I_1 x_2(t) x_3(t), \\ \dot{x}_2(t) = I_2 x_1(t) x_3(t), \\ \dot{x}_3(t) = I_3 x_1(t) x_2(t), \\ I_1 = -2, \quad I_2 = 1.25, \quad I_3 = -0.5, \\ \mathbf{x} = (0, 1, 1)^T, \quad t = [0, 12]. \end{cases}$$

Для проверки работы алгоритма управления шагом применялась численная схема вложенного метода Рунге–Кутты с $p = 4$ и $\hat{p} = 3$ к системе уравнений бруселятора:

$$\begin{cases} \dot{x}_1(t) = 1 + x_1^2 x_2 - 4x_1, \\ \dot{x}_2(t) = 3x_1 - x_1^2 x_2. \end{cases}$$

Для начальных условий $x_1(0) = 1.5$, $x_2(0) = 3$ уравнение бруселятора было численно проинтегрировано методом ERK43b на отрезке $0 \leq t \leq 20$ с абсолютной и относительной погрешностями $A_{tol} = R_{tol} = 10^{-4}$. Были построены графики, идентичные графикам из книги [1, с. 170, рис. 4.1]. При сравнении результатов из книги [1] (см. рис. 1) с полученными нами результатами (см. рис. 2 и 3) видно, что реализованный нами метод работает практически также, как и использованный в [1], однако выбирает размер шага аккуратнее.

Еще одна часто применяющаяся для тестирования численных схем система уравнений – частный случай ограниченной задачи трех тел, назы-

ваемый задачей по вычислению орбит Аренсторфа. Свое название данная задача получила в честь Ричарда Ф. Аренсторфа – американского физика, рассчитавшего стабильную орбиту малого тела между Луной и Землей.

В ограниченной задаче трех тел рассматривается движение малого тела в поле тяготения среднего и большого тел (Луна и Земля). Масса малого тела считается равной нулю, массы среднего и большого тел – μ_1 и μ_2 соответственно. Предполагается, что орбиты тел лежат в одной плоскости.

В задаче Аренсторфа в зависимости от начальных значений можно получить разные стабильные орбиты [1]. В безразмерных синодических координатах три группы начальных значений, дающих три разные орбиты, имеют следующий вид:

$$p_y = -1.00758510637908238, \quad p_x = 0.0, \\ q^x = 0.994, \quad q^y = 0.0; \quad (2)$$

$$p_y = -1.03773262955733680, \quad p_x = 0.0, \\ q^x = 0.994, \quad q^y = 0.0; \quad (3)$$

$$p_y = 0.15064248999999985, \quad p_x = 0.0, \\ q^x = 1.2, \quad q^y = 0.0. \quad (4)$$

Здесь p_x, p_y – обобщенные импульсы системы, q^x, q^y – обобщенные координаты системы.

Важно отметить, что начальные значения специально указаны с большой точностью, так как малое тело крайне близко подходит к среднему телу и даже небольшая погрешность вычислений может привести к ошибке, что физически интерпретируется как падение малого тела на среднее. Как раз из-за такой особенности данная задача хорошо подходит для проверки реализации численных схем.

Функция Гамильтона в синодических координатах записывается следующим образом:

$$H(p_x, p_y, q^x, q^y) = \\ = \frac{1}{2}(p_x^2 + p_y^2) + p_x q^y - p_y q^x - F(q^x, q^y),$$

где

$$F(q^x, q^y) = \frac{\mu_1}{r_1} + \frac{\mu_2}{r_2}, \quad \mu_1 + \mu_2 = 1, \\ r_1 = \sqrt{(q^x - \mu_2)^2 + (q^y)^2}, \\ r_2 = \sqrt{(q^x + \mu_1)^2 + (q^y)^2};$$

$$\frac{\partial F}{\partial q^x} = -\frac{\mu_1(q^x - \mu_2)}{r_1^3} - \frac{\mu_2(q^x + \mu_1)}{r_2^3},$$

$$\frac{\partial F}{\partial q^y} = -\frac{\mu_1 q^y}{r_1^3} - \frac{\mu_2 q^y}{r_2^3}.$$

Канонические уравнения, к которым и будет применяться численная схема, имеют следующий вид:

$$\begin{cases} \dot{p}_x = +p_y + \frac{\partial F}{\partial q^x}, & \dot{q}^x = p_x + q^y, \\ \dot{p}_y = -p_x + \frac{\partial F}{\partial q^y}, & \dot{q}^y = p_y - q^x. \end{cases}$$

Численное решение проводится для $0 \leq t \leq 17.065216560157962558$ с абсолютной и относительной погрешностями $A_{tol} = 10^{-17}$ и $R_{tol} = 0$ и средней массой $\mu_1 = 0.012277471$. В результате получаем орбиты Аренсторфа (рис. 4, 5 и 6) для начальных значений (2), (3) и (4) соответственно.

Для оценки погрешности система решается численно для временного промежутка, равного одному периоду. В конечной точке промежутка малое тело должно вернуться в начальную точку, т.е. $(q^x(0), q^y(0)) = (q^x(T), q^y(T))$, поэтому погрешность $\|\mathbf{q}_0 - \mathbf{q}_n\|$ даст глобальную погрешность метода. Значения погрешности для первой группы начальных значений приведено в табл. 1.

5. ЗАКЛЮЧЕНИЕ

Таким образом, можно сформулировать основные результаты нашей работы:

1. В качестве специализированной системы символьных вычислений применен язык шаблонизатора Jinja2. Относительно использования универсальной системы символьных вычислений данное решение позволило сделать программный комплекс более простым, компактным и увеличить его переносимость.

2. Создан набор сценариев на языке Python с использованием языка шаблонизатора Jinja2, которые генерируют код на языке Julia, реализующий численные схемы методов Рунге–Кутты без управления шагом, вложенных методов с управлением шагом и методов Розенброка также с управлением шагом (все программы находятся в открытом доступе и расположены по адресу https://bitbucket.org/mngev/rungekutta_generator).

3. Сгенерированные функции протестированы с помощью решения нескольких типовых задач (непосредственно в тексте статьи подробно разобрана задача Аренсторфа, как наиболее требовательная к точности численной схемы); модуль для языка Julia также доступен по адресу <https://bitbucket.org/mngev/rungekutta-autogen>.

6. БЛАГОДАРНОСТИ

Публикация подготовлена при поддержке Программы РУДН “5-100” и при финансовой поддержке РФФИ в рамках научного проекта 19-01-00645.

СПИСОК ЛИТЕРАТУРЫ

1. Hairer E., Nørsett S.P., G. Wanner. Solving Ordinary Differential Equations I. 2 edition. Berlin: Springer, 2008.
2. Hairer E., Wanner G. Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems. 2 edition. 1996.
3. Fortran and Matlab Codes. URL: <https://www.uni-ge.ch/~hairer/software.html>.
4. Dormand J.R., Prince P.J. A family of embedded Runge–Kutta formulae // Journal of computational and applied mathematics. 1980. Vol. 6, no. 1. P. 19–26.
5. Prince P.J., Dormand J.R. High order embedded Runge–Kutta formulae // Journal of Computational and Applied Mathematics. 1981. V. 7. № 1. P. 67–75.
6. Matlab. 2020. URL: <https://www.mathworks.com/products/matlab.html>.
7. GNU Octave. 2020. URL: <https://www.gnu.org/software/octave/>.
8. Jones E., Oliphant T., Peterson P., Others. SciPy: Open source scientific tools for Python. URL: <http://www.scipy.org/>.
9. Scilab. 2020. URL: <http://www.scilab.org/>.
10. Boost. Boost C++ Libraries. 2020. URL: <http://www.boost.org/>.
11. 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. Springer International Publishing, 2016. V. 470. P. 169–179. arXiv: 1606.06604.
12. Kulyabov D.S., Gevorkyan M.N., Demidova A.V. et al. Implementation Diffi«culties Analysis of Stochastic Numerical Runge–Kutta Methods // 2nd International Scientific Conference “Convergent Cognitive Information Technologies”, Convergent 2017 / Ed. by M. Shneps-Shneppe, V. Sukhomlin, E. Zubareva. Vol. 2064 of CEUR Workshop Proceedings. Moscow: CEUR-WS, 2017. 11. P. 28–40.
13. 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. Vol. 919 of Communications in Computer and Information Science. P. 532–546. arXiv: 1811.01719.
14. Bezanson J., Edelman A., Karpinski S., Shah V.B. Julia: A Fresh Approach to Numerical Computing. 2014.
15. Bezanson J., Karpinski S., Shah V.B., Edelman A. Julia: A Fast Dynamic Language for Technical Computing. 2012.

16. *Kwong T.* Hands-On Design Patterns and Best Practices with Julia. Birmingham B3 2PB, UK: Packt Publishing, 2020.
17. *Геворкян М.Н., Королькова А.В., Кулябов Д.С., Севастьянов Л.А.* Пример модульного расширения системы компьютерной алгебры // Программирование. 2020. № 2. С. 30–37.
18. Python Reference Manual: Rep.; Executor: Guido Rossum. Amsterdam, The Netherlands, The Netherlands: 1995.
19. Jinja2 official site. URL: <http://http://jinja.pocoo.org>.
20. *Butcher J.C.* Numerical Methods for Ordinary Differential Equations. 2 edition. New Zealand: Wiley, 2003.
21. *Dormand J.R., Prince P.J.* A reconsideration of some embedded Runge–Kutta formulae // Journal of Computational and Applied Mathematics. 1986. V. 15. № 2. P. 203–211.
22. *Fehlberg E.* Klassische Runge–Kutta Formeln fünfter und siebenter Ordnung mit Schrittweiten-Kontrolle // Computing. 1969. 6. Bd. 4, H. 2. S. 93–106.
23. *Fehlberg E.* Klassische Runge–Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungsprobleme // Computing. 1970. 3. Bd. 6, H. 1–2. S. 61–71.
24. *Cash J., Karp A.* A variable order Runge–Kutta method for initial value problems with rapidly varying right-hand sides // ACM Trans. Math. Softw. 1990. V. 16. № 3. P. 201–222.

МЕТОД ЦВЕТНЫХ ГРАФОВ ДЛЯ УПРОЩЕНИЯ ВЫРАЖЕНИЙ С ИНДЕКСАМИ

© 2021 г. Г. Б. Шпиз^{a,*}, А. П. Крюков^{a,**}

^a Московский государственный университет имени М.В. Ломоносова
119991 Москва, ГСП-1, Ленинские горы, д. 1, Россия

*E-mail: shpiz@theory.sinp.msu.ru

**E-mail: kryukov@theory.sinp.msu.ru

Поступила в редакцию 06.08.2020 г.

После доработки 28.08.2020 г.

Принята к публикации 14.09.2020 г.

Компьютерная алгебра все шире применяется в научных и прикладных вычислениях. В качестве примера приведем тензорные вычисления или в более широком смысле этого слова — упрощение выражений с индексами. В настоящей статье развивается метод цветных графов для упрощения абстрактных выражений с индексами на случай, когда индексы могут быть отнесены к нескольким различным типам. Примерами таких индексов могут быть верхние и нижние индексы в тензорных выражениях. Предложенный подход позволяет значительно уменьшить число перебираемых вариантов при поиске канонической формы выражения, что резко ускоряет процесс вычислений.

DOI: 10.31857/S0132347421010118

1. ВВЕДЕНИЕ

Выражения с индексами являются одним из самых распространенных математических объектов, которые используются для вычислений в различных областях математики, физике, других областях науки, а также в прикладных областях, а в первую очередь в инженерных науках. Наиболее типичный пример таких вычислений — это тензорные вычисления. Мы будем рассматривать абстрактные выражения с индексами, обладающие определенными свойствами симметрии по отношению к их перестановкам, и линейными соотношениями между собой.

Представленная работа является логическим продолжением работы [1], в которой были даны основные определения для случая, когда индексы одного типа. В настоящей работе мы даем обобщение предложенного формализма на случай индексов нескольких типов. Примерами таких индексов могут быть верхние и нижние индексы у тензоров.

Основная задача, которую мы решаем — это введение выражения с индексами к канонической форме. Для этого будут использованы цветные графы, связанные с выражениями, и их автоморфизмы.

Само выражение с индексами является коммутативным произведением индексированных символов (элементарных индексированных выражений), причем каждый индекс имеет “цвет”, зависящий только от места индекса, но не от его имени.

Заметим, что как и прежде нас не интересует природа индексных выражений, а мы рассматриваем их как некоторые абстрактные объекты, имеющие определенный набор свойств по отношению к перестановкам индексов.

В работе мы будем придерживаться правила суммирования Эйнштейна, когда наличие пары одинаковых индексов означает сумму по всем значениям данной пары:

$$T_i^i \equiv \sum_{i=1}^{i=n} T_i^i,$$

где n — это размерность пространства.

Такие индексы называются индексами суммирования или немymi индексами.

В данной работе мы не будем останавливаться на примерах использования выражений с индексами и типах алгоритмов, используемых для их

упрощения. Скажем только, что традиционно для упрощения выражений с индексами используются следующие четыре подхода и их вариации:

- двойной класс смежности [7];
- групповая алгебра [8];
- базис Гребнера [5];
- изоморфизм графов [1].

Желающие могут обратиться к работам А.В. Кольковой, Д.С. Кулябова, Л.А. Севастьянова [2] и Г. Шпица и А. Крюкова [1], в которых данные вопросы рассмотрены более подробно. Там же можно найти большой список литературы, в котором описаны традиционно применяемые подходы и алгоритмы.

Из работ, которые вышли в последнее время или не были упомянуты в работе [1], в первую очередь хочется указать на подробный обзор М. Маскалума [3] по использованию компьютерной алгебры в задачах теории гравитации, которые являются одним из наиболее частых примеров тензорных вычислений. Упомянем работу, которая использует алгебру графов, но для ограниченного случая полиномов из тензоров Римана, — это работа [4]. С точки зрения компьютерной алгебры, выражения с индексами могут быть разного типа. В частности, задача приведения многопетлевых интегралов в квантовой теории поля [6] может рассматриваться как такой пример.

Структура статьи следующая. В разделе 2 дадим постановку задачи. В разделе 3 дадим строгое определение выражений с абстрактными индексами и их свойств. В этом же разделе дадим строгое определение оператора приведения к каноническому виду. В заключение перечислим полученные результаты, а также направление дальнейшего развития предложенных методов.

2. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Прежде чем перейти к основной теме работы дадим несколько базовых определений объектов, с которыми мы будем работать в дальнейшем.

Определение 1. Назовем *цветным символом* или просто *символом* выражение вида $t(i_1, \dots, i_n, c_1, \dots, c_n)$, где $t \in T$ — множество типов символов, $i \in I$ — множество индексов, $c \in C$ — множество цветов индексов i .

Произведение символов t будем называть выражением с индексами.

Символы t , могут обладать различными свойствами симметрии по отношению к перестановкам индексов, которые являются подгруппой группы перестановок, а также линейными соотношениями, которые включают три или более символа. Индексы в символьном выражении, встречающиеся дважды (возможно с разными цветами), считаются индексами суммирования. Индексы

суммирования и только они будут именоваться натуральными числами.

Замечание. Если оба индекса суммирования принадлежат одному символу, то преобразования такого символа полностью факторизуются от остального выражения и нами рассматриваться не будут как тривиальные.

Будем считать, что на множестве символов определен лексикографический порядок, причем:

- любой тип символа меньше индекса;
- любой свободный индекс меньше любого индекса суммирования;
- индексы суммирования и только они являются натуральными числами;
- цвета соответствуют номерам индексов, а не их именам. Таким образом при переименовании индексов их цвета не меняются.

Таким образом, мы определяем лексикографический порядок на символах $t(i_1, \dots, i_n, c_1, \dots, c_n)$ следующим образом.

$t(i_1, \dots, i_n, c_1, \dots, c_n) < t'(i'_1, \dots, i'_n, c'_1, \dots, c'_n) \Leftrightarrow (t, i_1, \dots, i_n, c_1, \dots, c_n) < (t', i'_1, \dots, i'_n, c'_1, \dots, c'_n)$ в лексикографическом смысле. Мы также считаем что на множестве мономов (произведение символов) определен естественный лексикографический порядок, а множество всех мономов становится линейно упорядоченным.

Важную роль в дальнейших построениях будет играть понятие сигнатуры символа.

Определение 2. Сигатурой k -го индекса i_k символа $v = t(i_1, \dots, i_n, c_1, \dots, c_n)$ будем называть пару (c_k, i_k) и обозначать эту сигнатуру σ_k .

Раскраска индексов может использоваться для различения пространств с которыми эти индексы связаны. Например, верхние индексы могут быть связаны с некоторым (арифметическим) векторным пространством, а нижние с его сопряженным $(t_{\mu\nu}^{kl})$. Индексы в различных типах символов, также могут быть связаны с различными пространствами. Так, индексы в символах частных производных связаны с пространством линейных дифференциальных операторов с постоянными коэффициентами, а индексы в символах функций с пространством аргументов. Например, если символ соответствует дифференциальному оператору вида $p(i_1, \dots) d(j_1, \dots)$, то цвет может определять принадлежность индекса коэффициенту или дифференцированию.

Определение 3. Сигатурой символа $t(i_1, \dots, i_n, c_1, \dots, c_n)$ будем называть последовательность $\sigma(t) = (t, \sigma_1, \dots, \sigma_n)$, где $(\sigma_1, \dots, \sigma_n)$ — упорядоченная по возрастанию последовательность сигатур индексов i_1, \dots, i_n .

Будем считать, что любое множество символов с попарно различными сигнатурами линейно независимо, а символы с одинаковыми сигнатурами и только они могут быть связаны линейными соотношениями. Множество всех таких соотношений будем обозначать R и будем предполагать, что любое переименование индексов переводит линейное соотношение из R в R , то есть множество соотношений инвариантно относительно переименований индексов.

Определение 4. *Сверточным выражением будем называть такое произведение символов, в котором ни один индекс не встречается более 2-х раз. Индексы, встречающиеся в сверточном выражении один раз, будем называть свободными, а встречающиеся 2 раза – индексами суммирования.*

Сверточные выражения будем называть **слабо эквивалентными**, если они получаются друг из друга перестановкой сомножителей и переименованием индексов суммирования. Класс слабой эквивалентности сверточного выражения t будем называть **мономом** и обозначать $[t]$.

3. СТРУКТУРНЫЙ ГРАФ МОНОМА

Пусть задано сверточное выражение $t = v_1 \dots v_n$, где v_i – символ.

Рассмотрим граф, вершины которого совпадают с входящими в t символами. Вершины считаются соединенными ребром тогда и только тогда, когда они содержат хотя бы один общий индекс.

Цветом вершины будем называть ее сигнатуру, а цветом ребра, соединяющего вершины v_i и v_j причем $i \leq j$ – последовательность $(i, \sigma(v_i), j, \sigma(v_j), c)$, где c – последовательность цветов общих индексов (суммирования) для v_i и v_j , упорядоченная по возрастанию.

Определение 5. *Описанный выше раскрашенный граф называется **графом** сверточного выражения t и обозначается $G(t)$.*

Заметим, что раскраска графа $G(t)$ зависит от порядка сомножителей и от переименования индексов суммирования.

Определение 6. *Упорядоченную по возрастанию последовательность цветов всех ребер графа $G(t)$ назовем цветом графа.*

Очевидно, что граф однозначно определяется своим цветом и лексикографический порядок цветов графов определяет линейный порядок на множестве графов вида $G(u)$ при $u \in [t]$.

Определение 7. *Минимальный элемент этого множества называется **структурным графом** монома $[t]$ и обозначается $G_0([t])$.*

Определение 8. *Канонической формой \hat{G} цветного графа G называется **граф минимального цвета***

среди всех графов, получающихся из G перенумерацией вершин.

Таким образом структурный граф монома $[t]$ является канонической формой графа $G(t)$ при $t \in [m]$.

Сверточные выражения соответствующие структурному графу получаются друг из друга автоморфизмами структурного графа (естественно сохраняющими раскраску вершин и ребер), то есть перестановками сомножителей в исходном выражении, и перестановками имен индексов суммирования с одинаковыми сигнатурами. Таким образом группа “автоморфизмов монома” является прямым произведением группы автоморфизмов его структурного графа и группы перестановок имен индексов с одинаковой сигнатурой.

Нумерации, приводящие граф G к канонической форме \hat{G} определены с точностью до автоморфизмов графа \hat{G} . Алгоритм вычисления перенумераций вершин цветного графа легко получается из соответствующего алгоритма для не цветных графов, приведенного в нашей предыдущей работе [1]. В нем меняется только определение сигнатур.

Каноническая форма выражения $t \in [m]$ является средним арифметическим всех выражений из $[t]$, граф которых имеет каноническую форму. То есть результатом усреднения любого выражения $v \in [m]$, граф которого имеет каноническую форму по группе автоморфизмов монома $[m]$. Суммирование естественно следует ввести с учетом линейных соотношений для символов, входящих в выражение и законов операций сложения и умножения. При суммировании каждый символ перед перемножением следует приводить к каноническому виду, а затем раскрывать скобки.

4. ЗАКЛЮЧЕНИЕ

Представленная работа является развитием идеи определения канонического представления выражения с индексами с использованием автоморфизмов цветного графа, строящегося по определенным правилам в случае когда индексы выражения могут иметь различный тип.

В работе дано определение цветного графа и способ его построения для произвольного выражения с цветными индексами. Разобран способ приведения выражения с индексами к каноническому виду, сводящийся к вычислению нумераций вершин соответствующего ему цветного графа, приводящих этот граф к каноническому виду. Предложенный способ позволяет последовательно учесть все виды соотношений между элементарными выражениями с абстрактными индексами, включая линейные соотношения общего вида. Причем объем перебора в рамках предложенного подхода существенно сокращается, так как на

каждом этапе необходимо делать перебор не по всему групповому пространству форм индексного выражения, а только по группе автоморфизмов, связанных с цветным графом.

В настоящее время имеется экспериментальная программная реализация алгоритмов на языке Python, на которой отрабатываются предложенные методы. В дальнейшем предполагается сделать реализацию этих методов для наиболее популярных систем компьютерной алгебры таких как MAPLE [9] и REDUCE [10].

Авторы выражают благодарность В.А. Ильину за плодотворные обсуждения, С.А. Абрамову за стимулирование исследований и возможность выступить на семинаре по компьютерной алгебре [11].

Работа была выполнена в рамках Государственного задания № 115041410196.

СПИСОК ЛИТЕРАТУРЫ

1. *Shpiz G., Kryukov A.* Canonical Representation of Polynomial Expressions with Indices // Programming and Computer Software. 2019. V. 45. P. 81–87. <https://doi.org/10.1134/S0361768819020105>
2. *Korol'kova A.V., Kulyabov D.S., Sevast'yanov L.A.* Tensor Computations in Computer Algebra Systems // Programming and Computer Software. 2012. V. 39. P. 135–142.
3. *Malcolm A.H.* MacCallum. Computer algebra in gravity research // Living Reviews in Relativity. 2018. V. 21. P. 6. <https://doi.org/10.1007/s41114-018-0015-6>
4. *Li H., Li Zh., Li Y.* Riemann Tensor Polynomial Canonicalization by Graph Algebra Extension // Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation. July, 2017. P. 269–276. <https://doi.org/10.1145/3087604.3087625>
5. *Portugal R.* An algorithm to simplify tensor expressions // Comput. Phys. Commun. 1998. V. 115. P. 215–230.
6. *Steinwachs Ch.F.* Combinatorial aspects in the one-loop renormalization of higher derivative theories // ArXiv: 1909.00810, 2019. 24 p. <https://arxiv.org/abs/1909.00810>
7. *Rodionov A.Y., Taranov A.Y.* Combinatorial aspects of simplification of algebraic expressions // In Proc. Int. Conf. EUROCAL'89, Lecture Notes in Computer Science. 1989. V. 378. P. 192–201.
8. *Plyin V.A., Kryukov A.P.* ATENSOR – REDUCE program for tensor simplification // Computer Physics Communications 1996. V. 96. P. 36–52.
9. Maplesoft. <https://www.maplesoft.com/>
10. *Hearn A.C. and Schöpf R.* REDUCE User's Manual, Free Version. <https://reduce-algebra.sourceforge.io/manual/manual.html>
11. *Абрамов С.А., Боголюбская А.А.* Семинар по компьютерной алгебре в 2016–2017 гг. // Программирование. 2018. № 2. С. 3–4; <http://www.ccas.ru/sabramov/seminar/doku.php>

АЛГОРИТМ ДЛЯ РЕШЕНИЯ СЕМЕЙСТВА ДИОФАНТОВЫХ УРАВНЕНИЙ
ЧЕТВЕРТОЙ СТЕПЕНИ, УДОВЛЕТВОРЯЮЩИХ УСЛОВИЮ РУНГЕ© 2021 г. Н. Н. Осипов^{а,*}, А. А. Кытманов^{а,**}^а Институт космических и информационных технологий
Сибирского федерального университета
660074 Красноярск, ул. Киренского, 26, Россия

*E-mail: nnosipov@gmail.com

**E-mail: aakytm@gmail.com

Поступила в редакцию 19.07.2020 г.

После доработки 01.09.2020 г.

Принята к публикации 12.09.2020 г.

В статье предлагается алгоритмическая реализация элементарной версии метода Рунге для семейства диофантовых уравнений 4-й степени с двумя неизвестными. К уравнениям рассматриваемого типа сводится любое диофантово уравнение 4-й степени, старшая однородная часть которого разлагается в произведение линейного и кубического многочленов. Компьютерную реализацию алгоритма решения (в его оптимизированном виде) предполагается осуществить в системе компьютерной алгебры PARI/GP.

DOI: 10.31857/S0132347421010076

1. ВВЕДЕНИЕ

В современных системах компьютерной алгебры (таких, как Mathematica, Maple, PARI/GP, SageMath и др.) реализовано только небольшое число известных алгоритмов для решения диофантовых уравнений в целых числах. Как правило, речь идет о системах линейных уравнений с произвольным числом неизвестных, квадратных уравнениях с двумя неизвестными, а также кубических уравнений Туэ (но при некоторых дополнительных условиях, см. [1]). Между тем, имеется довольно широкий класс диофантовых уравнений с двумя неизвестными

$$f(x, y) = 0, \quad (1)$$

для которого можно предложить эффективный (предоставляющий явные верхние границы для целочисленных решений) метод решения – так называемый метод Рунге.

Краткое описание упрощенной версии метода Рунге можно найти в хорошо известных монографиях [2–4]. Необходимо отметить, что оригинальная версия метода Рунге является более общей (см. старую статью Рунге [5] или, в современном изложении, статью [6]). Несмотря на то, что метод Рунге известен уже более ста лет, его реализация в системах компьютерной алгебры весьма

ограничена. Лишь в очень небольшом числе публикаций (см. [7–9] и, особенно, [10]) обсуждаются алгоритмические аспекты реализации этого метода, которые в целом представляются нетривиальными. Последнее частично объясняется следующим обстоятельством.

Пусть многочлен $f(x, y) \in \mathbb{Z}[x, y]$ неприводим. Положим

$$d_0 = \max\{\deg_x f(x, y), \deg_y f(x, y)\}.$$

Можно доказать, что если $f(x, y)$ удовлетворяет так называемому стандартному условию Рунге (см. ниже), то справедлива оценка

$$\max\{|x|, |y|\} < (2d_0)^{18d_0^7} h^{12d_0^6} \quad (2)$$

для всех решений $(x, y) \in \mathbb{Z}^2$ уравнения (1) (см. [6]). Здесь h – высота многочлена $f(x, y)$ (максимум модулей его коэффициентов). В частности, этот результат показывает, что тривиальная реализация – полный перебор в предписанных границах – является практически бессмысленной даже в случае довольно малого d_0 .

Метод Рунге основан на конструктивном способе доказательства теоремы Рунге о конечности

множества решений уравнения (1) в целых числах. В литературе широко известна упрощенная версия этой теоремы, которая и приводится ниже (полную версию, помимо первоисточника, можно найти, например, в [6]).

Пусть $d = \deg f(x, y)$ и $f_d(x, y)$ – старшая однородная часть многочлена $f(x, y)$.

Теорема Рунге. Предположим, что $f_d(x, y)$ разложима в произведение двух взаимно простых многочленов из $\mathbb{Z}[x, y]$ положительных степеней. Тогда уравнение (1) имеет конечное множество решений $(x, y) \in \mathbb{Z}^2$.

Далее для краткости условие теоремы Рунге будем называть *стандартным условием Рунге*. В случае кубических уравнений ($d = 3$) при выполнении стандартного условия Рунге в работе [9] был предложен практически работающий алгоритм решения уравнения (1). Этот алгоритм основан на так называемой элементарной версии метода Рунге для диофантовых уравнений степени не выше четвертой (см. [11]). Алгоритмическая реализация элементарной версии метода Рунге при $d = 4$ пока получена только в некоторых частных случаях (см. [12, 13]). Следует также упомянуть хорошо известный элементарный алгоритм решения для наиболее простого типа уравнений 4-й степени, к которым применим метод Рунге (этот алгоритм подробно описан в [7]). Как отмечено в [10], имеет смысл “избегать разложений в ряды Пюизо и алгебраических коэффициентов”, поскольку это обычно приводит к плохим оценкам типа (2). Собственно, данное обстоятельство и составляет основную трудность в алгоритмической реализации метода Рунге.

Элементарная версия метода Рунге для диофантовых уравнений малой степени базируется на специальных параметризациях, которые позволяют перечислить возможные целочисленные решения. Как результат, решение данного диофантова уравнения может быть сведено к решению конечного множества уравнений (как правило, квадратных или кубических) с одним неизвестным в целых числах. Эта идея была воплощена в работах [9, 12, 13]. Особенно хорошо метод работает в случае кубических диофантовых уравнений, где необходимо решать только квадратные уравнения (и, следовательно, все сводится к алгоритмически простой задаче извлечения квадратных корней).

Пример 1. а) Для решения диофантова уравнения

$$x(y^2 - 2x^2) + Hx + y + 1 = 0$$

(см. пример 9 из [9]) достаточно решить порядка $|H|^{1/4}$ квадратных уравнений с одним неизвестным. В частности, это позволяет за приемлемое время решить уравнение для каждого H в диапазоне $-10^{10} \leq H < 0$.

б) Аналогично, диофантово уравнение

$$xy^2 = 2Hx^2 + y$$

сводится к решению порядка $|H|^{1/3}$ квадратных уравнений. Это не только теоретически проще, но и гораздо быстрее, чем нахождение целых точек на подходящей кривой Морделла (как предлагалось в [14], см. раздел 6.1).

в) Диофантово уравнение

$$z = y^4 + 8Hy^3 - 12y^2 + 4$$

(см. [15]) заменой

$$z = x + y^2 + 4Hy - 8H^2 - 6$$

приводится к виду (1), где

$$f(x, y) = 2xy^2 + \dots$$

(см. в других обозначениях пример 1.1 из [13]). Оптимизированный алгоритм из [9] сводит решение уравнения (1) с такой левой частью к решению порядка $|H|^2$ квадратных уравнений. Но и этот результат можно улучшить, понизив порядок числа решаемых квадратных уравнений до $|H|^{3/2}$. Для сравнения отметим, что стандартный алгоритм решения (см. [7]) потребовал бы решения порядка $|H|^3$ квадратных уравнений.

В настоящей статье рассматривается новое семейство диофантовых уравнений (1) 4-й степени с левой частью вида

$$f(x, y) = x(ax^3 + bx^2y + cxy^2 + dy^3) + xg(x, y) + h(y), \quad (3)$$

где, в свою очередь,

$$g(x, y) = p_0x^2 + (p_1y + p_2)x + p_3y^2 + p_4y + p_5, \\ h(y) = Ay^2 + By + C.$$

Коэффициент d предполагается ненулевым, что и гарантирует применимость метода Рунге. Как по-

казано в [11], произвольное уравнение со старшей однородной частью

$$f_4(x, y) = (a_1x + b_1y)(a_2x^3 + b_2x^2y + c_2xy^2 + d_2y^3)$$

сводится к уравнению с левой частью (3).

Статья организована следующим образом.

В разделе 2 предлагается алгоритм решения уравнения (1) с левой частью (3). Корректность его работы гарантируется теоремой 1. Технически данный алгоритм несколько отличается от подобных алгоритмов (см. [9, 12]), так как требует решения некоторого количества уравнений 4-й степени с одним неизвестным в целых числах. Это необходимо принимать во внимание, если мы хотим корректно оценить сложность алгоритма. Как и в работе [13], с этой целью вводится дополнительный параметр – *весовой коэффициент*, который зависит от той системы компьютерной алгебры, в которой мы планируем реализовывать алгоритм (PARI/GP, см. [1]). Далее можно оптимизировать алгоритм уже известным способом (см. [9, 13]). К сожалению, многочисленные технические детали на этом пути пока не преодолены, поэтому мы приводим только один пример такой оптимизации.

В разделе 3 делаются некоторые замечания относительно полученных результатов. Кроме того, обсуждаются дальнейшие перспективы применения элементарной версии метода Рунге для решения оставшихся типов диофантовых уравнений 4-й степени.

2. АЛГОРИТМ

Будем считать $A \neq 0$ (в случае $A = 0$ предлагаемый метод можно упростить, см. раздел 3). Напомним, что $d \neq 0$.

Предполагая $x \neq 0$, рассмотрим число

$$l = \frac{h(y)}{x}.$$

Очевидно, что l должно быть целым для каждого решения $(x, y) \in \mathbb{Z}^2$ с $x \neq 0$. Поделив на x обе части уравнения, получим

$$ax^3 + bx^2y + cxy^2 + dy^3 + g(x, y) + l = 0.$$

Это равенство влечет сравнение

$$dy^3 + p_3y^2 + p_4y + p_5 + l \equiv 0 \pmod{x}$$

в кольце \mathbb{Z} . Далее имеем

$$A^2(dy^3 + p_3y^2 + p_4y + p_5) \equiv B_1y + C_1 \pmod{h(y)}$$

для некоторых целых чисел B_1 и C_1 (это сравнение в кольце $\mathbb{Z}[y]$). А именно:

$$\begin{aligned} B_1 &= p_4A^2 - (p_3B + dC)A + dB^2, \\ C_1 &= p_5A^2 - p_3AC + dBC. \end{aligned} \tag{4}$$

Ввиду того, что $h(y) \equiv 0 \pmod{x}$, получим еще одно сравнение $A^2l + B_1y + C_1 \equiv 0 \pmod{x}$ (оба сравнения в кольце \mathbb{Z}). Наконец, положим

$$k = \frac{A^2l + B_1y + C_1}{x} = \frac{A^3y^2 + B_1xy + A^2By + C_1x + A^2C}{x^2}.$$

Ясно, что k также должно быть целым числом. Таким образом, приходим к следующему результату.

Теорема 1. Пусть $(x, y) \in \mathbb{Z}^2$ – произвольное решение уравнения с $x \neq 0$. Тогда число

$$k = \frac{A^3y^2 + B_1xy + A^2By + C_1x + A^2C}{x^2} \tag{5}$$

является целым. Здесь B_1 и C_1 определяются равенствами (4).

Отметим, что формальное доказательство теоремы 1 легко получить, опираясь на символичные вычисления в какой-нибудь системе компьютерной алгебры. Выразим из уравнения коэффициент C :

$$C = -x(ax^3 + bx^2y + cxy^2 + dy^3) - xg(x, y) - Ay^2 - By.$$

Далее подставим это выражение вместо C в правую часть (5). После сокращения на x^2 мы получим явное (но довольно громоздкое) выражение для k в виде многочлена из $\mathbb{Z}[x, y]$. Теперь очевидно, что значение k должно быть целым, поскольку $(x, y) \in \mathbb{Z}^2$. Отметим, что подобные “механические” рассуждения очень просты логически и актуальны всякий раз, когда мы хотим верифицировать сложное “синтетическое” доказательство или быстро удостовериться в справедливости гипотезы (см., например, [16]).

Пример 2. Для уравнения с левой частью

$$f(x, y) = x(y^3 - 2x^3) + x^3 + y^2 + 1$$

описанная выше процедура приводит к следующему результату:

$$\begin{aligned} k &= \frac{y^2 - xy + 1}{x^2} = \\ &= y^4 - (2x^3 - x^2)y + 2x^2 - x. \end{aligned}$$

Фактически речь идет о равенстве

$$\begin{aligned} y^2 - xy + 1 &= \\ = x^2(y^4 - (2x^3 - x^2)y + 2x^2 - x) \end{aligned}$$

в кольце классов вычетов $\mathbb{Z}[x, y]/\langle f(x, y) \rangle$.

Дальнейшие рассуждения базируются на следующей идее. Предположим, что многочлен

$$\phi(t) = a + bt + ct^2 + dt^3 \in \mathbb{Z}[t]$$

неприводим. Пусть α – любой вещественный корень $\phi(t)$. Для соответствующей ветви $y = \Psi(x)$ алгебраической функции, определяемой уравнением, имеет место оценка

$$\Psi(x) \sim \alpha x, \quad x \rightarrow \infty.$$

Как следствие, функция

$$\begin{aligned} R(x) &= \\ &= \frac{A^3\Psi^2(x) + B_1x\Psi(x) + A^2B\Psi(x) + C_1x + A^2C}{x^2} \end{aligned}$$

при $x \rightarrow \infty$ имеет пределом число

$$L(\alpha) = A^3\alpha^2 + B_1\alpha.$$

Таким образом, для любого числа $m > 0$ найдется такое число $Q_\alpha(m) > 0$, что

$$|R(x) - L(\alpha)| < m$$

для всех x с условием $|x| > Q_\alpha(m)$. Теперь можно предложить следующий алгоритм для решения уравнения (1) с левой частью (3).

Алгоритм решения. Для всех вещественных корней α многочлена $\phi(t)$ сделать:

1. Выбрать $m > 0$ и найти $Q_\alpha(m)$.
2. Для всех целых чисел x с условием

$$|x| \leq Q_\alpha(m)$$

решить уравнение $f(x, y) = 0$ как кубическое относительно y в целых числах. Найденные пары $(x, y) \in \mathbb{Z}^2$ добавить в множество решений.

3. Для всех целых чисел k с условием

$$|k - L(\alpha)| < m$$

найти все пары $(x, y) \in \mathbb{Z}^2$, удовлетворяющие системе уравнений

$$\begin{aligned} f(x, y) &= 0, \\ A^3y^2 + B_1xy + A^2By + C_1x + A^2C &= kx^2, \end{aligned} \quad (6)$$

и добавить их в множество решений.

Первая проблема в реализации нашего алгоритма состоит в том, что необходимо задать $Q_\alpha(m)$ как явную функцию *управляющего параметра* m . Эта проблема технически довольно сложна и пока полностью не решена (мы кратко обсудим это в разделе 3).

Вторую проблему можно сформулировать следующим образом: как выбрать оптимальное значение m ? Более формально, для каждого фиксированного α мы хотим минимизировать *функцию затрат* вида

$$\text{cost}(m) = 2qm + 2Q_\alpha(m), \quad (7)$$

где весовой коэффициент q предлагается определять экспериментально – в зависимости от той системы компьютерной алгебры, в которой предполагается реализовывать алгоритм (PARI/GP). Более точно, в качестве q следует взять отношение сложности решения систем уравнений вида (6) к сложности решения кубических уравнений (в обоих случаях – в целых числах).

Рассмотрим систему уравнений (6) более подробно. Исключив y , мы получим уравнение 4-й степени относительно x вида

$$K_0x^4 + K_1x^3 + K_2x^2 + K_3x + K_4 = 0, \quad (8)$$

где $K_j = K_j(k)$ – многочлены от k с целыми коэффициентами (мы не приводим их явных и довольно громоздких выражений, отметим лишь, что $\deg K_0(k) = 3$ и $\deg K_j(k) \leq 2$ для остальных j). Таким образом, необходимо определить, насколько более сложной является проблема решения уравнений четвертой степени в целых числах по сравнению с аналогичной проблемой для кубических уравнений. В PARI/GP мы намерены решать обе проблемы с помощью функции `nfroots`, которая позволяет, в частности, находить все рациональные корни.

нальные корни многочлена от одной переменной с целыми коэффициентами. Как итог, весовой коэффициент q может быть выбран с помощью компьютерных экспериментов и предварительного анализа возможной высоты многочлена в левой части равенства (8).

К сожалению, ожидаемое аналитическое выражение для $Q_\alpha(m)$ (см. раздел 3) не позволяет минимизировать функцию затрат (7) символьными методами. Обозначим через m^* значение m , доставляющее глобальный минимум $\text{cost}(m)$. Имеет смысл сосредоточиться на “разумных” оценках для m^* и $\text{cost}(m^*)$ – тогда мы сможем оценить теоретическую сложность так называемого *оптимизированного алгоритма* (алгоритма, в котором $m = m^*$). Практически мы можем найти m^* с помощью любого приближенного метода (при этом желательно решить проблему локализации для m^* аналитически).

Приведем один пример с целью продемонстрировать все трудности, которые необходимо преодолеть в общем случае.

Пример 3. Сконструируем “вручную” оптимизированный алгоритм для семейства уравнений

$$x(y^3 - 2x^3) + x^3 + y^2 + H = 0, \quad (9)$$

где $H > 0$. Единственный вещественный корень $\phi(t) = t^3 - 2$ есть $\alpha = 2^{1/3}$. Тогда

$$L(\alpha) = 2^{2/3} - 2^{1/3}H = l(H).$$

Кроме того, мы имеем

$$k = \frac{y^2 - Hxy + H}{x^2}.$$

Коэффициенты уравнения (8) таковы:

$$\begin{aligned} K_0 &= k^3 + 6Hk + 2H^3 - 4, \\ K_1 &= -3Hk - H^3 + 4, \\ K_2 &= -4Hk^2 + 4k - 4H^2 - 1, \\ K_3 &= -2k + 2H^2, \\ K_4 &= -k^2 + 2H^2k - H^4. \end{aligned}$$

Для $y = \Psi(x)$ можно доказать следующее: если $|x| > H^{1/2}$, то справедлива оценка

$$\left| \frac{y}{x} - 2^{1/3} \right| < \frac{1}{|x|}. \quad (10)$$

С помощью (10) нетрудно получить требуемую оценку для $|k - l(H)|$. А именно,

$$|k - l(H)| < H^{1/2} + 6$$

для тех x , для которых $|x| > H^{1/2}$. Поэтому, положив $m = H^{1/2}$, получим

$$\text{cost}(m) \asymp H^{1/2}, \quad H \rightarrow \infty.$$

Поскольку при $x = H^{1/2}$ имеем

$$|k - l(H)| \asymp H^{1/2}, \quad H \rightarrow \infty,$$

оценка для $\text{cost}(m^*)$ имеет такой же порядок по H . Тем самым оптимизированный алгоритм для семейства уравнений (9) построен.

3. ЗАМЕЧАНИЯ

При $A = 0$ предложенный алгоритм будет работать быстрее, чем в общем случае. Действительно, тогда второе уравнение в (6) сводится к линейному, так что уравнение (8) становится кубическим по x (но по-прежнему остается кубическим по k).

Конечно, узкое место нашего алгоритма – это получение явного символьного выражения для $Q_\alpha(m)$. Очевидный подход (успешно примененный в [9]) состоит в том, чтобы решить уравнение (8) как кубическое относительно k по формуле Кардано. Однако из-за громоздких выражений для коэффициентов такой подход представляется непродуктивным. Более реалистичный путь – решить исходное уравнение (1) как кубическое по y и далее явно оценить разность $y/x - \alpha$ при $x \rightarrow \infty$. Иными словами, мы можем конкретизировать теоретическую оценку

$$\frac{y}{x} - \alpha = \frac{\Psi(x)}{x} - \alpha = O\left(\frac{1}{x}\right), \quad x \rightarrow \infty,$$

подобно тому, как это было сделано в примере 3 (т. е. получить оценку типа (10) для всех достаточно больших x , причем с указанием явной границы для этих x). С этой целью можно рассмотреть несколько первых членов степенного разложения для функции $\Psi(x)/x$ при $x \rightarrow \infty$. В общем случае, помимо (очень осторожного) обращения к рядам Пюизо, можно воспользоваться алгоритмами степенной геометрии, хорошо себя зарекомендовавшими при разрешении алгебраических особенностей (см. [17, 18]).

Чтобы проиллюстрировать возможные трудности, вновь обратимся к примеру 3. Имеем

$$\frac{\Psi(x)}{x} - 2^{1/3} = -\frac{2^{1/3}}{6x} + O\left(\frac{1}{x^2}\right), \quad x \rightarrow \infty.$$

Здесь может показаться, что существует такая абсолютная константа M , что для всякого $x \neq 0$ верна оценка

$$\left| \frac{y}{x} - 2^{1/3} \right| < \frac{M}{|x|}. \quad (11)$$

Однако это не так — из-за того, что коэффициенты разложения зависят от параметра H , который может быть сколь угодно большим. В самом деле, если $x = 1$, то, как можно убедиться, $y \sim -H^{1/3}$ при $H \rightarrow \infty$. Таким образом, в общем случае нам следует указать явную нижнюю границу для $|x|$, которая гарантировала бы оценку типа (11) для $y/x - \alpha$. В конкретной ситуации (пример 3) такую границу указать легко.

В заключение скажем несколько слов о дальнейших перспективах элементарной версии метода Рунге, предложенной в [11].

По-видимому, самым нетривиальным в плане алгоритмической реализации является последний неисследованный случай, когда старшая однородная часть уравнения (1) допускает разложение вида

$$f_4(x, y) = (a_1x^2 + b_1xy + c_1y^2)(a_2x^2 + b_2xy + c_2y^2).$$

Обнадёживающим моментом можно считать то, что принципиально новых проблем в реализации здесь не предвидится, а те проблемы, что описаны выше, могут быть решены.

Работа поддержана Красноярским математическим центром, финансируемым Минобрнауки РФ в рамках мероприятий по созданию и развитию региональных НОМЦ (Соглашение 075-02-2020-1534/1).

СПИСОК ЛИТЕРАТУРЫ

1. <https://pari.math.u-bordeaux.fr>
2. *Mordell L.J.* Diophantine equations. London: Academic Press Inc., 1969.
3. *Спринджук В.Г.* Классические диофантовы уравнения от двух неизвестных. М.: Наука, 1982.
4. *Masser D.W.* Auxiliary Polynomials in Number Theory. Cambridge University Press, 2016.
5. *Runge C.* Ueber ganzzahlige Lösungen von Gleichungen zwischen zwei Veränderlichen // J. reine und angew. Math. 1887. V. 100. P. 425–435.
6. *Walsh P.G.* A quantitative version of Runge's theorem on diophantine equations // Acta Arith. 1992. V. 62. P. 157–172.
7. *Poulakis D.* A simple method for solving the diophantine equation $Y^2 = X^4 + aX^3 + bX^2 + cX + d$ // Elem. Math. 1999. V. 54. P. 32–36.
8. *Tengely Sz.* On the Diophantine equation $F(x) = G(y)$ // Acta Arith. 2003. V. 110. P. 185–200.
9. *Osipov N.N., Gulnova B.V.* An algorithmic implementation of Runge's method for cubic diophantine equations // J. Sib. Fed. Univ. Math. Phys. 2018. V. 11. 2. P. 137–147.
10. *Beukers F., Tengely Sz.* An implementation of Runge's method for diophantine equations // Preprint <https://arxiv.org/abs/math/0512418v1>
11. *Осипов Н.Н.* Метод Рунге для уравнений 4-й степени: элементарный подход // Математическое просвещение. Сер. 3. Вып. 19. М.: МЦНМО, 2015. С. 178–198.
12. *Osipov N.N., Medvedeva M.I.* An elementary algorithm for solving a diophantine equation of degree four with Runge's condition // J. Sib. Fed. Univ. Math. Phys. 2019. V. 12. № 3. P. 331–341.
13. *Osipov N.N., Dalinkevich S.D.* An algorithm for solving a quartic diophantine equation satisfying Runge's condition // Lecture Notes in Computer Science. V. 11661. Springer, 2019. P. 377–392.
14. *Stroeker R.J., de Weger B.M.M.* Solving elliptic diophantine equations: the general cubic case // Acta Arith. 1999. V. 87. P. 339–365.
15. *Masser D.W.* Polynomial Bounds for Diophantine Equations // Amer. Math. Monthly. 1986. V. 93. P. 486–488.
16. *Осипов Н.Н.* О механическом доказательстве планиметрических теорем рационального типа // Программирование. 2014. № 2. С. 41–50.
17. *Брюно А.Д.* Степенная геометрия в алгебраических и дифференциальных уравнениях. М.: Физматлит, 1998.
18. *Брюно А.Д., Бахтин А.Б.* Разрешение алгебраической сингулярности алгоритмами степенной геометрии // Программирование. 2012. № 2. С. 11–28.

ПОСТРОЕНИЕ ЧАСТИЧНЫХ ЛОРАНОВЫХ РЕШЕНИЙ УСЕЧЕННЫХ
ДИФФЕРЕНЦИАЛЬНЫХ СИСТЕМ© 2021 г. А. А. Панфёров^{a,b,*}^a Факультет вычислительной математики и кибернетики МГУ им. М.В. Ломоносова
119991, Москва, Воробьевы горы, Россия^b Федеральный исследовательский центр “Информатика и управление” РАН
119333 Москва, ул. Вавилова, 40, Россия

*E-mail: ast.a_s@mail.ru

Поступила в редакцию 01.08.2020 г.

После доработки 26.08.2020 г.

Принята к публикации 14.09.2020 г.

Рассматривается задача построения начальных членов формальных лорановых рядов, являющихся решениями для заданной компоненты y_k ($1 \leq k \leq m$) вектора неизвестных у дифференциальной системы $y' = Ay$, где $y = (y_1, \dots, y_m)^T$, $A - m \times m$ -матрица, элементами которой являются d -усечения формальных лорановых рядов, т.е. их начальные члены до степени $d \geq 0$ включительно. Предлагается алгоритм решения задачи с использованием алгоритма TSLS (Truncated Series Laurent Solution). Строящийся предлагаемым алгоритмом первые члены формальных лорановых решений для y_k являются инвариантными относительно возможных продолжений элементов матрицы исходной системы.

DOI: 10.31857/S0132347421010088

1. ВВЕДЕНИЕ

Дифференциальные системы вида

$$y' = Ay, \quad (1)$$

где A – матрица размера $m \times m$, а $y = (y_1, \dots, y_m)^T$ – вектор неизвестных, которые часто называют *нормальными* дифференциальными системами, находят широкое применение в различных областях математики. Под решением системы (1) понимается вектор y из m компонент. К настоящему времени известно много алгоритмов построения решений дифференциальных систем разного вида, многие из которых реализованы и доступны в современных системах компьютерной алгебры. В некоторых прикладных задачах интерес могут представлять не все компоненты вектора неизвестных, а только их часть. В этом случае целесообразна постановка задачи *частичного* решения системы, т.е. построение некоторых заранее выбранных компонент вектора решений. Методы построения частичных решений также известны (см., например, [1]).

Отдельный интерес представляют системы, коэффициентами которых являются формальные ряды (степенные или лорановы). Поскольку ряды являются потенциально бесконечными объектами, их использование вызывает ряд трудностей при реализации в компьютерных системах символьных вычислений. При этом некоторые задачи оказываются алгоритмически неразрешимыми (см. [2, 3]). Тем не менее, методы для построения решений дифференциальных систем, коэффициентами которых являются ряды, существуют ([4]).

Ряды, используемые в качестве коэффициентов систем, могут быть заданы в приближенном виде, а именно в виде так называемых *усечений* – начальных отрезков рядов. Задачи, связанные с дифференциальными уравнениями, коэффициенты которых являются усеченными рядами, рассматриваются в некоторых работах ([3, 5]).

В настоящей работе также будут рассматриваться дифференциальные системы вида (1), коэффициентами которых являются усечения формальных лорановых рядов. Нас будет интересовать задача построения частичного решения, т.е.

построение некоторой заранее заданной компоненты решения y_k ($1 \leq k \leq m$), в виде такого усечения лоранового ряда, которое не зависит от возможных продолжений коэффициентов системы.

В разделе 2 приводятся обозначения и определения основных понятий, используемых в работе. В разделе 3 приводится постановка задачи и общая схема решения. Описание предлагаемого алгоритма решения поставленной задачи приведено в подразделе 4.4, некоторые детали его реализации в системе Maple – в подразделе 4.5.

2. ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ

Пусть K – поле характеристики 0, под которым будет пониматься некоторое числовое поле, например, поле рациональных чисел \mathbb{Q} . Также будем использовать стандартное обозначение $K[x]$ для кольца многочленов переменной x с коэффициентами из K : если p – ненулевой многочлен из $K[x]$, то $p = a_0 + a_1x + \dots + a_nx^n = \sum_{i=0}^n a_i x^i$, где $a_i \in K$ ($i = 0, 1, \dots, n$), $a_n \neq 0$, $\deg p = n$ – степень многочлена p ; при этом $\deg 0 = -\infty$.

Наряду с обычными многочленами из $K[x]$, будем также рассматривать *многочлены Лорана* (лорановы многочлены) из кольца $K[x, x^{-1}]$: если p – ненулевой многочлен Лорана, то $p = \sum_{i=m}^n a_i x^i$, где $m, n \in \mathbb{Z}$, $m \leq n$, $a_i \in K$ ($m \leq i \leq n$), $a_m \neq 0$, $a_n \neq 0$, $\deg p = n$.

Кольца $K[x]$ и $K[x, x^{-1}]$ будем рассматривать как дифференциальные кольца с обычным дифференцированием $' = \frac{d}{dx}$, для которых K будет являться полем констант.

Для кольца квадратных матриц размера $m \times m$ над кольцом R будем использовать стандартное обозначение $\text{Mat}_m(R)$. Пусть $A = (a_{ij}) \in \text{Mat}_m(K[x, x^{-1}])$, тогда $\deg A = \max_{1 \leq i, j \leq m} \deg a_{ij}$.

Для кольца формальных степенных рядов и для кольца формальных лорановых рядов с коэффициентами из K будем использовать стандартные обозначения $K[[x]]$ и $K((x))$ соответственно. Для ненулевого элемента $a = \sum a_i x^i \in K((x))$ его *валуация* $\text{val} a$ определяется как $\min\{i \mid a_i \neq 0\}$, при этом $\text{val} 0 = \infty$. Пусть $A = (a_{ij}) \in \text{Mat}_m(K((x)))$, тогда $\text{val} A = \min_{1 \leq i, j \leq m} \text{val} a_{ij}$.

Пусть $a \in K((x))$, $d \in \mathbb{Z} \cup \{-\infty\}$, тогда d -усечение $a^{(d)}$ получается отбрасыванием всех членов a

степени большей, чем d ; если $d = -\infty$, то $a^{(d)} = 0$. Пусть $B = (b_{ij}) \in \text{Mat}_m(K((x)))$, тогда d -усечением матрицы B назовем такую матрицу $A = (a_{ij}) \in \text{Mat}_m(K[x, x^{-1}])$, элементы которой являются d -усечениями соответствующих элементов матрицы B : $a_{ij} = b_{ij}^{(d)}$; d -усечение матрицы B будем обозначать $B^{(d)}$.

Пусть $A = (a_{ij}) = B^{(d)}$ для некоторой матрицы $B \in \text{Mat}_m(K((x)))$. Матрицу $\tilde{A} = (\tilde{a}_{ij}) \in \text{Mat}_m(K((x)))$ будем называть *продолжением* матрицы A , если для всех $1 \leq i, j \leq m$ выполняется $a_{ij} - \tilde{a}_{ij} = O(x^{d+1})$, т.е. $\text{val}(a_{ij} - \tilde{a}_{ij}) > d$.

Дифференциальным модулем M называется m -мерное линейное пространство над K , на котором определено аддитивное отображение $\Delta: M \rightarrow M$, удовлетворяющее следующему свойству (см. [6]): $\Delta(fv) = f'v + f\Delta v$ для всех $f \in K$ и $v \in M$. Отображение Δ будем называть *дифференцированием в M* .

3. ПОСТАНОВКА ЗАДАЧИ

Будем рассматривать систему (1), ненулевая матрица A которой является d -усечением некоторой неизвестной матрицы $B \in \text{Mat}_m(K((x)))$: $A = B^{(d)}$. Нас будут интересовать частичные лорановы решения системы (1), т.е. решения в виде лорановых рядов для некоторых заранее фиксированных компонент вектора неизвестных. При этом нас не будет интересовать вопрос построения всех коэффициентов таких решений (если они есть), а возможность определения максимального количества начальных членов лорановых рядов, которые являются инвариантными относительно возможных продолжений A , а также вычисление коэффициентов этих членов.

Для простоты задачу частичного решения будем рассматривать в постановке определения одной компоненты решений y_k ($1 \leq k \leq m$). Таким образом задача сводится к определению, существуют ли решения исходной системы (1), k -я компонента которых может быть представлена в виде лоранового ряда вне зависимости от продолжения коэффициентов системы, и если ответ положительный, то требуется определить максимальное число первых членов такого ряда, не зависящих от возможных продолжений элементов матрицы A .

Для решения поставленной задачи сведем ее к задаче построения лорановых решений скалярных

линейных дифференциальных уравнений с коэффициентами в виде усеченных степенных рядов, которая подробно рассмотрена в работе [5]. Там же предложен алгоритм, который для краткости будем называть TSLS (Truncated Series Laurent Solution). Для решения задачи с использованием TSLS алгоритма потребуется:

1. для заданной компоненты y_k исходной системы (1) построить скалярное дифференциальное уравнение, позволяющее строить решения k -й компоненты исходной усеченной системы;
2. представить коэффициенты построенного скалярного уравнения в виде усеченных степенных рядов, приняв во внимание усеченность коэффициентов исходной системы;
3. применить алгоритм TSLS для построения начальных элементов лорановых решений для компоненты y_k исходной системы.

3.1. Алгоритм TSLS

В работе [5] авторами подробно рассмотрена задача построения лорановых решений скалярных дифференциальных уравнений вида

$$\sum_{i=0}^m (a_i + O(x^{t_i+1}))\theta^i y(x) = 0, \quad (2)$$

где $a_i \in K[x]$, неотрицательные целые $t_i \geq \deg a_i$, $\theta = x \frac{d}{dx}$. Числа t_i являются степенями усечения коэффициентов: $a_i = b_i^{(t_i)}$ для некоторого $b_i \in K[[x]]$.

Алгоритм TSLS (подробности см. [5]) по уравнению (2) строит индуцированное рекуррентное уравнение

$$u_0(n)c_n + u_{-1}(n)c_{n-1} + \dots = 0 \quad (3)$$

такое, что уравнение (2) имеет лораново решение $y(x) = c_v x^v + c_{v+1} x^{v+1} + \dots$ тогда и только тогда, когда двусторонняя последовательность

$$\dots, 0, 0, c_v, c_{v+1}, \dots$$

удовлетворяет уравнению (3), т.е.

$$\begin{aligned} u_0(v)c_v &= 0, \\ u_0(v+1)c_{v+1} + u_{-1}(v+1)c_v &= 0, \\ u_0(v+2)c_{v+2} + u_{-1}(v+2)c_{v+1} + u_{-2}(v+2)c_v &= 0, \\ \dots & \end{aligned}$$

где каждый из многочленов $u_{-k}(n) \in K[n]$ ($k = 0, 1, \dots$) строится по коэффициентам при степенях x^k в $a_i(x)$ ($0 \leq i \leq r$) и имеет степень, меньшую или

равную r ; коэффициент $u_0(n)$ играет роль *определяющего многочлена* — множество его целых корней содержит все возможные валюации лорановых решений. Для каждой возможной валюации выполняется попытка вычислить начальные коэффициенты лоранового решения. Анализ получающихся значений коэффициентов лорановых решений позволяет отфильтровать только те валюации, решения с которыми существуют и не зависят от возможных продолжений коэффициентов исходного уравнения (2), а также построить максимальное число начальных членов лорановых решений, инвариантных относительно продолжения коэффициентов исходного уравнения.

4. ЧАСТИЧНЫЕ ЛОРАНОВЫ РЕШЕНИЯ

Для применения алгоритма TSLS необходимо по исходной системе (1) построить скалярное уравнение вида (2), позволяющее строить решения для компоненты y_k . Также потребуется определить степени усечения коэффициентов построенного скалярного уравнения, чтобы учесть степени усечения коэффициентов исходной системы.

4.1. Построение скалярного уравнения

Задача решения нормальных дифференциальных систем относительно части неизвестных подробно рассмотрена в [1], где приводится алгоритм (АВ-алгоритм), который в частности позволяет построить по заданной системе для фиксированной неизвестной y_k ($1 \leq k \leq m$) скалярное дифференциальное уравнение, все решения которого и только они являются первыми компонентами всех решений исходной системы (см. [1, предл. 2]). Будем использовать метод, основанный на АВ-алгоритме.

Чтобы скалярное дифференциальное уравнение, строящееся АВ-алгоритмом, использовало в качестве дифференцирования θ , будет удобно перейти от системы (1) к системе

$$\theta y = \hat{A}y,$$

где матрица системы $\hat{A} = xA$. Рассмотрим дифференциальный модуль $K[x]^m$ с дифференцированием $\Delta_{xA} : v \rightarrow \theta v + xA^T v$. Определим матрицу $[\Delta_{xA} v] \in \text{Mat}_m(K[x, x^{-1}])$, столбцы которой $[\Delta_{xA} v]_{*,i} = \Delta_{xA}^{i-1} v$, $1 \leq i \leq m$, считая $\Delta_{xA}^0 v = v$.

Возьмем вектор $e_k = (0, \dots, 0, 1, 0, \dots, 0)^T \in K^m$, в котором “1” стоит в k -й позиции. Пусть $\det[\Delta_{xA}e_k] \neq 0$, тогда коэффициенты скалярного уравнения

$$a_m \theta^m y_k + a_{m-1} \theta^{m-1} y_k + \dots + a_0 y_k = 0, \quad (4)$$

строящегося АВ-алгоритмом, могут быть вычислены по формулам:

$$\begin{aligned} a_m &= \det[\Delta_{xA}e_k] \\ a_{i-1} &= -\det([\Delta_{xA}e_k]_{(i)}), \quad i = 1, 2, \dots, m. \end{aligned} \quad (5)$$

Здесь $[\Delta_{xA}e_k]_{(i)}$ – матрица, получающаяся из $[\Delta_{xA}e_k]$ заменой i -го столбца вектором $\Delta_{xA}^m e_k$.

Корректность формул (5) следует непосредственно из корректности АВ-алгоритма (см. [1]). Решениями уравнения (4) являются k -е компоненты решений системы (1) в произвольном дифференциальном расширении основного поля.

Поскольку элементами матрицы A являются многочлены Лорана, в случае $\text{val}A < -1$ коэффициенты a_i ($i = 0, 1, \dots, m$), вычисляемые по формулам (5), в общем случае являются многочленами Лорана. Чтобы гарантировано получить скалярное уравнение, коэффициенты которого являются многочленами, в случае $\text{val}A < -1$ домножим каждый из коэффициентов на $x^{qm(m+1)/2}$, где $q = -\text{val}A - 1$:

$$\begin{aligned} a_m &= x^{qm(m+1)/2} \det[\Delta_{xA}e_k] \\ a_{i-1} &= -x^{qm(m+1)/2} \det([\Delta_{xA}e_k]_{(i)}), \quad i = 1, 2, \dots, m. \end{aligned} \quad (6)$$

Замечание 1. При вычислении коэффициентов скалярного уравнения по формулам (5) или (6) возможно появление общих множителей. Для дальнейшего будет важно, что сокращение на общие множители не производится.

4.2. Оценки коэффициентов скалярного уравнения

Поскольку коэффициенты системы (1) являются усечениями лорановых рядов, мы не обладаем всей полнотой информации о системе. Различные продолжения коэффициентов системы могут влиять на строящиеся решения. Строящиеся коэффициенты скалярного уравнения для компоненты y_k , являющиеся многочленами, также нужно рассматривать как усечения рядов. На следующем этапе необходимо установить соответствие степеней усечения коэффициентов скалярного уравнения со степенями усечения коэффициентов исходной системы.

Определим $A^{(i_0, j_0)}$ – как матрицу, полученную продолжением (i_0, j_0) -го элемента A , тогда ее (i, j) -й элемент будет выражаться следующим образом:

$$A_{i,j}^{(i_0, j_0)} = A_{i,j} + \begin{cases} \alpha \cdot x^{d+1}, & \text{если } i = i_0 \text{ и } j = j_0 \\ 0, & \text{иначе} \end{cases}$$

где $d = \text{deg}A$, α – символьный коэффициент (константа). Матрицу $A^{(i_0, j_0)}$ будем называть *одноэлементным* (i_0, j_0) -*продолжением* A .

Поскольку формулы вычисления коэффициентов скалярного дифференциального уравнения для неизвестной y_k различаются в зависимости от значения $q = -\text{val}(xA) = -\text{val}A - 1$, рассмотрим случаи $q \leq 0$ и $q > 0$ отдельно.

Предложение 1. Пусть $1 \leq k \leq m$, $A \in \text{Mat}_m(K[x, x^{-1}])$, такая, что $\det[\Delta_{xA}e_k] \neq 0$, $d = \text{deg}A$, $q = -\text{val}A - 1 \leq 0$, a_i ($0 \leq i \leq m$) – коэффициенты скалярного дифференциального уравнения, вычисленные по формулам (5). Пусть $\tilde{A} = A + x^{d+1}B$, где $B \in \text{Mat}_m(K[x])$ – произвольная матрица такая, что $\det[\Delta_{x\tilde{A}}e_k] \neq 0$, и \tilde{a}_i ($0 \leq i \leq m$) – коэффициенты скалярного дифференциального уравнения, вычисленные по формулам (5) для матрицы \tilde{A} .

Тогда:

1) для $1 < i \leq m$: $a_i - \tilde{a}_i = O(x^{r_1})$, где $r_1 = d + 2 - q(m - 2)$;

2) $a_0 - \tilde{a}_0 = O(x^{r_2})$, где $r_2 = d + 2 - q(m - 1)$.

Доказательство. Достаточно рассмотреть одноэлементное продолжение $A^{(i_0, j_0)}$. Обозначим p_j ($j = 0, 1, \dots, m$) – минимальную степень x среди элементов вектора $\Delta_{(xA)^k}^{(i_0, j_0)} e_k$, в коэффициенте при которой появляется символьный коэффициент α ; если α вообще не появляется ни в одном из элементов вектора $\Delta_{(xA)^j}^{(i_0, j_0)} e_k$, то положим $p_j = \infty$. Через v_j ($j = 0, 1, \dots, m$) обозначим минимальную валюацию элементов вектора $\Delta_{(xA)^j}^{(i_0, j_0)} e_k$. Для p_j и v_j верны следующие оценки, непосредственно следующие из определения $\Delta_{(xA)^j}^{(i_0, j_0)}$: $p_0 = v_0 = 0$; для $j \geq 1$

$$\begin{aligned} p_j &\geq d + 2, \\ v_j &\geq -q. \end{aligned} \quad (7)$$

Символьный коэффициент α будет присутствовать в определителях, участвующих в формулах (5), в качестве коэффициента при минимальной степени x , если оценки (7) обратятся в равенства.

1) Поскольку $[\Delta_{(xA)^{(0, \cdot, 0)}} e_k]$ состоит из векторов $\Delta_{(xA)^{(0, \cdot, 0)}}^j e_k$ ($j = 0, 1, \dots, m - 1$), то минимальная степень x , в коэффициенте при которой возможно появление символьного коэффициента α в $\det[\Delta_{(xA)^{(0, \cdot, 0)}} e_k]$, определяется как

$$\min_{1 \leq j \leq m-1} \left(p_j + \sum_{\substack{1 \leq i \leq m-1 \\ i \neq j}} v_i \right) \geq d + 2 - q(m - 2).$$

Для $1 \leq n < m$ в коэффициентах a_n участвуют матрицы $[\Delta_{(xA)^{(0, \cdot, 0)}} e_k]_{(n+1)}$, отличающиеся от $[\Delta_{(xA)^{(0, \cdot, 0)}} e_k]$ одним $(n + 1)$ -м столбцом, вместо которого подставляется вектор $\Delta_{(xA)^{(0, \cdot, 0)}}^m e_k$. Минимальная степень x , в коэффициенте при которой возможно появление символьного коэффициента α в $\det([\Delta_{(xA)^{(0, \cdot, 0)}} e_k]_{(n)})$, определяется как

$$\min_{1 \leq j \leq m} \left(p_j + \sum_{\substack{1 \leq i \leq m \\ i \neq j, i \neq n}} v_i \right) \geq d + 2 - q(m - 2);$$

2) $[\Delta_{(xA)^{(0, \cdot, 0)}} e_k]_{(1)}$ отличается от $[\Delta_{(xA)^{(0, \cdot, 0)}} e_k]$ первым столбцом, в качестве которого используется вектор $\Delta_{(xA)^{(0, \cdot, 0)}}^m e_k$. Поэтому минимальная степень x , в коэффициенте при которой возможно появление символьного коэффициента α в $\det([\Delta_{(xA)^{(0, \cdot, 0)}} e_k]_{(1)})$, определяется как

$$\min_{1 \leq j \leq m} \left(p_j + \sum_{\substack{1 \leq i \leq m \\ i \neq j}} v_i \right) \geq d + 2 - q(m - 1). \quad \square$$

Предложение 2. Пусть $1 \leq k \leq m$, $A \in \text{Mat}_m(K[x, x^{-1}])$, такая, что $\det[\Delta_{xA} e_k] \neq 0$, $d = \deg A$, $q = -\text{val} A - 1 > 0$, a_i ($0 \leq i \leq m$) – коэффициенты скалярного дифференциального уравнения, вычисленные по формулам (6). Пусть $\tilde{A} = A + x^{d+1} B$, где $B \in \text{Mat}_m(K[x])$ – произвольная матрица такая, что $\det[\Delta_{x\tilde{A}} e_k] \neq 0$, и \tilde{a}_i ($0 \leq i \leq m$) – коэффициенты скалярного дифференциального уравнения, вычисленные по формулам (6) для матрицы \tilde{A} .

Тогда:

- 1) $a_m - \tilde{a}_m = O(x^{r_1})$, где $r_1 = d + 2 + q(m + 1)$;
- 2) для $1 < i < m$: $a_i - \tilde{a}_i = O(x^{r_2})$, где $r_2 = d + 2 + 2q$;
- 3) $a_0 - \tilde{a}_0 = O(x^{r_2})$, где $r_2 = d + 2 + q$.

Доказательство отличается от доказательства предложения 1 оценками величин p_j, v_j . В случае $q > 0$ для $j \geq 1$ будут справедливы следующие неравенства:

$$p_j \geq d + 2 - q(j - 1), \\ v_j \geq -qj.$$

1) Поскольку $[\Delta_{(xA)^{(0, \cdot, 0)}} e_k]$ состоит из векторов $\Delta_{(xA)^{(0, \cdot, 0)}}^i e_k$ ($i = 0, 1, \dots, m - 1$), то минимальная степень x , в коэффициенте при которой возможно появление символьного коэффициента α в $\det[\Delta_{(xA)^{(0, \cdot, 0)}} e_k]$, определяется как

$$\min_{1 \leq j \leq m-1} \left(p_j + \sum_{\substack{1 \leq i \leq m-1 \\ i \neq j}} v_i \right) \geq d + 2 - q \sum_{i=2}^{m-1} i$$

Учитывая множитель $x^{qm(m+1)/2}$ в (6), получаем $r_1 = d + 1 - q \sum_{i=2}^{m-1} i + q \sum_{i=1}^m i = d + 1 + q(m + 1)$;

2) Для $1 \leq n < m$ в коэффициентах a_n участвуют матрицы $[\Delta_{(xA)^{(0, \cdot, 0)}} e_k]_{(n+1)}$, отличающиеся от $[\Delta_{(xA)^{(0, \cdot, 0)}} e_k]$ одним $(n + 1)$ -м столбцом, вместо которого подставляется вектор $\Delta_{(xA)^{(0, \cdot, 0)}}^m e_k$. Минимальная степень x , в коэффициенте при которой возможно появление символьного коэффициента α в $\det([\Delta_{(xA)^{(0, \cdot, 0)}} e_k]_{(n)})$, определяется как

$$\min_{1 \leq j \leq m} \left(p_j + \sum_{\substack{1 \leq i \leq m \\ i \neq j, i \neq k}} v_i \right) \geq d + 2 + q - q \sum_{i=2}^m i = \\ = d + 2 + 2q - q \sum_{i=1}^m i.$$

Учитывая множитель $x^{qm(m+1)/2}$ в (6), получаем $r_2 = d + 2 + 2q - q \sum_{i=1}^m i + q \sum_{i=1}^m i = d + 2 + 2q$;

3) $[\Delta_{(xA)^{(0, \cdot, 0)}} e_k]_{(1)}$ отличается от $[\Delta_{(xA)^{(0, \cdot, 0)}} e_k]$ первым столбцом, в качестве которого используется вектор $\Delta_{(xA)^{(0, \cdot, 0)}}^m e_k$. Поэтому минимальная степень x , в коэффициенте при которой возможно появление

символьного коэффициента α в $\det([\Delta_{(xA)^{(0,0)}}e_k]_{(1)})$, определяется как

$$\min_{1 \leq j \leq m} \left(p_j + \sum_{\substack{1 \leq i \leq m \\ i \neq j}} v_i \right) \geq d + 2 + q - q \sum_{i=1}^m i.$$

Учитывая множитель $x^{qm(m+1)/2}$ в (6), получаем $r_3 = d + 2 + q - q \sum_{i=1}^m i + q \sum_{i=1}^m i = d + 2 + q$. \square

Пример 1. Рассмотрим усеченную систему $y' = Ay$, где

$$A = \begin{bmatrix} 1 & x \\ -x & 1 \end{bmatrix},$$

$y = (y_1, y_2)^T$ и степень усечения коэффициентов $d = 1$. Построим начальный отрезок лоранового решения для компоненты y_1 , не зависящий от продолжений коэффициентов системы. Коэффициенты скалярного дифференциального уравнения для y_1 вычисляются по формулам (5), само уравнение будет иметь вид:

$$x^2 \theta^2 y_1 + (-2x^3 - 2x^2) \theta y_1 + (x^6 + x^4 + x^3) y_1 = 0. \quad (8)$$

Не все слагаемые в коэффициентах уравнения (8) являются инвариантными относительно возможных продолжений коэффициентов исходной системы. Применяя оценки из предложения 1, уравнение (8) можно переписать в виде

$$(x^2 + O(x^3)) \theta^2 y_1 + (-2x^2 + O(x^3)) \theta y_1 + (x^3 + O(x^4)) y_1 = 0,$$

после сокращения на x^2 , окончательно получаем

$$(1 + O(x)) \theta^2 y_1 + (-2 + O(x)) \theta y_1 + (x + O(x^2)) y_1 = 0. \quad (9)$$

Алгоритм TSLS, примененный к уравнению (9), построит определяющий многочлен $u_0(n) = n^2 - 2n$, откуда получаем множество возможных валюаций решений $\{0, 2\}$. Далее будет построено решение с валюацией 2: $y_1 = c_2 x^2 + O(x^3)$, где c_2 — произвольная постоянная. При этом решение с валюацией 0 построено не будет, поскольку не при всех продолжениях коэффициентов уравнения (9) оно будет иметь лораново решение с валюацией 0. В то же время $\text{val} A = 0$ и, согласно [7, Theorem 13.1], эта система должна иметь два линейно независи-

мых решения в виде степенных рядов. Поскольку $\det[\Delta_{xA} e_1] \neq 0$, два линейно независимых решения в виде степенных рядов должна иметь компонентна y_1 .

Продолжения коэффициентов построенного скалярного дифференциального уравнения (9) не являются произвольными и зависят от продолжений коэффициентов исходной системы. Чтобы учесть это, применим метод *наращивания коэффициентов*, заключающийся в следующем: добавим к коэффициентам исходной системы символьные коэффициенты до x^n ($n > d$) включительно, построив матрицу $A^{[n]}$, элементы которой $A_{ij}^{[n]} = A_{ij} + \sum_{l=d+1}^n \alpha_{ijl} x^l$, где α_{ijl} — символьные коэффициенты. Добавленные коэффициенты считаются известными, поэтому они будут использованы алгоритмом TSLS в строящихся решениях. Будем последовательно наращивать коэффициенты, пока в каждое из решений, строящихся алгоритмом TSLS, не попадут добавленные символьные коэффициенты. После этого усечем полученные решения так, чтобы в них не осталось символьных коэффициентов. Так построенные решения будут содержать максимальное число начальных членов лорановых рядов, не зависящих от возможных продолжений коэффициентов исходной усеченной системы.

Для рассматриваемого примера матрица $A^{[2]}$ будет иметь вид:

$$A^{[2]} = \begin{bmatrix} 1 + \alpha_{112} x^2 & x + \alpha_{122} x^2 \\ -x + \alpha_{212} x^2 & 1 + \alpha_{222} x^2 \end{bmatrix}.$$

Скалярное уравнение для неизвестной y_1 системы с матрицей $A^{[2]}$, коэффициенты которого вычислены по формулам (5) и усечены, используя оценки предложения 1, будет иметь вид $(x^2 + x^3 \alpha_{122} + O(x^4)) \theta^2 y_1 + (-2x^2 + (-3\alpha_{122} - 2)x^3 + O(x^4)) \theta y_1 + (x^3 + (2\alpha_{122} + 1)x^4 + O(x^5)) y_1 = 0$, применение алгоритма TSLS к этому уравнению даст два линейно независимых решения $c_1 + c_1 x + c_2 x^2 + O(x^3)$ и $c_2 x^2 + \left(c_2 + \frac{2}{3} c_2 \alpha_{122} \right) x^3 + O(x^4)$ с валюациями 0 и 2 соответственно. Поскольку в первое решение не вошли элементы с добавленными символьными коэффициентами α , нет гарантии, что первое решение содержит максимальное число слагаемых, не зависящих от возможных продолжений коэффициентов исходной системы, и нужно добавить символьные коэффициенты с x^3 , т.е. рассмотреть систему с матрицей

$$A^{[3]} = \begin{bmatrix} 1 + \alpha_{112}x^2 + \alpha_{113}x^3 & x + \alpha_{122}x^2 + \alpha_{123}x^3 \\ -x + \alpha_{212}x^2 + \alpha_{213}x^3 & 1 + \alpha_{222}x^2 + \alpha_{223}x^3 \end{bmatrix}.$$

$$\begin{bmatrix} -x - x^2 + x^3 + \alpha_{114}x^4 & -1 + \alpha_{124}x^4 & \alpha_{134}x^4 \\ -1 + \alpha_{214}x^4 & 1 + x + \alpha_{224}x^4 & x^3 + \alpha_{234}x^4 \\ x + \alpha_{314}x^4 & 1 + \alpha_{324}x^4 & 1 + \alpha_{334}x^4 \end{bmatrix}.$$

Применение TSLS алгоритма к скалярному дифференциальному уравнению с усеченными коэффициентами, строящемуся по системе с матрицей $A^{[3]}$ относительно y_1 , дает два линейно независимых решения $c_1 + c_1x + c_2x^2 + \left(\frac{1}{3}c_1\alpha_{112} - \frac{1}{3}c_1\alpha_{122} + \frac{2}{3}c_2\alpha_{122} - \frac{1}{3}c_1 + c_2\right)x^3 + O(x^4)$ и $c_2x^2 + \left(\frac{2}{3}c_2\alpha_{122} + c_2\right)x^3 + \left(\frac{2}{3}c_2\alpha_{122} + \frac{1}{2}c_2 + \frac{1}{2}c_2\alpha_{123}\right)x^4 + O(x^5)$. Отбрасывая элементы, начиная с которых в коэффициенты входят добавленные символьные коэффициенты α , окончательно получаем два линейно независимых решения с максимальным числом элементов, инвариантных относительно возможных продолжений коэффициентов исходной системы: $c_1 + c_1x + c_2x^2 + O(x^3)$ и $c_2x^2 + O(x^3)$. Заметим, что второе решение является частным случаем первого решения при $c_1 = 0$.

4.3. Вполне определенные системы

Не всегда наращивание коэффициентов позволяет решить поставленную задачу. Рассмотрим следующий пример.

Пример 2. Пусть матрица усеченной дифференциальной системы (1) с вектором неизвестных $y = (y_1, y_2, y_3)^T$ имеет вид

$$A = \begin{bmatrix} -x - x^2 + x^3 & -1 & 0 \\ -1 & 1 + x & x^3 \\ x & 1 & 1 \end{bmatrix}. \tag{10}$$

Будем интересоваться лорановыми решениями для компоненты y_1 . Скалярное уравнение для y_1 после усечения коэффициентов будет иметь вид

$$O(x^6)\theta^3 y_1 + O(x^6)\theta^2 y_1 + O(x^6)\theta y_1 + O(x^7)y_1 = 0.$$

Алгоритм TSLS не применим к уравнениям такого вида, поскольку в коэффициентах содержится слишком мало информации, из-за чего не возможно построить определяющий многочлен $u_0(n)$ для определения возможных валуаций решений. Добавление символьных коэффициентов к элементам A до степени x^4 дает $A^{[4]}$ с элементами

Скалярное уравнение для y_1 , построенное по $A^{[4]}$, после усечения коэффициентов будет иметь вид $((-4\alpha_{134} + 1)x^6 + O(x^7))\theta^3 y_1 + ((24\alpha_{134} - 6)x^6 + O(x^7))\theta^2 y_1 + ((-20\alpha_{134} + 5)x^6 + O(x^7))\theta y_1 + O(x^8)y_1 = 0$. Определяющий многочлен, строящийся алгоритмом TSLS, будет иметь вид $u_0(n) = (-4\alpha_{134} + 1)n^3 + (24\alpha_{134} - 6)n^2 + (-20\alpha_{134} + 5)n = n(n - 1)(n - 5)(1 - 4\alpha_{134})$, т.е. будет зависеть от значения добавленного символьного коэффициента α_{134} . Добавление символьных коэффициентов с большими степенями не изменит определяющий многочлен. И в этом случае без дополнительных исследований алгоритм TSLS оказывается неприменим.

Определение 1. Пусть $A \in \text{Mat}_m(K[x, x^{-1}])$ – матрица системы (1), $d = \deg A$, $q = -\text{val}(xA) = -\text{val}A - 1$, $1 \leq k \leq m$. Пусть $q \leq 0$ ($q > 0$), a_i ($0 \leq i \leq m$) – коэффициенты, вычисленные по формулам (5) (по формулам (6)), $v = \min_{0 \leq i \leq m} \text{val} a_i$, $R = d + 2 - q(m - 2)$ ($R = d + 2 + q$). Систему (1) будем называть *вполне определенной* относительно неизвестной y_k , если $v < R$.

Алгоритм TSLS для построения частичных решений для неизвестной y_k может быть гарантированно применен только к вполне определенным относительно y_k системам. В этом случае определяющий многочлен, строящийся алгоритмом TSLS, гарантированно не зависит от возможных продолжений коэффициентов исходной системы. Возможность использования алгоритма TSLS для систем, не являющихся вполне определенными относительно y_k , требует дополнительного исследования, которое в настоящей работе не проводится.

Система с матрицей (10) из предыдущего примера не является вполне определенной относительно y_1 : в этом случае $d = 3$, $q = -1$, $R = 6$, $v = 6$ и условие $v < R$ не выполняется. Также система из примера 3 не является вполне определенной относительно y_2 , но в то же время она является вполне определенной относительно неизвестной y_3 .

4.4. Алгоритм

Предлагаемый алгоритм построения частичных решений усеченной дифференциальной системы (1) принимает на вход матрицу A системы, являющейся d -усечением некоторой неизвестной матрицы $B \in \text{Mat}_m(K((x)))$ и номер $1 \leq k \leq m$ неизвестной, для которой необходимо построить лораново решение.

Алгоритм состоит из следующих шагов:

1. $p := d$; $q := -\text{val}A - 1$;

2. $p := p + 1$;

3. построить матрицу $A^{[p]}$, добавив к элементам A члены до степени x^p с символьными коэффициентами: $A_{ij}^{[p]} = A_{ij} + \sum_{l=d+1}^p \alpha_{ijl} x^l$ ($1 \leq i, j \leq m$);

4. построить скалярное дифференциальное уравнение для компоненты y_k , используя для коэффициентов формулы (5), если $q \leq 0$, или формулы (6), если $q > 0$;

5. усечь коэффициенты скалярного дифференциального уравнения, используя оценки предложения 1 (при $q \leq 0$) или предложения 2 (при $q > 0$);

6. применить к полученному скалярному дифференциальному уравнению с усеченными коэффициентами алгоритм TSLS;

7. если результатом алгоритма TSLS является пустое множество, то закончить выполнение алгоритма, лорановых решений нет;

8. если хотя бы в одно из решений, выданных алгоритмом TSLS, не входят добавленные символьные коэффициенты α , то вернуться к шагу 2;

9. усечь полученные решения так, чтобы в них остались только члены, в коэффициенты которых не входят α , и выдать их в качестве результата работы.

Предложение 3. Пусть усеченная дифференциальная система (1) является вполне определенной относительно неизвестной y_k ($1 \leq k \leq m$). Тогда предложенный алгоритм построит максимальное число начальных членов лоранового решения k -й компоненты решений системы (1), независимых от продолжений коэффициентов системы, либо установит, что таких решений не существует.

Доказательство. Поскольку заданная усеченная система (1) является вполне определенной относительно неизвестной y_k , определяющий многочлен, строящийся алгоритмом TSLS по скалярному дифференциальному уравнению для y_k , не зависит от возможных продолжений коэффициентов системы и позволяет определить все возможные валуации лорановых решений для компоненты y_k или

установить, что лорановых решений нет. Докажем, что алгоритм всегда завершает свою работу, т.е. что процесс наращивания коэффициентов системы (шаги 2 и 3) обязательно приводит к тому, что все полученные алгоритмом TSLS решения будут зависеть от символьных коэффициентов α . Для этого покажем, что для достаточно больших значений p условие шага 8 не выполняется и возврата к шагу 2 не происходит.

Пусть $q = -\text{val}A - 1$. Рассмотрим отдельно случаи $q \leq 0$ и $q > 0$.

1) Пусть $q \leq 0$, тогда коэффициенты скалярного уравнения для y_k вычисляются по формулам (5) и усекаются, используя оценки предложения 1. Поскольку степень усечения коэффициента a_m не больше степени усечения других коэффициентов скалярного уравнения, достаточно, чтобы символьные коэффициенты α остались после усечения в коэффициенте a_m . Пусть $p > d = \text{deg} A$, обозначим $a_m^{[p]} = \det[\Delta_{x^A} e_k]$ – старший коэффициент скалярного уравнения, построенного для неизвестной y_k дифференциальной системы (1) с матрицей $A^{[p]}$. Не трудно видеть, что $\text{deg} a_m^{[p]} = m(m - 1)(p + 1)/2$, причем старший коэффициент, т.е. коэффициент при $x^{\text{deg} a_m^{[p]}}$ в $a_m^{[p]}$, вычисляется только через α_{ijp} ($1 \leq i, j \leq m$). После наращивания коэффициентов получаем $\text{deg} a_m^{[p+1]} > \text{deg} a_m^{[p]}$, причем коэффициент при $x^{\text{deg} a_m^{[p]}}$ в $a_m^{[p+1]}$ отличается от коэффициента при той же степени x в $a_m^{[p]}$ слагаемыми, в каждое из которых множителем входят добавленные коэффициенты $\alpha_{i,j,p+1}$, подстановка нулей вместо которых необходимо приводит к коэффициенту при $x^{\text{deg} a_m^{[p]}}$ в $a_m^{[p]}$. Таким образом, коэффициент при $x^{\text{deg} a_m^{[d+1]}}$ обязательно будет содержать символьные коэффициенты α при $p \geq d + 1$. Чтобы член с $x^{\text{deg} a_m^{[d+1]}}$ остался после усечения в коэффициенте $a_m^{[p]}$, должно выполняться условие $p + 2 - q(m - 2) > \text{deg} a_m^{[d+1]}$. Чтобы коэффициент при $x^{\text{deg} a_m^{[d+1]}}$ участвовал в построении решений (см. [3, Prop. 1]), дополнительно требуется выполнение условия $p + 2 - q(m - 2) - \min_{0 \leq i \leq m} \text{val} a_i > e^* - e_*$, где a_i ($0 \leq i \leq m$) – коэффициенты скалярного дифференциального уравнения для y_k , вычисленные по формулам (5), e^* , e_* – соответственно наибольший и наименьший целые корни определяющего многочлена, строящегося алгоритмом TSLS. Окончательно получаем, что для

$$p > \max\left(\frac{m(m-1)(d+2)}{2}, e^* - e_* + \min_{0 \leq i \leq m} \text{val} a_i\right) + q(m-2) - 2$$

в коэффициенты решений, строящихся алгоритмом TSLS, попадут добавленные символьные коэффициенты α , поэтому рост p на шаге 2 алгоритма ограничен сверху значением p_{\max} , для которого справедлива оценка

$$p_{\max} \leq \max\left(\frac{m(m-1)(d+2)}{2}, e^* - e_* + \min_{0 \leq i \leq m} \text{val} a_i\right) + q(m-2) - 1. \quad (11)$$

2) Пусть $q > 0$, коэффициенты скалярного уравнения для y_k вычисляются по формулам (6) и усекаются, используя оценки предложения 2. Поскольку наименьшая из оценок усечения соответствует коэффициенту a_0 , рассуждения аналогичны п. 1, но относительно коэффициента a_0 . Пусть $p > d = \deg A$, обозначим $a_0^{[p]} = -x^{qm(m+1)/2} \det[\Delta_{x^{A^{[p]}} e_k}]_{(1)}$ – младший коэффициент скалярного уравнения, построенного для неизвестной y_k дифференциальной системы (1) с матрицей $A^{[p]}$. Тогда $\deg a_0^{[p]} = qm(m+1)/2 + \deg[\Delta_{x^{A^{[p]}} e_k}]_{(1)} = m(m-1)(p+q+1)/2$, поскольку, как нетрудно показать, $\deg[\Delta_{x^{A^{[p]}} e_k}]_{(1)} = m(m+1)(p+1)/2$. Чтобы член с $x^{\deg a_0^{[p]}}$ остался после усечения в коэффициенте $a_0^{[p]}$, должно выполняться условие $p+2+q > \deg a_0^{[d+1]}$, и чтобы коэффициент при $x^{\deg a_0^{[d+1]}}$ участвовал в построении решений (см. [3, Prop. 1]), дополнительно требуется выполнение условия $p+2+q - \min_{0 \leq i \leq m} \text{val} a_i > e^* - e_*$, где a_i ($0 \leq i \leq m$) – коэффициенты скалярного дифференциального уравнения для y_k , вычисленные по формулам (6), e^* , e_* – соответственно наибольший и наименьший целые корни определяющего многочлена, строящегося алгоритмом TSLS. Окончательно получаем, что для

$$p > \max\left(\frac{m(m+1)(d+q+2)}{2}, e^* - e_* + \min_{0 \leq i \leq m} \text{val} a_i\right) - q - 2.$$

в коэффициенты решений, строящихся алгоритмом TSLS, попадут добавленные символьные коэффициенты α , поэтому рост p на шаге 2 алгоритма

ограничен сверху значением p_{\max} , для которого справедлива оценка

$$p_{\max} \leq \max\left(\frac{m(m+1)(d+q+2)}{2}, e^* - e_* + \min_{0 \leq i \leq m} \text{val} a_i\right) - q - 1. \quad (12)$$

□

Пример 3. Рассмотрим работу алгоритма на примере усеченной системы $y' = Ay$, где $y = (y_1, y_2)^T$,

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

Данная система является вполне определенной относительно y_1 . Найдем лорановы решения для компоненты y_1 , используя предложенный алгоритм.

Для заданной системы степень усечения $d = 0$. На шаге 1 устанавливаем $p := 0$, $q := -\text{val} A - 1 = 1$. На шаге 2, являющимся началом цикла, увеличиваем p до 1. На шаге 3 строим матрицу

$$A^{[1]} = \begin{bmatrix} x^{-2} + \alpha_{111}x & x^{-1} + \alpha_{121}x \\ -x^{-2} + \alpha_{211}x & -x^{-1} - 1 + \alpha_{221}x \end{bmatrix}.$$

На шаге 4 получаем скалярное дифференциальное уравнение для неизвестной y_1 , которое, после усечения коэффициентов на шаге 5, примет вид $(x^3 + \alpha_{121}x^5 + O(x^6))\theta^2 y_1 + (-x^2 + x^3 + (-\alpha_{121} + 1)x^4 + O(x^5))\theta y_1 + (x^2 - x^3 + O(x^4))y_1 = 0$. Лораново решение, строящееся для этого скалярного уравнения с помощью алгоритма TSLS на шаге 6, имеет вид $y_1 = c_1 x + c_1 x^2 + O(x^3)$. Поскольку построенное решение не содержит добавленных символьных коэффициентов α , возвращаемся к шагу 2 и увеличиваем p до 2. На шаге 3 строим матрицу $A^{[2]}$, которая примет вид

$$\begin{bmatrix} x^{-2} + \alpha_{111}x + \alpha_{112}x^2 & x^{-1} + \alpha_{121}x + \alpha_{122}x^2 \\ -x^{-2} + \alpha_{211}x + \alpha_{212}x^2 & -x^{-1} - 1 + \alpha_{221}x + \alpha_{222}x^2 \end{bmatrix}.$$

Скалярное дифференциальное уравнение, строящееся по $A^{[2]}$, после усечения коэффициентов на шаге 5, будет иметь вид: $(x^3 + \alpha_{121}x^5 + \alpha_{122}x^6 + O(x^7))\theta^2 y_1 + (-x^2 + x^3 + (-\alpha_{121} + 1)x^4 + (-\alpha_{111} - \alpha_{121} - \alpha_{122} - \alpha_{221})x^5 + O(x^6))\theta^2 y_1 + (x^2 - x^3 + (4\alpha_{121} + \alpha_{221})x^4 + O(x^5))y_1 = 0$. Применение к нему алгоритма

TSLs дает в результате решение $y_1 = c_1x + c_1x^2 + \left(\frac{3}{2}\alpha_{121} + \frac{1}{2}\alpha_{221} + 3\right)c_1x^3 + O(x^4)$. Отбрасывая члены, начиная со степени x^3 , поскольку в коэффициент перед ней вошли добавленные $\alpha_{121}, \alpha_{221}$, окончательно получаем $y_1 = c_1x + c_1x^2 + O(x^3)$.

Замечание 2. Оценки (11) и (12) степени наращивания коэффициентов системы зачастую оказываются завышенными. Для системы из примера 2, используя оценку (12), можно получить условие $p_{\max} \leq \max(9, 2) - 2 = 7$ хотя алгоритм завершил работу уже при $p = 2$. Последовательное наращивание коэффициентов зачастую позволяет избежать лишних вычислений.

4.5. Реализация

Предложенный алгоритм реализован в системе компьютерной алгебры Maple в виде процедуры `LaurentPartialSolution(A, k, d)`, принимающей в качестве параметров матрицу A системы (1), элементами которой являются многочлены или многочлены Лорана, порядковый номер k компоненты вектора неизвестных, для которой будут строиться частичные решения, и необязательный параметр d , задающий степень усечения коэффициентов системы. Если параметр d не указан, в качестве степени усечения берется максимальная степень элементов матрицы A .

Процедура `LaurentPartialSolution` использует для работы ряд вспомогательных процедур: `MinimalEquationTheta`, `MinimalTruncatedEquationTheta`, `ReduceSolutions`.

Процедура `MinimalEquationTheta(A, k)` предназначена для построения скалярного дифференциального уравнения, записанного с помощью оператора $\theta = x \frac{d}{dx}$, для компоненты решений y_k дифференциальной системы (1) с матрицей A . Для этого используются формулы (5) или (6) в зависимости от вычисляемого значения $q = -\text{val}A - 1$.

Процедура `MinimalTruncatedEquationTheta(A, k)` строит скалярное дифференциальное уравнение с помощью процедуры `MinimalEquationTheta` и усекает его коэффициенты с помощью оценок предложений 1 или 2 в зависимости от значения $q = -\text{val}A - 1$.

Процедура `ReduceSolutions(L)`, которой передается список L лорановых решений, строящихся алгоритмом TSLs, сокращает его, убирая решения с большими валюациями, являющиеся

частными случаями решений с меньшими валюациями при занулении свободных постоянных.

Результатом работы процедуры `LaurentPartialSolution` является список найденных лорановых решений (начальные члены, инвариантные относительно продолжения коэффициентов системы) для компоненты y_k дифференциальной системы (1), либо сообщение об ошибке, если система не является вполне определенной относительно неизвестной y_k . Строящийся список решений в частности может быть пустым, если для k -й компоненты нетривиальных лорановых решений не существует.

В своей работе процедура `LaurentPartialSolution` обращается к реализации алгоритма TSLs, разработанной авторами [5] и доступной по адресу <http://www.ccas.ru/ca/truncatedseries>. Исходный код процедуры `LaurentPartialSolution` и всех вспомогательных процедур доступен по адресу http://www.ccas.ru/ca/_media/tslps.mpl.

Пример 4. Продемонстрируем работу процедур на примере дифференциальной системы (1) с матрицей

```
> A:=Matrix([[1, x], [-x, 1]])
```

$$A := \begin{bmatrix} 1 & x \\ -x & 1 \end{bmatrix}$$

```
> MinimalEquationTheta(A, 1)
```

$$x^2\theta(y(x), x, 2) + (-2x^3 - 2x^2)\theta(y(x), x, 1) + (x^6 + x^4 + x^3)y(x)$$

```
> MinimalTruncatedEquationTheta(A, 1)
```

$$(x^2 + O(x^3))\theta(y(x), x, 2) + (-2x^2 + O(x^3))\theta(y(x), x, 1) + (x^3 + O(x^4))y(x)$$

```
> LaurentPartialSolution(A, 1)
```

$$[x^2_c_2 + x_c_1 + _c_1 + O(x^3)]$$

5. БЛАГОДАРНОСТИ

Частичная поддержка РФФИ, грант 19-01-00032.

СПИСОК ЛИТЕРАТУРЫ

1. *Абрамов С.А., Бронштейн М.* Решение линейных дифференциальных и разностных систем по отношению к части неизвестных // Журнал вычисл. матем. и матем. физ. 2006. № 2. С. 229–241.

2. *Abramov S.A., Barkatou M.A., Khmel'nov D.E.* On full rank differential systems with power series coefficients // *Journal of Symbolic Computation*. 2015. V. 68. P. 120–137.
3. *Abramov S., Barkatou M., Pflügel E.* Higher-order linear differential systems with truncated coefficients // *Proc. of CASC'2011. Lecture Notes in Computer Science*. 2011. V. 6885. P. 10–24.
4. *Абрамов С.А., Рябенко А.А., Хмельнов Д.Е.* Процедуры поиска локальных решений линейных дифференциальных систем с бесконечными степенными рядами в роли коэффициентов // *Программирование*. 2016. № 2. С. 75–86.
5. *Абрамов С.А., Рябенко А.А., Хмельнов Д.Е.* Линейные обыкновенные дифференциальные уравнения и усеченные ряды // *Журнал вычисл. матем. и матем. физ.* 2019. Т. 59. № 10. С. 1706–1717.
6. *van der Put M., Singer M.F.* Galois Theory of Linear Differential Equations. *Grundlehren der mathematischen Wissenschaften, 328*, Springer, Heidelberg, 2003.
7. *Hartman P.* Ordinary differential equations. — John Wiley and Sons, Inc., New York, 1964.

**ПОИСК РАВНОВЕСНЫХ СОСТОЯНИЙ МАШИНЫ АТВУДА
С ДВУМЯ КОЛЕБЛЮЩИМИСЯ ГРУЗАМИ С ПРИМЕНЕНИЕМ
КОМПЬЮТЕРНОЙ АЛГЕБРЫ**© 2021 г. А. Н. Прокопеня^{a,*}^a Варшавский университет естественных наук – SGGW,
Польша 02-776 Варшава, ул. Новоурсыновска, 159

*E-mail: alexander_prokopenya@sggw.pl

Поступила в редакцию 27.08.2020 г.

После доработки 02.09.2020 г.

Принята к публикации 12.09.2020 г.

Обсуждается проблема поиска равновесных состояний машины Атвуда, в которой шкив конечного радиуса заменяется двумя отдельными малыми шкивами и оба груза могут колебаться в вертикальной плоскости. Получены дифференциальные уравнения движения системы и вычислены их решения в виде степенных рядов по малому параметру. Показано, что в случае грузов одинаковой массы равновесное положение $r = \text{const}$ системы существует только при одинаковых амплитудах и частотах колебаний грузов и сдвиге фаз $\alpha = 0$ или $\alpha = \pi$. Кроме того, возможно состояние динамического равновесия, когда оба груза совершают колебания с одинаковыми амплитудами и частотами, а сдвиг фаз составляет $\alpha = \pm\pi/2$. При этом длины маятников также совершают колебания около некоторого равновесного значения. Сравнение полученных результатов с соответствующими численными решениями уравнений движения подтверждает их корректность. Все необходимые вычисления выполняются с помощью системы компьютерной алгебры Wolfram Mathematica.

DOI: 10.31857/S013234742101009X

1. ВВЕДЕНИЕ

Классическая машина Атвуда (см. [1]), в которой один из грузов может колебаться в вертикальной плоскости под действием силы тяжести, представляет собой консервативную механическую систему с двумя степенями свободы, которая была предметом исследования многих работ (см. [2–7]). Следует отметить, что учет колебаний груза приводит к значительному усложнению уравнений движения системы и их общее решение не может быть записано в символьной форме. Численное исследование уравнений движения показывает, что машина Атвуда с одним колеблющимся грузом демонстрирует весьма интересное поведение и может совершать различные виды движения, например, квазипериодическое и хаотическое движение (см. [3, 5, 7]).

Заметим, что колебания груза приводят к возрастанию средней силы натяжения нити, причем ее величина определяется амплитудой колебаний (см. [8]). Поэтому при небольшой разнице масс и соответствующей амплитуде колеблющийся груз меньшей массы может тянуть более тяжелый груз вверх, что невозможно в отсутствие колебаний. С другой стороны, при заданной разнице масс

можно выбрать такие начальные условия, при которых система будет находиться в состоянии динамического равновесия, когда груз большей массы уравновешивается колеблющимся грузом меньшей массы. В таком случае более тяжелый груз совершает поступательное движение и также колеблется около некоторого равновесного положения, а груз меньшей массы ведет себя как маятник, длина которого совершает малые колебания. При этом наблюдается резонанс частот вида 2 : 1, т.е. частота колебаний длины маятника в два раза превышает частоту колебаний угловой переменной. Соответствующее равновесное состояние системы описывается периодическим решением уравнений движения, построенным в [9].

Если оба груза имеют одинаковые массы ($m_1 = m_2$), то в отсутствие колебаний система может находиться в равновесии при любом положении грузов. Если же один из грузов совершает колебания, то равновесное состояние машины Атвуда не существует, поскольку колеблющийся груз будет двигаться вниз и тянуть второй груз вверх даже при малой амплитуде колебаний. Однако можно ожидать, что рассматриваемая система может находиться в состоянии динамического равновесия,

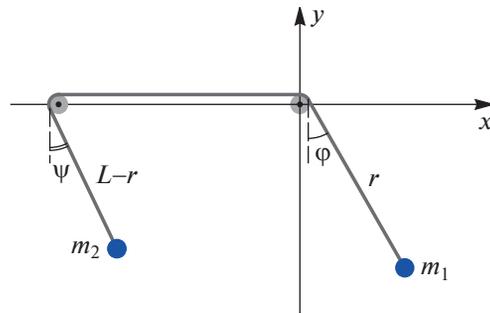


Рис. 1. Машина Атвуда с двумя колеблющимися грузами.

когда второй груз также совершает колебания. Численное решение уравнений движения машины Атвуда с двумя колеблющимися грузами показало, что такое состояние системы существует, если оба груза совершают синфазные колебания или колеблются в противофазе с одинаковыми амплитудами и частотами (см. [10]).

Целью данной работы является поиск решений уравнений движения машины Атвуда с двумя колеблющимися грузами одинаковой массы, определяющих состояния динамического равновесия системы, когда длина маятника r совершает малые колебания около некоторого равновесного значения, а также определение условий, при которых такие решения существуют. Поскольку уравнения движения системы существенно нелинейны и записать их точное решение не представляется возможным, будем искать приближенные решения в виде степенных рядов. Отметим, что построение и исследование таких решений обычно связано с выполнением весьма громоздких символьных вычислений, которые удобно выполнять с помощью систем компьютерной алгебры (см., напр., [11–15]). В данной работе для выполнения всех расчетов и визуализации полученных результатов используется система компьютерной алгебры *Wolfram Mathematica* [16].

2. УРАВНЕНИЯ ДВИЖЕНИЯ

Рассматривается машина Атвуда, в которой шкив конечного радиуса заменяется двумя отдельными малыми шкивами и оба груза могут колебаться в вертикальной плоскости (рис. 1). Такая модификация классической машины Атвуда (см. [1]) не изменяет ее физической природы, но позволяет исключить влияние размеров шкива и сосредоточиться на исследовании влияния колебаний на движение системы.

Рассматриваемая система имеет три степени свободы, а ее функция Лагранжа имеет вид

$$\mathcal{L} = \frac{m_1}{2}(\dot{r}^2 + r^2\dot{\phi}^2) + \frac{m_2}{2}(\dot{r}^2 + (L-r)^2\dot{\psi}^2) + m_1gr \cos \phi + m_2g(L-r) \cos \psi, \quad (2.1)$$

где точка над символом означает полную производную соответствующей функции по времени, g – ускорение свободного падения, r и $(L-r)$ – расстояния между шкивами и грузами m_1 и m_2 соответственно, а углы ϕ и ψ определяют отклонения грузов от вертикали (см. рис. 1). Выражение (2.1) записано в предположении, что радиусы шкивов пренебрежимо малы и изменением длины нити r и $(L-r)$ при колебаниях грузов за счет наматывания нити на шкив можно пренебречь.

Используя функцию Лагранжа (2.1) и выполняя дифференцирование с помощью встроенной функции *D* системы *Mathematica* (см. [16]), получаем уравнения движения в виде

$$r\ddot{\phi} = -g \sin \phi - 2\dot{r}\dot{\phi} \quad (2.2)$$

$$(L-r)\ddot{\psi} = -g \sin \psi + 2\dot{r}\dot{\psi}, \quad (2.3)$$

$$(m_1 + m_2)\ddot{r} = m_1g \cos \phi - m_2g \cos \psi + m_1r\dot{\phi}^2 - m_2(L-r)\dot{\psi}^2. \quad (2.4)$$

Для удобства дальнейших вычислений введем безразмерные переменные

$$r^*(t^*) = r(t)/R_0, \quad t^* = t\sqrt{g/R_0}, \quad (2.5)$$

где R_0 – длина нити r в положении равновесия $\phi = 0, \psi = 0$ в отсутствие колебаний при $m_1 = m_2$. Далее безразмерные переменные r^*, t^* будем обозначать обычным образом через r, t . Тогда уравнения движения (2.2)–(2.4) принимают вид

$$r\ddot{\phi} = -\sin \phi - 2\dot{r}\dot{\phi}, \quad (2.6)$$

$$(k-r)\ddot{\psi} = -\sin \psi + 2\dot{r}\dot{\psi}, \quad (2.7)$$

$$2\ddot{r} = \cos \phi - \cos \psi + r\dot{\phi}^2 - (k-r)\dot{\psi}^2, \quad (2.8)$$

где учтено равенство масс тел и введен параметр $k = L/R_0 > 1$.

Легко видеть, что при $r = \text{const}$ уравнения (2.6), (2.7) сводятся к двум независимым уравнениям колебаний математических маятников длиной r и $k - r$ соответственно:

$$\begin{aligned} \ddot{\phi} + \frac{1}{r} \sin \phi &= 0, \\ \ddot{\psi} + \frac{1}{k-r} \sin \psi &= 0. \end{aligned} \quad (2.9)$$

Уравнения (2.9) имеют равновесные решения $\phi = 0$, $\psi = 0$, подстановка которых в уравнение (2.8) обращает его в тождество при $r = \text{const}$. Это ожидаемый результат, поскольку в отсутствие колебаний система может находиться в равновесии при любом значении $r = \text{const}$. Кроме того, уравнение (2.8) имеет решение $r = \text{const}$ и в случае ненулевых решений уравнений (2.9), которые выражаются через эллиптические функции (см., например, [17]), если выполняется условие $\phi(t) = \pm\psi(t)$. Это возможно при $k = 2r$, когда оба маятника совершают нелинейные колебания одинаковой частоты и амплитуды (см. [10]), а их фазы совпадают ($\phi(t) = \psi(t)$) или противоположны ($\phi(t) = -\psi(t)$).

Однако система (2.6)–(2.8) может иметь и другие решения, которые описывают связанные колебания двух математических маятников переменной длины. Целью данной работы является поиск решений системы (2.6)–(2.8), описывающих состояния динамического равновесия, когда оба маятника совершают нелинейные колебания, а функция $r(t)$ совершает малые квазипериодические колебания около некоторого равновесного значения.

Отметим, что уравнения (2.6)–(2.8) являются нелинейными и существование квазипериодических движений грузов в окрестности состояний динамического равновесия, которыми мы интересуемся в данной работе, возможно только вследствие нелинейного взаимодействия между степенями свободы системы.

3. РЕШЕНИЯ УРАВНЕНИЙ ДВИЖЕНИЯ

Напомним, что колебания математического маятника приводят к увеличению средней силы натяжения нити, причем ее величина определяется амплитудой колебаний (см. [8]). Поскольку мы рассматриваем машину Атвуда с двумя колеблющимися грузами одинаковой массы, можно ожидать, что при одинаковых амплитудах колебаний система будет находиться в состоянии динамического равновесия, при котором переменные $r(t)$, $\phi(t)$, $\psi(t)$ совершают малые колебания около некоторых равновесных значений и являются ква-

зипериодическими функциями времени. Поскольку уравнения движения (2.6)–(2.8) являются нелинейными, частоты колебаний маятников будут зависеть от амплитуд (см., например, [18]). Далее будем считать, что амплитуды колебаний грузов определяются параметром ε , который предполагается малым, но конечным. При $\varepsilon \rightarrow 0$, функции $r(t)$, $\phi(t)$, $\psi(t)$ должны сводиться к равновесному решению $r = 1$, $\phi = 0$, $\psi = 0$.

Для удобства вычислений произведем замену переменных

$$\begin{aligned} r(t) &\rightarrow 1 + \varepsilon r(t), \\ \phi(t) &\rightarrow \sqrt{\varepsilon} \phi(t), \quad \psi(t) \rightarrow \sqrt{\varepsilon} \psi(t), \end{aligned} \quad (3.1)$$

и будем предполагать, что $r(t)$, $\phi(t)$, $\psi(t)$ являются ограниченными осциллирующими функциями. Подставляя (3.1) в (2.6)–(2.8) и заменяя тригонометрические функции их разложениями в степенные ряды с точностью до пятого порядка включительно, перепишем уравнения движения в виде, удобном для применения теории возмущений (см. [18, 19]):

$$\ddot{\phi} + \phi = -\varepsilon \left(r \ddot{\phi} + 2\dot{r} \dot{\phi} - \frac{1}{6} \phi^3 \right) - \frac{\varepsilon^2}{120} \phi^5, \quad (3.2)$$

$$(k-1) \ddot{\psi} + \psi = \varepsilon \left(r \ddot{\psi} + 2\dot{r} \dot{\psi} + \frac{1}{6} \psi^3 \right) - \frac{\varepsilon^2}{120} \psi^5, \quad (3.3)$$

$$\begin{aligned} \ddot{r} &= -\frac{1}{4} (\phi^2 - \psi^2) + \frac{1}{2} \phi^2 - \frac{1}{2} (k-1) \psi^2 + \\ &+ \frac{\varepsilon}{2} \left(r (\phi^2 + \psi^2) + \frac{1}{24} (\phi^4 - \psi^4) \right). \end{aligned} \quad (3.4)$$

Решение системы (3.2)–(3.4) будем искать в виде степенных рядов по малому параметру ε :

$$\phi(t) = \phi_0(t) + \varepsilon \phi_1(t) + \varepsilon^2 \phi_2(t) + \dots, \quad (3.5)$$

$$\psi(t) = \psi_0(t) + \varepsilon \psi_1(t) + \varepsilon^2 \psi_2(t) + \dots, \quad (3.6)$$

$$r(t) = r_0(t) + \varepsilon r_1(t) + \varepsilon^2 r_2(t) + \dots \quad (3.7)$$

Функции $\phi_0(t)$, $\psi_0(t)$ в разложениях (3.5), (3.6) представляют собой решения уравнений (3.2), (3.3) при $\varepsilon = 0$ и без ограничения общности рассуждений могут быть представлены в виде

$$\phi_0(t) = \cos(\omega_1 t), \quad \psi_0(t) = \cos(\omega_2 t + \alpha), \quad (3.8)$$

где начальная фаза переменной $\phi_0(t)$ равна нулю, а постоянная α определяет разность фаз колебаний маятников в момент времени $t = 0$. Амплитуды колебаний в (3.8) одинаковы, что обеспечивает равенство нулю постоянной составляющей функции в правой части уравнения (3.4) и не приводит к появлению линейной или квадратичной за-

висимости от времени функции $r_0(t)$ в (3.7), которая является решением уравнения (3.4) при $\varepsilon = 0$. Напомним, что интересующее нас решение (3.7) описывает малые колебания длины $r(t)$ около равновесного значения $r = 1$, а амплитуды колебаний переменных $\varphi(t)$ и $\psi(t)$ определяются параметром ε (см. (3.1)).

Частоты колебаний ω_1, ω_2 также можем представить в виде степенных рядов по ε :

$$\begin{aligned} \omega_1 &= \omega_{10} + \varepsilon\omega_{11} + \varepsilon^2\omega_{12} + \dots, \\ \omega_2 &= \omega_{20} + \varepsilon\omega_{21} + \varepsilon^2\omega_{22} + \dots \end{aligned} \quad (3.9)$$

где частоты гармонических колебаний $\omega_{10} = 1, \omega_{20} = 1/\sqrt{k-1}$ определяются уравнениями (3.2), (3.3) при $\varepsilon = 0$. Коэффициенты $\omega_{11}, \omega_{21}, \omega_{12}, \omega_{22}, \dots$ в разложениях (3.9) будут определяться из условия, что в правой части уравнений (3.2) и (3.3) отсутствуют резонансные члены, пропорциональные $\cos(\omega_1 t), \sin(\omega_1 t)$ и $\cos(\omega_2 t), \sin(\omega_2 t)$ соответственно, которые приводят к неограниченному возрастанию амплитуды колебаний (см. [17, 18]). Поскольку левые стороны уравнений (3.2), (3.3) должны обращаться строго в нуль при подстановке в них функций вида (3.8) с точными значениями частот (3.9), эти уравнения перепишем в эквивалентном виде

$$\begin{aligned} \ddot{\varphi} + \omega_1^2 \varphi &= (\omega_1^2 - 1)\varphi - \\ &- \varepsilon \left(r\dot{\varphi} + 2\dot{r}\varphi - \frac{1}{6}\varphi^3 \right) - \frac{\varepsilon^2}{120}\varphi^5, \end{aligned} \quad (3.10)$$

$$\begin{aligned} \ddot{\psi} + \omega_2^2 \psi &= \left(\omega_2^2 - \frac{1}{k-1} \right) \psi + \\ &+ \frac{\varepsilon}{k-1} \left(r\dot{\psi} + 2\dot{r}\psi + \frac{1}{6}\psi^3 \right) - \frac{\varepsilon^2}{120(k-1)}\psi^5. \end{aligned} \quad (3.11)$$

Полученная в результате система уравнений (3.4), (3.10), (3.11) записана в форме, удобной для вычисления неизвестных функций $r_k(t), \varphi_k(t), \psi_k(t)$ в разложениях (3.5)–(3.7). Соответствующий алгоритм символьных вычислений в данной работе реализуется с помощью системы компьютерной алгебры *Wolfram Mathematica* [16]. Сначала определяем правила замены вида

$$\begin{aligned} rul1 &= \{r \rightarrow (r_0[\#] + \varepsilon r_1[\#] + \varepsilon^2 r_2[\#])\&\}, \\ \varphi &\rightarrow (\varphi_0[\#] + \varepsilon \varphi_1[\#] + \varepsilon^2 \varphi_2[\#])\&\}; \\ \psi &\rightarrow (\psi_0[\#] + \varepsilon \psi_1[\#] + \varepsilon^2 \psi_2[\#])\&\}; \end{aligned}$$

и выполняем подстановку выражений (3.5)–(3.7) в уравнения (3.4), (3.10), (3.11). Отметим, что использование анонимных функций (pure functions)

в правилах замены приводит к автоматическому вычислению всех производных в уравнениях движения. Кроме того, в правой части каждого уравнения выполняем подстановку

$$\begin{aligned} rul2 &= \left\{ \omega_1 \rightarrow (1 + \varepsilon\omega_{11} + \varepsilon^2\omega_{12}), \right. \\ &\left. \omega_2 \rightarrow \frac{1}{\sqrt{k-1}}(1 + \varepsilon\omega_{21} + \varepsilon^2\omega_{22}) \right\}. \end{aligned}$$

Далее в каждом из полученных уравнений разлагаем левую и правую части в степенные ряды по параметру ε с помощью функции *Series*. Для выполнения разложений с точностью до второго порядка, например, достаточно к каждому уравнению применить функцию

$$Series[\#, \{\varepsilon, 0, 2\}]\&\&.$$

Встроенная функция *Coefficient[eq, \varepsilon, k]* позволяет выделить в выражении *eq* коэффициент при $\varepsilon^k, k = 0, 1, 2, \dots$

Выполняя описанные выше символьные вычисления и приравнивая коэффициенты при одинаковых степенях параметра ε в левой и правой части каждого уравнения (3.4), (3.10), (3.11), получаем следующую систему дифференциальных уравнений:

$$\ddot{\varphi}_0 + \omega_1^2 \varphi_0 = 0, \quad (3.12)$$

$$\ddot{\psi}_0 + \omega_2^2 \psi_0 = 0, \quad (3.13)$$

$$\ddot{r}_0 = \frac{1}{4}(\psi_0^2 - \varphi_0^2) + \frac{1}{2}\dot{\varphi}_0^2 - \frac{1}{2}(k-1)\dot{\psi}_0^2, \quad (3.14)$$

$$\ddot{\varphi}_1 + \omega_1^2 \varphi_1 = -r_0\dot{\varphi}_0 - 2\dot{r}_0\varphi_0 + 2\omega_{11}\varphi_0 + \frac{1}{6}\varphi_0^3, \quad (3.15)$$

$$\begin{aligned} \ddot{\psi}_1 + \omega_2^2 \psi_1 &= \\ &= \frac{1}{k-1} \left(r_0\dot{\psi}_0 + 2\dot{r}_0\psi_0 + 2\omega_{21}\psi_0 + \frac{1}{6}\psi_0^3 \right), \end{aligned} \quad (3.16)$$

$$\begin{aligned} \ddot{r}_1 &= -\frac{1}{2}\varphi_0\varphi_1 + \frac{1}{2}\psi_0\psi_1 + \dot{\varphi}_0\dot{\varphi}_1 - (k-1)\dot{\psi}_0\dot{\psi}_1 + \\ &+ \frac{1}{2}r_0(\dot{\varphi}_0^2 + \dot{\psi}_0^2) + \frac{1}{48}(\varphi_0^4 - \psi_0^4), \end{aligned} \quad (3.17)$$

$$\begin{aligned} \ddot{\varphi}_2 + \omega_1^2 \varphi_2 &= -r_0\ddot{\varphi}_1 - r_1\ddot{\varphi}_0 - 2\dot{r}_0\dot{\varphi}_1 - 2\dot{r}_1\dot{\varphi}_0 + \\ &+ \omega_{11}^2\varphi_0 + 2\omega_{12}\varphi_0 + 2\omega_{11}\varphi_1 + \frac{1}{2}\varphi_0^2\varphi_1 - \frac{1}{120}\varphi_0^5, \end{aligned} \quad (3.18)$$

$$\ddot{\psi}_2 + \omega_2^2 \psi_2 = \frac{1}{k-1} \left(r_0 \ddot{\psi}_1 + r_1 \ddot{\psi}_0 + 2(\dot{r}_0 \dot{\psi}_1 + \dot{r}_1 \dot{\psi}_0) + 2(\omega_{22} \psi_0 + \omega_{21} \psi_1) + \omega_{21}^2 \psi_0 + \frac{1}{2} \psi_0^2 \psi_1 - \frac{1}{120} \psi_0^5 \right), \quad (3.19)$$

$$\begin{aligned} \ddot{r}_2 = & \frac{1}{4}(\psi_1^2 - \phi_1^2) - \frac{1}{2}(\phi_0 \phi_2 - \psi_0 \psi_2) + \\ & + \frac{1}{2} r_1 (\phi_0^2 + \psi_0^2) + r_0 (\phi_0 \phi_1 + \psi_0 \psi_1) + \phi_0 \phi_2 + \\ & + \frac{1}{2} (\phi_1^2 - (k-1) \psi_1^2) - (k-1) \psi_0 \psi_2 + \\ & + \frac{1}{12} (\phi_0^3 \phi_1 - \psi_0^3 \psi_1), \dots \end{aligned} \quad (3.20)$$

Далее предполагаем, что функции $\phi_k(t)$, $k = 1, 2, \dots$ удовлетворяют нулевым начальным условиям $\phi_k(0) = 0$, $\dot{\phi}_k(0) = 0$, и будем искать такие решения уравнений (3.14)–(3.20), которые описывают малые колебания функций $\phi_k(t)$, $\psi_k(t)$, $r_k(t)$.

Очевидно, функции (3.8) являются решениями уравнений (3.12), (3.13), а их подстановка в (3.14) и замена произведений тригонометрических функций на сумму тригонометрических функций от кратных аргументов с помощью встроенной функции *TrigReduce* системы *Mathematica* приводит к дифференциальному уравнению

$$\ddot{r}_0 = -\frac{3}{8} \cos(2\omega_1 t) + \frac{3}{8} \cos(2\omega_2 t + 2\alpha). \quad (3.21)$$

Отметим, что выбор одинаковых амплитуд переменных $\phi_0(t)$, $\psi_0(t)$ в (3.8) обеспечивает обнуление постоянной составляющей функции в правой части (3.21) и позволяет найти частное решение дифференциального уравнения (3.21) в виде осциллирующих функций. Для получения такого решения достаточно дважды проинтегрировать правую часть уравнения (3.21) с помощью встроенной функции *Integrate[# , t, t] &*. В результате находим

$$r_0 = \frac{3}{32\omega_1^2} \cos(2\omega_1 t) - \frac{3}{32\omega_2^2} \cos(2\omega_2 t + 2\alpha). \quad (3.22)$$

Далее подставляем решение (3.22) в уравнение (3.15) и, преобразуя его правую часть в линейную комбинацию тригонометрических функций с помощью функции *TrigReduce*, получаем

$$\begin{aligned} \ddot{\phi}_1 + \omega_1^2 \phi_1 = & \left(2\omega_{11} - \frac{1}{64} \right) \cos(\omega_1 t) + \frac{53}{192} \cos(3\omega_1 t) + \\ & - \left(\frac{3(k-1)}{64} - \frac{3\sqrt{k-1}}{16} \right) \cos((2\omega_2 - \omega_1)t + 2\alpha) - \end{aligned} \quad (3.23)$$

$$- \left(\frac{3(k-1)}{64} + \frac{3\sqrt{k-1}}{16} \right) \cos((2\omega_2 + \omega_1)t + 2\alpha).$$

Аналогичным образом приводим уравнение (3.16) к виду

$$\begin{aligned} \ddot{\psi}_1 + \omega_2^2 \psi_1 = & \frac{2}{k-1} \left(\omega_{21} - \frac{1}{128} \right) \cos(\omega_2 t + \alpha) - \\ & - \left(\frac{3}{64(k-1)^2} - \frac{3}{16(k-1)^{3/2}} \right) \cos((\omega_2 - \\ & - 2\omega_1)t + \alpha) - \left(\frac{3}{64(k-1)^2} + \frac{3}{16(k-1)^{3/2}} \right) \times \\ & \times \cos((\omega_2 + 2\omega_1)t + \alpha) + \\ & + \frac{53}{192(k-1)} \cos(3\omega_2 t + 3\alpha). \end{aligned} \quad (3.24)$$

Дифференциальные уравнения (3.23), (3.24) описывают вынужденные колебания переменных $\phi_1(t)$, $\psi_1(t)$ и их решения будут ограниченными осциллирующими функциями только при условии, что правые части этих уравнений не содержат резонансных членов (см. [17–19]).

Далее рассмотрим специальный случай одинаковых частот $\omega_1 = \omega_2$, что возможно при условии $L = 2R_0$ или $k = 2$, и выделим резонансные члены в правой части (3.23):

$$\left(2\omega_{11} - \frac{1}{64} \right) \cos(\omega_1 t) + \frac{9}{64} \cos(\omega_1 t + 2\alpha).$$

Легко видеть, что существуют только четыре значения начальной фазы α , при которых сумма двух колебаний одинаковой частоты обращается в нуль. Если $\alpha = 0$ или $\alpha = \pi$, то резонансные члены исчезают при $\omega_{11} = -\frac{1}{16}$. В этом случае уравнение (3.23) принимает вид

$$\ddot{\phi}_1 + \omega_1^2 \phi_1 = \frac{1}{24} \cos(3\omega_1 t), \quad (3.25)$$

а его решение, удовлетворяющее начальным условиям $\phi_1(0) = \dot{\phi}_1(0) = 0$, легко вычисляется с помощью встроенной функции *DSolve*

$$\phi_1(t) = \frac{1}{192\omega_1^2} (\cos(\omega_1 t) - \cos(3\omega_1 t)). \quad (3.26)$$

Если же $\alpha = \pm\pi/2$, то условие обнуления резонансных членов в (3.23) дает $\omega_{11} = \frac{5}{64}$, а соответствующее решение уравнения (3.23) имеет вид

$$\phi_1(t) = \frac{49}{768\omega_1^2} (\cos(\omega_1 t) - \cos(3\omega_1 t)). \quad (3.27)$$

Условие обнуления резонансных членов в уравнении (3.24) приводит к таким же результатам $\omega_{21} = -\frac{1}{16} = \omega_{11}$ или $\omega_{21} = \frac{5}{64} = \omega_{11}$, что естественно ожидать, так как частоты колебаний предполагаются одинаковыми ($\omega_1 = \omega_2$). При $\alpha = 0$ решения уравнений (3.23) и (3.24) совпадают и имеют вид (3.26), а при $\alpha = \pi$ знак функции $\psi_1(t)$ изменяется на противоположный ($\psi_1(t) = -\varphi_1(t)$). Заметим, что при $\omega_1 = \omega_2$ функции $\varphi_0(t)$ и $\psi_0(t)$ в (3.8) также совпадают при $\alpha = 0$ и имеют противоположные знаки при $\alpha = \pi$. В обоих случаях подстановка найденных функций $\varphi_1(t)$ и $\psi_1(t)$ в (3.17) с учетом (3.8) приводит к обнулению правой части, что дает решение $r_1(t) = 0$. Поскольку $r_0(t) = 0$ при $\omega_1 = \omega_2$ (см. (3.22)), то можно утверждать, что с точностью до членов порядка $O(\epsilon^2)$ включительно длина обоих маятников остается постоянной $r(t) = 1$, а сами маятники совершают колебания с одинаковыми частотами и амплитудами синфазно ($\varphi(t) = \psi(t)$) или в противофазе ($\varphi(t) = -\psi(t)$).

При $\omega_{11} = \omega_{21} = \frac{5}{64}$ решение уравнения (3.23) имеет вид (3.27) при обоих значениях начальной фазы $\alpha = \pm\pi/2$, а решения уравнения (3.24), удовлетворяющие начальному условию $\psi_1(0) = 0$, запишем в виде

$$\psi_1(t) = C_1 \sin(\omega_1 t) \pm \frac{49}{768\omega_1^2} \cos(3\omega_1 t). \quad (3.28)$$

Поскольку при решении уравнения (3.24) начальное значение производной $\dot{\psi}_1(0)$ не задано, решение (3.28) содержит постоянную C_1 , которая далее будет найдена при решении уравнения (3.17).

Действительно, при подстановке функций (3.27), (3.28) в уравнение (3.17) получаем

$$\ddot{\eta}_i = \frac{49}{3072} \pm \frac{C_1}{4} + \left(\pm \frac{3}{4} C_1 - \frac{175}{1024} \right) \cos(2\omega_1 t). \quad (3.29)$$

При условии $C_1 = \mp 47/768$ постоянная составляющая функции в правой части (3.29) обнуляется, что приводит к осциллирующему частому решению

$$r_1(t) = \frac{7}{128\omega_1^2} \cos(2\omega_1 t). \quad (3.30)$$

Вычисления в более высоких порядках по ϵ выполняются аналогичным образом с применением встроенных функций системы *Mathematica*, таких как *Expand*, *TrigExpand*, *Collect*, *Coefficient*, *D*, *Integrate*, *Series*, *Normal*, *DSolve*. Так, подстановка найденных выше функций в (3.18) при $\alpha = 0$ приводит к уравнению

$$\ddot{\varphi}_2 + \omega_1^2 \varphi_2 = \left(2\omega_{12} - \frac{1}{1536} \right) \cos(\omega_1 t) - \frac{1}{384} \cos(3\omega_1 t) - \frac{3}{2560} \cos(5\omega_1 t). \quad (3.31)$$

Условие отсутствия резонансных членов, пропорциональных $\cos(\omega_1 t)$, дает $\omega_{12} = 1/3072$, а решение уравнения (3.31), удовлетворяющее начальным условиям $\varphi_2(0) = \dot{\varphi}_2(0) = 0$, получается в виде

$$\varphi_2(t) = -\frac{23}{61440} \cos(\omega_1 t) + \frac{1}{3072} \cos(3\omega_1 t) + \frac{1}{20480} \cos(5\omega_1 t). \quad (3.32)$$

Анализ уравнения (3.19) дает $\omega_{22} = 1/3072 = \omega_{12}$, а вычисление функции $\psi_2(t)$ приводит к результату $\psi_2(t) = \varphi_2(t)$ при $\alpha = 0$ и $\psi_2(t) = -\varphi_2(t)$ при $\alpha = \pi$. Далее из уравнения (3.20) находим $r_2(t) = 0$.

Описанный процесс последовательного решения дифференциальных уравнений, которые получаются путем приравнивания коэффициентов при одинаковых степенях параметра ϵ в левой и правой части каждого из уравнений (3.4), (3.10), (3.11), можно продолжить и вычислить решения (3.5)–(3.7) и частоты (3.9) с требуемой точностью, хотя в более высоких порядках такие вычисления становятся все более громоздкими. Например, с точностью до третьего порядка по ϵ находим

$$\varphi(t) = \sqrt{\epsilon} \cos(\omega_1 t) + \frac{\epsilon^{3/2}}{192} (\cos(\omega_1 t) - \cos(3\omega_1 t)) + \frac{\epsilon^{5/2}}{61440} \times (17 \cos(\omega_1 t) - 20 \cos(3\omega_1 t) + 3 \cos(5\omega_1 t)), \quad (3.33)$$

$$\omega_1 = \omega_2 = 1 - \frac{\epsilon}{16} + \frac{\epsilon^2}{3072} - \frac{23\epsilon^3}{737280}. \quad (3.34)$$

При этом длины маятников одинаковы и не изменяются $r(t) = 1 = k - r(t)$, а колебания маятников происходят синфазно ($\psi(t) = \varphi(t)$) или в противофазе ($\psi(t) = -\varphi(t)$).

Следует отметить, что при $\psi(t) = \varphi(t)$ или $\psi(t) = -\varphi(t)$ правая часть точного уравнения движения (2.8) обращается в нуль и при начальных условиях $r(0) = 1$, $\dot{r}(0) = 0$ его точным решением есть постоянная функция $r(t) = 1$. При этом уравнения (2.6), (2.7) являются независимыми и сводятся к (2.9). Таким образом, выражение (3.33) определяет приближенное решение нелинейного дифференциального уравнения колебаний математического маятника (2.9). Соответственно, разложение (3.34) дает приближенное выражение для частоты нелинейных колебаний маятника, которая является функцией амплитуды колебаний, и совпадает с соответствующим разложением точного выра-

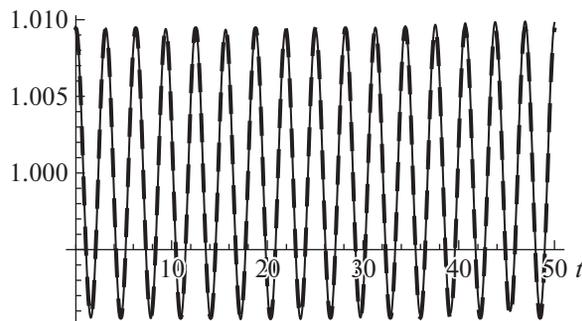


Рис. 2. Сравнение аналитического и численного решений $r(t)$ ($\varepsilon = 0.05$).

жения, выраженного через эллиптические функции Якоби (см. [17]).

Выбор начальной фазы $\alpha = \pi/2$ приводит к появлению зависимости функции $r(t)$ от времени, хотя вычисления функций $\varphi_k(t)$, $\psi_k(t)$, $r_k(t)$ во втором и более высоких порядках по ε производятся подобно случаю $\alpha = 0$. Так, подстановка выражений (3.8), (3.22), (3.27), (3.28), (3.30) в уравнения (3.18), (3.19) дает

$$\ddot{\varphi}_2 + \omega_1^2 \varphi_2 = \left(2\omega_{12} - \frac{1717}{24576}\right) \cos(\omega_1 t) + \frac{2335}{12288} \cos(3\omega_1 t) - \frac{5493}{40960} \cos(5\omega_1 t), \quad (3.35)$$

$$\ddot{\psi}_2 + \omega_1^2 \psi_2 = \left(-2\omega_{22} + \frac{1717}{24576}\right) \sin(\omega_1 t) + \frac{2335}{12288} \sin(3\omega_1 t) + \frac{5493}{40960} \sin(5\omega_1 t). \quad (3.36)$$

Из условия обнуления резонансных членов в правых частях уравнений (3.35), (3.36) получаем

$$\omega_{12} = \omega_{22} = \frac{1717}{49152}. \quad (3.37)$$

Решения уравнений (3.35), (3.36), удовлетворяющие начальным условиям $\varphi_2(0) = \dot{\varphi}_2(0) = 0$, $\psi_2(0) = \dot{\psi}_2(0) = 0$, легко вычисляются с помощью встроенной функции *DSolve*:

$$\varphi_2(t) = \frac{17857}{983040\omega_1^2} \cos(\omega_1 t) - \frac{2335}{98304\omega_1^2} \cos(3\omega_1 t) + \frac{1831}{327680\omega_1^2} \cos(5\omega_1 t), \quad (3.38)$$

$$\psi_2(t) = C_2 \sin(\omega_1 t) - \frac{2335}{98304\omega_1^2} \sin(3\omega_1 t) - \frac{1831}{327680\omega_1^2} \sin(5\omega_1 t). \quad (3.39)$$

Неизвестная постоянная C_2 в выражении (3.39) может быть найдена из уравнения (3.20). Действительно, подставляя полученные выше решения в уравнение (3.20), получаем

$$\ddot{i}_2 = \frac{17857}{3932160} + \frac{C_2}{4} + \left(\frac{3}{4}C_2 - \frac{139453}{1966080}\right) \cos(2\omega_1 t) - \frac{394633}{11796480} \cos(6\omega_1 t). \quad (3.40)$$

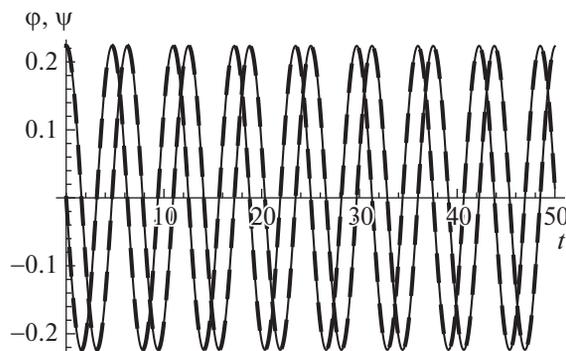


Рис. 3. Сравнение аналитического и численного решений $\varphi(t)$ и $\psi(t)$ ($\varepsilon = 0.05$).

Условие обнуления постоянной составляющей функции в правой части уравнения (3.40) дает

$$C_2 = -\frac{17857}{983040}. \quad (3.41)$$

Интегрируя дважды правую часть уравнения (3.40) с учетом (3.41), получаем

$$r_2(t) = \frac{332477}{15728640\omega_1^2} \cos(2\omega_1 t) + \frac{394633}{424623280\omega_1^2} \cos(6\omega_1 t). \quad (3.42)$$

Описанные вычисления можно продолжить и получить функции (3.5)–(3.7) с необходимой точностью, хотя вычисления становятся все более громоздкими и для их выполнения требуется применение систем компьютерной алгебры. Используя найденные решения, можно вычислить начальные значения функций $\varphi(0)$, $\psi(0)$, $r(0)$ и их производных $\dot{\varphi}(0)$, $\dot{\psi}(0)$, $\dot{r}(0)$ при выбранном значении параметра ε , а затем найти соответствующие численные решения уравнений движения (3.2)–(3.4) с помощью встроенной функции *NDSolve*. Визуализация найденных аналитических решений при $\varepsilon = 0.05$ (штриховые линии на рис. 2 и 3) демонстрирует хорошее совпадение с численными решениями (сплошные тонкие линии на рис. 2 и 3).

4. ЗАКЛЮЧЕНИЕ

В настоящей работе обсуждается проблема построения решений уравнений движения машины Атвуда с двумя колеблющимися грузами одинаковой массы, которые описывают состояние динамического равновесия системы. Показано, что при одинаковых амплитудах и частотах колебаний система может находиться в равновесии, если оба маятника имеют одинаковую длину $r(t) = 1$ и совершают синфазные колебания или колеблются в противофазе. Существование такого равновесного состояния соответствует нашим ожиданиям, так как система обладает симметрией и оба груза в каждый момент времени действуют на нить с одинаковыми силами. Эта симметрия нарушается при сдвиге фаз $\alpha = \pm\pi/2$ и, хотя в среднем действия обоих грузов на нить уравновешиваются, в каждый момент времени действие на нить одного из грузов оказывается большим. В результате длина каждого из маятников совершает малые колебания около равновесного значения $r = 1$, причем частота этих колебаний равняется удвоенной частоте колебаний угловых переменных $\varphi(t)$ и $\psi(t)$ и зависит от амплитуды. Соответствующие решения уравнений движения найдены в виде

степенных рядов по малому параметру ε , который определяет амплитуды колебаний. Последовательно описаны символьные вычисления, необходимые для определения коэффициентов этих рядов, а сами функции найдены с точностью до второго порядка по ε включительно. Сравнение найденного аналитического решения с численным решением уравнений движения показало справедливость полученных теоретических результатов.

Отметим также, что в данной работе все вычисления и визуализация результатов выполнены с использованием системы компьютерной алгебры *Wolfram Mathematica*.

СПИСОК ЛИТЕРАТУРЫ

1. *Atwood G.* A Treatise on the Rectilinear Motion and Rotation of Bodies. Cambridge University Press, 1784.
2. *Tuffillaro N.B., Abbott T.A., Griffiths D.J.* Swinging Atwood's machine // *American Journal of Physics*. 1984. V. 52 (3.1). P. 895–903.
3. *Tuffillaro N.B.* Motions of a swinging Atwood's machine // *J. Physique*. 1985. V. 46. P. 1495–1500.
4. *Tuffillaro N.B.* Integrable motion of a swinging Atwood's machine // *Amer. J. Phys.* 1986. V. 54. P. 142–143.
5. *Casasayas J., Nunes T.A., Tuffillaro N.B.* Swinging Atwood's machine: integrability and dynamics // *J. Physique*. 1990. V. 51. P. 1693–1702.
6. *Yehia H.M.* On the integrability of the motion of a heavy particle on a tilted cone and the swinging Atwood's machine // *Mech. R. Comm.* 2006. V. 33 (25). P. 711–716.
7. *Pujol O., Pérez J.P., Ramis J.P., Simo C., Simon S., Weil J.A.* Swinging Atwood machine: Experimental and numerical results, and a theoretical study // *Physica D*. 2010. V. 239 (12). P. 1067–1081.
8. *Prokopenya A.N.* Motion of a swinging Atwood's machine: simulation and analysis with Mathematica // *Mathematics in Computer Science*. 2017. V. 11(3–4). P. 417–425.
9. *Прокопеня А.Н.* Построение периодического решения уравнений движения обобщенной машины Атвуда с применением компьютерной алгебры // *Программирование*. 2020. Т. 46 (2). С. 53–59.
10. *Prokopenya A.N.* Modelling Atwood's Machine with Three Degrees of Freedom // *Mathematics in Computer Science*. 2019. V. 13 (1–2). P. 247–257.
11. *Абрамов С.А., Зима Е.Б., Ростовцев В.А.* Компьютерная алгебра // *Программирование*. 1992. № 5. С. 4–25.
12. *Васильев Н.Н., Еднерал В.Ф.* Компьютерная алгебра в физических и математических приложениях // *Программирование*. 1994. № 1. С. 70–82.
13. *Прокопеня А.Н.* Некоторые алгоритмы символьных вычислений в исследованиях проблем космиче-

- ской динамики // Программирование. 2006. Т. 32 (2.2). С. 16–22.
14. *Прокопеня А.Н.* Символьные вычисления в исследованиях устойчивости решений линейных систем дифференциальных уравнений с периодическими коэффициентами // Программирование. 2007. Т. 33 (2.2). С. 9–16.
15. *Прокопеня А.Н.* Нормализация гамильтониана в ограниченной задаче многих тел методами компьютерной алгебры // Программирование. 2012. Т. 38 (3). С. 65–78.
16. *Wolfram S.* An elementary introduction to the Wolfram Language, 2nd ed. Champaign, IL, USA, Wolfram Media, 2017.
17. *Маркеев А.П.* Теоретическая механика. Ижевск: НИЦ "Регулярная и хаотическая динамика", 2001. 592 с.
18. *Ландау Л.Д., Лифшиц Е.М.* Механика. 4-е изд. М.: Наука, 1988, 216 с.
19. *Nayfeh A.H.* Introduction to Perturbation Techniques. New York: John Wiley & Sons, 1981. 519 p.

ЭВРИСТИЧЕСКИЕ АЛГОРИТМЫ РАСПОЗНАВАНИЯ НЕКОТОРЫХ
КУБИЧЕСКИХ ГИПЕРПОВЕРХНОСТЕЙ© 2021 г. А. В. Селиверстов^{а,*}^а *Институт проблем передачи информации им. А.А. Харкевича
Российской академии наук
127051 Москва, Большой Каретный пер., д. 19, Россия***E-mail: slvstv@iitp.ru*

Поступила в редакцию 29.06.2020 г.

После доработки 26.08.2020 г.

Принята к публикации 12.09.2020 г.

Предложены эвристические вероятностные алгоритмы полиномиального времени с односторонней ошибкой для распознавания кубических гиперповерхностей, чьи сингулярные локусы не содержат никакого линейного подпространства достаточно большой размерности. Эти алгоритмы легко реализовать в системах компьютерной алгебры. Алгоритмы основаны на проверке условий, что гессиан кубической формы не обращается в нуль тождественно или не определяет конус в проективном пространстве. Проверка свойств гессиана, в свою очередь, выполняема вероятностными алгоритмами с односторонней ошибкой, основанными на лемме Шварца–Зиппеля.

DOI: 10.31857/S0132347421010106

1. ВВЕДЕНИЕ

Гиперповерхностью называется проективное многообразие, определяемое одной формой (однородным многочленом). Кривая на плоскости и поверхность в трехмерном пространстве — это гиперповерхности. Предполагается, что гиперповерхность задана свободной от квадратов формой над полем характеристики нуль, в котором арифметические операции вычислимы за полиномиальное число битовых операций, например, над полем рациональных чисел [1]. Точка называется особой, если все первые частные производные этой формы обращаются в нуль. Поиск особой точки сводится к поиску нетривиального решения системы однородных алгебраических уравнений. В малых размерностях эта задача легко решается посредством символьных вычислений, например, в Maple или MathPartner [2]. Однако в больших размерностях задачи распознавания гладкости и поиска особой точки на кубической гиперповерхности алгоритмически трудные.

Обсуждая вычислительную сложность, мы рассматриваем символьные вычисления, где не происходит ни округления числовых значений, ни приближения мероморфных функций рациональными. Работая над расширением поля рати-

ональных чисел, методы компьютерной алгебры нельзя заменить численными методами. Если не оговорено противное, временем работы алгоритма называется число арифметических операций над полем, операций сравнения и копирования чисел, а также операций над индексами. Более формально, мы рассматриваем вычисления на обобщенных регистровых машинах над некоторым полем [3]. Если каждая арифметическая операция над полем вычислима за полиномиальное число битовых операций, вычислимость за полиномиальное время на обобщенной регистровой машине над этим полем, вообще говоря, не влечет вычислимость за полиномиальное число битовых операций, поскольку запись ответа может иметь экспоненциальную длину. Например, этот эффект возникает при многократном возведении в степень $(...(x^2)^2 ...)^2$.

Если в худшем случае задача трудна, то в типичном случае решить ее может быть легче [4]. Поэтому интересны так называемые генерические алгоритмы, когда для почти всех входов — на большей доле входов данной длины — быстро вычисляется правильный ответ, но для небольшой доли входов результатом служит лишь предупреждение о невозможности вычисления [5, 6]. Гене-

рический алгоритм не ошибается в отличие от эвристического алгоритма, который для почти всех входов дает правильный ответ, но для некоторых входов ответ может быть ложным. Алгоритм с односторонней ошибкой может допускать ошибку только одного рода. Тривиальный алгоритм, принимающий все входы, служит примером эвристического алгоритма с односторонней ошибкой для задачи распознавания гладкости кубической гиперповерхности, поскольку общая форма определяет гладкую гиперповерхность. Трудно предложить для этой задачи эвристический алгоритм с односторонней ошибкой другого рода, безошибочно распознающий особые кубические гиперповерхности.

Вероятностные алгоритмы используют в ходе работы случайные биты [4]. Примером служит вероятностное сведение задачи о поиске двоичного решения системы алгебраических уравнений к поиску двоичного решения одного уравнения [7]. Замена случайных битов псевдослучайной последовательностью превращает вероятностный алгоритм в эвристический (но, вообще говоря, не в генерический) алгоритм.

Существует несколько методов поиска особых точек. Поиск решения системы алгебраических уравнений посредством вычисления базиса Гребнера реализован во многих системах компьютерной алгебры, но требует больших затрат времени и памяти [8, 9]. Исследование свойств особой точки увлекает многогранник Ньютона [10]. Впервые этот многогранник был применен А.Д. Брюно для поиска асимптотики решений систем дифференциальных уравнений [11], а его обобщение — многогранник Адамара — для параметризации [12, 13]. Для общей гиперповерхности с единственной особой точкой, которая служит обыкновенной двойной точкой, координаты этой особой точки выражаются рациональными функциями от коэффициентов некоторой формы, определяющей эту гиперповерхность [14].

Гиперповерхность степени d в \mathbb{P}^n соответствует гиперплоскому сечению многообразия Веронезе. Гиперповерхность особая, если секущая гиперплоскость касается многообразия Веронезе, то есть служит точкой двойственного многообразия. Степень этого двойственного многообразия равна $(n+1)(d-1)^n$. Это степень дискриминанта формы степени d от $n+1$ переменной [15]. Например, дискриминант квадратичной формы

пропорционален определителю матрицы Гессе, а его степень равна числу переменных $n+1$. Если бинарная форма служит гомогенизацией неоднородного многочлена от одной переменной, то дискриминант формы обращается в нуль, когда этот многочлен имеет кратный корень.

Гладкость плоских кривых может быть проверена специальными методами [16]. Кубическая кривая проективно эквивалентна кривой в форме Вейерштрасса. Ее аффинная часть задана уравнением $y^2 = x^3 + px + q$. Эта кривая особая, когда дискриминант многочлена в правой части, равный $-4p^3 - 27q^2$, обращается в нуль. Приведение к форме Вейерштрасса сводится к поиску точки перегиба, которая существует на каждой неприводимой плоской кубической кривой [17]. Алгоритм приведения к форме Вейерштрасса реализован в системе Maple в пакете `algcurves` [18].

Проверка гладкости или вычисление сингулярного локуса кубической поверхности может применяться для моделирования и визуализации сложных поверхностей [19]. Кубическая поверхность в \mathbb{P}^3 , заданная над некоторым полем K и содержащая некоторую K -точку, унирациональна над K , то есть некоторое всюду плотное в топологии Зарисского множество K -точек на этой поверхности допускает рациональную параметризацию [20, 21]. Поэтому кубические поверхности удобнее для моделирования, чем поверхности больших степеней [22, 23]. С другой стороны, гладкие вещественные кубические поверхности с двумя компонентами связности унирациональны, но иррациональны над полем вещественных чисел.

2. ПРЕДВАРИТЕЛЬНЫЕ РЕЗУЛЬТАТЫ

Мы предполагаем, что в каждой размерности фиксировано проективное пространство \mathbb{P}^n с однородными координатами $(x_0 : \dots : x_n)$. Значения однородных координат не обращаются в нуль одновременно, а два набора координат, отличающиеся общим ненулевым множителем, определяют одну и ту же точку в \mathbb{P}^n . Свободная от квадратов форма $f(x_0, \dots, x_n)$ определяет гиперповерхность, в точках которой форма f обращается в нуль. Если после некоторой линейной замены координат форма зависит от меньшего числа переменных, то она определяет конус в \mathbb{P}^n .

Матрица Гессе многочлена – это симметричная матрица вторых частных производных. Элементами матрицы Гессе многочлена третьей степени служат линейные функции. Определитель матрицы Гессе называется гессианом. Гессиан определяется гиперповерхностью с точностью до ненулевого множителя, что позволяет говорить о гессиане гиперповерхности. След матрицы Гессе называется лапласианом. В системе Maple матрица Гессе, гессиан и лапласиан вычисляются в пакете VectorCalculus. Команды называются Hessian с опцией determinant для вычисления гессиана и Laplacian, соответственно.

Вычислительная сложность определителя матрицы имеет тот же порядок роста, что и для матричного умножения. Известны алгоритмы, асимптотически более эффективные, чем метод Гаусса [24, 25]. Определитель легко вычислить, например, над полем рациональных чисел или полем рациональных функций от одной переменной с рациональными коэффициентами. Но для матриц над кольцом многочленов от многих переменных, вообще говоря, определитель нельзя вычислить за время, ограниченное многочленом от числа переменных. В этом случае для проверки отличия определителя от тождественно нулевого обычно используют лемму Шварца–Зиппеля [26].

Лемма 1 (Шварц–Зиппель). *Дан многочлен $f(x_0, \dots, x_n)$ положительной степени $d > 0$ над некоторым полем. Для любого конечного множества S элементов этого поля и для независимых случайных величин ξ_0, \dots, ξ_n , равномерно распределенных на множестве S , вероятность обращения многочлена в нуль $f(\xi_0, \dots, \xi_n) = 0$ не превосходит отношения $\frac{d}{|S|}$.*

Для конуса в \mathbb{P}^n гессиан обращается в нуль, поскольку строки матрицы Гессе линейно зависимы над полем коэффициентов. Однако гессиан может обращаться в нуль и в случаях, когда строки матрицы Гессе линейно зависимы лишь над полем рациональных функций [27]. Как показали в 1876 г. П. Гордан и М. Нётер (P. Gordan, M. Noether), это выполнено для кубической формы $x_0x_3^2 + x_1x_3x_4 + x_2x_4^2$, которая определяет отличную от конуса гиперповерхность в \mathbb{P}^4 .

Тернарные формы с тождественно нулевым гессианом рассмотрены П.В. Бибиковым [28]. Мы рассмотрим лишь кубические формы, но от многих переменных.

Если гиперповерхность X особая, то множество особых точек на X образует подмногообразие, называемое сингулярным локусом. Мы рассмотрим гиперповерхности, у которых сингулярный локус содержит достаточно большие линейные подпространства.

Обозначим через E_0, \dots, E_n точки в \mathbb{P}^n , где однородные координаты точки E_k равны нулю за исключением k -й координаты.

Теорема 1. *Дана кубическая форма $f(x_0, \dots, x_n)$, которая определяет особую проективную гиперповерхность $X \in \mathbb{P}^n$. Если линейное координатное подпространство, заданное уравнениями $x_{m+1} = 0, \dots, x_n = 0$, вложено в сингулярный локус на X , то*

$$f = \sum_{k=0}^m x_k q_k(x_{m+1}, \dots, x_n) + c(x_{m+1}, \dots, x_n),$$

где q_k – некоторые квадратичные формы, а c – кубическая форма.

Доказательство. Предположим, что форма f содержит моном $x_k^2 x_j$ для индексов $k \leq m$ и некоторого j , возможно, равного k . Тогда частная производная $\frac{\partial f}{\partial x_j}$ отлична от нуля в точке E_k , что противоречит условию, что точка E_k особая. Следовательно, для каждого индекса $k \leq m$ форма f не содержит мономов $x_k^2 x_j$ и, в частности, мономов x_k^3 .

Предположим, что для некоторых индексов, удовлетворяющих условиям $k < \ell \leq m$, $j \neq k$ и $j \neq \ell$, форма f содержит моном $x_k x_\ell x_j$. Тогда частная производная $\frac{\partial f}{\partial x_j}$ отлична от нуля в точке, где только k -я и ℓ -я однородные координаты отличны от нуля. Но по условию эта точка особая. Противоречие доказывает, что такого монома нет. \square

Существование сингулярного локуса малой размерности не влечет обращение в нуль гессиана. Например, зонтик Уитни, заданный формой $x_0x_2^2 + x_1x_3^2$, служит примером поверхности, чей сингулярный локус содержит прямую. Эта прямая определяется двумя уравнениями $x_2 = 0$ и $x_3 = 0$. Но гессиан этой формы равен $16x_2^2x_3^2$. Другая форма $x_0x_3^2 + x_1x_4^2 + x_2x_5^2$ определяет гиперповерхность в \mathbb{P}^5 , сингулярный локус которой со-

держит плоскость. Гессиан этой формы равен $-64x_3^2x_4^2x_5^2$.

Как обычно [29], для данного множества форм, коэффициенты которых зависят от параметров ξ_0, \dots, ξ_m , некоторое свойство $\Phi(\xi_0, \dots, \xi_m)$ выполнено для почти каждой формы этого множества, если существует такой многочлен $p(\xi_0, \dots, \xi_m)$, не обращающийся в нуль тождественно, что для каждого допустимого набора значений параметров ξ_0, \dots, ξ_m если свойство $\Phi(\xi_0, \dots, \xi_m)$ не выполнено, то многочлен обращается в нуль $p(\xi_0, \dots, \xi_m) = 0$. Свойство, которое выполнено для почти каждого набора значений параметров, не выполняется лишь на нигде не плотном множестве меры нуль. Но в общем случае множество наборов значений, на которых свойство Φ не выполняется, может быть собственным подмножеством множества нулей многочлена p .

Для некоторой матрицы A обозначим через A^T транспонированную матрицу.

Теорема 2. Дано нечетное число $n = 2m + 1$. Для почти каждой кубической формы $f(x_0, \dots, x_n)$, определяющей проективную гиперповерхность в \mathbb{P}^n , чей сингулярный локус содержит линейное подпространство, заданное уравнениями $x_{m+1} = 0, \dots, x_n = 0$, гессиан формы f — многочлен степени $n + 1$, который не зависит от переменных x_0, \dots, x_m .

Доказательство. Из теоремы 1 следует, что матрица Гессе формы f — это блочная матрица вида

$$H = \begin{pmatrix} 0 & A \\ A^T & B \end{pmatrix},$$

где A и B — квадратные матрицы порядка $m + 1$, причем элементами матрицы A служат линейные формы, которые не зависят от переменных x_0, \dots, x_m . Гессиан формы f , равный $\det H$, не зависит от блока B , но только от A .

Форма $x_0x_{m+1}^2 + \dots + x_mx_n^2$ удовлетворяет условиям теоремы и ее гессиан равен $(-4)^{m+1}x_{m+1}^2 \dots x_n^2$. Согласно теореме 1, все рассматриваемые кубические формы параметризуются наборами коэффициентов квадратичных форм q_k и кубической формы s . Следовательно, существует многочлен p от этих коэффициентов, значение которого равно коэффициенту при мономе $x_{m+1}^2 \dots x_n^2$ гессиана со-

ответствующей формы f . И степень многочлена p положительная. Следовательно, для почти каждой рассматриваемой формы f гессиан содержит моном $x_{m+1}^2 \dots x_n^2$ степени $n + 1$. Но мономов большей степени в этом гессиане нет. \square

Напомним достаточное условие обращения в нуль гессиана кубической формы [27].

Теорема 3 ([27]). Дана кубическая форма $f(x_0, \dots, x_n)$, которая определяет проективную гиперповерхность $X \in \mathbb{P}^n$. Если для некоторого $t > \frac{1}{2}(n-1)$ линейное подпространство, заданное уравнениями $x_{m+1} = 0, \dots, x_n = 0$, вложено в сингулярный локус на X , то гессиан формы f тождественно обращается в нуль.

Доказательство. Согласно теореме 1, матрица Гессе формы f содержит в первых строках и столбцах нулевую подматрицу порядка $t + 1$. Порядок матрицы Гессе равен $n + 1$. При условии $t + 1 > \frac{1}{2}(n + 1)$ определитель матрицы Гессе тождественно обращается в нуль. \square

3. ОСНОВНЫЕ РЕЗУЛЬТАТЫ

Теорема 4. Существует вероятностный алгоритм полиномиального времени, который получает на вход рациональное число $\varepsilon > 0$, целое число n и заданную список коэффициентов кубическую форму $f(x_0, \dots, x_n)$. Если форма f определяет гиперповерхность в \mathbb{P}^n , чей сингулярный локус содержит некоторое линейное подпространство размерности больше $\frac{1}{2}n$, то вход всегда отвергается. Однако для почти каждой кубической формы f вход принимает с вероятностью больше $1 - \varepsilon$.

Доказательство. На первом шаге вычисляется матрица Гессе формы f . На втором шаге посредством вероятностного алгоритма, основанного на лемме Шварца—Зиппеля [26], проверяется отличие гессиана формы f от тождественно нулевого. Если сингулярный локус гиперповерхности достаточно большой, то в силу теоремы 3 гессиан обращается в нуль при любой оценке переменных. Алгоритм вычисляет значение гессиана на случайно и независимо выбранных из отрезка от нуля до $\lceil (n + 1)/\varepsilon \rceil$ целочисленных оценок для переменных x_0, \dots, x_n . Если вычисленное значение

нулевое, то вход отвергается, иначе вход принимается. \square

Замечание. Матрица порядка $n + 1$, элементами которой служат линейные формы от переменных x_0, \dots, x_n , содержит не более $(n + 1)^3$ числовых коэффициентов. Используя симметричность матрицы Гессе, достаточно хранить в памяти меньшее количество чисел. Поэтому алгоритм из теоремы 4 выполняет $O(n^3)$ операций над полем коэффициентов. Алгоритм основан на вычислении определителя матрицы над этим полем. Если кубическая форма f определена над полем рациональных чисел или над конечным расширением этого поля, этот вероятностный алгоритм имеет полиномиальную битовую сложность.

Теорема 5. *Существует вероятностный алгоритм полиномиального времени, который получает на вход рациональное число $\varepsilon > 0$, нечетное целое число $n = 2t + 1$ и заданную списком коэффициентов кубическую форму $f(x_0, \dots, x_n)$. Если форма f определяет гиперповерхность в \mathbb{P}^n , чей сингулярный локус содержит некоторое t -мерное линейное подпространство, то вход всегда отвергается. Однако для почти каждой кубической формы f вход принимается с вероятностью больше $1 - \varepsilon$.*

Доказательство. Сначала вычисляется матрица Гессе формы f . Затем посредством вероятностного алгоритма, основанного на лемме Шварца–Зиппеля [26], проверяется линейная независимость градиента гессиана формы f в случайно выбранных точках.

Пусть $\xi_0 = 1$. Алгоритм выбирает случайные оценки ξ_1, \dots, ξ_n для переменных x_1, \dots, x_n , которые равномерно и независимо выбираются из набора целых чисел из отрезка от нуля до $\lceil n(n + 1)/\varepsilon \rceil$. Поскольку $\xi_0 = 1$, это однородные координаты некоторой точки в \mathbb{P}^n . Для вычисления в выбранной точке частной производной по переменной x_k сделаем подстановку

$$x_j = \begin{cases} \xi_j, & j \neq k \\ \xi_k + y, & j = k \end{cases}$$

и вычислим значение гессиана над полем рациональных функций от одной переменной y . Первая частная производная гессиана по переменной x_k равна коэффициенту при линейном члене по переменной y . Так вычисляется градиент в вы-

бранной точке. Все вычисления повторяются для $n + 1$ независимо от выбранной точки. Вычисления частных производных по разным переменным и в разных точках могут быть выполнены параллельно.

После этого вычисляется определитель матрицы порядка $n + 1$, составленной из первых частных производных гессиана в $n + 1$ точке. Если гессиан не зависит от некоторой переменной, то этот определитель обращается в нуль тождественно. Поэтому если он обращается в нуль в ходе вычислений, то вход отвергается, иначе вход принимается.

Гессиан формы f — это многочлен степени не выше $n + 1$ от переменных x_0, \dots, x_n . Первая частная производная гессиана — однородный многочлен степени не выше n . Определитель, составленный из таких производных в $n + 1$ точке, имеет степень не выше $n(n + 1)$. Согласно лемме Шварца–Зиппеля, вероятность обращения в нуль этого многочлена, если он не обращается в нуль тождественно, меньше ε . \square

Если ограничиться ортогональными преобразованиями координат, можно использовать не только гессиан, но и лапласиан. Рассмотрим гиперповерхность в \mathbb{P}^n , чей сингулярный локус может быть нульмерным, но содержит точки E_0, \dots, E_n , где однородные координаты точки E_k равны нулю за исключением k -й координаты. Примером служит поверхность Кэли, заданная формой $x_0x_1x_2 + x_0x_1x_3 + x_0x_2x_3 + x_1x_2x_3$.

Теорема 6. *Существует алгоритм полиномиального времени, который получает на вход целое число n и заданную списком коэффициентов кубическую форму $f(x_0, \dots, x_n)$. Если форма f определяет особую гиперповерхность в \mathbb{P}^n , чей сингулярный локус содержит образ точек E_0, \dots, E_n при некотором ортогональном преобразовании, то вход всегда отвергается. Однако для почти каждой кубической формы f вход принимается.*

Доказательство. Лапласиан (след матрицы Гессе) инвариантен относительно ортогональных преобразований координат. Поэтому без ограничения общности можно считать, что форма f определяет гиперповерхность с особыми точками E_0, \dots, E_n . Повторяя рассуждения из доказательства теоремы 1, получаем, что форма f мультилинейная, то есть никакая переменная не входит ни во второй ни в третьей степени. Следовательно,

лапласиан тождественно обращается в нуль. Напротив, для почти каждой кубической формы лапласиан равен линейной форме.

Алгоритм отвергает вход, если лапласиан обращается в нуль тождественно, иначе принимает вход. \square

4. ОБСУЖДЕНИЕ

Рассмотрено проверяемое за полиномиальное время вероятностным алгоритмом достаточное условие отсутствия достаточно большого сингулярного локуса на кубической гиперповерхности. В общем случае это не позволяет судить о гладкости этой гиперповерхности. Более того, поскольку гладкость эквивалентна отличию от нуля дискриминанта — неприводимого многочлена степени $(n + 1)2^n$ от коэффициентов кубической формы от $n + 1$ переменных, это свойство в типичном случае не может быть выражено многочленом меньшей степени. Тогда как степень гессиана такой кубической формы не превышает $n + 1$. Поэтому рассмотренный метод не позволяет распознавать сингулярные локусы малых размерностей. Также не известен детерминированный алгоритм проверки обращения гессиана в нуль.

В вероятностных алгоритмах из теорем 4 и 5 границы отрезка, из которого выбираются оценки для переменных, можно расширить так, чтобы случайные числа взаимно однозначно соответствовали наборам случайных битов. При этом вероятность ошибки не увеличится.

Алгоритмы из теорем 4–6 легко преобразовать в генерические алгоритмы, которые либо принимают вход, либо выдают предупреждение о невозможности дать точный ответ, но никогда не отвергают вход и не делают ошибок. Действительно, общая гиперповерхность гладкая, а гиперповерхности с особыми точками определяются формами с обращающимся в нуль дискриминантом. С другой стороны, при использовании псевдослучайной последовательности вместо случайных битов долю входов, которые не допускаются, но вызывают сообщение об отказе от вычисления ответа генерическим алгоритмом, можно сделать достаточно низкой, увеличивая длину отрезка, из которого выбираются случайные числа, или посредством повторения теста.

Автор благодарен М. Спиваковскому (Mark Spivakovsky) за обсуждение темы работы, а также

С.А. Абрамову и А.Б. Батхину за полезные замечания.

СПИСОК ЛИТЕРАТУРЫ

1. *Алаев П.Е., Селиванов В.Л.* Поля алгебраических чисел, вычисляемые за полиномиальное время. I // *Алгебра и логика*. 2019. Т. 58. № 6. С. 673–705. Перевод: *Alaev P.E., Selivanov V.L.* Fields of algebraic numbers computable in polynomial time. I // *Algebra and Logic*. 2020. V. 58. P. 447–469. <https://doi.org/10.1007/s10469-020-09565-0> <https://doi.org/10.33048/alglog.2019.58.601>
2. *Малашонок Г.И.* Система компьютерной алгебры MathPartner // *Программирование*. 2017. № 2. С. 63–71. Перевод: *Malaschonok G.I.* MathPartner computer algebra // *Programming and Computer Software*. 2017. V. 43. № 2. P. 112–118. <https://doi.org/10.1134/S0361768817020086>
3. *Neumann E., Pauly A.* A topological view on algebraic computation models // *Journal of Complexity*. 2018. V. 44. P. 1–22. <https://doi.org/10.1016/j.jco.2017.08.003>
4. *Абрамов С.А.* Лекции о сложности алгоритмов. М.: МЦНМО, 2012. 248 с. ISBN 978-5-4439-0204-3
5. *Рыбалов А.Н.* О генерической амплификации рекурсивно перечислимых множеств // *Алгебра и логика*. 2018. Т. 57, № 4. С. 448–455. Перевод: *Rybalov A.N.* Generic amplification of recursively enumerable sets // *Algebra and Logic*. 2018. V. 57. № 4. P. 289–294. [10.1007/s10469-018-9500-y](https://doi.org/10.1007/s10469-018-9500-y) <https://doi.org/10.17377/alglog.2018.57.403>
6. *Рыбалов А.Н.* О генерической NP-полноте проблемы выполнимости булевых схем // *Прикладная дискретная математика*. 2020. № 47. С. 101–107. <https://doi.org/10.17223/20710410/47/8>
7. *Селиверстов А.В.* О двоичных решениях систем уравнений // *Прикладная дискретная математика*. 2019. № 45. С. 26–32. <https://doi.org/10.17223/20710410/45/3>
8. *Eder C., Faugère J.-C.* A survey on signature-based algorithms for computing Gröbner bases // *Journal of Symbolic Computation*. 2017. V. 80. № 3. P. 719–784. <https://doi.org/10.1016/j.jsc.2016.07.031>
9. *Янович Д.А.* Вычисление инволютивных базисов Гребнера, используя табличное представление полиномов // *Программирование*. 2020. № 2. С. 67–72. Перевод: *Yanovich D.A.* Computation of involutive and Gröbner bases using the tableau representation of polynomials // *Programming and Computer Software*.

2020. V. 46. № 2. P. 162–166.
<https://doi.org/10.1134/S0361768820020115>
<https://doi.org/10.31857/S0132347420020120>
10. *Brzostowski S., Krasinski T., Walewska J.* Arnold's problem on monotonicity of the Newton number for surface singularities // *Journal of the Mathematical Society of Japan*. 2019. V. 71. № 4. P. 1257–1268.
<https://doi.org/10.2969/jmsj/78557855>
 11. *Брюно А.Д.* Асимптотика решений нелинейных систем дифференциальных уравнений // *Доклады АН СССР*. 1962. Т. 143, № 4. С. 763–766. Перевод: *Bryuno A.D.* The asymptotic behavior of solutions of nonlinear systems of differential equations // *Soviet Mathematics, Doklady*. 1962. V. 3. P. 464–467.
 12. *Брюно А.Д.* Алгоритмы решения одного алгебраического уравнения // *Программирование*. 2019. № 1. С. 59–72. Перевод: *Bryuno A.D.* Algorithms for solving an algebraic equation // *Programming and Computer Software*. 2018. V. 44. № 6. P. 533–545.
<https://doi.org/10.1134/S0361768819100013>
<https://doi.org/10.1134/S0132347419010084>
 13. *Брюно А.Д.* О параметризации алгебраической кривой // *Математические заметки*. 2019. Т. 106. № 6. С. 837–847. Перевод: *Bryuno A.D.* On the parametrization of an algebraic curve // *Mathematical Notes*. 2019. V. 106. No 6. P. 885–893.
<https://doi.org/10.1134/S0001434619110233>
<https://doi.org/10.4213/mzml2013>
 14. *Antipova I.A., Mikhalkin E.N., Tsikh A.K.* Singular points of complex algebraic hypersurfaces // *Journal of Siberian Federal University. Mathematics Physics*. 2018. V. 11. № 6. P. 670–679.
<https://doi.org/10.17516/1997-1397-2018-11-6-670-679>
 15. *Гельфанд И.М., Зелевинский А.В., Капранов М.М.* О дискриминантах многочленов от многих переменных // *Функциональный анализ и его приложения*. 1990. Т. 24. № 1. С. 1–4. Перевод: *Gel'fand I.M., Zelevinskii A.V., Kapranov M.M.* Discriminants of polynomials in many variables // *Functional Analysis and Its Applications*. 1990. V. 24. № 1. P. 1–4.
<https://doi.org/10.1007/BF01077912>
 16. *Селиверстов А.В.* О касательных прямых к аффинным гиперповерхностям // *Вестник Удмуртского университета. Математика. Механика. Компьютерные науки*. 2017. Т. 27. № 2. С. 248–256.
<https://doi.org/10.20537/vm170208>
 17. *Rubanov L.I., Seliverstov A.V.* Projective-invariant description of a meandering river // *Journal of Communications Technology and Electronics*. 2017. V. 62. № 6. P. 663–668.
<https://doi.org/10.1134/S1064226917060201>
 18. *van Hoeij M.* An algorithm for computing the Weierstrass normal form. In: *Levelt A.H.M. (ed.) Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC'95*. New York: ACM Press, 1995. P. 90–95.
 19. *Сляднев С.Е., Турлапов В.Е.* Упрощение САД-моделей путем автоматического распознавания и подавления цепочек скруглений // *Программирование*. 2020. № 3. С. 53–65. Перевод: *Slyadnev S.E., Turlapov V.E.* Simplification of CAD models by automatic recognition and suppression of blend chains // *Programming and Computer Software*. 2020. V. 46. № 3. P. 233–243.
<https://doi.org/10.1134/S0361768820030081>
<https://doi.org/10.31857/S0132347420030085>
 20. *Segre B.* A note on arithmetical properties of cubic surfaces // *Journal of the London Mathematical Society*. 1943. V. 18. P. 24–31.
 21. *Kollár J.* Unirationality of cubic hypersurfaces // *Journal of the Institute of Mathematics of Jussieu*. 2002. V. 1. № 3. P. 467–476.
<https://doi.org/10.1017/S1474748002000117>
 22. *Polo-Blanco I., Top J.* A remark on parameterizing nonsingular cubic surfaces // *Computer Aided Geometric Design*. 2009. V. 26, № 8. P. 842–849.
<https://doi.org/10.1016/j.cagd.2009.06.001>
 23. *González-Sánchez J., Polo-Blanco I.* Construction algorithms for rational cubic surfaces // *Journal of Symbolic Computation*. 2017. V. 79. P. 309–326.
<https://doi.org/10.1016/j.jsc.2016.02.010>
 24. *Пан В.Я.* Быстрое умножение матриц и смежные вопросы алгебры // *Математический сборник*. 2017. Т. 208. № 11. С. 90–138. Перевод: *Pan V.Ya.* Fast matrix multiplication and its algebraic neighbourhood // *Sbornik: Mathematics*. 2017. V. 208. № 11. P. 1661–1704.
<https://doi.org/10.1070/SM8833>
<https://doi.org/10.4213/sm8833>
 25. *Malaschonok G.* Recursive matrix algorithms, distributed dynamic control, scaling, stability // *2019 Computer Science and Information Technologies (CSIT), Yerevan, Armenia, 2019*, pp. 112–115.
<https://doi.org/10.1109/CSITechnol.2019.8895255>
 26. *Schwartz J.T.* Fast probabilistic algorithms for verification of polynomial identities // *Journal of the ACM*. 1980. V. 27. № 4. P. 701–717.
<https://doi.org/10.1145/322217.322225>

27. *Gondim R., Russo F.* On cubic hypersurfaces with vanishing hessian // *Journal of Pure and Applied Algebra*. 2015. V. 219. № 4. P. 779–806. <https://doi.org/10.1016/j.jpaa.2014.04.030>
28. *Бибиков П.В.* Классификация тернарных форм с нулевым гессианом // *Известия высших учебных заведений. Математика*. 2011. № 9. С. 99–101. Перевод: *Bibikov P.V.* Classification of ternary forms with zero hessian // *Russian Mathematics*. 2011. V. 55. № 9. P. 83–85. <https://doi.org/10.3103/S1066369X11090118>
29. *Селиверстов А.В.* Симметричные матрицы, элементами которых служат линейные функции // *Журнал вычислительной математики и математической физики*. 2020. Т. 60. № 1. С. 109–115. <https://doi.org/10.31857/S0044466920010147> Перевод: *Seliverstov A.V.* Symmetric matrices whose entries are linear functions // *Computational Mathematics and Mathematical Physics*. 2020. V. 60. P. 102–108. <https://doi.org/10.1134/S0965542520010121>