

# СОДЕРЖАНИЕ

---

---

Номер 2, 2020

---

---

## КОМПЬЮТЕРНАЯ АЛГЕБРА

Семинар по компьютерной алгебре в 2018–2019 гг. <i>С. А. Абрамов, А. А. Боголюбская</i>	3
Нормализация периодической системы Гамильтона <i>А. Д. Брюно</i>	6
Бифуркации периодических решений системы Гамильтона с дискретной группой симметрий <i>А. Б. Батхин</i>	14
Пример модульного расширения системы компьютерной алгебры <i>М. Н. Геворкян, А. В. Королькова, Д. С. Кулябов, Л. А. Севастьянов</i>	30
Алгоритм построения аналога формулы Бине <i>В. И. Кузоватов, А. А. Кытманов, О. И. Кузоватова</i>	38
О машинном доказательстве для арифметики дробей над кольцом с НОД <i>С. Д. Мешвелиани</i>	43
Построение периодического решения уравнений движения обобщенной машины Атвуда с применением компьютерной алгебры <i>А. Н. Прокопеня</i>	53
Дифференциальная геометрия и механика — источник задач для компьютерной алгебры <i>В. Н. Сальников, А. Хамдуни</i>	60
Вычисление инволютивных базисов и базисов Грёбнера используя табличное представление полиномов <i>Д. А. Янович</i>	67

---

---

# CONTENTS

---

---

No. 2, 2020

---

---

## COMPUTER ALGEBRA

Seminar on Computer Algebra in 2018–2019 <i>S. A. Abramov, A. A. Bogolyubskaya</i>	3
Normalization of the Hamilton Periodic System <i>A. D. Bruno</i>	6
Bifurcations of Periodic Solutions of the Hamiltonian System with Discrete Symmetry Group <i>A.B. Batkhin</i>	14
An Example of a Modular Extension of a Computer Algebra System <i>M.N. Gevorkyan, A.V. Korolkova, D.S. Kulyabov, L.A. Sevastyanov</i>	29
An Algorithm for Constructing an Analogue of the Binet Formula <i>V.I. Kuzovator, A.A. Kytmanov, O.I. Kuzovator</i>	36
On a Machine-checked Proof for Fraction Arithmetic over a GCD Domain <i>S.D. Meshveliani</i>	41
Construction of a Periodic Solution of the Equations of Motion Generalized Atwood Machine Using Computer Algebra <i>A. N. Prokopenya</i>	51
Differential Geometry and Mechanics – a Source for Computer Algebra Problems <i>V.N. Salnikov, A. Hamdouni</i>	57
Calculation of Involutive Bases and Gröbner Bases Using Tabular Representation of Polynomials <i>D. A. Yanovich</i>	64

---

---

## КОМПЬЮТЕРНАЯ АЛГЕБРА

УДК 004.421.6

### СЕМИНАР ПО КОМПЬЮТЕРНОЙ АЛГЕБРЕ В 2018–2019 гг.

© 2020 г. С. А. Абрамов<sup>a,\*</sup>, А. А. Боголюбская<sup>b,\*\*</sup>

<sup>a</sup> ВЦ ФИЦ ИУ РАН

119333 Москва, ул. Вавилова, 40, Россия

<sup>b</sup> Объединенный институт ядерных исследований

141980 Дубна Московской области, Россия

\*E-mail: sergeyabramov@mail.ru

\*\*E-mail: abogol@jinr.ru

Поступила в редакцию 05.06.2019 г.

После доработки 12.09.2019 г.

Принята к публикации 12.09.2019 г.

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

DOI: 10.31857/S0132347420020028

#### 1. О СЕМИНАРЕ

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

В 2018–2019 учебном году семинар собирался раз в месяц по третьим средам на факультете вычислительной математики и кибернетики МГУ, а в мае 2019 г. в Дубне, в Объединенном институте ядерных исследований (ОИЯИ) состоялось традиционное заседание, организованное совместно с Лабораторией информационных технологий ОИЯИ.

#### 2. РЕГУЛЯРНЫЕ СОБРАНИЯ СЕМИНАРА

С сентября по апрель были прочитаны следующие доклады<sup>1</sup>. Аннотации доступны на странице семинара <http://www.ccas.ru/sabramov/seminar/doku.php>, где также содержится информация о состоявшихся ранее докладах.

С.В. Пославский (НИЦ “Курчатовский институт” — ИФВЭ, Протвино; stvlpos@mail.ru) *Rings: эффективная JVM библиотека для коммутативной алгебры.*

В.П. Гердт (Лаборатория информационных технологий ОИЯИ, Дубна; gerdt@jinr.ru) *Декомпозиция Томаса систем дифференциальных уравнений и ее реализация в системе Maple.*

В.И. Суковых (DataArt, Факультет компьютерных наук ВГУ, Воронеж; sukovuh@gmail.com) *Компьютерные алгоритмы и символьные вычисления*

*в задаче коэффициентной классификации однородных поверхностей.*

А.А. Панферов (Вычислительный центр им. А.А. Дородницына ФИЦ ИУ РАН; Факультет вычислительной математики и кибернетики МГУ; ast.a\_s@mail.ru) *Алгоритмы символьных вычислений в системах компьютерной алгебры для линейных дифференциальных систем с выделенными неизвестными.*

А.А. Тютюнник (Российский университет дружбы народов; pastya.tyutyunnik@gmail.com) *Символьно-численное исследование векторной модели волнового распространения электромагнитного излучения.*

Н.Н. Осипов (Сибирский федеральный университет, г. Красноярск; nnosipov@gambler.ru) *Алгоритмическая реализация элементарной версии метода Рунге для кубических диофантовых уравнений.*

А.Д. Брюно (Институт прикладной математики им. М.В. Келдыша РАН; bruno@keldysh.ru) *Приведённая нормальная форма периодической системы Гамильтона.*

Р.Р. Гонцов (Институт проблем передачи информации РАН; gontsovrr@gmail.com) *Системы линейных дифференциальных уравнений с малыми коэффициентами: различные виды разрешимости и их проверка.*

А.Б. Батхин (Институт прикладной математики им. М.В. Келдыша РАН; batkhin@gmail.com) *Бифуркации симметричных периодических решений системы Гамильтона.*

Е.В. Зима (Университет Уилфрида Лорие, Ватерлоо, Канада; ezima@wlu.ca) *Факториальные*

<sup>1</sup> Перечень докладов, прочитанных в 1995–2018 гг., опубликован в [1]– [24].

*полиномы в задачах компьютерной алгебры, связанных с символьным суммированием.*

### 3. ДВУХДНЕВНАЯ КОНФЕРЕНЦИЯ В ОБЪЕДИНЕННОМ ИНСТИТУТЕ ЯДЕРНЫХ ИССЛЕДОВАНИЙ (ДУБНА)

По установившейся традиции в мае 2019 г. в Дубне прошло совместное заседание семинаров “Компьютерная алгебра” факультета ВМиК МГУ и ВЦ РАН и семинара Лаборатории информационных технологий ОИЯИ. По существу, это была двухдневная конференция по компьютерной алгебре и ее приложениям.

Вниманию участников были предложены следующие выступления (аннотации доступны на странице <http://compalg.jinr.ru/Dubna2019/index.html>).

С.А. Абрамов (Вычислительный центр им. А.А. Дородницына ФИЦ ИУ РАН; Факультет вычислительной математики и кибернетики МГУ; sergeyabramov@mail.ru) *Когда поиск решений может быть прерван.*

Д.А. Янович (Лаборатория информационных технологий ОИЯИ, Дубна; yan@jinr.ru) *Вычисление инволютивных базисов и базисов Грёбнера на основе табличного представления полиномов.*

Г.К. Гиоргадзе (Институт прикладной математики им. И. Векуа, Тбилиси, Грузия; gia.giorgadze@tsu.ge), Г. Гулагашвили (Тбилисский государственный университет, Грузия) *О проблеме вычислений частных индексов матриц функций.*

М.Д. Малых (Российский университет дружбы народов; malukhmd@yandex.ru), Л.А. Севастьянов (Российский университет дружбы народов; Лаборатория информационных технологий ОИЯИ, Дубна) *О вычислении абелевых интегралов в системе компьютерной алгебры.*

С.Ф. Адлай (Вычислительный центр им. А.А. Дородницына ФИЦ ИУ РАН, Москва; SemjonAdlaj@gmail.com) *Символьное интегрирование эллиптических функций.*

А.П. Крюков, Г.Б. Шпиз (Научно-исследовательский институт ядерной физики им. Д.В. Скобельцына МГУ, Москва; kryukov@theory.sinp.msu.ru) *Автоморфизм цветных графов и каноническое представление мультипликативных выражений со свертками.*

В.В. Корняк (Лаборатория информационных технологий ОИЯИ, Дубна; kornyak@jinr.ru) *Сплетения и моделирование многочастичных квантовых систем.*

К.А. Богданов (Московский физико-технический институт, Долгопрудный), А.В. Сумароков, С.Н. Тимаков (Московский физико-технический институт, Долгопрудный; ПАО РКК “Энергия”, Королёв; sergeytimakov333@gmail.com) *Примене-*

*ние обобщённых полиномов Баттерворта в задачах модального синтеза систем управления.*

Д.В. Диваков, А.А. Тютюнник, М.Д. Малых (Российский университет дружбы народов; divakov\_dv@rudn.university), Л.А. Севастьянов (Российский университет дружбы народов; Лаборатория информационных технологий ОИЯИ, Дубна) *Символьно-численная реализация метода четырех потенциалов отыскания нормальных мод на примере квадратного электромагнитного волновода с прямоугольной вставкой.*

М.Н. Геворкян, А.В. Демидова, А.В. Королькова (Российский университет дружбы народов; gevorgyan\_mn@rudn.university), Д.С. Кулябов (Российский университет дружбы народов; Лаборатория информационных технологий ОИЯИ, Дубна), А.А. Петрова (МГУ) *Тестирование генераторов случайных чисел для систем компьютерной алгебры.*

Н.Н. Васильев (Санкт-Петербургское отделение Математического института им. В.А. Стеклова РАН, Санкт-Петербург), В.С. Дужин, А.Д. Кузьмин (Санкт-Петербургский государственный электротехнический университет “ЛЭТИ”, Санкт-Петербург; vduzhin@gmail.com) *Исследование динамики эволюции случайных значений в алгоритме RSK.*

А.А. Гусев, О. Чулуунбаатар, Г. Чулуунбаатар, В.П. Гердт (Лаборатория информационных технологий ОИЯИ, Дубна; gooseff@jinr.ru), С.И. Виницкий (Лаборатория теоретической физики им. Н.Н. Боголюбова ОИЯИ, Дубна), Л.Л. Хай (Педагогический университет Хошимина, Хошимин, Вьетнам) *Алгоритмы вычисления интерполяционных полиномов Эрмита для метода конечных элементов.*

В.П. Гердт (Лаборатория информационных технологий ОИЯИ, Дубна; gerdt@jinr.ru), Ю.А. Блинков (Саратовский государственный университет им. Н.Г. Чернышевского, Саратов) *Линейные системы ДУЧП, разностные схемы, сильная согласованность и первое дифференциальное приближение.*

М. Спиридонова (Институт математики и информатики БАН, София, Болгария; mspirid@math.bas.bg) *Прикладные возможности программных средств компьютерной алгебры.*

С. Порязов, Е. Саранова, В. Андонов (Институт математики и информатики БАН, София, Болгария; stoyan@cc.bas.bg) *Масштабируемые модели нагрузки антропо-кибернетических систем.*

В. Андонов, С. Порязов, Е. Саранова (Институт математики и информатики БАН, София, Болгария; velin\_andonov@yahoo.com) *Аналитическая модель целостной телекоммуникационной системы с гарантией качества обслуживания, содержащей очереди на этапе коммутации.*

Ю.Г. Палий (Лаборатория информационных технологий ОИЯИ, Дубна; palii@jinr.ru) *Параметры на классах сопряженности специальной линейной группы.*

В.С. Рихвицкий (Лаборатория информационных технологий ОИЯИ, Дубна; rqvtsk@mail.ru) *Логическое программирование и метод форсинга.*

### СПИСОК ЛИТЕРАТУРЫ

1. *Абрамов С.А., Зима Е.В.* Семинар по компьютерной алгебре на факультете вычислительной математики и кибернетики МГУ в 1995–1996 г. // Программирование. 1997. № 1. С. 75–77.
2. *Абрамов С.А., Зима Е.В.* Научно-исследовательский семинар “Компьютерная алгебра” в 1996–1997 г. // Программирование. 1998. № 1. С. 69–72.
3. *Абрамов С.А., Ростовцев В.А.* Семинар по компьютерной алгебре в 1997–1998 г. // Программирование. 1998. № 6. С. 3–7.
4. *Абрамов С.А., Крюков А.П., Ростовцев В.А.* Семинар по компьютерной алгебре в 1998–1999 г. // Программирование. 2000. № 1. С. 8–12.
5. *Абрамов С.А., Крюков А.П., Ростовцев В.А.* Семинар по компьютерной алгебре в 1999–2000 г. // Программирование. 2001. № 1. С. 3–7.
6. *Абрамов С.А., Крюков А.П., Ростовцев В.А.* Семинар по компьютерной алгебре в 2000–2001 г. // Программирование. 2002. № 2. С. 6–9.
7. *Абрамов С.А., Крюков А.П., Ростовцев В.А.* Семинар по компьютерной алгебре в 2001–2002 г. // Программирование. 2003. № 2. С. 3–7.
8. *Абрамов С.А., Еднерал В.Ф., Ростовцев В.А.* Семинар по компьютерной алгебре в 2002–2003 г. // Программирование. 2004. № 2. С. 3–7.
9. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А., Еднерал В.Ф.* Семинар по компьютерной алгебре в 2003–2004 г. // Программирование. 2005. № 2. С. 3–9.
10. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А., Еднерал В.Ф.* Семинар по компьютерной алгебре в 2004–2005 г. // Программирование. 2006. № 2. С. 3–7.
11. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А., Еднерал В.Ф.* Семинар по компьютерной алгебре в 2005–2006 г. // Программирование. 2007. № 2. С. 3–8.
12. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А., Еднерал В.Ф.* Семинар по компьютерной алгебре в 2006–2007 г. // Программирование. 2008. № 2. С. 3–8.
13. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А., Еднерал В.Ф.* Семинар по компьютерной алгебре в 2007–2008 г. // Программирование. 2009. № 2. С. 3–9.
14. “Mathematical Modeling and Computational Physics (CAAP’2009)”. Book of abstracts of the international conference. Dubna, July 7–11. 2009. Dubna. 2009.
15. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А., Еднерал В.Ф.* Семинар по компьютерной алгебре в 2008–2009 г. // Программирование. 2010. № 2. С. 3–8.
16. *Абрамов С.А., Боголюбская А.А., Еднерал В.Ф., Ростовцев В.А.* Семинар по компьютерной алгебре в 2009–2010 г. // Программирование. 2011. № 1. С. 3–8.
17. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А.* Семинар по компьютерной алгебре в 2010–2011 г. // Программирование. 2012. № 2. С. 3–8.
18. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А.* Семинар по компьютерной алгебре в 2011–2012 г. // Программирование. 2013. № 2. С. 3–10.
19. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А.* Семинар по компьютерной алгебре в 2012–2013 г. // Программирование. 2014. № 2. С. 3–11.
20. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А.* Семинар по компьютерной алгебре в 2013–2014 г. // Программирование. 2015. № 2. С. 3–6.
21. *Абрамов С.А., Боголюбская А.А., Ростовцев В.А.* Семинар по компьютерной алгебре в 2014–2015 г. // Программирование. 2016. № 2. С. 4–7.
22. *Абрамов С.А., Боголюбская А.А.* Семинар по компьютерной алгебре в 2015–2016 г. // Программирование. 2017. № 2. С. 3–6.
23. *Абрамов С.А., Боголюбская А.А.* Семинар по компьютерной алгебре в 2016–2017 г. // Программирование. 2018. № 2. С. 3–4.
24. *Абрамов С.А., Боголюбская А.А.* Семинар по компьютерной алгебре в 2017–2018 г. // Программирование. 2019. № 2. С. 3–5.

НОРМАЛИЗАЦИЯ ПЕРИОДИЧЕСКОЙ  
СИСТЕМЫ ГАМИЛЬТОНА

© 2020 г. А. Д. Брюно\*

Институт прикладной математики им. М.В. Келдыша РАН  
125047 Москва, Миусская пл., д. 4, Россия

\*E-mail: abruno@keldysh.ru

Поступила в редакцию 31.08.2019 г.

После доработки 12.09.2019 г.

Принята к публикации 20.10.2019 г.

Сначала напомним нормальную форму вблизи стационарного решения автономной системы Гамильтона. Затем рассматриваются линейные периодические системы Гамильтона. Для них найдены нормальные формы функций Гамильтона в комплексном и вещественном случаях. Обнаружена специфика вещественного случая в ситуации параметрического резонанса. Затем найдены нормальные формы функций Гамильтона нелинейных периодических систем. Посредством дополнительного канонического преобразования координат такая нормальная форма всегда сводится к автономной системе Гамильтона, которая сохраняет все малые параметры и симметрии исходной системы. Ее локальным семействам неподвижных точек соответствуют семейства периодических решений исходной системы. Аналогичная теория строится вблизи периодического решения автономной системы. Все преобразования алгоритмичны и могут быть реализованы в системе компьютерной алгебры.

DOI: 10.31857/S0132347420020053

## 1. ВВЕДЕНИЕ

Резонансная нормальная форма автономной системы Гамильтона вблизи стационарного решения, учитывающая только собственные числа матрицы  $A$  ее линейной части и без ограничений на эту матрицу  $A$ , была введена в [1], § 12. Оказалось, что она эквивалентна системе Гамильтона с меньшим числом степеней свободы.

Позже была введена слегка более простая сверхрезонансная нормальная форма, которая учитывала жордановы клетки нормальной формы матрицы  $A$ . Но эти дополнительные упрощения не позволяли дополнительно понизить число степеней свободы [8].

Теория резонансной нормальной формы подробно изложена в гл. I книги [2], и здесь она кратко упоминается в разделе 2. В гл. II книги [2] изложена аналогичная теория резонансной нормальной формы для периодической системы Гамильтона. Однако там имеются две недоделки: плохо изложен случай параметрического резонанса и нормальная форма не приводится к автономной системе. Здесь исправляются эти упущения в разделах 3 и 4 соответственно. В разделе 5 эта теория переносится на окрестность периодического решения автономной системы Гамильтона.

Все описанные в работе преобразования носят алгоритмический характер и могут быть вполне реализованы в различных системах компьютерной алгебры, как коммерческих, например, Wolfram Mathematica [9] или Maplesoft Maple [10], так и открытых, например, SymPy [11] или MathPartner [12].

2. НОРМАЛЬНАЯ ФОРМА ВБЛИЗИ  
СТАЦИОНАРНОГО РЕШЕНИЯ

Рассмотрим систему Гамильтона

$$\dot{\xi}_j = \frac{\partial \gamma}{\partial \eta_j}, \quad \dot{\eta}_j = -\frac{\partial \gamma}{\partial \xi_j}, \quad j = 1, \dots, n \quad (1)$$

с  $n$  степенями свободы в окрестности неподвижной точки

$$\xi = (\xi_1, \dots, \xi_n) = 0, \quad \eta = (\eta_1, \dots, \eta_n) = 0. \quad (2)$$

Если функция Гамильтона  $\gamma(\xi, \eta)$  аналитична в этой точке, то она разлагается в степенной ряд

$$\gamma(\xi, \eta) = \sum \gamma_{\mathbf{p}\mathbf{q}} \xi^{\mathbf{p}} \eta^{\mathbf{q}}, \quad (3)$$

где  $\mathbf{p} = (p_1, \dots, p_n)$ ,  $\mathbf{q} = (q_1, \dots, q_n) \in \mathbb{Z}^n$ ,  $\mathbf{p}, \mathbf{q} \geq 0$ ,  $\xi^{\mathbf{p}} = \xi_1^{p_1} \xi_2^{p_2} \dots \xi_n^{p_n}$ . Поскольку точка (2) — неподвижная, то разложение (3) начинается с квадратич-

ных членов. Им соответствует линейная часть системы (1).

Собственные числа ее матрицы разбиваются на пары

$$\lambda_{j+n} = -\lambda_j, \quad j = 1, \dots, n.$$

Пусть  $\lambda = (\lambda_1, \dots, \lambda_n)$ . Канонические замены координат

$$(\xi, \eta) \rightarrow (x, y) \quad (4)$$

сохраняют гамильтоновость системы.

**Теорема 1.** *Существует каноническое формальное преобразование (4), приводящее систему (1) к нормальной форме*

$$\dot{x}_j = \frac{\partial g}{\partial y_j}, \quad \dot{y}_j = -\frac{\partial g}{\partial x_j}, \quad j = 1, \dots, n, \quad (5)$$

где ряд

$$g(x, y) = \sum g_{pq} x^p y^q$$

содержит только резонансные члены  $s \langle p - q, \lambda \rangle = 0$ , а квадратичная часть  $g_2(x, y)$  имеет свою нормальную форму (так что матрица линейной части системы является гамильтоновым аналогом жордановой нормальной формы). Здесь  $\langle p, \lambda \rangle = p_1 \lambda_1 + \dots + p_n \lambda_n$  – скалярное произведение.

Если  $\lambda \neq 0$ , то нормальная форма (5) эквивалентна системе с меньшим числом степеней свободы и дополнительными параметрами. При нормализующем преобразовании (4) сохраняются малые параметры и линейные автоморфизмы

$$(\xi, \eta) \rightarrow (\tilde{\xi}, \tilde{\eta}), \quad t \rightarrow \tilde{t}.$$

Локальные семейства периодических решений систем (1), (5) удовлетворяют системе уравнений

$$\frac{\partial g}{\partial y_j} = \lambda_j x_j a, \quad \frac{\partial g}{\partial x_j} = \lambda_j y_j a, \quad j = 1, \dots, n,$$

где  $a$  – свободный параметр.

Для вещественной исходной системы (1) коэффициенты  $g_{pq}$  комплексной нормальной формы (5) удовлетворяют специальным соотношениям вещественности, и при стандартной канонической линейной замене координат  $(x, y) \rightarrow (X, Y)$  система (5) переходит в вещественную систему.

Имеется несколько способов вычисления коэффициентов  $g_{pq}$  нормальной формы (5). Наиболее простой описан в книге Журавлёва, Петрова, Шундерюка [3].

### 3. НОРМАЛИЗАЦИЯ ЛИНЕЙНОЙ СИСТЕМЫ ГАМИЛЬТОНА

#### 3.1. Линейная система

Рассмотрим линейную систему

$$\frac{d\zeta}{d\psi} = A(\psi)\zeta, \quad (6)$$

где вектор  $\zeta = (\zeta_1, \dots, \zeta_m)$ ,  $A(\psi)$  – матрица, аналитически зависящая от  $\psi$ . После замены координат

$$\zeta = B(\psi)z \quad (7)$$

система (6) перейдет в систему

$$\frac{dz}{d\psi} = B^{-1} \left( AB - \frac{dB}{d\psi} \right) z. \quad (8)$$

Пусть теперь система (6) гамильтонова:

$$\frac{d\xi_j}{d\psi} = \frac{\partial \gamma}{\partial \eta_j}, \quad \frac{d\eta_j}{d\psi} = -\frac{\partial \gamma}{\partial \xi_j}, \quad j = 1, \dots, n, \quad (9)$$

т.е.  $m = 2n$ ,  $\zeta = (\xi, \eta) = (\xi_1, \dots, \xi_n, \eta_1, \dots, \eta_n)$ ,  $A(\psi) = J\Gamma(\psi)$ , где  $\Gamma(\psi)$  – симметрическая матрица,

$J = \begin{pmatrix} 0 & E_n \\ -E_n & 0 \end{pmatrix}$  и функция Гамильтона  $\gamma = \frac{1}{2} \langle \zeta, \Gamma(\psi)\zeta \rangle$ .

Здесь  $E_n$  – единичная  $n \times n$ -матрица и  $\langle \cdot, \cdot \rangle$  – скалярное произведение.

Если преобразование (7) каноническое, т.е.

$$B^*(\psi)JB(\psi) = \delta J, \quad \delta = \text{const} \quad (10)$$

(звездочка – символ транспонирования матрицы), то система (8) также гамильтонова с функцией Гамильтона

$$g = \frac{1}{2\delta} \langle z, B^* \Gamma B z \rangle + \frac{1}{2\delta} \left\langle z, B^* J \frac{dB}{d\psi} z \right\rangle \stackrel{\text{def}}{=} \frac{1}{2} \langle z, G z \rangle, \quad (11)$$

т.е.  $G = \delta^{-1} B^* \Gamma B + \delta^{-1} B^* J dB/d\psi$ ,  $z = (x, y)$ .

Рассмотрим теперь систему Гамильтона (9), в которой матрица  $A(\psi) = J\Gamma(\psi)$  имеет по  $\psi$  период  $2\pi$ , т.е.  $A(\psi + 2\pi) = A(\psi)$ . Посредством линейной канонической замены координат (7), (10) с  $2\pi$ -периодической матрицей  $B(\psi)$  постараемся получить гамильтониан (11) наиболее простого вида. Пусть  $Z(\psi)$  – фундаментальная матрица решений системы (6). Тогда

$$Z(\psi + 2\pi) = Z(\psi)N,$$

где  $N$  – постоянная матрица,  $\det N \neq 0$ . Для системы Гамильтона она каноническая.

Если для матрицы  $N$  существует представление

$$N = \exp(2\pi JL), \quad (12)$$

где  $L$  – постоянная симметрическая матрица, то, согласно § 1 гл. II книги [2],  $L = B_1^* G B_1$ , где  $B_1$  – постоянная каноническая матрица и  $G$  – нормальная форма матрицы  $L$ . Таким образом, преобразование (7) с

$$B(\psi) = Z(\psi) \exp(-\psi JG)$$

приводит систему Гамильтона (9) к нормальной форме

$$\frac{dz}{d\psi} = JGz, \quad G = \text{const}, \quad (13)$$

с функцией Гамильтона  $g = \frac{1}{2}\langle z, Gz \rangle$ . Однако представление (12) имеется не для всякой канонической матрицы  $N$  (см. Вильямсон [4]).

Пусть  $\nu_1, \dots, \nu_{2n}$  – собственные числа канонической матрицы  $N$ . Вместе с числом  $\nu_j = b$  среди них есть и число  $b^{-1}$ . Более того, элементарные делители матрицы  $\nu E - N$  обладают следующими свойствами:

- если  $b \neq \pm 1$  и имеется ровно  $k$  элементарных делителей  $(\nu - b)^l$ , то имеется ровно  $k$  элементарных делителей  $(\nu - b^{-1})^l$ ;
- если  $b = \pm 1$  и  $l$  нечетно, то элементарный делитель  $(\nu - b)^l$  встречается четное число раз.

### 3.2. Комплексная нормальная форма

Для комплексной системы (6) матрица  $N$  – комплексная. Неприводимые над полем комплексных чисел  $\mathbb{C}$  элементарные делители матрицы  $\nu E - N$  относятся к одному из следующих четырех случаев:

- C1)  $(\nu - b)^l$  и  $(\nu - b^{-1})^l$ ,  $b \neq \pm 1$ ;
- C2)  $(\nu - b)^l$  и  $(\nu - b)^l$ ,  $b = \pm 1$ ,  $l$  – нечетное;
- C3)  $(\nu - 1)^{2l}$ ;
- C4)  $(\nu + 1)^{2l}$ .

Посредством постоянной канонической замены координат  $\zeta$  матрицу  $\Gamma(\psi)$  можно привести к такому блочному виду, что каждому из перечисленных случаев отвечает своя четверка блоков порядка  $l$ , а вне блоков стоят нули. Поэтому достаточно рассмотреть каждый из этих случаев в предположении  $l = n$ .

В случаях C1)–C3) существует представление (12); при этом элементарные делители  $(\lambda - a)^l$  матрицы  $\lambda E - JL$  относятся к случаям C1)–C3) п. 1.Б гл. I книги [2], где

$$a = \frac{1}{2\pi} \text{Ln } b = \frac{1}{2\pi} \ln |b| + \frac{i}{2\pi} \arg b + im$$

и  $m$  – любое целое число; а именно:

- в случае C1)

$$G = \begin{pmatrix} 0 & C^* \\ C & 0 \end{pmatrix},$$

где  $C$  – жорданова клетка порядка  $l$ :

$$\begin{pmatrix} a & 0 & 0 & \dots & 0 & 0 & 0 \\ \varepsilon & a & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \varepsilon & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & \varepsilon & a & 0 \\ 0 & 0 & 0 & \dots & 0 & \varepsilon & a \end{pmatrix},$$

т.е.

$$g_2 = a \sum_{j=1}^l x_j y_j + \varepsilon \sum_{j=1}^{l-1} x_j y_{j+1}; \quad (14)$$

- случай C2) с  $b = 1$  относится к случаю C1) [2] с  $a = im$ ;
- случай C2) с  $b = -1$  относится к случаю C1) [2] с  $a = im + \frac{i}{2}$ ;
- в случае C3)

$$G = \begin{pmatrix} 0 & C^* \\ C & \sigma \Delta \end{pmatrix},$$

где  $C$  – жорданова клетка порядка  $l$  с  $a = 0$ ,  $\sigma = \pm 1$  и диагональная матрица  $\Delta = \{1, 0, \dots, 0\}$ , т.е.

$$g_2 = \varepsilon \sum_{j=1}^{l-1} x_j y_{j+1} + \frac{1}{2} \sigma y_l^2; \quad (15)$$

- в случае C4) представления (12) нет, и комплексная нормальная форма

$$\frac{dz}{d\psi} = JG(\psi)z$$

имеет

$$G = \begin{pmatrix} 0 & C^* \\ C & \sigma \Delta e^{i\psi} \end{pmatrix},$$

где  $C$  – жорданова клетка порядка  $l$  с  $a = im + \frac{i}{2}$ ,  $\sigma = \pm 1$  [5], т.е.

$$g_2 = a \sum_{j=1}^l x_j y_j + \varepsilon \sum_{j=1}^{l-1} x_j y_{j+1} + \frac{1}{2} \sigma y_l^2 e^{i\psi}. \quad (16)$$

Рассмотрим теперь удвоенные случаи C3) и C4).

C3\*) Двум элементарным делителям  $(\nu - 1)^{2l'}$  и  $(\nu - 1)^{2l'}$  можно поставить в соответствие нормальную форму (13) случая C2) с  $a = im$  и произвольным целым  $m$  (только теперь  $l = 2l' - \text{четно}$ ).

C4\*) Двум элементарным делителям  $(\nu + 1)^{2l'}$  и  $(\nu + 1)^{2l'}$  можно поставить в соответствие нормальную форму случая C1) с

$$a = (2\pi)^{-1} \text{Ln}(-1) = im + \frac{i}{2}.$$



Итак, посредством комплексной замены (7), где  $B(\psi)$  – каноническая  $2\pi$ -периодическая матрица, исходная функция Гамильтона

$$\gamma = \frac{1}{2} \langle \zeta, \Gamma(\psi)\zeta \rangle$$

приводится к нормальной форме, являющейся суммой форм вида (14), (15), (16). Она является постоянной, если каждый элементарный делитель вида  $(v + 1)^{2l}$  встречается четное число раз среди элементарных делителей матрицы  $vE - N$ . Вильямсон [4] доказал, что это условие не только достаточно, но и необходимо для комплексной приводимости.

### 3.3. Вещественные системы

Для вещественной системы (9) матрица  $N$  является вещественной. Поэтому элементарные делители матрицы  $vE - N$  обладают следующими свойствами. Пусть элементарный делитель  $(v - b)^l$  имеется точно  $k$  раз.

- Если число  $b$  комплексное, т.е.  $\text{Re}b \cdot \text{Im}b \neq 0$ , и  $|b| \neq 1$ , то элементарные делители  $(v - \bar{b})^l$ ,  $(v - b^{-1})^l$  и  $(v - \bar{b}^{-1})^l$  также имеются точно  $k$  раз.

- Если число  $b$  вещественное или единичного модуля,  $b \neq \pm 1$ , то  $(v - b^{-1})^l$  имеется точно  $k$  раз.

- Если  $b = \pm 1$  и  $l$  нечетно, то  $k$  должно быть четным.

Здесь черта сверху означает комплексное сопряжение.

Поэтому элементарные делители матрицы  $vE - N$  относятся к одному из следующих восьми случаев:

R1)  $(v - b)^l(v - \bar{b})^l$  и  $(v - b^{-1})^l(v - \bar{b}^{-1})^l$ ,  $b \in \mathbb{C}$ ,  $\text{Re}b \cdot \text{Im}b \neq 0$ ,  $|b| \neq 1$ ;

R2)  $(v - b)^l$  и  $(v - b^{-1})^l$ ,  $b \in \mathbb{R}$ ,  $b > 0$ ,  $b \neq 1$ ;

R3)  $(v - b)^l(v - \bar{b})^l$ ,  $|b| = 1$ ,  $b \neq \pm 1$ ;

R4)  $(v - 1)^l$  и  $(v - 1)^l$ ,  $l$  – нечетно;

R5)  $(v - 1)^{2l}$ ;

R6)  $(v + 1)^l$  и  $(v + 1)^l$ ,  $l$  – нечетно;

R7)  $(v - b)^l$  и  $(v - b^{-1})^l$ ,  $b \in \mathbb{R}$ ,  $b < 0$ ,  $b \neq -1$ ;

R8)  $(v + 1)^{2l}$ .

Посредством вещественной постоянной канонической замены координат матрицу  $\Gamma(\psi)$  можно привести к такому блочному виду, что каждому из перечисленных случаев отвечает своя группа блоков, а вне этих блоков стоят нули. Поэтому достаточно рассмотреть каждый из этих случаев в

предположении, что он исчерпывает матрицу  $N$ . В случаях R1)–R7) существует представление (12) с вещественной матрицей  $L$ ; при этом элементарные делители  $(\lambda - a)^l$  матрицы  $\lambda E - JL$  относятся к случаям R1)–R5) п. 1. В гл. I книги [2] соответственно, где

$$a = \frac{1}{2\pi} \text{Ln} b = \frac{1}{2\pi} \ln b + im.$$

При этом число  $\ln b$  однозначно определяется по  $b$ , а целое число  $m$  надо вычислять дополнительно следующим образом.

Вычисляется любое решение  $\zeta(\psi)$  линейной подсистемы вида (6), относящейся к одному из случаев R1)–R7). Количество колебаний каждой из его координат на периоде  $2\pi$  – это и есть число  $m$ . Если сделать дополнительное каноническое преобразование

$$\begin{aligned} \tilde{x}_j &= x_j \exp(-im\psi), \\ \tilde{y}_j &= y_j \exp(im\psi), \quad j = 1, \dots, l, \end{aligned}$$

то получим собственное число  $\tilde{\lambda} : 0 \leq \text{Im}\tilde{\lambda} \leq 1$ . При этом в случаях R3) и R5) имеется дополнительный вещественный инвариант  $\sigma = \pm 1$ .

Итак, в случаях R1)–R7) имеется постоянная комплексная нормальная форма гамильтониана

$$g_2 = \frac{1}{2} \langle z, Gz \rangle,$$

которая переводится в вещественную нормальную форму

$$f_2 = \frac{1}{2} \langle Z, FZ \rangle$$

с помощью стандартного канонического преобразования

$$Z = Qz, \quad \det Q = 1.$$

При этом подстановка

$$\bar{z} = Pz,$$

где  $2n$ -матрица  $P = \bar{Q}^{-1}Q$ , сохраняет гамильтониан. Конкретный вид матриц  $Q$  и  $P$  для каждого из случаев R1)–R7) описан в главе I книги [2]. Так, в случаях R2)–R7) либо

$$x_j = X_j = \bar{x}_j, \quad y_j = Y_j = \bar{y}_j, \quad j = 1, \dots, l, \quad (17)$$

либо

$$\begin{aligned} x_j &= \frac{1}{\sqrt{2i}} (iX_j - Y_j) = i\bar{y}_j, \\ y_j &= \frac{1}{\sqrt{2i}} (iX_j + Y_j) = i\bar{x}_j, \quad j = 1, \dots, l. \end{aligned} \quad (18)$$

**Теорема 2.** *Комплексная запись вещественной нормальной формы в случае R8) – это система (16) с таким стандартным преобразованием:*

$$\begin{aligned} x_j &= \frac{1}{2i} \{X_j[1 + ie^{-i\psi}] - Y_j[i + e^{-i\psi}]\}, \\ y_j &= \frac{1}{2i} \{X_j[i - e^{i\psi}] + Y_j[-1 + ie^{i\psi}]\}. \end{aligned} \quad (19)$$

Здесь  $\bar{x}_j = ix_j e^{i\psi}$ ,  $\bar{y}_j = -iy_j e^{-i\psi}$ ,  $\varepsilon_j = i$ ,  $\sigma = \pm i$ ,  $j = 1, \dots, l$ .

Тогда

$$\bar{G}(\psi) = G(\psi)$$

и гамильтониан нормальной формы (16) имеет вид

$$\begin{aligned} g_2 &= \lambda \sum_{j=1}^l x_j y_j + i \sum_{j=1}^{l-1} x_j y_{j+1} \pm \\ &\pm \frac{i}{2} [X_1^2 + Y_1^2 + (X_1^2 - Y_1^2) \sin \psi - 2X_1 Y_1 \cos \psi]. \end{aligned}$$

#### 4. НЕЛИНЕЙНАЯ НОРМАЛЬНАЯ ФОРМА

##### 4.1. Нелинейная нормализация

Рассмотрим систему Гамильтона с  $n$  степенями свободы

$$\frac{d\xi_j}{d\psi} = \frac{\partial \gamma}{\partial \eta_j}, \quad \frac{d\eta_j}{d\psi} = -\frac{\partial \gamma}{\partial \xi_j}, \quad j = 1, \dots, n, \quad (20)$$

где  $\gamma$  – степенной ряд по  $\xi$ ,  $\eta$  с  $2\pi$ -периодичности по  $\psi$  коэффициентами, который разлагается в сходящийся ряд Пуассона

$$\gamma = \sum_m \gamma_{\mathbf{p}\mathbf{q}\mathbf{m}} \xi^{\mathbf{p}} \eta^{\mathbf{q}} e^{im\psi}, \quad (21)$$

начинающийся с квадратичных членов  $g_2$  по  $\xi$ ,  $\eta$ .

Сделаем линейное каноническое преобразование  $\xi, \eta \rightarrow \mathbf{x}, \mathbf{y}$ , которое  $2\pi$ -периодично по  $\psi$  и приводит квадратичную часть гамильтониана системы (20) к комплексной нормальной форме, являющейся суммой частей вида (14), (15), (16). Тогда на главной диагонали матрицы  $JG$  стоят ее собственные числа  $\lambda_1, \dots, \lambda_n, -\lambda_1, \dots, -\lambda_n$ . Обозначим  $\lambda = (\lambda_1, \dots, \lambda_n)$ . Гамильтониан (21) примет вид

$$g(\mathbf{x}, \mathbf{y}, \psi) = \sum g_{\mathbf{p}\mathbf{q}\mathbf{m}} \mathbf{x}^{\mathbf{p}} \mathbf{y}^{\mathbf{q}} e^{im\psi}. \quad (22)$$

Назовем его нормальной формой, если

1) его форма  $g_2$  является нормальной формой (14), (15), (16),

2) в разложении (21) имеются только резонансные члены, для которых

$$\langle \mathbf{p} - \mathbf{q}, \lambda \rangle + im = 0. \quad (23)$$

Здесь  $\langle \cdot, \cdot \rangle$  – скалярное произведение.

**Теорема 3.** Для гамильтониана (22) существует формальная каноническая замена координат  $\mathbf{x}, \mathbf{y}$ ,  $\psi \rightarrow \mathbf{u}, \mathbf{v}, \varphi$ :

$$\mathbf{z} = (\mathbf{x}, \mathbf{y}) = \mathbf{w} + \mathbf{b}(\mathbf{w}, \varphi), \quad \psi = \varphi + b_{2n+1}(\mathbf{w}, \varphi),$$

$2\pi$ -периодическая по  $\varphi$ , которая переводит гамильтониан (22) в нормальную форму

$$h(\mathbf{u}, \mathbf{v}, \varphi) = \sum h_{\mathbf{p}\mathbf{q}\mathbf{m}} \mathbf{u}^{\mathbf{p}} \mathbf{v}^{\mathbf{q}} \exp(im\varphi) \quad (24)$$

со свойством (23).

Доказательство см. в гл. I и II книги [2].

**Теорема 4.** Каноническое преобразование  $u_j = \tilde{u}_j \exp(-i\text{Im}\lambda_j \varphi)$ ,  $v_j = \tilde{v}_j \exp(i\text{Im}\lambda_j \varphi)$ ,  $j = 1, \dots, n$ , приводит нормальную форму гамильтониана (24) к постоянному степенному ряду

$$\tilde{h}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) \sum \tilde{h}_{\mathbf{p}\mathbf{q}\mathbf{m}} \tilde{\mathbf{u}}^{\mathbf{p}} \tilde{\mathbf{v}}^{\mathbf{q}}, \quad (25)$$

где

$$\langle \mathbf{p} - \mathbf{q}, \text{Re}\lambda \rangle = 0, \quad \langle \mathbf{p} - \mathbf{q}, \text{Im}\lambda \rangle = -m, \quad (26)$$

$\mathbf{p}, \mathbf{q}, m$  – целочисленны,  $\mathbf{p}, \mathbf{q} \geq 0$ . При этом  $\tilde{h}_{\mathbf{p}\mathbf{q}\mathbf{m}} = h_{\mathbf{p}\mathbf{q}\mathbf{m}}$ , если  $\|\mathbf{p}\| + \|\mathbf{q}\| \geq 3$ , квадратичные члены имеют вид

$$\tilde{h}_2(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) = \frac{1}{2} \langle \tilde{\mathbf{w}}, \tilde{G}\tilde{\mathbf{w}} \rangle,$$

где матрица  $J\tilde{G} = JG - \text{Im}\Lambda$  с диагональной матрицей  $\Lambda = \{\lambda, -\lambda\}$ , и почти отсутствуют квадратичные члены по  $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) = \tilde{\mathbf{w}}$ .

Доказательство сводится к проверке равенства (10), которое здесь очевидно. Здесь  $\|\mathbf{p}\| = p_1 + p_2 + \dots + p_n$ .

Для исходного вещественного гамильтониана (21) комплексные координаты  $\mathbf{z}$  связаны с вещественными координатами  $\mathbf{Z} = (\mathbf{X}, \mathbf{Y})$  стандартным преобразованием, состоящим из замен (17), (18), (19), а координаты  $\mathbf{w}$  и  $\tilde{\mathbf{w}}$  связаны этим же преобразованием с соответствующими вещественными координатами  $\mathbf{W}$  и  $\tilde{\mathbf{W}}$  [2].

Таким образом, приходим к автономной системе Гамильтона с  $n$  степенями свободы.

##### 4.2. Малые параметры

Пусть исходный гамильтониан разлагается в степенной ряд по малым параметрам  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_s)$ . По теореме 5.1 гл. I книги [2] при нормализующем преобразовании малые параметры не меняются. Поэтому получаем автономный гамильтониан (25), (26), где коэффициенты  $\tilde{h}_{\mathbf{p},\mathbf{q},\mathbf{m}}$  суть степенные ряды по малым параметрам  $\boldsymbol{\mu}$ . При  $\boldsymbol{\mu} = 0$  эти коэффициенты с  $\|\mathbf{p}\| + \|\mathbf{q}\| = 1$  равны нулю, а с  $\|\mathbf{p}\| + \|\mathbf{q}\| = 2$  соответствуют (23). Но при  $\boldsymbol{\mu} \neq 0$  это не обязательно.

Для системы, соответствующей гамильтониану (25), (26), можно вычислить семейства неподвижных точек вблизи точки  $\tilde{\mathbf{u}} = \tilde{\mathbf{v}} = 0$ ,  $\boldsymbol{\mu} = 0$ . Это делается с помощью степенной геометрии (книга [6]). Им соответствуют семейства периодических

решений исходной периодической системы Гамильтона. Вообще укорочения функции Гамильтона и системы Гамильтона изучены в гл. IV книги [6]. Они не всегда совпадают. Примеры таких вычислений см. в [7].

4.3. Линейные канонические автоморфизмы

Пусть исходная система (20) обладает линейным каноническим автоморфизмом

$$\zeta^* = M\tilde{\zeta}^*, \quad \psi = \theta\tilde{\psi},$$

где  $M$  – постоянная матрица  $2n \times 2n$  и  $\theta = \text{const}$ . Согласно теореме 2.3 гл. I книги [2] приведенная нормальная форма (25), (26) также обладает соответствующим линейным каноническим автоморфизмом. Впрочем, она может иметь дополнительные автоморфизмы, которые не имеют соответствия в исходной системе.

5. НОРМАЛЬНАЯ ФОРМА В ОКРЕСТНОСТИ ПЕРИОДИЧЕСКОГО РЕШЕНИЯ

5.1. Локальные координаты

Пусть вещественная функция  $\tilde{\gamma}(\tilde{\xi}, \tilde{\eta})$  аналитична в некоторой области  $\mathcal{C}$  вещественного пространства  $\mathbb{R}^{2n+2}$  с координатами

$$\tilde{\xi} = (\tilde{\xi}_1, \dots, \tilde{\xi}_{n+1}), \quad \tilde{\eta} = (\tilde{\eta}_1, \dots, \tilde{\eta}_{n+1}).$$

Тогда через каждую точку области  $\mathcal{C}$  проходит одна траектория (она же решение)  $\xi(t), \eta(t)$  системы Гамильтона.

$$\dot{\xi}_j = \frac{\partial \tilde{\gamma}}{\partial \tilde{\eta}_j}, \quad \dot{\eta}_j = -\frac{\partial \tilde{\gamma}}{\partial \tilde{\xi}_j}, \quad j = 1, \dots, n+1. \quad (27)$$

Траектории образуют фазовое пространство этой системы. Пусть у системы (27) есть периодическое решение  $M \subset \mathcal{C}$  с периодом  $T = T(M)$ . В дальнейшем под  $\mathcal{U}$  будем понимать достаточно малую окрестность решения  $M, \mathcal{U} \subset \mathcal{C}$ . В окрестности  $\mathcal{U}$  существуют такие аналитические функции  $\xi = (\xi_1, \dots, \xi_n), \eta = (\eta_1, \dots, \eta_n), \rho, \psi$  от  $\tilde{\xi}, \tilde{\eta}$ , что

1) траектория  $M$  определяется равенствами  $\xi = \eta = 0, \rho = 0$ ;

2) функция  $\psi$  является циклической (угловой) по mod  $2\pi$ ;

3) окрестность  $\mathcal{U}$  является косым произведением  $(2n+1)$ -мерного шара на цикл  $\psi \in [0, 2\pi]$ ;

4) координаты  $\xi, \rho$  и  $\eta, \psi$  являются канонически сопряженными, так что в окрестности  $\mathcal{U}$  система (27) в этих координатах гамильтонова:

$$\begin{aligned} \dot{\xi}_j &= \frac{\partial \gamma}{\partial \eta_j}, & \dot{\eta}_j &= -\frac{\partial \gamma}{\partial \xi_j}, & j &= 1, \dots, n, \\ \dot{\rho} &= \frac{\partial \gamma}{\partial \psi}, & \dot{\psi} &= -\frac{\partial \gamma}{\partial \rho}. \end{aligned} \quad (28)$$

Гамильтониан  $\gamma$  в окрестности  $\mathcal{U}$  является  $2\pi$ -периодическим по  $\psi$  и разлагается в сходящийся ряд Тейлора

$$\gamma = \sum \gamma_{pq}(\psi) \xi^p \eta^q \rho^l, \quad (29)$$

где целочисленные  $p \geq 0, q \geq 0, l \geq 0, \xi^p = \xi_1^{p_1} \dots \xi_n^{p_n}$ , аналитические функции  $\gamma_{pq}(\psi)$  имеют по  $\psi$  период  $2\pi$  и разлагаются в ряды Фурье. Поскольку на решении  $M$  имеем  $\dot{\xi} = 0, \dot{\eta} = 0, \dot{\rho} = 0$ , то на  $M$  система (28) принимает вид

$$0 = \gamma_{0e_j 0}(\psi), \quad 0 = \gamma_{e_j 0 0}(\psi), \quad j = 1, \dots, n, \quad (30)$$

$$0 = \frac{d\gamma_{000}}{d\psi}, \quad \dot{\psi} = -\gamma_{001}(\psi), \quad (31)$$

где  $e_j$  –  $j$ -й единичный вектор. Из (31), в частности, следует, что  $\gamma_{000}(\psi) = \text{const}$ . Так как гамильтониан можно задавать с точностью до постоянного слагаемого, положим  $\gamma_{000} = 0$ .

Поскольку  $M$  – периодическое решение, то на нем нет неподвижных точек; следовательно,  $\gamma_{001}(\psi) \neq 0$  при вещественных  $\psi$ . Пусть  $\frac{1}{\lambda_0}$  – среднее значение функции  $\frac{1}{\gamma_{001}(\psi)}$ , тогда

$$\frac{\lambda_0}{\gamma_{001}(\psi)} = 1 + \sum_{m=1}^{\infty} (a_m \cos m\psi + b_m \sin m\psi).$$

Положим

$$\begin{aligned} g(\psi) &= \int \frac{\lambda_0 d\psi}{\gamma_{001}(\psi)} = \\ &= \psi + \sum_{m=1}^{\infty} \frac{1}{m} (a_m \sin m\psi - b_m \cos m\psi) \end{aligned}$$

и сделаем каноническую замену

$$\tilde{\rho} = \frac{\gamma_{001}(\psi)}{\lambda_0} \rho, \quad \tilde{\psi} = g(\psi). \quad (32)$$

Тогда на  $M$  уравнение для  $\tilde{\psi}$  есть  $\dot{\tilde{\psi}} = -\lambda_0$ , т.е.  $\tilde{\psi} = -\lambda_0 t + \text{const}$ . Поскольку  $M$  – периодическое решение с периодом  $T$ , а  $\tilde{\psi}$  имеет период  $2\pi$ , то  $-\lambda_0 = \frac{2\pi}{T}$ , где значение  $T$  может быть любого знака.

В дальнейшем будем считать, что преобразование (32) уже сделано, и будем опускать тильды над  $\rho$  и  $\psi$ . Если теперь разложить гамильтониан  $\gamma$  в ряд вида (29), где вместо  $\rho$  и  $\psi$  стоят  $\tilde{\rho}$  и  $\tilde{\psi}$ , то, опуская тильды, вместо (31) получим равенства

$$\gamma_{000}(\Psi) = 0, \quad -\gamma_{001}(\Psi) = \frac{2\pi}{T}. \quad (33)$$

Итак, определены свободный и линейный члены разложения (29).

### 5.2. Линейная нормализация

Пусть  $\zeta = (\xi, \eta)$ . Для разложения (29) через  $\gamma_{kl}(\zeta, \rho, \Psi)$  будем обозначать однородную форму по  $\zeta$  порядка  $k$ , содержащую  $\rho$  только в виде множителя  $\rho^l$ . Согласно (30) и (33), в гамильтониане (29) члены наименьшего порядка суть

$$\gamma_{20} + \gamma_{01} = \frac{2\pi}{T} \left( \frac{1}{2} \langle \zeta, \Gamma(\Psi)\zeta \rangle - \rho \right). \quad (34)$$

Пусть преобразование (7) каноническое в координатах  $\zeta$ , т.е. удовлетворяет равенству (10); если одновременно с ним сделать преобразование

$$\rho = \delta\tilde{s} - \frac{1}{2} \left\langle \mathbf{z}, B^* J \frac{dB}{d\Psi} \mathbf{z} \right\rangle, \quad \Psi = \psi,$$

то получим комплексную нормальную форму (см. (15)):

$$g_{20} + g_{01} = \frac{2\pi}{T} \left( \frac{1}{2} \langle \mathbf{z}, G\mathbf{z} \rangle - \tilde{s} \right), \quad G = \text{const}. \quad (35)$$

При этом на главной диагонали матрицы  $JG$  стоят ее собственные числа  $\lambda_1, \dots, \lambda_n, -\lambda_1, \dots, -\lambda_n$ .

### 5.3. Нелинейная нормализация

В результате канонической замены (7), (34) гамильтониан (29) примет вид  $g(\mathbf{x}, \mathbf{y}, \tilde{s}, \psi) = \delta^{-1} \gamma(\xi, \eta, \rho, \psi)$ . Разложим его в ряд Пуассона с  $\mathbf{z} = (\mathbf{x}, \mathbf{y})$

$$g(\mathbf{x}, \mathbf{y}, \tilde{s}, \psi) = \sum g_{\mathbf{p}\mathbf{q}lm} \mathbf{x}^{\mathbf{p}} \mathbf{y}^{\mathbf{q}} \tilde{s}^l e^{im\psi}. \quad (36)$$

Этот ряд сходится абсолютно для достаточно малых  $|\mathbf{x}|, |\mathbf{y}|, |\tilde{s}|, |\text{Im}\psi|$ .

Теперь будем искать наиболее простой гамильтониан

$$h(\mathbf{u}, \mathbf{v}, \tilde{r}, \varphi) = h_{20} + h_{01} + \sum h_{\mathbf{p}\mathbf{q}lm} \mathbf{u}^{\mathbf{p}} \mathbf{v}^{\mathbf{q}} \tilde{r}^l e^{im\varphi}, \quad (37)$$

к которому приводится гамильтониан (36) посредством нелинейной канонической замены координат  $\mathbf{x}, \mathbf{y}, \tilde{s}, \psi \rightarrow \mathbf{u}, \mathbf{v}, \tilde{r}, \varphi$ :

$$\begin{aligned} \mathbf{z} &= \mathbf{w} + \mathbf{b}(\mathbf{w}, \tilde{r}, \varphi), \\ \tilde{s} &= \tilde{r} + b_{2n+1}(\mathbf{w}, \tilde{r}, \varphi), \\ \psi &= \varphi + b_{2n+2}(\mathbf{w}, \tilde{r}, \varphi), \end{aligned} \quad (38)$$

где  $\mathbf{w} = (\mathbf{u}, \mathbf{v})$ .

Пусть форма  $g_{20} + g_{01}$  есть (36). Тогда на главной диагонали матрицы  $JG$  стоят ее собственные числа  $\lambda_1, \dots, \lambda_n, -\lambda_1, \dots, -\lambda_n$ . Обозначим  $\lambda = (\lambda_1, \dots, \lambda_n)$ . Га-

мильтониан (38) назовем *комплексной нормальной формой*, если:

1) его форма  $h_{20} + h_{01}$  является нормальной формой (35);

2) в разложении (37) имеются только такие члены, для которых

$$\langle \mathbf{p} - \mathbf{q}, \lambda \rangle + im = 0.$$

**Теорема 5** ([2]). *Для гамильтониана (36) существует формальное преобразование (38) к нормальной форме (37).*

### 5.4. Приведенная нормальная форма

**Теорема 6.** *Каноническое преобразование  $u_j = \tilde{u}_j \exp(-i\text{Im}\lambda_j \varphi)$ ,  $v_j = \tilde{v}_j \exp(i\text{Im}\lambda_j \varphi)$ ,  $j = 1, \dots, n$ , приводит нормальную форму гамильтониана (37) к постоянному степенному ряду*

$$\tilde{h}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}, \tilde{r}) = \sum \tilde{h}_{\mathbf{p}\mathbf{q}lm} \tilde{\mathbf{u}}^{\mathbf{p}} \tilde{\mathbf{v}}^{\mathbf{q}} \tilde{r}^l, \quad (39)$$

$$\langle \mathbf{p} - \mathbf{q}, \text{Re}\lambda \rangle = 0, \quad \langle \mathbf{p} - \mathbf{q}, \text{Im}\lambda \rangle = -m, \quad (40)$$

$\mathbf{p}, \mathbf{q}, l, m$  — целочисленны,  $\mathbf{p}, \mathbf{q}, l \geq 0$ . При этом  $\tilde{h}_{\mathbf{p}\mathbf{q}lm} = h_{\mathbf{p}\mathbf{q}lm}$ , если  $\|\mathbf{p}\| + \|\mathbf{q}\| \geq 3$ , квадратичные члены имеют вид

$$\tilde{h}_{20} + \tilde{h}_{01} = \frac{2\pi}{T} \left( \frac{1}{2} \langle \tilde{\mathbf{w}}, \tilde{G}\tilde{\mathbf{w}} \rangle - \tilde{r} \right),$$

где матрица  $J\tilde{G} = JG - i\text{Im}\Lambda$  с диагональной матрицей  $\Lambda = \{\lambda, -\lambda\}$  и почти отсутствуют квадратичные члены по  $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) = \tilde{\mathbf{w}}$ . Здесь  $\|\mathbf{p}\| = p_1 + p_2 + \dots + p_n$ .

Таким образом, приходим к автономной системе Гамильтона (39), (40) с  $n$  степенями свободы, двумя дополнительными уравнениями

$$\tilde{r} = \frac{\partial \tilde{h}}{\partial \varphi} = 0, \quad \dot{\varphi} = -\frac{\partial \tilde{h}}{\partial \tilde{r}} \quad (41)$$

и дополнительным малым параметром  $\tilde{r}$ , которую назовем *приведенной нормальной формой*. Она позволяет изучать бифуркации семейств периодических решений системы (20) в окрестности резонансного периодического решения.

Для исходного вещественного гамильтониана (29) комплексные координаты  $\mathbf{z}$  связаны с вещественными координатами  $\mathbf{Z} = (\mathbf{X}, \mathbf{Y})$  стандартным преобразованием (17)–(19), а координаты  $\mathbf{w}$  и  $\tilde{\mathbf{w}}$  связаны этим же преобразованием с соответствующими вещественными координатами  $\mathbf{W}$  и  $\tilde{\mathbf{W}}$ . Координаты  $\tilde{r}$  и  $\varphi$  вещественны.

Другие свойства приведенной нормальной формы (сохранение малых параметров и линейных автоморфизмов) аналогичны таким же свойствам нормальной формы (38), описанным в пп. 2.Г и 2.Д главы II книги [2].

## 5.5. Понижение числа степеней свободы

Пусть  $k$  – число линейно независимых решений  $\mathbf{p} \in \mathbb{Z}^n$  системы уравнений

$$\langle \mathbf{p}, \operatorname{Re} \lambda \rangle = 0, \quad \langle \mathbf{p}, \operatorname{Im} \lambda \rangle = 0.$$

Тогда приведенная нормальная форма (38)–(41) каноническим преобразованием сводится к автономной системе Гамильтона с  $k + 1$  степенью свободы и с  $n - k$  параметрами согласно § 3 главы I книги [2].

## 5.6. Локальные семейства неподвижных точек

Здесь справедливо все, что было сказано в конце подраздела 4.2, только надо учитывать еще один малый параметр.

## СПИСОК ЛИТЕРАТУРЫ

1. Брюно А.Д. Аналитическая форма дифференциальных уравнений. II // Труды Моск. матем. общества. 1972. Т. 26. С. 199–226
2. Брюно А.Д. Ограниченная задача трех тел. М.: Наука, 1990. 296 с.
3. Журавлев В.Ф., Петров А.Г., Шундерюк М.М. Избранные задачи гамильтоновой механики. М.: ЛЕНАНД, 2015. 291 с.
4. Williamson J. The exponential representation of canonical matrices // Amer. Math. J. 1939. V. 61. № 4. P. 897–911.
5. Брюно А.Д. Нормальная форма периодической системы Гамильтона с  $n$  степенями свободы // Препринты ИПМ им. М.В. Келдыша. 2018. № 223. 15 с. <https://doi.org/10.20948/prepr-2018-223> URL: <http://library.keldysh.ru/preprint.asp?id=18-223>
6. Брюно А.Д. Степенная геометрия в алгебраических и дифференциальных уравнениях. М.: Физматлит, 1998. 288 с.
7. Брюно А.Д. Нормальная форма системы Гамильтона с периодическим возмущением // Препринты ИПМ им. М.В. Келдыша. 2019. № 57. 27 с. <https://doi.org/10.20948/prepr-2019-57> URL: <http://library.keldysh.ru/preprint.asp?id=19-57>
8. Baider A., Sanders J.A. Unique normal forms: the nilpotent Hamiltonian case // Journal of Differential Equations. 1991. V. 92. P. 282–304.
9. Wolfram S. The Mathematica Book. Wolfram Media, Inc. 2003. 1488 p.
10. Thompson I. Understanding Maple. Cambridge University Press, 2016. 228 p.
11. Meurer A. et al. SymPy: symbolic computing in Python // PeerJ Computer Science. 2017. V. 3. P. e103. <https://doi.org/10.7717/peerj-cs.103>
12. Малашинок Г.И. Система компьютерной алгебры MathPartner // Программирование. 2017. № 2. С. 63–71.

БИФУРКАЦИИ ПЕРИОДИЧЕСКИХ РЕШЕНИЙ СИСТЕМЫ  
ГАМИЛЬТОНА С ДИСКРЕТНОЙ ГРУППОЙ СИММЕТРИЙ© 2020 г. А. Б. Батхин<sup>а,в,\*</sup><sup>а</sup> *Институт прикладной математики им. М.В. Келдыша РАН  
125047 Москва, Миусская пл., д. 4, Россия*<sup>в</sup> *Московский физико-технический институт (государственный университет)  
141701 г. Долгопрудный Московской области, Институтский переулок, д. 9, Россия*

\*E-mail: batkhin@gmail.com

Поступила в редакцию 25.08.2019 г.

После доработки 18.09.2019 г.

Принята к публикации 18.09.2019 г.

Рассматривается автономная система Гамильтона с двумя степенями свободы, инвариантная относительно четверной группы Клейна линейных канонических автоморфизмов расширенного фазового пространства системы. Строится последовательность симплектических преобразований матрицы монодромии симметричного периодического решения системы. С помощью этих преобразований исследуется структура и перестройка фазового потока в окрестности этого решения. Показано, что перестройки, соответствующие кратному увеличению периода, происходят иначе для решений, обладающих двойной симметрией, чем для решений с одной симметрией. Приведен пример критических периодических решений семейства двояко симметричных орбит плоской круговой задачи Хилла. Большинство трудоемких аналитических выкладок в работе выполнены с использованием пакетов для вычисления базисов Грёбнера и работы с полиномиальными идеалами системы компьютерной алгебры Maple.

DOI: 10.31857/S0132347420020041

## 1. ВВЕДЕНИЕ

Восходящий к Пуанкаре подход к исследованию фазового потока неинтегрируемой системы обыкновенных дифференциальных уравнений состоит в поиске некоторого “скелета”, состоящего из иерархической структуры инвариантных многообразий различных размерностей. Наиболее простыми компонентами этой структуры являются положения равновесия и семейства периодических решений. Один из способов вычисления периодических решений использует метод регулярных возмущений. Он состоит в том, что в исходной системе неким образом выделяется интегрируемая подсистема, а полная система рассматривается как ее некоторое малое возмущение. Для интегрируемой подсистемы находится множество общих решений, с помощью которых с учетом возмущения вычисляются порождающие решения — предельные положения решений возмущенной задачи при стремлении возмущения к нулю.

Хорошо известно, что интегрируемость системы ОДУ связана с наличием достаточного числа первых интегралов этой системы, что, в силу теоремы Эмми Нётер, означает наличие некоторого

количества непрерывных групп симметрий исходных уравнений. В общем случае возмущение интегрируемой системы разрушает часть этих групп инвариантов, превращая полную систему в неинтегрируемую. Однако зачастую в силу специфики возмущения от непрерывной группы симметрий может остаться та или иная дискретная группа. Как следствие, должны существовать решения возмущенной системы уравнений, инвариантные относительно такой группы. Например, рассматривая ограниченную задачу трех тел (ОЗТТ) при значении массового параметра  $\mu$  близкого к нулю как возмущение интегрируемой задачи Кеплера вращающейся системе координат, имеем ситуацию, при которой возмущение “разрушает” интеграл площадей, индуцированный группой  $SO(1)$  поворотов конфигурационного пространства задачи, но, тем не менее, сохраняет дискретную группу порядка 2 линейных канонических преобразований расширенного фазового пространства. В следствие этого ОЗТТ обладает периодическими решениями, орбиты которых симметричны относительно оси  $OX$ . Другие примеры систем Гамильтона с различными дискретными группами симметрий см., например, в [1].

Наличие даже дискретной симметрии уравнений ОДУ часто позволяет существенно упростить процедуру поиска периодических решений. В данной работе рассматривается ситуация, когда некоторая автономная система Гамильтона с двумя степенями свободы, обладающая единственным первым интегралом, — интегралом энергии — имеет дополнительно дискретную группу линейных канонических автоморфизмов расширенного фазового пространства системы. Эта группа изоморфна четверной группе Клейна  $K_4$ , две образующие которой суть обратимые преобразования. Исследуется структура матрицы монодромии невырожденного двояко симметричного периодического решения. В критических решениях семейства рассматриваются различные сценарии ветвления семейства.

Работа состоит из введения, пяти разделов и списка литературы. В разделе 2 приведено описание представлений четверной группы Клейна и классификация решений системы Гамильтона с учетом симметрии уравнений. В разделе 3 рассматриваются особенности структуры матрицы монодромии симметричного периодического решения, а в разделе 4 рассматривается преобразование матрицы монодромии к виду, удобному для ее последующего исследования в критических случаях. Раздел 5 посвящен анализу возможных сценариев ветвления семейства двояко симметричных решений. Наконец, в разделе 6 приводится пример численного исследования семейства  $f_3$  двояко симметричных периодических решений плоской круговой задачи Хилла и его поведения в критических решениях. Отметим, что для численного интегрирования системы Гамильтона и ее уравнений в вариациях используются пакеты TAYLOR [2] и TIDES [3], основанные на алгоритмах автоматического дифференцирования с применением системы компьютерной алгебры Wolfram Mathematica.

В работе интенсивно используются пакеты LinearAlgebra, Groebner и PolynomialIdeals системы компьютерной алгебры Maple [4] для проведения некоторых вычислений, связанных с преобразованиями симплектических матриц. Эти вычисления проводились по следующей схеме:

- 1) рассматривается матрица  $\mathbf{M}$  в символьном виде;
- 2) строится некоторый полиномиальный идеал  $\mathcal{J}$ , описывающий то или иное условие на элементы матрицы  $\mathbf{M}$ , например, симплектичность или ортогональность матрицы, существование собственного числа  $\pm 1$  и т. п.;
- 3) вычисляется базис Грёбнера  $\mathcal{GB}\mathcal{J}$  этого идеала для соответствующего лексикографического порядка;
- 4) выполняются полилинейные преобразования матрицы  $\mathbf{M}$ , после чего полиномиальные

элементы новой матрицы приводятся к канонической форме по модулю идеала  $\mathcal{GB}\mathcal{J}$ .

Такого рода вычисления можно выполнить в любой системе компьютерной алгебры, позволяющей вычислять базис Грёбнера, например в открытой системе SymPy [5] или проприетарных Maplesoft Maple или Wolfram Mathematica [6].

Предварительные результаты работы докладывались на международных конференциях: Polynomial Computer Algebra'2019, 15–20 апреля в Санкт-Петербурге [7], Computer Algebra, 17–21 июня в Москве [8] и опубликованы в препринте [9].

## 2. ДИСКРЕТНАЯ ГРУППА АВТОМОРФИЗМОВ СИСТЕМЫ ГАМИЛЬТОНА

Рассмотрим автономную систему Гамильтона с двумя степенями свободы с гамильтонианом  $H(\mathbf{z})$  в случае общего положения, где  $\mathbf{z} = (\mathbf{x}, \mathbf{y})$  — вектор фазового пространства  $\mathbb{M} \equiv \mathbb{R}^4$ , т.е. система канонических уравнений

$$\dot{\mathbf{z}} = \mathbf{J}H(\mathbf{z}), \quad \mathbf{J} = \begin{pmatrix} 0 & \mathbf{E}^2 \\ -\mathbf{E}^2 & 0 \end{pmatrix}, \quad (2.1)$$

имеет единственный первый интеграл

$$H(\mathbf{z}) = h. \quad (2.2)$$

Положим, что система (2.1) инвариантна относительно конечной группы линейных канонических автоморфизмов расширенного фазового пространства  $\tilde{\mathbb{M}} \equiv \mathbb{R} \times \mathbb{M}$ , изоморфной четверной группе Клейна  $V_4$ .

Четверная группа Клейна  $V_4$  (или диэдральная группа  $D_2 \equiv \mathbb{Z}_2 \otimes \mathbb{Z}_2$ ) — это группа порядка 4, являющаяся наименьшей нециклической абелевой группой, а также 2-примарной абелевой группой. Она задается либо соотношениями

$$\langle g_1^2 = g_2^2 = (g_1 \circ g_2)^2 = id \rangle, \quad (2.3)$$

где  $g_{1,2}$  — два генератора группы, либо соотношениями

$$\langle r^2 = s^2 = id, s^{-1} \circ r \circ s = r^{-1} \rangle,$$

где  $r$  — поворот, а  $s$  — симметрия. Известно, что любую группу диэдра  $D_n$  можно представить как группу линейных преобразований плоскости, состоящей из  $n$ -операторов поворота  $r_k$ ,  $k = 0, \dots, n - 1$ , на угол  $2\pi k/n$ , и  $n$ -операторов симметрии  $s_k$ ,  $k = 0, \dots, n - 1$ , относительно прямой, составляющей угол  $\pi k/n$  с осью абсцисс. Для  $D_2$  выберем

два генератора  $g_1 \cong s_0$  и  $g_2 \cong s_1$ , тогда  $g_3 = g_1 \circ g_2 = g_2 \circ g_1 = r_1$ . Для конфигурационного пространства  $\mathbb{R}^2$  автоморфизмы  $g_1$  и  $g_2$  задаются соответственно матрицами  $S_0 = \{1, -1\}$  и  $S_1 = \{-1, 1\}$ . Следовательно, с учетом того, что автоморфизмы  $g_{1,2}$  обращают время, в расширенном фазовом пространстве  $\mathbb{M}$  они имеют вид

$$\begin{aligned} g_1 : (t, \mathbf{z}) &\rightarrow (-t, \mathbf{G}_1 \mathbf{z}), & \mathbf{G}_1 &= \{+1, -1, -1, +1\}, \\ g_2 : (t, \mathbf{z}) &\rightarrow (-t, \mathbf{G}_2 \mathbf{z}), & \mathbf{G}_2 &= \{-1, +1, +1, -1\}. \end{aligned}$$

Автоморфизм  $g_3$  в этом случае является центральной симметрией фазового пространства  $\mathbb{M}$ :

$$g_3 : (t, \mathbf{z}) \rightarrow (t, \mathbf{G}_3 \mathbf{z}), \quad \mathbf{G}_3 = \{-1, -1, -1, -1\}.$$

Итак, каждый из автоморфизмов  $g_i$ ,  $i = 1, 2, 3$  в силу соотношений (2.3), является *инволюцией*, но при этом преобразования  $g_1$  и  $g_2$ , в свою очередь, будут *обратимыми*, поскольку

$$\frac{d}{dt} g_i(\mathbf{z}) = -\mathbf{J} \text{grad} H(g_i(\mathbf{z})), \quad i = 1, 2.$$

Инвариантные множества  $\Sigma_{1,2} \equiv \{\mathbf{z} | g_{1,2}(\mathbf{z}) = \mathbf{z}\}$  преобразований  $g_{1,2}$  являются двумерными координатными плоскостями – плоскостями симметрии преобразований

$$\Sigma_1 : \{x_2 = y_1 = 0\}, \quad \Sigma_2 : \{x_1 = y_2 = 0\}. \quad (2.4)$$

Инвариантное множество преобразования  $g_3$  состоит из одной точки – начала координат.

**Определение 1** ([10]). Пусть  $\mathbf{o}(\mathbf{z}_0)$  – орбита системы (2.1):

$$\mathbf{o}(\mathbf{z}_0) = \{\mathbf{z}(t, \mathbf{z}_0) | t \in \mathbb{R}\}.$$

Тогда орбита  $\mathbf{o}(\mathbf{z}_0)$  называется  $g$ -симметричной, если она является инвариантным множеством преобразования  $g$ :  $g(\mathbf{o}(\mathbf{z}_0)) = \mathbf{o}(\mathbf{z}_0)$ .

**Теорема 1** ([10]). Пусть  $\mathbf{o}(\mathbf{z}_0)$  – орбита системы (2.1) с обратимой симметрией  $g$ . Тогда

(1) орбита  $\mathbf{o}(\mathbf{z}_0)$  является  $g$ -симметричной тогда и только тогда, когда она пересекает соответствующее инвариантное множество

$$\Sigma(g) = \{\mathbf{z} | g(\mathbf{z}) = \mathbf{z}\}.$$

При этом орбита пересекает  $\Sigma(g)$  не более чем в двух точках и целиком содержится в  $\Sigma(g^2)$ .

(2) орбита  $\mathbf{o}(\mathbf{z}_0)$  пересекает  $\Sigma(g)$  в точности в двух точках тогда и только тогда, когда она – орбита периодического решения (но не положения рав-

новесия) и симметрична относительно автоморфизма  $g$ .

Каждое решение  $\mathbf{z}(t, \mathbf{z}_0)$  системы (2.1) с начальным условием  $\mathbf{z}(0) = \mathbf{z}_0$  принадлежит к одному из следующих классов в зависимости от типа симметрии.

- **Несимметричные** решения, фазовые траектории которых неинвариантны относительно любого из преобразований  $g_i$ ,  $i = 1, 2, 3$ .

- **Однократно симметричные** решения, фазовые траектории которых инвариантны лишь относительно какого-либо одного преобразования  $g_i$ ,  $i = 1, 2, 3$ .

- **Двойко симметричные** решения, фазовые траектории которых инвариантны относительно любого из преобразований  $g_i$ ,  $i = 1, 2, 3$ .

Заметим, что если решение  $\mathbf{z}(t, \mathbf{z}_0)$  симметрично относительно двух обратимых автоморфизмов  $g_1$  и  $g_2$ , то оно симметрично относительно преобразования центральной симметрии  $g_3$ . Обратное, вообще говоря, неверно, т.е. среди однократно симметричных решений следует выделять обратимо симметричные и центрально симметричные.

Пусть  $\mathbf{z}(t, \mathbf{z}_0)$  – периодическое решение с периодом  $T > 0$ , т.е. решение удовлетворяющее условию периодичности для любого  $t$ :

$$\mathbf{z}(t + T, \mathbf{z}_0) = \mathbf{z}(t, \mathbf{z}_0).$$

Периодические решения автономной системы Гамильтона не являются изолированными, а образуют семейства (см. [11, Lemma 6.4.1]). Размерности этих семейств равны числу независимых первых интегралов системы Гамильтона. Для автономной неинтегрируемой системы Гамильтона семейства периодических решений являются обычно однопараметрическими, где параметром может служить значение, например, величина  $h$  интеграла (2.2). Итак, каждое семейство периодических решений описывается парой функций  $\mathbf{z}_0(\lambda)$  и  $T(\lambda)$ , где  $\lambda$  – параметр семейства. Здесь, следуя [12, п. 2.5], под  $T(\lambda^*)$  понимается не минимальный период конкретного решения, соответствующего значению параметра  $\lambda^*$  на семействе, а предел минимальных периодов решений семейства при стремлении  $\lambda \rightarrow \lambda^*$ . Это позволяет рассматривать функцию  $T(\lambda)$  как непрерывную функцию параметра  $\lambda$  даже для ситуаций, когда одно семейство пересекает другое с решениями кратных периодов.

Как известно (см., например, [13]), семейства периодических решений могут быть либо замкнутыми, либо натурально ограниченными, когда одна или более количественных характеристик решения – наибольшее расстояние от орбиты до начала координат, значение интеграла  $h$  или пе-



риода  $T$  – неограниченно возрастают. При продолжении семейства по параметру  $\lambda$  эти и другие характеристики (например, индекс устойчивости  $S$ ) периодического решения меняются гладко вдоль семейства, в то время как симметрия решения является глобальным инвариантом семейства. Семейство решений может пересекаться с другим семейством, но оно всегда может быть единственным образом продолжено через точку пересечения.

### 3. ФАЗОВЫЙ ПОТОК СИСТЕМЫ ГАМИЛЬТОНА ВБЛИЗИ ПЕРИОДИЧЕСКОГО РЕШЕНИЯ

#### 3.1. Свойства фазового потока

Динамика фазового потока  $\Phi(t)$  в окрестности решения  $\mathbf{z}(t)$  описывается матрицей  $\mathbf{Z}(t, \mathbf{z})$ , которая есть решение задачи Коши уравнения в вариациях Пуанкаре

$$\dot{\mathbf{Z}} = \mathbf{J} \text{Hess}H(\mathbf{z})\mathbf{Z}, \quad \mathbf{Z}(0) = \mathbf{Z}_0, \quad \text{rang}\mathbf{Z}_0 = 4, \quad (3.1)$$

где  $\text{Hess}H(\mathbf{z})$  – гессиан функции  $H(\mathbf{z})$ , вычисляемый вдоль решения  $\mathbf{z}(t, \mathbf{z}_0)$ . Решение  $\mathbf{Z}(t, \mathbf{Z}_0)$  называется *матрицантом* и, если матрица  $\mathbf{Z}_0$  симплектическая, т.е. удовлетворяет матричному тождеству

$$\mathbf{Z}_0^t \mathbf{J} \mathbf{Z}_0 = \mathbf{J}, \quad (3.2)$$

то такой же будет и матрицант  $\mathbf{Z}(t)$  для любого  $t$  [14, Гл. II, п. 3.4].

Здесь и далее значок  $^t$  означает операцию транспонирования матрицы или вектора.

**Определение 2.** Матрицей монодромии периодического решения  $(\mathbf{Z}_0, T)$  называется матрица  $\mathbf{M} = \mathbf{Z}_0^{-1} \cdot \mathbf{Z}(T, \mathbf{Z}_0)$ , а ее собственные числа  $\rho_i, i = 1, \dots, 4$ , называются *мультипликаторами*.

В дальнейшем, если не оговорено противное, будем считать, что  $\mathbf{Z}_0 \equiv \mathbf{E}^4$ , т.е. матрица  $\mathbf{Z}(t, \mathbf{Z}_0)$  есть нормированная фундаментальная матрица системы в вариациях (3.1).

Приведем здесь основные свойства матрицы монодромии  $\mathbf{M}$ , которые будут использоваться в дальнейшем.

1) Пусть  $\mathbf{Z}(t)$  – фундаментальная матрица системы (3.1), тогда

$$\mathbf{Z}(t + T) = \mathbf{Z}(t)\mathbf{M}. \quad (3.3)$$

2) В силу симплектичности матрицы  $\mathbf{M}$  ее характеристический многочлен  $P_\lambda(\mathbf{M})$  возвратный [15]. Для симплектической матрицы  $\mathbf{M}$  обратная к ней вычисляется по формуле

$$\mathbf{M}^{-1} = \mathbf{J}^t \mathbf{M}^t \mathbf{J} = -\mathbf{J} \mathbf{M}^t \mathbf{J}. \quad (3.4)$$

3) В силу вещественности системы (3.1) и симплектичности  $\mathbf{M}$  ее мультипликаторы попарно взаимно комплексно-сопряженные и взаимно обратные.

4) Вектор-столбец фазовой скорости  $\mathbf{v}_0 \equiv \mathbf{J} \text{grad}H(\mathbf{z}_0)$  является правым собственным вектором матрицы  $\mathbf{M}$  с соответствующим мультипликатором  $\rho_1 = +1$  [11, Lemma 6.4.1]. Это следует непосредственно из дифференцирования по  $t$  тождества, задающего групповую структуру  $\mathbf{z}(\tau, \mathbf{z}(t, \mathbf{z}_0)) = \mathbf{z}(\tau + t, \mathbf{z}_0)$ , при подстановке в него значений  $\tau = T$  и  $t = 0$ :

$$\mathbf{M} \cdot \dot{\mathbf{z}}(T) = \mathbf{M} \cdot \mathbf{v}_0 = \mathbf{v}_0.$$

5) Мультипликатор  $\rho_1 = \rho_2 = +1$  имеет алгебраическую кратность 2.

6) Поскольку система (2.1) автономна и обладает первым интегралом (2.2), то, дифференцируя по  $\mathbf{z}_0$  тождество  $H(\mathbf{z}(T, \mathbf{z}_0)) = H(\mathbf{z}_0)$ , получим, что вектор-строка  $\tilde{\mathbf{v}}_1 \equiv \text{grad}H(\mathbf{z}_0)$  является левым собственным вектором матрицы монодромии  $\mathbf{M}$  [11, Lemma 6.5.1]:

$$\text{grad}H(\mathbf{z}_0)\mathbf{M} = H(\mathbf{z}_0).$$

7) Характеристический многочлен  $P_\lambda(\mathbf{M})$  матрицы монодромии  $\mathbf{M}$  раскладывается на множители

$$P_\lambda(\mathbf{M}) = (\lambda - 1)^2(\lambda^2 - 2S\lambda + 1),$$

где

$$S = \text{Tr}\mathbf{M}/2 - 1 \quad (3.5)$$

– индекс устойчивости периодического решения  $(\mathbf{z}_0, T)$ .

Величина  $S$  позволяет определить характер устойчивости периодического решения в линейном приближении. А именно,

- если  $|S| > 1$ , то  $\rho_{3,4} \in \mathbb{R}$ ,  $\rho_3 = 1/\rho_4$  и решение  $\mathbf{z}(t, \mathbf{z}_0)$  неустойчиво;
- если  $|S| < 1$ , то  $\rho_3 = \bar{\rho}_4$ ,  $|\rho_3| = |\rho_4| = 1$  и решение  $\mathbf{z}(t, \mathbf{z}_0)$  орбитально устойчиво;
- критический случай  $|S| = 1$  требует дополнительного исследования.

Особо интересен случай, когда индекс устойчивости  $S = \cos(2\pi p/q)$ , где  $p, q \in \mathbb{N}$ . Его впервые рассмотрел А. Пуанкаре в своем знаменитом трактате *Les méthodes nouvelles de la mécanique céleste* [16, Гл. XXX]. Этот случай соответствует появлению в окрестности исходного периодического решения так называемых *периодических решений второго рода* по Пуанкаре с периодом  $T' = qT$ .

### 3.2. Вычисление матрицы монодромии

В общем случае вычисление матрицы монодромии  $\mathbf{M}$  требует одновременного интегрирования двух систем уравнений: исходной системы (2.1), дающей интегральную кривую  $\mathbf{Z}(t, \mathbf{Z}_0)$ , и системы уравнений в вариациях (3.1) для четырех начальных условий. Как отмечено в [17], подставляя в (3.3) значение  $t = -T/2$ , получим  $\mathbf{Z}(T/2) = \mathbf{Z}(-T/2)\mathbf{M}$ , откуда

$$\mathbf{M} = \mathbf{Z}^{-1}(-T/2)\mathbf{Z}(T/2) = -\mathbf{J}\mathbf{Z}^1(-T/2)\mathbf{J}\mathbf{Z}(T/2).$$

При этом матрицы  $\mathbf{Z}(-T/2)$  и  $\mathbf{Z}(T/2)$  можно вычислять параллельно. Численное интегрирование систем (2.1) и (3.1) на половине периода дает более точные результаты, чем интегрирование на всем периоде, особенно для сильно неустойчивых решений.

Рассмотрим вычисление матрицы монодромии симметричного периодического решения. Если периодическое решение симметрично относительно обратимого преобразования  $g_i$ ,  $i = 1, 2$ , то, согласно теореме 1, начальное условие  $\mathbf{z}_0$  можно выбрать на инвариантном множестве  $\Sigma(g_i)$ , и тогда через полпериода  $\mathbf{z}(T/2, \mathbf{z}_0) \in \Sigma(g_i)$ . Инвариантность системы (2.1) относительно  $g_i$  означает инвариантность гамильтониана  $H(\mathbf{z})$  и равносильна матричному тождеству

$$\text{Hess}H(\mathbf{z}(-t)) = \mathbf{J}\mathbf{G}_i\mathbf{J}\text{Hess}H(\mathbf{z}(t))\mathbf{G}_i^{-1}, \quad i = 1, 2.$$

Это означает, что для системы (2.1) выполнено условие  $t$ -инвариантности [14], а матрицы  $\mathbf{Z}(-t)$  и  $\mathbf{Z}(t)$  для таких систем связаны соотношением  $\mathbf{Z}(-t) = \mathbf{G}_i^{-1}\mathbf{Z}(t)\mathbf{G}_i$ , в частности,  $\mathbf{Z}(-T/2) = \mathbf{G}_i^{-1}\mathbf{Z}(T/2)\mathbf{G}_i$ .

Так как матрица  $\mathbf{Z}(-T/2)$  симплектическая, то с учетом формулы (3.4) получаем матрицу монодромии  $\mathbf{M}_1$ , вычисленную от точки  $\mathbf{z}_0$ ,

$$\mathbf{M}_1 = \tilde{\mathbf{G}}_i\mathbf{Z}^1(T/2)\tilde{\mathbf{G}}_i\mathbf{Z}(T/2), \quad (3.6)$$

где  $\tilde{\mathbf{G}}_i = \mathbf{G}_i\mathbf{J}$ ,  $\tilde{\mathbf{G}}_i^{-1} = \tilde{\mathbf{G}}_i^1 = \tilde{\mathbf{G}}_i$ . Нетрудно получить матрицу монодромии  $\mathbf{M}_2$ , вычисленную от точки  $\mathbf{z}(T/2)$

$$\mathbf{M}_2 = \mathbf{Z}(T/2)\tilde{\mathbf{G}}_i\mathbf{Z}^1(T/2)\tilde{\mathbf{G}}_i. \quad (3.7)$$

Вычисление матриц монодромии  $\mathbf{M}_1, \mathbf{M}_2$  по формулам (3.6) и (3.7) соответственно предпочтительнее по следующим соображениям. Во-первых, интегрирование системы (3.1) на половине периода требует меньших вычислительных затрат и дает более точные результаты. Во-вторых, в случае бифуркации потери симметрии (см. подраздел 1) требуется информация о структуре матрицы монодромии как в точке  $\mathbf{z}(0)$ , так и в точке  $\mathbf{z}(T/2)$ .

Воспользуемся этими соображениями для выявления свойств матрицы монодромии двояко симметричного периодического решения. Тогда для ее вычисления достаточно знать решение системы (3.1) на четверти периода  $\mathbf{Z}(T/4)$ . Действительно, пусть начальная точка  $\mathbf{z}_0$  дважды симметричной орбиты расположена на множестве  $\Sigma_1$ , тогда через четверть периода  $\mathbf{Z}(T/4) \in \Sigma_2$ . Обозначим через  $\mathbf{Z}_1(T/4)$  решение системы (3.1), соответствующее начальной точке  $\mathbf{z}(0)$ , а через  $\mathbf{Z}_2(T/4)$  – решение, соответствующее начальной точке  $\mathbf{z}(T/4)$ . Согласно групповому свойству решений имеем  $\mathbf{Z}_1(T/2) = \mathbf{Z}_2(T/4)\mathbf{Z}_1(T/4)$ . С другой стороны, в силу свойства  $t$ -инвариантности уравнения в вариациях относительно преобразования  $g_2$ , имеем

$$\begin{aligned} \mathbf{Z}_2(-T/4) &= \mathbf{G}_2^{-1}\mathbf{Z}_2(T/4)\mathbf{G}_2 \Rightarrow \\ &\Rightarrow \mathbf{Z}_2(T/4) = \mathbf{G}_2\mathbf{Z}_2(-T/4)\mathbf{G}_2^{-1}. \end{aligned}$$

Используя тот факт, что  $\mathbf{Z}_2(-T/4) = \mathbf{Z}_1^{-1}(T/4) = -\mathbf{J}\mathbf{Z}_1^1(T/4)\mathbf{J}$ , получаем для  $\mathbf{Z}_1(T/2)$  выражение

$$\mathbf{Z}_1(T/2) = \tilde{\mathbf{G}}_1\mathbf{Z}_1^1(T/4)\tilde{\mathbf{G}}_2\mathbf{Z}_1(T/4).$$

Применяя  $t$ -инвариантность системы (3.1) относительно преобразования  $g_1$ , в итоге получим формулы для матриц монодромии  $\mathbf{M}_1$  и  $\mathbf{M}_2$

$$\begin{aligned} \mathbf{M}_1 &= [\tilde{\mathbf{G}}_1\mathbf{Z}_1^1(T/4)\tilde{\mathbf{G}}_3\mathbf{Z}_1(T/4)]^2, \\ \mathbf{M}_2 &= [\mathbf{Z}_1(T/4)\tilde{\mathbf{G}}_1\mathbf{Z}_1^1(T/4)\tilde{\mathbf{G}}_2]^2. \end{aligned} \quad (3.8)$$

Непосредственными вычислениями убеждаемся в том, что для двояко симметричного периодического решения матрицы монодромии, вычисленные в точках  $\mathbf{z}(0)$  и  $\mathbf{z}(T/2)$ , а также в точках  $\mathbf{z}(T/4)$  и  $\mathbf{z}(3T/4)$ , попарно равны.

Наконец, рассмотрим случай  $g_3$ -инвариантных периодических решений. Пусть  $\mathbf{z}_0$  – начальная точка такого решения, тогда нетрудно видеть, что  $\mathbf{z}_1 \equiv \mathbf{z}(T/2, \mathbf{z}_0) = -\mathbf{z}_0$ . В силу симметрии векторы фазовой скорости, вычисленные в точках  $\mathbf{z}_0$  и  $\mathbf{z}_1$ , будут равны по абсолютной величине и противоположны по направлению. Следовательно, вектор  $\mathbf{v}_0$  является собственным вектором матрицы  $\mathbf{Z}(T/2, \mathbf{z}_0)$  и ему соответствует собственное число  $-1$ . Матрица монодромии в этом случае есть

$$\mathbf{M} = \mathbf{Z}^2(T/2, \mathbf{z}_0). \quad (3.9)$$

Этот факт будет использован в дальнейшем при исследовании удвоения периода двояко симметричных периодических решений.

Особая структура матрицы монодромии обратимо-симметричного периодического решения

приводит к наличию некоторой внутренней симметрии матрицы  $M$ :

$$\begin{aligned} m_{33} &= m_{11}, & m_{44} &= m_{22}, & m_{43} &= -m_{12}, \\ m_{23} &= -m_{14}, & m_{34} &= -m_{21}, & m_{41} &= -m_{32}. \end{aligned} \quad (3.10)$$

Заметим, что наличие двух симметрий у периодического решения не добавляет дополнительных связей к равенствам (3.10).

Величина индекса устойчивости  $S$  центрально симметричного периодического решения ограничена снизу значением  $-1$ . Это следует непосредственно из формул (3.8) или (3.9) и того, что след матрицы  $M$  такого решения равен сумме квадратов собственных чисел вещественной матрицы  $Z(T/2, z_0)$ , т.е., является неотрицательной величиной, а значение  $S$  вычисляется по формуле (3.5).

#### 4. ПРОДОЛЖЕНИЕ СЕМЕЙСТВА ПЕРИОДИЧЕСКИХ РЕШЕНИЙ

Процедура вычисления семейства периодического решения обычно строится по предикторно-корректорной схеме. Вначале предиктор находит малые поправки  $\delta z$  и  $\delta T$  к начальным условиям и периоду, а затем корректор уточняет поправки и в случае успеха получает новое периодическое решение семейства.

Пусть известно некоторое невырожденное периодическое решение  $z(t, z_0)$  семейства с начальным условием и периодом  $(z_0, T)$ , тогда вблизи него должно быть периодическое решение  $z(t) + \delta z(t)$  с периодом  $T + \delta T$ . Раскладывая левую часть условия периодичности  $z(T + \delta T, z_0 + \delta z) = z_0 + \delta z$  в ряд Тейлора, оставляя в разложении члены не выше первого порядка малости по  $\delta z$  и  $\delta T$ , получаем, что малые добавки  $\delta z$  и  $\delta T$  должны удовлетворять линейной однородной системе

$$(M - E)\delta z(T) + v_1 \delta T = 0, \quad (4.1)$$

где  $v_1 = J \text{grad} H(z(T, z_0))$ . Множество решений этой системы определяется структурой матрицы монодромии  $M$ .

Для обратимых симметричных периодических решений (которые рассматриваются в данной работе) вместо системы (4.1) удобнее продолжать однопараметрическое семейство по другой схеме – схеме продолжения по длине дуги [18, Гл. 5]. Выбирая начальное приближение симметричного решения на инвариантном множестве  $\Sigma_b$ ,  $b = 1, 2$ , в силу теоремы 1 условие периодичности эквивалентно условию равенства нулю отображения

$$g : \mathbb{R}^3 \rightarrow \mathbb{R}^2 : \xi \mapsto g(\xi),$$

где  $\xi = (z_0^j, z_0^k, \tau)$ ,  $g(\xi) = (z_l(\tau, z_0^j, z_0^k), z_m(\tau, z_0^j, z_0^k))^t$ . Величина  $\tau$  равна  $T/2$  для однократно и  $T/4$  для двойко симметричного решения, а индексы  $j, k, l, m$  выбираются так, чтобы пара  $j, k$  соответствовала ненулевым компонентам вектора начального условия  $z_0 \in \Sigma_b$ , а пара  $l, m$  – нулевым компонентам вектора  $g \in \Sigma_e$ ,  $e = 1, 2$ , согласно формулам (2.4). Тогда решая однородную систему  $Dg(\xi)\delta\xi = 0$  по формулам

$$\frac{\delta\xi_1}{D_1} = \frac{\delta\xi_2}{D_2} = \frac{\delta\xi_3}{D_3},$$

где  $D_i$  – определитель матрицы Якоби  $Dg(\xi)$  с удаленным  $i$ -м столбцом, взятый со знаком  $(-1)^i$ , находим поправки  $\delta\xi$  в виде

$$\delta\xi_i = D_i \Delta s, \quad \tilde{D}_i = D_i / \sqrt{D_1^2 + D_2^2 + D_3^2}.$$

Шаг смещения  $\Delta s$  вдоль семейства выбирается таким образом, чтобы обеспечить сходимость корректора.

Когда найдены поправки  $\delta\xi$  первого порядка применяется корректор, который обычно реализуется в виде итерационной процедуры по модифицированной схеме Ньютона. Пусть  $\xi^{(0)}$  – вычисленное на шаге предиктора приближенное начальное условие периодического решения, тогда итерационная схема имеет вид

$$\xi^{(k+1)} = \xi^{(k)} - [Dg(\xi^{(k)})]^+ g(\xi^{(k)}),$$

где знаком  $^+$  обозначена псевдообратная матрица [19, Гл. 1, § 5] матрицы Якоби  $Dg(\xi)$  размера  $2 \times 3$ , состоящей из соответствующих компонентов матрицанта  $Z(\tau)$  и компонентов вектора фазовой скорости  $v(\tau)$ . Псевдообратная матрица  $A^+$  матрицы  $A$  размера  $m \times n$ ,  $m > n$ , может быть вычислена по формуле

$$A^+ = A^t (A \cdot A^t)^{-1}$$

при условии, что  $\text{rang} A = n$ .

Описанная схема продолжения семейства по длине дуги обладает тем преимуществом, что позволяет проходить “складки”, т.е. когда одна из величин  $\xi_i$ ,  $i = 1, 2, 3$ , достигает экстремального значения.

Покажем, что собственному числу  $+1$  матрица монодромии  $M$  невырожденного периодического решения  $(z_0, T)$  автономной неинтегрируемой системы Гамильтона соответствует жорданова клетка второго порядка.

**Определение 3.** Периодическое решение  $(z_0, T)$  автономной неинтегрируемой системы Гамильтона (2.1) называется *невырожденным* или *простым*, если ее матрица монодромии  $\mathbf{M}$  имеет единственный элементарный делитель  $(\lambda - 1)^2$ , соответствующий мультипликатору  $\rho_{1,2} = 1$ , а все остальные элементарные делители просты.

Действительно, в случае общего положения, т.е. когда периодическое решение невырождено, согласно [11, Ch. 6. Theorem 6.5.1], оно (решение) лежит на гладкой двумерной цилиндрической поверхности периодических решений, образующая которой параметризуется значением интеграла  $H(\mathbf{z})$ . Из этой теоремы следует, что размерность пространства решений системы (4.1) равна двум и, следовательно, ранг матрицы  $(\mathbf{M} - \mathbf{E} : \mathbf{v}_0)$  системы (4.1) должен быть равен трем. Очевидно, что если у некоторой матрицы  $\mathbf{M}$  размера  $n \times n$  собственное число  $\lambda$  полупростое и имеет кратность  $k$ , то  $\text{rang}(\mathbf{M} - \lambda \mathbf{E}) = n - k$ . Следовательно, собственное число  $+1$  кратности 2 не является полупростым и ему соответствует жорданова клетка размера  $2 \times 2$ . Эти же рассуждения верны и для произвольной автономной системы Гамильтона с  $n$  степенями свободы.

**Утверждение 1.** Матрица монодромии  $\mathbf{M}$  невырожденного периодического решения автономной неинтегрируемой системы Гамильтона в случае общего положения, т.е. когда имеется единственный первый интеграл  $H(\mathbf{z}) = h$ , всегда имеет жорданову клетку второго порядка, соответствующую собственному числу  $+1$  кратности 2.

Анализ решений системы (4.1) удобнее проводить в том базисе, в котором матрица  $\mathbf{M}$  имеет наиболее простую форму, т.е. нормальную форму. Классификация всех нормальных форм постоянных канонических матриц была дана еще Вильямсоном в 1937 году (см. [20, 21, Гл. II]), однако здесь воспользуемся несколько иным подходом. Этот подход основан на идее перехода к некоторому сопутствующему ортонормированному базису, в котором один орт — это касательный вектор к фазовой траектории, а другой — нормаль к изоэнергетической поверхности  $H(\mathbf{z}) = h$ .

#### 4.1. Преобразования матрицы монодромии $\mathbf{M}$

Выполним линейное преобразование матрицы  $\mathbf{M}$ , упрощающее ее структуру и облегчающее ее анализ. Потребуем, чтобы матрица  $\mathbf{A}$  этого преобразования принадлежала группе  $Sp(4, \mathbb{R}) \cap SO(4, \mathbb{R})$ , т.е. была одновременно и симплектической и ортогональной. Дополнительным условием должно быть сохранение внутренней симметрии (3.10) для обратимых симметричных периодических решений.

Как указано в [22], задача построения такой матрицы  $\mathbf{A}$  эквивалентна проблеме построения невырожденного непрерывного касательного векторного поля на сфере  $S^3$ . Как известно, для  $S^3$  это возможно, следовательно, такое преобразование всегда существует.

Условие симплектичности матрицы  $\mathbf{A}$  выполнено, если ее столбцы  $\mathbf{a}_i$  удовлетворяют условиям  $\mathbf{a}_{i+2} = -\mathbf{J}\mathbf{a}_i$ ,  $i = 1, 2$  и  $|\mathbf{a}_j| = 1$ ,  $j = 1, \dots, 4$ . Условие ортогональности матрицы  $\mathbf{A}$  записывается в виде  $(\mathbf{a}_i, \mathbf{a}_j) = \delta_{ij}$ ,  $i, j = 1, \dots, 4$ ,  $\delta_{ij}$  — символ Кронекера. Столбцы  $\mathbf{a}_1$  и  $\mathbf{a}_3$ , как указано выше, определены как нормированные векторы фазовой скорости и градиента первого интеграла соответственно. Остается найти компоненты столбца  $\mathbf{a}_2$  или  $\mathbf{a}_4$ . Из указанных выше условий составляется полиномиальный идеал. Его размерность оказалась равна 1, т.е. имеется однопараметрическое семейство матриц преобразования  $\mathbf{A}$ . Условие сохранения внутренней симметрии (3.10) матрицы монодромии  $\mathbf{M}$  в итоге выделяет из этого семейства только две матрицы, отличающиеся порядком столбцов  $\mathbf{a}_2$  и  $\mathbf{a}_4$ .

Одна из таких матриц  $\mathbf{A}$  была предложена Б.Б. Крейсманом при исследовании периодических решений ограниченной задачи трех тел (см., например, [17]). Пусть  $\mathbf{a}_3 = (H_1, H_2, H_3, H_4)$  — нормированный вектор  $\text{grad}H(\mathbf{z}_0)$ . Тогда матрица  $\mathbf{A}$  имеет вид

$$\mathbf{A} = \begin{pmatrix} H_3 & -H_4 & H_1 & H_2 \\ H_4 & H_3 & H_2 & -H_1 \\ -H_1 & -H_2 & H_3 & -H_4 \\ -H_2 & H_1 & H_4 & H_3 \end{pmatrix}. \quad (4.2)$$

**Утверждение 2** ([17]). Пусть симплектическая матрица  $\mathbf{M}$  имеет собственное значение  $+1$ , которому соответствует собственный вектор  $\mathbf{v}_0 = \mathbf{a}_1$ .

Тогда матрица  $\mathbf{N} = \mathbf{A}^t \mathbf{M} \mathbf{A}$  имеет вид

$$\mathbf{N} = \begin{pmatrix} 1 & n_{12} & n_{13} & n_{14} \\ 0 & n_{22} & n_{23} & n_{24} \\ 0 & 0 & 1 & 0 \\ 0 & n_{42} & n_{43} & n_{44} \end{pmatrix}. \quad (4.3)$$

Это утверждение приведено в [17] без доказательства. В данной работе истинность утверждения проверена с использованием системы компьютерной алгебры Maple, хотя такие же преобразования могут быть выполнены в любой системе компьютерной алгебры, позволяющей вычислять базис Грёбнера полиномиального идеала и с его помощью находить каноническое представление произвольного многочлена по модулю идеала. Общая схема проверки такова.

1) Для произвольной  $4 \times 4$  матрицы  $\mathbf{M}$  с помощью условия симплектичности (3.2) составляется полиномиальный идеал  $\mathcal{F}_1$ . Для него вычисляется базис Грёбнера  $\mathcal{GB}\mathcal{F}_1$  для чистого лексикографического порядка от элементов  $m_{ij}$ ,  $i, j = 1, \dots, 4$  матрицы  $\mathbf{M}$ .

2) Согласно свойству 4) на стр. 4 матрица  $\mathbf{M}$  имеет собственное число, равное  $+1$ . Поэтому вычисляется характеристический многочлен  $P_\lambda(\mathbf{M})$ , его коэффициенты упрощаются по модулю идеала  $\mathcal{GB}\mathcal{F}_1$ , и к этому идеалу добавляется полином, равный сумме коэффициентов многочлена  $P_\lambda(\mathbf{M})$ . Для нового идеала вычисляется его базис Грёбнера  $\mathcal{GB}\mathcal{F}_2$  с тем же лексикографическим порядком.

3) Записывается в символьной форме собственный вектор  $\mathbf{v}_0$  матрицы  $\mathbf{M}$  как решение однородной системы линейных алгебраических уравнений с матрицей  $\mathbf{M} - \mathbf{E}$  в виде

$$\frac{v_0^1}{M_1} = -\frac{v_0^2}{M_2} = \frac{v_0^3}{M_3} = -\frac{v_0^4}{M_4},$$

где  $M_i$  – миноры элементов первой строки матрицы  $\mathbf{M} - \mathbf{E}$ . Компоненты вектора  $\mathbf{v}_0$  упрощаются по модулю идеала  $\mathcal{GB}\mathcal{F}_2$ .

4) В идеал  $\mathcal{GB}\mathcal{F}_2$  добавляется многочлен  $v_0^2 - 1$ , обеспечивающий условие нормировки вектора  $\mathbf{v}_0$ , и вычисляется новый базис Грёбнера  $\mathcal{GB}\mathcal{F}_3$ . Из компонентов вектора  $\mathbf{v}_0$  составляется матрица  $\mathbf{A}$  по формуле (4.2) и с ее помощью выполняется переход к матрице  $\mathbf{N}$ .

5) После этого каждый элемент матрицы  $\mathbf{N}$  упрощается по модулю базиса  $\mathcal{GB}\mathcal{F}_3$ . Матрица  $\mathbf{N}$  принимает форму (4.3).

В силу громоздкости получаемых в процессе вычисления выражений они здесь не приводятся, но легко могут быть восстановлены по данному выше описанию с помощью какой-либо системы компьютерной алгебры.

4.2. *Общее решение уравнения продолжения для невырожденного периодического решения*

Преобразование матрицы монодромии  $\mathbf{M}$  с помощью матрицы  $\mathbf{A}$  позволяет, с одной стороны, прояснить структуру решений уравнения продолжения (4.1) для несимметричного периодического решения [17], а с другой стороны, упростить ее анализ в критических случаях (см. далее п. 5).

Отметим, что случай двояко симметричного периодического решения ничем не отличается от случая однократно симметричного, рассмотренного в [17].

Для обратимого симметричного периодического решения  $\mathbf{z}(t, \mathbf{z}_0)$  с начальным условием  $\mathbf{z}_0 \in \Sigma_i$  матрица  $\mathbf{A}_i$  имеет вид

$$\mathbf{A}_1 = \begin{pmatrix} 0 & -H_4 & H_1 & 0 \\ H_4 & 0 & 0 & -H_1 \\ -H_1 & 0 & 0 & -H_4 \\ 0 & H_1 & H_4 & 0 \end{pmatrix},$$

$$\mathbf{A}_2 = \begin{pmatrix} H_3 & 0 & 0 & H_2 \\ 0 & H_3 & H_2 & 0 \\ 0 & -H_2 & H_3 & 0 \\ -H_2 & 0 & 0 & H_3 \end{pmatrix}.$$

Внутренняя симметрия (3.10) матрицы  $\mathbf{M}$  приводит к следующим соотношениям для преобразованной матрицы  $\mathbf{N}$ :

$$n_{22} = n_{44}, \quad n_{23} = -n_{14}, \quad n_{43} = -n_{12},$$

а индекс устойчивости в этом случае  $S = n_{22}$ .

Вектор  $\mathbf{b}$  продолжения семейства имеет вид

$$\mathbf{b} = (0, 2n_{23}, 2(1 - n_{22}), 0, -(n_{12}m_2 + n_{13}m_3)/v). \quad (4.4)$$

Его проекция  $\tilde{\mathbf{b}}$  на фазовое пространство  $\mathbb{M}$  определяется только вторым и третьим столбцами матрицы  $\mathbf{A}$ , что, очевидно, подтверждает приведенные выше рассуждения о продолжении обратимого симметричного решения.

**Утверждение 3 ([17]).** Продолжение невырожденного симметричного периодического решения вдоль семейства сохраняет его симметрию.

5. КРИТИЧЕСКИЕ РЕШЕНИЯ СЕМЕЙСТВА ДВОЯКО СИММЕТРИЧНОГО РЕШЕНИЯ

В предыдущем пункте рассматривалась ситуация, когда периодическое решение невырожденное. Здесь выясним, как симметрия периодического решения влияет на поведение его семейства в критических решениях, т.е. для тех решений, для которых либо у матрицы монодромии  $\mathbf{M}$ , либо у ее  $q$ -й степени  $\mathbf{M}^q$  соответствующего критического решения появляются дополнительные элементарные делители. Это решение семейства будем называть *бифуркационным решением*.

Отметим, что соответствующий бифуркационный анализ однократно симметричных периодических решений был проведен ранее. Так, в Главе VIII книги [21] это было сделано с использованием метода нормальной формы. В статье [17] анализ симметричного решения проводился с использованием последовательных симплектических преобразований матрицы  $\mathbf{M}$ . Фактически эта работа является продолжением методики по-

следней работы на случай двояко симметричных решений.

Мы рассмотрим три типичных критических случая, которые обычно относят к одному из видов бифуркации периодического решения [23, Ch. 11]:

- бифуркация рождения-гибели (седло-узел);
- бифуркация потери симметрии (типа “вилки”);
- бифуркация кратного увеличения периода.

Первые два случая соответствуют значению индекса устойчивости  $S = 1$ , последний случай соответствует значению  $S = \cos(2\pi p/q)$ ,  $p, q \in \mathbb{N}$ ,  $p$  и  $q$  взаимно просты. Отдельно рассмотрим важный случай бифуркации удвоения периода при  $S = -1$ .

5.1. Случай  $S = 1$

В этом случае  $n_{22} = n_{44} = 1$ , матрица  $\mathbf{N}$  принимает вид

$$\mathbf{N} = \begin{pmatrix} 1 & n_{12} & n_{13} & n_{14} \\ 0 & 1 & -n_{14} & n_{24} \\ 0 & 0 & 1 & 0 \\ 0 & n_{42} & -n_{12} & 1 \end{pmatrix},$$

а два компонента  $m_3, m_4$  вектора  $\mathbf{b}$  согласно (4.4) равны 0. Условие симплектичности матрицы  $\mathbf{N}$  приводит к условиям

$$2n_{14} = n_{24}n_{12}, \quad n_{24}n_{42} = 0, \quad n_{14}n_{42} = 0.$$

Здесь возможны два случая в зависимости от значения единственного нетривиального компонента  $m_2$ .

**Случай  $m_2 \neq 0$ .** Тогда из (4.4) следует, что  $n_{42} = 0$ , и матрица  $\mathbf{N}$  принимает вид

$$\mathbf{N} = \begin{pmatrix} 1 & n_{12} & n_{13} & \frac{1}{2}n_{12}n_{24} \\ 0 & 1 & -\frac{1}{2}n_{12}n_{24} & n_{24} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -n_{12} & 1 \end{pmatrix} \sim \exp(\mathbf{JC}),$$

где матрица  $\mathbf{C}$  согласно [21, Гл. II, § 1] подобна матрице

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

и, следовательно, имеет элементарный делитель  $(\lambda - 1)^4$ . В этом случае вектор продолжения семейства  $\mathbf{b} = (0, -n_{12}, 0, 0, -n_{12}m_2/\nu)^t$ , а его проекция  $\tilde{\mathbf{b}}$  ор-

тогональна вектору  $H(\mathbf{z}_0)$ . Следовательно, первый интеграл  $H(\mathbf{z})$  достигает экстремума на семействе периодических решений, на характеристике семейства в координатах  $(h, S)$  имеется складка, а само критическое решение называется *бифуркацией рождения-гибели*. При прохождении через это решение меняется устойчивость семейства.

**Случай  $m_2 = 0$ .** Тогда из (4.4) следует, что  $n_{24} = 0$ ,  $n_{42} \neq 0$ , и матрица  $\mathbf{N}$  принимает вид

$$\mathbf{N} = \begin{pmatrix} 1 & n_{12} & n_{13} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & n_{42} & -n_{12} & 1 \end{pmatrix}.$$

Выполним дополнительное симплектическое преобразование

$$\tilde{\mathbf{A}} = \begin{pmatrix} 1 & 0 & 0 & n_{12}/n_{42} \\ 0 & 1 & n_{12}/n_{42} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\tilde{\mathbf{N}} = \begin{pmatrix} 1 & 0 & n_{13} + n_{12}^2/n_{42} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & n_{42} & 0 & 1 \end{pmatrix}.$$

Нетрудно видеть, что матрица  $\tilde{\mathbf{N}}$  имеет два элементарных делителя  $(\lambda - 1)^2$  и  $(\lambda - 1)^2$ . Корневой вектор, соответствующий первому элементарному делителю, задает направление главного семейства  $\mathbf{b}_1 = (0, 0, 1, 0, -\tilde{n}_{13}/\nu)^t$ . Собственный вектор  $\mathbf{b}_2 = (0, 0, 0, 1, 0)$ , соответствующий второму элементарному делителю, задает направление нового семейства периодических решений с периодом, равным периоду решения главного семейства. В этом случае новое решение будет обладать только одной из двух симметрий. Поскольку для двояко симметричных периодических решений матрицы монодромии  $\mathbf{M}_1$  и  $\mathbf{M}_2$ , вычисленные от взаимно симметричных точек, отстоящих через полпериода, равны, то в результате получаем два порожденных семейства однократно симметричных решений. Периодические решения этих двух порожденных семейств взаимно симметричны относительно другого автоморфизма  $g_i$ . Другими словами, если порожденные семейства состоят из  $g_1$ -симметричных орбит, то эти орбиты взаимно  $g_2$ -симметричны.

5.2. Случай  $S = -1$

Рассмотрим подробнее критический случай, соответствующий бифуркации удвоения периода.

В общем случае компонента  $m_3 \neq 0$  и можно выполнить дополнительное симплектическое преобразование

$$\tilde{\mathbf{A}} = \begin{pmatrix} 1 & 0 & 0 & m_2/m_3 \\ 0 & 1 & m_2/m_3 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (5.1)$$

$$\tilde{\mathbf{N}} = \begin{pmatrix} 1 & 0 & \tilde{n}_{13} & 0 \\ 0 & -1 & 0 & n_{24} \\ 0 & 0 & 1 & 0 \\ 0 & n_{42} & 0 & -1 \end{pmatrix}.$$

с дополнительным условием  $n_{24}n_{42} = 0$ . Здесь  $\tilde{n}_{13} = n_{12}m_2/m_3 + n_{13}$ .

Для однократно симметричного периодического решения имеем либо  $n_{24} = 0, n_{42} \neq 0$ , либо  $n_{42} = 0, n_{24} \neq 0$ . Тогда матрица  $\tilde{\mathbf{N}}$  имеет два элементарных делителя  $(\lambda - 1)^2$  и  $(\lambda + 1)^2$ . Корневой вектор  $\mathbf{r}_1$ , соответствующий первому элементарному делителю, задает направление главного семейства  $\mathbf{b}_1 = \mathbf{r}_1$ . Для второго элементарного делителя возможны два варианта с собственным вектором  $\mathbf{b}_2^1 = (0, 1, 0, 0)$  в первом случае и  $\mathbf{b}_2^2 = (0, 0, 0, 1)$  во втором. Этот вектор задает направление нового семейства периодических решений с периодом, равным периоду решения главного семейства. Анализ, проведенный в [17], показывает, что на порожденном семействе всегда имеет место экстремум  $H(\mathbf{z})$  и оно (семейство) сохраняет тип симметрии порождающего решения.

Рассмотрим случай двойко симметричного решения. В этом случае матрица монодромии  $\mathbf{M}$ , согласно (3.9), является квадратом матрицы решения  $\mathbf{Z}(T/2, \mathbf{z}_0)$  уравнения в вариациях (3.1) на половине периода. Но тогда, в силу вещественности, у нее не может быть элементарного делителя  $(\lambda + 1)^2$ , поскольку вещественного квадратного корня из жордановой клетки второго порядка, соответствующей этому делителю, нет [23, Ch. 3].

**Утверждение 4.** Матрица монодромии  $\mathbf{M}$  двойко симметричного периодического решения при  $S = -1$  в случае общего положения имеет три элементарных делителя:  $(\lambda - 1)^2, (\lambda + 1)$  и  $(\lambda + 1)$ . Двум последним соответствуют однократно симметричные решения с разными типами симметрии и периодом  $T = 2T$ . На каждом из семейств однократно симметричных решений первый интеграл  $H(\mathbf{z})$  достигает экстремума в бифуркационном решении.

Имеется два сценария бифуркации удвоения периода:

1) на обоих новых порожденных семействах интеграл  $H(\mathbf{z})$  имеет один и тот же тип экстремума

(максимум или минимум) в критическом решении,

2) на новых порожденных семействах интеграл  $H(\mathbf{z})$  имеет разные типы экстремума (у одной пары минимум, у другой — максимум).

При первом сценарии в результате бифуркации появляется четыре пары периодических решений с удвоенным периодом — две пары с одним типом симметрии, две с другим. В этом случае одна пара семейств двойко периодических решений имеет устойчивые орбиты, а другая неустойчивые.

При втором сценарии в окрестности бифуркационного решения имеется по одной паре периодических решений с удвоенным периодом. Поскольку индекс устойчивости  $S$  порождающего семейства всегда больше или равен  $-1$ , то порожденные семейства двойко периодических решений всегда имеют только неустойчивые орбиты.

### 5.3. Случай кратного увеличения периода

Случай бифуркации удвоения периода является частным случаем бифуркации кратного увеличения периода когда  $S = \cos \varphi, \varphi = 2\pi p/q$ . Количественный анализ аналогичный тому, что был проведен в [17], показывает, что при  $S \neq \pm 1$  компонента  $m_3$  вектора  $\mathbf{b}$  не равна нулю. Тогда произведение матриц  $\tilde{\mathbf{A}}$  из (5.1) и  $\mathbf{C} = \text{diag}\{1, c_{22}, 1, c_{44}\}$ , где  $c_{44} = \sqrt{n_{42}/\sin \varphi}, c_{22} = 1/c_{44}$ , дает матрицу симплектического преобразования  $\tilde{\mathbf{A}}\mathbf{C}$ , приводящего матрицу монодромии  $\tilde{\mathbf{N}}$  к виду

$$\tilde{\mathbf{M}} = \begin{pmatrix} 1 & 0 & \tilde{n}_{13} & 0 \\ 0 & \cos \varphi & 0 & -\sin \varphi \\ 0 & 0 & 1 & 0 \\ 0 & \sin \varphi & 0 & \cos \varphi \end{pmatrix}. \quad (5.2)$$

При этом угол  $\varphi$  определяется из условий

$$\cos \varphi = S, \quad n_{42} \sin \varphi \geq 0.$$

Из (5.2) следует, что для нечетных и взаимно простых чисел  $p, q$  матрица  $\tilde{\mathbf{M}}^q$  имеет структуру элементарных делителей  $(\lambda - 1)^2, (\lambda - 1)$  и  $(\lambda - 1)$ . Если же одно из чисел  $p, q$ , четно, то у матрицы  $\mathbf{Z}(qT/2, \mathbf{z}_0)$  структура элементарных делителей  $(\lambda - 1)^2, (\lambda + 1)$  и  $(\lambda + 1)$ , т.е. такая же как в утверждении 4. Как и в предыдущих случаях, корневой вектор жордановой клетки второго порядка с собственным числом  $+1$  дает направление продолжения главного семейства. Собственные векторы, соответствующие простым элементарным делителям, дают направление либо одному двойко симметричному порожденному семейству для нечетных  $p, q$ , либо двум парам семейств однократ-

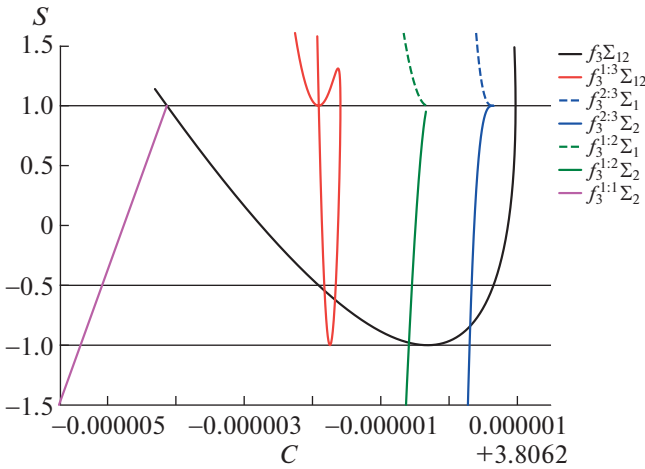


Рис. 1. Характеристика семейства  $f_3$  и связанных с ним семейств на плоскости  $(C, S)$ .

но симметричных периодических решений, когда одно из  $p, q$  четно. Качественное объяснение этого эффекта состоит в следующем.

Рассмотрим двояко симметричное периодическое решение, орбита которого последовательно ортогонально пересекает плоскости симметрии  $\Sigma_{1,2}$  через четверть периода. Пусть это решение с периодом  $T$  является порождающим решением для периодического решения второго рода с периодом  $qT$ . Для того чтобы орбита порожденного решения также имела две симметрии, необходимо выполнение указанного выше условия ортогональности. Пусть начальная точка порожденной орбиты лежит на плоскости  $\Sigma_1$ , тогда через четверть периода следующая ортогональная точка будет лежать на плоскости  $\Sigma_2$ , при этом точка орбиты должна совершить  $(2k - 1)/4$  оборота вокруг начала координат, где  $k \in \mathbb{N}$ . Поэтому за весь период  $qT$  точка орбиты совершит  $2k - 1$  оборотов, то есть число  $q$  всегда нечетное. С другой стороны, орбита, соответствующая резонансу  $p/q$ , может быть представлена в фазовом пространстве как замкнутая винтовая линия, лежащая на некотором торе [24], которая делает за один период  $q$  оборотов по одной образующей тора и  $p$  оборотов по другой образующей. Поэтому для двояко симметричной орбиты число  $p$  тоже должно быть нечетным.

**Утверждение 5.** Пусть двояко симметричное периодическое решение с периодом  $T$  имеет значение индекса устойчивости  $S$ , равное  $\cos(2\pi p/q)$ , где  $p, q \in \mathbb{N}$  и взаимно просты.

- Если оба числа  $p, q$  нечетные, то в окрестности исходного решения имеется одно семейство двояко симметричных решений с периодом  $T = qT$ .

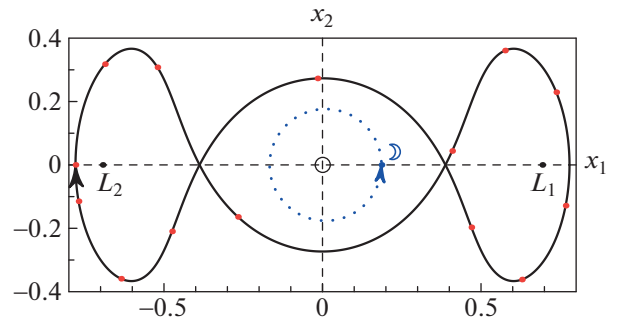


Рис. 2. Орбита семейства  $f_3$ .

- Если одно из чисел  $p$  или  $q$  четно, то в окрестности исходного решения имеется две пары семейств однократно симметричных решений с различными типами симметрий для каждой пары и с периодами  $T = qT$ .

Для всех случаев, кроме случая  $p/q = 1/3$ , интеграл  $H$  достигает экстремума на порожденных семействах (подробнее см. [21, Гл. VIII]).

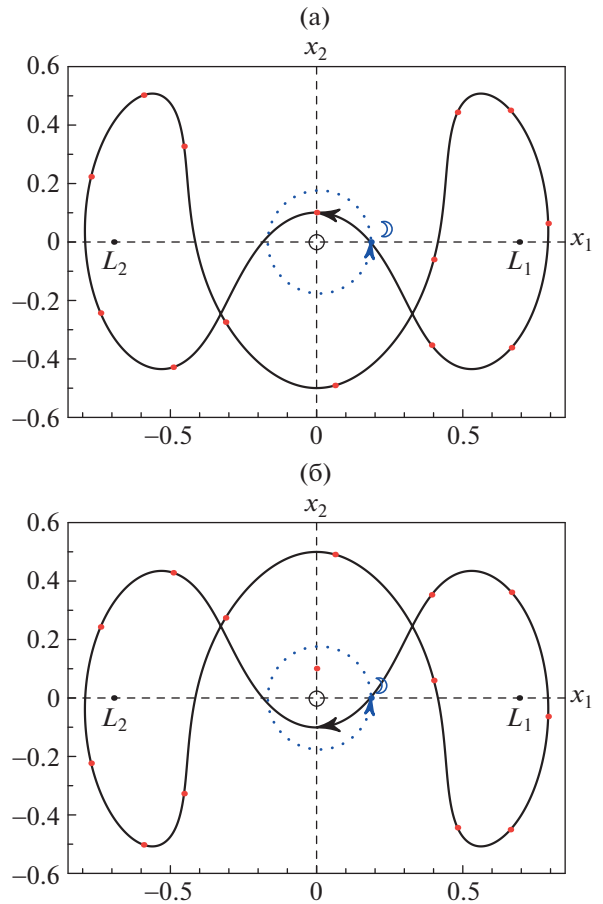


Рис. 3. Орбиты  $g_2$ -симметричных семейств  $f_3\Sigma_2$ .



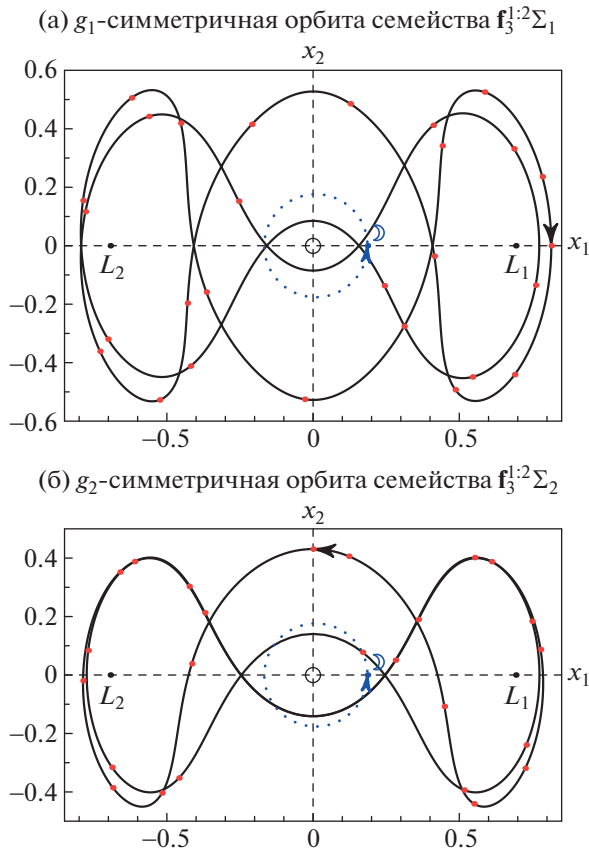


Рис. 4. Орбиты семейств двояко периодических решений.

### 6. ИССЛЕДОВАНИЕ СЕМЕЙСТВ ДВОЯКО СИММЕТРИЧНЫХ РЕШЕНИЙ ЗАДАЧИ ХИЛЛА

#### 6.1. Задача Хилла и ее свойства

Плоская круговая задача Хилла [25, 26, § 10.4] используется для исследования динамики тела “нулевой” массы (спутника), движущегося в окрестности меньшего из двух тяготеющих тел и остающегося в плоскости их движения. Эта задача является некоторым предельным вариантом известной *ограниченной задачи трех тел* (ОЗТТ) [26], когда

i) массовый параметр  $\mu$  в ОЗТТ стремится к нулю и

ii) большее из двух массивных тел стремится к  $+\infty$  вдоль оси абсцисс в равномерно вращающейся (синодической) системе координат.

Хотя задача Хилла неинтегрируема [27], но ее уравнения движения в отличие от уравнений ОЗТТ не содержат массового параметра  $\mu$ , что делает их существенно проще уравнений ОЗТТ. Вторым важным отличием уравнений задачи Хилла от уравнений ОЗТТ является наличие у них двух дополнительных инвариантов относительно четверной группы Клейна линейных автоморфизмов с двумя образующими, обозначаемыми  $g_1$  и  $g_2$  (см. раздел 2). Каждое из этих преобразований имеет инвариантное множество в виде двумерной плоскости в расширенном фазовом пространстве задачи.

Единственным первым интегралом системы канонических уравнений задачи Хилла

$$\begin{cases} \dot{x}_1 = y_1 + x_2, & \dot{y}_1 = y_2 + 2x_1 + \frac{x_1}{|\mathbf{x}|^3}, \\ \dot{x}_2 = y_2 - x_1, & \dot{y}_2 = -y_1 - x_2 + \frac{x_2}{|\mathbf{x}|^3}, \end{cases}$$

остается интеграл энергии, традиционно записываемый в форме интеграла Якоби

$$\mathcal{J} \equiv 3x_1^2 + \frac{2}{|\mathbf{x}|} - \dot{x}_1^2 - \dot{x}_2^2 = C,$$

где постоянная  $C$  удовлетворяет соотношению  $C = -2H(\mathbf{z})$ .

#### 6.2. Семейство $\mathbf{f}_3$ двояко симметричных решений

Для плоской круговой задачи Хилла было найдено большое количество семейств периодических решений (см., например, книгу [25] и библиографию в ней, а также [28]). Среди этих семейств можно выделить основные, которые прямо или опосредованно связаны другими семействами через порождающие и/или критические решения [29]. Такими семействами можно считать семейства

Таблица 1. Параметры критических периодических решений семейства  $\mathbf{f}_3$

№	$x_1(0)$	$y_2(0)$	$T$	$S$	$x_2(T/4)$	$y_1(T/4)$
1	0.7790205	0.0162955	6.6496347	1.0	0.2725126	-2.1521173
2	0.7790055	0.0162955	6.6477869	-0.5	0.2723290	-2.1532489
3	0.7789904	0.0162951	6.6459314	-1.0	0.2721447	-2.1543862
4	0.7789752	0.0162942	6.6440671	-0.5	0.2719594	-2.1555297
5	0.7789601	0.0162930	6.6421949	1.0	0.2717735	-2.1566791

**Таблица 2.** Начальные условия периодических решений семейства  $f_3^{1:2}\Sigma_1$

№	$x_1(0)$	$y_2(0)$	$T$	$S$	$x_1(T/2)$	$y_2(T/2)$
1	0.7789945	0.0162954	13.2918621	1.0029942	0.7789862	0.0162946
2	0.7790215	0.0162972	13.2918633	10.189443	0.7789594	0.0162912
3	0.8146104	-0.2266505	13.7419253	$3.0 \times 10^{11}$	0.7734352	-0.2359037

**Таблица 3.** Начальные условия периодических решений семейства  $f_3^{1:2}\Sigma_2$

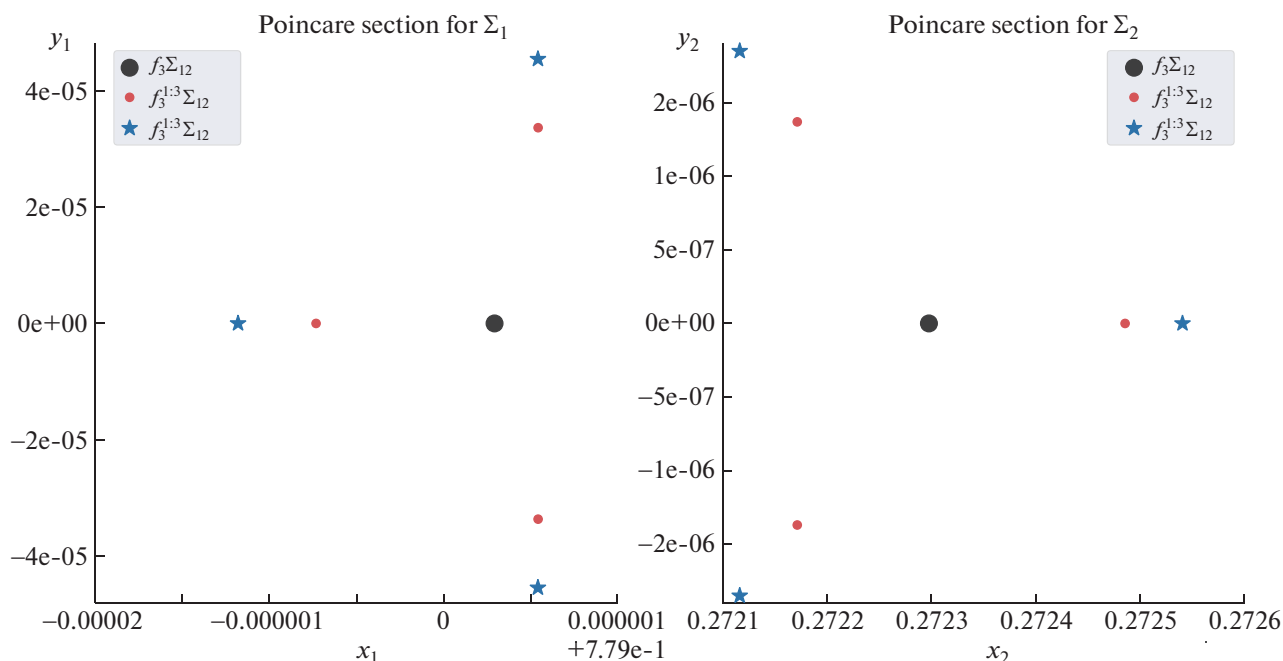
№	$x_2(0)$	$y_1(0)$	$T$	$S$	$x_2(T/2)$	$y_1(T/2)$
1	0.2722264	-2.1538817	13.2917712	0.9998048	0.2720628	-2.1548915
2	0.2725404	-2.1519454	13.2901165	-1.0088341	0.2717481	-2.1568361
3	0.4308632	-1.4454805	12.1324885	$-4.57 \times 10^9$	0.1405653	-3.3987653

двойко симметричных прямых и обратных спутниковых орбит  $\mathbf{g}$  и  $\mathbf{f}$  соответственно (см. [30, 25]).

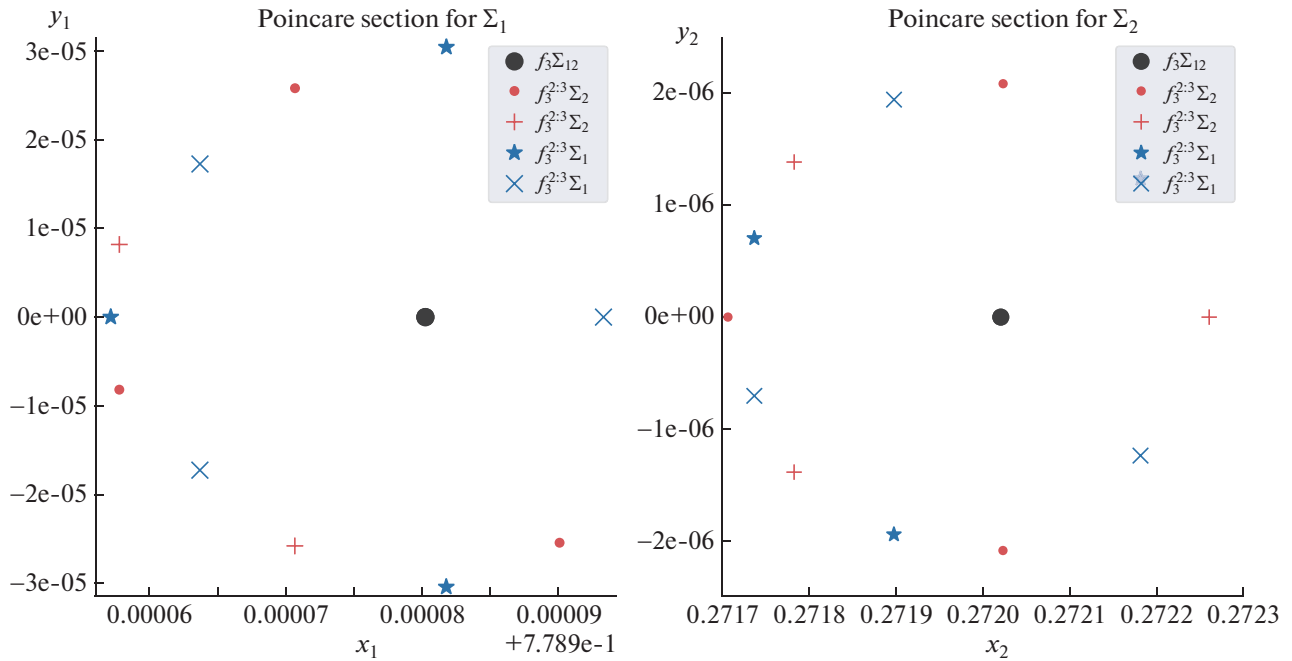
Семейства  $\mathbf{g}$  и  $\mathbf{f}$  не имеют участков, где индекс устойчивости менялся бы на всем интервале устойчивости, поэтому здесь в качестве примера рассмотрим семейство двойко симметричных трехоборотных периодических решений  $\mathbf{f}_3$ . Это семейство найдено М. Эноном в [31] (где названо  $\mathbf{g}_3$ ), подробно исследовано в [32, 25], где приведены примеры его орбит. В табл. 1 приведены начальные условия

на плоскостях симметрии, период, значения константы  $C$  и индекса устойчивости  $S$  для критических орбит семейства  $\mathbf{f}_3$ .

Участок характеристики этого семейства в координатах  $(C, S)$ , а также других семейств, пересекающихся с ним в критических решениях, показан на рис. 1, а пример орбиты семейства — на рис. 2. Здесь и далее на рис. 3 и 4 показано положение точек либрации  $L_1, L_2$ , а также орбита Луны (пунктир). Точ-



**Рис. 5.** Сечение Пуанкаре плоскостями  $\Sigma_1$  (слева) и  $\Sigma_2$  (справа) двойко симметричных периодических решений вблизи семейства  $\mathbf{f}_3$  для случая  $p/q = 1/3$ .



**Рис. 6.** Сечение Пуанкаре плоскостями  $\Sigma_1$  (слева) и  $\Sigma_2$  (справа) двойко симметричных периодических решений вблизи семейства  $f_3$  для случая  $p/q = 2/3$ .

ками обозначено положение спутника на орбите с временным интервалом в один месяц.

Индекс устойчивости  $S$  семейства  $f_3$  пересекает значение  $+1$  в двух точках (см. рис. 1). В самой правой точке (строка 5 табл. 1) константа  $C$  на семействе достигает экстремума и, следовательно, имеет место бифуркация рождения-гибели. В самой левой точке (строка 1 табл. 1) семейство пересекает значение  $+1$  монотонно, здесь происходит появление двух семейств  $g_2$ -периодических решений  $f_3\Sigma_2$ . Эти семейства были найдены в [33], где обозначены  $H_k$ . Более подробно они описаны в [25, п. 4.3.5]. Примеры орбит этих семейств показаны на рис. 3.

Рассмотрим появление двух семейств двойко периодических однократно симметричных решений, когда индекс устойчивости  $S$  главного семейства  $f_3$  проходит значение  $-1$  (строка 3 табл. 1). Для этого начального условия вычисляются матрицы  $A_1$  и  $A_2$  — каждая для точки на соответствующей плоскости симметрии. Вторые столбцы этих матриц дают векторы  $b_1^1$  и  $b_2^2$ , в направлении которых ответвляются двукратные периодические решения симметричные относительно преобразований  $g_1$  и  $g_2$  соответственно. Эти векторы равны

$$\begin{aligned} \tilde{b}_1^1 &= (-0.9953708, 0.0, 0.0, -0.0961093)^t, \\ \tilde{b}_2^2 &= (0.0, 0.1599010, 0.9871331, 0.0)^t. \end{aligned}$$

В соответствии с утверждением 4 в этом случае наблюдается появление двух семейств согласно сценарию 1) на стр. 23, поскольку оба семейства имеют максимум по  $C$ . Два семейства  $f_3^{1:2}\Sigma_1$   $g_1$ -симметричных орбит содержат только неустойчивые решения. Некоторые начальные условия этого семейства даны в табл. 2. Пример орбиты приведен на рис. 4а. Ей соответствует последняя строка таблицы.

Два семейства  $f_3^{1:2}\Sigma_2$   $g_2$ -симметричных орбит имеют небольшой участок с устойчивыми решениями. Некоторые начальные условия этого семейства даны в таб. 3. Пример орбиты приведен на рис. 4б). Ей соответствует последняя строка таблицы.

Наконец, рассмотрим появление трехкратно периодических порожденных решений. Оно происходит, когда  $S = -1/2$ , что возможно в двух ситуациях.

**При  $p/q = 1/3$ .** В этой ситуации согласно утверждению 5 появляется одно семейство двойко симметричных решений, которое вблизи порождающего семейства имеет только неустойчивые решения. Однако здесь на семействе при продолжении достигается максимум константы  $C$ , что видно по его характеристике на рис. 1. Следо-

вательно, при одном и том же значении интеграла  $C$  на сечении Пуанкаре должны наблюдаться два набора неподвижных точек: один набор устойчивых, другой – неустойчивых. Это можно наблюдать на рис. 5.

При  $p/q = 2/3$ . Согласно утверждению 5 появляется четыре семейства однократно симметричных решений – одна пара семейств  $g_1$ -симметричных неустойчивых решений, другая пара семейств  $g_2$ -симметричных устойчивых решений. Таким образом, на сечении Пуанкаре при определенных условиях можно наблюдать 12 неподвижных точек: по три точки на периодическое решение каждого из семейств, что видно на рис. 6.

Все результаты, полученные в разделе 6, получены путем численного интегрирования системы Гамильтона задачи Хилла и ее уравнений в вариациях с использованием пакетов TAYLOR [2] и TIDES [3].

#### БЛАГОДАРНОСТИ

Пользуясь возможностью, автор благодарит профессора А.Д. Брюно за плодотворное обсуждение работы и ее поддержку.

#### СПИСОК ЛИТЕРАТУРЫ

1. Barrio R., Blesa F. Systematic search of symmetric periodic orbits in 2DOF Hamiltonian systems // *Chaos, Solitons and Fractals*. 2009. V. 41. № 2. P. 560–582.
2. Jorba A., Zou M. A software package for the numerical integration of ODE by means of high order Taylor methods. 2004. P. 32. Preprint.
3. Algorithm 924: TIDES, a Taylor Series Integrator for Differential EquationS / Alberto Abad, Roberto Barrio, Fernando Blesa, Marcos Rodríguez // *ACM Transactions on Mathematical Software*. 2012. November. V. 39. № 1. Article No. 5.
4. Thompson I. *Understanding Maple*. Cambridge University Press, 2016. 228 p.
5. SymPy: symbolic computing in Python / Aaron Meurer, Christopher P. Smith, Mateusz Paprocki et al. // *PeerJ Computer Science*. 2017. V. 3. P. e103. Access mode: <https://doi.org/10.7717/peerjcs.103>
6. Wolfram S. *The Mathematica Book*. Wolfram Media, Inc., 2003. 1488 p.
7. Batkhin A.B. Some properties of doubly symmetric periodic solutions to Hamiltonian system // *International Conference Polynomial Computer Algebra '2019*; St. Petersburg, April 15–20, 2019 / Ed. by N.N. Vasiliev; Euler International Mathematical Institute. St. Petersburg, Russia: VVM Publisher, 2019. P. 25–28.
8. Батхин А.Б. Бифуркации периодических решений системы Гамильтона с дискретными симметриями // *Компьютерная алгебра: материалы Международной конференции*. Москва, 17–21 июня 2019 г. / Под ред. С.А. Абрамова, Л.А. Севастьянова. М.: РУДН, 2019. С. 90–97.
9. Батхин А.Б. О структуре фазового потока в окрестности симметричного периодического решения системы Гамильтона // *Препринты ИПМ им. М.В. Келдыша*. 2019. № 69.
10. Lamb J.S.W., Roberts J.A.G. Time-reversal symmetry in dynamical systems: A survey // *Physica D*. 1998. V. 112. P. 1–39.
11. Meyer K.R. *Periodic Solutions of the N-body Problem*. Lecture Notes in Mathematics, V. 1719. Berlin, Heidelberg, New York: Springer–Verlag, 1999.
12. Hénon M. *Generating Families in the Restricted Three-Body Problem*. Lecture Note in Physics. Monographs no. 52. Berlin, Heidelberg, New York: Springer, 1997. 278 p.
13. Уинтнер А. *Аналитические основы небесной механики*. М.: Наука, 1967. 523 с.
14. Якубович В.А., Старжинский В.М. *Линейные дифференциальные уравнения с периодическими коэффициентами*. М.: Наука, 1972. 720 с.
15. Маркеев А.П. *Точки либрации в небесной механике и космодинамике*. М.: Гл. ред. физ.-мат. литер. изд-ва. Наука., 1978. 352 с.
16. Пуанкаре А. *Новые методы небесной механики. Топология. Теория чисел*. М.: Наука, 1972. Т. II. С. 9–356.
17. Крейман Б.Б. Семейства периодических решений гамильтоновой системы с двумя степенями свободы. Несимметричные периодические решения плоской ограниченной задачи трех тел // *Космические исследования*. 2005. Т. 43. № 2. С. 1–23.
18. Методы анализа нелинейных динамических моделей / М. Холодниок, А. Клич, М. Кубичек, М. Марек. М.: Мир, 1991. 368 с.
19. Гантмахер Ф.П. *Теория матриц*. 4-е изд. изд. М.: Наука, 1988. 552 с.
20. Williamson J. On the normal forms of linear canonical transformations in dynamics // *American Journal of Mathematics*. 1937. Jul. V. 59. № 3. P. 599–617.
21. Брюно А.Д. *Ограниченная задача трех тел: Плоские периодические орбиты*. М.: Наука, 1990. 296 с.
22. Каримов С.Р., Сокольский А.Г. Метод продолжения по параметрам естественных семейств периодических движений гамильтоновых систем. 1990. Препринт ИТА АН СССР. № 9.
23. Meyer K.R., Offin D.C. *Introduction to Hamiltonian Dynamical Systems and the N-Body Problem*. Applied Mathematical Sciences, V. 90. 3rd edition edition. New York: Springer International Publishing, 2017.
24. Теория бифуркаций / В.И. Арнольд, В.С. Афраймович, Ю.С. Ильяшенко, Л.П. Шильников. М.: ВИНТИ АН СССР, 1985. Т. 5 из Современные проблемы математики. Фундаментальные направления.
25. Батхин А.Б., Батхина Н.В. *Задача Хилла*. Волгоград: Волгоградское научное издательство, 2009. 200 с.

26. *Себехей В.* Теория орбит: ограниченная задача трех тел. М.: Наука, 1982. 656 с.
27. *Morales-Ruiz J., Simó C., Simon S.* Algebraic proof of the non-integrability of hill's problem // *Ergodic Theory and Dynamical Systems*. 2005. V. 25. № 4. P. 1237–1256.
28. *Батхин А.Б.* Симметричные периодические решения задачи Хилла. II // *Космические исследования*. 2013. Т. 51. № 6. С. 497–510.
29. *Батхин А.Б.* Сеть семейств периодических орбит обобщенной задачи хилла // *ДАН*. 2014. Т. 458. № 2. С. 131–137.
30. *Hénon M.* Numerical exploration of the restricted problem. V. Hill's case: periodic prbits and their stability // *Astron. & Astrophys*. 1969. V. 1. P. 223–238.
31. *Hénon M.* Numerical exploration of the restricted problem. VI. Hill's case: non-periodic orbits // *Astron. & Astr.* 1970. no. 9. P. 24–36.
32. *Hénon M.* New families of periodic orbits in Hill's problem of three bodies // *Celest. Mech. Dyn. Astr.* 2003. V. 85. P. 223–246.
33. *Hénon M.* Families of asymmetric periodic orbits in Hill's problem of three bodies // *Celest. Mech. Dyn. Astr.* 2005. V. 93. P. 87–100.

---



---

**КОМПЬЮТЕРНАЯ  
АЛГЕБРА**


---



---

УДК 004.422.8

**ПРИМЕР МОДУЛЬНОГО РАСШИРЕНИЯ СИСТЕМЫ  
КОМПЬЮТЕРНОЙ АЛГЕБРЫ**

© 2020 г. М. Н. Геворкян<sup>a,\*</sup>, А. В. Королькова<sup>a,\*\*</sup>,  
Д. С. Кулябов<sup>a,b,\*\*\*</sup>, Л. А. Севастьянов<sup>a,c,\*\*\*\*</sup>

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

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

<sup>c</sup> Лаборатория теоретической физики, Объединенный институт ядерных исследований,  
ул. Жолио-Кюри 6, 141980 Дубна, Московская область, Россия

\*E-mail: [gevorkyan-mn@rudn.ru](mailto:gevorkyan-mn@rudn.ru)

\*\*E-mail: [korolkova-av@rudn.ru](mailto:korolkova-av@rudn.ru)

\*\*\*E-mail: [kulyabov-ds@rudn.ru](mailto:kulyabov-ds@rudn.ru)

\*\*\*\*E-mail: [sevastianov-la@rudn.ru](mailto:sevastianov-la@rudn.ru)

Поступила в редакцию 08.09.2019 г.

После доработки 19.10.2019 г.

Принята к публикации 19.10.2019 г.

Системы компьютерной алгебры представляют из себя сложные программные комплексы, охватывающие широкий спектр научных и практических проблем. Однако абсолютная полнота недостижима. И зачастую возникает задача создания пользовательского расширения существующей системы компьютерной алгебры. При этом следует учитывать расширяемость самой системы. В статье рассматривается технология расширения системы компьютерной алгебры SymPy низкоуровневым модулем, реализующим генератор случайных чисел.

DOI: 10.31857/S0132347420020065

## 1. ВВЕДЕНИЕ

Система компьютерной алгебры SymPy [1] является по своей сути модулем, написанным на языке Python [2, 3], поэтому для расширения функциональности SymPy достаточно написать функцию или модуль на самом Python.

Хотя язык Python является универсальным языком программирования и на нем можно реализовать любые алгоритмы, однако ему присущ существенный недостаток — малая производительность. Падение быстродействия особенно заметно в случае если в алгоритме присутствуют циклы. Данная проблема обусловлена динамической природой языка из-за которой даже элементарные типы данных, такие как `int` и `float` реализованы в стандартном интерпретаторе `cpython` в виде составных структур данных.

Язык Python хорошо подходит для прототипирования приложения. Также язык Python является в некотором роде клеем — он хорошо подходит для связывания разных библиотек вместе [4]. Но попытка создать на основе этого языка большую

быструю программу скорее всего обречена на провал. Причина, по которой он так успешно справляется с научными и инженерными задачами состоит в том, что Python использует низкоуровневый интерфейс к библиотекам, написанным на более вычислительно эффективных языках программирования. Таким образом создается впечатление, что Python работает так же быстро, как и код, написанный, например, на C++. Побочным эффектом этого является то, что для практического программирования на Python необходимо также программировать и на языках более низкого уровня. Если для задачи достаточно использования стандартных библиотек, то ничего кроме Python может и не понадобится. Однако, если необходимо добавить новую функциональность, то следует использовать более низкоуровневые языки программирования.

В данной статье рассмотрено использование модуля языка Python `ctypes` для интеграции C-функций в Python-программу. В качестве примера рассматривается библиотека, реализующая

генератор случайных чисел. Этот пример интересен тем, что он является достаточно ресурсоемким и его нецелесообразно делать на чистом Python.

Следует заметить, что идея использования компилируемого языка со статической типизацией для повышения производительности отдельных элементов программы, написанной на интерпретируемом языке с динамической типизацией, не является новой [5, 6]. Напротив, исторически сложилось множество технологических подходов, позволяющих осуществить такое расширение. Такое многообразие подходов затрудняет доступ начинающим в эту область. В нашей статье дается краткий обзор подходов, позволяющих повысить быстродействие Python-программ и краткая характеристика каждого из них.

Основной упор делается на использование встроенного модуля `ctypes` для непосредственного вызова C-функций из Python-программ. Обосновывается выбор именно данного модуля. Изложение носит практический характер и затрагивает вопросы создания библиотеки на языке Си, ее компиляции для дальнейшего использования с `ctypes` под различные платформы. Данная часть статьи может использоваться как введение в возможности `ctypes` для начинающих. В последней части статьи описывается созданный нами модуль, использующий `ctypes` для вызова функций из библиотеки на языке C, которая реализует ряд генераторов псевдослучайных чисел. Приведены тесты для сравнения предложенного модуля генератора псевдослучайных чисел с имеющимися генераторами из модуля `random` и библиотеки `NumPy`.

## 2. СПОСОБЫ ПОВЫШЕНИЯ БЫСТРОДЕЙСТВИЯ SYMPY/PYTHON ПРОГРАММ

Перечислим основные средства, которые используются в настоящее время для повышения быстродействия Python-программ.

- Использование сторонних библиотек, таких как `NumPy` [7, 8] и `SciPy` [9], в которых ресурсоемкие алгоритмы реализованы на языках C/C++ и Fortran, а сам Python используется как язык-связка для предоставления удобного программного интерфейса.

- Оптимизирующий статический компилятор `Cython` [10, 11] который позволяет транслировать Python-код в код на C/C++. При этом сама программа пишется на специальном диалекте Python, который позволяет применять статическую

типизацию в критичных для производительности участках кода.

- Проект `Numba` [12], представляющий собой JIT-компилятор Python кода. `Numba` позволяет писать программу на чистом Python, используя декораторы для функций и циклов, производительность которых необходимо повысить.

- Модуль `ctypes` [13] из стандартной библиотеки интерпретатора `Cpython`, который позволяет непосредственно вызвать C-функции, из статических или разделяемых (`shared`) библиотек, как обычные Python-функции.

Заметим, что ни одно из перечисленных средств не является универсальным. Специализированные библиотеки `NumPy` и `SciPy` нацелены в основном на научные и инженерные вычислительные задачи, поэтому реализуемый ими набор алгоритмов хоть и обширен, но ограничен рамками этой специализации. JIT-компилятор `Numba` дает существенный прирост скорости, однако проект пока находится на стадии разработки и его функциональные возможности ограничены стандартными сценариями применения.

Статический компилятор `Cython` на сегодняшний день является одним из самых часто используемых средств для повышения производительности. Его используют многие библиотеки, в том числе `NumPy` и `SciPy` для повышения производительности и интеграции библиотек на C/C++.

В данной работе мы используем модуль `ctypes`, поскольку для наших задач он имеет ряд преимуществ над `Cython`:

- это стандартный модуль `Cpython`, тогда как `Cython` необходимо устанавливать отдельно;
- не смешиваются несколько диалектов языка Python;

- `ctypes` особенно полезен, если необходимая функциональность уже реализована на C. В этом случае подготовка вызова функций крайне проста и занимает буквально несколько строк кода. Также его разумно применять в случае, если реализуемые C-функции просты, но активно используют циклы и работу с примитивными типами данных.

Таким образом, при использовании `ctypes` работа над программой делится на два этапа. На первом этапе программист реализует функции на языке C, компилирует и собирает из них статическую или разделяемую (динамическую) библиотеку. Второй шаг – отдельная реализация ряда функций-обертки уже на языке Python. Данные функции-обертки по сути представляют собой интерфейс для удобного вызова уже реализованных C-функций из Python-программ. Отметим, что функциональность вызова C-функций

реализуется с помощью модуля `ctypes`, входящего в стандартную библиотеку интерпретатора CPython.

### 3. МОДУЛЬ CTYPES

В данном разделе описан полный цикл создания библиотеки на языке Си и ее использование совместно с `ctypes`. Официальная документация `ctypes` приводит примеры использования функций из данного модуля, однако в ней не затрагивается процесс создания библиотеки на языке C.

Использование модуля `ctypes` начинается с загрузки файла библиотеки, поэтому прежде чем переходить к описанию базовой функциональности `ctypes`, приведем пример сборки статической и динамической (разделяемой) библиотек на примере компилятора языка C из набора компиляторов `gcc`. Авторы использовали `gcc` версии 8.3.0 под операционной системой Gnu Linux, дистрибутив Ubuntu 19.04.

#### 3.1. Компиляция C-функций и сборка библиотеки

Код типичной библиотеки на языке C состоит из набора файлов с исходным кодом (`src_01.c`, `src_02.c` и т.д.) и ряда заголовочных файлов (`header_01.h` и т.д.). Общепринятой практикой [14] является размещение всех файлов с исходным кодом в поддиректории `src` проекта, а заголовочных файлов в поддиректории `include`.

Для компиляции исходных файлов используются следующие ключи компилятора.

- `-c` – позволяет создать объектный файл, без сборки всей программы или библиотеки.
- `-Wall` – компилятор будет распечатывать сообщения не только о синтаксических ошибках, но и предупреждения, которые потенциально могут привести к некорректной работе программы.
- `-Werror` – все предупреждения будут интерпретироваться компилятором как ошибки.
- `-fPIC` – указывает компилятору о необходимости транслировать программу в позиционно-независимый машинный код (`position-independent code – PIC`), где все переходы осуществляются только по относительным адресам. Этот флаг важен, так как библиотека потенциально может быть загружена в любом месте программы.
- `-I./include` – указывает компилятору, что файлы *заголовков* следует искать в локальной директории `include` нашего проекта.
- `-L./lib` – указывает компилятору, что файлы *библиотек* следует искать в локальной директории `lib` нашего проекта. Важно соблюдать

последовательность и указывать флаг `-L` только после флага `-I`.

- После отладки программы можно также добавить флаг оптимизации `-O2` или `-O3`, помня, однако, что агрессивная оптимизация в некоторых случаях может привести к некорректной работе функций.

Все перечисленные флаги сохраняем в переменную окружения `CFLAGS` и для компиляции файла с исходным кодом в объектный файл для каждого файла выполняется следующая команда:

```
gcc -c $CFLAGS src/src_01.c -o
↪lib/obj_01.o
```

После того, как будут созданы все объектные файлы, их можно запаковать в статическую библиотеку с помощью утилиты `ar`, выполнив следующую команду:

```
ar crs libmy.a lib/obj_01.o lib/obj_02.o
```

Опции `crs` говорят о том, что нужно создать архив с заменой файлов, если таковые уже в нем есть и дополнительно создать индекс. После успешного выполнения команды получим файл статической библиотеки `libmy.a`, который можно будет использовать для подключения средствами `ctypes`.

Для создания статической библиотеки в среде Windows следует использовать опцию `-Wl`, которая позволит передать дополнительные опции компоновщику (`linker`) и указать с помощью опции компоновщика `--out-implib` путь к файлу статической библиотеки, которую необходимо создать.

```
gcc -shared lib/obj_01.o lib/obj_02.o
↪libmy.so -o bin/libmy.dll
↪-Wl,-out-implib,lib/win/libmy.a
```

При необходимости, можно создать не статическую, а разделяемую библиотеку:

```
gcc -shared lib/obj_01.o lib/obj_02.o
↪libmy.so
```

В результате получим файл разделяемой библиотеки `libmy.so`. Та же команда позволяет получить динамическую библиотеку и в системе Windows. Следует лишь указать расширение файла как `dll` вместо `so`.

#### 3.2. Загрузка библиотеки в Python-программу

Предполагая, что мы успешно скомпилировали и собрали библиотеку `libmy.so`, опишем процедуру ее импорта в Python-программу. Предполагаем, что файл библиотеки будет находиться в той же директории, что и наша Python-программа. Разберем следующий фрагмент кода.



```
import ctypes
import sys
import os
path = os.path.dirname(os.path.realpath(
    ↪ __file__ ))
if sys.platform.startswith('win'):
    clib = ctypes.CDLL(os.path.join(path,
    ↪ 'libmy.dll'))
else:
    clib = ctypes.CDLL(os.path.join(path,
    ↪ 'libmy.so'))
```

Вначале необходимо загрузить модуль `ctypes` и ряд дополнительных модулей. Далее получаем абсолютный путь до директории с программой. Затем определяем тип операционной системы и в зависимости от этого загружаем файл `.dll` или `.so`.

Стоит отметить, что загрузка файла библиотеки по абсолютному пути обязательна в том случае, если мы организуем нашу Python-программу в виде модуля и хотим хранить файл библиотеки внутри директории модуля.

### 3.3. Вызов функций из библиотеки

После импорта все функции библиотеки будут доступны для вызова в виде атрибутов объекта `clib`. Пусть, например, в библиотеке `libmy` присутствует следующая функция:

```
uint64_t uint64_var(uint64_t var) {
    uint64_t i = 9223372036854775807llu;
    printf("Function uint64_var, arg
    ↪ uint64 var = %llu\n", i);
    return i;
}
```

Для ее вызова из Python-программы можно использовать следующий код:

```
clib.uint64_var.argtypes =
    ↪ [ctypes.c_uint64]
clib.uint64_var.restype =
    ctypes.c_uint64
res = clib.uint64_var(0)
```

Перед вызовом функции мы указали тип аргумента используя список из одного элемента, так как аргумент единственный. Далее указывается тип возвращаемого значения, после чего можно вызвать требуемую функцию. В `ctypes` определены все стандартные типы языка C и вызов любой простой функции, принимающей и возвращающей аргументы базовых типов, укладывается в вышеприведенные три строки кода.

Рассмотрим чуть более сложный пример, когда аргумент передается в функцию по указателю. Пусть имеется следующая C-функция:

```
void change_var(double* var) {
    *var = 2.0;
}
```

Следующий код показывает способ вызвать эту функцию с помощью `ctypes`:

```
x = ctypes.c_double(1.0)
print(f"x = {x.value}")
clib.change_var(ctypes.byref(x))
print(f"x = {x.value}")
```

Здесь мы вначале с помощью конструктора `c_double` присвоили значение переменной `x`, а затем передали ее в виде аргумента функции `change_var`, указав дополнительно с помощью `byref`, что аргумент передается по ссылке. Так как функция не возвращает никаких значений, то не нужно указывать `restype`, а так как мы передали в качестве аргумента переменную уже известного типа, то указывать тип аргумента тоже не пришлось.

Наконец рассмотрим вызов функции, принимающей в качестве аргумента массив:

```
double avg_value(long long int array[],
    ↪ size_t len) {
    double avg = 0.0;
    for (size_t i = 0; i < len; ++i) {
        avg += (double) array[i] /
        ↪ (double) len;
        array[i] = 0.0;
    }
    return avg;
}
```

Функция `avg_value` принимает в качестве первого аргумента массив, а в качестве второго целое число – размер массива. Для удобного вызова этой функции из Python-кода можно написать следующую функцию-обертку.

```
def avg_value(l: list) -> float:
    """Обертка для avg_value"""
    clib.avg_value.restype =
    ↪ ctypes.c_longdouble
    A = (ctypes.c_longlong * len(l))(*l)
    n = ctypes.c_size_t(len(l))
    return clib.avg_value(A, n)
```

Вначале определяется возвращаемый тип (`long double`), затем выделяется память для массива, который сразу же инициализируется значениями из списка `l`. После чего создается переменная `n` типа `size_t` и происходит вызов C-функции.

Использование оберточных функций оправданно в большинстве случаев, так как позволяет спрятать рутинные действия по инициализации аргументов и указанию типов данных, предоставляя пользователю удобный интерфейс.

#### 4. ГЕНЕРАЦИЯ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

Получение истинно случайных чисел представляет достаточно трудную задачу. Обычно для этого используют разнообразные физические процессы. Основной проблемой генераторов истинно случайных чисел является низкая интенсивность генерации случайных чисел [15]. Поэтому для практических целей используют генераторы псевдослучайных чисел [16–20].

Пакет SymPy не реализует отдельных генераторов псевдослучайных чисел, так как вся необходимая функциональность присутствует в стандартном модуле `random` и в подмодуле `numpy.random` библиотеки NumPy.

Функции обоих модулей основаны на алгоритме под названием *вихрь Мерсенна* [21], который генерирует псевдослучайные равномерно распределенные последовательности беззнаковых целых чисел. Данный алгоритм позволяет получить качественную последовательность псевдослучайных чисел, однако отличается сравнительно низкой производительностью ввиду громоздкости самого алгоритма. В настоящее время появился ряд альтернативных алгоритмов [22–25], которые также генерируют качественные последовательности псевдослучайных чисел, но при этом выигрывают в быстродействии.

Современные алгоритмы генераторов псевдослучайных чисел используют побитовые логические операции и операции сдвига, поэтому естественным выбором для реализации таких алгоритмов являются системные языки программирования, обеспечивающие минимум абстракций в пользу максимального уровня быстродействия. Большинство разработчиков данных алгоритмов предоставляют также примеры реализаций на языках C или C++.

Такие реализации представляют собой компактные функции, сигнатура которых имеет следующий вид:

```
uint32_t generator(uint32_t seed[]);
/* или */
```

```
uint64_t generator(uint64_t seed[]);
```

где массив `seed` представляет собой начальные значения для инициализации генератора. Для генерации последовательности псевдослучайных чисел данную функцию достаточно вызвать в цикле  $N$  раз.

```
void rand_n(uint64_t N, uint64_t
seed[],
↪ uint64_t res[]) {
    for (uint64_t i=0; i < N; ++i) {
        res[i] = generator(seed);
    }
}
```

Внутреннее состояние генератора определяется набором чисел `seed` и сохраняется от вызова к вызову, так как массив `seed` передается по ссылке.

Для получения чисел из полуинтервала  $[0, 1)$  достаточно нормировать генерируемые числа. Нижеследующий код показывает как это сделать.

```
double normed_gen(uint64_t seed[]) {
    return (double) generator(seed) /
    ↪ (double) UINT64_MAX;
}
```

##### 4.1. Структура библиотеки

Авторами была реализована компактная библиотека на языке C [26], в которой был собран ряд современных генераторов псевдослучайных чисел [27, 22–25]. Библиотека имеет следующую структуру.

- В директории `src` располагаются файлы с исходным кодом, реализующие различные алгоритмы генераторов псевдослучайных чисел. Файлы названы именем алгоритма, реализация которого содержится внутри.

- В директории `include` содержится единственный файл заголовка, в котором объявлены все функции, реализованные в библиотеке.

- В директории `tools` находятся C-программа, реализующая утилиту командной строки `random`, с помощью которой можно запустить любой генератор для вывода сгенерированной последовательности на печать.

- Для сборки библиотеки и утилиты под операционной системой типа Unix написан `makefile`, а для сборки под ОС Windows командный `bat`-файл.

- Результатом компиляции и сборки будут файлы разделяемой и статической библиотек, расположенные в директориях `lib/shared` и `lib/static` соответственно. Также в директории `bin` будет собрана командная утилита `random`

##### 4.2. Обертка библиотеки с помощью `ctypes`

Вышеописанная библиотека была интегрирована в среду Python/SymPy с помощью стандартного модуля `ctypes` и оформлена в виде Python-модуля под названием `crandom`. Все функциональные возможности модуля реализованы в файле `crandom.py` в виде класса `Random`.

Для корректного функционирования требуются стандартные модули `random`, `typing`, `ctypes`, `sys` и `os`. Также для генерации массивов псевдослучайных чисел требуется библиотека NumPy.

Рассмотрим основные возможности `crandom` на примерах. Для запуска примеров использовал-

ся дистрибутив языка Python 3.6.8 Miniconda и интерактивная оболочка Jupyter 4.4.0.

Работа с модулем начинается с выбора и инициализации генератора. Рассмотрим пример:

```
import crandom
gen = crandom.Random('xorshift+')
gen.set_seed([233, 43])
```

Здесь мы создали объект `gen` который будет использовать алгоритм *xorshift+* своей работе. Также мы инициализировали генератор, передав ему два целых числа с помощью функции-метода `set_seed`. Если генератор не инициализировать явным вызовом `set_seed`, то будет использована функция `randint` из стандартного модуля `random`. Также следует отметить, что при выборе начальных значений следует придерживаться ряда рекомендаций [28] и числа 233, 43 были выбраны только для того, чтобы не загромождать пример.

Состояние генератора сохраняется в атрибуте `seed` объекта `gen`. В зависимости от типа генератора `seed` может быть как единственным беззнаковым целым числом, так и последовательностью беззнаковых целых чисел. Вызванная без аргументов, функция-метод `set_seed` самостоятельно определяет сколько целых чисел необходимо для инициализации.

После того, как объект `gen` создан и инициализирован, его можно использовать для получения последовательности псевдослучайных чисел заданного размера. Сделать это можно следующими способами.

```
r = gen.generate(size=10)
r = gen.generate(size=10, type=float)
r = gen(size=10)
```

При первом вызове будет сгенерирована последовательность из 10 беззнаковых целых чисел. При втором вызове необязательному аргументу `type` передано `float`, что приводит к генерации последовательности чисел с плавающей запятой из полуинтервала  $[0, 1)$ . Наконец третья строка показывает, что необязательно использовать функцию-метод `generate` так как в классе определен метод `__call__`, и сам объект можно вызывать как функцию.

Генерацию массива псевдослучайных чисел полностью осуществляет функция на языке C (для каждого генератора своя). Затем сгенерированный массив конвертируется в `numpy`-массив. Для конвертации вызывается функция `np.array` с опцией `copy=False`, что позволяет не копировать массив в памяти, а заместить по месту (*in place*).

Состояние генератора сохраняется средствами Python-программы. После того, как последовательность сгенерирована, в атрибут `seed` записываются последние элементы этой последовательности. Они будут использованы в качестве новых начальных значений.

Если приоритетом является экономия памяти, то генератор можно использовать в режиме итератора, так как в классе реализованы специальные функции-методы `_next_` и `_iter_`. Следующий пример иллюстрирует как это сделать.

```
gen.set_iterator(10, int)
for i in gen:
    print(i)
```

Инициализация итератора осуществляется функцией `set_iterator`. В качестве аргументов указывается количество чисел, которое должен произвести генератор, и тип чисел (`int` или `float`). Затем объект `gen` можно использовать в цикле, как стандартный `python`-итератор. При этом работает цикл, реализованный на Python, в результате чего производительность ниже чем при использовании функции `generate`. Состояние генератора сохраняется также, как и при использовании `generate`.

#### 4.3. Тестирование производительности

Использование C-функций позволяет достичь высокой производительности. Сравним, например, работу нашего генератора с генератором из библиотеки `NumPy randint`. Быстродействие измеряется командой `%timeit`, встроенной в интерактивные оболочки *iPython* и *Jupyter*. В качестве аргумента ей передается фрагмент кода, быстродействие которого следует замерить. В качестве результата распечатывается значение среднего времени работы кода, среднееквадратичное отклонение и количество выполнений кода.

Для получения последовательности 64-битных беззнаковых целых чисел, при вызове функции `randint` необходимо указать значение `np.uint64` аргумента `dtype`, а также указать нижнюю (`low`) и верхнюю (`high`) границы.

Для обеих функций команда `%timeit` произвела 7 запусков по 10000 повторений в каждом. Среднее время работы функции `randint` составило 85.4 со стандартным отклонением в 1.67 микросекунды. Для функции `generate` – 39.4 микросекунды, со стандартным отклонением 1.18 микросекунды.

Так как в библиотеке `NumPy` генераторы также реализованы на языке C, то полученную разницу можно объяснить большей эффективностью алгоритма *xorshift*. Следует также отметить,

что компиляция библиотеки выполнялась с ключом оптимизации `-O3`.

Отметим, что в стандартном модуле `random` нет функции, позволяющей сгенерировать последовательность целых чисел. Взамен этого можно воспользоваться многократным вызовом функции `randint` и списковой сборкой, что будет заведомо медленней NumPy-версии (замер времени дает значение 11.6 миллисекунды).

Для проверки корректности реализации генераторов был проведен ряд визуальных тестов. Были построены следующие диаграммы:

- диаграмма рассеяния (scatter plot);
- диаграмма лага (Lag-plot);
- диаграмма автокорреляции в зависимости от лага (auto-correlation function plot, ACF-plot).

Данные визуальные тесты позволяют оценить насколько полученная последовательность псевдослучайных чисел является независимо распределенной и выявить лишь грубые ошибки. В качестве более строгих тестов были использованы наборы тестов DieHarder [29], TestU01 [30, 31], PractRand [32] и gjrand [33]. Отчеты по тестам DieHarder доступны в репозитории [26].

## 5. ЗАКЛЮЧЕНИЕ

Созданная нами библиотека и модуль для ее интеграции в среду SymPy/Python могут быть легко расширены добавлением новых функций на языке C и соответствующих оберточных функций на языке Python.

Отметим также, что в случае генераторов псевдослучайных чисел выбор модуля `ctypes` был обоснован, так как реализуемые алгоритмы используют побитовые операции и примитивные типы данных, поэтому их реализация полностью на системном языке программирования дает существенное увеличение производительности и уменьшение расхода памяти.

## БЛАГОДАРНОСТИ

Публикация подготовлена при поддержке Программы РУДН “5-100”.

## СПИСОК ЛИТЕРАТУРЫ

1. *Lamy R.* Instant SymPy Starter. Packt Publishing, 2013. 52 p.
2. *Слаткин Б.* Секреты Python. М.: Вильямс, 2017. 272 с.
3. *Любанович Б.* Простой Python. Современный стиль программирования. Бестселлеры O'Reilly. М.: Питер, 2019. 480 с.
4. *Кулябов Д.С., Королькова А.В., Севастьянов Л.А.* Новые возможности второй версии пакета компьютерной алгебры Cadabra // Программирование. 2019. № 2. С. 41–48.
5. *Aladjev V., Bogdevicius M.* Maple: Programming, Physical and Engineering Problems. 2006. 403 p. ISBN: 1596820802.
6. *Corless R.M.* Essential Maple 7: An Introduction for Scientific Programmers. Springer Science and Business Media, 2007. 282 p. ISBN: 9780387215570.
7. *Idris I.* NumPy Cookbook. Packt Publishing, 2012. 226 p.
8. *Oliphant T.E.* Guide to NumPy. 2 edition. CreateSpace Independent Publishing Platform, 2015. 364 p.
9. *Oliphant T.E.* Python for Scientific Computing // Computing in Science and Engineering. 2007. V. 9. № 3. P. 10–20.
10. *Behnel S., Bradshaw R., Citro C.* et al. Cython: The Best of Both Worlds // Computing in Science and Engineering. 2011. mar. V. 13. № 2. P. 31–39.
11. *Smith K.* Cython. A Guide for Python Programmers. O'Reilly Media, 2015. 238 p.
12. *Lam S.K., Pitrou A., Seibert S.* Numba: a LLVM-based Python JIT compiler // Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC. LLVM '15. Austin, Texas: ACM Press, 2015. nov. P. 7.1–6.
13. *Spreitzenbarth M., Uhrmann J.* Mastering Python Forcensics. Packt Publishing, 2015. 192 p.
14. *Клеменс Б.* Язык C в XXI веке. М.: ДМК Пресс, 2017. 376 с.
15. *Galton F.* Dice for statistical experiments // Nature. 1890. V. 42. № 1070. P. 13–14.
16. *Кнут Д.Э.* Искусство программирования. 3 изд. М.: Вильямс, 2001. Т. 2. 832 с.
17. *Дроздова И.И., Жилин В.В.* Генераторы случайных и псевдослучайных чисел // Технические науки в России и за рубежом: материалы VII Междунар. науч. конф. М.: Буки-Веди, 2017. nov. С. 13–15.
18. *Колчин В.Ф., Севастьянов Б.А., Чистяков В.П.* Случайные размещения. М.: Наука, 1976. 224 с.
19. *Тюрин Ю.Н., Макаров А.А.* Статистический анализ данных на компьютере / Под ред. В.Э. Фигурнова. М.: ИНФРА, 1998. 528 с.
20. *Gevorkyan M.N., Demidova A.V., Korolkova A.V.* et al. Pseudo-random number generator based on neural network // Selected Papers of the 8th International Conference “Distributed Computing and Gridtechnologies in Science and Education” / Ed. by Vladimir Koronkov, Andrey Nechaevskiy, Tatiana Zaikina, Elena Mazhitova. Vol. 2267 of CEUR Workshop Proceedings. Dubna, 2018. sep. P. 568–572.
21. *Matsumoto M., Nishimura T.* Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-random Number Generator // ACM Trans. Model. Comput. Simul. 1998. V. 8. № 1. P. 3–30.
22. *Marsaglia G.* Xorshift RNGs // Journal of Statistical Software. 2003. V. 8. № 1. P. 1–6.
23. *Panneton F., L'Ecuyer P.* On the Xorshift Random Number Generators // ACM Trans. Model. Comput. Simul. 2005. V. 15. № 4. P. 346–361.

24. *Boldi P., Vigna S.* On the Lattice of Antichains of Finite Intervals // *Order*. 2018. mar. V. 35. № 1. P. 57–81.
25. PCG: A Family of Simple Fast Space-Efficient Statistically Good Algorithms for Random Number Generation: Rep.: HMC-CS-2014-0905 / Harvey Mudd College; Executor: Melissa E O’Neill. Claremont, CA: 2014.
26. *Gevorkyan M.N., Kulyabov D.S., Korolkova A.V., Sevastianov L.A.* Random Number Generators for Computer Algebra Systems. 2019. URL: <https://bitbucket.org/yamadharma/articles-2019-rng-generator-code>.
27. *Rose G.G.* KISS: A bit too simple // *Cryptography and Communications*. 2018. V. 10. № 1. P. 123–137.
28. *Jones D.* Good practice in (pseudo) random number generation for bioinformatics applications. 2010.
29. *Brown R.G., Eddelbuettel D., Bauer D.* Dieharder: A Random Number Test Suite. 2017. URL: [http://www.phy.duke.edu/~rgb/General/rand\\_rate.php](http://www.phy.duke.edu/~rgb/General/rand_rate.php).
30. *L’Ecuyer P., Simard R.* TestU01: A C library for empirical testing of random number generators // *ACM Transactions on Mathematical Software (TOMS)*. 2007. V. 33. № 4. P. 22.
31. *L’Ecuyer P., Simard R.* TestU01 – Empirical Testing of Random Number Generators. 2009. URL: <http://simul.iro.umontreal.ca/testu01/tu01.html>.
32. *Doty-Humphrey C.* PractRand official site. 2018. URL: <http://prcrand.sourceforge.net/>.
33. Gjrnd random numbers official site. 2014. URL: <http://gjrnd.sourceforge.net/>.

## АЛГОРИТМ ПОСТРОЕНИЯ АНАЛОГА ФОРМУЛЫ БИНЕ

© 2020 г. В. И. Кузоватов<sup>a,\*</sup>, А. А. Кытманов<sup>a,\*\*</sup>, О. И. Кузоватова<sup>a,\*\*\*</sup><sup>a</sup> Сибирский федеральный университет,  
660041 Красноярск, пр. Свободный, 79, Россия

\*E-mail: kuzovатов@yandex.ru

\*\*E-mail: aakytm@gmail.com

\*\*\*E-mail: oik17@yandex.ru

Поступила в редакцию 31.08.2019 г.

После доработки 12.09.2019 г.

Принята к публикации 20.10.2019 г.

Приведен алгоритм построения аналога формулы Бине, которая используется при нахождении функционального соотношения для классической дзета-функции Римана. Алгоритм реализован в системе компьютерной алгебры Maple. Приведен пример, демонстрирующий работу данного алгоритма.

DOI: 10.31857/S0132347420020077

## 1. ВВЕДЕНИЕ

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

Вместе с тем, неалгебраические системы уравнений возникают в различных областях знания. В частности, в процессах, описываемых системами дифференциальных уравнений с правыми частями, разложимыми в ряд Тейлора, актуален вопрос об определении числа стационарных состояний в множествах определенного вида (и их локализации). Эта проблема приводит к задачам построения алгоритмов для определения числа корней заданной системы уравнений в различных множествах, определения самих корней, исключения части неизвестных из системы. В частности, в монографии [2] приведены многочисленные примеры из химической кинетики, где требуются алгоритмы исключения неизвестных. Здесь важно применение разработанных методов для качественного и численного анализа математических

моделей термодинамики процессов горения и катализа с целью получения условий воспламенения, взрыва и критических явлений в химически реагирующих системах. Для приложений, в том числе, например, для уравнений химической кинетики, важной задачей является исследование зависимостей решений систем нелинейных, в том числе и неалгебраических, уравнений от параметров. В вычислительном плане эта задача является достаточно трудоемкой. Ее степень сложности сильно зависит от размерности пространства неизвестных. Поэтому снижение этой размерности за счет исключения переменных может привести к упрощению исходной задачи.

Метод исключения неизвестных из систем нелинейных алгебраических уравнений, основанный на теории многомерных вычетов, был предложен Л.А. Айзенбергом в [3] в 1977 г. Дальнейшие модификации метода были предложены А.П. Южаковым, А.К. Цихом, В.И. Быковым, А.М. Кытмановым, М.З. Лазманом в конце прошлого века [2]. Эти идеи были в последствии развиты в работах [4–6]. Алгоритмический метод (разработанный на основе идей Л.А. Айзенберга и А.П. Южакова) был предложен М. Елкади и А. Ижером в работе [7]. Идея метода заключалась в нахождении определенных вычетов интегралов, связанных со степенными суммами корней (в положительных степенях) заданной системы уравнений, избегая нахождения самих корней и применяя затем к ним рекуррентные формулы Ньютона. По сравнению с классическим мето-

дом, данный метод сокращал время работы алгоритма, не повышая при этом кратность корней.

Еще одним методом исключения неизвестных служит построение результата двух целых функций. Хорошо известен классический результат Сильвестра для двух многочленов и метод исключения неизвестных, на нем основанный. Для неалгебраических функций такое понятие не было изучено ранее. Лишь в последние года в работах [8, 9] обсуждается один подход к нахождению результата двух целых функций, основанный на рекуррентных формулах Ньютона.

В работе [10] В.И. Кузоватовым и А.А. Кытмановым была предложена программная реализация алгоритма построения семейства аналогов формулы Плана (см., например, пример 7 главы 7 из [11]), впервые полученных В.И. Кузоватовым и А.М. Кытмановым в работе [12] при некоторых ограничениях. В работе [13] В.И. Кузоватовым эти ограничения были сняты.

Среди физических приложений классической формулы Плана и некоторых ее обобщений можно отметить их использование в теории квантованных полей для перенормировки тензора энергии импульса скалярного поля в различных фридмановских моделях Вселенной, а также при вычислении вакуумного среднего тензора энергии импульса квантованных полей в различных полных и неполных многообразиях (эффект Казимира). Подробное изложение этих вопросов можно найти в [14].

Целью данной работы является разработка и программная реализация алгоритма построения семейства аналогов формулы Бине (см., например, глава 12, п. 12.32 из [11]), полученных В.И. Кузоватовым и А.М. Кытмановым в работе [15]. Данный алгоритм будет использоваться для алгоритмизации и программной реализации функциональных соотношений на многомерные аналоги дзета-функции Римана, которые являются важным инструментом в создании методов исключения неизвестных из систем нелинейных уравнений, как было показано в работе [16].

## 2. КЛАССИЧЕСКАЯ ФОРМУЛА БИНЕ И ЕЕ ПРИМЕНЕНИЕ

Классическая формула Бине (см., например, глава 12, п. 12.32 из [11]) выражает значение логарифмической производной гамма-функции Эйлера  $\Gamma(z)$  (в случае, если вещественная часть  $z$  положительна) через следующие интегралы:

$$\frac{d}{dz} \ln \Gamma(z) = -\frac{1}{2z} + \ln z - 2 \int_0^{\infty} \frac{t dt}{(z^2 + t^2)(e^{2\pi t} - 1)}.$$

Данное интегральное представление используется при нахождении функционального соотношения

(см., например, глава 2, п. 9 из [17]) для классической дзета-функции Римана  $\zeta(s)$ . Рассмотрим для вещественных  $x$ :

$$\frac{\Gamma(1+x)}{\Gamma(1+x)} - \ln x = -2 \int_0^{\infty} \frac{t}{t^2 + x^2} \left( \frac{1}{e^{2\pi t} - 1} - \frac{1}{2\pi t} \right) dt.$$

Подставляя найденное соотношение в интегральное представление для дзета-функции Римана  $\zeta(s)$  и меняя порядок интегрирования, можно получить функциональное соотношение (см., например, глава 2, п. 9 из [17]) для классической дзета-функции Римана.

Напомним (см., например, глава 2, п. 9 из [17]), что интегральное представление для дзета-функции Римана  $\zeta(s)$  в полосе  $0 < \text{Re } s < 1$  имеет вид:

$$\zeta(s) = -\frac{\sin \pi s}{\pi} \int_0^{\infty} \left( \frac{\Gamma'(1+x)}{\Gamma(1+x)} - \ln x \right) x^{-s} dx.$$

Если говорить об обобщениях дзета-функции, то И.М. Гельфанд, Б.М. Левитан и Л.А. Дикий изучали (см., например, работы [18, 19, 20]) дзета-функцию, ассоциированную с собственными значениями оператора Штурма–Лиувилля в 50-х годах прошлого века. Ее значение оказалось связанным со следом данного оператора. Их подход был развит [21] далее В.Б. Лидским и В.А. Садовничим (60 годы), которые рассмотрели класс целых функций одного переменного, определили для них дзета-функцию корней и исследовали ее область аналитического продолжения. В работе [22] С.А. Смагин и М.А. Шубин строят дзета-функцию эллиптических операторов и операторов более общего вида, доказывают возможность мероморфного продолжения дзета-функции и дают некоторую информацию о полюсах.

## 3. ПОСТАНОВКА ЗАДАЧИ И ИЗВЕСТНЫЕ РЕЗУЛЬТАТЫ

Пусть  $f(z)$  – целая функция конечного порядка роста  $\rho$  в  $\mathbb{C}$ . Рассмотрим уравнение

$$f(z) = 0. \tag{1}$$

Обозначим через  $N_f = f^{-1}(0)$  множество всех корней уравнения (1) (каждый корень считается столько раз, какова его кратность). Число корней не более чем счетно.

Дзета-функция  $\zeta_f(s)$  корней  $z_n$  уравнения (1) определяется следующим образом:

$$\zeta_f(s) = \sum_{z_n \in N_f} (-z_n)^{-s},$$

где  $s \in \mathbb{C}$ .

В работе [23] В.И. Кузоватовым и А.А. Кытмановым с использованием теории вычетов получе-

ны два интегральных представления для дзета-функции, построенной по нулям целой функции конечного порядка роста на комплексной плоскости. С помощью этих представлений описана область, в которую эта дзета-функция продолжается.

Будем предполагать, что  $z_n = -q_n$ ,  $q_n > 0$ , где  $q_n$  образуют некоторую последовательность натуральных чисел.

Рассмотрим функцию ( $z = x + iy$ )

$$F(f, 2\pi iz) = \sum_{n=1}^{\infty} e^{z_n 2\pi iz} = \sum_{n=1}^{\infty} e^{-q_n 2\pi iz}. \quad (2)$$

С помощью замены  $e^{-2\pi iz} = w$  ряд (2) приводится к виду  $\sum_{n=1}^{\infty} w^{q_n}$ , то есть

$$G(w) = \sum_{n=1}^{\infty} f_n w^n, \quad (3)$$

где коэффициенты  $f_n$  определяются следующим образом:

$$f_n = \begin{cases} 1, & n = q_k; \\ 0, & n \neq q_k, \end{cases}$$

и, следовательно,  $\lim_{n \rightarrow \infty} \sqrt[n]{|f_n|} = 1$ .

Заметим, что в ряде (3) бесконечное число коэффициентов  $f_n$  отлично от нуля.

В дальнейшем ограничимся рассмотрением классов рациональных функций  $G(w)$ , для которых справедливо представление (3). Как показывает теорема Сеге (см., например, [24 §6.1]), степенной ряд (3), коэффициенты которого  $f_n$  могут принимать лишь конечное число различных значений, или представляет собой рациональную функцию, или непродолжаем за пределы единичного круга. В случае рациональности суммы ряда (3)

$$G(w) = \frac{P(w)}{1 - w^N},$$

где  $P(w)$  – многочлен, а  $N$  – некоторое натуральное число.

Предположим, что  $\deg P(w) = N$ , то есть

$$P(w) = a_1 w + a_2 w^2 + \dots + a_{N-1} w^{N-1} + w^N, \quad (4)$$

где коэффициенты  $a_j \in \{0, 1\}$  ввиду разложения (3),  $j = 1, \dots, N - 1$ . Коэффициенты  $a_j$  зависят от распределения  $\{q_n\}$ .

Приведем явное выражение функции  $F(f, 2\pi iz)$  через функцию  $G(w)$ . Будем иметь

$$F(f, 2\pi iz) = \frac{P(e^{-2\pi iz})}{1 - e^{-2\pi iz N}}.$$

Заметим, что функции  $F(f, 2\pi y)$  и  $1 + F(f, -2\pi y)$ , входящие в правую часть формулы (5), определяются [10] следующим образом ( $t = e^{2\pi y}$ ):

$$F(f, 2\pi y) = -\frac{a_1 t^{N-1} + a_2 t^{N-2} + \dots + a_{N-1} t + 1}{1 - t^N},$$

$$1 + F(f, -2\pi y) = \frac{1 + a_1 t + a_2 t^2 + \dots + a_{N-1} t^{N-1}}{1 - t^N}.$$

Для сокращения записи обозначим

$$W(x, y) = \varphi(x - iy) F(f, 2\pi y) + \varphi(x + iy) [1 + F(f, -2\pi y)].$$

**Теорема 1** (аналог формулы Бине, [15]). *Справедливо равенство*

$$\frac{P(1)}{N} \sum_n \frac{1}{(z + q_n)^2} = \frac{P(1)}{N} \left( \frac{1}{2} \varphi(x_2) - \frac{1}{2} \varphi(0) \right) + \int_0^{x_2} \varphi(z) dz + \frac{1}{i} \int_0^{\infty} W(0, y) dy - \frac{1}{i} \int_0^{\infty} W(x_2, y) dy, \quad (5)$$

где суммирование берется по всем точкам  $q_n$ , лежащим в отрезке  $[1; x_2]$ ,  $x_2$  – целое положительное число, полином  $P(w)$  определен формулой (4).

Здесь

$$\varphi(\zeta) = \frac{Q_1(\zeta) Q_2(\zeta) e^{\zeta^2}}{(z + \zeta)^2},$$

$$Q_1(\zeta) = \prod_{j \in J} \left( 1 - \frac{\zeta}{\alpha_j} \right) e^{\frac{\zeta}{\alpha_j}}, \quad \alpha_j = \frac{j}{N},$$

$$Q_2(\zeta) = \sum_{n=1}^{\infty} \frac{1}{Q_1(q_n) e^{q_n^2} \Psi'(q_n) (\zeta - q_n)},$$

$$\Psi(\zeta) = \prod_{n=1}^{\infty} \left( 1 - \frac{\zeta}{q_n} \right),$$

$$J = \{j : j = 1, 2, \dots, j \neq mN, m \in \{q_n\}\}.$$

Если говорить о методах исследования, то локальные вычеты обобщают обычный вычет Коши функции одного переменного, и их вычисление в наиболее важных для приложений случаях является конструктивной процедурой, которую несложно реализовать с помощью символьных вычислений на компьютере. Интегралы, выражаемые через локальные вычеты, появляются в различных прикладных задачах. В монографии [25] подробно обсуждается применение многомерных вычетов и, в частности, дается решение задачи о вычислении ошибки квантования двумерных рекурсивных цифровых фильтров. Для современной теоретической физики одним из важных классов инте-



гралов являются кратные интегралы Меллина–Барнса, которые изучаются с помощью локальных вычетов. Еще одно направление исследований, в котором используются интегралы, выражающиеся через локальные вычеты, связано с изучением достаточных условий алгебраичности интегралов, зависящих от параметра. Данные методы могут быть использованы в теории формальных языков и грамматик [26].

#### 4. ОПИСАНИЕ АЛГОРИТМА

**Алгоритм 1:** Алгоритм построения семейства аналогов формулы Бине

**Input:** Список коэффициентов  $a_j \in \{0, 1\}$ ,  $j = 1, \dots, N - 1$ ; целое положительное число  $x_2$ .

**Output:** Список, состоящий из левой и правой частей формулы (5).

**begin**

$$P(w) := a_1 w + \dots + a_{N-1} w^{N-1} + w^N$$

$$G(w) := \frac{P(w)}{1 - w^N}$$

$$Q := \{1 \cdot a_1, 2 \cdot a_2, \dots, (N - 1) \cdot a_{(N-1)}, N, (N + 1) \cdot a_1, (N + 2) \cdot a_2, \dots, x_2 \cdot a_{x_2}\}$$

$Q$  := список ненулевых элементов из  $Q$

$$J := \{1, \dots, x_2 \cdot N\} \setminus \{Q \cdot N\}$$

$$A = (\alpha_1, \dots, \alpha_J) := J/N$$

$$Q_1(\zeta) = \prod_{j \in J} \left(1 - \frac{\zeta}{\alpha_j}\right) e^{\frac{\zeta}{\alpha_j}}$$

$$\Psi(\zeta) = \prod_{q_n \in Q} \left(1 - \frac{\zeta}{q_n}\right)$$

$$Q_2(\zeta) = \sum_{q_n \in Q} \frac{1}{Q_1(q_n) e^{q_n} \Psi'(q_n) (\zeta - q_n)}$$

$$\varphi(\zeta) = \frac{Q_1(\zeta) Q_2(\zeta) e^{\zeta^2}}{(z + \zeta)^2}$$

$$L := \frac{P(1)}{N} \sum_{q_n \in Q} \frac{1}{(z + q_n)^2}$$

$$R := \frac{P(1)}{N} \left( -\frac{1}{2} \varphi(0) + \frac{1}{2} \varphi(x_2) \right) + \int_0^{x_2} \varphi(z) dz + \frac{1}{i} \int_0^\infty (\varphi(-iy) G(e^{-2\pi y}) + \varphi(iy) [1 + G(e^{2\pi y})]) dy - \frac{1}{i} \int_0^\infty (\varphi(x_2 - iy) G(e^{-2\pi y}) + \varphi(x_2 + iy) [1 + G(e^{2\pi y})]) dy$$

**return**  $\{L, R\}$

#### 5. ПРИМЕР

Алгоритм был реализован в среде Maple 2016 64bit. Полный код программы доступен по адресу [https://github.com/aakytmanov/Binet\\_formula](https://github.com/aakytmanov/Binet_formula). Вычисления производились на машине Intel Core i7-4790 (3.6 GHz) с 32 Gb RAM под управлением Windows 7 Enterprise x64 SP1. Время счета для приведенного примера составило менее 0.1 секунды.

Пример 1.

Пусть  $q_n = \{1, 2\}$ ,  $x_2 = 2$ . Тогда  $N = 2$ ,  $\alpha_j = \{1/2, 3/2\}$ ,  $P(w) = w + w^2$ ,

$$F(f, 2\pi y) = \frac{1}{e^{2\pi y} - 1},$$

$$1 + F(f, -2\pi y) = \frac{1}{1 - e^{2\pi y}},$$

$$Q_1(\zeta) = (1 - 2\zeta) \left(1 - \frac{2}{3}\zeta\right) e^{8\zeta/3},$$

$$\Psi(\zeta) = (1 - \zeta) \left(1 - \frac{\zeta}{2}\right), \quad \Psi'(\zeta) = -\frac{3}{2} + \zeta,$$

$$Q_2(\zeta) = \frac{1}{Q_1(1) e \Psi'(1) (\zeta - 1)} + \frac{1}{Q_1(2) e^4 \Psi'(2) (\zeta - 2)} = \frac{-3}{e^{11/3} \left(-\frac{1}{2}\right) (\zeta - 1)} + \frac{1}{e^{28/3} \frac{1}{2} (\zeta - 2)} = \frac{6\Psi(\zeta)}{e^{11/3} (\zeta - 1)} + \frac{2\Psi(\zeta)}{e^{28/3} (\zeta - 2)} = 3\zeta e^{-11/3} - 6e^{-11/3} + \zeta e^{-28/3} - e^{-28/3}.$$

Таким образом,

$$\varphi(\zeta) = \frac{1}{3} \frac{e^{\zeta^2} e^{8\zeta/3} (2\zeta - 1)(2\zeta - 3)}{(z + \zeta)^2} \times (3\zeta e^{-11/3} - 6e^{-11/3} + \zeta e^{-28/3} - e^{-28/3}) \quad (6)$$

и (5) принимает вид

$$\frac{1}{(z + 1)^2} + \frac{1}{(z + 2)^2} = \frac{1}{2} (\varphi(2) - \varphi(0)) + \int_0^2 \varphi(z) dz + \frac{1}{i} \int_0^\infty \frac{\varphi(-iy) - \varphi(iy)}{e^{2\pi y} - 1} dy - \frac{1}{i} \int_0^\infty \frac{\varphi(2 - iy) - \varphi(2 + iy)}{e^{2\pi y} - 1} dy,$$

где функция  $\varphi(\zeta)$  в данном случае определена формулой (6).

Входными данными алгоритма в этом случае будут список коэффициентов  $a_j$ , состоящий из [1],

некоторое заданное положительное целое число  $x_2 = 2$  и построенная функция  $\varphi(z)$ , например:  
`> Binet([1], 2, z->varphi(z), Phi);`

### БЛАГОДАРНОСТИ

Исследования, представленные в работе, были выполнены при поддержке следующих фондов: второй автор использовал поддержку гранта РФФ 18-71-10007 (разработка и программная реализация алгоритма); первый и третий авторы поддержаны грантом РФФИ 18-31-00019 (постановка технического задания для разработки алгоритма, тестирование, вычисление примеров).

### СПИСОК ЛИТЕРАТУРЫ

1. *von zur Gathen J., Gerhard J.* Modern Computer Algebra (3rd edition), Cambridge University Press, 2013.
2. *Вуков V.I., Кутманов A.M., Лазман M.Z.* Elimination methods in polynomial computer algebra, Kluwer Academic Publishers, Dordrecht-Boston-Basel, 1998.
3. *Айзенберг Л.А.* Об одной формуле обобщенного многомерного логарифмического вычета и решении систем нелинейных уравнений // Докл. АН СССР. 1977. Т. 234. № 3. С. 505–508.
4. *Кутманов А.А.* Об аналогах рекуррентных формул Ньютона // Изв. вузов. Математика. 2009. № 10. С. 40–50.
5. *Кутманов А.А.* Алгоритм вычисления степенных сумм корней для класса систем нелинейных уравнений // Программирование. 2010. Т. 36. № 2. С. 55–63.
6. *Кутманов А.А., Кутманов А.М., Мышкина Е.К.* Finding Residue Integrals for Systems of Non-algebraic Equations in  $\mathbb{C}^n$  // Journal of Symbolic Computation. 2015. V. 66. P. 98–110.
7. *Elkadi M., Yger A.* Residue calculus and applications // Publ. Res. Inst. Math. Sci. 2007. V. 43. № 1. P. 55–73.
8. *Кутманов А.М., Наприенко Y.M.* An approach to define the resultant of two entire functions // Journal Complex Variables and Elliptic Equations. 2017. V. 62. № 2. P. 269–286.
9. *Кутманов А.М., Мышкина Е.К.* On Some Approach for Finding the Resultant of Two Entire Functions // J. Siberian Federal Univ. Math. Phys. 2019. V. 12. № 4. P. 434–438.
10. *Кузоватов В.И., Кутманов А.А.* Алгоритм построения аналога формулы Плана // Программирование. 2018. № 2. С. 35–41.
11. *Whittaker E.T., Watson G.N.* A Course of Modern Analysis, Cambridge University Press, Cambridge, 1927.
12. *Кузоватов V.I., Кутманов A.M.* On an Analog of the Plan's Formula // Journal of Contemporary Mathematical Analysis. 2018. V. 53. № 3. P. 139–146.
13. *Кузоватов В.И.* Об одном обобщении формулы Плана // Известия вузов. Математика. 2018. № 5. С. 41–51.
14. *Саарян А.А.* К формуле суммирования Абеля–Плана // Изв. АН Армянской ССР. Физика. 1986. Т. 21. № 5. С. 262–265.
15. *Кузоватов В.И., Кутманов А.М.* Об одном аналоге интегрального представления Бине // Сибирские электронные математические известия. 2020. Т. 17. С 1–14.
16. *Кутманов А.М., Myslivets S.G.* On the Zeta-Function of Systems of Nonlinear Equations // Siberian Math. J. 2007. V. 48. № 5. P. 863–870.
17. *Titchmarsh E.C.* The Theory of the Riemann Zeta-Function, Oxford University Press, Oxford, 1951.
18. *Гельфанд И.М., Левитан Б.М.* Об одном простом тождестве для собственных значений дифференциального оператора второго порядка // Доклады ак. наук СССР. 1953. Т. 88. № 4. С. 593–596.
19. *Дикий Л.А.* Дзета-функция обыкновенного дифференциального уравнения на конечном отрезке // Изв. АН СССР. Сер. матем., 1955. Т. 19. № 4. С. 5187–200.
20. *Дикий Л.А.* Формулы следов для дифференциальных операторов Штурма–Лиувилля // УМН. 1958. Т. 13. № 3. С. 111–143.
21. *Lidskii V.B., Sadovnichii V.A.* Regularized Sums of Zeros of a Class of Entire Functions // Functional Analysis and Its Applications. 1967. V. 1. № 2. P. 133–139.
22. *Smagin S.A., Shubin M.A.* On the Zeta-Function of a Transversally Elliptic Operator // Russian Mathematical Surveys. 1984. V. 39. № 2. P. 201–202.
23. *Кузоватов V.I., Кутманов A.A.* On the Zeta-Function of Zeros of Some Class of Entire Functions // J. Siberian Federal Univ. Math. Phys. 2014. V. 7. № 4. P. 489–499.
24. *Bieberbach L.* Analytische Fortsetzung, Springer-Verlag, Berlin, 1955.
25. *Tsikh A.K.* Multidimensional residues and their applications, AMS, Providence, 1992.
26. *Egorushkin O.I., Kolbasina I.V., Safonov K.V.* On application of multidimensional complex analysis in formal language and grammar theory // Applied Discrete Mathematics. 2017. V. 37. P. 76–89.

О МАШИННОМ ДОКАЗАТЕЛЬСТВЕ  
ДЛЯ АРИФМЕТИКИ ДРОБЕЙ НАД КОЛЬЦОМ С НОД

© 2020 г. С. Д. Мешвелиани\*

*Институт программных систем им. А.К. Айламазяна РАН  
152021 Ярославская обл., Переславский р-н, с. Веськово, ул. Петра Первого, д. 4 “а”, Россия**\*E-mail: mechvel botik.ru*

Поступила в редакцию 31.08.2019 г.

После доработки 20.09.2019 г.

Принята к публикации 10.10.2019 г.

Представлена разработка доказательных программ арифметики дробей в общем случае области с операцией наибольшего общего делителя. Это есть малая часть библиотеки DoCon - А машинно-проверяемых программ для компьютерной алгебры, разработанной автором. В этой системе программы включают определения соответствующих математических понятий и доказательства главных свойств применяемых алгоритмов. Эти доказательства проверяются компилятором. Применяется чисто функциональный язык программирования Agda, который также предоставляет конструкции зависимых типов. Описано построение формальных машинно-проверяемых доказательств для оптимизированного способа сложения дробей.

DOI: 10.31857/S0132347420020089

## ВВЕДЕНИЕ

Вычисления в поле частных в общем случае кольца с НОД выполняют различные системы компьютерной алгебры. Но здесь мы рассматриваем только алгоритмы и программы, сопровождаемые машинно-проверяемыми доказательствами. Наша программа опирается на базовую библиотеку вычислительной алгебры DoCon - А [1, 2] доказательных программ, написанных автором на языке Agda [4, 5]. Этот язык основан на чистой функциональности, “ленивом” способе вычисления по умолчанию, поддержке обобщенного программирования (generic programming), аппарате зависимых типов. Приблизительно можно считать, что это язык Haskell, расширенный аппаратом зависимых типов. Зависимые типы позволяют адекватно описать алгебраическую область, зависящую от обычного значения ([1], Введение). Кроме того, этот аппарат позволяет вставлять в программу доказательства утверждений, и эти доказательства будут проверяться компилятором. Становится возможным описывать метод вычисления в программе так, как это делается в учебниках и научных статьях, вместе с доказательством правильности.

Проект DoCon - А является попыткой перенести обширную библиотеку программ вычислительной алгебры DoCon [3] с языка Haskell на математически значительно более адекватный язык.

Техника проверки доказательств компилятором (точнее – проверяльщиком типов, type checker) основана на изоморфизме Карри–Ховарда [6, 7]. Всякое утверждение  $S$  представляется подходящим типом  $T$  (зависящим от значений). Доказательство для  $S$  есть любая функция (завершающийся алгоритм), которая выдает любое значение  $v$  типа  $T$ . Проверяльщик типов проверяет отношение  $v : T$  посредством символьных вычислений с выражениями типов. Таким образом доказательство утверждения проверяется до начала компиляции в исполняемый код.

Недоказуемое высказывание соответствует пустому типу  $\perp$  с пустым множеством значений. Отрицание высказывания соответствует функции, отображающей соответствующий высказыванию тип в пустой тип. Импликация выражена типом  $S \rightarrow T$  всех функций из  $S$  в  $T$ , где  $S$  и  $T$  представляют соответствующие утверждения. Конъюнкция выражена произведением  $S \times T$  типов, дизъюнкция выражена суммой  $S \cup T$  типов. Доказательство индукцией по построению данного выражается в виде рекурсивно заданной функции. Применение леммы выражается в виде вызова функции, представляющей доказательство леммы.

Пока ограничиваемся только конструктивными доказательствами [8, 9] (без использования принципа Маркова для доказательства завершаемости алгоритмов). В частности, всякий объект существует лишь как итог некоторого данного ал-

горитма, и должно быть дано доказательство завершаемости этого алгоритма. Мы часто пользуемся законом исключенного третьего – это возможно в тех случаях, когда приведен алгоритм разрешения соответствующего отношения.

В этой статье мы не можем дать более подробные объяснения по данной системе программирования. Объяснения и примеры содержатся в [4, 1] и в руководстве по библиотеке [2].

Выпуск DoCon-A 2.02 библиотеки реализует задание иерархии общих классических структур (теорий) алгебры: Полугруппа (Semigroup), Группа (Group), Кольцо (Ring), Целостное кольцо (IntegralRing), Евклидово кольцо (EuclideanRing), Поле (Field) и некоторых других. Из конструкторов алгебраических областей реализованы конструкторы поля частных для кольца с НОД и кольца остатков для евклидова кольца.

Часть проекта DoCon-A, посвященная разработке доказательной программы для общей арифметики дробей, опирается на реализацию этой иерархии понятий.

#### Обозначения и словоупотребление:

- ниже мы называем систему DoCon-A “библиотека”,
- имена НОД, gcd, GCD обозначают наибольший общий делитель,
- имя Agda иногда пишем кириллицей и склоняем по падежам.

#### 1.1. О синтаксисе языка Agda

Имена операторов и отношений отделяются пробелами. Например, в строке

$$a+b\approx 0 : a + b \approx 0\#$$

программы  $a+b\approx 0$  есть имя значения, двоеточие отделяет имя значения от выражения типа, символы ‘+’ и ‘≈’ в выражении типа суть соответственно имя оператора сложения и имя двуместного отношения,  $0\#$  есть имя нулевой постоянной. Вся эта строка означает объявление: значение  $a+b\approx 0$  имеет тип  $a + b \approx 0\#$  (и этот тип зависит от значений  $a, b, 0\#$ ).

Знак подчеркивания в имени отношения или операции означает, что в синтаксисе программы во входном выражении на этой позиции ставится выражение аргумента этой операции.

Знак равенства ‘=’ – это определение идентификатора (присваивание) – часть синтаксиса программы. Равенство  $\approx$  – это знак отношения равенства на некоторой области, это отношение задается для каждой области программой пользователя или стандартной библиотекой.

## 2. ПОЛЕ ЧАСТНЫХ

Поле частных над кольцом  $R$  (в программе – Fraction  $R$ ) ([10] параграф 13) имеет смысл для целостного кольца (в программе – IntegralRing), то есть для коммутативного кольца без делителей нуля (если  $x * y = 0$ , то  $x = 0$  или  $y = 0$ ). Здесь также рассматривается более сильное условие кольца с НОД – GCDRing).

Обозначим  $\approx$  отношение равенства на  $R$ ,  $+$ ,  $*$  – действия сложения и умножения в области  $R$ .

Элементы области частных  $Q = \text{Fraction } R$  (дроби) определены как формальные записи

$$n/d, \quad n, d \in R, \quad d \neq 0.$$

Равенство  $\approx$  на  $Q$  определено как

$$n/d \approx n'/d' = n * d' \approx n' * d$$

Умножение, сложение и деление на  $Q$  определены как

$$n/d * n'/d' = (n * n') / (d * d')$$

$$n/d + n'/d' = (n * d' + n' * d) / (d * d')$$

$$\text{divide } n/d \ n'/d' = n/d * d'/n'$$

для  $n' \neq 0$ .

Нетрудно доказать ([10] параграф 13), что для целостного кольца  $R$  операции, заданные таким образом, удовлетворяют закону поля (коммутативное кольцо, в котором каждый ненулевой элемент обратим по умножению).

#### Цель исследования:

выразить на языке Agda известные оптимизированные алгоритмы сложения и умножения дробей в случае кольца с НОД, сопровождая их машинно-проверяемыми доказательствами. Также рассматривается особый способ сложения дробей, в котором сокращение на НОД применяется более интенсивно.

Вышеописанные наивные способы вычисления действий  $*$ ,  $+$  на практике в системах вычислительной алгебры не применяются из-за того, что во многих случаях вычисления в цикле приводят к быстрому росту размера знаменателя. Как пример худшего случая предложим вычисление отрезка гармонического ряда:

$$\sum_{k=1}^n 1/k \quad \text{для } n = 4, 8, 12, 16, \dots$$

В этой статье нет теоретических оценок стоимости вычисления, наша цель обозначена выше.

## 3. НАИВНЫЕ И ОПТИМИЗИРОВАННЫЕ АЛГОРИТМЫ АРИФМЕТИКИ ДРОБЕЙ

Объявление (на языке Agda)

```
record Prefraction : Set where
  constructor preFr
  field num : C
        denom : C
        denom≠0 : denom ≠ 0#
```

определяет, что дробь над целостным кольцом (с носителем, обозначенным  $C$ ) состоит из числителя  $num$ , знаменателя  $denom$  и свидетельства (доказательства, данного на языке  $Agda$ )  $denom \neq 0$  того, знаменатель не равен нулю.

Равенство на  $C$  обозначено  $\approx$ , неравенство обозначено  $\neq$ .

Отношение  $\equiv$  равенства на типе  $Prefraction$  выражается кодом

```

_+'_ : Op2 Prefraction


```

Здесь функция  $nz * nz$  применяется к значениям  $d \neq 0$ ,  $d' \neq 0$  (свидетельствам соответствующих неравенств) и возвращает свидетельство неравенства  $d * d' \neq 0$ . Эта функция выражает закон целостного кольца.

Полуоптимизированная арифметика опирается на понятия делимости и взаимной простоты. Отношение делимости определяется в программе для произвольной полугруппы в виде объявления типа

```

Coprime : Rel C _
Coprime a b = (c : C) → c | a → c | b → c | ε
    
```

—“для любого  $c$  (если  $c$  делит  $a$  и  $c$  делит  $b$ , то  $c$  обратим — делит единицу  $\epsilon$ )”.

Понятие дроби выражается объявлением *записи* (*record*)

```

record Fraction : Set where
  constructor fr'
  field num : C
        denom : C
        denom≠0 : denom ≠ 0#
        coprime : Coprime num denom
    
```

Оно включает требование *coprime* взаимной простоты числителя и знаменателя. Назовем это

```

record GCD (a b : C) : Set where
  constructor gcd'
  field
    proper : C          -- собственно НОД(a, b)
    divides1 : proper | a
    divides2 : proper | b
    greatest : ∀ {d} → d | a → d | b → d | proper
    
```

```

_=_'_ : Rel Prefraction _
f ='_ g = (num f * denom g) ≈ (num g * denom f)
    
```

Заметим, что в правой части определения стоит выражение типа, зависящего от значений  $f$  и  $g$ , это способ выразить утверждения в виде типов.

В данной статье мы не обсуждаем действие умножения дробей, а сосредоточимся на более сложном действии, на их сложении. Сложение в наивной арифметике дробей программируется на типе  $Prefraction$  и записывается как

```

_|_ : Rel C _
x | y = ∃ \q → x * q ≈ y
    
```

Здесь квантор существования имеет конструктивный смысл. Именно, всякое значение типа  $x | y$  является парой  $(q, eq)$ , где  $eq$  свидетельство для равенства  $x * q \approx y$ .

Свойство взаимной простоты определяется для произвольного моноида объявлением

представление сокращенным. Равенство  $\equiv_{fr}$  на типе  $Fraction$  такое же, как задано выше на типе  $Prefraction$ . Чтобы получить дробь в сокращенном виде применяется функция  $gcd$  и сокращение дроби на полученное значение. Это требует наличия структуры кольца с НОД ( $GCDRing$ ) для кольца  $R$ , то есть такого целостного кольца, на котором заданы функция (алгоритм)

```
gcd : (a b : C) → GCD a b,
```

и функция  $\_|\_?$  разрешения отношения  $\_|_$  деления. При этом понятие НОД определяется в виде объявления записи

Последние три поля записи представляют условия, которым значение `proper` должно удовлетворять. Например, условие `greatest` значит “для любого `d` (если `d` делит `a` и делит `b`, то `d` делит `proper`)”. Условия делимости включают в себя значения дополняющих множителей. Например, значение `divides1` имеет вид пары  $(q_1, eq_1)$ , где  $eq_1$  есть свидетельство равенства  $proper * q_1 \approx a$ .

Далее, функция

---

```

+_1_ : Op2 Fraction
(fr' n1 d1 d1≠0) +1 (fr' n2 d2 d2≠0) =
  fraction (n1 * d2 + n2 * d1) (d1 * d2) d1d2≠0
where
d1d2≠0 = nz*nz d1≠0 d2≠0

```

---

– сложить наивным способом, потом сократить на НОД. Назовем этот способ полуоптимизированным.

Оптимизированный (как мы его условно называем) способ сложения дробей применяет сокращение на НОД на ранних стадиях.

Здесь и ниже мы используем следующие обозначения, связанные со структурой GCD. Функция `gcd` возвращает запись GCD, из которой извлекаются некоторые готовые значения помимо объявленных в полях записи. Например, вычисление `gcd d1 d2` для знаменателей дает запись, для которой собственно значение НОД переименовано в `g`, частное  $d_1/g$  переименовано в  $d_1'$ ,  $d_2/g$  переименовано в  $d_2'$ .

Оптимизированный способ сложения дробей  $n_1/d_1$  и  $n_2/d_2$  выражается через следующие предварительные вычисления:

```

g = gcd d1 d2;
d1' = d1 /' g;          d2' = d2 /' g
s = n1 * d2' + n2 * d1'; g1 = gcd s g
s' = s /' g1

```

(в наших обозначениях и в программе знак умножения пишется явно). Здесь `/'` обозначает деление нацело в области  $\mathbb{R}$  в случае, когда доказано отношение делимости для аргументов. Здесь и

`fraction : (a b : C) → b ≠ 0 → Fraction` принимает значения для числителя и знаменателя и свидетельство неравенства нулю знаменателя и выдает соответствующую дробь в сокращенном виде. Она применяет вычисление НОД и деление нацело в  $\mathbb{R}$ . Также эта функция строит доказательство взаимной простоты полученных после сокращения числителя и знаменателя. В этой разновидности арифметики дробей сумма дробей вычисляется функцией

---

```

+fr_ : Op2 Fraction
(fr' n1 d1 d1≠0 coprime-n1d1) +fr (fr' n2 d2 d2≠0 coprime-n2d2) =
  fr' s' ddg' ddg'≠0 coprime-s'-ddg'

```

ниже символ `/'` дан как комментарий. В программе вместо этого действия готовые значения дополняющих множителей  $d_1'$ ,  $d_2'$ ,  $s'$ ,  $g'$  извлекаются из записи итога применения функции `gcd`, они накапливаются заодно в ходе вычисления НОД. Оптимизированный способ взят, с переобозначениями, из книги [11], параграф 4.5.1, первая половина страницы 355. Буквально способ вычисления в [11] в наших обозначениях выражается формулой

$$s' / (d_1' * (d_2 /' g_1))$$

В программе он для удобства доказательства изменен к формуле

$$s' / ((d_1' * (d_2 /' g)) * (g /' g_1))$$

Дробь получаются равные, а стоимость вычисления существенно не отличается. В целом, программируется способ

```

(FSum)
g = gcd d1 d2;
d1' = d1 /' g;          d2' = d2 /' g
s = n1 * d2' + n2 * d1'; g1 = gcd s g
s' = s /' g1;          g' = g /' g1
dd = d1' * d2';      ddg' = dd * g'

```

И в этих обозначениях функция сложения задается как

Здесь `fr'` конструктор (`тер`) дроби, `s'` и `ddg'` полученные числитель и знаменатель. Значения `ddg' ≠ 0` и `coprime-s'-ddg'` суть соответственно свидетельства неравенства нулю знаменателя и взаимной простоты пары  $(s', ddg')$ .

Способ из [11] дан для области целых чисел. Здесь же он применяется к произвольному кольцу с НОД. Законность и способ такого обобщения доказываются ниже. В [11] пишется о начальной проверке часто встречающегося условия взаимной простоты двух знаменателей. Для него сумма дробей вычисляется легче. Но мы эту проверку пропускаем потому, что в случае обратимости `g` вычисления по формулам (`FSum`) все-равно сильно удешевляются.

Этот способ на примере гармонического ряда (Раздел 2) показывает большое убыстрение в сравнении с полуоптимизированным способом `+1`. Для  $n = 16000$  он в 500 раз быстрее, чем `+1`, и в 8 раз быстрее, чем встроенная арифметика рациональных чисел библиотеки Glasgow Haskell - 7.10.2. Такое быстрое действие можно объяснить на интуитивном уровне тем, что при формировании числителя и конечном вычислении НОД в способе `+1` всякий лишний общий множитель в аргументах этих действий сильно увеличивает стоимость последних умножения и вычисления НОД. Конечно, каждый из трех вышеприведенных способов окажется быстрее двух других на некотором нарочно подобранном семействе примеров. Имеет смысл выводить оценки средней стоимости вычисления сложения дробей как функции размера числителей и знаменателей для способов `+1` и `+fr`. Но здесь мы рассматриваем другие вопросы.

Приведем некоторые соображения об оптимизации. В каких случаях описанный выше оптимизированный способ оправдывает свое название? В случае целых чисел сокращение на нетривиальный НОД операндов умножения приводит к укорачиванию двоичного кода операндов и к удешевлению умножения. И в среднем приводит к удешевлению последующих действий при построении числителя суммы. Не зря этот способ приводится в [11]. В случае области  $K[x_1, \dots, x_n]$  многочленов над полем  $K$  деление операндов на нетривиальный НОД уменьшает полную степень операндов. При прочих равных условиях уменьшение полной степени ведет к удешевлению умножения многочленов, и так далее. Еще очевиднее это в случае конечного поля  $K$ , ибо тогда стоимость действий над коэффициентами ограничена постоянной. Эти соображения показывают, что имеет смысл рассматривать оптимизированный способ сложения дробей в общем случае и строить машинно-проверяемое доказательство его правильности.

### 3.1. Доказательство правильности в части равенства

Во-первых, требуется построить формальное доказательство на языке Agda того, что оптимизированный способ действительно выдает сумму дробей. При этом наивный способ сложения (Раздел 3) есть определение сложения дробей. То есть для доказательства правильности оптимизированного способа необходимо доказать, что формальные выражения

$$s'/ddg' \text{ и } (n_1 * d_2 + n_2 * d_1) / (d_1 * d_2)$$

представляют равные дроби. То есть доказать равенство

$$s' * (d_1 * d_2) \approx (n_1 * d_2 + n_2 * d_1) * (d_1' * d_2' * g')$$

(Corr1)

Но в программе наивное и оптимизированное сложения определены на разных типах (второй тип включает условие взаимной простоты). Для формализации доказательства рассматриваются два представления дроби: `Prefraction` и `Fraction`. Также рассматриваются два отношения равенства: `'` на типе `Prefraction` и `=fr` на типе `Fraction`. Для доказательства равенства (Corr1) рассматриваются три способа сложения дробей:

- наивный способ `+` на типе `Prefraction`,
- полуоптимизированный способ `+1` на типе `Fraction`,
- оптимизированный способ `+fr` на типе `Fraction`.

Нужные доказательства переносятся с версии (`Prefraction`, `'`, `+`) на версию (`=fr`, `+1`), и далее, на (`=fr`, `+fr`). В этом подходе законы поля доказываются для трех версий арифметики дробей: для (`Prefraction`, `+`, `*`), затем для (`Fraction`, `+1`, `*1`), затем для (`Fraction`, `+fr`, `*fr`). Такой подход обусловлен следующими причинами.

- Первая версия работает в наибольшей общности.
- Вторая версия может оказаться более эффективной, чем третья для некоторых задач.
- Третья версия часто оказывается наиболее эффективной.
- Законы поля просто доказываются для первой версии. И остается доказать, что вторая и третья версии выдают итог равный (как дробь) итогу, выдаваемому первой версией. Таким способом законы поля формально доказываются для второй и третьей версий арифметики дробей.

Например, покажем, как сочетательное свойство сложения доказывается для этих трех версий арифметики дробей. Преобразование между двумя представлениями дроби выражается двумя функциями:

```

toPre : Fraction → Prefraction
toPre (fr' n d d≠0 _) = preFr n d d≠0
                                     - удалить условие взаимной простоты

fromPre : Prefraction → Fraction
fromPre (preFr n d d≠0) = fraction n d d≠0
                                     - привести к сокращенному виду

```

### Функции

```

fromPre◦toPre : ∀{f} → fromPre (toPre f) =fr f
fromPre◦toPre {f} = fr-sym {f} {fromPre (toPre f)} (f=fr=fraction-f f)

toPre◦fromPre : ∀{f} → toPre (fromPre f) =' f
toPre◦fromPre {preFr n d d≠0} = <пропуск>

```

выражают доказательство того, что отображения  $\text{toPre}$  и  $\text{fromPre}$  являются взаимно обратными (и, следовательно, взаимно-однозначными). Функции

```

+1via+' : (f g : Fraction) → (f +1 g) =fr (fromPre (toPre f +' toPre g))
+1via+' f g = =fr-refl {f +1 g}

```

```

toPre+homo : (f g : Fraction) → toPre (f +1 g) =' (toPre f +' toPre g)
toPre+homo = <тело функции пропущено>

```

```

fromPre+homo : (f g : Prefraction) →
  fromPre (f +' g) =fr
  (fromPre f +1 fromPre g)
fromPre+homo = <тело функции пропущено>

```

представляют простое доказательство того, что отображения  $\text{toPre}$  и  $\text{fromPre}$  являются гомоморфизмами относительно операций  $+_1$  и  $+'$ . Функция

```

+1cong : ==+1 Preserves2 ==fr_ → ==fr_ → ==fr_

```

представляет доказательство того, что операция  $+_1$  конгруэнтна, то есть согласована с равенством  $=fr$ . Это доказательство выводится из свойства конгруэнтности отображения  $\text{toPre}$  и из леммы  $+_1\text{via}'$ .

Сочетательный закон для наивного сложения  $+'$  доказывается так. Для дробей  $n_1/d_1, n_2/d_2, n_3/d_3$  свойство сочетательности выражается равенством

$$n * d' \approx n' * d, \quad (2)$$

где  $n$  выражение для числителя дроби  $(n_1/d_1 +' n_2/d_2) +' n_3/d_3$ , и  $n'$  выражение для числителя дроби  $n_1/d_1 +' (n_2/d_2 +' n_3/d_3)$ . Подобным образом составляются выражения для знаменателей  $d$  и  $d'$ . Левая и правая части равенства (2) суть многочленные выражения с целыми коэффициентами от переменных  $n_1, d_1, n_2, d_2, n_3, d_3$ . Доказательство равенства (2) состоит в приведении этих выражений к нормальному виду многочлена

(раскрытие скобок, приведение подобных членов и так далее) с использованием законов коммутативного кольца. При этом каждый шаг такой нормализации записывается в виде шага доказательства на языке Agda для равенства  $\approx$  в кольце  $R$ . Это делается автоматически по вызову функции специального доказывателя равенств (что важно, написанной на Agде – так же, как все в библиотеке Agды) из модуля `InCommutativeSemiring` библиотеки. Это сокращает большой участок кода формального доказательства до одного вызова библиотечной функции.

Следующая функция строит доказательство сочетательного закона для операции  $+_1$  путем сведения его к доказательству этого закона для  $+'$ .

Поясним, как нужно читать нижеприведенное доказательство. Сначала надо прочесть присваивания, данные после ключевого слова `where`. Затем читать левую колонку выражений между `begin-fr` и `end-fr`, затем читать соответствующую



шую правую колонку выражений. Левая колонка представляет последовательность дробей, в которой каждая дробь  $f$  равна следующей в смысле от-

ношения  $=fr$ , а правая часть против выражения  $f$  представляет применение функции, воплощающей доказательство этого равенства.

```

+_1assoc : FP=fr.Associative _+_1_
+_1assoc f g h =
begin-fr
  (f +_1 g) +_1 h           =fr[ +_1cong e0 h=from-h' ]
  from (f' +' g') +_1 (from h') =fr[ e1 ]
  from ((f' +' g') +' h')     =fr[ fromPre-cong (+'assoc f' g' h') ]
  from (f' +' g'+h')         =fr[ fromPre+homo f' (g' +' h') ]
  f'' +_1 (from (g' +' h'))   =fr[ +_1cong f''=f (fromPre+homo g' h') ]
  f +_1 (g'' +_1 h'')         =fr[ +_1cong_2 (+_1cong g''=g h''=h) ]
  f +_1 (g +_1 h)
end-fr
where
from = fromPre; f' = toPre f
g' = toPre g; h' = toPre h
f'' = from f'; g'' = from g'
h'' = from h'
f''=f = fromPreotoPre {f}
g''=g = fromPreotoPre {g}
h''=h = fromPreotoPre {h}
h=from-h' = =fr-sym (fromPreotoPre {h})

e0 : (f +_1 g) =fr from (f' +' g')
e0 =
=fr-sym
(begin-fr
  from (f' +' g') =fr[ fromPre+homo f' g' ]
  f'' +_1 g''      =fr[ +_1cong f''=f g''=g ]
  f +_1 g
end-fr
)
e1 : (from (f' +' g')) +_1 (from h') =fr from ((f' +' g') +' h')
e1 = =fr-sym (fromPre+homo f'+g' h')

```

Доказательство каждого шага из правой колонки дано в виде конструкции  $=fr[...]$ . Это не особая конструкция языка, а вызов стандартной функции  $\_=[\_]\_$  трех аргументов (написанной на Агде). Также стандартными функциями являются префиксная функция  $begin\_$  и постфиксная  $\_□$  — они в программе переименованы соответственно в  $begin-fr\_$  и  $\_end-fr$ . Вся конструкция  $begin-fr \dots \_end-fr$  наглядно показывает вывод равенства с повторным приме-

нением закона транзитивности  $trans$ . Доказательство использует свидетельство  $+\'assoc$  сочетательного закона для  $+$ , то, что  $fromPre$  является изоморфизмом относительно двух операций сложения дробей и что операция  $+$  конгруэнтна по каждому аргументу (лемма  $+_1cong$ ).

Теперь покажем перенесение доказательств для операции  $+_1$  на доказательства для операции  $+fr$  — с полуоптимизированного способа на оптимизированный. Функция

---

```

+fro+1 : ∀ f g → (f +fr g) =fr (f +1 g)
+fro+1 (fr' n1 d1 d1≠0 coprime-n1d1) (fr' n2 d2 d2≠0 coprime-n2d2) =
s' *den2≈num2*ddg'
where
sum2 = n1 * d2 + n2 * d1
G     = proper (gcd sum2 d1d2)
num2  = div sum2 G
den2  = div d1d2 G
s' *den2 ≈ num2*ddg' : s' * den2 ≈ num2 * ddg'
...

```

---

представляет доказательство того, что эти два способа сложения дают равные дроби. Здесь обозначения взяты из определения операции +fr данного в формуле (FSum). Целевым равенством является

$$s' * den2 \approx num2 * ddg' \quad (\text{Corr2})$$

Четыре значения в этом равенстве суть некоторые рациональные выражения от значений  $n_1, d_1, n_2, d_2, g, g', g_1, G, n_1', d_1', n_2', d_2'$ . Чтобы упростить цель до равенства многочленных выражений обе части (Corr2) умножаются на значение  $Gg_1 = G * (g * g_1)$ . Получается равенство (Corr2'), не содержащее знаменателей. Равенство (Corr2') доказывается приведением его частей к одному и тому же нормальному виду многочлена, а также подстановкой равенств, полученных из формул (FSum) (и извлеченных из структур GCD), например,  $g * d_1' \approx d, g_1 * g' \approx g$ . Наконец, целевое равенство (Corr2) выводится из (Corr2') вызовом (cancelNonzeroLFactor Gg<sub>1</sub>). Функция cancelNonzeroLFactor представляет доказательство леммы:

*В целостном кольце для каждого ненулевого элемента а выполнено свойство сокращения*

$$\forall b \ c \ (a * b \approx a * c \implies b \approx c).$$

Библиотека включает простое доказательство этого, вместе с сотней или двумя лемм, сопровождающих описание классических структур алгебры.

Далее, доказательство сочетательного закона для действия +fr составляется с использованием доказательства +<sub>1</sub>assoc для действия +<sub>1</sub> и доказательства равенства +fr<sup>o</sup>+<sub>1</sub> для двух способов сложения дробей. Это делается так же, как в выше описанном перенесении доказательства сочетательного закона с версии '+' на версию +<sub>1</sub>. Таким же образом другие доказательства переносятся с версии '+' на +<sub>1</sub>, и далее на +fr.

Рассмотрим теперь построение свидетельств необходимых для составления итоговой дроби. Эти свидетельства (конструктивные доказательства) составляются для утверждений

---

```

coprime : Coprime s' ddg'
d1d2≠0# : d1 * d2 ≠ 0#
g≠0      : g ≠ 0#
ddg'≠0   : (d1 * d2) * g' ≠ 0#
g'≠0     : g' ≠ 0#
d1'≠0   : d1' ≠ 0#

```

и некоторых других. Первое из них обсуждается ниже. Остальные весьма просто получаются из свойства отсутствия делителей нуля в R и из леммы о делимости  $\forall x \ y \rightarrow x \mid y \rightarrow y \neq 0 \# \rightarrow x \neq 0 \#$ .

### 3.2. Доказательство взаимной простоты для итога суммы дробей

Продолжаем использовать обозначения формул (FSum) Раздела 3. Осталась главная часть доказательства: составление свидетельства для свойства взаимной простоты значений s' и ddg':

$$\text{Coprime } s' \ ((d_1' * d_2') * g').$$

Из доказательства этого свойства следует, что в оптимизированном сложении нахождение последнего НОД можно пропустить. Доказательство основано на обобщенной лемме Евклида:

(LE) для любых элементов a,b,c кольца c

НОД выполнено свойство

если Coprime a b и Coprime a c, то Coprime a (b \* c).

Цель (Coprime s' ddg') доказывается по следующей схеме.

- Coprime d<sub>1</sub>' n<sub>1</sub> (== Coprime n<sub>1</sub> d<sub>1</sub>/g) следует из Coprime n<sub>1</sub> d<sub>1</sub>.

- Coprime d<sub>1</sub>' d<sub>2</sub>' выполнено из-за того, что d<sub>2</sub>' и d<sub>1</sub>' суть дополняющие множители в структуре GCD d<sub>1</sub> d<sub>2</sub>.

- По лемме Евклида, выполнено отношение Coprime d<sub>1</sub>' n<sub>1</sub>d<sub>2</sub>'.

- Докажем теперь Coprime s d<sub>1</sub>'. Предположим, что некоторый x делит s и делит d<sub>1</sub>'. Из равенства  $s = n_1 * d_2' + n_2 * d_1'$  следует  $x \mid n_1 d_2'$ .

Из выведенного выше отношения  $\text{Coprime } d_1' \ n_1 d_2'$  следует, что  $x$  обратим.

Таким же способом доказывается утверждение  $\text{Coprime } s \ d_2'$ .

Далее, по лемме Евклида, имеем  $\text{Coprime } s \ d_1' d_2'$ .

Так как  $s'$  делит  $s$ , то выполнено отношение  $\text{Coprime } s' \ d_1' d_2'$ .

$s'$  и  $g'$  суть дополняющие множители в  $\text{GCD } s \ g$ . Поэтому выполнено  $\text{Coprime } s' \ g'$ . Из леммы Евклида теперь следует  $\text{Coprime } s' \ ddg'$ .

По этой схеме в программе построено формальное доказательство в виде функции на языке Agda.

Ниже  $'|'$  обозначает отношение делимости в кольце  $R$ ,

$x \sim y$  обозначает, что  $x$  и  $y$  ассоциированы, то есть делят друг друга.

### 3.3. Отступление: обобщенная лемма Евклида

В учебниках лемма Евклида формулируется так:

если простое число  $p$  делит  $a \cdot b$ ,

то  $p \mid a$  или  $p \mid b$ .

И доказательство использует или свойство однозначности разложения на множители, или свойство кольца главных идеалов. В нашем случае произвольного кольца с НОД такая формулировка леммы и такие условия не применимы. Поэтому в предыдущей версии программы добавлялось условие кольца с однозначным разложением на множители, а лемма Евклида формулировалась не для простого числа  $p$ , а для простого элемента этого кольца. При этом доказательство последнего свойства в оптимизированном сложении сильно усложнялось. Но позже оказалось, что известное изящное доказательство леммы Евклида в более общей формулировке, в условии кольца с НОД — как выражено выше в утверждении (LE). И это позволяет сильно упростить программу и повысить ее общность. Доказательство обобщенной леммы Евклида является, по-видимому, фольклором. Позже автор нашел доказательство в электронном сообщении на форуме любителей математики в компьютерной сети. Приведем его здесь, переписав его в более ясном виде.

**Лемма GCD\*.** Для кольца с НОД  $R$  и для любых  $a, b, c$  из  $R$  выполнено отношение  $\text{gcd}(ac, bc) \sim c \cdot \text{gcd}(a, b)$ .

*Доказательство.*

В случае  $c = 0$  отношение  $\text{gcd}(0, 0) \sim 0$  выполнено, так как из определения НОД (Раздел 3) следует, что  $\text{gcd}(0, 0) = 0$ .

Рассмотрим случай  $c \neq 0$ . Делимость  $c \cdot \text{gcd}(a, b) \mid \text{gcd}(ac, bc)$  сразу следует из того, что  $c \cdot \text{gcd}(a, b)$  делит  $ac$  и делит  $bc$ .

Осталось доказать  $\text{gcd}(ac, bc) \mid c \cdot \text{gcd}(a, b)$ . Так как  $c \mid ac$ ,  $c \mid bc$ , то  $c \mid \text{gcd}(ac, bc)$ , и  $\text{gcd}(ac, bc) = xc$

для некоторого  $x$ , и по свойству НОД выполнено  $xc \mid ac$ ,  $xc \mid bc$ . То есть  $xy = ac$ ,  $xz = bc$  для некоторых  $y, z$ . Из последних двух равенств следуют равенства  $c \cdot (xy - a) = 0 = c(xz - b)$ . Раз  $c \neq 0$ , то по свойству целостного кольца выполнены равенства  $xy - a = 0 = xz - b$ . Следовательно,  $x \mid a$ ,  $x \mid b$ ,  $x \mid \text{gcd}(a, b)$ ,  $xc \mid c \cdot \text{gcd}(a, b)$ . Подставляя сюда равенство  $\text{gcd}(ac, bc) = xc$ , выведенное раньше, получаем требуемую делимость.

### Доказательство леммы Евклида.

Даны отношения  $\text{Coprime}(a, b)$ ,  $\text{Coprime}(a, c)$ . Выведем  $\text{Coprime}(a, bc)$ .

Предположим, что  $x$  делит  $a$  и делит  $bc$ . Тогда  $x \mid \text{gcd}(ac, bc)$ . По лемме  $\text{GCD}^*$ ,  $x \mid c \cdot \text{gcd}(a, b)$ . Из условия  $\text{gcd}(a, b) \sim 1$  взаимной простоты следует  $x \mid c$ . Из этого и из условий  $x \mid a$ ,  $\text{Coprime}(a, c)$  следует, что  $x$  обратим. Это доказывает лемму.

В библиотеке доказательство леммы Евклида формализовано на языке Agda в виде функции `euclid` в модуле `OfGCD.agda`.

**Реализация алгоритма и доказательства:** описанный метод сложения дробей воплощен в библиотеке `DoCon-A 2.02 [2]` (файлы `source/Fraction/*.agda`).

Файл `source/demoTest/FractionTest.agda` содержит демонстрацию.

## 4. О ДРУГИХ РАБОТАХ ПО ДАННОМУ ПРЕДМЕТУ

Система `Axiom [12]` программирования научных вычислений известна с 1970-х годов. Ее язык `Spad` — это инструмент для обобщенного программирования, и библиотека `Axiom` реализует башню классических структур алгебры. В этой библиотеке конструктор `Fraction` поля частных принимает как аргумент целостное кольцо. И при наличии в области операции НОД применяется и арифметика дробей, основанная на сокращенном виде дроби. Но язык `Spad` не поддерживает ни зависимые типы, ни, вообще, средства доказательства программного программирования.

Система `Coq [13]` имеет большое множество библиотек, с включением машинно-проверяемых доказательств различных непростых теорем в математике. На странице сети <https://github.com/math-comp/math-comp/blob/master/mathcomp/algebra/fraction.v> дана доказательная программа для арифметики поля частных целостного кольца. Рассматривается только наивная арифметика дробей, которая проста для доказательств в теории.

Алгебраическая часть библиотеки системы `Lean [14]` содержит выражение конструкции локального кольца:

<https://github.com/leanprover-community/mathlib/>

`tree/master/src/ringt_theory`,  
 модуль `localization.lean`. Это обобщение понятия поля частных. Но не видно никакой оптимизации вычисления суммы дробей. Очевидно, в системах `Coq` и `Lean` можно реализовать такой доказательный алгоритм. Но это пока не сделано, так как эти системы направлены в первую очередь на теоретические доказательства в математике.

## 5. ЗАКЛЮЧЕНИЕ

Описанный выше опыт с доказательной программой арифметики дробей в общем случае показывает, что язык `Agda` и его реализация позволяют адекватно выразить эту алгебраическую конструкцию, вместе с необходимым основанием в виде библиотеки классических категорий алгебры, и провести необходимые полные формальные доказательства для оптимизированного способа сложения дробей.

Также библиотека `DoCon-A` имеет то значение, что она является первой общей и сравнительно обширной библиотекой вычислительной алгебры для языка `Agda`.

## БЛАГОДАРНОСТИ

Автор благодарен неизвестному рецензенту, который указал на существование обобщенной леммы Евклида.

Исследование поддержано Министерством науки и высшего образования РФ, исследовательский проект No AAAA-A19-119020690043-9.

## СПИСОК ЛИТЕРАТУРЫ

1. *Мешвелиани С.Д.* О зависимых типах и интуиционизме в программировании математики. В электронном журнале Программные системы: теория и приложения. 2014. Т. 5. Вып. 3. С. 27–50. [http://ps-ta.psiras.ru/read/psta2014\\_3\\_2\\_7-50.pdf](http://ps-ta.psiras.ru/read/psta2014_3_2_7-50.pdf)
2. *Мешвелиани С.Д.* DoCon-A. Библиотека доказательных программ компьютерной алгебры. Переславль-Залесский, 2015–2018. <http://www.botik.ru/pub/local/Mechveliani/docon-A/>
3. *Mechveliani S.D.* Computer algebra with Haskell: applying functional-ategorial-'lazy' programming. In Proceedings of International Workshop CAAP-2001, Ed. V.P. Gerdt. Dubna, Russia. С. 203–211.
4. *Norell U.* Dependently Typed Programming in Agda. AFP 2008: Advanced Functional Programming, Lecture Notes in Computer Science, vol. 5832, Springer, Berlin–Heidelberg, 2008. С. 230–266.
5. Agda. A proof assistant. A dependently typed functional programming language and its system. <http://wiki.portal.chalmers.se/agda/pmwiki.php>.
6. *Curry H.B., Feys R.* Combinatory Logic, vol I. Amsterdam, North-Holland, 1958. 417 с.
7. *Howard W.A.* The formulae-as-types notion of construction. To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism. Boston, Academic Press, 1980. С. 479–490.
8. *Марков А.А.* О конструктивной математике. Проблемы конструктивного направления в математике. 2. Конструктивный математический анализ, Сборник работ. Тр. МИАН СССР, 67. Изд-во АН СССР, М.–Л., 1962, С. 8–14.
9. *Martin-Loef, Per.* Intuitionistic type theory. Bibliopolis, ISBN 88-7088-105-9 (1984), 91 с.
10. *ван дер Варден Б.Л.* Алгебра. Наука, Москва, 1979, 624 с.
11. *Кнут Д.Е.* Искусство программирования для ЭВМ. Том 2. Мир, Москва, 1977. 55 печ. л.
12. *Jenks R.D., Sutor R.S. и др.* Axiom, the Scientific Computation System. Springer-Verlag, New York–Heidelberg–Berlin, 1992.
13. *Chlipala A.* Certified Programming with Dependent Types: A Pragmatic Introduction to the Coq Proof Assistant. MIT Press, 2013. <http://adam.chlipala.net/cpdt/>
14. *de Moura L., Kong S., Avigad J., van Doorn F., von Raumer J.* The Lean Theorem Prover. 25th International Conference on Automated Deduction (CADE-25), Berlin, Germany, 2015. <https://leanprover.github.io/papers/system.pdf>.

ПОСТРОЕНИЕ ПЕРИОДИЧЕСКОГО РЕШЕНИЯ УРАВНЕНИЙ  
ДВИЖЕНИЯ ОБОБЩЕННОЙ МАШИНЫ АТВУДА  
С ПРИМЕНЕНИЕМ КОМПЬЮТЕРНОЙ АЛГЕБРЫ

© 2020 г. А. Н. Прокопеня\*

*Варшавский университет естественных наук – SGGW,  
Польша 02-776 Варшава, ул. Новоурсыновска, Польша**\*E-mail: alexander\_prokopenya@sggw.pl*

Поступила в редакцию 13.08.2019 г.

После доработки 18.09.2019 г.

Принята к публикации 20.10.2019 г.

Обсуждается проблема поиска периодического движения обобщенной машины Атвуда, в которой шкив конечного радиуса заменяется двумя отдельными малыми шкивами и один груз может колебаться в вертикальной плоскости. Выполнив необходимые символьные вычисления, мы получили уравнения движения системы и построили их периодические решения в виде степенных рядов по малому параметру в случае малых колебаний. Показано, что при небольшой разнице в массах грузов существует состояние динамического равновесия системы, когда колеблющийся груз 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. 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/S0132347420020090

## 1. ВВЕДЕНИЕ

Классическая машина Атвуда, первоначально разработанная для демонстрации равноускоренного движения, состоит из двух грузов, соединенных невесомой нерастяжимой нитью, перекинутой через шкив, который может вращаться без трения вокруг горизонтальной оси (см. [1]). Предполагая, что грузы могут перемещаться только вдоль вертикали, а нить не проскальзывает по шкиву, получаем простую механическую систему с одной степенью свободы, уравнение движения которой легко интегрируется. Простейшее обобщение этой системы, когда один из грузов во время движения может совершать колебания в вертикальной плоскости, приводит к появлению дополнительной степени свободы, что существенно усложняет уравнения движения и их общее решение не может быть записано в символьной форме [2]. Однако динамика такой системы была исследована довольно подробно [3]. В частности, было показано, что уравнения движения являются интегрируемыми только в том случае, когда отношение масс грузов равняется 3. При этом система может демонстрировать раз-

личные виды движения, например, квазипериодическое и хаотическое движение (см. [4–6]). Радиус шкива также влияет на динамику системы (см. [7]), но его изменение не приводит к каким-либо качественным изменениям движения.

Следует отметить, что именно колебания груза существенно влияют на характер движения системы, особенно при небольшой разнице масс грузов. Исследование движения машины Атвуда с одним колеблющимся грузом показало (см. [8]), что колебания приводят к возрастанию средней силы натяжения нити, причем ее величина зависит от амплитуды колебаний. Если оба груза одинаковой массы ( $m_1 = m_2$ ) находятся в равновесии и одному из них сообщается горизонтальная начальная скорость, то этот колеблющийся груз начинает движение вниз и тянет второй груз вверх даже в случае малых колебаний. При этом длина отрезка нити  $r$  между колеблющимся грузом и шкивом увеличивается, а амплитуда его колебаний и средняя сила натяжения нити убывают. Таким образом, при наличии колебаний одного из двух одинаковых грузов равновесное решение уравнений движения машины Атвуда  $r = \text{const}$  не

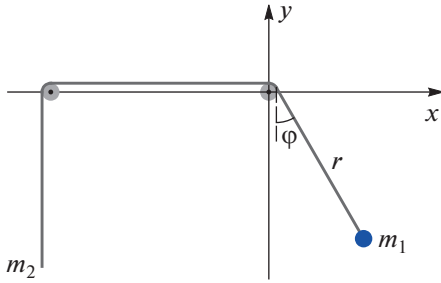


Рис. 1. Обобщенная машина Атвуда.

существует. Такое решение возможно только в том случае, когда колебания отсутствуют или же оба груза совершают колебания в противофазе с одинаковыми амплитудами и частотами (см. [9]).

С другой стороны, при небольшой разнице масс колебания груза меньшей массы приводят к возможности квазипериодического движения системы (см. [8]). Естественно предположить, что при подходящих начальных условиях система может находиться в состоянии динамического равновесия, когда груз большей массы совершает поступательное движение и колеблется около некоторого равновесного положения, а движение груза меньшей массы представляет собой наложение двух взаимно перпендикулярных колебаний. Соответствующее решение уравнений движения системы должно быть периодическим.

Целью данной работы является построение периодического решения уравнений движения машины Атвуда в виде степенного ряда по малому параметру и определение начальных условий, при которых такое движение может происходить. 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., [10–14]). В данной работе для выполнения всех расчетов и визуализации полученных результатов используется система компьютерной алгебры *Wolfram Mathematica* [15].

## 2. УРАВНЕНИЯ ДВИЖЕНИЯ

Рассматривается обобщенная модель машины Атвуда, в которой шкив конечного радиуса заменяется двумя отдельными малыми шкивами и один груз может колебаться в вертикальной плоскости (Рис. 1). Такая модификация классической машины Атвуда не изменяет ее физической природы, но позволяет исключить влияние размеров шкива и сосредоточиться на исследовании влияния колебаний на движение системы.

Функция Лагранжа рассматриваемой системы имеет вид

$$L = \frac{m_1}{2} (\dot{r}^2 + r^2 \dot{\varphi}^2) + \frac{m_2}{2} \dot{r}^2 + m_1 g r \cos \varphi - m_2 g r, \quad (2.1)$$

где точка над символом означает полную производную соответствующей функции по времени,  $g$  – ускорение свободного падения,  $r$  – расстояние между шкивом и грузом  $m_1$ , а угол  $\varphi$  определяет отклонение груза от вертикали (см. рис. 1). Выражение (2.1) записано в предположении, что радиус шкива пренебрежимо мал и изменением длины нити  $r$  при колебаниях груза за счет наматывания нити на шкив можно пренебречь.

Уравнения движения, записанные в форме Лагранжа с использованием (2.1), имеют вид

$$(m_1 + m_2) \ddot{r} = -m_2 g + m_1 g \cos \varphi + m_1 r \dot{\varphi}^2, \quad (2.2)$$

$$r \ddot{\varphi} = -g \sin \varphi - 2 \dot{r} \dot{\varphi}. \quad (2.3)$$

Легко видеть, что при  $r = R_0 = \text{const}$  уравнение (2.3) сводится к уравнению колебаний математического маятника

$$\ddot{\varphi} + \frac{g}{R_0} \sin \varphi = 0,$$

однократное интегрирование которого приводит к закону сохранения механической энергии

$$\frac{1}{2} \dot{\varphi}^2 = \frac{g}{R_0} (\cos \varphi - \cos \Phi_0), \quad (2.4)$$

где  $\Phi_0$  – амплитуда колебаний. При  $r = \text{const}$  уравнение (2.2) с учетом (2.4) принимает вид

$$3 \cos \varphi = 2 \cos \Phi_0 + \frac{m_2}{m_1}. \quad (2.5)$$

Поскольку  $\dot{\varphi}^2 \geq 0$  и, следовательно,  $|\varphi| \leq \Phi_0$ , уравнение (2.5) имеет единственное решение  $\varphi = \Phi_0 = 0$  только в случае равных масс грузов ( $m_1 = m_2$ ). Таким образом, система (2.2)–(2.3) имеет равновесное решение  $r = R_0 = \text{const}$ ,  $\varphi = 0$  только при условии  $m_1 = m_2$ .

Далее будем предполагать, что колебания груза  $m_1$  являются малыми (выполняется условие  $|\varphi| \ll 1$ ), и заменим тригонометрические функции в уравнениях (2.2), (2.3) их разложениями в степенные ряды с точностью до второго порядка включительно. Тогда уравнения движения примут вид

$$(m_1 + m_2) \ddot{r} = -m_2 g + m_1 g \left( 1 - \frac{\varphi^2}{2} \right) + m_1 r \dot{\varphi}^2, \quad (2.6)$$

$$r\ddot{\phi} = -g\phi - 2\dot{r}\dot{\phi}. \quad (2.7)$$

Заметим, что уравнения (2.6), (2.7) являются нелинейными и существование периодических движений грузов, которыми мы интересуемся в данной работе, возможно только вследствие нелинейного взаимодействия между степенями свободы системы. В линейном приближении переменные  $r(t)$  и  $\phi(t)$  разделяются и интересные нелинейные эффекты, которые проявляются даже в случае малых колебаний, теряются.

### 3. ПЕРИОДИЧЕСКОЕ РЕШЕНИЕ

Напомним, что колебания груза  $m_1$  приводят к увеличению средней силы натяжения нити и она становится больше силы тяжести  $m_1g$  (см. [8]). Следовательно, можно ожидать, что при небольшой разнице в массах грузов, когда выполняется условие  $(|m_2 - m_1|/m_1 \equiv \varepsilon \ll 1)$ , средняя сила натяжения нити может уравновесить силу тяжести  $m_2g$  и система будет находиться в состоянии динамического равновесия, при котором обе переменные  $r(t)$ ,  $\phi(t)$  осциллируют около равновесного решения и являются периодическими функциями времени. Поскольку уравнения движения (2.6)–(2.7) являются нелинейными, амплитуда и частота колебаний будут зависеть от амплитуды и, следовательно, от параметра  $\varepsilon$ . При  $\varepsilon \rightarrow 0$ , когда грузы имеют одинаковые массы, функции  $r(t)$ ,  $\phi(t)$  должны сводиться к равновесному решению  $r = R_0$ ,  $\phi = 0$ .

Для удобства вычислений произведем замену переменных

$$r(t) \rightarrow R_0(1 + \varepsilon r^*(t^*)), \quad \phi(t) \rightarrow \sqrt{\varepsilon}\phi(t^*), \quad (3.1)$$

где  $r^*$  и  $t^* = t\sqrt{g/R_0}$  – безразмерные переменные, которые далее будем обозначать обычным образом через  $r$ ,  $t$ . В результате уравнения движения (2.6)–(2.7) принимают вид, удобный для применения теории возмущений:

$$(2 + \varepsilon)\ddot{r} = -1 - \frac{1}{2}\varphi^2 + (1 + \varepsilon r)\dot{\varphi}^2, \quad (3.2)$$

$$\ddot{\phi} + \phi = -\varepsilon(r\dot{\phi} + 2\dot{r}\dot{\phi}). \quad (3.3)$$

Решение системы (3.2)–(3.3) можно искать в виде степенных рядов по малому параметру  $\varepsilon$ :

$$r(t) = r_1(t) + \varepsilon r_2(t) + \varepsilon^2 r_3(t) + \dots, \quad (3.4)$$

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

Поиск неизвестных функций  $r_k(t)$ ,  $\phi_k(t)$  в разложениях (3.4)–(3.5) требует выполнения стандартных, но достаточно громоздких символьных вычислений, которые выполняются в данной работе с помощью системы компьютерной алгебры *Wolfram Mathematica*. Для подстановки решений  $r(t)$ ,  $\phi(t)$  в уравнения (3.2)–(3.3) и автоматического вычисления всех производных удобно воспользоваться анонимными функциями (pure functions) и определить правила замены вида

$$r \rightarrow (r_1[\#] + \varepsilon r_2[\#] + \varepsilon^2 r_3[\#] \&),$$

$$\phi \rightarrow (\phi_0[\#] + \varepsilon \phi_1[\#] + \varepsilon^2 \phi_2[\#] \&);$$

Для разложения правых частей полученных уравнений в ряды по степеням параметра  $\varepsilon$  с точностью до второго порядка, например, достаточно к каждому уравнению применить функцию

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

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

Выполняя описанные символьные вычисления и приравнявая коэффициенты при одинаковых степенях параметра  $\varepsilon$  в левой и правой части каждого уравнения (3.2)–(3.3), получаем следующую систему линейных дифференциальных уравнений:

$$\ddot{\phi}_0 + \phi_0 = 0, \quad (3.6)$$

$$2\ddot{r}_1 = -1 - \frac{1}{2}\phi_0^2 + \dot{\phi}_0^2, \quad (3.7)$$

$$\ddot{\phi}_1 + \phi_1 = -r_1\ddot{\phi}_0 - 2\dot{r}_1\dot{\phi}_0, \quad (3.8)$$

$$2\ddot{r}_2 = -\ddot{r}_1 + 2\dot{\phi}_0\dot{\phi}_1 + r_1\dot{\phi}_0^2 - \phi_0\phi_1, \quad (3.9)$$

$$\ddot{\phi}_2 + \phi_2 = -r_1\ddot{\phi}_1 - r_2\ddot{\phi}_0 - 2\dot{r}_1\dot{\phi}_1 - 2\dot{r}_2\dot{\phi}_0, \quad (3.10)$$

$$2\ddot{r}_3 = -\ddot{r}_2 + 2\dot{\phi}_0\dot{\phi}_2 + \dot{\phi}_1^2 + 2r_1\dot{\phi}_0\dot{\phi}_1 + r_2\dot{\phi}_0^2 - \frac{1}{2}\phi_1^2 - \phi_0\phi_2, \dots \quad (3.11)$$

Отметим, что линейные дифференциальные уравнения второго порядка (3.6)–(3.11) последовательно решаются с помощью встроенной функции *DSolve*. Без ограничения общности рассуждений можем предположить, что в начальный момент времени груз  $m_1$  находится в отклоненном от вертикали положении и начинает движение без начальной скорости. Тогда решение уравнения (3.6) при начальном условии  $\dot{\phi}_0(0) = 0$  можно записать в виде

$$\varphi_0 = C_0 \cos t, \quad (3.12)$$

где  $C_0$  – неизвестная постоянная, равная амплитуде колебаний в нулевом приближении по  $\varepsilon$ , которая находится из условия, что в правой части уравнения (3.7) присутствуют только осциллирующие функции. Подставляя решение (3.12) в (3.7) и выполняя несложные символьные преобразования, получаем:

$$2\ddot{r}_1 = -1 + \frac{1}{4}C_0^2 - \frac{3}{4}C_0^2 \cos(2t). \quad (3.13)$$

Отметим, что наличие постоянного слагаемого в правой части уравнения (3.13) приводит к возрастанию или убыванию функции  $r_1(t)$ . Поскольку нас интересует периодическое решение, не зависящее от времени слагаемое следует приравнять к нулю. В результате находим:

$$C_0 = 2. \quad (3.14)$$

Поскольку груз  $m_1$  начинает движение без начальной скорости, а его начальное положение не известно, дифференциальное уравнение второго порядка (3.13) интегрируем с одним начальным условием  $\dot{r}_1(0) = 0$  и с учетом (3.14) получаем:

$$r_1 = B_1 + \frac{3}{8} \cos(2t). \quad (3.15)$$

В решении (3.15) появляется новая неизвестная постоянная  $B_1$ , которая находится из условия, что в правой части уравнения (3.8) отсутствуют резонансные члены, пропорциональные  $\cos t$  и  $\sin t$ . Отметим, что появление резонансных членов в правой части (3.8) привело бы к неограниченному возрастанию функции  $\varphi_1(t)$ , что невозможно в замкнутой консервативной системе (см., напр., [16]).

Подставляя решения (3.12) и (3.15) в уравнение (3.8), получаем:

$$\ddot{\varphi}_1 + \varphi_1 = \left(2B_1 - \frac{9}{8}\right) \cos t + \frac{15}{8} \cos(3t). \quad (3.16)$$

Условие отсутствия резонансов в (3.16) дает

$$B_1 = \frac{9}{16}. \quad (3.17)$$

Интегрируя (3.16) с учетом (3.17) при начальном условии  $\dot{\varphi}_1(0) = 0$ , находим:

$$\varphi_1(t) = \left(C_1 - \frac{15}{32}\right) \cos t - \frac{15}{64} \cos(3t), \quad (3.18)$$

где  $C_1$  – неизвестная постоянная, которая опять находится из условия, что в правой части уравнения (3.9) присутствуют только осциллирующие функции.

Подставляя решения (3.12), (3.15) и (3.18) в уравнение (3.9) и учитывая (3.14), (3.17), получаем:

$$2\ddot{r}_2 = C_1 + \frac{9}{32} + \left(\frac{87}{64} - 3C_1\right) \cos(2t) + \frac{81}{64} \cos(4t). \quad (3.19)$$

Из уравнения (3.19) находим

$$C_1 = -\frac{9}{32}. \quad (3.20)$$

Интегрирование (3.19) с учетом (3.20) при начальном условии  $\dot{r}_2(0) = 0$  дает:

$$r_2 = B_2 - \frac{141}{512} \cos(2t) - \frac{81}{2048} \cos(4t), \quad (3.21)$$

где  $B_2$  – неизвестная постоянная, которая находится из условия, что в правой части уравнения (3.10) отсутствуют резонансные члены.

Подставляя решения (3.12), (3.15), (3.18) и (3.21) в уравнение (3.10) и учитывая (3.14), (3.17), (3.20), получаем:

$$\begin{aligned} \ddot{\varphi}_2 + \varphi_2 = & \left(2B_2 + \frac{981}{1024}\right) \cos t - \\ & - \frac{6123}{2048} \cos(3t) - \frac{2619}{2048} \cos(5t). \end{aligned} \quad (3.22)$$

Условие отсутствия резонансов в (3.22) дает

$$B_2 = -\frac{981}{2048}. \quad (3.23)$$

Интегрируя (3.22) с учетом (3.23) при начальном условии  $\dot{\varphi}_2(0) = 0$ , находим:

$$\begin{aligned} \varphi_2(t) = & \left(C_2 + \frac{6123}{8192}\right) \cos t + \\ & + \frac{6123}{16384} \cos(3t) + \frac{873}{16384} \cos(5t), \end{aligned} \quad (3.24)$$

где  $C_2$  – неизвестная постоянная, которая опять находится из условия, что в правой части уравнения (3.11) присутствуют только осциллирующие функции.

Подставляя решения (3.12), (3.15), (3.18), (3.21) и (3.24) в уравнение (3.11) и учитывая (3.14), (3.17), (3.20), (3.23), получаем:

$$\begin{aligned} 2\ddot{r}_3 = & C_2 - \frac{6345}{16384} - \left(3C_2 + \frac{20595}{16384}\right) \cos(2t) - \\ & - \frac{9747}{4096} \cos(4t) - \frac{4455}{8192} \cos(6t). \end{aligned} \quad (3.25)$$

Из уравнения (3.25) находим

$$C_2 = \frac{6345}{16384}. \quad (3.26)$$



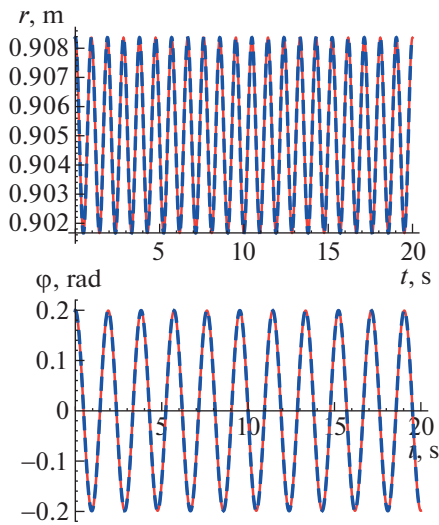


Рис. 2. Сравнение аналитического и численного решений ( $\varepsilon = 0.01$ ).

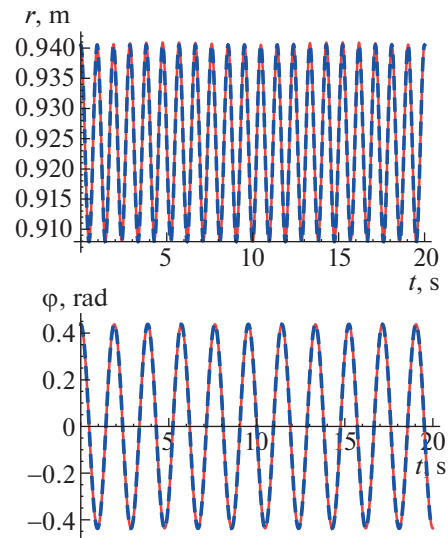


Рис. 3. Сравнение аналитического и численного решений ( $\varepsilon = 0.05$ ).

Интегрирование (3.25) с учетом (3.26) при начальном условии  $\dot{r}_3(0) = 0$  дает:

$$r_3 = B_3 + \frac{19815}{65536} \cos(2t) + \frac{9747}{131072} \cos(4t) + \frac{495}{65536} \cos(6t), \quad (3.27)$$

где  $B_2$  – неизвестная постоянная, которая находится из условия, что в правой части уравнения для  $\varphi_3(t)$  отсутствуют резонансные члены.

В результате с точностью до второго порядка по  $\varepsilon$  получаем решение уравнений движения (3.2)–(3.3) в виде

$$r(t) = \frac{1}{16} (9 + 6 \cos(2t)) - \frac{\varepsilon}{2048} (981 + 564 \cos(2t) + 81 \cos(4t)) + \frac{\varepsilon^2}{131072} (80883 + 39630 \cos(2t) + 9747 \cos(4t) + 990 \cos(6t)), \quad (3.28)$$

$$\varphi(t) = 2 \cos t - \frac{\varepsilon}{64} (48 \cos t + 15 \cos(3t)) + \frac{\varepsilon^2}{16384} (18591 \cos t + 6123 \cos(3t) + 873 \cos(5t)). \quad (3.29)$$

Легко видеть, что полученное решение является периодическим, причем частота колебаний длины маятника  $r(t)$  равняется удвоенной частоте

колебаний угловой переменной  $\varphi(t)$ , т.е. наблюдается резонанс частот  $2 : 1$ .

Напомним, что переменные  $r(t)$  и  $\varphi(t)$  в (3.28), (3.29) являются безразмерными (см. (3.1)). Используя выражения (3.28), (3.29), определяем начальные условия в исходных переменных:

$$r(0) = R_0 \left( 1 + \frac{15}{16} \varepsilon - \frac{813}{1024} \varepsilon^2 + \frac{65625}{65536} \varepsilon^3 \right), \\ \varphi(0) = \sqrt{\varepsilon} \left( 2 - \frac{63}{64} \varepsilon + \frac{25587}{16384} \varepsilon^2 \right), \quad (3.30) \\ \dot{r}(0) = \dot{\varphi}(0) = 0.$$

Отметим, что амплитуда колебаний маятника, определяемая выражением (3.30) и равная  $\varphi(0)$ , зависит от разницы масс грузов. При заданной частоте колебаний, определяемой параметром  $R_0$  и равной  $\sqrt{g/R_0}$ , начальная длина маятника  $r(0)$  также зависит от  $\varepsilon$  (см. (3.30)).

Для демонстрации корректности найденного аналитического решения (3.28), (3.29) выберем некоторые реалистичные значения параметров, например,  $R_0 = 0.9$  м,  $\varepsilon = 0.01$  и  $g = 9.8$  м/с<sup>2</sup> и найдем численное решение системы (3.2)–(3.3), используя встроенную функцию *NSolve* и начальные условия (3.30). Найденному численному решению соответствуют на рис. 2 сплошные линии, а штриховые линии изображают функции (3.28), (3.29). Как видим, обе кривые совпадают. Отметим также, что увеличение разницы масс грузов ( $\varepsilon = 0.05$ ) приводит к изменению начальных значений  $r(0)$  и  $\varphi(0)$  (см. (3.30)), что означает возрастание

тание амплитуды колебаний и изменение равновесного значения длины маятника  $r(t)$  (рис. 3). Очевидно, при дальнейшем возрастании параметра  $\varepsilon$  амплитуда периодических колебаний системы также должна возрасти, что приведет к нарушению условия малости колебаний и необходимости учета членов более высоких порядков в разложениях тригонометрических функций в уравнениях движения (2.2)–(2.3).

Если предположить, что в начальный момент времени грузу  $m_1$ , который покоится в положении  $\varphi(0) = 0$ , сообщают горизонтальную начальную скорость, то уравнение (3.6) следует решать при соответствующем начальном условии, что дает

$$\varphi_0 = C_0 \sin t, \quad (3.31)$$

где  $C_0$  – неизвестная постоянная. Повторяя вычисления с функцией  $\varphi_0(t)$ , определяемой выражением (3.31), получаем новое решение системы (3.2)–(3.3):

$$\begin{aligned} r(t) = & \frac{1}{16}(9 - 6 \cos(2t)) - \\ & - \frac{\varepsilon}{2048}(981 - 564 \cos(2t) + 81 \cos(4t)) + \\ & + \frac{\varepsilon^2}{131072}(80883 - 39630 \cos(2t) + \\ & + 9747 \cos(4t) - 990 \cos(6t)), \end{aligned} \quad (3.32)$$

$$\begin{aligned} \varphi(t) = & 2 \sin t - \frac{\varepsilon}{64}(48 \sin t - 15 \sin(3t)) + \\ & + \frac{\varepsilon^2}{16384}(18591 \sin t - 6123 \sin(3t) + \\ & + 873 \sin(5t)). \end{aligned} \quad (3.33)$$

Заметим, что при замене  $t \rightarrow t + \pi/2$  решение (3.32), (3.33) переходит в полученное ранее решение (3.28), (3.29). Аналогичный результат получается при выборе решения уравнения (3.6) в общем виде

$$\varphi_0 = C_0 \cos t + D_0 \sin t, \quad (3.34)$$

где  $C_0, D_0$  – неизвестные постоянные, и повторения всех вычислений. При этом следует следить за тем, чтобы в дифференциальных уравнениях (3.8), (3.10) для коэффициентов  $\varphi_k(t)$  не появились резонансные члены, пропорциональные  $\cos t$  и  $\sin t$ , а в правых частях уравнений (3.7), (3.9), (3.11) сохранились только осциллирующие члены.

Таким образом, при заданных параметрах  $\varepsilon$  и  $R_0$  существует единственное периодическое решение уравнений движения (3.2), (3.3), представимое в

виде степенных рядов (3.4), (3.5). Выполняя вычисления в высших порядках по  $\varepsilon$ , можно найти искомое периодическое решение с необходимой точностью, хотя вычисления становятся все более громоздкими и для их выполнения требуется применение систем компьютерной алгебры.

#### 4. ЗАКЛЮЧЕНИЕ

В настоящей работе обсуждается проблема построения периодического решения уравнений движения обобщенной машины Атвуда. Показано, что при небольшой разнице масс двух грузов и малых колебаниях груза меньшей массы уравнения движения имеют единственное периодическое решение, представимое в виде степенных рядов по малому параметру  $\varepsilon$ . Последовательно описаны символьные вычисления, необходимые для определения коэффициентов этих рядов, и найдено соответствующее решение с точностью до второго порядка по  $\varepsilon$ . Сравнение найденного аналитического решения с численным решением уравнений движения показало справедливость полученных теоретических результатов.

Отметим, что хотя найденные решения (3.28), (3.29) не выглядят слишком громоздкими, их получение с точностью до третьего и более высоких порядков по  $\varepsilon$  связаны с большим объемом символьных вычислений, которые можно эффективно выполнять с помощью систем компьютерной алгебры. 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. *Tufillaro 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. *Tufillaro N.B.* Motions of a swinging Atwood's machine // *J. Physique*. 1985. V. 46. P. 1495–1500.
4. *Tufillaro 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., Tufillaro 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 (2.4). P. 711–716.
7. *Pujol O., Pérez J.P., Ramis J.P., Simo C., Simon S., Weil J.A.* Swinging Atwood machine: Experimental and nu-

- merical results, and a theoretical study // *Physica D*. 2010. V. 239 (3.3). 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. *Prokopenya A.N.* Modelling Atwood's Machine with Three Degrees of Freedom // *Mathematics in Computer Science*. 2019. V. 13 (1–2). P. 247–257.
  10. *Абрамов С.А., Зима Е.Б., Ростовцев В.А.* Компьютерная алгебра // *Программирование*. 1992. № 5. С. 4–25.
  11. *Васильев Н.Н., Еднерал В.Ф.* Компьютерная алгебра в физических и математических приложениях // *Программирование*. 1994. № 1. С. 70–82.
  12. *Прокопеня А.Н.* Некоторые алгоритмы символьных вычислений в исследованиях проблем космической динамики // *Программирование*. 2006. Т. 32 (2.2). С. 16–22.
  13. *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.
  14. *Прокопеня А.Н.* Нормализация гамильтониана в ограниченной задаче многих тел методами компьютерной алгебры // *Программирование*. 2012. Т. 38 (2.3). С. 65–78.
  15. *Wolfram S.* An elementary introduction to the Wolfram Language, 2nd ed. – Champaign, IL, USA, Wolfram Media, 2017.
  16. *Маркеев А.П.* Теоретическая механика. Ижевск: НИЦ “Регулярная и хаотическая динамика”, 2001. 592 с.

## ДИФФЕРЕНЦИАЛЬНАЯ ГЕОМЕТРИЯ И МЕХАНИКА – ИСТОЧНИК ЗАДАЧ ДЛЯ КОМПЬЮТЕРНОЙ АЛГЕБРЫ

© 2020 г. В. Н. Сальников<sup>a,\*</sup>, А. Хамдуни<sup>b,\*\*</sup>

<sup>a</sup> ЦНРС и Университет города Ла Рошель  
17042 Ла Рошель, Проспект Мишеля Крепо, Франция

<sup>b</sup> Университет города Ла Рошель  
Ла Рошель, Проспект Мишеля Крепо, Франция

\*E-mail: vladimir.salnikov@univ-lr.fr,

\*\*E-mail: aziz.hamdouni@univ-lr.fr

Поступила в редакцию 04.08.2019 г.

После доработки 23.09.2019 г.

Принята к публикации 11.11.2019 г.

В этой статье мы обсуждаем возможности компьютерной алгебры в процессе моделирования и изучения качественных свойств механических систем и задач теоретической физики.

Мы описываем структуры из так называемой обобщенной геометрии: алгеброиды Куранта и структуры Дирака. Они являются удобным языком для изучения структуры дифференциальных уравнений порт-Гамильтоновых и неявных Лагранжевых систем, описывающих, соответственно, взаимодействующие механические системы с диссипацией и системы со связями. Для обоих классов систем мы формулируем нерешенные задачи, в решении которых могут быть использованы методы компьютерной алгебры.

Мы также напоминаем определения из градуированной геометрии: градуированных многообразий и  $Q$ -структур. На конкретных примерах мы объясняем, как классическая дифференциальная геометрия описывается в рамках градуированной и какие вычислительные вопросы могут возникать. Это направление, по-видимому, является очень малоисследованным в контексте компьютерной алгебры.

DOI: 10.31857/S0132347420020107

### 1. ВВЕДЕНИЕ/МОТИВАЦИЯ

Эта статья – часть большого проекта по “геометризации механики”, включающего в себя серию работ, цель которых описать формализм (связанный с дифференциальной или алгебраической геометрией), удобный для качественного анализа и моделирования достаточно обширного класса механических систем. В этом проекте мы исследуем весь спектр вопросов от описания математической модели изучаемой системы и определения соответствующих геометрических структур до подбора (или создания) численных методов, сохраняющих построенные структуры, а также эффективной компьютерной реализации и применения методов. Такие численные методы называются *геометрическими интеграторами*. Недавние публикации показывают преимущества такого подхода: рассмотрение законов сохранения, симметрий системы, ограничений, наложенных на нее, улучшает точность численных методов и, как следствие, надежность результатов

моделирования, а также описания качественных свойств системы.

В данной статье мы остановимся на некоторых задачах, возникающих в этом процессе, которые трудно или невозможно решить классическими аналитическими методами. Как нам кажется, методы компьютерной алгебры могут быть с успехом использованы в этом контексте. Некоторые из возникающих задач являются чисто техническими, и компьютерная алгебра просто ускоряет расчеты, но большинство достаточно концептуальные, где актуальны вопросы существования решений и оптимизации алгоритмов.

Подчеркнем, что речь идет именно об *открытых вопросах*. В них понятно, что нужно сделать, и примерно понятно, как, то есть можно выделить некоторые перспективные направления. Мы считаем, что к решению этих задач нет концептуальных препятствий, поскольку для конкретных примеров есть понятные конструктивные ответы. Отсутствие же решений в общем случае объясняется отсутствием регулярных контактов между

специалистами по компьютерной алгебре и прикладниками из других областей: физиками, механиками и особенно геометрами. В данной статье мы постараемся ликвидировать этот пробел, наиболее доступно описав геометрические конструкции, минимально необходимые для понимания возникающих задач и их непосредственного решения.

### 1.1. Организация статьи

В следующей части (раздел 3) мы определяем структуры из обобщенной геометрии, возникающие в приложениях к механике и теоретической физике. Затем в разделах 7 и 8 мы описываем механические системы, для которых такая геометрия является удобным “языком”, и объясняем, какие задачи могут быть решены методами компьютерной алгебры. Раздел 9 посвящен градуированной геометрии: мы определим необходимые математические понятия и снова опишем важные задачи компьютерной алгебры, насколько нам известно, не решенные в стандартных пакетах.

## 2. ОБОБЩЕННАЯ ГЕОМЕТРИЯ

В этом разделе мы коротко определим упомянутые во введении геометрические конструкции, а именно, алгеброид Куранта и структуры Дирака. По сравнению с оригинальной статьей [1], мы упростим изложение, опустив некоторые технические детали, не принципиальные для интересующих нас результатов, однако данного раздела должно быть достаточно, чтобы понять и сформулировать задачи компьютерной алгебры. В частности, мы будем писать слова “многообразие” и “расслоение”, не определяя их; для понимания можно читать “векторное пространство соответствующей размерности”.

### 2.1. Векторные поля

Итак, пусть  $M$  — многообразие размерности  $n$ . Обозначим  $TM$  — касательное расслоение к нему (множество всех касательных векторов в каждой точке) и  $T^*M$  — кокасательное расслоение. Локально  $TM$  можно рассматривать как

$$TM \simeq \mathbb{R}^{2n} = \mathbb{R}^n \times V,$$

где  $V$  — векторное пространство той же размерности, что и  $M$ , тогда

$$T^*M \simeq \mathbb{R}^{2n} = \mathbb{R}^n \times V^*$$

Множество сечений  $TM$  обозначается  $\Gamma(TM)$  — это векторные поля на  $M$ . На них определен коммутатор

$$[\cdot, \cdot]: \Gamma(TM) \times \Gamma(TM) \rightarrow \Gamma(TM).$$

В компонентах он имеет вид:

$$[v, w]^i = \sum_j (v^j \partial_j w^i - w^j \partial_j v^i),$$

где  $\partial_j$  — производная по  $j$ -й координате.

### 2.2. Дифференциальные формы

На  $V$  можно определить кососимметричные  $k$ -линейные формы. Если это сделать в каждой точке  $M$ , с некоторыми условиями на регулярность, получится объект, называемый *дифференциальная  $k$ -форма*, то есть, по сути, кососимметричная “функция”, “аргументами” которой являются  $k$  векторных полей на  $M$ .

Функции на  $M$  можно рассматривать как 0-формы. Стандартные примеры 1-форм, то есть ковекторных полей — это объекты, двойственные векторным полям  $\partial_i$ , их обозначают  $dx^i$ . Двойственность понимается в смысле  $dx^i(\partial_j) = \delta_j^i$ , где  $\delta_j^i$  — символ Кронекера. В размерности 2 примером 2-формы будет ориентированная площадь.

На дифференциальных формах определены две естественные операции:

- *Свертка* формы  $\alpha(\cdot, \cdot, \dots)$  с векторным полем  $v$ :

$$\iota_v(\alpha) := \alpha(v, \cdot, \dots).$$

Она понижает степень формы.

- *Внешний дифференциал* формы  $\alpha$ :

$$d\alpha(v_0, v_1, \dots, v_k) := \sum_i (-1)^i v_i \alpha(v_0, \dots, v_i, \dots, v_k) + \sum_{i < j} (-1)^{i+j} \alpha([v_i, v_j], v_0, \dots, \hat{v}_i, \dots, \hat{v}_j, \dots, v_k),$$

где  $\hat{v}$  обозначает пропуск аргумента.

Внешний дифференциал повышает степень формы.

Обозначение  $dx^i$  выше не случайно: двойственные к  $\partial_i$  объекты, действительно являются дифференциалами координат. Если  $d\alpha = 0$ , то форма  $\alpha$  называется *замкнутой*. Если форма  $\alpha$  сама является дифференциалом другой формы:  $\alpha = d\beta$ , то форма  $\alpha$  называется *точной*, а форма  $\beta$  иногда называется ее *интегралом*.

Если 2-форма невырождена (в каждой точке, в смысле линейной алгебры), она называется *почти симплектической*, если она к тому же замкнута, то *симплектической*. Векторное поле  $v$ , свертка которого с симплектической формой  $\omega$  является точной формой с интегралом  $H$  ( $\iota_v \omega = dH$ ), называется *гамильтоновым векторным полем с гамильтонианом  $H$* .

Заметим, что подобная конструкция возможна также на кокасательном расслоении — результатом будут поливекторные поля. А аналогом сим-

плектической формы будет Пуассонов бивектор, который мы упомянем в разделе 5.

### 2.3. Алгеброид куранта и структуры Дирака

Рассмотрим теперь расслоение  $E = TM \oplus T^*M$ , иногда его называют обобщенным касательным расслоением или расслоением Понтрягина. Его сечения — это пары  $(v, \alpha)$ , где  $v$  — векторное поле,  $\alpha$  — 1-форма, то есть ковекторное поле. На парах сечений определяются две операции:

Аналог скалярного произведения (в каждой точке, со значениями в  $\mathbb{R}$ ):

$$\langle (v, \alpha), (w, \beta) \rangle := \iota_v \beta + \iota_w \alpha.$$

Аналог векторного произведения:

$$[(v, \alpha), (w, \beta)]_{CD} := ([v, w], (d\iota_v + \iota_v d)\beta - \iota_w d\alpha).$$

Вторая операция называется *скобкой Куранта–Дорфман*, а вся конструкция — пример (достаточно общий) *алгеброида Куранта*.

*Структура Дирака* это подрасслоение  $D \subset E$ , которое, во-первых, максимально изотропно (то есть  $\langle \cdot, \cdot \rangle|_D \equiv 0$ , и его ранг максимален и равен размерности  $M$ ), во-вторых, устойчиво под действием  $[\cdot, \cdot]_{CD}$ :

$$[\Gamma(D), \Gamma(D)]_{CD} \subset \Gamma(D),$$

где  $\Gamma(D)$  обозначает множество сечений  $D$ . Если ослабить второе условие, получится *почти структура Дирака*.

Тривиальным примером структуры Дирака будет  $D = TM \subset E$ . Более интересный задается графиком 2-формы  $\omega$ , т.е. множеством пар

$$(v, \alpha), \quad \text{где} \quad \alpha = \iota_v \omega.$$

Условие на скалярное произведение гарантируется кососимметричностью формы. Условие на  $[\cdot, \cdot]_{CD}$  выполнено, когда форма замкнута. Таким образом, симплектическая структура — частный случай структуры Дирака.

Другие примеры включают Пуассоновы структуры — мы обсудим их в разделе (9), а общее описание можно найти в оригинальной статье ([1]).

Структуры Дирака были введены Т. Курантом с некоторой мотивацией из механики: систему можно рассматривать в терминах координат и скоростей или координат и импульсов, геометрическое описание тогда происходит на  $TM$  или  $T^*M$  соответственно. В некотором смысле, структуры Дирака могли бы обеспечить связь этих двух способов описания. Однако в дальнейших работах Т. Куранта изучались лишь их геометрические свойства. Приложения к механике возникли позже в работах о неявных Лагранжевых ([2–5]) и порт-Гамильтоновых системах ([6, 7]) — о них мы расскажем в двух следующих разделах.

### 3. НЕЯВНЫЙ ЛАГРАНЖЕВ ФОРМАЛИЗМ И СИСТЕМЫ СО СВЯЗЯМИ

Одна из причин, почему структуры Дирака не сразу нашли свое применение в механике, заключается в том, что просто рассматривать  $TM$  и  $T^*M$  как фазовое пространство системы не совсем верно. Оказывается, правильная конструкция подразумевает построение структур Дирака на многообразии, которое уже является расслоением:  $M = T^*Q$ , где  $Q$  — конфигурационное пространство системы. Тогда необходимый формализм использует двойные расслоения и некоторые их фундаментальные свойства, описанные в [8].

Такая конструкция позволяет пересмотреть классический формализм систем со связями. Рассмотрим с теми же оговорками, что и в предыдущем разделе,  $TQ \simeq \mathbb{R}^{2d} = \mathbb{R}^d \times V$  и  $T^*Q \simeq \mathbb{R}^{2d} = \mathbb{R}^d \times V^*$ , где  $d$  — число степеней свободы системы.  $V$  представляет собой пространство скоростей  $\mathbf{v}$  в каждой точке  $\mathbf{q}$  конфигурационного пространства, а  $V^*$ , соответственно, пространство импульсов  $\mathbf{p}$ . Наложить связи на данную систему значит ограничить разрешенные точки в конфигурационном пространстве и в каждой разрешенной точке выбрать подпространство допустимых скоростей — обозначим это множество  $\Delta$ . Такая геометрическая конструкция называется *распределением*, при некоторых условиях невырожденности, можно считать, что оно задается как ядро некоторого количества 1-форм.

Тогда, следуя построению из предыдущего раздела, рассмотрим  $\mathcal{V} = TT^*Q \simeq \mathbb{R}^{4d}$  и  $\mathcal{V}^* = T^*T^*Q \simeq \mathbb{R}^{4d}$ . Распределение  $\Delta$  и линейная оболочка 1-форм поднимаются на это двойное расслоение — обозначим результат поднятия  $\tilde{\Delta}$  и  $\Delta_0$  соответственно. Заметим, что на  $T^*Q$  существует каноническая симплектическая форма  $\Omega$ , которая определяет отображение  $\Omega^b \mathcal{V} \rightarrow \mathcal{V}^*$ , как в примере в предыдущем разделе. Тогда (почти) структура Дирака для системы со связями задается следующим условием:

$$\mathbb{D}_\Delta = \{(w, \beta) \in \mathcal{V} \times \mathcal{V}^* \mid w \in \tilde{\Delta}, \beta - \Omega^b w \in \Delta_0\}.$$

Динамика системы, в свою очередь, задается Лагранжианом  $L : TQ \rightarrow \mathbb{R}$ , дифференциал которого можно поднять до  $\mathcal{V}^*$ , и векторным полем — сечением  $\mathcal{V}$ . Такая динамика сохраняет связи, если эта пара принадлежит  $\mathbb{D}_\Delta$ .

Эта конструкция, конечно, не обязательна для вывода уравнений движения системы: связи можно изучать, например, с помощью множителей Лагранжа. Однако у геометрического подхода есть важное преимущество: все встречающиеся объекты и операции имеют естественные дискретные аналоги. Это позволяет построить численный метод для решения полученных уравнений движе-

ния, который будет сохранять связи лучше, чем классические методы.

Более подробно эта конструкция описана в [4, 5]. Опустив технические детали, приведем здесь только конечные уравнения, задающие  $\mathbb{D}_\Delta$ :

$$\begin{aligned} \iota_v \alpha^a &= 0, \quad a = 1, \dots, m, \\ \dot{\mathbf{q}} &\in \Delta, \quad \mathbf{p} = \frac{\partial L}{\partial \mathbf{v}}, \\ \dot{\mathbf{q}} &= \mathbf{v}, \quad \dot{\mathbf{p}} - \frac{\partial L}{\partial \mathbf{q}} \in \Delta_0 \end{aligned} \quad (3.1)$$

Первая строчка здесь определяет связи:  $\alpha^a$  – 1-формы, задающие распределение  $\Delta$ . Дискретная версия этих уравнений имеет вид:

$$\begin{aligned} \langle \alpha_d^a, v_k \rangle &= 0, \quad a = 1, \dots, m, \\ p_{k+1} &= \frac{1}{h} \frac{\partial L_d}{\partial v_k} \\ p_k - \frac{1}{h} \frac{\partial L_d}{\partial v_k} + \frac{\partial L_d}{\partial q_k} &= \sum_{a=1}^m \lambda^a \frac{\partial \langle \alpha_d^a, v_k \rangle}{\partial v_k} \end{aligned} \quad (3.2)$$

$L_d := hL(q_k, v_k)$  – дискретный Лагранжиан системы, индексы  $k$  или  $k+1$  обозначают момент времени, в который вычисляется соответствующая переменная. В уравнения явно входят переменные  $p$  на  $k$ -м и  $(k+1)$ -м шаге, а  $v_k$  задается некоторым приближением скорости  $v$  с участием  $q_k$ . Таким образом, легко проверить, что система определена и полна, кроме того она линейна по всем переменным, кроме, возможно  $v$ .

С точки зрения компьютерной алгебры, здесь естественно возникают два вопроса. Первый чисто технический: вывод уравнений (3.1) и их дискретизация. Второй гораздо более концептуальный: дискретные уравнения (3.2) нужно решать, причем делать это на каждом шаге интегрирования, то есть возникает вопрос эффективности. Отсюда первая задача, которую мы хотим сформулировать в данной статье.

**Задача 1.** Для уравнений вида (3.1) и (3.2) определить, когда полученный численный метод является явным, а когда нет. Какие геометрические (например, вид связей) и численные (например, приближение  $v$ ) факторы на это влияют? Для каждого класса предложить алгоритм точного (символьного) или приближенного (итеративного) решения системы (3.2).

Ответ на первый вопрос, по-видимому, довольно естественный: он зависит от того, насколько нелинейными соотношениями связаны известные и искомые переменные, а это классические вопросы компьютерной алгебры. В зависимости от этого ответа и производится выбор нужного алгоритма решения. При этом геометрическая и механическая интерпретация ответа может быть достаточно интересной.

#### 4. ПОРТ-ГАМИЛЬТОНОВ ФОРМАЛИЗМ И ВЗАИМОДЕЙСТВУЮЩИЕ СИСТЕМЫ

В разделе 2 мы определили симплектическую структуру и Гамильтоновы векторные поля. Это удобный формализм для описания консервативных механических систем, не взаимодействующих с окружением. В обозначениях предыдущих разделов, на кокасательном расслоении  $T^*Q$  с канонической симплектической формой такие системы имеют вид:

$$\dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}}, \quad \dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{q}}. \quad (4.1)$$

Можно проверить, что поток Гамильтонова векторного поля  $v$  сохраняет функцию  $H$ , что, с точки зрения механики, соответствует сохранению энергии. Пожалуй, первые геометрические интеграторы появились в этом контексте ([9, 10]) – они сохраняют симплектическую структуру и объем в фазовом пространстве и, как следствие, позволяют контролировать сохранение энергии.

Однако большинство механических систем, рассматриваемых в приложениях, взаимодействуют друг с другом и не являются консервативными. Для таких ситуаций была предложена модификация формализма, связанная с так называемыми *порт-Гамильтоновыми системами* ([6, 7]) – изучаются уравнения вида (4.1), в которые добавляются слагаемые, отвечающие за взаимодействие и диссипацию:

$$\dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}}, \quad \dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{q}} + \mathbf{F}.$$

Динамика тогда полностью определяется функцией  $H = H(\mathbf{q}, \mathbf{p})$  и силой  $\mathbf{F}$ . Или в более общем случае (для неканонических симплектических форм) рассмотрим систему уравнений

$$\dot{\mathbf{x}} = (J(\mathbf{x}) - R(\mathbf{x})) \frac{\partial H}{\partial \mathbf{x}} + g(\mathbf{x}) \mathbf{f}, \quad (4.2)$$

где  $J(\mathbf{x})$  – кососимметрическая матрица, а  $R(\mathbf{x})$  и  $\mathbf{f}$  новые слагаемые, отвечающие за взаимодействие и диссипацию, – они отсутствуют в классических Гамильтоновых системах. В работах [6, 7] приведена классификация этих слагаемых и описан их физический смысл.

Важно понимать, что порт-Гамильтонов формализм сам по себе не дает новой информации о системе. Однако он позволяет “упорядочить” систему, то есть разбить ее на простые блоки, взаимодействующие по понятным законам – это может быть полезно для моделирования (см. например, [11]). Кроме того, оказывается ([7]) порт-Гамильтоновы системы можно описывать с помощью структур Дирака, а это очень важно в контексте геометрических интеграторов. Таким образом, мы подходим ко второй задаче этой статьи.

**Задача 2.** По заданной системе дифференциальных уравнений построить ее оптимальное порт-Гамильтоново представление. Описать соответствующую структуру Дирака.

Обратим внимание на слово “оптимальное” в задаче. Нетрудно убедиться, что любую систему дифференциальных уравнений можно переписать в порт-Гамильтоновой форме: достаточно “забыть” Гамильтонову структуру и объявить все правые части “портами”. Но тогда потеряется информация об энергетическом балансе системы, и она потенциально увеличится втрое. С точки зрения приложений, такое решение задачи не имеет большого смысла. Оптимальность можно понимать, например, как ограничение количества добавленных переменных. Тогда возникают стандартные вопросы существования решения и быстрых тестов, исключающих нерешаемые случаи.

Построение структуры Дирака, в свою очередь, нужно для последующего использования в контексте геометрических интеграторов. Здесь необходимо понять, в какой форме ответ будет наиболее конструктивен, а именно, как описать структуру Дирака в виде, удобном для дискретизации. В этом контексте возможна также некоторая связь с вопросами из следующего раздела (9).

## 5. ГРАДУИРОВАННАЯ ГЕОМЕТРИЯ

В этом разделе мы опишем геометрическую конструкцию, еще более общую, чем в разделе 3: речь идет о *градуированных многообразиях*. По-прежнему, вместо многообразий можно думать о векторных пространствах, хотя в градуированном случае могут возникать более интересные последствия такого упрощения.

Будем считать, что на многообразии  $M$  определена *градуировка*, то есть для каждой координаты  $x^i$  на  $M$  определена величина  $\deg(x^i) \in \mathbb{Z}$ , называемая *степенью*. Для простоты будем считать, что  $\deg(x^i)$  неотрицательна для всех координат.

Эта градуировка используется для определения коммутационных соотношений между координатами. В отличие от классического случая,

$$x^i \cdot x^j = (-1)^{\deg(x^i)\deg(x^j)} x^j \cdot x^i.$$

Кроме того, для всех координат выполнено соотношение

$$\deg(x^i \cdot x^j) = \deg(x^i) + \deg(x^j).$$

Наиболее общие функции на таком многообразии представляют собой формальные ряды по всем переменным, однако в интересных случаях достаточно ограничиться многочленами от координат с ненулевыми степенями, коэффициенты которых суть гладкие функции от координат с нулевой степенью. Для них корректно определено

понятие однородной функции и ее степени, а значит и коммутационные соотношения:

$$f \cdot g = (-1)^{\deg(f)\deg(g)} g \cdot f.$$

На градуированных многообразиях определены многие объекты и операции классической дифференциальной геометрии. Важное отличие заключается в том, что каждый раз, когда градуированные объекты переставляются, возникает знак, зависящий от их степени. Например, градуированное векторное поле  $V$  четной степени коммутирует само с собой автоматически:

$$[V, V] \equiv VV - (-1)^{\deg(V)\deg(V)} VV = 0.$$

Условие коммутирования векторного поля  $Q$  нечетной степени нетривиально:

$$[Q, Q] \equiv QQ - (-1)^{\deg(Q)\deg(Q)} QQ = 2Q^2.$$

Коммутирующее с собой векторное поле степени 1 называется *Q-структурой*.

Градуированное многообразие с таким векторным полем называется *дифференциальным градуированным многообразием* (или в физической литературе *Q-многообразием*). Мы рассмотрим несколько таких многообразий, на примере которых проиллюстрируем вычисления в градуированном случае.

### 5.1. Дифференциальные формы

Рассмотрим, как раньше, касательное расслоение к обычному многообразию  $\Sigma$ , линейные координаты  $\theta$  на слое которого имеют степень 1 – такой объект обозначается  $T[1]\Sigma$ . Так как  $\deg(\sigma^\mu) = 0$ ,  $\sigma^{\mu_1}\sigma^{\mu_2} = \sigma^{\mu_2}\sigma^{\mu_1}$  и  $\deg(h(\sigma^1, \dots, \sigma^d)) = 0$ . Равенство  $\deg(\theta^\mu) = 1$  означает, что  $\theta^{\mu_1}\theta^{\mu_2} = -\theta^{\mu_2}\theta^{\mu_1}$ . Произвольная однородная функция на  $T[1]\Sigma$  степени  $p$  имеет вид

$$f = \sum f_{\mu_1 \dots \mu_p}(\sigma^1, \dots, \sigma^d) \theta^{\mu_1} \dots \theta^{\mu_p}.$$

И как и раньше,

$$f \cdot g = (-1)^{\deg(f)\deg(g)} g \cdot f.$$

Легко заметить, что эта конструкция повторяет операции с дифференциальными формами, определенные в разделе 2:

$$f \leftrightarrow \alpha = \sum f_{\mu_1 \dots \mu_p} d\sigma^{\mu_1} \wedge \dots \wedge d\sigma^{\mu_p} \in \Omega(\Sigma).$$

Рассмотрим векторное поле  $Q = \sum \theta^\mu \frac{\partial}{\partial \sigma^\mu}$ . За-

метим, что  $\deg Q = 1$  и  $[Q, Q] \equiv 2Q^2 = 0$ , значит, оно является *Q-структурой*, повторяющей в точности внешний дифференциал.



### 5.2. Пуассоновы структуры

Пусть на многообразии  $M$  определена скобка Пуассона, то есть кососимметричная операция  $\{, \} : C^\infty(M) \times C^\infty(M) \rightarrow C^\infty(M)$ , удовлетворяющая тождеству Лейбница

$$\{f, gh\} = \{f, g\}h + g\{f, h\}$$

и тождеству Якоби

$$\{f, \{g, h\}\} + \{g, \{h, f\}\} + \{h, \{f, g\}\} = 0$$

С геометрической точки зрения, скобка Пуассона может быть переписана как  $\{f, g\} = \pi(df, dg)$ , где  $\pi \in \Gamma(\Lambda^2 TM)$  – бивекторное поле с компонентами  $\pi^{ij}(x) = \{x^i, x^j\}$  (см. замечание в разделе 3). Бивекторное поле является бидифференцированием, поэтому удовлетворяет тождеству Лейбница. Тождество Якоби в компонентах  $\pi$  имеет вид

$$\sum_i \left( \frac{\partial \pi^{ij}(x)}{\partial x^i} \pi^{jk}(x) + \frac{\partial \pi^{ki}(x)}{\partial x^i} \pi^{ji}(x) + \frac{\partial \pi^{jk}(x)}{\partial x^i} \pi^{li}(x) \right) = 0 \quad (5.1)$$

Рассмотрим теперь  $T^*[1]M$  – кокасательное расслоение к  $M$  с координатами на слое степени 1, то есть  $\deg(x^i) = 0$ ,  $\deg(p_i) = 1$ . На нем определена каноническая симплектическая форма  $\omega = \sum_i dp_i \wedge dx^i$ . Бивекторное поле тогда определяет функцию на  $T^*[1]M$ :  $H = \frac{1}{2} \pi^{ij} p_i p_j$ , а соответствующее Гамильтоново векторное поле имеет вид

$$Q_\pi = \pi^{ij}(x) p_j \frac{\partial}{\partial x^i} - \frac{1}{2} \frac{\partial \pi^{jk}}{\partial x^i} p_j p_k \frac{\partial}{\partial p_i}$$

Полезно проверить, что  $\deg(Q_\pi) = 1$ , а также что условие  $Q_\pi^2 = 0$  эквивалентно (5.1).

Таким образом, градуированная геометрия описывает аналогичным способом достаточно разные объекты, такие как дифференциальные формы и Пуассоновы структуры. Кроме того, в этот формализм вписываются структуры Дирака, описанные выше (см. общее утверждение в [12] и конкретные механические примеры в [13, 14]), а также алгебры Ли и Куранта и многие другие объекты. А они, как упоминалось выше, находят применение в механике и теоретической физике.

Стоит отметить, что при всей изящности определений градуированных объектов реальные вычисления для конкретных примеров достаточно объемны и не всегда прямолинейны. Таким образом, мы подходим к третьей задаче компьютерной алгебры, которую мы довольно расплывчато сформулируем следующим образом:

**Задача 3.** *Расширить стандартные пакеты дифференциальной геометрии для работы с градуированными объектами.*

Насколько нам известно, существующие пакеты для работы с некоммутирующими объектами (а, по сути, это и есть основное отличие градуированных многообразий) написаны для конкретных довольно узкоспециализированных задач. Пакеты для классической дифференциальной геометрии, напротив, развиты очень хорошо – их расширение сделало бы неоценимый вклад в соответствующие прикладные области. Более конкретно, во все стандартные операции (дифференцирование, подсчет сверток и коммутаторов векторных полей, алгебраическое упрощение, ...) полезно было бы включить знаки, обусловленные градуировками – это исключительно вопрос программирования. Более того, нужно рассмотреть соответствие довольно простых условий вида  $Q^2 = 0$  с классическими условиями вида (5.1), которые являются уравнениями с частными производными, и использовать их при упрощении выражений. Это нетривиальная задача даже без учета градуировок, одним из возможных подходов может быть описание уравнений с частными производными в пространстве джетов ([15, 16]).

## 6. ЗАКЛЮЧЕНИЕ/ОБСУЖДЕНИЕ

Итак, мы сформулировали три задачи компьютерной алгебры, возникающие при изучении геометрических аспектов механических систем. Первая из них довольно техническая – компьютерная алгебра может помочь избежать длинных ошибкоопасных вычислений. Вторая является более концептуальной, она призвана помочь в качественном анализе систем дифференциальных уравнений и избежать “угадывания” их корректной формы. Третья задача выглядит, как достаточно неизученное, потенциально очень интересное направление развития компьютерной алгебры.

Подчеркнем еще раз, что решения приведенных задач важно для нас не “из спортивного интереса”. Для решения механических задач они позволят создавать программные комплексы “полного цикла” (от описания физики системы до результатов численного счета). Мы упомянули этот подход во введении: в настоящее время каждый пример, разобранный по такой схеме, достоин публикации. Именно поэтому автоматизация процесса в рамках задачи 2, кажется естественным и важным направлением вычислительной механики. Это верно и для задачи 3, возникающей, например, в расчетах по теоретической физике или физике высоких энергий. В любом случае, мотивированный читатель может восприни-

мать эти задачи как приглашение к совместной работе.

Заметим в заключение, что кроме собственно описанных “непрерывных” конструкций, для многих из них важно построить дискретные аналоги. Это направление тоже интересует нас ([17, 18]) в рамках уже упомянутого проекта по “геометризации механики”, оно очевидным образом связано со сформулированными задачами.

#### БЛАГОДАРНОСТИ

Автор благодарен С.А. Абрамову, С.Я. Степанову, А. Прокопене за интересные вопросы и плодотворные обсуждения, посвященные данной тематике, во многом повлиявшие на эту статью, а также за полезные замечания при подготовке рукописи.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Courant T.J.* Dirac manifolds. *Trans. Am. Math. Soc.* 1990. V. 319. P. 631–661.
2. *Yoshimura H., Marsden J.E.* Dirac Structures in Lagrangian Mechanics Part I: Implicit Lagrangian Systems, *Journal of Geometry and Physics.* 2006. V. 57.
3. *Yoshimura H., Marsden J.E.* Dirac structures in Lagrangian mechanics. Part II: Variational structures, *Journal of Geometry and Physics.* 2006. V. 57.
4. *Salnikov V., Hamdouni A.* From modelling of systems with constraints to generalized geometry and back to numerics, *Z Angew Math Mech.* 2019.
5. *Razafindralandy D., Salnikov V., Hamdouni A., Deeb A.* Some robust integrators for large time dynamics, *Advanced Modeling and Simulation in Engineering Sciences,* 2019.
6. *Maschke B.M., van der Schaft A.J., Breedveld P.C.* An intrinsic Hamiltonian formulation of network dynamics: non-standard Poisson structures and gyrators. *J. Franklin Inst.* 1992. V. 329.
7. *van der Schaft A.* Port-Hamiltonian systems: an introductory survey, *Proceedings of the International Congress of Mathematicians, Madrid,* 2006.
8. *Tulczyjew W.M.* The Legendre transformation, *Ann. Inst. H. Poincaré, Sect. A.* 1977. V. 27 (1). P. 101–114.
9. *Verlet L.* Computer “Experiments” on Classical Fluids, *Physical Review.* 1967. V. 159. P. 98–103.
10. *Yoshida H.* Construction of higher order symplectic integrators. *Phys. Lett. A.* 1990. V. 150. P. 262.
11. *Falaize A., Hélie T.* Passive simulation of the nonlinear port-hamiltonian modeling of a rhodes piano, *Journal of Sound and Vibration,* 2016.
12. *Kotov A., Schaller P., Strobl T.* Dirac Sigma Models, *Commun. Math. Phys.* 2005. V. 260.
13. *Salnikov V., Hamdouni A.* Geometric integrators in mechanics – the need for computer algebra tools, *Proceedings of The third International Conference “Computer algebra”.* Moscow, Russia, 2019.
14. *Salnikov V., Hamdouni A.* Géométrie généralisée et gradué pour la mécanique, *proceedings of the Congrès Français de Mécanique.* Brest, France, 2019.
15. *Kushner A., Lychagin V., Rubtsov V.* Contact geometry and non-linear differential equations. *Encyclopedia of Mathematics and its Applications,* 101, Cambridge University Press, 2007.
16. *Krasil'shchik I.S., Vinogradov A.M. et al.* Symmetries and conservation laws for differential equations of mathematical physics, *AMS,* Providence, RI, 1999.
17. *Hamdouni A., Salnikov V.* Dirac integrators for port-Hamiltonian systems, in preparation.
18. *Salnikov V., Hamdouni A.* Discretization in the graded world, in preparation.

ВЫЧИСЛЕНИЕ ИНВОЛЮТИВНЫХ БАЗИСОВ И БАЗИСОВ ГРЁБНЕРА  
ИСПОЛЬЗУЯ ТАБЛИЧНОЕ ПРЕДСТАВЛЕНИЕ ПОЛИНОМОВ

© 2020 г. Д. А. Янович\*

Объединенный институт ядерных исследований 141980 Дубна Московской области, Россия

\*E-mail: yan@jinr.ru

Поступила в редакцию 19.08.2019 г.

После доработки 20.10.2019 г.

Принята к публикации 19.11.2019 г.

При работе с полиномами обычно используются такие представления данных, как списки термов, геобакеты, кучи. В работе предпринята попытка по новому взглянуть на проблему представления полиномов для задач вычисления инволютивных базисов и базисов Грёбнера систем нелинейных полиномиальных уравнений. Также новый подход позволяет передать часть этой вычислительной задачи на GPU, что открывает перспективы решения более сложных проблем.

DOI: 10.31857/S0132347420020120

## 1. ВВЕДЕНИЕ

## Обозначения, используемые в работе

 $X$  – множество переменных  $X = \{x_1, \dots, x_n\}$ ; $\mathbb{R} = \mathbb{K}[X]$  – кольцо многочленов над кольцом  $\mathbb{K}$ ; $f, g, h, q, r$  – многочлены из  $\mathbb{R}$ ; $a, b, c$  – элементы из  $\mathbb{K}$ ; $F, G, H$  – конечные подмножества из  $\mathbb{R}$ ; $\langle F \rangle$  – идеал из  $\mathbb{R}$  порожденный  $F$ ; $\mathbb{Z}_{\geq 0}$  – множество неотрицательных чисел; $\mathbb{M} = \{x_1^{d_1} \dots x_n^{d_n} | d_i \in \mathbb{Z}_{\geq 0}\}$  – множество мономов из  $\mathbb{R}$ ; $u, v, w, s, t$  – множество мономов или термов; $U, V, W$  – конечные подмножества из  $\mathbb{M}$ ; $\deg_i(u)$  – степень переменной  $x_i$  в  $u$ ; $\deg(u)$  – полная степень монома  $u$ ; $\succ$  – допустимое мономиальное упорядочение с порядком переменных  $x_1 \succ \dots \succ x_n$ ; $\text{It}(f)$  – лидирующий терм относительно упорядочения  $\succ$ ; $\text{lm}(f)$  – лидирующий моном  $f$ ; $\text{lm}(F) = \{\text{lm}(f) | f \in F\}$  – множество лидирующих мономов из  $F$ ; $\text{lcm}(F)$  – наименьшее общее кратное множества мономов из  $\text{lm}(F)$ ; $u|v$  означает, что моном  $u$  делит моном  $v$ .

## Инволютивные базисы и базисы Грёбнера

Базис Грёбнера (и инволютивный базис) является известным математическим инструментом, широко применяемым в задачах алгебраической геометрии, биоинформатики, вычислительной физики, робототехники и криптографии: везде, где необходимо решать системы нелинейных алгебраических, разностных и дифференциальных уравнений.

Определение и алгоритм построения базисов Грёбнера были предложены Б. Бухбергером [1].

**Определение 1.** [1, 2] Пусть заданно допустимое мономиальное упорядочение. Конечное подмножество  $G = \{g_1, \dots, g_m\}$  элементов идеала  $I$  называется его базисом Грёбнера, если

$$\langle \text{lm}(g_1), \dots, \text{lm}(g_m) \rangle = \langle \text{lm}(I) \rangle.$$

В данной работе для вычисления базиса Грёбнера применяется инволютивный подход, который описан в работах [3, 4].

Если, некоторым самосогласованным способом запретить деление по некоторым переменным, называемыми *немультимпликативными* и разрешить по другим, называемыми *мультимпликативными*, то мы получим сужение операции деления – *инволютивное деление*.

**Определение 2.** [3, 4] Мы будем говорить, что на множестве мономов  $\mathbb{M}$  определено инволютивное деление  $L$ , если для любого конечного не пустого  $U \subset \mathbb{M}$  и для любого  $u \in U$  задано  $M_L(u, U) \subseteq X$ , генерирующее подмоноид  $L(u, U) \subset M$ , состоящий из мономов с переменными из  $M_L(u, U)$ , удовлетворяющий следующим условиям:

(a) Из  $u, v \in U$  и  $uL(u, U) \cap vL(v, U) \neq \emptyset$  следует  $u \in vL(v, U)$  или  $v \in uL(u, U)$ ;

(b) Из  $v \in U$  и  $v \in uL(u, U)$  следует  $L(v, U) \subseteq L(u, U)$ ;

(c) Из  $u \in V$  и  $V \subseteq U$  следует  $L(u, U) \subseteq L(u, V)$  для всех  $u \in V$ .

Элементы  $M_L(u, U)$  называются  $L$ -мультипликативными для  $u$ , элементы  $NM_L(u, U) = X \setminus M_L(u, U)$  называются  $L$ -немultiпликативными. Если  $w \in uL(u, U)$ , то  $u$  называется ( $L$ -)инволютивным делителем  $w$  и обозначается  $u|_L w$ . В свою очередь, моном  $w$  называется ( $L$ -)кратным  $u$ .

**Определение 3.** [3, 4] Множество мономов называется  $L$ -инволютивным, если

$$(\forall u \in U)(\forall w \in \mathbb{M})(\exists v \in U)[v|_L uw].$$

**Определение 4.** [3] Авторедуцированное множество  $F$  называется частично инволютивным до монома  $v$  в допустимом мономиальном упорядочении  $\succ$  если выполнено следующее условие

$$(\forall f \in F)(\forall u \in \mathbb{M})(\text{lm}(f) \succ v)[NF_L(fu, F) = 0],$$

где  $NF_L$  означает инволютивную нормальную форму, т.е. полином, ни один из мономов термов которого не делится лидирующими мономами множества  $F$ .

Таким образом, используя конструктивное и нётерово деление (например деление Жане), пополняя частичный базис ненулевыми инволютивными нормальными формами немultiпликативных продолжений элементов частично инволютивного множества  $F$ , мы в конце концов получим его инволютивный базис.

Приведем упрощенный алгоритм 1 для вычисления инволютивного базиса, не использующий критерии для пропуска редукций  $p$  к нулю. Полный алгоритм приведен в [3].

**Алгоритм 1.** Вычисление инволютивного базиса

**Input:**  $F, L, \prec$

**Output:**  $T$  – инволютивный базис  $F$

```

1:  $T := \emptyset$ 
2:  $Q := F$ 
3: while  $Q \neq \emptyset$  do
4:    $P := \{p \in Q | \text{lm}(p) = \min(\text{lm}(Q))\}$ ,
5:    $NF_L(p, T) \neq 0\}$ 
6:    $T := T \cup P$ 
7:    $Q := Q \setminus P$ 
8:   for all  $\{p \in P\}$  do
9:      $Q := Q \cup \{p \cdot NM_L(p, T)\}$ 
10:  od
11: for all  $\{r \in T\}$  do

```

```

12:   if  $\text{lm}(r) \sqsupseteq \text{lm}(p)$  then

```

```

13:      $Q := Q \cup \{r\}$ ;

```

```

14:      $T := T \setminus \{r\}$ 

```

```

15:   fi

```

```

16: od

```

```

17: od

```

Связь между инволютивным базисом и базисом Грёбнера определяется следующей теоремой:

**Теорема 1.** [3] *Инволютивный базис, авторедуцированный в смысле обычного деления, является редуцированным базисом Грёбнера.*

## 2. МОТИВАЦИЯ

Получение базисов Грёбнера (и инволютивных базисов) является тяжелой вычислительной задачей как по расходу оперативной памяти, так и по времени, необходимому для завершения алгоритма. Наибольшую трудность представляют вычисления нормальной формы полинома, которые занимают более 90% времени работы. Также известно, что сложность можно снизить сначала вычислив базисы с коэффициентами из кольца модулярных чисел (или с числами Фарей [5]) и потом восстановив базис с коэффициентами из  $\mathbb{Q}$ . Предлагаемое представление полиномов делает вычисления более подходящими современной архитектуре процессоров и адаптированными к способам работы с оперативной памятью. Также данное представление позволяет перенести часть операций с полиномами на GPU.

## 3. ПРЕДСТАВЛЕНИЕ ПОЛИНОМОВ

При вычислениях с модулярными коэффициентами в известных представлениях данных для полиномов от многих переменных (списках термов, кучах [6], геобакетах [7] и т.п.) большую часть места в оперативной памяти занимают именно мономы. В задачах из области криптографии [8, 9] и задачах SAT-выполнимости могут присутствовать тысячи переменных, что ведет к “разбуханию” этих структур данных до таких масштабов, что 16Гб RAM не хватает даже на несколько секунд вычислений. В работе [10] автор уже предлагал один из методов снижения потребления памяти, развитие идей этой работы и привело к разработке нового представления полиномов, основанного на следующих идеях:

- нам не надо представлять *все возможные* комбинации мультииндексов, в задачах вычисления базисов Грёбнера множество возможных мономов ограничено широко известной фигурой *staircase* [11];

- в процессе вычислений в множестве полиномов  $Q$ , которые необходимо рассмотреть в ходе

алгоритма, образуется большое количество полиномов, содержащих много одинаковых мономов.

Подсчитав количество различных мономов, встречающихся в ходе вычислений базисов на примерах из [12] (см. таблицу 1), можно утверждать, что оно очень мало по сравнению с теоретической оценкой, которую можно найти в работе [13]. Наибольшее количество мономов, встреченных автором в примере *noon9*: 131860, а теоретическая оценка составляет  $\binom{9+17}{9} = 3124550$ .

Представим все полиномы, которые находятся в частичном базисе и в множестве  $Q$ , в виде единой таблицы  $T_K$ , столбцам которой сопоставим мономы, в данный момент входящие в базис, а строкам – сами полиномы. В ячейках таблицы находятся коэффициенты при мономах.

Для обеспечения произвольного доступа к коэффициентам полинома по мономам и для упрощения работы с данными к таблице добавим еще несколько структур:

- связанный список мономов  $L_{\prec}$ , отсортированный по убыванию по упорядочению  $\prec$ , каждому элементу сопоставим номер его столбца в таблице  $T_K$ ;
- хэш-таблицу  $H_{col}$ , по ключу-моному выдающему номер его столбца в таблице  $T_K$ ;
- хэш-таблицу  $H_{mon}$ , по номеру столбца таблицы  $T_K$  выдающую моном, ему соответствующий.

#### 4. ДЕТАЛИ РЕАЛИЗАЦИИ

Для задач вычисления базисов Грёбнера и инволютивных базисов представление полиномов должно обеспечивать высокую эффективность следующих операций, перечисленных в порядке снижения важности:

- редукция полиномов (как по лидирующим, так и по хвостовым термам), включающая поиск подобных термов;
- умножение полинома на переменную (продолжение);
- нахождении старшего монома в полиноме

*Умножение полинома на переменную* фактически является переносом всех его коэффициентов в другие столбцы таблицы, соответствующие мономам, которые получаются при умножении каждого монома полинома на переменную. Это является достаточно сложной операцией (см. алгоритм 2), которая потенциально приводит к необходимости расширения таблицы при добавлении новых столбцов, соответствующих мономам, которых еще не было в полиномах частичного базиса. С течением времени все необходимые для вычислений мономы будут найдены и изме-

нения таблицы прекратятся. Очевидно, что этот же алгоритм применяется для умножения полинома на моном.

**Алгоритм 2.** Умножение полинома на переменную

---

**Input:**  $i$  – строка полинома в  $T_K$ ,  
 $x_j$  – переменная, на которую умножаем,  
 $i'$  – строка произведения

- 1: **for all**  $k$  – номер столбца  $T_K$ ,  $T_K[i][k] \neq 0$  **do**
- 2:      $u = H_{mon}[k] \cdot x_j$
- 3:     **if**  $u \notin H_{col}$  **then**
- 4:         **if**  $T_K$  – заполнена **then**
- 5:             расширяем  $T_K$  с запасом
- 6:         **fi**
- 7:          $l$  – свободный столбец  $T_K$
- 8:          $H_{col}[l] = u$
- 9:          $H_{mon}[u] = l$
- 10:         в  $L_{\prec}$  вставляем  $(u, l)$
- 11:         **fi**
- 12:          $T_K[i'][H_{col}[u]] = T_K[i][k]$
- 13: **od**

---

*Редукция полинома  $f$  по  $g$*  выполняется по алгоритму 3.

**Алгоритм 3.** Редукция полиномов

---

**Input:**  $f$  – полином,  $i$  – его строка в  $T_K$ ,  
 $t_f$  – терм, который надо редуцировать,  
 $g$  – полином,  $j$  – его строка в  $T_K$ ,  
 $t_g$  – терм, по которому надо редуцировать

- 1: находим  $u \in \mathbb{M}$ ,  $a, b \in \mathbb{K} : b \cdot t_f = a \cdot t_g \cdot u$
- 2: составляем временный полином в строке  $j'$ , умножая  $g$  на  $u$
- 3: **for all**  $k$  – номер столбца  $T_K$  **do**
- 4:      $T_K[i][k] = b \cdot T_K[j][k] - a \cdot T_K[j'][k]$
- 5: **od**

---

После некоторого количества редукций обязательно надо проверить плотность таблицы и осуществить компрессию убрав нулевые столбцы.

Одним из преимуществ предлагаемого представления является ориентированность на архитектуру современных процессоров и учет кэширования данных памяти. Процессоры намного быстрее обрабатывают данные из кэша, чем из оперативной памяти, разница может достигать нескольких десятков раз. Поэтому очень важно, и в каком порядке будут обработаны, чтобы успеть доставить их в кэш. Если выполнять редукцию по алгоритму 3, то процессор уверен, ка-

**Таблица 1.** Количество мономов в вычислениях

Пример	#	Пример	#
assur44	544	eco11	6215
butcher8	586	eco12	19077
chemequs	77	extcyc5	907
chemkin	1124	extcyc6	4365
cohn3	651	f744	1611
cpdm5	515	f855	4404
cyclic5	168	fabrice24	478
cyclic6	583	filter9	2673
cyclic7	4738	hairer2	10111
cyclic8	32122	hairer3	11690
d1	658	hcyclic7	6208
des18_3	367	hcyclic8	42504
des22_24	506	hf744	8069
discret3	1101	hf855	59707
dl	3520	hietarinta1	870
eco8	482	i1	1043
eco9	1222	ilias13	3609
eco10	2548	ilias_k_2	1579

кие данные следует получить, что выражается в значительном снижении кэш-промахов. Данное утверждение можно проверить, используя инструмент `valgrind-tool=che grind` [14]. Для примера *reimer6* списочное представление имеет 9.6% кэш-промахов, а табличное – всего 5.4%, аналогичная картина наблюдается и для остальных примеров.

Также код, более подходящий для современных процессоров, должен содержать как можно меньше условий, которые могут приводить к сбросу конвейера выполнения инструкций и вследствие этого приводящих к уменьшению скорости работы суперскалярных блоков выполнения CPU. Поэтому редукции с нулевыми коэффициентами не обходятся, а выполняются так же, как и значащие редукции, поскольку условный их обход приводил бы к понижению производительности.

*Нахождение старшего монома в полиноме.* После проведения операции редукции может понадобиться обновить указатель на старший моном полинома. Нахождение его производится простой итерацией по списку  $L_k$  до первого ненулевого коэффициента в соответствующем столбце строки полинома в таблице  $T_k$ .

## 5. ВОЗМОЖНОСТЬ ВЫЧИСЛЕНИЯ БАЗИСОВ НА GPU

Современные GPU являются мощнейшими процессорами для вычислений общего типа, об-

**Таблица 2.** Типичное профилирование GPU-вычислений

Время(%)	Вызов	Процедура
15.48%	470890	gpuReduceEle(..)
84.83%	941780	cudaMemcpy
14.01%	470890	cudaLaunchKernel

ладающими некоторыми особенностями, которые надо учитывать при адаптации алгоритмов для них. Во-первых, нужно учитывать архитектуру SIMD: на GPU запускается так называемое ядро вычислений (kernel), которое во множестве потоков обрабатывает одинаковым образом входные данные, организованные в batch и warp разделы (используется терминология CUDA от фирмы Nvidia). Во-вторых, наиболее эффективно происходят вычисления с большими объемами данных, поскольку GPU имеют собственную локальную память, в которую копируются исходные данные и из которой они потом должны быть ско-

**Таблица 3.** Сравнение списочного и табличного представлений

Пример	$t_{list}$	$t_{table}$
cyclic7	2.39	5.16
cyclic8	273.54	559.85
dl	14.6	56.91
eco10	2.86	6.84
eco11	42.68	59.64
eco12	1538.1	<b>631.47</b>
extcyc6	3.8	9.56
f855	2.82	14.8
hairer2	6.65	10.17
hcyclic7	2.56	10.99
hcyclic8	279.67	1426.53
hf744	2.38	16.72
hf855	259.93	1462.96
ilias13	10.88	35.01
katsura9	17.29	21.56
katsura10	225.78	240.00
katsura11	2513.21	2619.63
noon7	3.94	22.69
noon8	139.72	791.98
noon9	5797.72	*
redcyc7	2.17	5.03
redco11	20.91	34.56
redco12	244.33	376.01
reimer6	1.63	1.98
reimer7	228.27	<b>146.60</b>

пированы обратно в память, принадлежащую CPU. Из этого возникает третье условие эффективной адаптации: передач CPU  $\leftrightarrow$  GPU должно быть как можно меньше.

На данный момент автором реализован простейший вариант адаптации: перенос части алгоритма 3 на GPU. В процессе вычислений на CPU конструируются два полинома в табличном представлении, которые передаются на GPU, редуцируются ядром

```

__global__ void gpuReduceEle
    (mtype* aWhat, mtype* aByWhat,
     int aN,
     const mtype aA, const mtype aB,
     const mtype aM,
     const int aK, const mtype aMod) {
    const int i = blockIdx.x * blockDim.x
        + threadIdx.x;
    if (i < aN) {
        mtype c = aWhat[i] * aA
            + aByWhat[i] * aB;
        gpuBarretRed(c, aM, aK, aMod);
        aWhat[i] = c;
        aByWhat[i] = 0;
    }
}

```

и передаются обратно. Такая реализация является скорее концептом, который позволяет удостовериться, что вычисления инволютивных базисов и базисов Грёбнера с использованием GPU возможно. В таблице 2 показаны результаты профилирования запуска вычислений примера *reimerb* на видеокарте начального уровня NVidia GT710, способной запускать одно ядро вычислений используя до 1024 потоков обработки данных с использованием CUDA 10.0 (то, что сумма процентов больше 100 является особенностью работы профилировщика от NVidia). Видно, что собственно сами вычисления в ядре занимают лишь малую часть, основные же затраты идут на копирование данных и накладные расходы на запуск вычислений. Времена вычислений на GPU приведены в данной работе не будут, поскольку целью было показать именно принципиальную возможность вычислений используя предложенное представление данных, так как довольно очевидно, что ни одно из классических представлений полиномов не позволит получить даже такого результата.

## 6. РЕЗУЛЬТАТЫ ТЕСТОВЫХ ПРОГОНОВ

В таблице 3 представлены результаты сравнения реализаций алгоритма вычисления инволютивных базисов с использованием табличного и классического списочного представлений полиномов. В программах различались только пред-

ставления данных, остальная реализация (ввод-вывод, работа с инволютивными делениями и т.д.) одинакова. Вычисления велись на компьютере с CPU Intel Core i5-3470@3.20GHz, RAM 8Gb под управлением Ubuntu 16.04LTS. В качестве хэш-таблицы использовалась *spp::sparse\_hash\_map* от Google [15], примеры взяты из [12]. Видно, что табличное представление сравнимо со списочным, в большинстве случаев медленнее, но в двух из представленных примеров существенно быстрее.

## 7. ЗАКЛЮЧЕНИЕ

Несмотря на то, что в большинстве примеров при вычислениях на CPU текущая реализация табличного представления оказалось медленнее списочного, автор считает данное представление полиномов весьма перспективным как с точки зрения ускорения расчетов на CPU, так и с точки зрения переноса вычислений на GPU.

## СПИСОК ЛИТЕРАТУРЫ

1. Бухбергер Б. Базисы Грёбнера. Алгоритмический метод в теории полиномиальных идеалов // Компьютерная алгебра. Символьные и алгебраические вычисления. Москва: Мир, 1986. С. 331–372.
2. Кокс Д., Литтл Дж., О’ши Д. Идеалы, многообразия и алгоритмы. Введение в вычислительные аспекты алгебраической и коммутативной алгебры: Пер. с англ. // М.: Мир, 2000. 687 с., ил.

3. *Gerdt V.P., Blinkov Yu.A.* Involutive Bases of Polynomial Ideals // *Math. Comp. Sim.* 1998. V. 45. P. 519–542.
4. *Гердт В.П., Блинков Ю.А.* Инволютивное деление мономов // *Программирование.* 1998. Т. 6. С. 22–24.
5. *Янович Д.А.* Параллельное модулярное вычисление базисов Грёбнера и инволютивных базисов // *Программирование.* 2013. Т. 39. № 2. С. 75–80.
6. *Johnson S.C.* Sparse polynomial arithmetic // *ACM SIGSAM Bulletin.* 1974. V. 8 (3). P. 63–71.
7. *Yan T.* The Geobucket Data Structure for Polynomials // *Journal of Symbolic Computation.* 1998. V. 25. P. 285–293.
8. *Cid C., Murphy S., Robshaw M.J.B.* Small Scale Variants of the AES // *12th International Workshop, FSE 2005, Paris, France, February 21–23, Revised Selected Papers.* 2005. P. 145–162.
9. *Jean-Charles Faugère, Antoine Joux* Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases // *23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17–21. Proceedings.* 2003. P. 44–60.
10. *Янович Д.А.* Компактное представление полиномов для алгоритмов вычисления базисов Грёбнера и инволютивных базисов // *Программирование.* 2015. Т. 41. № 2. С. 63–68.
11. *Zimmermann P., Casamayou A., Cohen N., Connan G. et al.* Computational Mathematics with SageMath // *SIAM.* 2018. P. 203–205.
12. *Vershelde J.* The Database with Test Examples. <http://www.math.uic.edu/jan/demo.html>
13. *Monagan M.* Sparse polynomial division using a heap // *Journal of Symbolic Computation.* 2011. V. 46. P. 807–822
14. <http://www.valgrind.org>
15. <https://github.com/sparsehash/sparsehash>