

Санкт-Петербургский Государственный Университет
Факультет Прикладной Математики - Процессов Управления
Системный анализ, информатика и управление

Саакян Артур Темиевич

Выпускная квалификационная работа

**Построение схем для ускорения численного интегрирования
дифференциальных уравнений**

Научный руководитель,
доктор физ.-мат. наук,
профессор

Бабаджанянц Л.К.

Рецензент,
ООО Институт компьютерных технологий
менеджер проекта,
кандидат физ.-мат. наук,
старший преподаватель

Брэгман К.М.

Санкт-Петербург

2021

Оглавление

	Стр.
Введение	4
Глава 1. Сведение дифференциальных уравнений к полиномиальной форме	7
1.1 Метод дополнительных переменных	7
1.2 Примеры	10
Глава 2. Схемы и быстрое вычисление систем мономов многих переменных	15
2.1 Основные определения	15
2.2 Быстрое вычисление систем мономов многих переменных	17
2.2.1 Системы мономов до третьей степени	18
2.2.2 Системы мономов третьей степени и выше	19
2.3 Примеры построения схем	20
Глава 3. Методы рядов Тейлора	24
3.1 Метод рядов Тейлора для полиномиальных систем	24
3.1.1 Коэффициенты Тейлора	24
3.1.2 Формулировка метода рядов Тейлора	27
3.1.3 Оценки локальной погрешности	28
3.1.4 Вспомогательные алгоритмы	29
3.1.5 Общий алгоритм метода рядов Тейлора	32
3.2 Реализация метода рядов Тейлора (МРТ)	33
Глава 4. Численные эксперименты	37
4.1 Произвольный набор мономов	37
4.2 Задача N тел	38
Заключение	40
Список литературы	41

Список таблиц 53

Приложение А. Программа расчёта схемы 54

Введение

Вначале рассмотрим структуру работы: кратко изложим её содержание по главам, скажем о её практической значимости и результатах, сформулируем цели работы, её актуальность, новизну и положения, выносимые на защиту. Далее более обстоятельно обсудим связанные с нашей работой проблемы математического моделирования динамических процессов и основные проблемы, решаемые в работе.

В первой главе "Сведение дифференциальных уравнений к полиномиальной форме" изложены необходимые понятия, и приводится алгоритм метода дополнительных переменных и алгоритм сведения дифференциальных уравнений к полиномиальной форме, метод применим как для обыкновенных дифференциальных уравнений, так и для полных систем дифференциальных уравнений в частных производных. Рассмотрено пять примеров сведения.

Во второй главе "Схемы и быстрое вычисление систем мономов многих переменных" представлены необходимые определения, сформулирована задача быстрого вычисления систем мономов многих переменных, представлен алгоритм решения проблемы и приведены примеры, показывающие эффективность работы алгоритма.

В третьей главе "Методы рядов Тейлора" представлены алгоритм реализации метода рядов Тейлора, алгоритм вычисления коэффициентов Тейлора, оценка локальной погрешности, вспомогательные алгоритмы и общий алгоритм метода рядов Тейлора.

В четвертой главе "Численные эксперименты" представлен численный анализ эффективности схем как на произвольном наборе мономов, так и на примере задачи N тел в различных полиномиальных формах.

Целью данной работы является разработка общих подходов, методов и алгоритмов моделирования в символьной и численной формах в задачах Динамики, основанных на применении систем дифференциальных уравнений.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Разработать детали построения схем,
2. Реализовать алгоритмы построения схем для произвольного набора мономов,

3. Разработать компьютерные программы, реализующие алгоритмы построения схем.
4. Провести численные эксперименты, исследовать эффективность разработанных алгоритмов.

Научная новизна:

1. Впервые представлены алгоритмы построения схем для быстрого вычисления произвольного набора мономов.
2. Впервые выполнено исследование, показавшее высокую эффективность представленных алгоритмов для численного интегрирования произвольных полиномиальных систем дифференциальных уравнений.

Практическая значимость. Ускорение численного интегрирования дифференциальных уравнений в полиномиальной форме, описывающих как реальные, так и статистически сформированные модели.

Основные положения, выносимые на защиту:

1. Построение и полный анализ алгоритмов и соответствующих компьютерных программ построения схем вычисления всех мономов произвольного набора мономов.
2. Численные эксперименты для реальных и статистически сформированных моделей Динамики.

Достоверность. Все результаты работы получены строгими математическими методами, проверены при помощи многочисленных вычислений и опираются на шесть публикаций в российских и международных рецензируемых журналах. Все эти результаты докладывались на многочисленных международных конференциях.

Апробация работы. Основные результаты работы докладывались на международной научной конференции "Процессы управления и устойчивость" (г. Санкт-Петербург, март 2016 г.), международной конференции "Устойчивость и колебания нелинейных систем управления (конференция Пятницкого)" (г. Москва, июнь 2016 г.), на 14th и 15th "International Conference of Numerical Analysis and Applied Mathematics" (г. Родос и г. Салоники, Греция, сентябрь 2016 г. и сентябрь 2017 г.), международной научной конференции по механике "VIII Поляховские чтения" (г. Санкт-Петербург, февраль 2018 г.) и на "International Multidisciplinary Scientific GeoConference Surveying, Geology and Mining, Ecology and Management" (г. София, Болгария, июль 2019 г.).

Личный вклад. Автор принимал активное участие в разработке и имплементации алгоритмов построения схем и проведении анализа эффективности представленных алгоритмов. Все результаты, представленные в работе, получены лично автором.

Публикации. Основные результаты изложены в 6 печатных изданиях [3, 22, 26, 30, 35, 43], 5 из которых изданы в журналах, рекомендованных ВАК.

Объем и структура работы. Работа состоит из введения, четырех глав и заключения. Полный объем работы составляет 55 страниц, включая 3 таблицы. Список литературы содержит 127 наименований.

Глава 1. Сведение дифференциальных уравнений к полиномиальной форме

При написании данной главы использовались многие источники: [1 – 12, 22, 26, 30, 39, 40].

Данная глава состоит из двух разделов. В первом разделе описана теоретическая и алгоритмическая основа методов сведения систем к полиномиальной форме (система дифференциальных уравнений первого порядка с полиномами по неизвестным в правой части), для полных систем дифференциальных уравнений и систем функций. Метод дополнительных переменных сводит их к системе дифференциальных уравнений в полиномиальной форме. Отметим метод дополнительных переменных для полных систем дифференциальных уравнений в частных производных, необходимые и достаточные условия применимости и их реализация были представлены в работе [9], метод дополнительных переменных для обыкновенных дифференциальных уравнений был представлен в работе А. Пуанкаре [105].

Во втором разделе приведены примеры сведения систем дифференциальных уравнений и систем функций: задача Коши для ОДУ, задача Коши для систем ОДУ, задача Коши для полных систем, математический маятник, вращательное движение спутника вокруг своего центра масс, задача N тел, сведенная к трем полиномиальным формам: пятой, четвертой и третьей степени, система из двух функций, экзотическая функция и многоэтажная экспонента.

Кроме того, в пятой главе представлены 10 категорий функций, соответствующие замены, необходимые для сведения их к дифференциальным уравнениям, и соответствующие дифференциальные уравнения.

1.1 Метод дополнительных переменных

Метод дополнительных переменных направлен на приведение систем функций и/или дифференциальных уравнений (полных дифференциальных уравнений в частных производных и, в частности, обыкновенных дифференциальных уравнений) к полиномиальной форме. Идея метода для ОДУ

рассмотрена в работе Пуанкаре [105]. В небесной механике эта идея использовалась для решения задачи трех тел с использованием степенных рядов. Необходимые и достаточные условия для полных дифференциальных уравнений в частных производных, позволяющие компьютеризировать применение метода, были предложены в [9], а алгоритм метода и его реализация приведены в [11].

Рассмотрим основные обозначения, которые будем использовать в дальнейшем. Пусть

$x = (x_1, \dots, x_m) \in C^m$, $t = (t_1, \dots, t_s) \in C^s$, $\alpha = (\alpha_1, \dots, \alpha_\omega) \in C^\omega$, $f_i^j \in C$, $(y_1, \dots, y_N) \in C^N$, $g_r \in C$, предполагая x функцией переменной t и параметра α , а y - функцией переменной x и параметра α . Символ C обозначает поле комплексных чисел. Отметим, что везде символ C можно заменить на символ R , который обозначает поле вещественных чисел, так как все описанные ниже алгоритмы символьные.

Рассмотрим полную систему дифференциальных уравнений в частных производных первого порядка, разрешенную относительно производной. Такие системы дифференциальных уравнений можно записать в следующих формах:

$$\begin{aligned} \frac{\partial x_i}{\partial t_j} &= f_i^j(x, \alpha), \quad i \in [1 : m], \quad j \in [1 : s], \\ \frac{\partial x}{\partial t} &= f(x, \alpha), \quad dx = f(x, \alpha)dt, \end{aligned} \tag{1.1}$$

где $x = (x_1, \dots, x_m) \in C^m$, $t = (t_1, \dots, t_s) \in C^s$, $\alpha = (\alpha_1, \dots, \alpha_\omega) \in C^\omega$, $dx = (dx_1, \dots, dx_m)$, $dt = (dt_1, \dots, dt_s)$, $\frac{\partial x}{\partial t} = \frac{\partial x_i}{\partial t_j}$, $f = (f_i^j)$, $f_i^j \in C$.

В случае обыкновенных дифференциальных уравнений (т.е. при $s = 1$) эти формы можно свести к следующей:

$$\frac{dx_i}{dt} = f_i(x, \alpha), \quad i \in [1 : m], \quad \frac{dx}{dt} = f(x, \alpha). \tag{1.2}$$

Также рассмотрим систему функций:

$$y_r = g_r(x, \alpha), \quad r \in [1 : N], \tag{1.3}$$

где $s, N \in [0 : +\infty)$ и $(s, N) \neq 0$. Систему (1.1) (или (1.3)), правая часть которой является полиномом по неизвестным x_1, \dots, x_m , называют полиномиальной системой.

Рассмотрим отдельно: метод дополнительных переменных для полных систем, метод дополнительных переменных для систем функций и метод дополнительных переменных для смешанных систем.

Метод дополнительных переменных для полных систем

Пусть дополнительные переменные x_{m+1}, \dots, x_{m+k} удовлетворяют условиям:
 — все производные $\frac{\partial x_{m+l}}{\partial x_i}$, ($l \in [1 : k]$, $i \in [1 : m]$) — некоторые полиномы $P_{m+l,i}(x_1, \dots, x_{m+k})$ по переменным x_1, \dots, x_{m+k} ;
 — все правые части уравнений (1.1) — полиномы $Q_i^j(x_1, \dots, x_{m+k})$,
 то x_1, \dots, x_{m+k} удовлетворяют полиномиальной системе:

$$\begin{aligned} \frac{\partial x_i}{\partial t_j} &= Q_i^j(x_1, \dots, x_{m+k}), \quad i \in [1 : m], \quad j \in [1 : s], \quad l \in [1 : k], \\ \frac{\partial x_{m+l}}{\partial t_j} &= \sum_{i=1}^m Q_i^j(x_1, \dots, x_{m+k}) P_{m+l,i}(x_1, \dots, x_{m+k}). \end{aligned}$$

Метод дополнительных переменных для систем функций

Пусть дополнительные переменные x_{m+1}, \dots, x_{m+k} удовлетворяют условиям:
 — все производные $\frac{\partial x_{m+l}}{\partial x_i}$, ($l \in [1 : k]$, $i \in [1 : m]$) — некоторые полиномы $P_{m+l,i}(x_1, \dots, x_{m+k})$ по переменным x_1, \dots, x_{m+k} ;
 — все функции $g_r(x, \alpha)$, $r \in [1 : N]$, — полиномы $R_r(x_1, \dots, x_{m+k})$, то x_1, \dots, x_{m+k} удовлетворяют полиномиальной системе:

$$\begin{aligned} y_r &= R_r(x_1, \dots, x_{m+k}), \quad r \in [1 : N], \\ \frac{\partial x_{m+l}}{\partial t_j} &= P_{m+l,i}(x_1, \dots, x_{m+k}), \quad i \in [1 : m], \quad l \in [1 : k]. \end{aligned}$$

Дифференцируя y_r по переменным x_1, \dots, x_m , можно получить полную систему:

$$\begin{aligned} \frac{\partial y_r}{\partial x_i} &= \frac{\partial R_r}{\partial x_i} + \sum_{s=1}^k P_{m+s,i} \frac{\partial R_r}{\partial x_{m+s}}, \quad r \in [1 : N], \quad i \in [1 : m], \\ \frac{\partial x_{m+l}}{\partial t_j} &= P_{m+l,i}(x_1, \dots, x_{m+k}), \quad l \in [1 : k]. \end{aligned}$$

Метод дополнительных переменных для смешанных систем

Пусть дополнительные переменные x_{m+1}, \dots, x_{m+k} удовлетворяют условиям:

- все производные $\frac{\partial x_{m+l}}{\partial x_i}$, ($l \in [1 : k]$, $i \in [1 : m]$) — некоторые полиномы $P_{m+l,i}(x_1, \dots, x_{m+k})$ по переменным x_1, \dots, x_{m+k} ;
- все правые части уравнений (1.1) — полиномы $Q_i^j(x_1, \dots, x_{m+k})$,
- все функции $g_r(x, \alpha)$, $r \in [1 : N]$, — полиномы $R_r(x_1, \dots, x_{m+k})$, тогда x_1, \dots, x_{m+k} удовлетворяют полиномиальной системе:

$$\begin{aligned} \frac{\partial x_i}{\partial t_j} &= Q_i^j(x_1, \dots, x_{m+k}), \quad i \in [1 : m], \quad j \in [1 : s], \quad l \in [1 : k], \\ \frac{\partial x_{m+l}}{\partial t_j} &= \sum_{i=1}^m Q_i^j(x_1, \dots, x_{m+k}) P_{m+l,i}(x_1, \dots, x_{m+k}), \\ y_r &= R_r(x_1, \dots, x_{m+k}), \quad r \in [1 : N], \end{aligned}$$

причем, дифференцируя y_r по переменным x_1, \dots, x_m , можно получить полную систему:

$$\begin{aligned} \frac{\partial x_i}{\partial t_j} &= Q_i^j(x_1, \dots, x_{m+k}), \quad i \in [1 : m], \quad j \in [1 : s], \quad l \in [1 : k], \\ \frac{\partial x_{m+l}}{\partial t_j} &= \sum_{i=1}^m Q_i^j(x_1, \dots, x_{m+k}) P_{m+l,i}(x_1, \dots, x_{m+k}), \\ \frac{\partial y_r}{\partial x_i} &= \frac{\partial R_r}{\partial x_i} + \sum_{s=1}^k P_{m+s,i} \frac{\partial R_r}{\partial x_{m+s}}, \quad r \in [1 : N], \quad i \in [1 : m]. \end{aligned}$$

1.2 Примеры

В первых трех из рассматриваемых здесь десяти примерах используются функции Гильберта (трех аргументов p_1, p_2, p_3).

Замечание. Эти функции были введены в связи с 13 проблемой Гильберта. К моменту формулировки Гильбертом этой проблемы было известно преобразование, сводящее уравнение n -ой степени, в котором свободный член равен 1, коэффициент при старшей степени равен 1, а коэффициенты при степенях $n - 1$, $n - 2$, $n - 3$ равны нулю. Проблема Гильберта: можно ли решить общее уравнение седьмой степени с помощью функций, зависящих только от двух переменных? Как указывалось, уравнение седьмой степени можно рассматривать,

как уравнение, решение которого зависит только от трёх коэффициентов.

Первый из них является решением уравнения:

$$\varphi_1^7(p_1, p_2, p_3) + p_3 \varphi_1^3(p_1, p_2, p_3) + p_2 \varphi_1^3(p_1, p_2, p_3) + p_1 \varphi_1(p_1, p_2, p_3) + 1 = 0,$$

при условии $\varphi_1(0, 0, 0) = -1$, а второй определяется равенством:

$$\varphi_2(p_1, p_2, p_3) = (7\varphi_1^6(p_1, p_2, p_3) + 3p_3\varphi_1^2(p_1, p_2, p_3) + 2p_2\varphi_1(p_1, p_2, p_3) + p_1)^{-1}.$$

Они удовлетворяют задаче Коши:

$$\begin{cases} \frac{\partial \varphi_1}{\partial p_j} = -\varphi_1^j \varphi_2, & j = 1, 2, 3, \\ \frac{\partial \varphi_2}{\partial p_j} = (42\varphi_1^5 + 6p_3\varphi_1 + 2p_2)\varphi_1^j \varphi_2^3 - j\varphi_1^{j-1} \varphi_2^2, \\ \varphi_1(0, 0, 0) = -1, \varphi_2(0, 0, 0) = \frac{1}{7}. \end{cases}$$

В четвертом и пятом примерах рассматриваются две задачи динамики – математический маятник и вращательное движение спутника около своего центра масс.

Примеры метода дополнительных переменных для полных систем.

Пример 1: Задача Коши для ОДУ.

Рассмотрим уравнение:

$$\frac{dx}{dt} = a \sin \varphi_1(x, x^2, x^3) + b \cos \varphi_1(x, x^2, x^3)$$

и начальные условия $x(0) = 0$ (a, b - вещественные параметры).

Введем дополнительные переменные:

$$\psi_1 = \varphi_1(x, x^2, x^3), \psi_2 = \varphi_2(x, x^2, x^3), \psi_3 = \sin \varphi_1(x, x^2, x^3), \psi_4 = \cos \varphi_1(x, x^2, x^3)$$

и получим их полные производные по t в силу этого уравнения:

$$\begin{aligned} \frac{d\psi_1}{dt} &= \sum_{j=1}^3 \left(\frac{\partial \varphi_1}{\partial x_j} \right)_{x_1=x, x_2=x^2, x_3=x^3} \cdot jx^{j-1} \frac{dx}{dt} = \\ &= \sum_{j=1}^3 \left(-j\varphi_1^{j-1} \varphi_2 \right)_{x_1=x, x_2=x^2, x_3=x^3} \cdot jx^{j-1} \frac{dx}{dt} = - \sum_{j=1}^3 jx^{j-1} \psi_1^j \psi_2 \cdot \frac{dx}{dt}, \\ \frac{d\psi_2}{dt} &= \sum_{j=1}^3 \left(\frac{\partial \varphi_2}{\partial x_j} \right)_{x_1=x, x_2=x^2, x_3=x^3} \cdot jx^{j-1} \frac{dx}{dt} = \end{aligned}$$

$$\begin{aligned}
&= \sum_{j=1}^3 \left((42\varphi_1^5 + 6x_3\varphi_1 + 2x_2)\varphi_1^j\varphi_2^3 - j\varphi_1^{j-1}\varphi_2^2 \right)_{x_1=x, x_2=x^2, x_3=x^3} \cdot jx^{j-1} \frac{dx}{dt} = \\
&= \sum_{j=1}^3 jx^{j-1} \left((42\psi_1^5 + 6x^3\psi_1 + 2x^2)\psi_1^j\psi_2^3 - j\psi_1^{j-1}\psi_2^2 \right) \frac{dx}{dt}, \\
&\quad \frac{d\psi_3}{dt} = \cos \varphi_1(x, x^2, x^3) \frac{d\varphi_1(x, x^2, x^3)}{dt} = \psi_4 \frac{d\psi_1}{dt}, \\
&\quad \frac{d\psi_4}{dt} = -\sin \varphi_1(x, x^2, x^3) \frac{d\varphi_1(x, x^2, x^3)}{dt} = -\psi_3 \frac{d\psi_1}{dt}.
\end{aligned}$$

Тогда можно записать исходное уравнение и начальные условия в форме полиномиальной задачи Коши:

$$\begin{cases} \frac{dx}{dt} = a\psi_3 + b\varphi_4, & \frac{d\psi_1}{dt} = -\sum_{j=1}^3 jx^{j-1}\psi_1^j\psi_2 \cdot \frac{dx}{dt}, \\ \frac{d\psi_2}{dt} = \sum_{j=1}^3 jx^{j-1} \left((42\psi_1^5 + 6x^3\psi_1 + 2x^2)\psi_1^j\psi_2^3 - j\psi_1^{j-1}\psi_2^2 \right) \frac{dx}{dt}, \\ \frac{d\psi_3}{dt} = \psi_4 \frac{d\psi_1}{dt}, & \frac{d\psi_4}{dt} = -\psi_3 \frac{d\psi_1}{dt}, \\ x(0) = 0, & \psi_1(0) = -1, \psi_2(0) = \frac{1}{7}, \psi_3(0) = -\sin 1, \psi_4(0) = \cos 1. \end{cases}$$

Пример 2: Задача Коши для системы ОДУ.

Рассмотрим уравнение:

$$\frac{dx_i}{dt} = a_i \sin \varphi_1(x, x_2, x_3) + b_i \cos \varphi_1(x, x_2, x_3)$$

и начальные условия $x_i(0) = 0$, $i \in [1 : 3]$, ($a_i = a_i(x, x_2, x_3)$, $b_i = b_i(x, x_2, x_3)$ - алгебраические полиномы).

Введем дополнительные переменные:

$$\psi_1 = \varphi_1(x, x_2, x_3), \psi_2 = \varphi_2(x, x_2, x_3), \psi_3 = \sin \varphi_1(x, x_2, x_3), \psi_4 = \cos \varphi_1(x, x_2, x_3)$$

и получим их полные производные по t в силу этого уравнения:

$$\begin{aligned}
\frac{d\psi_1}{dt} &= \sum_{j=1}^3 \frac{\partial \varphi_1}{\partial x_j} \frac{dx_j}{dt} = -\sum_{j=1}^3 \varphi_1^j \varphi_2 \frac{dx_j}{dt} = -\sum_{j=1}^3 x^{j-1} \psi_1^j \psi_2 \frac{dx_j}{dt}, \\
\frac{d\psi_2}{dt} &= \sum_{j=1}^3 \frac{\partial \varphi_2}{\partial x_j} \frac{dx_j}{dt} = \sum_{j=1}^3 \left((42\varphi_1^5 + 6x_3\varphi_1 + 2x_2)\varphi_1^j\varphi_2^3 - j\varphi_1^{j-1}\varphi_2^2 \right) \frac{dx_j}{dt} = \\
&= \sum_{j=1}^3 \left((42\psi_1^5 + 6x_3\psi_1 + 2x_2)\psi_1^j\psi_2^3 - j\psi_1^{j-1}\psi_2^2 \right) \frac{dx_j}{dt},
\end{aligned}$$

$$\begin{aligned}\frac{d\psi_3}{dt} &= \cos \varphi_1(x, x_2, x_3) \frac{d\varphi_1(x_1, x_2, x_3)}{dt} = \psi_4 \frac{d\psi_1}{dt}, \\ \frac{d\psi_4}{dt} &= -\sin \varphi_1(x, x_2, x_3) \frac{d\varphi_1(x_1, x_2, x_3)}{dt} = -\psi_3 \frac{d\psi_1}{dt}.\end{aligned}$$

Тогда можно записать исходную систему и начальные условия в форме полиномиальной задачи Коши:

$$\left\{ \begin{aligned} \frac{dx_i}{dt} &= a_i \psi_3 + b_i \varphi_4, \quad i \in [1 : 3], \quad \frac{d\psi_1}{dt} = -\sum_{j=1}^3 \psi_1^j \psi_2 \cdot \frac{dx_j}{dt}, \\ \frac{d\psi_2}{dt} &= \sum_{j=1}^3 \left((42\psi_1^5 + 6x_3\psi_1 + 2x_2) \psi_1^j \psi_2^3 - j\psi_1^{j-1} \psi_2^2 \right) \frac{dx_j}{dt}, \\ \frac{d\psi_3}{dt} &= \psi_4 \frac{d\psi_1}{dt}, \quad \frac{d\psi_4}{dt} = -\psi_3 \frac{d\psi_1}{dt}, \\ x_i(0) &= 0, \quad i \in [1 : 3], \quad \psi_1(0) = -1, \quad \psi_2(0) = \frac{1}{7}, \quad \psi_3(0) = -\sin 1, \quad \psi_4(0) = \cos 1. \end{aligned} \right.$$

Пример 3. Задача Коши для полной системы.

Рассмотрим уравнение:

$$\frac{\partial x_i}{\partial t_j} = a_{i,j} \sin \varphi_1(x, x_2, x_3) + b_{i,j} \cos \varphi_1(x, x_2, x_3)$$

и начальные условия $x_i(0) = 0$, $i \in [1 : m]$, $j \in [1 : 3]$, ($a_{i,j} = a_{i,j}(x, \dots, x_m)$, $b_{i,j} = b_{i,j}(x, \dots, x_m)$ – алгебраические полиномы).

Введем дополнительные переменные:

$$\psi_1 = \varphi_1(x, x_2, x_3), \quad \psi_2 = \varphi_2(x, x_2, x_3), \quad \psi_3 = \sin \varphi_1(x, x_2, x_3), \quad \psi_4 = \cos \varphi_1(x, x_2, x_3)$$

и получим их полные производные по t в силу этих уравнений:

$$\begin{aligned} \frac{d\psi_1}{dt_j} &= \sum_{k=1}^3 \frac{\partial \varphi_1}{\partial x_k} \frac{\partial x_k}{\partial t_j} = -\sum_{k=1}^3 \varphi_1^k \varphi_2 \frac{\partial x_k}{\partial t_j} = -\sum_{k=1}^3 \psi_1^k \psi_2 \frac{\partial x_k}{\partial t_j}, \\ \frac{d\psi_2}{dt_j} &= \sum_{k=1}^3 \frac{\partial \varphi_2}{\partial x_k} \frac{\partial x_k}{\partial t_j} = \sum_{k=1}^3 \left((42\varphi_1^5 + 6x_3\varphi_1 + 2x_2) \varphi_1^k \varphi_2^3 - k\varphi_1^{k-1} \varphi_2^2 \right) \frac{\partial x_k}{\partial t_j} = \\ &= \sum_{k=1}^3 \left((42\psi_1^5 + 6x_3\psi_1 + 2x_2) \psi_1^k \psi_2^3 - k\psi_1^{k-1} \psi_2^2 \right) \frac{\partial x_k}{\partial t_j}, \\ \frac{d\psi_3}{dt_j} &= \cos \varphi_1(x_1, x_2, x_3) \frac{\partial \varphi_1(x_1, x_2, x_3)}{\partial t_j} = \psi_4 \frac{\partial \psi_1}{\partial t_j}, \\ \frac{d\psi_4}{dt_j} &= -\sin \varphi_1(x_1, x_2, x_3) \frac{\partial \varphi_1(x_1, x_2, x_3)}{\partial t_j} = -\psi_3 \frac{\partial \psi_1}{\partial t_j}. \end{aligned}$$

Тогда можно записать исходную систему и начальные условия в форме полиномиальной задачи Коши:

$$\left\{ \begin{array}{l} \frac{\partial x_i}{\partial t_j} = a_{i,j}\psi_3 + b_{i,j}\varphi_4, \quad i \in [1 : m], \quad j \in [1 : 3], \quad \frac{\partial \psi_1}{\partial t_j} = - \sum_{k=1}^3 \psi_1^k \psi_2 \cdot \frac{\partial x_k}{\partial t_j}, \\ \frac{\partial \psi_2}{\partial t_j} = \sum_{k=1}^3 \left((42\psi_1^5 + 6x_3\psi_1 + 2x_2)\psi_1^k \psi_2^3 - k\psi_1^{k-1}\psi_2^2 \right) \frac{\partial x_k}{\partial t_j}, \\ \frac{\partial \psi_3}{\partial t_j} = \psi_4 \frac{\partial \psi_1}{\partial t_j}, \quad \frac{\partial \psi_4}{\partial t_j} = -\psi_3 \frac{\partial \psi_1}{\partial t_j}, \\ x_i(0, 0, 0) = 0, \quad i \in [1 : m], \quad \psi_1(0, 0, 0) = -1, \quad \psi_2(0, 0, 0) = \frac{1}{7}, \\ \psi_3(0, 0, 0) = -\sin 1, \quad \psi_4(0, 0, 0) = \cos 1. \end{array} \right.$$

Пример 4. Математический маятник.

Пусть $x_1 = x$, $x_2 = \dot{x}$, тогда уравнение математического маятника

$$\ddot{x} + k^2 \sin x = 0$$

запишем в виде системы обыкновенных дифференциальных уравнений

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -k^2 \sin x_1.$$

Вводя дополнительные переменные $x_3 = \sin x_1$, $x_4 = \cos x_1$, получаем полиномиальную (квадратичную) систему:

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -k^2 x_3, \quad \dot{x}_3 = x_2 x_4, \quad \dot{x}_4 = -x_2 x_3.$$

Пример 5. Вращательное движение спутника.

Рассмотрим движение спутника вокруг своего центра масс в предположении, что сам центр масс движется по круговой орбите с угловой скоростью ω . Обычно движение спутника вокруг своего центра масс описывается шестью фазовыми переменными, пусть это будут x_1, \dots, x_6 . При учете основных возмущающих факторов оказывается, как правило, что уравнения относительно этих переменных имеют вид:

$$\dot{x}_j = P_j(x_1, \dots, x_6, \sin x_1, \dots, \sin x_6, \cos x_1, \dots, \cos x_6, \sin \omega t, \cos \omega t),$$

причем P_j - полиномы по всем своим аргументам. Пусть $x_7 = \sin x_1, \dots$, $x_{13} = \cos x_1, \dots$, а также $x_{19} = \sin \omega t$, $x_{20} = \cos \omega t$, получаем полиномиальную систему:

$$\left\{ \begin{array}{l} \dot{x}_j = P_j(x_1, \dots, x_{20}), \quad \dot{x}_{6+j} = x_{13+j} P_j(x_1, \dots, x_{20}), \\ \dot{x}_{6+j} = -x_{6+j} P_j(x_1, \dots, x_{20}), \quad j = 1, \dots, 6, \quad \dot{x}_{19} = \omega x_{20}, \quad \dot{x}_{20} = -\omega x_{19}. \end{array} \right.$$

Глава 2. Схемы и быстрое вычисление систем мономов многих переменных

При написании данной главы использовались многие источники: [3, 8, 22, 30, 35].

Данная глава состоит из трех разделов. В первом разделе описаны основные определения, введенные для формулировки проблемы построения схемы и её алгоритма.

Во втором разделе представлены алгоритмы построения схем для произвольного набора мономов: до третьей степени и выше.

В третьей разделе представлены примеры построения схем для уравнений Пенлеве и задачи N тел в различных полиномиальных формах.

2.1 Основные определения

Введем необходимые определения и обозначения.

Моном – функция, определяемая формулой:

$$x^i = \prod_{k=1}^n x_k^{i_k} = x_1^{i_1} \cdot \dots \cdot x_n^{i_n}, \quad i_k \geq 0, \quad k \in [1, n].$$

Набор $T = (x^{i(1)}, \dots, x^{i(n)}, x^{i(n+1)}, \dots, x^{i(M)})$ мономов рассмотрим как упорядоченный при условии, что

$$2 \leq |i(n+1)| \leq |i(n+2)| \leq \dots \leq |i(M)| \leq L,$$

и обсудим задачу вычисления мономов $x^{i(n+1)}, \dots, x^{i(M)}$ набора T в предположении, что известны первые n его мономов, то есть мономы

$$x^{i(1)} = x_1, \dots, x^{i(n)} = x_n.$$

Вычислить T (то есть мономы $x^{i(n+1)}, \dots, x^{i(M)}$ в T) можно всегда, так как каждый из мономов можно получить, перемножив соответствующее число раз исходные величины x_1, \dots, x_n , например, $x_1^2 x_2^3 x_3 = x_1 x_1 x_2 x_2 x_2 x_3$. Ясно, что это не самый эффективный способ вычисления T . Вычисляя мономы последовательно слева направо в T , можно для получения очередного монома использовать в

качестве сомножителей уже вычисленные ранее мономы, если таковые есть. Например, $x_1^2 x_2^3 x_3$ можно вычислить как $(x_1^2 x_2^3) x_3$ или как $x_1^2 (x_2^3 x_3)$, если моном $x_1^2 x_2^3$ или мономы $x_2^3 x_3$ и x_1^2 ранее были вычислены.

Если очередной моном $x^{i(r)}$ в наборе можно вычислить как произведение ранее вычисленных мономов $x^{i(p)}$, $x^{i(q)}$, то это означает, что $i(r) = i(p) + i(q)$, причем $1 \leq p < r$, $1 \leq q < r$. Если при $|i(r)| \geq 3$ все мономы $x^{i(r)}$ можно вычислить таким образом последовательно (мономы в T упорядочены), то можно ввести в рассмотрение **схему** – упорядоченный набор

$$S(T) = ((p(n+1), q(n+1)), \dots, (p(M), q(M)))$$

из $M - n$ пар натуральных чисел $((p(r), q(r)))$, удовлетворяющих условию $r > p(r), q(r)$ для любых $r \in [n+1, M]$.

Набор T может не иметь схемы, а может иметь одну или несколько схем. Например, набор $\{x, x^2, x^4, x^7\}$ не имеет схемы, а наборы $\{x, x^2, x^4, x^5, x^7\}$, $\{x, x^2, x^3, x^4, x^7\}$ имеют одну и две схемы соответственно. Если T имеет схему, ее можно использовать при вычислении $T : x^{i(n+1)}, \dots, x^{i(M)}$ вычисляются слева направо в этом порядке, причем очередной моном $x^{i(r)}$ вычисляется как произведение уже вычисленных $x^{i(p)}$ и $x^{i(q)}$. Если набор T не имеет схемы, то его всегда можно дополнить до набора T' , имеющего схему. Это очевидно, так как любой набор T мономов степени не выше L можно дополнить до набора T' , содержащего все мономы степени не выше $L - 1$.

Пусть T, T' – упорядоченные наборы мономов. Будем писать $T \subset T'$ или $T' \supset T$ в том случае, когда все мономы набора T содержатся и в T' . Если $T' \supset T$ и T' имеет схему, то будем называть T' **оболочкой** для $T : T' = \text{span}(T)$. Набор T' с минимальным количеством добавленных мономов будем называть **минимальной оболочкой** и обозначать $T' = \text{mspan}(T)$. Если мы хотим вычислять T на основе некоторой схемы, а T не имеет схемы, то следует вычислять не T , а его оболочку T' , причем чем меньше дополнительных мономов в T' , тем быстрее, вообще говоря, будет процедура вычисления набора T . В то же время для данного набора T могут существовать различные оболочки с одинаковым количеством мономов (например, у набора $\{x, x^2, x^4, x^7\}$ есть оболочки $\{x, x^2, x^3, x^4, x^7\}$, $\{x, x^2, x^4, x^5, x^7\}$, $\{x, x^2, x^4, x^6, x^7\}$).

Исходный набор $T^0 = (x^{i(1)}, \dots, x^{i(n)}, \dots, x^{i(n+1)}, \dots, x^{i(M^0)})$ можно задать соответствующим набором индексов $I^0 = (i(1), \dots, i(n), \dots, i(n+1), \dots, i(M^0))$. Набор мономов, индексов и количество элементов оболочки обозначим

T^h, I^h, M^h . Эти же величины на текущем шаге алгоритма обозначим T, I, M (в начале и в конце алгоритма они равны T^0, I^0, M^0 и T^h, I^h, M^h).

Неравенства для индексов означают, что они выполнены для всех компонентов индексов, например, $i(\mathbf{v}) = (i_1(\mathbf{v}), \dots, i_n(\mathbf{v})) \geq 0$ означает, что $i_j(\mathbf{v}) \geq 0$, $j = 1, \dots, n$. Если $i - k \geq 0$ ($i - k > 0$), то будем говорить, что i содержит (строго содержит) k , или что k содержится (строго содержится) в i (будем говорить еще, что k является подиндексом i , а i – надиндексом k).

Величину $s = |i| = i_1 + \dots + i_n$ будем называть **степенью индекса** i , причем при $s = 1, 2, 3, \dots, m$ будем говорить о линейном, квадратичном, кубическом индексе и m – индексе (или индексе степени m).

Нелинейный индекс $i(v) \in I^0$ будем называть **согласованным** или **несогласованным** в наборе I^0 , если он представляется или нет суммой двух индексов из I^0 . Все квадратичные индексы i согласованы. Исходный набор индексов $I^0 = (i(1), \dots, i(n), \dots, i(n+1), \dots, i(M^0))$ разделим на два набора, $I_+^0 = (i(1), \dots, i(n), \dots, i(n+M_+))$, $I_-^0 = (i(n+r), \dots, i(n), \dots, i(n+R))$, первый из которых состоит из линейных и согласованных нелинейных индексов, а второй – из несогласованных индексов. Предполагается, что в наборах порядок следования индексов тот же, что у них был в I^0 . Очевидно, что целые r, R, M_+ должны удовлетворять неравенствам

$$0 < r \leq R \leq M^0 - n, \quad n + 1 \leq M \leq M^0.$$

Для $i(k) \in I$, примем также и обозначение $i(s, \mu(s))$, где $s = |i(k)| = i_1(k) + \dots + i_n(k)$, а $\mu(s) = 1, 2, \dots, m(s)$ – порядковый номер индекса в множестве

$$I(s) = \{i(\mathbf{v}) \in I \mid |i(\mathbf{v})| = s\},$$

как и соответствующего монома в

$$T(s) = \{x^{i(\mathbf{v})} \in T \mid |i(\mathbf{v})| = s\}$$

(если $I(s)$ пустое множество, то полагаем $\mu(s) = m(s) = 0$).

2.2 Быстрое вычисление систем мономов многих переменных

Данный раздел разбит на две части. В первой части описан алгоритм решения проблемы построения минимальной оболочки для произвольного набора

мономов до третьей степени включительно. Во второй части описан алгоритм для вычисления оболочки для произвольного набора мономов произвольной степени.

2.2.1 Системы мономов до третьей степени

В данном разделе описана и решена проблема построения минимальной оболочки и схемы для набора мономов до третьей степени включительно. Материал взят из [3, 22, 26].

Формулировка проблемы: Задача состоит во введении меньшего количества таких квадратичных индексов (мономов второй степени), что любой несогласованный кубический индекс (несогласованный моном третьей степени) $i(3, \mu_1(3)), \dots, i(3, \mu_m(3))$ исходного набора содержит хотя бы один из них.

Каждый из кубических индексов содержит до трех квадратичных. Например, кубический индекс $(3, 0, 0, \dots, 0)$ содержит один квадратичный $(2, 0, 0, \dots, 0)$, индекс $(2, 1, 0, \dots, 0)$ содержит два $(2, 0, 0, \dots, 0)$, $(1, 1, 0, \dots, 0)$, а $(1, 1, 1, 0, \dots, 0)$ – три $(1, 1, 0, \dots, 0)$, $(1, 0, 1, \dots, 0)$ и $(0, 1, 1, \dots, 0)$.

Квадратичный подиндекс индекса k^r обозначим k_j^r , а их количество – $\sigma(r) \in [1 : 3]$. Все различные подиндексы k_j^r $j \in [1 : \sigma(r)]$, $r \in [1 : m]$ рассмотрим в виде упорядоченного набора i^1, \dots, i^l , а каждый подиндекс k_j^r запишем как $i^{\nu(r, j)}$. Сопоставив каждой величине i^v переменную d_v , которая может принимать два значения – 0 и 1, приходим к задаче:

при известных $\sigma(r) \in [1 : 3]$, $\nu(i, j) \in [1 : l]$

найти целые d_1, \dots, d_l , удовлетворяющие неравенствам:

$$\sum_{j=1}^{\sigma(r)} d_{\nu(i, j)} \geq 1, \quad 0 \leq d_{\nu} \leq 1, \quad \nu \in [1 : l], \quad r \in [1 : m],$$

и минимизировать сумму $L(d_1, \dots, d_l) = d_1 + \dots + d_l$.

Решение проблемы минимизации количества квадратичных подиндексов сведена к задаче бинарного линейного программирования. Решив полученную задачу бинарного программирования, получим оптимальное решение сформулированной проблемы.

2.2.2 Системы мономов третьей степени и выше

В данном разделе описана и решена проблема построения оболочки и схемы для набора мономов произвольной степени. Материал взят из [43].

Формулировка проблемы: Рассмотрим набор T мономов до степени s включительно. Задача состоит во введении, по возможности, меньшего количества дополнительных мономов для построения оболочки $\text{span}(T)$ и схемы $S(T)$.

Рассмотрим упорядоченный набор мономов T_s степени s из исходного набора $T : T_s \subseteq T$ и рассмотрим проблему: найти минимальный набор мономов $D(T_s)$, состоящий из мономов $s - 1$ степени, такой, что любой моном из набора T_s можно вычислить умножением монома из набора $D(T_s)$ и одного монома первой степени.

Введем рекурсивные обозначения:

$$T^{(L-1)} = D(T_s) \cup T_{L-1}, \quad T^{(L-2)} = D(T^{(L-1)}) \cup T_{L-2}, \quad \dots, \quad T^3 = D(T^2) \cup T_2,$$

мы будем использовать следующую эвристическую формулу:

$$\text{span}(T) = T^{(2)} \cup \dots \cup T^{(L-1)} \cup T_L.$$

Кроме того, будем использовать обозначения, введенные выше, и обозначения:

$m(s) = \text{card } T_s$ – количество индексов в T_s ;

$k^{(r,s)} (r \in [1 : m(s)])$ – r – ый индекс в T_s ;

$G(r, s) \in [1 : s]$ – количество подиндексов $s - 1$ степени в индексе $k^{(r,s)}$;

$k_j^{(r,s)} (r \in [1 : m], j \in [1 : G(r, s)], G(r, s) \in [1 : s])$ – j – ый подиндекс $s - 1$ степени индекса $k^{(r,s)}$;

$l(s) = \text{card } \bigcup_{r=1}^{m(s)} \bigcup_{j=1}^{G(r,s)} k_j^{(r,s)}$ – количество различных подиндексов $s - 1$ степени для всех индексов $k^{(r,s)}$;

i^1, \dots, i^l – упорядоченный набор $\text{card } \bigcup_{r=1}^{m(s)} \bigcup_{j=1}^{G(r,s)} k_j^{(r,s)}$;

$v(r, s, j)$ – такая функция, которая $k_j^{(r,s)} = i^{v(r,s,j)}$;

Связывая с каждой переменной i^v бинарную переменную d_v ($d_v = 0$ или $d_v = 1$), возникает следующая проблема бинарного линейного программирования:

при известных $m(s)$, $l(s)$, и $\sigma(r, s) \in [1 : s]$, $v(r, s, j) \in [1 : l(s)]$

минимизировать $L(d_1, \dots, d_{l(s)}) = d_1 + \dots + d_{l(s)}$

при ограничениях: $0 \leq d_v \leq 1$, $v \in [1 : l(s)]$, $\sum_{1 \leq j \leq G(r,s)} d_{v(r,s,j)} \geq 1$.

Для решения таких задач использовалась соответствующая программа на Wolfram Mathematica (Приложение А).

2.3 Примеры построения схем

В данном разделе рассмотрены шесть уравнений Пенлеве в полиномиальной форме, оболочки для данных полиномиальных систем и соответствующие схемы для этих оболочек. Рассмотрим шесть уравнений Пенлеве в полиномиальной форме и построим для них оболочку и схему, соответствующую этой оболочке.

Первое уравнение Пенлеве:

$$\begin{aligned}x'_1 &= x_2, \\x'_2 &= 6x_1^2 + x_3, \\x'_3 &= 1.\end{aligned}$$

Набор мономов, стоящих в правой части дифференциальных уравнений, состоит из мономов первой и второй степени, поэтому схема строится автоматически.

Набор мономов: $T(P_I) = \{x_1, x_2, x_3, x_1^2\}$, Схема: $S(P_I) = \{(1, 1)\}$.

Второе уравнение Пенлеве:

$$\begin{aligned}x'_1 &= x_2, \\x'_2 &= 2x_1^3 + x_1x_3 + \alpha, \\x'_3 &= 1.\end{aligned}$$

Набор мономов: $T(P_{II}) = \{x_1, x_2, x_3, x_1^3, x_1x_3\}$.

Добавим один моном $\{x_1^2\}$, получим оболочку:

$$T'(P_{II}) = \{x_1, x_2, x_3, x_1^2, x_1^3, x_1x_3\}.$$

Схема для полученной оболочки:

$$S(P_{II}) = \{(1, 1), (1, 4), (1, 3)\}.$$

Третье уравнение Пенлеве:

$$\begin{aligned}x_1' &= x_2, \\x_2' &= x_2^2 x_3 - x_2 x_4 + \alpha x_1^2 x_4 + \beta x_4 + \gamma x_1^3 + \delta x_3, \\x_3' &= -x_2 x_3^2, \\x_4' &= -x_4^2.\end{aligned}$$

Набор мономов:

$$T(P_{III}) = \{x_1, x_2, x_3, x_4, x_2 x_4, x_4^2, x_1^2 x_4, x_2^2 x_3, x_2 x_3^2, x_1^3\}.$$

Добавим два монома $\{x_2 x_3, x_1^2\}$, получим оболочку:

$$T'(P_{III}) = \{x_1, x_2, x_3, x_4, x_2 x_3, x_2 x_4, x_1^2, x_4^2, x_1^2 x_4, x_2^2 x_3, x_2 x_3^2, x_1^3\}.$$

Схема для полученной оболочки:

$$S(P_{III}) = \{(2, 3), (2, 4), (1, 1), (4, 4), (4, 7), (2, 5), (3, 5), (1, 7)\}.$$

Четвёртое уравнение Пенлеве:

$$\begin{aligned}x_1' &= x_2, \\x_2' &= \frac{1}{2} x_2^2 x_3 - \frac{3}{2} x_1^3 + 4x_1^2 x_4 + 2x_1 x_4^2 - 2\alpha x_1 + \beta x_3, \\x_3' &= -x_2 x_3^2, \\x_4' &= 1.\end{aligned}$$

Набор мономов:

$$T(P_{IV}) = \{x_1, x_2, x_3, x_4, x_1^2 x_4, x_1 x_4^2, x_2 x_3^2, x_2^2 x_3, x_1^3\}.$$

Добавим три монома $\{x_1 x_4, x_2 x_3, x_1^2\}$, получим оболочку:

$$T'(P_{IV}) = \{x_1, x_2, x_3, x_4, x_1 x_4, x_2 x_3, x_1^2, x_1^2 x_4, x_1 x_4^2, x_2 x_3^2, x_2^2 x_3, x_1^3\}.$$

Схема для полученной оболочки:

$$S(P_{IV}) = \{(1, 4), (2, 3), (1, 1), (1, 5), (4, 5), (3, 6), (2, 6), (1, 7)\}.$$

Пятое уравнение Пенлеве:

$$\begin{aligned}x_1' &= x_2, \\x_2' &= \frac{1}{2} x_2^2 x_3 + x_2^2 x_4 - x_2 x_5 + \alpha x_1^3 x_5^2 + \beta x_1^2 x_3 x_5^2 - 2\alpha x_1^2 x_5^2 - \\&\quad - 2\beta x_1 x_3 x_5^2 + \alpha x_1 x_5^2 + \beta x_3 x_5^2 + \gamma x_1 x_4 + \gamma x_1 x_4, \\x_3' &= -x_2 x_3^2, \\x_4' &= -x_2 x_4^2, \quad x_5' = -x_5^2.\end{aligned}$$

Набор мономов:

$$T(P_V) = \{x_1, x_2, x_3, x_4, x_5, x_1x_4, x_1x_5, x_2x_5, x_1^2x_4, x_2^2x_3, x_5^2, \\ x_1^2x_4, x_2^2x_3, x_2^2x_4, x_2x_3^2, x_2x_4^2, x_1x_5^2, x_3x_5^2, x_1^2x_5^2, x_1x_3x_5^2, x_1^3x_5^2, x_1^2x_3x_5^2\}.$$

Добавим три монома $\{x_2x_3, x_2x_4\}$, получим оболочку:

$$T'(P_V) = \{x_1, x_2, x_3, x_4, x_5, x_1x_4, x_1x_5, x_2x_5, x_2x_3, x_2x_4, x_1^2x_4, x_2^2x_3, x_5^2, \\ x_1^2x_4, x_2^2x_3, x_2^2x_4, x_2x_3^2, x_2x_4^2, x_1x_5^2, x_3x_5^2, x_1^2x_5^2, x_1x_3x_5^2, x_1^3x_5^2, x_1^2x_3x_5^2\}.$$

Схема для полученной оболочки:

$$S(P_V) = \{(1, 4), (1, 5), (2, 5), (2, 3), (2, 4), (1, 6), (2, 9), (5, 5), (1, 6), \\ (2, 9), (2, 10), (3, 9), (4, 10), (5, 7), (3, 13), (7, 7), (3, 19), (1, 21), (3, 21)\}.$$

Шестое уравнение Пенлеве:

$$\begin{aligned} x'_1 &= x_2, \\ x'_2 &= \frac{1}{2}x_2^2x_3 + \frac{1}{2}x_2^2x_4 + \frac{1}{2}x_2^2x_5 - x_2x_5 - x_2x_6 - x_2x_7 + \\ &+ \alpha x_1(x_1 - 1)(x_1 - x_8)x_6^2x_7^2 + \beta(x_1 - 1)(x_1 - x_8)x_3x_6x_7^2 + \\ &+ \gamma x_1(x_1 - x_8)x_4x_6^2x_7 + \delta x_1(x_1 - 1)x_6x_6x_7, \\ x'_3 &= -x_2x_3^2, \quad x'_4 = -x_2x_4^2, \quad x'_5 = -x_5^2(x_2 - 1), \\ x'_6 &= -x_6^2, \quad x'_7 = -x_7^2, \quad x'_8 = 1. \end{aligned} \tag{2.1}$$

Набор мономов:

$$T(P_{VI}) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_2x_5, x_2x_6, x_2x_7, x_5^2, x_6^2, x_7^2, \\ x_2^2x_3, x_2^2x_4, x_2^2x_5, x_2x_3^2, x_2x_4^2, x_2x_5^2, x_1x_5x_6x_7, x_1x_3x_6x_7^2, x_3x_6x_7^2x_8, \\ x_1^2x_5x_6x_7, x_1^2x_6^2x_7^2, x_1x_6^2x_7^2x_8, x_1^2x_3x_6x_7^2, x_1^2x_3x_6x_7^2x_8, x_1^2x_4x_6^2x_7, \\ x_1x_4x_6^2x_7x_8, x_1^3x_6^2x_7^2, x_1^2x_6^2x_7^2x_8\}.$$

Добавим десять мономов:

$$\{x_1x_5, x_1x_7, x_2x_3, x_2x_4, x_3x_7, x_6x_7, x_6^2x_7, x_3x_6x_7^2, x_1x_6^2x_7^2, x_1x_4x_6^2x_7\},$$

получим оболочку:

$$T'(P_{VI}) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_1x_5, x_1x_7, x_2x_3, x_2x_4, x_3x_7,$$

$$\begin{aligned}
& x_6x_7, x_2x_5, x_2x_6, x_2x_7, x_5^2, x_6^2, x_7^2, x_6^2x_7, x_2^2x_3, \\
& x_2^2x_4, x_2^2x_5, x_2x_3^2, x_2x_4^2, x_2x_5^2, x_3x_6x_7^2, x_1x_6^2x_7^2, x_1x_5x_6x_7, x_1x_3x_6x_7^2, \\
& x_3x_6x_7^2x_8, x_1x_4x_6^2x_7, x_1^2x_5x_6x_7, x_1^2x_6^2x_7^2, x_1x_6^2x_7^2x_8, x_1^2x_3x_6x_7^2, x_1^2x_3x_6x_7^2x_8, \\
& x_1^2x_4x_6^2x_7, x_1x_4x_6^2x_7x_8, x_1^3x_6^2x_7^2, x_1^2x_6^2x_7^2x_8\}.
\end{aligned}$$

Схема для полученной оболочки:

$$\begin{aligned}
S(P_{VI}) = \{ & (1, 5), (1, 7), (2, 3), (2, 4), (3, 7), (6, 7), (2, 5), (2, 6), (2, 7), (5, 5), \\
& (6, 6), (7, 7), (2, 11), (2, 12), (2, 13), (3, 11), (4, 12), (5, 13), (6, 18), \\
& (10, 18), (17, 18), (1, 10), (8, 30), (1, 29), (15, 28), (9, 28), (1, 34), \\
& (8, 34), (1, 31), (8, 31), (1, 35), (8, 35)\}.
\end{aligned}$$

Глава 3. Методы рядов Тейлора

При написании данной главы использовались источники: [4, 5, 8, 98].

Настоящая глава состоит из двух разделов.

В первом разделе представлено полное описание метода рядов Тейлора в полиномиальной форме. Выведены формулы подсчета коэффициентов Тейлора, сформулирован метод рядов Тейлора для полиномиальной системы, представлены оценки локальной погрешности для линейной и нелинейной задачи. Описаны вспомогательные алгоритмы, основываясь на которых, строится сам алгоритм метода рядов Тейлора в полиномиальной форме. В последнем подразделе второго раздела (3.2.5) алгоритм представлен в операторной форме.

Во втором разделе описана структура реализации представленного метода (из четвертого разделе данной главы) с описанием основных функций и подпрограмм.

3.1 Метод рядов Тейлора для полиномиальных систем

При написании данного раздела использовались источники: [8, 10].

В данном разделе рассматриваются три формы задания полиномиальной задачи Коши, формулы для коэффициентов Тейлора и различные оценки локальных решений.

3.1.1 Коэффициенты Тейлора

Рассмотрим автономную задачу Коши

$$\frac{dx}{dt} = f(x), \quad x(t_0) = x_0, \quad (3.1)$$

где $f = (f_1, \dots, f_n)$, $x = (x_1, \dots, x_n)$, $x_0 = (x_{10}, \dots, x_{n0}) \in R^n$, $t, t_0 \in R$. Решение ее обозначим $x(t, t_0, x_0)$, $x(t)$, или x . Полиномиальной называют систему (3.24) (и задачу Коши), в которой все функции f_i – алгебраические полиномы по

x_1, \dots, x_n . Далее используются две формы задания полиномиальной системы.

Первая форма представления полиномиальной задачи Коши имеет вид:

$$\frac{dx}{dt} = a + \sum_{m=1}^{L+1} \sum_{i \in I(m)} a[i]x^i, \quad x(t_0) = x_0, \quad (3.2)$$

где

$i = (i_1, \dots, i_n)$, $i_1, \dots, i_n \in Z$, $L \in [0 : \infty)$, $I(m) = \{i \in Z^n \mid i_1, \dots, i_n \geq 0, |i| = m\}$,
 $|i| = i_1 + \dots + i_n$, $x = (x_1, \dots, x_n) \in R^n$, $x^i = x_1^{i_1} \cdot \dots \cdot x_n^{i_n}$, $x_0 = (x_{10}, \dots, x_{n0}) =$
 $= (x_1(t_0), \dots, x_n(t_0)) \in R^n$, $a = (a_1, \dots, a_n) \in R^n$, $a[i] = (a_1[i], \dots, a_n[i])$.

Вторая форма представления задачи (3.24) следующая:

$$\frac{dx_j}{dt} = \sum_{k=0}^u a_{j,k} x^{i(k)}, \quad x_j(t_0) = x_{j0}, \quad j \in [1 : n], \quad (3.3)$$

где $x^{i(0)} = 1, \dots, x^{i(n)} = x_n$, а $x^{i(n+1)}, \dots, x^{i(u)}$ – все различные нелинейные мономы в правых частях уравнений (3.25).

Третья форма – иерархическая, она представляется формулами:

$$\frac{dx_j}{dt} = \sum_{k=0}^{u(1)} a_{s,j,k} y_1^{i(1,k)}, \quad j \in [1 : n], \quad (3.4)$$

$$y_r^{i(r,j)} = \sum_{k=0}^{u(r+1)} a_{r,j,k} y_{r+1}^{i(r+1,k)}, \quad j \in [1 : m(r)], \quad r \in [1 : s-1], \quad (3.5)$$

в которых использованы обозначения:

$y_r = (y_{r,1}, \dots, y_{r,m(r)})$, $r \in [1 : s]$, $y_s = x$, т.е. $y_{s,1} = x_1, \dots, y_{s,m(s)} = x_n$, $m(s) = n$;

$y_r^{i(r,0)} = 1$, $y_r^{i(r,1)} = y_{r,1}$, ..., $y_r^{i(r,m(s))} = y_{r,m(s)}$, $r \in [1 : s]$;

при каждом $r \in [1 : s]$ величины $y_r^{i(r,m(s)+1)}, \dots, y_r^{i(r,u(r))}$ – различные нелинейные мономы по $y_{r,1}, \dots, y_{r,m(r)}$ (в частности, при $r = s$ это мономы по x_1, \dots, x_n).

В формулах (3.27), (3.28) величина s – любое фиксированное натуральное число. Будем называть его числом уровней третьей формы. При $s = 1$ третья форма совпадает со второй. При $s \in [2 : \infty)$ третья форма является обобщением второй. Вместе с тем простым перемножением полиномов третью форму всегда можно свести к форме с меньшим числом уровней и в конце концов – ко второй. Необходимость в третьей форме может возникнуть по разным причинам, но важно, что к такой форме полиномиальной системы сводятся системы ОДУ

в результате применения к ним метода дополнительных переменных в самом общем случае в автоматизированном режиме.

Дополним, если необходимо, набор мономов T (мономы стоящие в правой части дифференциального уравнения) до оболочки T' со схемой

$$S(T) = ((p(n+1), q(n+1)), \dots, (p(u), q(u)))$$

и обратимся к уравнениям (3.26) с тем, чтобы получить искомые формулы для коэффициентов Тейлора. Подставив

$$x^{i(k)} = \sum_{p=0}^{\infty} x_{k,p} (t - t_0)^p, \quad k \in [0 : u], \quad (3.6)$$

в равенство

$$x^{i(k)} = x^{i(p(k))} \cdot x^{i(q(k))}, \quad k \in [n+1, u], \quad (3.7)$$

и в уравнения (3.26), находим

$$\sum_{p=0}^{\infty} [x_{k,p} - \sum_{l=0}^p x_{p(k),l} x_{q(k),p-l}] (t - t_0)^p = 0, \quad (3.8)$$

$$\sum_{p=0}^{\infty} [(p+1)x_{j,p+1} - \sum_{k=0}^u a_{j,k} x_{k,p}] (t - t_0)^p = 0. \quad (3.9)$$

Приравнявая нулю коэффициенты при $(t - t_0)^p$ в равенствах (3.31), (3.32) и вспоминая начальные условия $x(t_0) = x_0$, получаем расчетные формулы

$$x_{k,0} = x_k(t_0), \quad k \in [1 : n], \quad (3.10)$$

$$x_{k,p} = \sum_{l=0}^p x_{p(k),l} x_{q(k),p-l}, \quad k \in [n+1 : u], \quad (p = 0, 1, \dots); \quad (3.11)$$

$$x_{k,p+1} = (p+1)^{-1} \sum_{l=0}^u a_{k,l} x_{l,p}, \quad k \in [1 : n], \quad (p = 0, 1, \dots);$$

Формулы (3.33), (3.34) позволяют вычислять последовательно все коэффициенты $x_{k,p}$ разложений (3.29) величин $x_1, \dots, x_n, x_i(n+1), \dots, x_i(u)$ в ряд Тейлора. Для того чтобы использовать эти формулы, достаточно иметь в своем распоряжении:

- начальные данные: $x_k(t_0)$ при $k \in [1 : n]$,
- коэффициенты: $a_{k,l}$ при $k \in [1 : n]$, $l \in [0 : u]$,

– соответствующую схему: $S(T) = ((p(n+1), q(n+1)), \dots, (p(u), q(u)))$. Важно, что теми же величинами задается и сама полиномиальная задача Коши, и такой способ задания может быть предпочтительным не только при ее пошаговом интегрировании (методом рядов Тейлора, методом Рунге - Кутты и т. п.), но и при решении других задач, связанных с дифференциальными уравнениями.

3.1.2 Формулировка метода рядов Тейлора

В данном подразделе рассматривается формулировка метода рядов Тейлора и используемые в его реализации оценки радиуса сходимости и остаточного члена для линейной и нелинейной задач.

Введем обозначения:

$$x^{(k)} = \frac{\partial^k x}{\partial t^k}, \quad x_0^{(k)} = x^{(k)}(t_0), \quad |x| = \max_{i \in [1:n]} |x_i|, \quad O_\rho(t_0) = \{t \in C \mid |t - t_0| < \rho\},$$

$$T_M x(t, t_0, x_0) = \sum_{m=0}^M x_0^{(m)} \frac{(t - t_0)^m}{m!}, \quad \delta T_M x(t, t_0, x_0) = x(t, t_0, x_0) - T_M x(t, t_0, x_0),$$

где T_M и δT_M – операторы, которые решению $x(t, t_0, x_0)$ задачи (3.24) сопоставляют полином Тейлора $T_M x(t, t_0, x_0)$ и остаточный член $\delta T_M x(t, t_0, x_0)$ соответственно. Радиус сходимости ряда $T_\infty x(t, t_0, x_0)$ обозначим через $R(t_0, x_0)$.

Метод рядов Тейлора решения задачи Коши (3.29) заключается в построении таблицы приближенных значений $\tilde{x}_w = \tilde{x}(t_w)$ по формулам

$$\tilde{x}_1 = T_{M_1} x(t_1, t_0, x_0), \dots, \tilde{x}_w = T_{M_w} x(t_w, t_{w-1}, x_{w-1}), \dots,$$

где M_1, M_2, \dots – натуральные числа, $t_1 = t_0 + h_1$, $t_2 = t_1 + h_2, \dots$, h_1, h_2, \dots удовлетворяют неравенствам

$$|h_w| < R(t_{w-1}, \tilde{x}_{w-1}).$$

Вычисление каждого значения $\tilde{x}(t_w)$ называют шагом метода, а число h_w – величиной шага.

3.1.3 Оценки локальной погрешности

В данном подразделе рассматриваются оценки локальной погрешности, чтобы автоматизировать выбор порядка M_k и величины шага h_k , можно воспользоваться оценками величин $R(t_0, x_0)$ и $\delta T_M x(t, t_0, x_0)$.

Оценки для линейной задачи.

Рассмотрим задачу Коши

$$\begin{aligned} \frac{dx}{dt} &= a + Ax, \quad x(t_0) = x_0, \quad x = (x_1, \dots, x_n) \in R^n, \\ x_0 &= (x_{10}, \dots, x_{n0}) \in R^n, \quad a = (a_1, \dots, a_n) \in R^n, \quad A = (a_{i,j}), \quad t, t_0, a_{i,j} \in R. \end{aligned} \quad (3.12)$$

Введем замену $x_j = \alpha_j y_j$, $j \in [1 : n]$, зависящую от "масштабирующих множителей" $\alpha = (\alpha_1, \dots, \alpha_n)$, $\alpha_1, \dots, \alpha_n > 0$, и обозначения

$$\begin{aligned} \rho(\alpha) &= \frac{1}{s(\alpha)}, \quad s(\alpha) = \max_{i \in [1:n]} s_i(\alpha), \quad s_i(\alpha) = \alpha_i^{-1} \sum_{j=1}^n \alpha_j |a_{i,j}|, \quad |b| = \max_{i \in [1:n]} \alpha_i^{-1} |\alpha_i|, \\ |y_0| &= \max_{i \in [1:n]} \alpha_i^{-1} |x_{i0}|, \quad T_M e^\tau = \sum_{m=0}^M \frac{\tau^m}{m!}, \quad u(\tau) = \delta T_M e^\tau = e^\tau - T_M e^\tau. \end{aligned} \quad (3.13)$$

Утверждение 1. Решение $x(t, t_0, x_0)$ задачи (3.25) удовлетворяет неравенству

$$|\delta T_M x_i(t, t_0, x_0)| \leq \alpha_i (|y_0| + |b| \rho) \delta T_M e^{\frac{|t-t_0|}{\rho}}, \quad i \in [1 : n], \quad t \in C.$$

Утверждение 2. Пусть u^{-1} – функция, обратная u при $\tau > 0$, χ_1, \dots, χ_n , $\varepsilon_1, \dots, \varepsilon_n$ – положительные числа и

$$\varepsilon = (|y_0| + \rho |b|)^{-1} \min_{i \in [1:n]} \frac{\varepsilon_i \chi_i}{\alpha_i}.$$

Тогда

$$(|t - t_0| \geq \rho u^{-1}(\varepsilon)) \Rightarrow (|\delta T_M x_i(t, t_0, x_0)| \leq \varepsilon_i \chi_i, \quad i \in [1 : n]).$$

Оценки для нелинейной задачи.

В задаче (3.25) положим, что $x_j = \alpha_j y_j$, $j \in [1 : n]$, $\alpha = (\alpha_1, \dots, \alpha_n)$, $\alpha_1, \dots, \alpha_n > 0$, и предположим дополнительно, что x_{j0} и α удовлетворяют неравенствам $0 < |x_{j0}| < \alpha_j$, $j \in [1 : n]$. Введем обозначения:

$$\rho(\alpha) = \frac{1}{Ls(\alpha)}, \quad s(\alpha) = \max_{j \in [1:n]} s_j(\alpha), \quad s_j(\alpha) = \alpha_j^{-1} \left(|a_j| + \sum_{m=1}^{L+1} \sum_{|i|=m} \alpha^i |a_j[i]| \right),$$

$$O_\rho(t_0) = \{t \in C \mid |t - t_0| < \rho\}, \quad b(\tau) = (1 - \tau)^{\frac{-1}{L}}, \quad T_M b(\tau) = \sum_{m=0}^M \prod_{l=0}^{m-1} \frac{(\frac{1}{L} + l)\tau^m}{m!}, \quad (3.14)$$

$$v(\tau) = \delta T_M b(\tau) = b(\tau) - T_M b(\tau), \quad \tau \in [0, 1).$$

Утверждение 3. Решение $x(t, t_0, x_0)$ задачи (3.25) регулярно в круге $O_\rho(t_0)$ и удовлетворяет там неравенству

$$|\delta T_M x_j(t, t_0, x_0)| \leq \alpha_j \delta T_M b\left(\frac{|t - t_0|}{\rho}\right).$$

Утверждение 4. Если функция v^{-1} обратна v , а

$$\varepsilon = \min_{i \in [1:n]} \left(\frac{\varepsilon_i \chi_i}{\alpha_i} \right), \quad \varepsilon_i > 0,$$

то

$$(|t - t_0| \geq \rho v^{-1}(\varepsilon)) \Rightarrow (|\delta T_M x_i(t, t_0, x_0)| \leq \varepsilon_i \chi_i, \quad i \in [1 : n]).$$

Далее будем использовать эти утверждения при $\varepsilon_1 = \varepsilon_2 = \dots = \varepsilon_n$, $\chi_1 = \alpha_1, \dots, \chi_n = \alpha_n$.

3.1.4 Вспомогательные алгоритмы

В основе реализации МРТ для полиномиальных ОДУ лежат формулы (3.33) и (3.34) для коэффициентов Тейлора и алгоритмы автоматического выбора величины шага и порядка, опирающиеся как на утверждения 1 – 4, так и на обычно применяемые в численном анализе эвристические соображения. Мы изложим вначале некоторые вспомогательные алгоритмы, потом опирающиеся на них алгоритмы автоматического выбора величины шага и порядка, а затем алгоритм интегрирования полиномиальной системы ОДУ на заданном промежутке.

Далее величины $t = t_k$, $x = x^k = \tilde{x}(t_k)$, $h = h_k = t_{k+1} - t_k$ будут обозначать текущий узел, приближенное значение решения в нем и величину текущего шага, а ε, Δ – заданные пользователем допустимые относительную и абсолютную погрешности решения на шаге. Кроме того, при фиксированных M, K, t_k, x^k величины $T_M x(t_{k+1}, t_k, x^k)$ и

$$T_M x(t_{k+1}, t_k, x^k), \quad \delta T_{M,K} x(t_{k+1}, t_k, x^k) = T_{M+K} x(t_{k+1}, t_k, x^k) - T_M x(t_{k+1}, t_k, x^k),$$

$$\varepsilon_{M,K}(t_{k+1}, t_k, x^k) = |\delta T_{M+K}x(t_{k+1}, t_k, x^k)| \left(|T_m x(t_{k+1}, t_k, x^k)| + \Delta \right)^{-1}$$

будут записаны упрощенно как $T(h)$, $\delta T(h)$, $\varepsilon(h)$.

1. Таблицы для вычисления значений функций u^{-1} и v^{-1} (утверждения 2 и 4). Таблицы содержатся в [124] и используются в алгоритме 2 (см. ниже).

Для каждой пары $L = 0, \dots, 99$, $M = 1, \dots, 99$ таблицы содержат набор значений $(u(\tau_j), v(\tau_j))$, упорядоченный по возрастанию $\tau_j = 0.01, 0.02, \dots, 0.99$. Они вычислены при помощи программы Wolfram Mathematica [126] по формулам

$$u(\tau_j) = e^{\tau_j} - T_M e^{\tau_j}, \quad v(\tau_j) = b(\tau_j) - T_M b(\tau_j).$$

Из равенств $u^{-1}(u(\tau_j)) = \tau_j$ и $v^{-1}(v(\tau_j)) = \tau_j$ следует, что значение функций u^{-1}, v^{-1} в промежуточных точках можно получить интерполированием по значениям в ближайших узловых точках $(u(\tau_j), v(\tau_j))$.

2. Априорный выбор шага. Величины x_0 (значение решения в узле, для которого выбирается шаг h), ε, Δ (допустимые относительная и абсолютная погрешности) считаются заданными. Вначале вычисляются масштабирующие множители α_i , $i \in [1 : n]$ по процедуре *tsmr_alp*, если она задана пользователем, иначе при $i \in [1 : n]$

$$\text{если } \max_{j \in [1:n]} |x_j(t_0)| = 0, \text{ то } \alpha_j = 1, \text{ иначе } \alpha_j = \max_{j \in [1:n]} (|x_j(t_0)|).$$

Затем вычисляется величина τ :

для нелинейной системы:

$$\tau = u^{-1} \left(\varepsilon \min_{i \in [1:n]} \frac{|x_i(t_0)| + \Delta}{\alpha_i} \right),$$

для линейной системы:

$$\tau = v^{-1} \left(\varepsilon \min_{i \in [1:n]} \frac{|x_i(t_0)| + \Delta}{\alpha_i} \right).$$

3. Итеративный алгоритм коррекции шага. Введем следующие обозначения: D – натуральное число, например, $D = 5$, $d = \frac{h_0}{D}$, $s = \text{sgn}(\varepsilon - \varepsilon(h_0))$,

$$\text{если } \text{sgn}(\varepsilon - \varepsilon(h)) > 0, \text{ то } \sigma(i, h) = i - 1, \text{ иначе } \sigma(i, h) = i,$$

где h_0 – шаг, который требуется скорректировать так, чтобы получить максимально возможную его величину при заданной пользователем допустимой

относительной погрешности ε .

Алгоритм в операторной форме:

- 1) $i := 1$;
- 2) $h := h_0 + sid$. Если $sgn(\varepsilon - \varepsilon(h)) < 0$, то $h := h_0 + s\sigma(i, h)d$ и переход к 5;
- 3) если $i = D - 1$, то $h_0 := h$, $d := \frac{h_0}{D}$, и переход к 1;
- 4) $i := i + 1$ и переход к 2;
- 5) выход.

4. Стандартный алгоритм коррекции шага. Он основан на апостериорной информации. Скорректированная величина h для текущего узла при известном значении $x = (x_1, \dots, x_n)$ в нем и известном приближении h_0 вычисляется как

$$h = h_0 \left(\sqrt{n^{-1} \sum_{i=1}^n \left((\delta T(h_0))^2 \cdot (\Delta + \varepsilon \max(|x_i|, |T_i(h_0)|)) \right)} \right)^{1/(M+1)},$$

которая аналогична используемым в методах Рунге - Кутты (M – порядок МРТ).

5. Градуировка. Она заключается в том, что для каждого порядка $p \in [M_{min} : M_{max}]$ (где, например, $M_{min} = 5$, $M_{max} = 60$) определяется процессорное время $t(p)$ вычисления всего набора коэффициентов Тейлора решения. Градуировка производится на начальном этапе работы программы до начала интегрирования системы на заданном промежутке, а ее результаты используются для выбора порядка и величины шага на первом шаге интегрирования и для корректировки порядка на следующих шагах.

6. Выбор величины шага и порядка на первом шаге. Начальное приближение h_0 для величины первого шага вычисляется по алгоритму 2. Затем при $p \in [M_{min} : M_{max}]$ вычисляются величины $V(p) = \frac{h(p)}{t(p)}$ – ”пошаговые скорости”, где $h(p)$ – величина шага, полученная МРТ порядка p с помощью алгоритма 3 и с использованием начального приближения h_0 . В качестве порядка на первом шаге выбирается величина M такая, что

$$V(M) = \max_{p \in [M_{min}, M_{max}]} V(p),$$

а в качестве величины этого шага принимается $h = h(M)$.

(а) Алгоритм автоматического выбора шага. На каждом шаге (кроме первого) за начальное приближение h_0 принимается значение, полученное

на предыдущем шаге интегрирования. Затем с помощью алгоритмов 2 и 4 вычисляются две величины шага h_1 и h_2 . Из них выбирается максимальная — $h = \max\{h_1, h_2\}$, которая корректируется по алгоритму 3.

(б) Алгоритм автоматического выбора порядка. Если на очередном шаге МРТ величина шага h изменилась в m (например, $m = 5$) раз по сравнению с величиной H (H изменяется после каждого изменения величины порядка M , а на первом шаге полагается равной величине первого шага), то производится корректировка порядка M . Предполагается, что известны коэффициенты Тейлора до порядка M . Для $p = M - 1, \dots, M_{min}$ последовательно вычисляются длины шагов $h(p)$ и $V(p) = \frac{h(p)}{t(p)}$. Как только (и если) окажется, что $V(p) \leq V(M)$, новый порядок M полагается равным этому p . Если же не найдется таких $p = M - 1, \dots, M_{min}$, то для $p = M + 1, \dots, M_{max}$ последовательно вычисляются коэффициенты Тейлора порядка p (с учетом того, что коэффициенты порядка до p уже известны), соответствующий шаг $h(p)$ и скорость $V(p)$. Как только (и если) окажется, что $V(p) > V(M)$, новый порядок M полагается равным этому p . Если же не найдется таких $p = M + 1, \dots, M_{max}$, то использовавшийся порядок не корректируется.

3.1.5 Общий алгоритм метода рядов Тейлора

Ниже этот алгоритм представлен в операторной форме.

1. Выбор величины шага и порядка на первом шаге (см. алгоритм б) и присвоение этих значений переменным H и M (см. (б)).
2. Вычисление в следующей точке t_k коэффициентов Тейлора по формулам (3.29), (3.30) и шага h (см. (а)).
3. Если шаг h изменился более чем в m раз по сравнению с величиной H , то вычисляется новый порядок M (см. (б)) и соответствующий шаг h (см. (а)).
4. Если $t_k + h < T$, то $t_{k+1} = t_k + h$, вычисляется решение в точке t_{k+1} и переход к 2.
5. Вычисление решения в точке T (шаг: $h = T - t_k$), выход.

3.2 Реализация метода рядов Тейлора (МРТ)

Описанная ниже программа взята из [10, 124].

В данном разделе описана программа, реализованная на языке Фортран, которая численно интегрирует системы дифференциальных уравнений в полиномиальной форме (методом, описанным в главе 3).

Программа состоит из главной программы, подпрограммы чтения конфигурационного файла, подпрограммы интегрирования на промежутке, подпрограммы вычисления коэффициентов Тейлора, подпрограммы вычисления полиномов Тейлора, функции вычисления шага по априорному алгоритму, функции вычисления шага стандартной коррекцией, функции вычисления шага итеративной коррекцией, функции автоматического выбора шага, функции вычисления ρ , подпрограммы чтения файлов с данными, функции вычисления степени правой части, функции вычисления u^{-1} и v^{-1} , процедуры замера времени расчета коэффициентов, функции вычисления $\alpha_1, \dots, \alpha_n$, процедуры вычисления оптимального порядка.

Главная программа.

1. Считывает имя конфигурационного файла из командной строки.
2. Считывает данные из конфигурационного файла.
 - *cfg.dat* – задаются: требуемая относительная погрешность, требуемая абсолютная погрешность и пути к остальным файлам конфигурации;
 - *coef.dat* – набор ненулевых коэффициентов системы. Коэффициент задается тройкой (коэффициент, номер уравнения, номер монома);
 - *sch.dat* – схема: последовательность пар чисел. Если система линейна, схема не задается;
 - *ic.dat* – начальные данные для задачи Коши;
 - *points.dat* – последовательность значений аргумента, в которых выдаются значения решения.
3. Запускает интегрирование на промежутке.

Программа чтения конфигурационного файла.

Аргументы: *cfg* - имя конфигурационного файла *ipar* :

1 - размерность,

2 - число мономов,

3 - число ненулевых коэффициентов,

4 - число точек вывода;

rpar - требуемая погрешность:

1 - относительная, 2 - абсолютная; *ln* - линейность;

fpar - файлы:

1. - с начальными данными,

2. - со схемой,

3. - файлы с коэффициентами,

4. - с таблицей,

5. - для записи результатов,

6. - с точками вывода.

Возвращает: *ipar*, *rpar*, *ln*, *fpar*.

Подпрограмма интегрирования на промежутке.

Аргументы:

n - размерность; *u* - число мономов; *na* - число ненулевых коэффициентов; *nr*

- число точек для выдачи решения; *ln* - линейность; *rtol* - требуемая относи-

тельная погрешность; *atol* - требуемая абсолютная погрешность; *fpar* - файлы:

1 - с начальными данными, 2 - со схемой, 3 - файл с коэффициентами, 4 - с

таблицей, 5 - для записи результатов, 6 - с точками вывода.

Возвращает: значение *x*.

Подпрограмма вычисления коэффициентов Тейлора.

Аргументы:

x - массив коэффициентов Тейлора; *tx* - массив значений решения в точке;

pl, *pu* - нижний и верхний порядок.

Возвращает: значение *x*.

Глобальные переменные: *n*, *u*, *na*, *sch*, *a*, *ia*, *ja*, *ptax*.

Подпрограмма вычисления полиномов Тейлора.

Аргументы:

tx - массив значений решения в точке; x - массив коэффициентов Тейлора; h - длина шага; pl , pu - нижний и верхний порядок.

Возвращает: tx .

Глобальные переменные: n , u , $pmax$.

Функция вычисления шага по априорному алгоритму.

Аргументы: tx - массив значений решения в точке; $alp - \alpha_1, \dots, \alpha_n$; $pwork$ - порядок.

Глобальные переменные: n .

Функция вычисления шага стандартной коррекцией.

Аргументы: tx - массив значений решения в точке; x - массив коэффициентов Тейлора; $h0$ - приближение к шагу; rx , tx - вспомогательные массивы; $pwork$ - порядок.

Глобальные переменные: n , $pmax$, $atol$, $rtol$.

Функция вычисления шага итеративной коррекцией.

Аргументы: tx - массив значений решения в точке; x - массив коэффициентов Тейлора; $h0$ - приближение к шагу; rx , tx - вспомогательные массивы; $pwork$ - порядок.

Глобальные переменные: n , $pmax$, $atol$, $rtol$.

Функция автоматического выбора шага.

Аргументы: h - приближение к шагу; tx - массив значений решения в точке; x - массив коэффициентов Тейлора; rx , tx - вспомогательные массивы; $alp - \alpha_1, \dots, \alpha_n$; $pwork$ - порядок; dir - направление.

Глобальные переменные: n , u .

Функция вычисления ρ по $\alpha_1, \dots, \alpha_n$.

Аргументы: $alp - \alpha_1, \dots, \alpha_n$.

Глобальные переменные: n , u , na , ut , aa , ia , ja , $rdeg$.

Программа чтения файлов с данными.

Аргументы: tx - значение решения в точке; $fpar$ - файлы: 1 - с начальными данными, 2 - со схемой, 3 - файл с коэффициентами, 4 - с таблицей, 5 - для записи результатов, 6 - с точками вывода.

Возвращает: tx , sch , a , ia , ja , $rdeg$, ox

Глобальные переменные: n , u , na , sch , ox , np , a , ia , ja , $rdeg$.

Функция вычисления степени правой части.

Глобальные переменные: n , u , sch .

Функция вычисления u^{-1} и v^{-1} .

Аргументы: alp — $-\alpha_1, \dots, \alpha_n$; $pwork$ - текущий порядок метода.

Глобальные переменные: n , $atol$, $rtol$, vtb , vtn , $pmin$, $pmax$.

Процедура замера времени расчета коэффициентов

Аргументы: x - массив коэффициентов Тейлора; tx - массив значений решения в точке.

Возвращает: $time$.

Глобальные переменные: n , u , $pmax$, $time$.

Функция вычисления $\alpha_1, \dots, \alpha_n$

Аргументы: alp — $-\alpha_1, \dots, \alpha_n$; tx - массив значений решения в точке; $rpar$, $ipar$ - вспомогательные массивы.

Глобальные переменные: n .

Процедура вычисления оптимального порядка.

Аргументы: x - массив коэффициентов Тейлора; tx - массив значений решения в точке.

Возвращает: $pwork$, h .

Глобальные переменные: n , u , $pmax$, $time$.

Глава 4. Численные эксперименты

При написании данной главы использовались многие источники: [26, 30, 35, 43].

В данной главе рассматривается серия экспериментов, показывающая эффективность схем и метода рядов Тейлора для полиномиальных систем, использующего схем как инструмент подсчёта коэффициентов Тейлора.

4.1 Произвольный набор мономов

В данном эксперименте случайным образом генерируются системы из N_m мономов из N_v переменных. Учитывая систему мономов, можно вычислить $\tau = t/t_s$ (эффективность схемы), где t_s, t – время, необходимое для вычисления всех мономов сгенерированной системы с помощью схемы и без неё соответственно. Учитывая N_m, N_v для сотни систем случайно выбранных мономов, мы можем вычислить минимальные и максимальные значения $\tau : \tau_{min}, \tau_{max}$. Результаты эксперимента представлены в Таблице 1.

Таблица 1 — Рандомный набор мономов

$N_m = 1000$			$N_v = 200$		
N_v	τ_{min}	τ_{max}	N_m	τ_{min}	τ_{max}
50	1.6	1.7	1 000	4.5	4.7
100	2.6	2.8	2 000	4.9	5.2
150	3.5	3.8	3 000	5.1	5.3
200	4.5	4.7	4 000	5.3	5.8
250	5.2	5.5	5 000	5.6	5.9
300	6.3	6.7	6 000	5.8	6.2
350	7.5	8.2	7 000	5.6	6.0
400	8.6	9.7	8 000	5.7	6.6
450	9.9	11.0	9 000	5.7	6.9
500	11.2	13.5	10 000	5.8	7.0

4.2 Задача N тел

В [30] рассмотрена система дифференциальных уравнений классической задачи N тел с использованием декартовых координат. Затем эта система дифференциальных уравнений сведена к трем различным системам уравнений с правыми частями, которые являются многочленами пятой, четвертой и третьей степени соответственно.

Рассмотрим два численных эксперимента (Таблица 2, Таблица 3). Они связаны со сравнением стоимости оценки при вычислении правых частей уравнений исходной системы и систем одночленов в правых частях полиномиальных уравнений пятой, четвертой и третьей степени (с помощью схем и без них).

Здесь используются следующие обозначения:

N – количество тел;

t_0 – время, необходимое для вычисления всех мономов из правых частей исходных уравнений задачи N тел;

t_5, t_4, t_3 – время, необходимое для вычисления всех различных мономов в правых частях систем пятой, четвертой и третьей степени соответственно;

$t_{j,s}, t_j$ – время, необходимое для вычисления всех мономов степени j в правых частях системы пятой ($j = 5$), четвертой ($j = 4$) и третьей степени ($j = 3$) соответственно (с помощью схемы и без неё);

$t_{j,s}^+, t_j^+$ – время, необходимое для вычисления дополнительных кубических мономов (возмущений) и всех одночленов степени j в правых частях системы пятой ($j = 5$), четвертой ($j = 4$) и третьей степени ($j = 3$) соответственно (с помощью схемы и без нее).

Таблица 2 — Мономы задачи N тел без возмущений

N	t_0/t_5	$t_0/t_{5,s}$	t_0/t_4	$t_0/t_{4,s}$	t_0/t_3	$t_0/t_{3,s}$
3	7.2	8.6	4.2	5.0	4.2	5.5
4	9.6	15.4	4.7	7.0	4.6	7.8
5	9.7	20.4	4.8	11.5	4.7	14.1
6	9.8	22.5	4.9	14.2	4.8	16.3
7	10.2	25.5	5.2	17.7	5.1	19.3
8	12.1	38.7	5.3	20.1	5.3	22.8
9	13.0	49.9	5.6	23.0	5.5	25.8
10	13.1	57.6	5.8	26.6	5.7	29.6

Таблица 3 — Мономы задачи N тел с возмущениями

N	$t_5/t_{5,s}$	$t_5^+/t_{5,s}^+$	$t_4/t_{4,s}$	$t_4^+/t_{4,s}^+$	$t_3/t_{3,s}$	$t_3^+/t_{3,s}^+$
3	1.2	1.2	1.2	1.1	1.3	1.2
4	1.6	1.5	1.5	1.5	1.7	1.6
5	2.1	2.0	2.4	2.3	3.0	2.7
6	2.3	2.1	2.9	2.7	3.4	3.1
7	2.6	2.2	3.4	3.0	3.8	3.3
8	3.2	2.6	3.8	3.3	4.3	3.6
9	3.8	3.0	4.1	3.5	4.7	3.8
10	4.4	3.2	4.6	3.8	5.2	4.1

Заключение

Основные результаты работы заключаются в следующем:

1. Разработаны детали построения схем,
2. Реализованы алгоритмы построения схем для произвольного набора мономов,
3. Разработаны компьютерные программы, реализующие алгоритмы построения схем,
4. Проведены численные эксперименты, исследована эффективность алгоритмов и программ, разработанных для численного интегрирования дифференциальных уравнений.

В заключение автор выражает благодарность и большую признательность научному руководителю Бабаджанянцу Л. К. за поддержку, помощь, обсуждение результатов и научное руководство. Автор также благодарит всех, кто сделал настоящую работу возможной.

Список литературы

- [1] *Абалакин В., Аксенов Е., Гребеников Е., Демин В., Рябов Ю.* Справочное руководство по небесной механике и астродинамике // М.: Наука, 1976.
- [2] *Алесова И., Бабаджанянц Л., Потоцкая И., Саакян А.* Оптимальное управление нелинейными колебаниями спутника на эллиптической орбите // Устойчивость и колебания нелинейных систем управления. 2016. С. 14 - 17.
- [3] *Алесова И., Бабаджанянц Л., Потоцкая И., Саакян А.* Оптимизация пошагового интегрирования дифференциальных уравнений динамики // Устойчивость и колебания нелинейных систем управления. 2016. С. 17 - 19.
- [4] *Алферов Г., Бабаджанянц Л., Ковригин Д., Сенатова С.* Лабораторный практикум по механике управляемого движения. // Учебное пособие. Издательство ЛГУ. 1989.
- [5] *Арушанян О., Залеткин С.* Численное решение обыкновенных дифференциальных уравнений на Фортране // М.: Издательство Московского университета, 1990.
- [6] *Бабаджанянц Л.* Продолжаемость и представление решений в задачах небесной механики // Труды ИТА АН СССР, Вып. 17. 1978. С. 3 – 45.
- [7] *Бабаджанянц Л., Чекашкин Ю.* Аналитический метод вычисления возмущений в координатах планет // Вестник Ленингр. ун-та. Сер. 1: Математика, механика, астрономия. Вып. 3. 1990. С. 101 – 107.
- [8] *Бабаджанянц Л.* Метод рядов Тейлора // Вестник Санкт-Петербургского университета. Сер. 10. № 3. 2010. С. 13 – 29.
- [9] *Бабаджанянц Л.* Метод дополнительных переменных // Вестник Санкт-Петербургского университета. Сер. 10. № 1. 2010. С. 3 – 11.
- [10] *Бабаджанянц Л., Большаков А.* Реализация метода рядов Тейлора для решения обыкновенных дифференциальных уравнений // Вычислительные

методы и программирование. Научно-исследовательский вычислительный центр МГУ им. М.В. Ломоносова. Т. 13. 2012. С. 497 – 510.

- [11] *Бабаджанянц Л., Брэгман К.* Алгоритм метода дополнительных переменных // Вестник Санкт-Петербургского университета. Сер. 10. № 2. 2012. С. 3 – 12.
- [12] *Бабаджанянц Л., Мгоян П.* Оценка голоморфных решений обыкновенных дифференциальных уравнений // Известия АН Арм. ССР. Серия "Математика". Том XVII. №2. 1982. С. 83 – 91.
- [13] *Бабаджанянц Л., Мгоян П.* Оценка голоморфных решений обыкновенных дифференциальных уравнений // Вестник ЛГУ. 1984. № 7.
- [14] *Банди Б.* Основы линейного программирования // Москва "Радио и Связь". 1989. С. 17 – 20.
- [15] *Беллман Р.* Процессы регулирования с адаптацией // пер. с англ. Ю. П. Леонова; под ред. А. М. Летова. М.: Наука, 1964. 360 с.
- [16] *Беллман Р.* Введение в теорию матриц // пер. с англ.; под ред. В. Б. Лидского. М.: Наука. 1969. 368 с.
- [17] *Брумберг В.* Ряды полиномов в задаче трех тел // Бюл. Ин-та теор. астрономии АН СССР. Т. 9, №4., 1963, С. 234 – 256.
- [18] *Брэгман К.* Математические модели возмущенного движения в центральных полях // Диссертация. 2014.
- [19] *Гантмахер Ф.* Теория матриц // Наука. 1988.
- [20] *Мерман Г.* О представлении общего решения задачи трех тел сходящимися рядами // Бюл. Ин-та теор. астрономии АН СССР. Т. 6, № 10., 1958, С. 713 – 769.
- [21] *Пуанкаре А.* О кривых, определяемых дифференциальными уравнениями // Под ред. А. Андропова, М.: Гостехиздат. 1947. 392 с.
- [22] *Саакян А.* Ускорение численного интегрирования уравнений динамики при помощи схем // Процессы управления и устойчивость. Том 3. Номер 1. 2016.

- [23] *Чернышева Н.* Метод вычисления возмущений в поле вращающегося тела // Вестник Ленинградского университета. Сер. 1: Математика, механика, астрономия. Вып. 4. 1987, С. 83 – 89.
- [24] *Abad A., Barrio R., Blesa F., Rodriguez M.* Breaking the limits: the Taylor series method // Applied Mathematics and Computation. 217, №20. 2011. pp. 7940 – 7954.
- [25] *Abad A., Barrio R., Blesa F., Rodriguez M.* Algorithm 924: TIDES, a Taylor Series Integrator for Differential EquationS // ACM Transactions on Mathematical Software. 2012.
- [26] *Alesova I., Babadzanjan L., Pototskaya I., Pupysheva Y., Saakyan A.* Taylor Series Method of Numerical Integration of the N-body problem // AIP Conference Proceeding. Volume 1863. Issue 1. 2017.
- [27] *Alesova I., Babadzanjan L., Pototskaya I., Pupysheva Y., Saakyan A.* Piecewise Polynomial Control in Mechanical Systems // AIP Conference Proceeding. Volume 1863. Issue 1. 2017.
- [28] *Alesova I., Babadzanjan L., Pototskaya I., Pupysheva Y., Saakyan A.* Fuel/Time Optimal Control of Satellite Oscillations // AIP Conference Proceeding. Volume 1863. Issue 1. 2017.
- [29] *Alesova I., Babadzanjan L., Pototskaya I., Pupysheva Y., Saakyan A.* Control of Mechanical Systems by the Mixed "Time and Expenditure" Criterion // AIP Conference Proceeding. Volume 1959. Issue 1. 2018.
- [30] *Alesova I., Babadzanjan L., Pototskaya I., Pupysheva Y., Saakyan A.* High-Precision Numerical Integration of Equations in Dynamics // AIP Conference Proceeding. Volume 1959. Issue 1. 2018.
- [31] *Alesova I., Babadzanjan L., Pototskaya I., Pupysheva Y., Saakyan A.* Optimal Control of Parametric Oscillations of Compressed Flexible Bars // AIP Conference Proceeding. Volume 1959. Issue 1. 2018.
- [32] *Alesova I., Babadzanjan L., Bregman A., Bregman K., Pototskaya I., Pupysheva Y., Saakyan A.* Piecewise Constant Control of Linear Mechanical

- Systems in the General Case // AIP Conference Proceeding. Volume 1978. Issue 1. 2018.
- [33] *Alesova I., Babadzanjanz L., Bregman A., Bregman K., Pototskaya I., Pupysheva Y., Saakyan A.* Perturbations of Calculation Technique for Central Fields // AIP Conference Proceeding. Volume 1978. Issue 1. 2018.
- [34] *Alesova I., Babadzanjanz L., Bregman A., Bregman K., Pototskaya I., Pupysheva Y., Saakyan A.* Control of Satellite Aerodynamic Oscillations // AIP Conference Proceeding. Volume 1978. Issue 1. 2018.
- [35] *Alesova I., Babadzanjanz L., Bregman A., Bregman K., Pototskaya I., Pupysheva Y., Saakyan A.* Schemes of Fast Evaluation of Multivariate Monomials for Speeding Up Numerical Integration of Equations in Dynamics // AIP Conference Proceeding. Volume 1978. Issue 1. 2018.
- [36] *Alesova I., Babadzanjanz L., Pototskaya I., Saakyan A.* Fuel Optimal Control of Non-Linear Oscillations of a Satellite on Elliptical Orbit // International Conference Stability and Oscillations of Nonlinear Control Systems. 2016.
- [37] *Andrade R., Rauh A.* The Lorenz model and the method of Carleman embedding // Physics Letters. Vol. 82A. 1981. pp. 276 – 278.
- [38] *Azamed Yehuala G.* Qualitative Models of Neural Activity and the Carleman Embedding Technique // East Tennessee State University M.S thesis. 2009. pp. 11 – 17.
- [39] *Babadzanjanz L.* Existence of the Continuations in the N-body problem // Celestial Mechanics. 20, 1979. pp. 43 – 57.
- [40] *Babadzanjanz L.* Error estimates for numerical integration of the N-body problem // American Institute of Physics (Sov.Astron.Lett.7(6) Nov.-dec.-1981), 1982. pp. 416 – 418.
- [41] *Babadzanjanz L.* On the global solution of N-body problem // Celestial Mechanics. 56. 1993. pp. 427 – 449.
- [42] *Babadzanjanz L., Sarkissian D* Taylor series method for dynamic systems with control: convergence and error estimates // Journal of Mathematical Sciences. Springer New York, Vol. 139. № 6. 2006. pp. 7025 – 7046.

- [43] *Babadzanjanz L., Pototskaya I., Pupysheva Y., Saakyan A.* Fast evaluation of multivariate monomials for speeding up numerical integration in space dynamics // International Multidisciplinary Scientific GeoConference Surveying Geology and Mining Ecology Management. Volume 19. 2019. pp. 647 - 654.
- [44] *Babadzanjanz L., Pototskaya I., Pupysheva Y., Saakyan A.* Quality-cost optimal control in Lotka-Volterra populations model // International Multidisciplinary Scientific GeoConference Surveying Geology and Mining Ecology Management. Volume 19. 2019. pp. 3 - 10.
- [45] *Bank B., Giusti M., Heintz J., Safey El Din.* Intrinsic complexity estimates in polynomial optimization // Journal of Complexity, 2014.
- [46] *Bates D., Hauenstein J., Sommese A., Wampler C.* Numerically solving polynomial systems with Bertini // Society for Industrial and Applied Mathematics, SIAM. Philadelphia. Software, Environments, and Tools, Vol. 25. 2015.
- [47] *Bellman R.* Introduction to matrix analysis // MCGRAW-HILL book company, inc. 1960.
- [48] *Bellman R.* Adaptive control processes: a guided tour // Princeton University Press. 1961.
- [49] *Bellman R., Richardson J.* On Some questions arising in the approximate solution of nonlinear differential equations // Quart. Math., Vol. 20. 1963. pp. 333 – 339.
- [50] *Bellman R.* Addition chains of vectors (advanced problem 5125) // American Mathematical Monthly Vol. 70. 1963.
- [51] *Berz M., Bischof C., Corliss G., Griewank A.* Computational differentiation: techniques, applications, and tools // Society for Industrial and Applied Mathematics, SIAM. Philadelphia. 1996.
- [52] *Berz M.* Cosy infinity version 8 reference manual // Technical Report MSUCL-1088. National Superconducting Cyclotron Lab., Michigan State University, East Lansing. Mich. 2003.

- [53] *Brauer A.* On addition chains // Bulletin of the American Mathematical Society. Vol.45. 1939. pp. 736 – 739.
- [54] *Brening L., Fairen V.* Analytic approach to initiate value problems in nonlinear systems // Journal of Mathematical Physics. Vol. 22. 1981. pp. 649 – 652.
- [55] *Broucke R.* Solution of the N-body problem with recurrent power series // Celestial Mechanics. № 4. 1971. pp. 110 – 115.
- [56] *Captain Brain Gaude/W.* Solving nonlinear Aeronautical problems using Carleman linearization method // Sandia National Laboratories. 2001.
- [57] *Carleman T.* Application de la theories des equations integrales lineaire aux systemes dequations differentielles nonlineaires // Acta mathematica, Vol. 59, 1963. pp. 63-87.
- [58] *Ceberio M., Kreinovich V.* Greedy Algorithms for Optimizing Multivariate Horner Schemes // ACM Sigsam Bull, 38, 2004. pp. 8 – 15.
- [59] *Corliss G., Chang Y.* Solving ordinary differential equations using Taylor series // ACM Transactions on Mathematical Software. 1982. pp. 114 – 144.
- [60] *Carothers D., Parker G., Sochacki J., Warne P.* Some properties of solutions to polynomial systems of differential equations // Electronic journal of differential equations. № 40. 2005. pp. 1 – 17.
- [61] *Chang Y., Corliss G.* ATOMFT: solving ODEs and DAEs using Taylor series // Computers and Mathematics with Applications. 1994. № 10 – 12. pp. 209 – 233.
- [62] *Charnyi V.* Two Methods of integrating the equations of motion // Cosmic Research. Vol. 8. № 5. 1970.
- [63] *Chen B., He S., Li Z., Zhang S.* Maximum block improvement and polynomial optimization // Society for Industrial and Applied Mathematics Journal on Optimization, SIOPT. 2012.
- [64] *Dekker K., Verwer J.* Stability of Runge - Kutta Methods for Stiff Nonlinear Differential Equations // Elsevier Science Publishers B. V., North-Holland, Amsterdam - New York - Oxford. 1984.

- [65] *Estes R., Lancaster E.* An algorithm for integrating stepwise the restricted problem in Thiele's coordinates // *Celestial Mechanics*. № 1. 1970. pp. 297 – 300.
- [66] *Gordon D.* A survey of fast exponentiation methods // *Journal of Algorithms* 27. 1998. pp. 129 – 146.
- [67] *Griewank A.* Evaluating derivatives // *Society for Industrial and Applied Mathematics, SIAM*. Philadelphia. 2000.
- [68] *Griffith J.* On a generalized Taylor scheme for numerical integration // *Astronomy and Astrophysics*. Vol. 8. № 2. 1970. pp. 267 – 272.
- [69] *Guckenheimer J., Holmes P.* *Nonlinear Oscillations; Dynamical Systems; and Bifurcations of Vector Fields* // Springer - Verlag. 1983.
- [70] *Hairer E., Norsett S., Wanner G.* *Solving Ordinary Differential Equations I. Nonstiff Problems* // Springer - Verlag, Berlin - Heildeberg. 1987.
- [71] *Hairer E., Wanner G., Norsett S.* *Solving ordinary differential equations: nonstiff problems* // Berlin: Springer. 2009.
- [72] *Hoefkens J., Berz M., Makino K* Computing validated solutions of implicit differential Equations // *Advances in Computational Mathematics* 19. 2003. pp. 231 – 253.
- [73] *Holmes H.* *Introduction to Perturbation Methods* // *Text in Applied Mathematics*, Springer - Verlag New York, Inc. 1998.
- [74] *Hoppensteadt F.* *Analysis and Simulation of Chaotic Systems* // *Second Edition*, Springer - Verlag, Inc. 1993.
- [75] *Horner W.* *Phil. Trans. Soc. London* (1819) 308 – 305. Reprinted in: D.E. Smith, *A Source Book in Mathematics* McGraw-Hill. 1959.
- [76] *Jorba A., Zou M* A software package for the numerical integration of ODEs by means of high-order Taylor methods // *Experimental Mathematics* 14, № 1. 2005. pp. 99 – 117.
- [77] *Kerner E.* Universal formats for ordinary differential systems // *Journal of Mathematical Physics*, Vol. 22. 1981. pp. 1366 – 1371.

- [78] *Knuth D., Papadimitriou C.* Duality in addition chains // Bulletin of the European Association for Theoretical Computer Science 13. 1981. pp. 2 – 4.
- [79] *Kochergin V.* On the complexity of computations of system of three monomials in three variables // Matematicheskie Voprosy Kibernetiki, vyp.15 , 2006. pp. 79 – 155.
- [80] *Kochergin V.* Correction of estimations of complexity of evaluation of a monomial and systems of monomials in Bellman's and Knuth's problems // Diskretnyi Analiz i Issl. Oper. Vol. 21. № 6. 2014. pp. 51 – 72.
- [81] *Kojima M.* Efficient evaluation of polynomials and their partial derivatives in continuation methods // Journal of the Operations Research Society of Japan. Vol. 51. 2008. pp. 29 – 54.
- [82] *Kowalski K.* Hilbert space description of classical dynamical systems // Physica., Vol. 145A. 1987. pp. 408 – 427.
- [83] *Kowalski K., Steeb W.* Nonlinear Dynamical Systems and Carleman linearization // World Scientist. 1991.
- [84] *Lara M., Elipe A., Palacios M.* Automatic programming of recurrent power series // Mathematics and Computers in Simulation. Vol. 49. 1999. pp. 351 – 362.
- [85] *Leiserson C., Liyun Li, Maza, Moreno M., Xie, Yuzhen.* Efficient Evaluation of Large Polynomials // In K. Fukuda et al. (Eds.): Mathematical Software – ICMS 2010, Lecture Notes in Computer Science, Vol. 6327. 2010. pp. 342 – 353.
- [86] *Levi M.* Qualitative analysis of the periodically forced relaxation oscillations // Memoirs of the American Mathematical Society. Vol. 32. № 244. 1981.
- [87] *Makino K., Berz M.* Taylor models and other validated functional inclusion methods // International Journal of Pure and Applied Mathematics. Vol. 6. № 3. 2003. pp. 239 – 316.
- [88] *Miletics E., Molnarka G.* Taylor series method with numerical derivatives for initial value problems // Journal of Computational Methods in Sciences and Engineering. Vol. 4. № 1 – 2. 204. pp. 105 – 114.

- [89] *Molnarka G., Miletics E.* Implicit extension of Taylor series method with numerical derivatives for initial value problems // Computers and Mathematics with Applications. Vol. 50. № 7. 2005. pp. 1167 – 1177.
- [90] *Montroll E., Helleman R.* On a nonlinear perurbation theory without secular terms // AIP Conference Proceedings. Vol. 17. 1976. pp. 75 – 111.
- [91] *Nedialkov N., Jackson K., Corliss G.* Validated solutions of initial value problems for ordinary differential equations // Applied Mathematics and Computation. Vol. 105. 1999. pp. 21 – 68.
- [92] *Nedialkov N., Pryce J.* Solving differential-algebraic equations by Taylor series. I. Computing Taylor coefficients // BIT. Vol. 45. № 3. 2005. pp. 561 – 591.
- [93] *Nedialkov N., Pryce J.* Solving differential-algebraic equations by Taylor series. II. Computing the system Jacobian // BIT. Vol. 47. № 1. 2007. pp. 121 – 135.
- [94] *Nedialkov N., Pryce J.* Solving differential algebraic equations by Taylor series. III. The DAETS code // Journal of Numerical Analysis, Industrial and Applied Mathematics. Vol. 3. № 1 – 2. 2008. pp. 61 – 80.
- [95] *Oesterwinter C., Cohen C.* New orbital elements for Moon and planets // Celestial Mechanics. Vol. 5. № 3. 1972. pp. 317 – 395.
- [96] *Okhotsimskii D., Sikharulidze Yu.* Foundations of Space Flight Mechanics // Science. Moscow. 1994.
- [97] *Pan V.* Some schemes for the evaluation of polynomials with real coefficients // Problems of Cybernetics. Pergamon Press 5. 1961. pp. 14 – 32.
- [98] *Parker G., Sochacki J.* Implementing the Picard iteration // Neural, Parallel and Scientific Computation. Vol. 4. 1996. pp. 97 – 112.
- [99] *Parker G., Sochacki J.* A Picard–McLaurin theorem for initial value PDE's // Abstract and Applied Analysis. 5. 2000. pp. 47 – 63.
- [100] *Paterson M., Stockmeyer L.* On the number of nonscalar multiplications necessary to evaluate polynomials // SIAM Journal on Computing. Vol. 2. 1973. pp. 60 – 66.

- [101] *Pena J., Sauer T.* On the multivariate Horner scheme // SIAM Journal on Numerical Analysis. Vol. 37, 2000. pp. 1186 – 1197.
- [102] *Pena J., Sauer T.* On the multivariate Horner scheme. II: Running error analysis // Computing. Vol. 65. 2000. pp. 313 – 322.
- [103] *Pippenger N.* On evaluation of powers and related problems // Proceedings 17th Annual IEEE Symposium on Foundations of Computer Science, Houston. 1976. pp. 258 – 263.
- [104] *Pippenger N.* On evaluation of powers and monomials // SIAM Journal on Computing. Vol. 9(2). 1980. pp. 230 – 250.
- [105] *Poincare H.* Sur les courbes definies par les equations differentielles (IV) // Journal de mathematiques pures et appliques 4e serie. Tome 2. 1886. pp. 151 – 218.
- [106] *Pruett C., Rudmin J., Lacy J.* An adaptive N-body algorithm of optimal order // Journal of Computational Physics. Vol. 187. 2003. pp. 298 – 317.
- [107] *Rall L.* Automatic differentiation: techniques and applications // Lecture Notes in Computer Science. Vol. 120. Berlin: Springer - Verlag. 1981.
- [108] *Rauch L.* Iterative solution of the N-body problem for real time // ARS Journal. № 30. 1960. pp. 284 – 286.
- [109] *Rauch L., Riddel W.* The iterative solution of the analytical N-body problem // Journal of the Society for Industrial and Applied Mathematics. № 8. 1960. pp. 568 – 581.
- [110] *Rodriguez M., Barrio R.* Reducing rounding errors and achieving Brouwer's law with Taylor series method // Applied Numerical Mathematics. Vol. 62. № 8. 2012. pp. 1014 – 1024.
- [111] *Simai He, Zhening Li., Zhang S.* Inhomogeneous polynomial optimization over a convex set: An approximation approach // Mathematics of Computation. Vol. 84. pp. 715 – 741.
- [112] *Steeb W., Wilhelm F.* Nonlinear autonomous system of differential equations and Carleman linearization procedure // Journal of Mathematical Analysis and Applications. Vol. 44. 1980. pp. 601 – 611.

- [113] *Steeb W.* A note on Carleman linearization // *Physics Letters*. Vol. 140A. 1989. pp. 336 – 338.
- [114] *Steffensen J.* On the restricted problem of three bodies // *Mat.-Fys. Medd. Danske, Videnskab. Selskab.* Vol. 30, № 18. 1956. pp. 75 – 83.
- [115] *Steffensen J.* On the problem of three bodies in the plane // *Mat.-Fys. Medd. Danske, Videnskab. Selskab.* Vol. 31. № 3. 1957. pp. 98 – 123.
- [116] *Tsiligiannis C., Lyberatos G.* Steady state bifurcations and exact multiplicity conditions via Carleman linearization // *Journal of Mathematical Analysis and Applications*. Vol. 126. 1987. pp. 143 – 160.
- [117] *Tsiligiannis C., Lyberatos G.* Normal forms, resonance and bifurcation analysis via the Carleman linearization // *Journal of Mathematical Analysis and Applications*. Elsevier. Vol. 139, 1989. pp. 123 – 138.
- [118] *Van Barel M., Humet M., Sorber L.* Approximating optimal point configurations for multivariate polynomial interpolation // *Electronic Transactions on Numerical Analysis*. Vol. 42. 2014. pp. 41 – 63.
- [119] *Van der Pol B.* A theory of Amplitude of free and forced triode vibrations // *Radio Review*. Vol. 1. 1920. pp. 701 – 754.
- [120] *Van der Pol B.* On relaxation oscillation // *Philosophical Magazine*. Vol. 2. 1962. pp. 978 – 992.
- [121] *Wong W.* Carleman transformation and Ovsyannikov Treves operators // *Nonlinear analysis*. Vol. 6. 1982. pp. 1296 – 1303.
- [122] *Xiao S., Zeng G.* Equality-constrained minimization of polynomial functions // *Science China Mathematics*. Vol. 58, Issue 10. 2015.
- [123] *Yao A C.* On the evaluation of powers // *SIAM Journal on Computing*. Vol. 5. 1976. pp. 100 – 103.
- [124] *Babadzanjanz L.* webpage URL: (TSMR) <http://www.apmath.spbu.ru/en/staff/babadzhanyants/index.html>
- [125] *Hairer E.* webpage URL: <http://www.unige.ch/~hairer/>

- [126] *Wolfram Mathematica*. webpage URL: <https://reference.wolfram.com/language/>
- [127] *NASA Jet Propulsion Laboratory*. webpage URL: ssd.jpl.nasa.gov/?constants

Список таблиц

1	Рандомный набор мономов	37
2	Мономы задачи N тел без возмущений	39
3	Мономы задачи N тел с возмущениями	39

Приложение А

Программа расчёта схемы

В данном разделе представлена программа *моно3* реализованная в Wolfram Mathematica, алгоритм которой описан в главе 2.

На вход программе подаётся набор мономов третьей степени. На выходе программа выдаёт схему для введенного набора мономов. В случае набора мономов выше третьей степени, необходимо рекуррентно вызвать основную часть (Листинг А.2) программы.

Листинг А.1 — Программа *моно3*: Ввод и обработка набора мономов.

```

# utilits
im[t_] := Times @@ ((x[#] &) /@ t);
mi[t_] := Flatten[t /. {Power -> ConstantArray, Times -> List}
  /. x -> Identity];

5 # input
m = DeleteDuplicates[{ x[1] x[2], x[1] x[3] x[4]}];

# preprocessing
monomial1 = im /@ Transpose[{Sort[DeleteDuplicates[
10 Catenate[mi /@ m]]]}];
monomial2 = Select[m, Length[mi[#]] == 2 &];
monomial3 = Select[m, Length[mi[#]] == 3 &];
m3 = DeleteCases[Complement[m, monomial2], t_ /;
  Length[DeleteDuplicates[Subsets[mi[t], {2}]] \[Intersection]
15 mi /@ monomial2] > 0];
m3Subsets = (DeleteDuplicates[Subsets[mi[#], {2}]] &) /@ m3;
m3AllSubsets = DeleteDuplicates@Catenate[m3Subsets];

```

Листинг А.2 — Программа *моно3*: Основная часть.

```

% linear programming part
% use this part recursively if the set consists of
% monomials higher than third degree
m3OptimalSubsets = If[m3AllSubsets == {}, {},
5   im /@ m3AllSubsets[[Transpose[Position[
   LinearProgramming[Table[1.0, {i, Length[m3AllSubsets]}],
   (ReplacePart[Table[0, {i, Length[m3AllSubsets]}],
   Transpose[{#}] -> 1] &) /@
   Table[FirstPosition[m3AllSubsets,
10   m3Subsets[[r, j]][[1]],
   {r, Length[m3]}, {j, Length[m3Subsets[[r]]}],
   Table[1, {i, Length[m3]}],
   Table[{0, 1}, {i, Length[m3AllSubsets]}],
   Integers], 1]][[1]]]
15 ];

```

Листинг А.3 — Программа *моно3*: Схема.

```

monomial230OptimalSubsets = Join[monomial2, m3OptimalSubsets];

Join[
  monomial2scheme =
5   Catch[Do[If[# == monomial1[[i]] monomial1[[j]], Throw[{i, j
   }]],
   {i, Length[monomial1]}, {j, Length[monomial1]}] & /@
   Select[Join[m, m3OptimalSubsets], Length[mi[#]] == 2 &],

  monomial3scheme =
10  Catch[Do[If[# == monomial1[[i]] monomial230OptimalSubsets[[j]],
   Throw[{i, Length[monomial1] + j}]],
   {i, Length[monomial1]}, {j, Length[monomial230OptimalSubsets
   ]}] & /@
   Select[m, Length[mi[#]] == 3 &]
]
15

```