Федеральный исследовательский центр
«Информатика и управление» Российской академии наук

На правах рукописи
УДК 517.929+519.688

Панфёров Антон Александрович

Алгоритмы символьных вычислений в системах компьютерной алгебры для линейных дифференциальных систем с выделенными неизвестными

Специальность 05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

ДИССЕРТАЦИЯ
на соискание ученой степени кандидата
физико-математических наук

Научный руководитель –
д.ф.-м.н., профессор
С. А. Абрамов

Москва – 2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Актуальность темы исследования. Компьютерная алгебра, называемая также символьными вычислениями, является разделом алгоритмической математики ([1]). Её алгоритмы предназначены в основном для решения с помощью компьютера задач, в которых исходные данные и результаты имеют вид математических выражений, формул. Выполнение каждого такого алгоритма состоит в проведении некоторых формульных выкладок.

Одной из ранних реализаций систем аналитических вычислений, обладающей общностью и эффективностью, была аппаратная реализация языка и системы АНАЛИТИК, созданной коллективом под руководством В. М. Глушкова [2] на малой машине МИР-2. Это была одна из первых в мире ЭВМ, способных выполнять аналитические вычисления наряду с численными.

Также следует отметить созданный в 1966 году В. Ф. Турчиным язык программирования Рефал, основанный на новом принципе программирования — ассоциативной обработке текстов на основе теории рекурсивных функций, без адресного управления программой. Компьютерная алгебра оказалась среди потенциальных областей приложения этого языка. Первым применением Рефала в компьютерной алгебре было решение В. Ф. Турчиным и др. задач ядерной физики [3].

Современные системы компьютерной алгебры можно разделить на системы общего назначения (универсальные) и специализированные. Системы компьютерной алгебры общего назначения, к которым относятся Reduce, Macsyma, Maple, Matematica и другие, находят своё применение при решении физических задач, в разнообразных областях научных и технических исследований, а также в учебном процессе (подробнее см. [4]). Специализированные системы отличаются более высокой эффективностью, но область их применения ограничена. К специализированным системам относятся такие системы, как CALEY и GAP — специализированные системы для вычислений в теории групп, MACAULEY, CoCoA, Singular — системы разной степени универсальности для вычислений в кольце многочленов, SCHOONSHIP — специализированная система для вычислений в физике высоких энергий, muMATH и её правонаследница

DERIVE — системы, широко используемые в учебном процессе, и многие другие.

Развитие систем компьютерной алгебры происходит в разных направлениях. На ранней стадии в основном внимание уделялось созданию самих систем и их приложениям. В последнее время всё большую популярность приобретают вопросы разработки специфических символьных алгоритмов и их реализация в существующих системах компьютерной алгебры.

Одна из важных областей компьютерной алгебры посвящена разработке алгоритмов решения задач, связанных с дифференциальными уравнениями и их системами. Рассматриваемые в этом направлении задачи не ограничиваются поиском и построением в символьном виде решений дифференциальных уравнений и систем, но также включают в себя различные алгоритмы преобразования уравнений, систем или ассоциированных операторов с целью установления различных свойств исходных уравнений или систем. Не все задачи в этой области оказываются алгоритмически разрешимыми, но даже для разрешимых задач вычислительная сложность алгоритмов может быть настолько высокой, что такие алгоритмы оказываются неприменимыми на практике.

Решением дифференциальной системы является вектор, компоненты которого соответствуют неизвестным (переменным) системы. В некоторых случаях, интерес могут представлять не все компоненты этого вектора, а какая-то их *выделенная* часть. Соответствующие выделенным компонентам решений неизвестные будем также называть *выделенными*. Задачи, приводящие к системам с выделенными неизвестными, возникают в разных областях. Системы, куда помимо «основных» (выделенных) также входят и избыточные переменные, типичны для математических моделей реальных явлений. Общее число переменных, как правило, больше числа интересующих исследователя переменных. Ещё одной причиной появления в системе избыточных переменных является составление систем дифференциальных уравнений возмущённого движения при изучении устойчивости тех или иных движений системы. Эволюция избыточных и «нефизических» переменных, как правило, интереса не представляет. Поэтому оправдана и целесообразна постановка задачи частичной устойчивости и стабилизации по отношению к выделенным переменным (см. [5]).

Как правило, системы компьютерной алгебры не предоставляют специализированных средств решения дифференциальных систем относительно части неизвестных. Существующие и реализованные алгоритмы поиска решений дифференциальных систем позволяют находить решения разного вида (полиномиальные, рациональные, в виде рядов и др.), но только такие, все компоненты которых имеют указанный вид. При частичном построении решений (поиск только выделенных компонент) вид невыделенных компонент интереса не представляет. Несоответствие вида выделенных и невыделенных компонент решений может приводить к невозможности использования общих алгоритмов построения решений дифференциальных систем. Для решения этой проблемы С. А. Абрамовым и М. Бронштейном был разработан AB-алгоритм (см. [6]), реализация которого вошла в систему компьютерной алгебры Maple. AB-алгоритм позволяет построить по заданной нормальной дифференциальной системе $y' = Ay$ с выделенными неизвестными новую нормальную дифференциальную систему, неизвестными в которой будут только выделенные неизвестные исходной системы и их производные. Решение построенной системы позволяет получить выделенные компоненты решений исходной системы.

Линейные однородные систем обыкновенных дифференциальных уравнений представимы в виде

$$A_r y^{(r)} + A_{r-1} y^{(r-1)} + \dots + A_1 y' + A_0 y = 0,$$

где $r \in \mathbb{N}$ — порядок системы, A_i — матрицы размера $m \times m$, $A_r \neq 0$ — *ведущая* матрица, $y = (y_1, \dots, y_m)^T$ — вектор неизвестных. Нормальные дифференциальные системы являются частным случаем линейных однородных дифференциальных систем первого порядка с обратимой ведущей матрицей. AB-алгоритм, реализованный в Maple, позволяет работать только с нормальными дифференциальными системами. И одной из задач, решаемых в настоящей работе, является разработка для системы компьютерной алгебры Maple процедуры, позволяющей применять реализованный AB-алгоритм для общего случая систем линейных однородных дифференциальных уравнений.

Также в работе вводится понятие *сателлитных* неизвестных: в дифференциальной системе с выделенными неизвестными мы называем *сателлитной* такую невыделенную неизвестную, соответствующая которой компонента решений принадлежит тому же дифференциальному расширению, что и выделенные компоненты, т. е. если выделенные компоненты решения имеют «хороший» вид, то и сателлитные тоже автоматически будут иметь «хороший» вид. Существование возможности эффективно определять сателлитные неизвестные позволяет при частичном решении систем относительно выделенных компонент почти задаром получать и сателлитные компоненты.

Цель диссертационной работы. Целью диссертационной работы является разработка для системы компьютерной алгебры Maple процедуры, позволяющий обобщить имеющуюся реализацию AB-алгоритма на случай линейных однородных дифференциальных систем произвольного порядка, а также реализация процедур распознавания сателлитных неизвестных в линейных однородных дифференциальных системах с выделенными неизвестными.

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

- разработать алгоритм преобразования линейных однородных дифференциальных систем с выделенными неизвестными к виду, допускающему применение процедуры, реализующей алгоритм Абрамова–Бронштейна ([6]);
- разработать алгоритмы распознавания сателлитных неизвестных в линейных однородных дифференциальных системах с выделенными неизвестными;
- разработать программный комплекс, реализующий алгоритмы для работы с линейными дифференциальными системами с выделенными неизвестными.

Научная новизна. В диссертационной работе получен ряд результатов, обладающих научной новизной:

- Разработан алгоритм Extract, позволяющий получать по линейной дифференциально-алгебраической системе полного ранга с выделенными неизвестными пару систем: нормальную дифференциальную и алгебраическую, — которые являются согласованными с исходной системой и множеством выделенных неизвестных. Алгоритм Extract позволяет обобщить на случай линейных однородных дифференциальных систем произвольного порядка известные алгоритмы, сформулированные для нормальных дифференциальных систем с выделенными неизвестными.

- Разработаны алгоритмы распознавания сателлитных неизвестных в линейных однородных дифференциальных системах с выделенными неизвестными.

- В среде Maple разработан программный комплекс, включающий реализацию алгоритма Extract, а также процедуры распознавания сателлитных неизвестных.

Теоретическая и практическая значимость. Предложенные в диссертационной работе алгоритмы могут быть реализованы и встроены в существующие системы компьютерной алгебры. Реализация для Maple доступна по адресу `http://www.ccas.ru/ca/`. Разработанные алгоритмы могут быть использованы совместно с процедурами поиска решений линейных дифференциальных систем. Предложенное понятие сателлитных неизвестных может быть полезно при описании свойств решений линейных однородных дифференциальных систем, а также в контексте задач построения частичных решений таких систем.

Положения, выносимые на защиту.

- Алгоритм Extract, позволяющий обобщить на случай полноранговых линейных однородных дифференциальных систем произвольного порядка известные алгоритмы, сформулированные для нормальных дифференциальных систем с выделенными неизвестными.

- Алгоритмы распознавания сателлитных неизвестных в линейных однородных дифференциальных системах с выделенными неизвестными.

- Программный комплекс для системы компьютерной алгебры Maple, включающий реализацию предложенных алгоритмов.

Апробация работы. Основные результаты по теме диссертации были представлены на следующих конференциях и семинарах:

- Объединенный семинар «Компьютерная алгебра» МГУ и ЛИТ ОИЯИ, г. Дубна, 2014, 2015 и 2016 гг.

- Семинар «Компьютерная алгебра» факультета ВМК МГУ и ВЦ РАН, г. Москва, 2015 и 2016 гг.

- Конференция «Актуальные вопросы программирования», посвящённая 90-летию со дня рождения Н. П. Трифонова, г. Москва, 2015 г.

- Международная конференция «FELIM», г. Лимож, Франция, 2016 и 2017 гг.

- Международная конференция «Компьютерная алгебра», г. Москва, 2016 и 2017 гг.

- Научная конференция «Ломоносовские чтения», г. Москва, 2017 и 2018 гг.

- Семинар «Аналитическая теория дифференциальных уравнений» Математического института РАН им. Стеклова, г. Москва, 2018 г.

- Семинар «Математическое моделирование» кафедры прикладной информатики и теории вероятностей РУДН, г. Москва, 2018 г.

Публикации. Основные результаты по теме диссертации изложены в 10 печатных работах, из них 5 статей в рецензируемых журналах из перечня ВАК и индексируемых Web of Science.

Личный вклад автора. Все представленные в диссертации результаты получены лично автором.

Структура и объём диссертации. Диссертация состоит из введения, пяти глав, заключения и списка литературы. Полный объём диссертации составляет 101 страницу. Список литературы содержит 39 наименований.

В первой главе даётся обзор средств построения решений линейных дифференциальных систем, предоставляемых системой компьютерной алгебры Maple. Указываются сложности применения общих алгоритмов при решении задачи построения частичных решений дифференциальных систем. Также приводятся основные понятия и обозначения, использующиеся в остальных главах.

Вторая глава посвящена алгоритму Extract, позволяющему обобщить на случай дифференциально-алгебраических систем алгоритмы, формулируемые для нормальных дифференциальных систем с выделенными неизвестными. Как следствие, алгоритм Extract делает возможным применение AB-алгоритма к полноранговым линейным дифференциальным системам произвольного порядка.

В третьей главе вводится понятие сателлитных неизвестных, а также описывается алгоритм распознавания сателлитных неизвестных для случая нормальных дифференциальных систем над полем рациональных функций. Описывается схема обобщения алгоритма распознавания сателлитных неизвестных на случай линейных дифференциальных систем высоких порядков, основанная на использовании алгоритма Extract.

Четвёртая глава посвящена ещё одному виду сателлитных неизвестных, а именно, линейно сателлитным неизвестным. Предлагается алгоритм распознавания таких неизвестных и обсуждается его обобщение на случай линейных дифференциальных систем высоких порядков.

Пятая глава посвящена описанию программного комплекса символьных вычислений, выполненного в среде компьютерной алгебры Maple. Программный комплекс включает в себя процедуру `Extract` — реализация одноимённого алгоритма, а также пакет `Satellite`, предоставляющий процедуры для работы с сателлитными неизвестными.

В заключение кратко сформулированы основные результаты работы.

Результаты диссертации опубликованы в работах [7–16].

ГЛАВА 1. ПРИМЕНЕНИЕ СИСТЕМ КОМПЬЮТЕРНОЙ АЛГЕБРЫ ДЛЯ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ СИСТЕМ

1.1. ОСОБЕННОСТЬ СИМВОЛЬНЫХ АЛГОРИТМОВ ПОСТРОЕНИЯ РЕШЕНИЙ ДИФФЕРЕНЦИАЛЬНЫХ СИСТЕМ

Системы компьютерной алгебры предоставляют возможность получать в символьном виде решения разного рода математических задач. В настоящей работе нас будет интересовать задача построения решений линейных однородных дифференциальных систем. Особенностью символьных алгоритмов решения дифференциальных систем, реализованных в системах компьютерной алгебры, является то, что они позволяют строить решения, исходя из его вида. Поэтому в современные системы компьютерной алгебры, как правило, входит набор процедур решения дифференциальных систем, каждая из которых позволяет получать решения определённого вида: полиномиальные, рациональные, экспоненциальные, решения в виде рядов и др. Например, в стандартную поставку системы компьютерной алгебры Maple входит пакет `DEtools`, в котором имеются процедуры построения в замкнутом виде решений разного вида, в частности процедура `polysols` (для построения полиномиальных решений), `ratsols` (для построения рациональных решений) и др. Пакет `Slode` предоставляет процедуры построения решений в виде формальных степенных рядов. Также отметим пакет `LinearFunctionalSystems`, процедуры которого также могут быть использованы для решения линейных дифференциальных систем.

Решением дифференциальной системы является вектор-столбец, компоненты которого соответствуют неизвестным системы. Использование процедур построения решений конкретного вида способно находить лишь такие решения систем, все компоненты которого имеют заданный вид. Иногда интерес представляют не все компоненты вектора неизвестных, а лишь их часть. При частичном решении дифференциальных систем использование общих процедур построения решений не всегда позволяет достичь результата. Например, в результате применения процедур поиска полиномиальных решений к системе

$$y' = \begin{bmatrix} -1/x & 2x \\ 0 & -1/x \end{bmatrix} y,$$

где $y = (y_1, y_2)^T$, будет получено лишь тривиальное решение, поскольку таких решений, обе компоненты которого имели бы вид ненулевых многочленов, у данной системы нет. В то же время, эта система имеет решения, в которых компонента y_1 имеет вид ненулевого многочлена.

Для решения проблемы частичного построения решений дифференциальных систем предназначен AB-алгоритм ([6]), реализация которого входит в стандартную поставку Maple (процедура `ReducedSystem` пакета `OreTools:-Consequences`). Поскольку алгоритмы, разработанные в рамках настоящего исследования, связаны с AB-алгоритмом, их реализация также выполнена в Maple. Процедура `ReducedSystem` позволяет построить новую дифференциальную систему, неизвестными которой будут лишь выделенные неизвестные исходной системы и их производные, тем самым делая возможным использовать общие методы построения решений дифференциальных систем для поиска только выделенных компонент. Процедура `ReducedSystem` позволяет работать лишь с наиболее простым видом линейных дифференциальных систем. Её обобщение на общий случай является одной из задач, решаемых в настоящей работе.

1.2. ДИФФЕРЕНЦИАЛЬНЫЕ СИСТЕМЫ С ВЫДЕЛЕННЫМИ НЕИЗВЕСТНЫМИ

Процедуры, предоставляемые системами компьютерной алгебры, в основном являются реализациями алгоритмов решения разного рода математических задач. Поэтому основная сложность создания таких процедур заключается непосредственно в разработке математического алгоритма решения задачи. В настоящем разделе приводятся сведения из теории дифференциальных уравнений и систем, которые необходимы для формулировки разрабатываемых алгоритмов. Также приводится описание AB-алгоритма и его свойств.

1.2.1. Основные понятия

Пусть K — дифференциальное поле характеристики 0 с операцией дифференцирования $\partial = {}'$, обладающей следующими свойствами: $\forall a, b \in K$: $(a+b)' = a' + b'$, $(ab)' = a'b + ab'$. Поле констант K обозначается $\mathrm{Const}(K)$ и определяется как множество тех элементов K, которые дифференцированием переводятся в 0: $\mathrm{Const}(K) := \{c \in K \mid c' = 0\}$. При разработке алгоритмов для систем компьютерной алгебры, необходимо, чтобы основное поле было *конструктивным* (см. [17]), т. е. чтобы полевые операции, включая дифференцирование и проверку на равенство нулю, было возможно выполнять алгоритмически (было возможно реализовать). Конструктивным, например, является поле $\mathbb{Q}(x)$ рациональных функций с рациональными коэффициентами.

Будем рассматривать линейные однородные системы обыкновенных дифференциальных уравнений, среди которых наиболее простой вид имеют *нормальные* дифференциальные системы, т. е. системы вида

$$y' = Ay, \tag{1.1}$$

где A — квадратная матрица над K размера $m \times m$, $y = (y_1, \ldots, y_m)^T$ — вектор неизвестных.

Определение 1. Дифференциальное поле K_0 с дифференцированием ∂_0 называется *дифференциальным расширением* K, если $K_0 \supseteq K$ и для всех $x \in K$ выполняется равенство $\partial_0 x = \partial x$.

Обозначим систему (1.1) через S.

Определение 2 ([18, Lem. 1.8])**.** *Пространство решений* $V_{K_0}(S)$ системы S в $K_0 \supseteq K$ определяется как $\{v \in K_0^m \mid v' = Av\}$ и является векторным линейным пространством над $\mathrm{Const}(K_0)$, размерность которого не превосходит m.

Как было замечено во введении, в компьютерной алгебре известно не мало алгоритмов построения решений дифференциальных систем вида (1.1) принадлежащих K и его расширениям, но эти алгоритмы способны находить лишь такие решения, все компоненты которых принадлежат заданному расширению. Наряду с построением решений дифференциальных систем, большой интерес представляет также задача построения частичных решений, т. е. получение решений не для всех, а только для части компонент вектора неизвестных. Эта задача перекликается с известной задачей из теории устойчивости, а именно с задачей частичной устойчивости ([5]).

Пусть часть компонент вектора неизвестных являются *выделенными*. Множество выделенных неизвестных обозначим s. Для дифференциальных систем с выделенными неизвестными можно рассматривать различные задачи, например, проверка существования решений, выделенные компоненты которых принадлежат заданным классам, поиск только выделенных компонент решений, частичная устойчивость решений по выделенным компонентам и др.

В работе [6] авторами рассматривается задача построения решений по отношению к части неизвестных системы и предлагается алгоритм (AB-алгоритм), позволяющий для случая нормальных дифференциальных систем решать упомянутые задачи. Отметим, что сама работа [6] явилась отправной точкой настоящего исследования. В следующем разделе мы подробно опишем AB-алгоритм, поскольку именно он положен в основу методов и алгоритмов, ставших предметом настоящей работы.

Введём ещё несколько понятий, которые потребуются в дальнейшем.

Определение 3. Дифференциальное расширение M поля K называется *расширением Пикара–Вессио* для дифференциальной системы $y' = Ay$, где $A \in K^{m\times m}$, $y = (y_1, \dots, y_m)^T$, если

1) существует обратимая матрица $B \in M^{m\times m}$ такая, что $B' = AB$ (таким образом, B является фундаментальной матрицей для рассматриваемой системы);

2) $\mathrm{Const}(M) = \mathrm{Const}(K)$;

3) M порождено как поле над K элементами фундаментальной матрицы B.

Известно (см. [18, Sec. 1.3]), что в случае, когда $\mathrm{Const}(K)$ алгебраически замкнуто, расширение Пикара–Вессио существует для любой нормальной дифференциальной системы.

Определение 4. Дифференциальное расширение M поля K называется *универсальным дифференциальным расширением (универсальным расширением*

Пикара–Вессио), если оно является расширением Пикара–Вессио для любой системы вида $y' = Ay$, в которой $A \in K^{m\times m}$ и $y = (y_1, \dots, y_m)^T$.

Может быть показано (см. [18, Ch. 10]), что для поля K, для которого $\mathrm{Const}(K)$ алгебраически замкнуто, универсальное дифференциальное расширение существует.

Лемма 1. *Пусть $\widetilde{K}$ — универсальное дифференциальное расширение K. Тогда любая неоднородная система*

$$y' = Ay + b, \tag{1.2}$$

где $A \in K^{m\times m}$, $b = (b_1, \dots, b_m)^T \in K^m$ и $y = (y_1, \dots, y_m)^T$ имеет решение в $\widetilde{K}$.

Доказательство. Рассмотрим нормальную дифференциальную систему

$$\tilde{y}' = \tilde{A}\tilde{y}, \tag{1.3}$$

где $\tilde{y} = (y_1, \dots, y_m, y_{m+1})^T$, а матрица $\tilde{A}$ имеет вид

$$\tilde{A} = \left[\begin{array}{ccc|c} & & & b_1 \\ & A & & \vdots \\ & & & b_m \\ \hline 0 & \dots & 0 & 0 \end{array}\right].$$

Поскольку $\tilde{A}$ является матрицей над K, дифференциальная система (1.3) имеет полное пространство решений в $\widetilde{K}$. Очевидно, что если $(y_1, \dots, y_m, 1)^T$ является решением (1.3), то $(y_1, \dots, y_m)^T$ — решение (1.2). Из последнего уравнения системы (1.3) следует, что решениями для компоненты y_{m+1} являются произвольные постоянные: $y_{m+1} = C_{m+1}$. Поэтому, потребовав дополнительно $C_{m+1} = 1$, получаем решения (1.2). $\square$

Определение 5. Две системы $y' = Ay$ и $z' = Bz$ $(A, B \in K^{m\times m})$ называются *эквивалентными (над K)*, если существует такая обратимая матрица $T \in K^{m\times m}$, что $T' = AT - TB$ и $z = Ty$.

Заметим, что расширения Пикара–Вессио эквивалентных систем совпадают.

1.2.2. АВ-алгоритм

Пусть S — система дифференциальных уравнений вида (1.1) с вектором неизвестных $y = (y_1, \dots, y_m)^T$, некоторые компоненты которого выделены.

Рассмотрим пространство P линейных форм от $y_1, \dots, y_m$ с коэффициентами из K: $P = \{a_1y_1 + \dots + a_my_m \mid a_i \in K\}$. Очевидно, что $P \cong K^m$, поэтому линейные формы из P будем изображать в виде векторов-столбцов коэффициентов: $f = a_1y_1 + \dots + a_my_m = (a_1, \dots, a_m)^T$. В P можно определить операцию дифференцирования $\Delta : P \to P$ в силу системы S:

$$\Delta f = f' + A^T f, \tag{1.4}$$

где f' обозначает результат покомпонентного дифференцирования.

Пусть $e_1, \dots, e_m$ — стандартный базис P:

$$e_1 = y_1, \dots, e_m = y_m.$$

Для простоты положим, что множество выделенных неизвестных состоит из первых n ($n \leq m$) неизвестных: $s = \{y_1, \dots, y_n\}$. Размерность l линейного пространства $\mathrm{Span}_n(S) \subseteq P$ линейных форм, порождённого $e_1, \dots, e_n$ и всеми линейными формами $\Delta^j e_i$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots$, называется *n-рангом* системы S. Если неотрицательные $l_1, \dots, l_n$ таковы, что линейные формы

$$\begin{array}{c} e_1, \Delta e_1, \Delta^2 e_1, \dots, \Delta^{l_1-1} e_1, \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ e_n, \Delta e_n, \Delta^2 e_n, \dots, \Delta^{l_n-1} e_n, \end{array} \tag{1.5}$$

образуют базис $f_1, \dots, f_l$ пространства $\mathrm{Span}_n(S)$, $l = l_1 + \ldots + l_n$, то упорядоченный набор $(l_1, \dots, l_n)$ называется *допустимым разбиением* n-ранга системы S.

Пример 1. Пусть система имеет вид

$$y' = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} y,$$

где $y = (y_1, y_2, y_3)^T$. 2-ранг l этой системы равен трём и единственное допустимое разбиение l — это $(2, 1)$. Разбиение $(1, 2)$ не является допустимым, так как $e'_2 = e_2$. С другой стороны, 3-ранг этой системы имеет три допустимых разбиения: $(2, 1, 0)$, $(0, 1, 2)$, $(1, 1, 1)$.

Каждая из производных Δf_i $(i = 1, 2, \dots, l)$ является линейной комбинацией над K линейных форм $f_1, \dots, f_l$. Введём новые неизвестные $z_1, \dots, z_l$, которые выражаются через $y_1, \dots, y_m$ посредством линейных форм $f_1, \dots, f_l$. Пусть матрица $B \in K^{l \times l}$ такова, что вектор-столбец коэффициентов Δf_i равен произведению B на вектор-столбец коэффициентов f_i, $i = 1, 2, \dots, l$. Выпишем систему уравнений

$$z' = Bz, \tag{1.6}$$

$z = (z_1, \dots, z_l)^T$. Базис $f_1, \dots, f_l$ можно расширить до базиса

$$f_1, \dots, f_l, f_{l+1}, \dots, f_m$$

пространства P. Использование линейных форм из этого базиса для введения новых неизвестных $Z = (z_1, \dots, z_l, z_{l+1}, \dots, z_m)^T$, соответствует замене вида

$$Z = Hy$$

с невырожденной матрицей $H \in K^{m \times m}$. Исходная система S преобразуется в

$$Z' = \begin{bmatrix} B & 0 \\ U & V \end{bmatrix} Z. \tag{1.7}$$

Таким образом, действие AB-алгоритма заключается в определении для заданного множества выделенных неизвестных s допустимого разбиения n-ранга и построении новой системы (1.6) для неизвестных $z_1 = f_1, \dots, z_l = f_l$, которые выражаются через $y_1, \dots, y_m$ посредством соответствующих линейных форм. Систему (1.6), которая является результатом применения AB-алгоритма к системе S по отношению к множеству выделенных неизвестных s, будем обозначать S_s^{AB}.

Предложение 1 ([6, Предл. 1]). (i) *Проекции на выделенные неизвестные пространств решений в произвольном расширении исходного дифференциального поля исходной системы $y' = Ay$ и системы $z' = Bz$ совпадают.*

(ii) *Если решение системы* $z' = Bz$ *таково, что его компоненты, соответствующие выделенным компонентам исходной системы, принадлежат некоторому дифференциальному расширению* K_0 *исходного дифференциального поля, то и все его компоненты принадлежат этому расширению.*

(iii) *Если размер* B *равен размеру* A *и исходная система обладает решением, выделенные компоненты которого принадлежат некоторому расширению исходного дифференциального поля, то* все *компоненты такого решения принадлежат этому расширению.*

Доказательство. (i) Положим $z = (z_1, \ldots, z_l)^T$, $\tilde{z} = (z_{l+1}, \ldots, z_m)^T$. Исходя из решения z системы (1.6), мы с помощью системы (1.7) получаем для $\tilde{z}$ неоднородную систему

$$\tilde{z}' = V\tilde{z} + Uz, \tag{1.8}$$

которая согласно лемме 1 должна иметь решение (в универсальном дифференциальном расширении $\widetilde{K}$ для расширения Пикара–Вессио, построенного для системы (1.6)). С другой стороны, n из неизвестных $z_1, \ldots, z_l$ равны $y_1, \ldots, y_n$ (преобразование H не изменяет значения этих неизвестных):

$$\begin{gathered} z_1 = y_1, z_2 = y_1', \ldots, z_{l_1} = y_1^{(l_1-1)}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ z_k = y_n, z_{k+1} = y_n', \ldots, z_l = y_n^{(l_n-1)}. \end{gathered}$$

(ii) Очевидно, так как K_0 замкнуто относительно дифференцирования.

(iii) В этом случае система (1.7) имеет вид $Z' = BZ$, и при этом $y = H^{-1}Z$; далее принимаем во внимание (ii). □

Замечание 1. Утверждение (iii) предложения 1 может быть сформулировано в более сильной форме: если размер B равен размеру A, тогда системы S и S_s^{AB} *эквивалентны* над K. Действительно, в этом случае для обратимой матрицы $H \in K^{m\times m}$ мы имеем $z = Hy$ и $H' = AH - HB$. Как следствие получаем, что любая неизвестная системы S может быть выражена линейно с коэффициентами из K через неизвестные системы S_s^{AB}, т. е. через выделенные неизвестные из множества s и их производные.

В дальнейшем нам потребуется оценка сложности AB-алгоритма, причём будет достаточно довольно грубой оценки.

Предложение 2. *Сложность AB-алгоритма допускает оценку* $O(m^4)$.

Доказательство. Вычисление производной в силу системы Δf произвольной линейной формы $f \in P$ согласно (1.4) осуществляется за $O(m^2)$ операций в K. Базис (1.5) строится за n шагов. Начиная с множества $\{e_1, \ldots, e_n\}$, на i-м шаге вычисляются и добавляются производные $\Delta^j e_i$ $(j = 1, 2, \ldots)$, пока получающаяся система остаётся линейно независимой. Проверка линейной независимости может быть осуществлена за $O(m^3)$ операций (приведение соответствующей матрицы к ступенчатому виду). Таких проверок потребуется провести не более m раз. Когда базис $f_1, \ldots, f_l$ $(l \leq m)$ построен, вычисление элементов матрицы B сводится к решению не более m систем линейных алгебраических уравнений размера не больше $m \times m$, каждая из которых может быть решена за $O(m^3)$ операций. Таким образом, сложность AB-алгоритма полиномиальна и допускает оценку $O(m^4)$. $\square$

Реализация AB-алгоритма входит в стандартную поставку системы компьютерной алгебры Maple в виде процедуры `ReducedSystem` подпакета `Consequences` стандартного пакета `OreTools`.

1.3. ДИФФЕРЕНЦИАЛЬНЫЕ СИСТЕМЫ ВЫСОКИХ ПОРЯДКОВ

Как отмечалось ранее, нормальные дифференциальные системы представляют собой наиболее простой вид линейных однородных дифференциальных систем. В общем случае линейные однородные дифференциальные системы имеют вид

$$A_r y^{(r)} + A_{r-1} y^{(r-1)} + \cdots + A_1 y' + A_0 y = 0, \tag{1.9}$$

где натуральное r называется *порядком* системы, $A_i \in K^{m \times m}$ $(i = 0, \ldots, r)$, $A_r \neq 0$, $y = (y_1, \ldots, y_m)^T$ — вектор неизвестных. Матрица A_r называется *ведущей* матрицей.

Обозначим через $K[\partial]$ кольцо дифференциальных операторов с коэффициентами из K. Система (1.9) может быть записана в операторном виде

$$Ly = 0, \tag{1.10}$$

где $L = A_r\partial^r + \cdots + A_1\partial + A_0 \in K[\partial]^{m\times m}$ ($A_r \neq 0$) представляет собой матричный дифференциальный оператор (операторную матрицу), для которого порядок $\mathrm{ord}(L) = r$. Матрица A_r называется *ведущей* матрицей оператора и обозначается $\ell c(L)$.

Для случая $L = 0$ будем считать $\mathrm{ord}(L) = -\infty$ и $\ell c(L) = 0$.

Матричный дифференциальный оператор L можно представить в виде матрицы скалярных дифференциальных операторов

$$L = \begin{bmatrix} L_{1,1} & \dots & L_{1,m} \\ \dots & \dots & \dots \\ L_{m,1} & \dots & L_{m,m} \end{bmatrix}. \tag{1.11}$$

Обозначим i-ю строку матрицы L через $L_{i,*}$.

Определение 6. Пусть $J \subseteq \{1, \dots, m\}$, тогда строки $L_{i,*}$, где $i \in J$, называются $K[\partial]$-*линейно зависимыми*, если существуют дифференциальные операторы $\{W_i\} \subset K[\partial]$, не все одновременно равные нулю, такие, что $\sum_{i\in J} W_i L_{i,*} = 0$; в противном случае они называются $K[\partial]$-*линейно независимыми.*

Ранг операторной матрицы L определим как максимальное число её $K[\partial]$-линейно независимых строк. Будем говорить, что система (1.10) имеет *полный ранг*, если ранг L равен m.

Определение 7. Операторная матрица $U \in K[\partial]^{m\times m}$ называется *унимодулярной*, если для неё найдётся операторная матрица $\widetilde{U} \in K[\partial]^{m\times m}$ такая, что $\widetilde{U}U = U\widetilde{U} = I_m$, где I_m — единичная матрица размера $m \times m$.

Определение 8. Пусть $L \in K[\partial]^{m\times m}$ и пусть $\delta_i = \max_{1\le j\le m}(0, \mathrm{ord}(L_{i,j}))$ ($i = 1, \dots, m$).

- Вектор-строка $\delta = (\delta_1, \dots, \delta_m)$ называется *вектором порядков строк* оператора L.

- Матрица $M \in K^{m\times m}$ такая, что её (i,j)-й элемент равен коэффициенту при ∂^{δ_i} скалярного дифференциального оператора $L_{i,j}$ представления (1.11), называется *фронтальной матрицей* L и обозначается $\ell c_{\mathrm{row}}(L)$.

- Операторная матрица L называется *приведённой по строкам*, если ненулевые строки $\ell c_{\mathrm{row}}(L)$ линейно независимы над K.

Алгоритмы, описанные в следующей главе, принимают на вход операторную матрицу в приведённом по строкам виде. Как показывает следующая лемма, любую операторную матрицу можно привести унимодулярными преобразованиями к виду, приведённому по строкам.

Лемма 2 ([19, Th. 2.2])**.** *Для любой операторной матрицы* $L \in K[\partial]^{m\times n}$ *ранга* $s \le \min(m,n)$ *существует унимодулярная матрица* $U \in K^{m\times m}$ *такая, что* UL *имеет форму*

$$UL = \left[\begin{array}{c} L^* \\ \hline 0 \end{array}\right],$$

где L^* *— приведённая по строкам операторная матрица размера* $s\times n$ *такая, что все её строки ненулевые и* $\mathrm{ord}(L^*) \le \mathrm{ord}(L)$.

Процесс, положенный в основу доказательства [19, Th. 2.2], определяет алгоритм, получивший название Row-Reduction. Его формальное описание можно также найти в [20, Sect. 2.1].

Заметим, что линейная дифференциальная система вида (1.9) порядка $r > 1$ с помощью перехода к вектору неизвестных

$$Y = \left(y_1, \ldots, y_m, y_1', \ldots, y_m', \ldots, y_1^{(r-1)}, \ldots, y_m^{(r-1)}\right)^T$$

может быть сведена к линейной однородной дифференциальной системе первого порядка

$$\tilde{A}_1 Y' + \tilde{A}_0 Y = 0, \tag{1.12}$$

в которой

$$\tilde{A}_1 = \begin{bmatrix} I_m & 0 & \ldots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & I_m & 0 \\ 0 & \ldots & 0 & A_r \end{bmatrix}, \quad \tilde{A}_0 = \begin{bmatrix} 0 & -I_m & & 0 \\ & \ddots & \ddots & \\ 0 & & 0 & -I_m \\ A_0 & A_1 & \ldots & A_{r-1} \end{bmatrix}.$$

В том случае, если ведущая матрица A_r системы (1.9) обратима, также обратимой будет и матрица $\tilde{A}_1$, и систему (1.12) можно переписать в нормальном виде $Y' = AY$, где

$$A = -\tilde{A}_1^{-1}\tilde{A}_0 = \begin{bmatrix} 0 & I_m & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & I_m \\ -A_r^{-1}A_0 & -A_r^{-1}A_1 & \dots & -A_r^{-1}A_{r-1} \end{bmatrix}. \qquad (1.13)$$

Таким образом, алгоритмы, предназначенные для обработки нормальных дифференциальных систем, обобщаются на случай линейных однородных дифференциальных систем высокого порядка с невырожденными ведущими матрицами. Заметим также, что матрица (1.13) является *разряженной* (содержит большое количество нулевых элементов). В некоторых случаях это позволяет использовать более эффективные алгоритмы обработки таких матриц, но в настоящей работе этот вопрос не рассматривается.

Если A_r вырождена, то построить по дифференциальной системе высокого порядка (1.9) нормальную дифференциальную систему нельзя. В эквивалентной дифференциальной системе первого порядка (1.12) ведущая матрица $\tilde{A}_1$ также будет вырожденной. Системы вида (1.12), ведущая матрица которых $\tilde{A}_1$ ненулевая, но вырожденная, называются линейными *дифференциально-алгебраическими* системами, и для обработки таких систем требуются специальные алгоритмы.

ГЛАВА 2. АЛГОРИТМ EXTRACT

Для программной реализации процедуры, позволяющей обобщить встроенную в Maple процедуру `ReducedSystem` (реализацию AB-алгоритма) на случай линейных однородных дифференциальных систем произвольного порядка, необходимо разработать алгоритм, который преобразует исходную систему к виду, позволяющему использовать `ReducedSystem`. Целью настоящей главы является разработка такого алгоритма. Как было отмечено ранее, произвольная система вида (1.9) может быть сведена к системе первого порядка вида (1.12), причём в случае обратимости ведущей матрицы можно построить эквивалентную нормальную систему, к которой AB-алгоритм применим. Нас будет интересовать случай дифференциально-алгебраических систем, т. е. случай, когда ведущие матрицы исходной системы высокого порядка и получающейся системы первого порядка вырождены.

Далее для простоты изложения будем полагать, что $\mathrm{Const}(K)$ алгебраически замкнуто. Кроме того будем считать, что все рассматриваемые дифференциальные расширения K имеют одно и то же поле констант $\mathrm{Const}(K)$.

Рассматривается дифференциально-алгебраическая система S вида

$$A_1 y' + A_0 y = 0, \tag{2.1}$$

где $A_1, A_0 \in K^{m\times m}$, $A_1 \neq 0$, $\det A_1 = 0$, $y = (y_1, \ldots, y_m)^T$ — вектор неизвестных. Системе S сопоставим линейный дифференциальный оператор $L = A_1\partial + A_0$. Предполагается, что система имеет полный ранг, и часть неизвестных $\varnothing \neq s = \{y_{i_1}, \ldots, y_{i_k}\} \subseteq \{y_1, \ldots, y_m\}$ являются *выделенными*. Система (2.1) является дифференциально-алгебраической системой и может быть разбита на две подсистемы, одна из которых дифференциальная, а вторая — алгебраическая. Существуют алгоритмы, осуществляющие такое преобразование (например, см. [21], [20, Sect. 5]). Алгоритм Extract, описание которого будет представлено в следующем разделе, также позволяет построить по дифференциально-алгебраической системе пару систем (дифференциальную и алгебраическую), но более специальным образом.

2.1. ОПИСАНИЕ АЛГОРИТМА

Основная цель алгоритма Extract — преобразовать исходную систему (2.1) к виду, пригодному для применения AB-алгоритма, т. е. получить нормальную систему вида (1.1) с выделенной частью неизвестных.

На вход алгоритма поступает дифференциальная система (2.1) с приведенной по строкам операторной матрицей. При этом, так как ведущая матрица A_1 вырождена, вектор порядков строк $\vec{\delta}$ будет содержать нулевые компоненты, а система разделится на две части: дифференциальную — состоящую из уравнений, соответствующих единичным элементам $\vec{\delta}$, и алгебраическую — соответствующую нулевым элементам $\vec{\delta}$.

Суть алгоритма состоит в том, чтобы из дифференциальной части исключить некоторые неизвестные, используя уравнения алгебраической части, и тем самым получить дифференциальную систему с числом уравнений, равным числу неизвестных, и имеющую обратимую ведущую матрицу. При исключении необходимо учитывать, какие неизвестные были изначально выделены. В связи с этим, исключения проводятся в два этапа: на первом этапе из дифференциальной части системы исключаются невыделенные неизвестные, а на втором — выделенные.

Алгоритм состоит из трёх этапов:

этап 1: исключение (из дифференциальной части системы) невыделенных неизвестных;

этап 2: исключение выделенных неизвестных;

этап 3: выражение исключённых выделенных неизвестных через оставшиеся в дифференциальной системе выделенные неизвестные.

Первый этап выполняется, если среди алгебраических уравнений системы есть такое, в которое входит какая-либо невыделенная неизвестная с ненулевым коэффициентом. Этот этап состоит из следующих шагов:

1. Из множества алгебраических уравнений выбрать уравнение, в которое какая-нибудь невыделенная неизвестная входит с ненулевым коэффициентом.

2. Используя выбранное на предыдущем шаге уравнение, произвести исключение этой невыделенной неизвестной из всех остальных уравнений системы как дифференциальных, так и алгебраических.

3. Использованное алгебраическое уравнение удалить из системы.

4. Если в системе остались алгебраические уравнения, в которые входят невыделенные неизвестные с ненулевыми коэффициентами, то вернуться к шагу 1.

В терминах операторных матриц исключение, выполняемое на шаге 2 первого этапа, можно описать как домножение операторной матрицы, соответствующей дифференциальной системе, слева на унимодулярную матрицу. Представим операторную матрицу, соответствующую системе, в виде

$$L = \begin{bmatrix} L_{11} & \dots & L_{1m} \\ \multicolumn{3}{c}{\dotfill} \\ L_{m1} & \dots & L_{mm} \end{bmatrix}.$$

Будем считать, что используемое алгебраическое уравнение соответствует m-й строке этой матрицы ($\operatorname{ord} L_{mj} = 0$ для $j = 1, \dots, m$), $L_{mm} \neq 0$ и что исключается m-я неизвестная (этого всегда можно добиться перестановкой уравнений и перенумерацией неизвестных). Тогда соответствующая унимодулярная матрица будет иметь вид:

$$U = \begin{bmatrix} 1 & 0 & \dots & -L_{1m}L_{mm}^{-1} \\ 0 & \ddots & & \vdots \\ \vdots & & 1 & -L_{m-1,m}L_{mm}^{-1} \\ 0 & \dots & 0 & 1 \end{bmatrix} \tag{2.2}$$

Если после первого этапа в системе не осталось алгебраических уравнений, то алгоритм заканчивает свою работу. Если же ещё остались алгебраические уравнения, то в них с ненулевыми коэффициентами будут присутствовать только выделенные неизвестные. Исключение из дифференциальной части некоторых из выделенных неизвестных осуществляется на втором этапе работы алгоритма.

Второй этап состоит из следующих шагов:

1. Из множества алгебраических уравнений выбрать произвольное уравнение.

2. Используя выбранное уравнение, произвести исключение какой-либо выделенной неизвестной, входящей в это уравнение с ненулевым коэффициентом, из всех остальных уравнений системы также, как и на шаге 2 первого этапа.

3. Использованное уравнение удалить из системы. Впоследствии (на третьем этапе алгоритма) оно найдет применение для выражения выделенных неизвестных, не вошедших в результирующую дифференциальную систему, через оставшиеся в ней выделенные неизвестные.

4. Если в системе остались алгебраические уравнения, то вернуться к шагу 1.

На заключительном третьем этапе алгоритма строится матрица T, посредством которой выделенные неизвестные, не вошедшие в результирующую дифференциальную систему, выражаются через выделенные неизвестные, вошедшие в неё. Размер T таков, что число её строк l равно числу выделенных неизвестных, не вошедших в результирующую дифференциальную систему (что соответствует числу исключений на втором этапе), а число столбцов равно числу выделенных неизвестных, попавших туда.

Третий этап состоит из следующих шагов:

1. Если на втором этапе были удалены все выделенные неизвестные, то

$$T = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix},$$

где число строк T равно числу выделенных неизвестных (выделенные неизвестные в этом случае тождественно равны нулю), и третий этап завершается. Иначе осуществляется переход к следующему шагу.

2. По алгебраическим уравнениям, использовавшимся на втором этапе для исключения выделенных неизвестных, строится матрица $D \in K^{l \times m}$ коэффициентов при неизвестных. При этом строки матрицы упорядочиваются согласно тому, в каком порядке использовались соответствующие алгебраические уравнения.

3. Столбцы матрицы, соответствующие невыделенным неизвестным, будут содержать только нулевые элементы (иначе соответствующие алгебраические уравнения можно было бы использовать на первом этапе алгоритма), поэтому они исключаются. Тем самым осуществляется переход к матрице $\tilde{D} \in K^{l \times k}$, где k — количество выделенных неизвестных в исходной системе.

4. Оставшиеся столбцы упорядочиваются так, чтобы первый соответствовал первой исключаемой на втором этапе выделенной неизвестной, второй — второй и т. д. После столбцов, соответствующих исключаемым на втором этапе неизвестным, следуют столбцы, соответствующие выделенным неизвестным, оставшимся в системе. Они упорядочиваются согласно тому, под какими индексами соответствующие неизвестные входят в результирующую систему. Тогда $\tilde{D}$ будет иметь вид

$$\tilde{D} = \left[\, D_1 \;\middle|\; D_2 \,\right],$$

где $D_1 \in K^{l \times l}$ имеет верхнетреугольную форму без нулей на главной диагонали, а $D_2 \in K^{l \times (k-l)}$. Искомая матрица T будет вычисляться как

$$T = -D_1^{-1} D_2.$$

Предложение 3. *Алгоритм Extract завершается, и результирующая дифференциальная система имеет невырожденную ведущую матрицу.*

Доказательство. На шаге 3 первого и второго этапов происходит удаление из системы одного из алгебраических уравнений, соответствующих нулевым компонентам вектора порядков строк $\vec{\delta}$, а так как число таких компонент не превосходит числа уравнений системы, алгоритм заканчивает свою работу через конечное число шагов.

После второго этапа в системе гарантированно не останется ни одного алгебраического уравнения. Ведущая матрица исходной дифференциальной системы не была тождественно равна нулю, поэтому как минимум одно дифференциальное уравнение в результирующей системе останется. Невырожденность ведущей матрицы получающейся дифференциальной системы обеспечивается тем, что каждое исключение неизвестной (с оговорками, приведенными в шаге 2 первого этапа алгоритма) осуществляется домножением на унимодулярную матрицу (2.2):

$$UL = \begin{bmatrix} 1 & 0 & \dots & -L_{1m}L_{mm}^{-1} \\ 0 & \ddots & & \vdots \\ \vdots & & 1 & -L_{m-1,m}L_{mm}^{-1} \\ 0 & \dots & 0 & 1 \end{bmatrix} \times L = \left[\begin{array}{ccc|c} & & & 0 \\ & \tilde{L} & & \vdots \\ & & & 0 \\ \hline L_{m1} & \dots & L_{m,m-1} & L_{mm} \end{array}\right]$$

Структура матрицы U такова, что ранги фронтальных матриц $\ell c_{\mathrm{row}}(L)$ и $\ell c_{\mathrm{row}}(UL)$ совпадают. Действительно, обозначив через $(UL)_{ij}$ элемент (i, j) матрицы UL, получаем:

$$(UL)_{ij} = \begin{cases} L_{ij} - L_{im}L_{mm}^{-1}L_{mj}, & \text{для } i < m \\ L_{ij}, & \text{для } i = m \end{cases}$$

Принимая во внимание тот факт, что порядок элементов L_{ij} $(1 \le i, j \le m)$ не превосходит единицы, каждый из них можно представить в виде

$$L_{ij} = a_{ij}\partial + b_{ij},$$

где $a_{ij}, b_{ij} \in K$, причем $a_{mj} = 0$, так как по предположению m-я строка является алгебраическим уравнением. В таких обозначениях для $(UL)_{ij}$ $(i < m)$ будем иметь

$$\begin{aligned} (UL)_{ij} &= a_{ij}\partial + b_{ij} - (a_{im}\partial + b_{im})b_{mm}^{-1}b_{mj} \\ &= a_{ij}\partial + b_{ij} - a_{im}\partial(b_{mm}^{-1}b_{mj}) - b_{im}b_{mm}^{-1}b_{mj} = \\ &= (a_{ij} - a_{im}b_{mm}^{-1}b_{mj})\partial + (b_{ij} - a_{im}(\partial(b_{mm}^{-1}b_{mj})) - b_{im}b_{mm}^{-1}b_{mj}) \end{aligned} \tag{2.3}$$

Пусть i-я строка L имеет первый порядок. Тогда элементами i-х строк фронтальных матриц $\ell c_{\mathrm{row}}(L)$ и $\ell c_{\mathrm{row}}(UL)$ будут a_{ij} и $a_{ij} - a_{im}b_{mm}^{-1}b_{mj}$ соответственно. Как видно, i-я строка фронтальной матрицы $\ell c_{\mathrm{row}}(UL)$ получается как результат вычитания из элементов i-й строки соответствующих элементов m-й строки матрицы $\ell c_{\mathrm{row}}(L)$, домноженных на константу $a_{im}b_{mm}^{-1}$.

Пусть теперь i-я строка L $(i < m)$ имеет нулевой порядок, т. е. соответствует алгебраическому уравнению. Тогда $a_{ij} = 0$ для всех $j = 1, \ldots, m$, и элементами i-х строк фронтальных матриц $\ell c_{\mathrm{row}}(L)$ и $\ell c_{\mathrm{row}}(UL)$ будут b_{ij} и $b_{ij} - b_{im}b_{mm}^{-1}b_{mj}$ соответственно. При этом i-я строка фронтальной матрицы $\ell c_{\mathrm{row}}(UL)$ получается как результат вычитания из элементов i-й строки соответствующих элементов m-й строки матрицы $\ell c_{\mathrm{row}}(L)$, домноженных на константу $b_{im}b_{mm}^{-1}$.

Таким образом, строки $\ell c_{\mathrm{row}}(UL)$ получаются как результат применения к $\ell c_{\mathrm{row}}(L)$ элементарных преобразований, которые не меняют ранг.

Операторная матрица L имеет фронтальную матрицу ранга m, так как L является приведенной по строкам операторной матрицей дифференциальной системы полного ранга. А так как $L_{mm} \neq 0$, операторная матрица $\tilde{L}$, к которой осуществляется переход на очередной итерации алгоритма, так же будет приведенной по строкам и иметь полный ранг. $\square$

Поскольку общее число исключений, требующихся алгоритму Extract не превосходит m, а каждое исключение требует $O(m^2)$ операций в K (следует из (2.3)), сложность самого алгоритма допускает оценку $O(m^3)$.

Теорема 1. *Пусть дифференциальная система* $A_1y' + A_0y = 0$ $(A_1, A_0 \in K^{m\times m}, A_1 \neq 0)$ *имеет полный ранг. Пусть выделена часть неизвестных, т. е. часть компонент вектора* y*. В этом случае для некоторой части* $\tilde{y}$ *компонент вектора* y *всегда можно построить систему вида* $\tilde{y}' = B\tilde{y}$ *так, что выделенные компоненты* y*, не вошедшие в* $\tilde{y}$*, линейно выражаются через вошедшие в* $\tilde{y}$ *выделенные компоненты.*

Доказательство. В случае, если A_1 обратима, $\tilde{y} = y, B = -A_1^{-1}A_0$, и утверждение доказано.

Пусть A_1 вырождена. В этом случае, применяя алгоритм Row-Reduction [19], всегда можно построить приведенную по строкам систему, к которой применим описанный выше алгоритм. Согласно предложению 3, в результате работы описанного алгоритма для некоторого подмножества $\tilde{y} \subset y$ неизвестных будет построена дифференциальная система $\tilde{A}_1\tilde{y}' + \tilde{A}_0\tilde{y} = 0$ с обратимой ведущей матрицей. Искомую матрицу B можно выразить как $B = -\tilde{A}_1^{-1}\tilde{A}_0$. Если в $\tilde{y}$ вошли не все выделенные неизвестные, то не вошедшие в $\tilde{y}$ неизвестные будут линейно выражаться лишь через выделенные неизвестные, вошедшие в $\tilde{y}$, с помощью матрицы T, построенной на третьем этапе алгоритма. Если в $\tilde{y}$ не попадает ни одной выделенной компоненты y, то для выделенных компонент возможны только нулевые решения. □

Пример 2. Пусть $K = \mathbb{Q}(x)$, $\partial = \frac{d}{dx}$, $y = (y_1, y_2, y_3, y_4)^T$. Рассмотрим систему

$$\begin{bmatrix} 0 & x & -2x & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} y' + \begin{bmatrix} 0 & 1 & -2x^2-2 & 0 \\ 1 & 0 & 1 & -x^2 \\ 1 & 2x & 0 & x^2 \\ 0 & 1 & 0 & x \end{bmatrix} y = 0 \tag{2.4}$$

с выделенными y_1 и y_2. Применяя описанный алгоритм, на первом этапе с использованием четвертого уравнения системы будет исключена неизвестная y_4. После первого этапа, в частности, третье уравнение системы примет вид

$$y_1 + xy_2 = 0. \tag{2.5}$$

На втором этапе, используя уравнение (2.5), будет исключена неизвестная y_1. В результате будет получена дифференциальная система с обратимой ведущей матрицей

$$\begin{bmatrix} x & -2x \\ 0 & x \end{bmatrix} \tilde{y}' + \begin{bmatrix} 1 & -2x^2-2 \\ 0 & 1 \end{bmatrix} \tilde{y} = 0, \tag{2.6}$$

где $\tilde{y} = (y_2, y_3)^T$, в которую не вошла выделенная неизвестная y_1. Но y_1 согласно (2.5) выражается линейно через y_2 — выделенную неизвестную, вошедшую в (2.6), при этом $T = [-x]$ и $y_1 = Ty_2 = -xy_2$.

В случае, когда ни одна из выделенных неизвестных не попадает в дифференциальную систему, на третьем этапе алгоритма в качестве матрицы T строится нулевой вектор-столбец с числом компонент равным числу выделенных неизвестных. Это означает, что выделенные неизвестные в этом случае будут иметь только нулевые решения. Действительно, если в результирующей системе не останется ни одной выделенной неизвестной, алгебраические уравнения, использованные на втором этапе алгоритма, будут представлять собой однородную алгебраическую систему с обратимой матрицей (в виду того, что исходная дифференциальная система была приведена по строкам и имела полный ранг).

Пример 3. Пусть $K = \mathbb{Q}(x)$, $\partial = \frac{d}{dx}$, $y = (y_1, y_2, y_3)^T$. Рассмотрим систему

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & x \end{bmatrix} y' + \begin{bmatrix} 1 & x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix} y = 0$$

с выделенными y_1, y_2. В результате применения алгоритма результирующая дифференциальная система примет вид

$$xy_3' - 2y_3 = 0$$

т. е. ни одной выделенной неизвестной в ней не останется. Очевидно, что первые два уравнения системы являются алгебраической системой относительно выделенных неизвестных:

$$y_1 + xy_2 = 0, \ \ y_2 = 0.$$

Отсюда следует, что выделенные неизвестные тождественно равны нулю.

Для случая, когда выделены все неизвестные, входящие в систему, первый этап алгоритма пропускается, а выполнение начинается непосредственно со второго этапа. Описанный алгоритм при этом позволяет произвести преобразование дифференциально-алгебраической системы так, чтобы её дифференциальная часть стала самостоятельной системой (подсистемой) с обратимой ведущей матрицей для части (не всех) неизвестных, причём те неизвестные, что не попали в дифференциальную подсистему, будут выражаться линейно с помощью алгебраической части через неизвестные дифференциальной подсистемы.

Пример 4. Пусть $K = \mathbb{Q}(x)$, $\partial = \frac{d}{dx}$, $y = (y_1, y_2, y_3, y_4)^T$. Рассмотрим систему

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & x^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} y' + \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & -2x^2 & x & -2x^3 \\ 0 & 1 & 0 & x \end{bmatrix} y = 0,$$

считая все входящие в систему неизвестные выделенными.

Результатом применения алгоритма Extract будет дифференциальная система с обратимой ведущей матрицей

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -x \\ 0 & x^2 & 0 \end{bmatrix} \tilde{y}' + \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -2 \\ -1 & x & 0 \end{bmatrix} \tilde{y} = 0, \tag{2.7}$$

где $\tilde{y} = (y_1, y_3, y_4)^T$, и матрица $T = \begin{bmatrix} 0 & 0 & -x \end{bmatrix}$, с помощью которой неизвестная y_2, не вошедшая в дифференциальную систему (2.7), выражается через вошедшую в эту систему неизвестную y_4: $y_2 = T \times (y_1, y_3, y_4)^T = -xy_4$.

Согласно теореме 1, результатами работы алгоритма Extract можно считать пару систем: нормальную дифференциальную систему, которую будем обозначать S_d и алгебраическую систему S_a специального вида, позволяющую выразить те выделенные неизвестные исходной системы, которые не вошли в S_d, только через выделенные неизвестные, которые в S_d вошли. Следующий раздел посвящён анализу размеров этих систем.

2.2. СОГЛАСОВАННЫЕ СИСТЕМЫ

Пусть F — дифференциальное расширение K. Множество решений системы S в F, т. е. $\{y \in F^m \mid L(y) = 0\}$, образует линейное пространство над $\mathrm{Const}(F) = \mathrm{Const}(K)$. Это пространство решений будем обозначать $V_F(S)$.

Проекцией пространства решений $V_F(S)$ *на неизвестные* $s = \{y_{l_1}, \ldots, y_{l_k}\} \subseteq \{y_1, \ldots, y_m\}$ будем называть множество

$$\pi_s(V_F(S)) = \{P_s y \mid y \in F^m, L(y) = 0\},$$

где P_s — матрица размера $k \times m$, элементы p_{ij} которой определяются следующим образом:

$$p_{ij} = \begin{cases} 1, & \text{если } l_i = j, \\ 0, & \text{иначе.} \end{cases} \tag{2.8}$$

Проекция $V_F(S)$ на s является линейным пространством над $\mathrm{Const}(F) = \mathrm{Const}(K)$.

Вектор-столбец, составленный из неизвестных множества s, будем обозначать y_s.

Пусть для системы S и множества выделенных неизвестных s существует такое его разбиение $s = s_1 \cup s_2$ ($s_1 \cap s_2 = \varnothing$), что найдется пара систем — дифференциальная S_d и алгебраическая S_a, удовлетворяющие следующим условиям:

C1: S_d является нормальной дифференциальной системой, т. е. системой вида (1.1), неизвестными которой являются неизвестные из s_1 и, может быть, другие неизвестные системы S, не являющиеся выделенными,

C2: S_a является линейной алгебраической системой, представимой в виде:

- если $s_1 \neq \varnothing$: $y_{s_2} = T y_{s_1}$, где T — матрица над K,
- если $s_1 = \varnothing$: $y_{s_2} = 0$.

В случае $s_2 = \varnothing$ (т. е. $s_1 = s$ и S_d содержит все выделенные неизвестные) будем считать, что множество уравнений S_a пусто.

Количество уравнений произвольной системы S будем называть *размером системы*. Размер S_d совпадает с количеством неизвестных, входящих в неё, и не может быть меньше $\#(s_1)$ — количества элементов множества s_1. Размер S_a в точности совпадает с $\#(s_2)$.

Определение 9. Системы S_d, S_a называются *согласованными* с (S, s) если выделенные компоненты s решений S в любом дифференциальном расширении исходного дифференциального поля однозначно определяются системами S_d, S_a.

Если системы S_d и S_a согласованы с (S, s), то компоненты решений S_d, S_a, соответствующие s, могут быть дополнены до решений S и не существует решений S, выделенные компоненты которых не могут быть получены из S_d, S_a.

Как не трудно видеть, системы S_d, S_a, строящиеся алгоритмом Extract, являются согласованными с исходной системой S и множеством выделенных неизвестных s. Перейдём к анализу их размеров.

2.2.1. Размер алгебраической системы

Лемма 3. *Пусть S_d, S_a согласованы с (S, s). Разбиение $s = s_1 \cup s_2$ в виде $s_1 = \varnothing$, $s_2 = s$ возможно тогда и только тогда, когда для любого дифференциального расширения $\widetilde{K} \supseteq K$ верно, что*

$$\pi_s(V_{\widetilde{K}}(S)) = \{0\}. \tag{2.9}$$

Доказательство. Необходимость очевидна, поскольку в случае $s_1 = \varnothing$ система S_a имеет вид $y_s = 0$. Достаточность докажем от противного. Пусть S_d имеет вид $\tilde{y}' = A\tilde{y}$, где $\tilde{y}$ — вектор-столбец длины n, который содержит выделенные неизвестные из $s_1 \neq \varnothing$. Возьмём в качестве $\widetilde{K} \supseteq K$ расширение Пикара–Вессио, построенное для S_d. Тогда по определению должна существовать обратимая матрица $B \in \widetilde{K}^{n\times n}$, являющаяся фундаментальной, т. е. каждый столбец B является решением S_d. Выполнение (2.9) означает, что выделенные компоненты s_1 могут иметь лишь нулевые значения, т. е. соответствующие строки B должны быть нулевыми, что противоречит её обратимости. Таким образом, множество s_1 выделенных неизвестных, попадающих в S_d, обязано быть пустым. □

Случай, когда проекция пространства решений S на выделенные неизвестные s имеет нулевую размерность, может быть выявлен алгоритмом Extract: в качестве матрицы алгебраической системы будет выдан нулевой вектор-столбец, длина которого равна общему числу выделенных неизвестных (третий этап алгоритма). В дальнейшем будем предполагать, что эта размерность положительна. Тогда, согласно лемме 3, $s_1 \neq \varnothing$.

Положим $Q = \begin{bmatrix} I_{\#(s_1)} \\ T \end{bmatrix}$ (в случае, когда $s_2 = \varnothing$, $Q = I_{\#(s)}$).

Лемма 4. *Каково бы ни было разбиение $s = s_1 \cup s_2$ ($s_1 \cap s_2 = \varnothing$, $s_1 \neq \varnothing$), если для него существуют системы S_d, S_a, удовлетворяющие условиям **C1**, **C2** и такие, что для любого дифференциального расширения $\widetilde{K} \supseteq K$ выполняется*

$$\pi_s(V_{\widetilde{K}}(S)) = \left\{ v \mid v = \tilde{I}Qv_1, v_1 \in \pi_{s_1}(V_{\widetilde{K}}(S_d)) \right\} \tag{2.10}$$

для некоторой перестановочной матрицы $\tilde{I}$ размера $\#(s) \times \#(s)$, тогда значения $\#(s_1)$, $\#(s_2)$ определяются однозначно исходной системой S и множеством выделенных неизвестных s.

Доказательство. Выполнение (2.10) для произвольного $\widetilde{K} \supseteq K$ эквивалентно условию согласованности систем S_d, S_a с (S, s). Возьмем в качестве $\widetilde{K}$ универсальное дифференциальное расширение. Пусть $\dim V_{\widetilde{K}}(S) = n$ и $e_1, \ldots, e_n$ — базис $V_{\widetilde{K}}(S)$. Рассмотрим матрицу $B \in \widetilde{K}^{\#(s)\times n}$, столбцами которой являются $P_s e_1, \ldots, P_s e_n$, где P_s — матрица проектирования, элементы которой определены как в (2.8). Каждая строка матрицы B соответствует одной из выделенных неизвестных S. Рассмотрим максимальный набор линейно независимых над K строк матрицы B. Количество строк в этом наборе определяет $\#(s_1)$. Остальные строки B линейно выражаются через этот линейно независимый набор с коэффициентами из K. Получающиеся линейные выражения с заменой строк на соответствующие им выделенные неизвестные составляют уравнения S_a, количество которых равно $\#(s_2) = \#(s) - \#(s_1)$. Тем самым получаем утверждение леммы. □

Предложение 4. *Пусть системы S_d, S_a согласованы с (S, s). Тогда размер S_a и количество выделенных неизвестных S, попадающих в S_d, определяются однозначно исходной системой S и множеством выделенных неизвестных s.*

Доказательство. Размер S_a равен $\#(s_2)$, $\#(s_1) = \#(s) - \#(s_2)$ равно числу выделенных неизвестных, попадающих в S_d. В случае, когда размерность проекции пространства решений S на s нулевая, истинность утверждения следует из леммы 3. Когда эта размерность положительна, условие согласованности S_d, S_a с (S, s) обеспечивает выполнение (2.10) в произвольном $\widetilde{K} \supseteq K$, и требуемое утверждение является следствием леммы 4. □

Отметим, что требование выполнения условия (2.10) для любого расширения $\widetilde{K} \supseteq K$ является существенным. При его нарушении лемма 4 и, следовательно, предложение 4 не обязаны выполняться, что демонстрирует следующий пример.

Пример 5. Пусть $K = \overline{\mathbb{Q}}(x)$, $\partial = \frac{d}{dx}$ и исходная система имеет вид:

$$\begin{bmatrix} x & 0 & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} y' + \begin{bmatrix} 1 & 0 & 0 & 0 \\ -x^3 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ -x & 0 & 0 & 1 \end{bmatrix} y = 0,$$

где $y = (y_1, y_2, y_3, y_4)^T$ — вектор неизвестных, из которых y_1 и y_2 — выделены.

Пространство решений дифференциальной системы

$$\tilde{y}' = \begin{bmatrix} -1/x & 0 \\ 1 & 0 \end{bmatrix} \tilde{y},$$

$\tilde{y} = (y_1, y_2)^T$, совпадает с проекцией пространства решений на y_1, y_2 исходной системы в произвольном дифференциальном расширении поля K, в частности, в самом K. Алгебраическая система при этом отсутствует. Но если ограничиться лишь рациональными решениями, т. е. интересоваться решениями, выделенные компоненты которых лежат в K, то можно перейти и к другой паре систем, из одного уравнения каждая:

$$y_2' = 0, \quad y_1 = 0.$$

Нетрудно проверить, что эти две системы (два уравнения) полностью описывают проекцию пространства решений исходной системы на неизвестные y_1, y_2 в K.

Предложение 5. *Для произвольной дифференциально-алгебраической системы (2.1) полного ранга с выделенными неизвестными можно построить такие удовлетворяющие условиям* ***C1***, ***C2*** *дифференциальную и алгебраическую системы S_d, S_a, что условие (2.10) будет выполнено для любого дифференциального расширения поля K.*

Доказательство. Описанный в предыдущем разделе алгоритм Extract решает поставленную задачу. Применяемые в алгоритме преобразования эквивалентны домножениям на унимодулярные операторные матрицы с элементами из $K[\partial]$, а значит, пространство решений остается неизменным не зависимо от того, какому дифференциальному расширению эти решения принадлежат. Это обеспечивает выполнение условия (2.10) для произвольного дифференциального расширения $\widetilde{K} \supseteq K$. Поэтому те системы, которые строит алгоритм Extract, удовлетворяют сформулированным требованиям. □

2.2.2. Размер дифференциальной системы

Алгоритм Extract позволяет получить нормальную дифференциальную систему S_d, размер которой, вообще говоря, не является минимальным из возможных. Причина в невыделенных неизвестных, которые не оказывают влияние на выделенные неизвестные, но присутствуют в S_d.

Пример 6. Пусть $K = \overline{\mathbb{Q}}(x)$, $\partial = \frac{d}{dx}$ и система S имеет вид:

$$\begin{bmatrix} x & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} y' + \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x & 0 & 0 & 1 \end{bmatrix} y = 0, \tag{2.11}$$

где $y = (y_1, y_2, y_3, y_4)^T$ — вектор неизвестных, у которого первые две компоненты — y_1 и y_2 — выделены. Применяя Extract к (2.11), получим следующую нормальную дифференциальную систему S_d:

$$\tilde{y}' = \begin{bmatrix} -1/x & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tilde{y}, \tag{2.12}$$

где $\tilde{y} = (y_1, y_2, y_3)^T$. Очевидно, что неизвестная y_3 из (2.12) может быть исключена, и тем самым размер системы может быть уменьшен.

Рассмотрим возможность уменьшения размера дифференциальной системы, получаемой в результате работы алгоритма Extract, с помощью AB-алгоритма. Для этого несколько изменим условие **C1**, накладываемое на S_d, так, чтобы оно соответствовало свойствам дифференциальной системы, получающейся в результате применения этого алгоритма:

C1$'$: S_d является нормальной дифференциальной системой, неизвестными которой являются только неизвестные из s_1 и, может быть, некоторые их производные.

Пример 7. Продолжая пример 6 и применяя AB-алгоритм к системе (2.12), получим дифференциальную систему, размер которой меньше:

$$\tilde{y}' = \begin{bmatrix} -1/x & 0 \\ 1 & 0 \end{bmatrix} \tilde{y}, \quad \tilde{y} = (y_1, y_2)^T. \tag{2.13}$$

Ясно, что получить дифференциальную систему для двух неизвестных меньшего размера нельзя, поэтому размер системы (2.13) является минимальным.

Предложение 6. *Пусть системы S_d, S_a согласованы с (S, s) и удовлетворяют условиям **C1′**, **C2**. Тогда размер S_d определяется однозначно исходной системой S и множеством выделенный неизвестных s.*

Доказательство. Возьмём в качестве $\widetilde{K}$ расширение Пикара–Вессио дифференциального поля K, построенное для системы S_d. Тогда размер S_d будет равен $\dim V_{\widetilde{K}}(S_d)$. В то же время $\dim \pi_s(V_{\widetilde{K}}(S)) = \dim \pi_{s_1}(V_{\widetilde{K}}(S)) = \dim \pi_{s_1}(V_{\widetilde{K}}(S_d))$. Общее решение для некоторой выделенной неизвестной y_i системы S_d в $\widetilde{K}$ может быть представлено в виде $y_i = \sum_{k=1}^{n} C_k g_k$, где $g_k \in \widetilde{K}$, а C_k — произвольные постоянные, количество которых (n) определяется размерностью $V_{\widetilde{K}}(S_d)$. Если неизвестная y_j системы S_d является некоторой производной y_i, т. е. $y_j = y_i^{(p)}$ для некоторого $p \in \mathbb{N}$, то общее решение для y_j представляется в виде $y_j = \sum_{k=1}^{n} C_k g_k^{(p)}$ с тем же или меньшим (в случае $g_k^{(p)} = 0$ для некоторых k) количеством произвольных постоянных. Таким образом, дополнительные неизвестные S_d, являясь производными неизвестных из s_1, не оказывают влияния на $\dim V_{\widetilde{K}}(S_d)$, т. е. $\dim V_{\widetilde{K}}(S_d) = \dim \pi_{s_1}(V_{\widetilde{K}}(S_d))$. Размер S_d совпадает со значением $\dim \pi_{s_1}(V_{\widetilde{K}}(S_d))$, которое определяется только исходной системой, откуда получаем требуемое утверждение. □

Дифференциальную систему, удовлетворяющую условию **C1′**, обозначим S_d^{AB}, и перейдем к рассмотрению случая, когда требования на состав неизвестных дифференциальной системы максимально ослаблены. Как и раньше будем рассматривать S_d и S_a, согласованные с (S, s), где S_a удовлетворяет условию **C2**, а условия на S_d переформулируем следующим образом:

C1″: S_d является нормальной дифференциальной системой, неизвестными которой являются неизвестные из s_1 и, может быть, какие-то другие (любые) неизвестные.

Нас будет интересовать вопрос, можно ли ослаблением условия на вид возможных дополнительных неизвестных S_d, добиться уменьшения её размера (сделать его меньше, чем размер S_d^{AB}). При условии согласованности систем S_d, S_a с (S, s), ответ на этот вопрос даёт следующее утверждение:

Теорема 2. *Пусть системы S_d, S_a согласованы с (S, s). Тогда размер дифференциальной системы S_d, на возможные дополнительные неизвестные которой не накладывается никаких ограничений, не может быть меньше, чем размер S_d^{AB}.*

Доказательство. Размер S_d не может быть меньше, чем $\dim \pi_{s_1}(V_{\widetilde{K}}(S))$, где $\widetilde{K}$ — расширение Пикара–Вессио, построенное для этой системы. В свою очередь, размер S_d^{AB} равен этой величине. □

2.2.3. Краткие выводы

Был рассмотрен процесс перехода от линейной дифференциально-алгебраической системы с выделенными неизвестными к паре систем: дифференциальной, в которую входит лишь часть выделенных неизвестных исходной системы, и алгебраической, выражающей оставшуюся часть выделенных неизвестных лишь через выделенные неизвестные, вошедшие в дифференциальную. При этом на дифференциальную систему накладывались различные дополнительные условия, обозначенные **C1**, **C1$'$**, **C1$''$**.

Условие **C1** соответствует дифференциальной системе, получаемой из исходной лишь унимодулярными преобразованиями. Такая дифференциальная система получается применением алгоритма Extract к исходной системе.

Дифференциальная система, удовлетворяющая условию **C1$'$** соответствует результату применения AB-алгоритма к дифференциальной системе, полученной с помощью Extract, и обладает тем свойством, что является минимальной по количеству неизвестных, входящих в неё.

Условие **C1$''$** отличается тем, что описывает дифференциальные системы без ограничений на возможные дополнительные неизвестные, и условия **C1** и **C1$'$** являются его частными случаями. Рассмотрение дифференциальных систем, удовлетворяющих **C1$''$**, позволяет обосновать предложенный алгоритм минимизации размера дифференциальной системы: к нормальной дифференциальной системе, полученной с помощью Extract, применить AB-алгоритм, получив тем самым нормальную дифференциальную систему S_d^{AB} для возможно части выделенных неизвестных исходной дифференциально-алгебраической системы, остальная часть выделенных неизвестных линейно выражается через выделенные неизвестные, попавшие в S_d^{AB}. Этот алгоритм может, например, использоваться для решения дифференциально-алгебраических систем относительно части неизвестных в ситуации, когда есть алгоритм нахождения таких решений нормальных систем, все компоненты которых принадлежат некоторому фиксированному расширению поля K. В этом случае минимальность размера получаемой дифференциальной системы представляет определенную ценность, уменьшая вычислительные затраты при поиске её решений.

Пример 8. Пусть $K = \overline{\mathbb{Q}}(x)$, $\partial = \frac{d}{dx}$ и система S имеет вид:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} y' + \begin{bmatrix} 1 & -x^2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & -1 \end{bmatrix} y = 0, \tag{2.14}$$

где $y = (y_1, y_2, y_3, y_4)^T$ — вектор неизвестных. Будем интересоваться рациональными решениями для компонент y_1, y_2. Если применить процедуру поиска рациональных решений непосредственно к системе (2.14), то для выделенных компонент будут получены лишь нулевые решения, хотя эта система обладает решениями, выделенные компоненты которых представляют собой ненулевые рациональные функции. Для их поиска воспользуемся описанной выше схемой.

Применение алгоритма Extract к исходной системе (2.14) позволяет получить согласованные системы

$$S_d\colon\ \tilde{y}' = \begin{bmatrix} -1/x & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad S_a\colon\ y_1 = x^2 y_2,$$

где $\tilde{y} = (y_2, y_3, y_4)^T$.

Далее воспользуемся AB-алгоритмом для получения по S_d дифференциальной системы наименьшего размера для поиска компоненты y_2 решений S:

$$S_d^{\mathrm{AB}}\colon\ y_2' = -\frac{1}{x}\,y_2.$$

Процедура поиска рациональных решений, применённая уже к системе S_d^{AB}, позволяет определить $y_2 = C/x$, где C — произвольная постоянная. Далее из S_a получаем $y_1 = Cx$.

Вид S_d^{AB} также позволяет заключить, что для любого решения исходной системы (2.14) его компоненты y_1, y_2 являются рациональными функциями.

2.2.4. Частичные решения

Пусть F — некоторое линейное пространство над K. Тогда можно поставить вопрос о существовании решений дифференциальной системы (2.1), выделенные компоненты которых принадлежат F. При этом дифференцирование предполагается расширенным на рассматриваемое линейное пространство.

Описанный в разделе 2.1 алгоритм позволяет для произвольной системы вида (2.1) построить нормальную систему для части неизвестных. При этом в построенную дифференциальную систему помимо выделенных неизвестных могут входить и другие невыделенные неизвестные. Эти невыделенные неизвестные могут препятствовать применению алгоритмов поиска решений в рассматриваемом линейном пространстве, если они ему не принадлежат. Для разрешения этой ситуации к построенной системе можно применить AB-алгоритм, в результате чего будет построена дифференциальная система, в которую, кроме выделенных неизвестных и их производных, не будут входить никакие другие «лишние» неизвестные.

Предложение 7. *Пусть F — некоторое линейное пространство над K, на которое распространено дифференцирование, определенное в K. Пусть имеется алгоритм поиска решений дифференциальных систем вида (1.1), целиком принадлежащих F. Тогда для систем вида (2.1) с выделенной частью неизвестных можно определять, существуют ли решения, выделенные компоненты которых принадлежат F, и при утвердительном ответе находить эти компоненты.*

Доказательство. Согласно теореме 1, по системе (2.1) можно построить нормальную систему
$$\tilde{y}' = B\tilde{y}, \tag{2.15}$$
куда войдут, быть может, не все неизвестные исходной системы, но при этом выделенные неизвестные, не попадающие в (2.15), будут выражаться линейно только через оставшиеся в (2.15) выделенные неизвестные. Далее, применяя АВ-алгоритм к системе (2.15), можно получить нормальную дифференциальную систему
$$z' = Rz, \tag{2.16}$$
неизвестными которой будут только выделенные неизвестные исходной системы, попавшие в (2.15), и некоторые их производные. Применяя к системе (2.16) алгоритм поиска решений, принадлежащих F, находим их, если они существуют. Тем самым, находим выделенные компоненты решений дифференциальной системы (2.1), попавшие в (2.15). Согласно предложению 1, они могут быть дополнены до решений (2.15), причем невыделенные неизвестные из (2.15), вообще говоря, не принадлежат F, а принадлежат некоторому расширению G, в качестве которого можно рассматривать универсальное дифференциальное расширение. Остальные выделенные неизвестные (2.1) линейно выражаются через найденные выделенные неизвестные, т. е. тоже оказываются в F. Невыделенные неизвестные из (2.1), не попавшие в (2.15), линейно выражаются через выделенные и невыделенные неизвестные (2.15), так что в общем случае принадлежат G. □

Полиномы не образуют линейного пространства над полем рациональных функций. Поэтому поиск полиномиальных компонент решений может вызывать дополнительные сложности: получив полиномиальные решения для выделенных компонент, попавших в (2.15), по ним ещё нужно найти остальные выделенные компоненты, не попавшие в дифференциальную систему (если такие были). Как уже было сказано, оставшиеся выделенные компоненты линейно выражаются через найденные компоненты, но коэффициенты этих линейных выражений представляют собой рациональные функции. В этом случае можно так ограничить значения произвольных констант, чтобы в результате остались полиномы.

Пример 9. Пусть $K = \overline{\mathbb{Q}}(x)$, $\partial = \frac{d}{dx}$, $y = (y_1, y_2, y_3)^T$. Рассмотрим систему

$$\begin{bmatrix} x & 0 & -2x \\ 0 & 0 & x \\ 0 & 0 & 0 \end{bmatrix} y' + \begin{bmatrix} 1 & 0 & -2x^2-2 \\ 0 & 0 & 1 \\ x & 1 & 0 \end{bmatrix} y = 0$$

и будем интересоваться полиномиальными решениями для y_1, y_2. Исключая неизвестную y_1 согласно описанному в разделе 2.1 алгоритму, получаем нормальную систему для $\tilde{y} = (y_2, y_3)^T$

$$\tilde{y}' = \begin{bmatrix} 0 & -2x^2 \\ 0 & -1/x \end{bmatrix} \tilde{y} \tag{2.17}$$

и выражение для y_1 через y_2: $y_1 = -\frac{1}{x} y_2$.

Применяя AB-алгоритм к системе (2.17), получаем систему

$$z' = \begin{bmatrix} 0 & 1 \\ 0 & 1/x \end{bmatrix} z,$$

где $z = (y_2, y_2')^T$, из которой находим полиномиальное решение $y_2 = C_1 x^2 + C_2$, где C_1 и C_2 — произвольные постоянные.

Подставляя найденное значение y_2 в выражение для y_1, получаем $y_1 = -C_1 x - C_2/x$, которое будет полиномом только в том случае, если положить $C_2 = 0$. В итоге получаем, что исходная система имеет решения с полиномиальными компонентами y_1 и y_2, при этом значения этих компонент равны: $y_1 = -C_1 x$, $y_2 = C_1 x^2$, где C_1 — произвольная постоянная.

В общем случае для выбора значений произвольных постоянных для получения полиномов можно использовать следующий метод. Выражение для выделенной неизвестной, не вошедшей в дифференциальную систему, нужно представить в виде с общим знаменателем:

$$y_j = \frac{C_1 p_1(x) + \cdots + C_s p_s(x)}{p(x)},$$

где $p(x), p_i(x)$ — полиномы, C_i — произвольные постоянные. Далее нужно найти $r_i(x)$ — остатки от деления $p_i(x)$ на $p(x)$. Тогда значения произвольных постоянных C_i, при которых y_j будет являться полиномом, определяются из соотношения $C_1 r_1(x) + \cdots + C_s r_s(x) = 0$, которое эквивалентно системе линейных алгебраических уравнений для $C_1, \dots, C_s$, число уравнений в которой равно $\max_{1 \le i \le s} \deg r_i(x)$.

2.2.5. Операторные матрицы

Алгоритм Extract может быть описан на языке операторных матриц. При этом выделенным неизвестным системы будут соответствовать выделенные столбцы операторной матрицы, сопоставленной системе.

Теорема 3. *Для операторной матрицы* $L = A_1\partial + A_0 \in K[\partial]^{m\times m}$ *полного ранга, в которой первые* k *столбцов являются выделенными, найдутся унимодулярные матрицы* U *и* V *такие, что*

$$\breve{L} = ULV = \begin{array}{c} \overbrace{\hphantom{C_1 \quad D_1}}^{k \text{ столбцов}} \hphantom{\quad 0 \qquad 0} \\ \left[\begin{array}{cc|cc} C_1 & D_1 & 0 & 0 \\ 0 & \multicolumn{2}{|c|}{\hat{A}_1\partial + \hat{A}_0} & 0 \\ D_2 & D_3 & D_4 & C_2 \end{array}\right] \end{array}, \tag{2.18}$$

где $C_1 \in K^{l\times l}$, $D_1 \in K^{l\times(k-l)}$, $\hat{A}_1, \hat{A}_0 \in K^{n\times n}$, $D_2 \in K^{(m-n-l)\times l}$, $D_3 \in K^{(m-n-l)\times(k-l)}$, $D_4 \in K^{(m-n-l)\times(n+l-k)}$, $C_2 \in K^{(m-n-l)\times(m-n-l)}$, *причём* C_1, C_2, $\hat{A}_1$ *— невырождены, а*

$$V = \begin{bmatrix} \tilde{I}_k & 0 \\ 0 & \tilde{I}_{m-k} \end{bmatrix},$$

где $\tilde{I}_k, \tilde{I}_{m-k}$ *— перестановочные матрицы порядка* k *и* $m-k$. *При этом размеры* l *и* n *определяются однозначно исходной операторной матрицей* L.

Доказательство. При помощи алгоритма Row-Reduction [19] может быть построена такая унимодулярная матрица U_1, что $L_1 = U_1L$ является приведенной по строкам. К L_1 применим алгоритм Extract, проводимые которым исключения осуществляются при помощи умножений на унимодулярные матрицы специального вида (см. раздел 2.1). Таким образом, при помощи алгоритма Extract могут быть получены унимодулярные матрицы U_2 и V такие, что матрица $\breve{L} = U_2L_1V = (U_2U_1)LV = ULV$ имеет вид (2.18) и также является приведенной по строкам. Матрица V используется только для перестановки столбцов таким образом, чтобы выделенные столбцы оставались среди первых k столбцов. При этом строки $\breve{L}$ с C_2 соответствуют первому этапу алгоритма — исключение невыделенных неизвестных, а строки с C_1 — второму — исключение выделенных неизвестных. Однозначность определения размера l следует из предложений 4 и 5.

Пусть $\vec{\delta} = (\delta_1, \ldots, \delta_m)^T$ — вектор порядков строк $\breve{L}$, элементы которого могут принимать два значения: $\delta_i = 0$, если элементы i-й строки $\breve{L}$ не содержат ∂ (соответствующее уравнение системы $\breve{L}(y) = 0$ является алгебраическим), и $\delta_i = 1$ иначе (соответствующее уравнение системы $\breve{L}(y) = 0$ является дифференциальным). Тогда $n = \sum_{i=1}^{m} \delta_i$, и однозначность его определения следует из [22, Th. 2]. $\square$

2.3. АЛГОРИТМ ExtrAB

Алгоритм Extract, хотя и представляет собой самостоятельный алгоритм со своей областью применения, оригинально был задуман как предварительный этап перед применением AB-алгоритма. Алгоритм, получающийся в результате такого объединения, получил название ExtrAB.

Алгоритм ExtrAB применяется к линейной однородной дифференциальной системе общего вида (1.9) с выделенными неизвестными и состоит из трёх этапов:

1. Преобразование исходной системы к системе первого порядка (1.12).

2. Применение к полученной системе алгоритма Extract — получение систем S_d и S_a.

3. Применение AB-алгоритма к системе S_d по отношению к той части выделенных неизвестных, что осталась в S_d — получение системы S_d^{AB}.

Результатом работы алгоритма являются системы S_d^{AB} и S_a (последняя может отсутствовать).

Тем самым алгоритм ExtrAB, позволяет обобщить AB-алгоритм на произвольные линейные дифференциальные системы полного ранга. Получаемые системы S_d^{AB} и S_a являются согласованными с исходной системой и множеством выделенных неизвестных (т. е. позволяют определять выделенные компоненты решений исходной системы в произвольном расширении основного дифференциального поля K) и их размеры являются минимальными (среди всех согласованных систем). При этом система S_d^{AB} такова, что если она имеет решение, выделенные компоненты которого принадлежат некоторому дифференциальному расширению K, то и остальные компоненты этого решения также принадлежат этому расширению.

Примеры применения ExtrAB уже были рассмотрены в предыдущих разделах (см. примеры 7, 8, 9, 20).

ГЛАВА 3. САТЕЛЛИТНЫЕ НЕИЗВЕСТНЫЕ

Понятие *сателлитных* неизвестных в дифференциальных системах с выделенными неизвестными, которое будет введено в настоящей главе, относится скорее к теории дифференциальных систем, нежели к компьютерной алгебре. Как будет показано, распознавание сателлитных неизвестных возможно алгоритмически, хотя высокая сложность алгоритма распознавания делает бесполезным его реализацию в системах компьютерной алгебры. В следующей главе будет рассмотрен один подкласса сателлитных неизвестных, допускающий эффективное распознавание с возможным практическим использованием.

Пусть $K = \overline{\mathbb{Q}}(x)$ — дифференциальное поле рациональных функций с обычной производной $' = \frac{d}{dx}$. Пусть S — нормальная дифференциальная система вида (1.1), часть неизвестных которой выделены. Множество выделенных неизвестных $s = \{y_{i_1}, \dots, y_{i_k}\}$ не пустое и не содержит сразу все неизвестные системы $(0 < k < m)$.

В этой главе будет введено понятие *сателлитных* неизвестных. Невыделенная неизвестная y_j данной системы S называется *сателлитной* для множества выделенных неизвестных s, если j-я компонента любого решения S принадлежит некоторому дифференциальному расширению F_s поля K, содержащему все выделенные компоненты всех решений S. Формальное определение сателлитных неизвестных будет дано в следующем разделе.

Пример 10. Рассмотрим следующую дифференциальную систему:

$$y' = \begin{bmatrix} 1/x & 0 & 0 \\ 3 & -1/x & 0 \\ 0 & 0 & 1 \end{bmatrix} y, \tag{3.1}$$

где вектор неизвестных $y = (y_1, y_2, y_3)^T$. Общее решение системы (3.1) может быть представлено в виде

$$y = C_1 \begin{bmatrix} x \\ x^2 \\ 0 \end{bmatrix} + C_2 \begin{bmatrix} 0 \\ 1/x \\ 0 \end{bmatrix} + C_3 \begin{bmatrix} 0 \\ 0 \\ e^x \end{bmatrix}, \tag{3.2}$$

где C_1, C_2, C_3 — произвольные постоянные. Из вида (3.2) следует, что компоненты любого решения (3.1), соответствующие неизвестным y_1 и y_2, являются рациональными функциями от x. В то же время, компоненты решений, соответствующие y_3, в общем случае не могут быть представлены в виде рациональных функций.

Множество сателлитных неизвестных данной системы зависит от множества выделенных неизвестных. В случае, если выделена только неизвестная y_1 (т. е. $s = \{y_1\}$), неизвестная y_2 будет являться сателлитной, а y_3 — нет. Поскольку $y_3 \in \overline{\mathbb{Q}}(x, e^x) \supset \overline{\mathbb{Q}}(x)$, обе другие неизвестные — y_1 и y_2 — будут сателлитными для множества выделенных неизвестных $s = \{y_3\}$.

В настоящей главе будет также представлен алгоритм, позволяющий найти все сателлитные неизвестные для заданной линейной дифференциальной системы с фиксированным множеством выделенных неизвестных.

Проблема распознавания сателлитных неизвестных может быть рассмотрена в контексте задачи частичного построения решений линейных дифференциальных систем. Как отмечается во введении [6], в компьютерной алгебре известно не мало алгоритмов, позволяющих строить решения линейных дифференциальных систем, принадлежащих полю $\overline{\mathbb{Q}}(x)$ и его расширениям. Но практически все эти алгоритмы позволяют находить решения, все компоненты которых принадлежат заданному расширению. Применение предлагаемого алгоритма распознавания сателлитных неизвестных позволит получить дополнительную информацию о данной системе. Эта информация далее может быть использована при частичном построении решений.

3.1. ОПРЕДЕЛЕНИЕ

Обозначим через K_S расширение Пикара–Вессио, построенное для системы S, V_S — пространство решений системы S в K_S^m.

Рассмотрим дифференциальное расширение $F_s \supseteq K$, удовлетворяющее следующим свойствам:

1. $\pi_s(V_S) \subseteq F_s^k$;

2. для любого расширения $F \supseteq K$ такого, что $\pi_s(V_S) \subseteq F^k$, выполняется условие $F_s \subseteq F$.

Расширение F_s можно рассматривать как дифференциальное поле, порождённое над K выделенными компонентами всех решений системы S.

Определение 10. Невыделенная неизвестная y_j системы S называется *сателлитной* для множества выделенных неизвестных s, если j-е компоненты всех решений S принадлежат F_s.

В следующем разделе будет описан алгоритм, позволяющий для заданной системы S и множества выделенных неизвестных s построить множество всех сателлитных неизвестных. Этот алгоритм использует в своей работе AB-алгоритм.

Не трудно видеть, что поле F_s из определения 10 может быть формально определено как расширение Пикара–Вессио для системы S_s^{AB}, т. е. $F_s = K_{S_s^{\text{AB}}}$. Поэтому определение 10 можно переформулировать в следующем виде.

Определение 11. Невыделенная неизвестная y_j системы S называется *сателлитной* для множества выделенных неизвестных s, если j-я компонента любого решения S принадлежит $K_{S_s^{\text{AB}}}$.

3.2. АЛГОРИТМ РАСПОЗНАВАНИЯ

Алгоритм, который будет представлен в этом разделе, позволяет определять для некоторой невыделенной неизвестной системы S, является ли она сателлитной для множества выделенных неизвестных или нет. Этот алгоритм может быть использован для построения множества всех сателлитных неизвестных. Далее через y_j будет обозначаться невыделенная неизвестная системы S, для которой решается вопрос, является ли она сателлитной, s — множество выделенных неизвестных, $\tilde{s} = s \cup \{y_j\}$.

Размер нормальной дифференциальной системы S (количество уравнений в системе) будем обозначать $|S|$.

Алгоритм 1. *Распознавание сателлитных неизвестных в нормальных дифференциальных системах.*

Вход: Нормальная дифференциальная система S с множеством выделенных неизвестных s; невыделенная неизвестная y_j.

Выход: Ответ «ДА», если y_j является сателлитной для s в S, и ответ «НЕТ» иначе.

1. Построить $S_s^{\text{АВ}}$.
2. Если $|S_s^{\text{АВ}}| = |S|$, **return** «ДА».
3. Построить $S_{\tilde{s}}^{\text{АВ}}$, где $\tilde{s} = s \cup \{y_j\}$.
4. Если $|S_s^{\text{АВ}}| = |S_{\tilde{s}}^{\text{АВ}}|$, **return** «ДА».
5. Если $K_{S_s^{\text{АВ}}} = K_{S_{\tilde{s}}^{\text{АВ}}}$, **return** «ДА».
6. **return** «НЕТ».

Прежде чем перейти к доказательству корректности алгоритма 1, прокомментируем некоторые из его шагов.

Системы $S_s^{\text{АВ}}$ и $S_{\tilde{s}}^{\text{АВ}}$ на шагах 1 и 3 строятся по исходной системе S с помощью применения АВ-алгоритма по отношению к множествам выделенных неизвестных s и $\tilde{s}$ соответственно. Шаг 5 является наиболее сложным с вычислительной точки зрения. В случае, когда $|S| \neq |S_s^{\text{АВ}}| \neq |S_{\tilde{s}}^{\text{АВ}}|$, неизвестная y_j будет являться сателлитной при выполнении условия

$$K_{S_s^{\text{АВ}}} \supseteq K_{S_{\tilde{s}}^{\text{АВ}}}. \tag{3.3}$$

Задача проверки, вложено ли расширение Пикара–Вессио одной системы в расширение Пикара–Вессио другой, может быть решена алгоритмически. Один такой алгоритм был предложен в [23]. Этот алгоритм (см. [23, Sect. 5.3.3 (H)]) основан на другом известном алгоритме, а именно алгоритме Хрушовского [24], который используется для вычисления дифференциальных групп Галуа. Алгоритм Хрушовского имеет довольно высокую сложность, но позволяет решать задачу в общем случае. В некоторых частных случаях могут быть использованы другие известные алгоритмы построения дифференциальных групп Галуа, например, алгоритм Ковачича [25] полностью решает задачу для систем размера два, в работе [26] описывается алгоритм построения дифференциальной группы Галуа, если она редуктивная.

Далее нам понадобится понятие *сводимости*. Одна задача *сводится* к другой, если возможно эффективное преобразование экземпляров первой задачи в экземпляры второй таким образом, что решение преобразованной задачи даёт ответ к исходной. Две задачи называются *эквивалентными*, если они сводятся друг к другу. Как будет показано, задача проверки вложения расширений Пикара–Вессио, которая возникает на шаге 5 алгоритма 1, эквивалентна задаче распознавания сателлитных неизвестных. Поэтому эти задачи имеют сопоставимую вычислительную сложность, и любое алгоритмическое улучшение в решении одной из этих задач автоматически влечёт улучшение в решении другой.

Предложение 8. *Проверка вложения расширений Пикара–Вессио двух заданных нормальных дифференциальных систем эквивалентна задаче распознавания сателлитных неизвестных.*

Доказательство. Пусть $A_1 \in K^{n\times n}$, $A_2 \in K^{m\times m}$ — матрицы двух заданные нормальных дифференциальных систем S_1 и S_2 соответственно. Рассмотрим новую дифференциальную систему S_3 следующего вида:

$$Y' = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} Y.$$

Вектор неизвестных Y состоит из $n+m$ компонент, из которых первые n будут считаться выделенными: $s = \{Y_1, Y_2, \ldots, Y_n\}$. Не сложно видеть, что $K_{S_3} \supseteq K_{S_1}$ и $K_{S_3} \supseteq K_{S_2}$. Поэтому условие $K_{S_1} \supseteq K_{S_2}$ будет верно тогда и только тогда, когда все невыделенные неизвестные в S_3 (т. е. $Y_{n+1}, \ldots, Y_{n+m}$) будут являться сателлитными для s. Таким образом, любой алгоритм распознавания сателлитных неизвестных может быть использован для решения задачи проверки вложений расширений Пикара–Вессио. $\square$

Предложение 9. *Алгоритм 1 решает задачу распознавания сателлитных неизвестных.*

Доказательство. Необходимо доказать корректность всех шагов алгоритма, содержащих оператор **return**.

Корректность шага 2 следует из (iii) предложения 1. Действительно, если $|S_s^{\text{AB}}| = |S|$, не только y_j, но и остальные невыделенные неизвестные будут сателлитными для s.

Для обоснования корректности шага 4 заметим, что применение AB-алгоритма к системе $S_{\tilde{s}}^{\text{AB}}$ по отношению к множеству выделенных неизвестных s даёт систему S_s^{AB}:

$$(S_{\tilde{s}}^{\text{AB}})_s^{\text{AB}} = S_s^{\text{AB}}.$$

Поэтому, если рассматривать систему $S_{\tilde{s}}^{\text{AB}}$ как исходную с выделенными неизвестными $s \subset \tilde{s}$, принимая во внимание (iii) предложения 1, получаем, что все невыделенные неизвестные в $S_{\tilde{s}}^{\text{AB}}$ являются сателлитными для s. Неизвестная y_j является невыделенной неизвестной $S_{\tilde{s}}^{\text{AB}}$, и потому является сателлитной.

На шаге 5 выполняется условие $|S_s^{\text{AB}}| \neq |S_{\tilde{s}}^{\text{AB}}|$, поэтому для того, чтобы y_j была сателлитной, необходимо и достаточно, чтобы

$$K_{S_s^{\text{AB}}} = K_{S_{\tilde{s}}^{\text{AB}}}. \tag{3.4}$$

Из свойств AB-алгоритма следует, что $K_{S_s^{\text{AB}}} \subseteq K_{S_{\tilde{s}}^{\text{AB}}}$, поэтому условие (3.4) эквивалентно условию (3.3) шага 5. Получаем, что корректность шага 5 следует непосредственно из определения 11. □

Пример 11. Рассмотрим дифференциальную систему S вида

$$y' = \begin{bmatrix} 1 & x & x^2 \\ 0 & -1/x & 1 \\ 0 & 0 & -1/x \end{bmatrix} y, \tag{3.5}$$

где $y = (y_1, y_2, y_3)^T$.

Положим, что выделена неизвестная y_1, т. е. $s = \{y_1\}$. Применяя AB-алгоритм к системе (3.5), получаем следующую систему S_s^{AB}:

$$z' = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} z,$$

где $z = (z_1, z_2, z_3)^T = (y_1, y_1', y_1'')^T$. Таким образом мы получаем, что $|S_s^{\text{AB}}| = |S| = 3$, следовательно неизвестные y_2 и y_3 являются сателлитными для $s = \{y_1\}$.

Положим, что выделенной является неизвестная y_2, т. е. $s = \{y_2\}$. Применение AB-алгоритма к системе (3.5) даёт систему S_s^{AB} следующего вида:

$$u' = \begin{bmatrix} 0 & 1 \\ 0 & -2/x \end{bmatrix} u,$$

где $u = (u_1, u_2)^T = (y_2, y_2')^T$. Для проверки того, является ли неизвестная y_3 сателлитной для $s = \{y_2\}$, нужно с помощью AB-алгоритма построить систему $S_{\tilde{s}}^{\text{AB}}$, где $\tilde{s} = s \cup \{y_3\} = \{y_2, y_3\}$. Система $S_{\tilde{s}}^{\text{AB}}$ будет иметь вид:

$$v' = \begin{bmatrix} -1/x & 1 \\ 0 & -1/x \end{bmatrix} v, \tag{3.6}$$

где $v = (y_2, y_3)^T$. Поскольку $|S_s^{\text{AB}}| = |S_{\tilde{s}}^{\text{AB}}| = 2$, неизвестная y_3 является сателлитной для $\{y_2\}$.

Наконец, проверим, является ли неизвестная y_1 сателлитной для $s = \{y_2, y_3\}$. Система S_s^{AB} будет иметь вид (3.6) и $|S_s^{\text{AB}}| = 2$. Пусть $\tilde{s} = s \cup \{y_1\} = \{y_1, y_2, y_3\}$. Применение AB-алгоритма к системе S по отношению к множеству выделенных неизвестных $\tilde{s}$ даст в результате исходную систему (3.5): $S_{\tilde{s}}^{\text{AB}} = S$. Получаем, что $|S_{\tilde{s}}^{\text{AB}}| = 3$, и проверка того, является ли y_1 сателлитной для $\{y_2, y_3\}$ должна осуществляться путём сравнения расширений Пикара–Вессио систем $S_{\tilde{s}}^{\text{AB}}$ и S_s^{AB}. В данном примере y_1 не является сателлитной, потому что $K_{S_{\tilde{s}}^{\text{AB}}} \neq K_{S_s^{\text{AB}}}$, но использовать алгоритм Хрушовского для установления этого факта нет необходимости. В этом конкретном случае достаточно использовать известные алгоритмы построения рациональных и экспоненциальных решений (например, см. [18, Sect. 4.1]) для установления того факта, что компоненты решений системы S, соответствующие y_1, не являются в общем случае рациональными функциями и принадлежат расширению $\overline{\mathbb{Q}}(x, e^x)$. В то же время компоненты решений, соответствующие y_2 и y_3, являются рациональными функциями.

3.3. ЛИНЕЙНЫЕ ДИФФЕРЕНЦИАЛЬНО-АЛГЕБРАИЧЕСКИЕ СИСТЕМЫ

Алгоритм распознавания сателлитных неизвестных (алгоритм 1), как он описан в предыдущем разделе, работает только с нормальными дифференциальными системами вида (1.1). Но он может может быть обобщён и на случай линейных дифференциальных систем высоких порядков вида (1.9).

Пусть дифференциальная система (1.9) имеет полный ранг и часть неизвестных (компонент вектора неизвестных) выделена. Мы можем определить понятие *сателлитных* неизвестных для систем (1.9) как и в определении 10 с небольшим изменением: вместо расширения Пикара–Вессио мы должны рассматривать универсальное дифференциальное расширение $\widetilde{K}$ и пространство решений системы (1.9) в $\widetilde{K}^m$.

Как отмечалось ранее (см. раздел 1.3), произвольная линейная однородная дифференциальная система (1.9) с помощью перехода к новым неизвестным может быть преобразована в систему первого порядка нормального вида (1.1) или в дифференциально-алгебраическую систему (1.12). Чтобы обобщить алгоритм распознавания сателлитных неизвестных из предыдущего раздела на случай произвольных линейных дифференциальных систем, остаётся рассмотреть случай дифференциально-алгебраических систем.

Будем рассматривать дифференциально-алгебраическую систему S вида (2.1) с выделенными неизвестными $s = \{y_{i_1}, \ldots, y_{i_k}\}$ $(0 < k < m)$. Для того, чтобы воспользоваться алгоритмом распознавания сателлитных неизвестных из предыдущего раздела, преобразуем систему S при помощи алгоритма Extract (см. раздел 2.1).

Алгоритм Extract по дифференциально-алгебраической системе S с выделенными неизвестными s строит новую нормальную дифференциальную систему S_d для части неизвестных $\tilde{y} \subseteq y$. Если какая-то выделенная неизвестная не входит в $\tilde{y}$ (т. е. не входит в систему S_d), она может быть выражена линейно над K только через выделенные неизвестные из $\tilde{y}$. Эти линейные выражения формируют алгебраическую систему S_a. Таким образом, алгоритм Extract по дифференциально-алгебраической системе с выделенными неизвестными строит две системы: нормальную дифференциальную систему S_d и алгебраическую систему S_a, — обе с коэффициентами из K.

Как и раньше, будем рассматривать задачу определения, является ли заданная невыделенная неизвестная y_j системы S сателлитной для множества выделенных неизвестных s. Для применения алгоритма распознавания сателлитных неизвестных к системам вида (2.1) будем в качестве начального этапа использовать алгоритм Extract с небольшой модификацией, суть которой заключается в следующем.

Входом алгоритма, наряду с матрицами A_1, A_0 системы (2.1) и множеством s выделенных неизвестных, будет также являться невыделенная неизвестная y_j, которая в ходе алгоритма будет считаться выделенной, как и неизвестные из s. Как и для оригинального Extract, будем полагать, что входная система приведена по строкам. Выходом алгоритма является либо нормальная дифференциальная система S_d для части неизвестных системы (2.1), включая y_j, либо линейное выражение y_j через выделенные неизвестные множества s. Заметим, что последнее сразу будет означать, что y_j является сателлитной неизвестной. Первый этап алгоритма остаётся без изменений. Второй этап — исключение выделенных неизвестных — изменяется следующим образом: при исключении очередной выделенной неизвестной (y_j при этом также считается выделенной) приоритет отдается неизвестной y_j и при её исключении алгоритм завершается, возвращая соответствующее линейное выражение y_j через неизвестные из s. В третьем этапе оригинального алгоритма Extract нет необходимости, и он не используется.

Модифицированный алгоритм Extract будем называть Extract-M.

Алгоритм 2. *Extract-M*

Вход: Матрицы A_1, A_0 дифференциально-алгебраической системы S; множество выделенных неизвестных s; невыделенная неизвестная y_j.

Выход: Либо нормальная дифференциальная система S_d для части неизвестных S, включая y_j, либо линейное выражение y_j через выделенные неизвестные множества s.

1. Провести исключение невыделенных неизвестных (неизвестных, не принадлежащих множеству $s \cup \{y_j\}$) с помощью алгебраических уравнений системы.

2. Провести исключение выделенных неизвестных (неизвестных из множества $s \cup \{y_j\}$) с помощью алгебраических уравнений системы, отдавая приоритет исключению неизвестной y_j. При исключении y_j завершить работу, вернув линейное выражение для y_j через неизвестные из s.

3. Построить систему S_d, преобразовав систему, полученную в результате исключений, к нормальному виду.

4. **return** S_d.

С помощью алгоритма Extract-M, алгоритм распознавания сателлитных неизвестных (алгоритм 1) может быть обобщен на случай дифференциально-алгебраических систем S вида (2.1) следующим образом:

Алгоритм 3. *Распознавание сателлитных неизвестных (для линейных дифференциально-алгебраических систем)*

Вход: Матрицы A_1, A_0 дифференциально-алгебраической системы S; множество выделенных неизвестных s; невыделенная неизвестная y_j.

Выход: Ответ «ДА», если y_j является сателлитной для s в S, и ответ «НЕТ» иначе.

1. Применить к системе S алгоритм Extract-M по отношению к s и y_j.

2. Если результатом шага 1 является линейное выражение y_j через неизвестные из s, **return** «ДА».

3. Результатом шага 1 является нормальная дифференциальная система S_d. Обозначим через s_1 ту часть выделенных неизвестных s, которые вошли в S_d.

4. Если $s_1 \neq \varnothing$, то применить к S_d алгоритм распознавания сателлитных неизвестных (алгоритм 1), используя s_1 в качестве множества выделенных неизвестных, и завершить работу, вернув полученный результат.

5. Построить систему $(S_d)^{\mathrm{AB}}_{\{y_j\}}$.

6. Если $K_{(S_d)^{\mathrm{AB}}_{\{y_j\}}} = K$, **return** «ДА».

7. **return** «НЕТ».

Замечание 2. Предлагаемая модификация алгоритма Extract не является необходимой. Extract-M обладает по сравнению с Extract более удобным интерфейсом, не выполняет лишнюю работу (построение алгебраической системы) и незначительно ускоряет получение положительного ответа при использовании совместно с алгоритмом распознавания сателлитных неизвестных (случай, когда y_j выражается линейно через выделенные неизвестные). Вместо Extract-M может быть использован оригинальный алгоритм Extract по отношению к пополненному множеству выделенных неизвестных $\tilde{s} = s \cup \{y_j\}$. Распознавание случая, когда y_j выражается линейно через неизвестные множества s осуществляется анализом алгебраической системы: либо y_j явно выражается через неизвестные из s, либо y_j входит с ненулевым коэффициентом в выражение некоторой неизвестной из s.

Предложение 10. *Алгоритм 3 позволяет определять сателлитные неизвестные в дифференциально-алгебраических системах полного ранга.*

Доказательство. Коэффициенты линейного выражения y_j через неизвестные из s, которое может обнаружить и вернуть алгоритм Extract-M, принадлежат K (следует из свойств алгоритма Extract). Существование такого линейного выражения означает, что для любого решения исходной дифференциально-алгебраической системы его y_j-я компонента линейно выражается через компоненты, соответствующие неизвестным из s (в частности, если для y_j-х компонент решения допустимы лишь нули). Следовательно, неизвестная y_j является сателлитной для множества выделенных неизвестных s. В этом случае, на шаге 2 будет возвращено «ДА».

Если же результатом работы Extract-M является нормальная дифференциальная система S_d, то в ней в качестве неизвестных присутствуют неизвестные подмножества s_1 множества выделенных неизвестных s ($s_1 \subseteq s$), неизвестная y_j и, возможно, другие невыделенные неизвестные исходной системы.

Если $s_1 \neq \varnothing$, то в этом случае y_j является сателлитной тогда и только тогда, когда она является сателлитной для множества неизвестных s_1 в S_d. Это следует из свойств алгоритма Extract: нормальная дифференциальная система, которую строит алгоритм Extract, является следствием исходной системы;

при исключении неизвестных используются алгебраические соотношения с коэффициентами из K, поэтому $\pi_s(V_S)$ полностью определяется $\pi_{s_1}(V_S)$, которая совпадает с $\pi_{s_1}(V_{S_d})$.

Если $s_1 = \varnothing$, то согласно лемме 3, это означает, что выделенные компоненты решений исходной системы могут принимать только нулевые значения. Для того, чтобы y_j была сателлитной необходимо и достаточно, чтобы j-я компонента любого решения S_d в K_{S_d} принадлежала K, что, с учётом свойств AB-алгоритма, эквивалентно условию шага 6. □

Пример 12. Рассмотрим следующую дифференциально-алгебраическую систему

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} y' + \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & (x-1)/x & (x^2-1)/x & x-1 \\ x & x & 1 & 0 \end{bmatrix} y = 0,$$

где $y = (y_1, y_2, y_3, y_4)^T$. Пусть множество выделенных неизвестных состоит из первых двух неизвестных: $s = \{y_1, y_2\}$. Проверим, что y_3 является сателлитной для s. Применим алгоритм Extract-M по отношению к множеству выделенных неизвестных $\tilde{s} = s \cup \{y_3\} = \{y_1, y_2, y_3\}$. На первом шаге алгоритм Extract-M будет использовать алгебраическое уравнение $y_1 + \frac{x-1}{x} y_2 + \frac{x^2-1}{x} y_3 + (x-1) y_4 = 0$ для исключения невыделенной неизвестной y_4. На втором шаге Extract-M получит алгебраическое уравнение $xy_1 + xy_2 + y_3 = 0$, в котором все входящие неизвестные являются выделенными. Получаем $y_3 = -xy_1 - xy_2$, откуда заключаем, что y_3 является сателлитной для s.

3.4. ЧАСТИЧНЫЕ АЛГОРИТМЫ РАСПОЗНАВАНИЯ САТЕЛЛИТНЫХ НЕИЗВЕСТНЫХ

Как было отмечено ранее, проверка условия шага 5 алгоритма распознавания сателлитных неизвестных (алгоритм 1) возможна алгоритмически, но соответствующий алгоритм имеет довольно высокую вычислительную сложность. Здесь мы рассмотрим модификацию алгоритма 1 в части шага 5, которая позволит решать задачу распознавания сателлитных неизвестных в некоторых частных случаях, и при этом имеет сравнительно небольшую вычислительную сложность.

Известно, что расширения Пикара–Вессио эквивалентных систем совпадают (см. [18, Ex. 1.16(1)]). Мы используем этот факт в нашем частичном алгоритме определения сателлитных неизвестных. Такой алгоритм позволит распознавать сателлитные неизвестные, т. е. получать утвердительный ответ, в некоторых случаях (не всегда). В то же время, он не позволит распознать случай, когда наперёд взятая неизвестная сателлитной не является.

Алгоритм 4. *Частичный алгоритм распознавания сателлитных неизвестных*

Вход: Нормальная дифференциальная система S с множеством выделенных неизвестных s; невыделенная неизвестная y_j.

Выход: Ответ «ДА», если удалось определить, что y_j является сателлитной для s в S, и ответ «НЕИЗВЕСТНО», иначе.

1. Построить S_s^{AB}.
2. Если $|S_s^{\mathrm{AB}}| = |S|$, **return** «ДА».
3. Построить $S_{\tilde{s}}^{\mathrm{AB}}$, где $\tilde{s} = s \cup \{y_j\}$.
4. Если $|S_s^{\mathrm{AB}}| = |S_{\tilde{s}}^{\mathrm{AB}}|$, **return** «ДА».
5. Если системы S_s^{AB} и $S_{\tilde{s}}^{\mathrm{AB}}$ эквивалентны, **return** «ДА».
6. **return** «НЕИЗВЕСТНО».

Замечание 3. Эквивалентными в смысле определения 5 раздела 1.2 могут быть только системы одинакового размера. Системы S_s^{AB} и $S_{\tilde{s}}^{\mathrm{AB}}$ на шаге 5 алгоритма 4 имеют заведомо разные размеры. Но меньшую из систем S_s^{AB} всегда можно дополнить до размера системы $S_{\tilde{s}}^{\mathrm{AB}}$ уравнениями вида $z_i' = 0$, где z_i — новые неизвестные. Не трудно видеть, что добавление таких новых уравнений не влияет на расширение Пикара–Вессио системы.

Задача проверки эквивалентности нормальных дифференциальных систем уже рассматривалась ранее (см. [27]) и сводится к поиску решений в K дифференциальной системы

$$T' = AT - TB. \tag{3.7}$$

В работе [27] приводятся некоторые необходимые условия для того, чтобы две системы были эквивалентными. В частности, одним из таких условий (см. [27, Prop. 1]) является наличие у скалярного уравнения

$$z' - (\operatorname{tr} A - \operatorname{tr} B)z = 0 \tag{3.8}$$

ненулевого рационального решения. Эти необходимые условия довольно важны с практической точки зрения, поскольку просты в вычислительном плане и позволяют дать быстрый ответ для большинства случаев, когда системы не эквивалентны.

Отметим, что условие эквивалентности систем S_s^{AB} и $S_{\tilde{s}}^{\mathrm{AB}}$, являясь достаточным, не является необходимым, что демонстрирует следующий пример.

Пример 13. Рассмотрим дифференциальную систему S следующего вида:

$$y' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} y,$$

в которой $y = (y_1, y_2)^T$. Очевидно, что компоненты решений, соответствующие обеим неизвестным, принадлежат расширению $\bar{\mathbb{Q}}(x, e^x)$. Поэтому неизвестная y_2 является сателлитной для $s = \{y_1\}$. В то же время, системы S_s^{AB} и $S_{\tilde{s}}^{\mathrm{AB}}$ эквивалентными не будут. В данном случае система S_s^{AB} будет состоять из одного уравнения $y_1' = y_1$, а система $S_{\tilde{s}}^{\mathrm{AB}}$ совпадает с исходной системой S. Тем самым будет нарушено упомянутое выше необходимое условие: уравнение (3.8) примет вид $z' + z = 0$, у которого нет нетривиальных решений в виде рациональных функций.

Алгоритм 4 также можно использовать в алгоритме 3 на шаге 4, получая тем самым частичный алгоритм распознавания сателлитных неизвестных в линейных дифференциально-алгебраических системах.

Процедуры, реализующие частичные алгоритмы распознавания сателлитных неизвестных, вошли в состав пакета `Satellite` для системы компьютерной алгебры Maple. Исходный код пакета доступен по адресу `http://www.ccas.ru/ca/satellite`. Описание пакета и примеры работы процедур представлены в главе 5.

ГЛАВА 4. ЛИНЕЙНО САТЕЛЛИТНЫЕ НЕИЗВЕСТНЫЕ

В этой главе будет введено новое понятие *линейно сателлитных неизвестных* — особый вид сателлитных неизвестных, алгоритм распознавания которых значительно проще с вычислительной точки зрения алгоритма распознавания сателлитных неизвестных. Как будет показано, алгоритм распознавания линейно сателлитных неизвестных может быть довольно просто реализован в системе компьютерной алгебры Maple. Также будут рассмотрены некоторые приложения, в частности, частичное построение решений линейных дифференциальных систем.

4.1. ПОНЯТИЕ ЛИНЕЙНО САТЕЛЛИТНЫХ НЕИЗВЕСТНЫХ

Будем рассматривать дифференциальное поле K характеристики 0, поле констант которого алгебраически замкнуто, и дифференциальную систему S вида (1.1) над K с неизвестными $y = (y_1, \dots, y_m)^T$, некоторые из которых выделены. Множество выделенных неизвестных $s = \{y_{i_1}, \dots, y_{i_k}\}$ не пустое и не содержит сразу все неизвестные (т. е. $0 < k < m$).

В главе 3 было введено понятие *сателлитных* неизвестных: невыделенная неизвестная y_j системы S называется сателлитной для множества выделенных неизвестных s, если j-я компонента любого решения S принадлежит дифференциальному расширению $F_s \supseteq K$, содержащему все выделенные компоненты всех решений S (см. определение 10). Одной из задач, связанной с дифференциальными системами, является задача частичного построения решений, когда интерес представляют не все компоненты вектора неизвестных, а только их часть (см. [6]). С этим также связана задача частичной устойчивости решений дифференциальных систем — устойчивости решений относительно части неизвестных, а также частичное управление (см. [5]). Сателлитные неизвестные обладают схожими с выделенными неизвестными свойствами. Поэтому алгоритмическая возможность распознавания сателлитных неизвестных может быть полезна в контексте упомянутых задач.

В настоящей главе будет рассмотрен один подвид сателлитных неизвестных, а именно *линейно сателлитные неизвестные.* В отличие от сателлитных неизвестных, в определении линейно сателлитных неизвестных не используются выделенные компоненты сразу всех решений системы. Невыделенная неизвестная y_j называется *линейно сателлитной* неизвестной для s, если j-я компонента любого решения S может быть выражена линейно над K через выделенные компоненты этого решения и их производные. Другими словами, невыделенная неизвестная y_j называется линейно сателлитной, если j-я компонента любого решения системы S принадлежит минимальному дифференциальному расширению K, содержащему все ненулевые выделенные компоненты этого решения. В случае, если все выделенные компоненты решения равны нулю, компоненты решений, соответствующие линейно сателлитным неизвестным, также должны быть равны нулю.

Распознавание линейно сателлитных неизвестных оказывается значительно менее трудоёмким по сравнению с распознаванием сателлитных неизвестных.

Пример 14. Рассмотрим следующую дифференциальную систему:

$$y' = \begin{bmatrix} 1/x & 1/(2x) & -1/x \\ 2/x & 3/x & -2/x \\ 1/x & 3/(2x) & -1/x \end{bmatrix} y, \tag{4.1}$$

где $y = (y_1, y_2, y_3)^T$. Компоненты любого решения системы (4.1) являются рациональными функциями. Это означает, что любая неизвестная y_j является сателлитной для любого непустого множества $s \subset \{y_1, y_2, y_3\}$, не содержащего y_j. В частности неизвестная y_2 является сателлитной для $\{y_1\}$. В то же время, поскольку система (4.1) обладает решением $(0, x^2, x^2/2)^T$, неизвестная y_2 не является линейно сателлитной для $\{y_1\}$, но является линейно сателлитной для $\{y_3\}$: у всех решений (4.1), для которых $y_3 = 0$, компоненты, соответствующие y_2 также нулевые.

4.1.1. Определение

Дифференциальное расширение F_s, определённое в разделе 3.1, может быть также рассмотрено как линейное пространство над K. Пусть $y =$

$(y_1, \ldots, y_m)^T$ — решение системы S. Определим $F_s(y)$ как минимальное линейное подпространство F_s, содержащее все выделенные компоненты y и их производные. $F_s(y)$ также может быть рассмотрено как линейное пространство, порождённое над K выделенными компонентами y и их производными. Важно заметить, что $F_s(y)$ в общем случае дифференциальным полем не является: если все выделенные компоненты решения y равны нулю, то $F_s(y) = \{0\}$.

Пример 15. Вернёмся к системе (3.1) из примера 10 предыдущей главы. Положим $s = \{y_1\}$. Несложно проверить, что $y = (x, x^2, e^x)^T$ является решением (4.1), $F_s(y)$ этого решения совпадает с основным дифференциальным полем $\overline{\mathbb{Q}}(x)$. Для решения $\tilde{y} = (0, -1/x, 0)^T$ получаем $F_s(\tilde{y}) = \{0\}$, что дифференциальным расширением $K = \overline{\mathbb{Q}}(x)$ не является.

Определение 12. Невыделенная неизвестная y_j называется *линейно сателлитной* неизвестной для множества выделенных неизвестных s в нормальной дифференциальной системе S, если для любого решения y верно, что его j-я компонента принадлежит $F_s(y)$.

Непосредственно из определения получаем следующие свойства линейно сателлитных неизвестных.

1. Если y_j — линейно сателлитная неизвестная для s в S, то j-я компонента любого решения системы S с нулевыми выделенными компонентами равна нулю.

2. Если y_j — линейно сателлитная неизвестная для s в S, то y_j — сателлитная неизвестная для s в S (поскольку для любого решения $y = (y_1, \ldots, y_m)^T$ верно, что $y_j \in F_s(y) \subseteq F_s$).

Распознавание линейно сателлитных неизвестных может быть проведено алгоритмически. Формальное описание такого алгоритма будет представлено в следующем подразделе.

4.1.2. Алгоритм распознавания линейно сателлитных неизвестных

Алгоритм распознавания линейно сателлитных неизвестных основан на AB-алгоритме и повторяет шаги алгоритма распознавания сателлитных неизвестных, исключая наиболее сложный с вычислительной точки зрения шаг 5. Как будет показано, оставшихся шагов достаточно для проверки, является ли наперёд взятая невыделенная неизвестная y_j линейно сателлитной для множества выделенных неизвестных s.

Алгоритм 5. *Распознавание линейно сателлитных неизвестных в нормальных дифференциальных системах.*

Вход: Нормальная дифференциальная система S с множеством выделенных неизвестных s; невыделенная неизвестная y_j.

Выход: Ответ «ДА», если y_j является линейно сателлитной для s в S, и ответ «НЕТ» иначе.

1. Построить S_s^{AB}.
2. Если $|S_s^{\mathrm{AB}}| = |S|$, **return** «ДА».
3. Построить $S_{\tilde{s}}^{\mathrm{AB}}$, где $\tilde{s} = s \cup \{y_j\}$.
4. Если $|S_s^{\mathrm{AB}}| = |S_{\tilde{s}}^{\mathrm{AB}}|$, **return** «ДА».
5. **return** «НЕТ».

Предложение 11. *Алгоритм 5 решает задачу распознавания линейно сателлитных неизвестных.*

Для доказательства предложения 11 потребуется следующая лемма.

Лемма 5. *Пусть s_1 и s_2 — непустые множества неизвестных системы S такие, что $s_1 \neq s_2$ и $s_1 \subset s_2$. Тогда следующие утверждения эквивалентны:*

(i) $|S_{s_1}^{\mathrm{AB}}| < |S_{s_2}^{\mathrm{AB}}|$;

(ii) $\dim \pi_{s_1}(V_S) < \dim \pi_{s_2}(V_S)$;

(iii) *У системы S существует решение в K_S^m, у которого все соответствующие неизвестным из s_1 компоненты равны нулю и хотя бы одна из компонент, соответствующих неизвестным из $s_2 \setminus s_1$, отлична от нуля.*

Доказательство. Для доказательства эквивалентности (i) и (ii) необходимо заметить, что для любого непустого множества неизвестных s расширение Пикара–Вессио $K_{S_s^{\mathrm{AB}}}$ является подполем K_S, поскольку неизвестными системы S_s^{AB} являются лишь неизвестные из s и их производные. Обозначим пространства решений систем S и S_s^{AB} через V_S и $V_{S_s^{\mathrm{AB}}}$ соответственно. Тогда согласно (i) предложения 1 получаем, что $\pi_s(V_S) = \pi_s(V_{S_s^{\mathrm{AB}}})$. Более того, будучи производными неизвестных из s, дополнительные неизвестные в S_s^{AB} не влияют на размерность пространства решений, и поэтому $\dim \pi_s(V_{S_s^{\mathrm{AB}}}) = \dim V_{S_s^{\mathrm{AB}}}$. Для завершения доказательства эквивалентности (i) и (ii) остаётся лишь заметить, что $\dim V_{S_s^{\mathrm{AB}}} = |S_s^{\mathrm{AB}}|$.

Для доказательства эквивалентности (ii) и (iii) рассмотрим подпространство $W \subseteq V_S$, содержащее только такие решения S, у которых соответствующие неизвестным из s_1 компоненты равны нулю:

$$W = \left\{ z = (z_1, \ldots, z_m)^T \in V_S \mid \forall i : y_i \in s_1 \Rightarrow z_i = 0 \right\}.$$

Тогда (ii) становится эквивалентно условию $\dim \pi_{s_2}(W) > 0$, что эквивалентно (iii). □

Доказательство предложения 11. Корректность шага 2 следует немедленно из пункта (iii) предложения 1 (см. замечание 1).

Для доказательства корректности шага 4 необходимо заметить, что $(S_{\tilde{s}}^{\mathrm{AB}})_s^{\mathrm{AB}} = S_s^{\mathrm{AB}}$. Поэтому если рассматривать систему $S_{\tilde{s}}^{\mathrm{AB}}$ как исходную с множеством выделенных неизвестных $s \subset \tilde{s}$, то согласно (iii) предложения 1 (см. замечание 1) получаем, что все невыделенные неизвестные в $S_{\tilde{s}}^{\mathrm{AB}}$, включая неизвестную y_j, являются линейно сателлитными для s.

Для завершения доказательства остаётся показать корректность шага 5. В этом случае выполняется неравенство $|S_s^{\mathrm{AB}}| < |S_{\tilde{s}}^{\mathrm{AB}}|$, и по лемме 5 получаем, что система S обладает решением с нулевыми выделенными компонентами и ненулевой компонентой y_j. Существование такого решения означает, что y_j не является линейно сателлитной неизвестной для s. □

Пример 16. Обозначим систему (4.1) из примера 14 через S. Положим $s = \{y_1\}$ и проверим, является ли неизвестная y_2 линейно сателлитной для s. Применение AB-алгоритма даёт системы S_s^{AB} и $S_{s\cup\{y_2\}}^{\mathrm{AB}}$ с матрицами

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{и} \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 2 & 2/x \end{bmatrix}$$

соответственно. Поскольку размеры матриц не совпадают, y_2 не является линейно сателлитной для $\{s_1\}$, как это было показано в примере 14.

Для $s = \{y_3\}$ AB-алгоритм построит систему с матрицей 3×3 следующего вида:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Поскольку $|S_s^{\mathrm{AB}}| = |S|$, обе неизвестные y_1 и y_2 являются линейно сателлитными для $\{y_3\}$.

4.1.3. Линейные дифференциально-алгебраические системы

Алгоритм 5 может быть обобщён на случай полноранговых систем линейных однородный дифференциальных уравнений высокого порядка вида (1.9). Как и раньше, через s будет обозначаться непустое множество выделенных неизвестных такой системы. Определение линейно сателлитных неизвестных для систем (1.9) дословно повторяет определение 12, где $F_s(y)$ — линейное пространство порождённое над K выделенными компонентами решения y и их производными. Поскольку произвольная система (1.9) порядка $r > 1$ может быть сведена к системе вида (1.1) или дифференциально-алгебраической системе вида (1.12), остаётся рассмотреть возможность распознавания линейно сателлитных неизвестных именно в дифференциально-алгебраических системах.

Будем рассматривать систему вида (2.1), где $A_1, A_0 \in K^{m\times m}$, $A_1 \neq 0$, $\det A_1 = 0$, $y = (y_1, \dots, y_m)^T$ — вектор неизвестных, $s \subset \{y_1, \dots, y_m\}$ — непустое множество выделенных неизвестных.

Для обобщения алгоритма 5 на системы (2.1) используем тот же подход, что и в разделе 3.3, основанный на использовании алгоритма Extract.

Применяя алгоритм Extract к дифференциально-алгебраической системе (2.1) по отношению к множеству выделенных неизвестных $\tilde{s} = s \cup \{y_j\}$, где y_j ($y_j \notin s$) является тестируемой неизвестной (для которой решается вопрос, является ли она линейно сателлитной для множества выделенных неизвестных s), будут построены системы S_d и S_a, согласованные с исходной системой (2.1) и множеством выделенных неизвестных $\tilde{s}$. Если тестируемая неизвестная y_j при этом окажется в алгебраической системе S_a (см. замечание 2), тогда она может быть выражена линейно через выделенные неизвестные из s. Это означает, что y_j в этом случае является линейно сателлитной для s в (2.1). Если же y_j отсутствует в S_a, тогда эта неизвестная необходимо окажется среди неизвестных нормальной дифференциальной системы S_d, и тем самым проблема распознавания линейно сателлитных неизвестных для дифференциально-алгебраической системы будет сведена к проблеме распознавания для нормальной системы S_d и решена с помощью алгоритма 5.

4.2. НЕПРИВОДИМЫЕ СИСТЕМЫ

В этом разделе в качестве системы S будем рассматривать нормальную дифференциальную систему вида (1.1) с выделенными неизвестными, образующими множество s.

В предыдущей главе было рассмотрено понятие сателлитных неизвестных, а также был предложен алгоритм распознавания сателлитных неизвестных. В настоящей главе был рассмотрен подкласс сателлитных неизвестных, а именно линейно сателлитные неизвестные, а также алгоритм распознавания линейно сателлитных неизвестных. В отличие от алгоритма распознавания сателлитных неизвестных, алгоритм распознавания линейно сателлитных неизвестных обладает довольно низкой сложностью и сводится не более чем к двукратному применению АВ-алгоритма. Таким образом, время работы алгоритма распознавания линейно сателлитных неизвестных полиномиально ограничено (см. предложение 2).

Как уже отмечалось, в системе S с выделенными неизвестными, образующими множество s, любая линейно сателлитная для s неизвестная является также и сателлитной неизвестной. Тем самым, в некоторых случаях распознавание сателлитных неизвестных может быть осуществлено с помощью алгоритма распознавания линейно сателлитных неизвестных. Существуют системы, в которых верно и обратное утверждение: для фиксированного множества выделенных неизвестных s любая сателлитная неизвестная также является и линейно сателлитной. Примером таких систем является класс неприводимых систем. Настоящий раздел будет посвящён обоснованию этого факта.

4.2.1. Факторизация систем на основе AB-алгоритма

Задача факторизации напрямую связана с понятием приводимости дифференциальных систем.

Определение 13 ([26, Sect. 2.3])**.** Дифференциальная система $y' = Ay$, $A \in K^{m\times m}$, называется *приводимой*, если она эквивалентна системе $z' = Bz$, у которой матрица B имеет блочную нижнетреугольную форму

$$B = \begin{bmatrix} B_{1,1} & 0 \\ B_{2,1} & B_{2,2} \end{bmatrix}, \tag{4.2}$$

где $B_{1,1}$, $B_{2,2}$ — квадратные матрицы. В противном случае, система называется *неприводимой*.

В литературе также можно встретить эквивалентное определение приводимых систем, в котором матрица B имеет блочную верхнетреугольную форму (см. [28, Sect. 1.1], [29, с. 113]). Для нас будет удобнее иметь дело именно с нижнетреугольной матрицей.

Определение 14 ([28, Def. 1.2])**.** Под *факторизацией над K дифференциальной системы $y' = Ay$* понимается либо нахождение эквивалентной над K системы $z' = Bz$, матрица которой имеет вид (4.2), либо заключение, что такой системы не существует.

Пусть для системы S вида (1.1) и некоторого i $(1 \le i \le m)$ выполняется условие

$$|S^{\mathrm{AB}}_{\{y_i\}}| = n < m = |S|. \tag{4.3}$$

Как будет показано далее, условие (4.3) является достаточным для того, чтобы система S была приводимой. При этом возможен эффективный процесс факторизации системы, построенный только на основе AB-алгоритма.

Не ограничивая общности, далее будем полагать $i = 1$ в (4.3).

В [6, Разд. 2] отмечалось, что базис из неизвестных системы $S^{\mathrm{AB}}_{\{y_1\}}$ (т. е. $y_1, y_1', \dots, y_1^{(n-1)}$) может быть дополнен до базиса пространства всех линейных форм от $y_1, \dots, y_m$ так, что соответствующее преобразование перехода к новому базису будет являться невырожденной над K заменой неизвестных $z = Ty$, приводящей исходную систему к виду (4.2). Но вопрос о построении такого преобразования T не обсуждался. Следующий алгоритм описывает это построение.

Алгоритм 6. *Построение эквивалентной системы с блочной нижнетреугольной матрицей.*

Вход: Дифференциальная система S такая, что $|S^{\mathrm{AB}}_{\{y_1\}}| = n < m$.

Выход: Блочная нижнетреугольная матрица B (матрица эквивалентной системы $z' = Bz$) и матрица T преобразования $z = Ty$.

1. $s := \{y_1\}$;
2. **for** y_i **in** $\{y_2, \dots, y_m\}$ **do**
   (a) **if** $|S^{\mathrm{AB}}_{s\cup\{y_i\}}| > |S^{\mathrm{AB}}_{s}|$ **then** $s := s \cup \{y_i\}$;
   (b) **if** $|S^{\mathrm{AB}}_{s}| = m$ **then break**;
3. $B :=$ матрица системы S^{AB}_{s};
4. Неизвестными новой системы $z' = Bz$ служат неизвестные, входящие в множество s (часть неизвестных исходной системы), и некоторые их производные, причём
   - первые компоненты $z_1, \dots, z_n$ соответствуют $y_1, y_1', \dots, y_1^{(n-1)}$;
   - если $z_j = y_i^{(k)}$, $k > 0$, то $z_{j-1} = y_i^{(k-1)}$, считая $y_i^{(0)} \equiv y_i$.

Матрица T строится построчно. Обозначим i-ю строку T через T_i.

for i **from** 1 **to** m **do**

(a) **if** $z_i = y_j$ **then** $T_i := (0, \ldots, 0, 1, 0, \ldots, 0)$, где «1» находится в j-й позиции;

(b) **if** $z_i = y_j^{(p)}$ **then** $T_i := T'_{i-1} + T_{i-1}A$;

5. **return** B, T.

Неизвестные строящихся в ходе алгоритма систем являются неизвестными исходной системы S и их производными, т. е. являются линейными формами из P. Выполнение условия шага 2a гарантирует, что очередная неизвестная y_i линейно независима с неизвестными системы S_s^{AB}, а значит, y_i линейно независима с неизвестными $S_{\{y_1\}}^{\text{AB}}$, т. е. $y_1, y'_1, \ldots, y_1^{(n-1)}$. Поэтому неизвестные $z_1, \ldots, z_n$ построенной после шага 2 системы будут соответствовать $y_1, y'_1, \ldots, y_1^{(n-1)}$. Таким образом, матрица B построенной системы будет иметь вид (4.2), причём $B_{1,1}$ будет соответствовать матрице системы $S_{\{y_1\}}^{\text{AB}}$. Порядок перебора неизвестных на шаге 2 алгоритма влияет лишь на матрицы $B_{2,1}$ и $B_{2,2}$ представления (4.2), а также на строки с $(n+1)$-й по m-ю матрицы T.

Матрица T является матрицей преобразования $z = Ty$, поэтому для всех i от 1 до m должно выполняться условие

$$z_i = T_i y. \tag{4.4}$$

При истинности условия шага 4a, выполнение (4.4) очевидно. В случае, если истинно условие шага 4b, $z_i = z'_{i-1} = (T_{i-1}y)' = T'_{i-1}y + T_{i-1}y' = T'_{i-1}y + T_{i-1}Ay$. Поэтому $T_i = T'_{i-1} + T_{i-1}A$.

Пример 17. Рассмотрим работу алгоритма на примере системы S вида (1.1), где

$$A = \begin{bmatrix} \frac{2x^4-2x+1}{x^5+2x^2-x} & \frac{3x}{x^4+2x-1} & \frac{3x^2}{x^4+2x-1} & -\frac{3x}{x^4+2x-1} \\ \frac{x^2}{x^4+2x-1} & \frac{1}{x^4+2x-1} & \frac{x}{x^4+2x-1} & -\frac{1}{x^4+2x-1} \\ \frac{x}{x^4+2x-1} & \frac{1}{x^5+2x^2-x} & -\frac{x^4+x-1}{x^5+2x^2-x} & -\frac{1}{x^5+2x^2-x} \\ \frac{2x^2-x}{x^4+2x-1} & -\frac{x^3}{x^4+2x-1} & -\frac{x^4}{x^4+2x-1} & \frac{x^3}{x^4+2x-1} \end{bmatrix},$$

$y = (y_1, y_2, y_3, y_4)^T$. Система $S^{\mathrm{AB}}_{\{y_1\}}$ имеет вид:

$$z' = \begin{bmatrix} 0 & 1 \\ 2/x^2 & 0 \end{bmatrix} z,$$

где $z = (y_1, y_1')^T$. Положим $s = \{y_1\}$ и начинаем последовательный перебор остальных неизвестных. $|S^{\mathrm{AB}}_s| < |S^{\mathrm{AB}}_{s\cup\{y_2\}}| = 3 < |S|$, поэтому добавляем y_2 к s ($s := \{y_1, y_2\}$) и переходим к рассмотрению y_3. $|S^{\mathrm{AB}}_{s\cup\{y_3\}}| = |S|$, поэтому добавляем к s неизвестную y_3 и заканчиваем цикл шага 2.

Система S^{AB}_s, где $s = \{y_1, y_2, y_3\}$, имеет вид:

$$z' = Bz = \left[\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ 2/x^2 & 0 & 0 & 0 \\ \hline 1/(3x^2) & 1/(3x) & 0 & 0 \\ 1/(3x^3) & 1/(3x^2) & 0 & -1/x \end{array}\right] z,$$

где вектор неизвестных $z = (y_1, y_1', y_2, y_3)^T$ и матрица B имеет требуемую блочную нижнетреугольную форму. Матрица преобразования, строящаяся на шаге 4 алгоритма по z и матрице исходной системы, примет вид:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{2x^4-2x+1}{(x^4+2x-1)x} & \frac{3x}{x^4+2x-1} & \frac{3x^2}{x^4+2x-1} & -\frac{3x}{x^4+2x-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Представленный алгоритм решает задачу факторизации дифференциальных систем лишь частично, а именно, только для систем, удовлетворяющих предусловию ($|S^{\mathrm{AB}}_{\{y_1\}}| < |S|$). Заметим, что проверка этого условия, достаточно проста (осуществима за полиномиальное время), поскольку заключается в не более чем m-кратном применении AB-алгоритма (вместо y_1 может выступать любая неизвестная системы).

Задача факторизации чаще рассматривается по отношению к скалярным дифференциальным операторам (см. [30–36]). Нужно заметить, что вычислительная сложность описываемых в указанных работах алгоритмов, довольно высокая. Очевидно, что для произвольного скалярного дифференциального оператора $L \in K[\partial]$ вида

$$L = \partial^m + a_{m-1}\partial^{m-1} + \cdots + a_0\partial^0$$

можно построить эквивалентную систему $\tilde{S}$: $y' = A_L y$, где $y = (y_1, \dots, y_m)^T$ и $A_L \in K^{m\times m}$ имеет вид

$$A_L = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{m-1} \end{bmatrix}, \tag{4.5}$$

и тем самым, свести задачу факторизации скалярного оператора к задаче факторизации дифференциальной системы. Алгоритм 6 в том виде, в котором он сформулирован, к системам с матрицами вида (4.5) не применим, поскольку $|\tilde{S}^{\mathrm{AB}}_{\{y_1\}}| = m$. Однако использование вместо y_1 другой неизвестной не обязательно приводит к системе размера m.

Пример 18. Рассмотрим скалярный дифференциальный оператор $L = \partial^3$. Матрица эквивалентной ему системы $\tilde{S}$ $y' = A_L y$, $y = (y_1, y_2, y_3)^T$, будет иметь вид:

$$A_L = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Матрица системы $\tilde{S}^{\mathrm{AB}}_{\{y_1\}}$ совпадает с A_L, но матрица системы $\tilde{S}^{\mathrm{AB}}_{\{y_2\}}$ имеет вид

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Имеются работы, рассматривающие задачу факторизации непосредственно для дифференциальных систем. Так, в работе [28] приводится алгоритм факторизации систем с небольшой сложностью, но описанный алгоритм работает лишь для случая, когда поле констант основного дифференциального поля имеет положительную характеристику, что не позволяет сравнивать его с алгоритмом 6. В работе [37] описывается алгоритм, проверяющий неприводимость системы, а в случае, если система приводима, осуществляющий её факторизацию. Сложность алгоритма оценивается трехкратной экспонентой от размера системы. Но поскольку алгоритм 6 применим не всегда, сравнение его с алгоритмом из [37] не является корректным.

4.2.2. Сателлитные неизвестные в неприводимых системах

Существуют дифференциальные системы, для которых для фиксированного множества выделенных неизвестных любая сателлитная неизвестная является одновременно и линейно сателлитной. Тем самым, понятия сателлитных и линейно сателлитных неизвестных для таких систем являются эквивалентными, т. е. для распознавания сателлитных неизвестных достаточно использовать алгоритм распознавания линейно сателлитных неизвестных. Как будет показано в этом разделе, таким классом систем является класс *неприводимых* систем.

Предложение 12. *Пусть S — неприводимая система. Тогда $|S^{\mathrm{AB}}_{\{y_i\}}| = |S|$ для всех $i = 1, \ldots, m$.*

Доказательство. Проведём доказательство от противного: пусть для некоторой неизвестной y_i выполняется условие

$$|S^{\mathrm{AB}}_{\{y_i\}}| = n < m = |S|. \tag{4.6}$$

Покажем, что в этом случае исходная система S является приводимой. Не ограничивая общности, можно считать, что условие (4.6) выполняется для неизвестной y_1, т. е. $|S^{\mathrm{AB}}_{\{y_1\}}| = n < m$.

Неизвестными системы $S^{\mathrm{AB}}_{\{y_1\}}$ являются $y_1, y_1', \ldots, y_1^{(n-1)}$. Поскольку y_1 и любая её производная принадлежат P (могут быть представлены в виде линейных форм от $y_1, \ldots, y_m$), неизвестные $S^{\mathrm{AB}}_{\{y_1\}}$ образуют базис подпространства в P. Этот базис может быть дополнен до базиса всего пространства P, причём только с помощью $y_2, \ldots, y_m$ и их производных (см. алгоритм 6). Использование линейных форм из этого базиса для введения новых неизвестных $z = (z_1, \ldots, z_m)^T$ соответствует замене $z = Ty$ с невырожденной матрицей $T \in K^{m \times m}$. Исходная система S преобразуется в систему $z' = Bz$, где B имеет вид (4.2) и $B_{1,1}$ является матрицей системы $S^{\mathrm{AB}}_{\{y_1\}}$. Тем самым система S приводима, что противоречит сделанному предположению. Значит, $|S^{\mathrm{AB}}_{\{y_i\}}| = m = |S|$. □

Таким образом, в случае, если система S неприводима, для любого непустого множества выделенных неизвестных s будет выполняться условие $|S^{\mathrm{AB}}_s| = |S|$, и оба алгоритма распознавания (алгоритмы 1 и 5) закончат свою работу с положительным ответом на шаге 2. Тем самым справедливы следующие утверждения.

Следствие 1. *Пусть S — неприводимая система. Тогда для любого множества s выделенных неизвестных любая невыделенная неизвестная является линейно сателлитной.*

Следствие 2. *Для неприводимых систем понятия сателлитных и линейно сателлитных неизвестных эквивалентны.*

Замечание 4. Неприводимые системы не исчерпывают множество дифференциальных систем, в которых понятия сателлитных и линейно сателлитных неизвестных совпадают. Существуют приводимые системы, в которых каждая сателлитная неизвестная для произвольного фиксированного множества выделенных неизвестных также является и линейно сателлитной. Примером может служить система:

$$y' = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} y,$$

$y = (y_1, y_2)^T$. Не сложно проверить, что неизвестная y_1 является линейно сателлитной (а, значит, и просто сателлитной) для $\{y_2\}$, а неизвестная y_2 является линейно сателлитной для $\{y_1\}$. Заменой

$$z = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} y$$

система приводится к виду

$$z' = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} z.$$

Таким образом, распознавание неприводимости системы не может быть построено на основе алгоритмов распознавания сателлитных неизвестных.

Ещё одно следствие, которое можно сделать из предложения 12, касается понятия *циклического вектора.*

Определение 15. Линейная форма $v \in P$ называется *циклическим вектором* для системы S, если $v, \Delta v, \ldots, \Delta^{m-1} v$ образуют базис P.

Циклический вектор используется для построения по заданной дифференциальной системе вида (1.1) эквивалентного скалярного дифференциального оператора. В [6, Разд. 4] отмечалось, что если 1-ранг системы совпадает с её размером, то неизвестная y_1 является циклическим вектором для этой системы. Заметим, что если множество s выделенных неизвестных содержит неизвестную, которая является циклическим вектором для системы, то любая невыделенная неизвестная будет линейно-сателлитной для s. Для неприводимых систем получаем следующее утверждение:

Следствие 3. *Любая неизвестная неприводимой системы S является для неё циклическим вектором.*

4.3. ПРИЛОЖЕНИЯ

4.3.1. Частичное решение дифференциальных систем

Компьютерная алгебра насчитывает довольно большое число алгоритмов, позволяющих находить решения дифференциальных систем, принадлежащих различным расширениям основного дифференциального поля K. В случае, когда компоненты решений дифференциальной системы принадлежат различным расширениям, эти алгоритмы не всегда применимы. При наличии дополнительной информации относительно некоторых компонент решений дифференциальной системы (каким дифференциальным расширениям они принадлежат), использование алгоритмов распознавания линейно сателлитных неизвестных совместно с AB-алгоритмом позволяет построить большее число компонент решений исходной системы.

Рассмотрим этот подход на примере.

Пример 19. Рассмотрим следующую линейную однородную дифференциальную систему

$$
\begin{cases}
-y_1' + xy_3' - xy_4' + y_3 + y_5 + y_6 = 0 \\
(x+1)xy_4' + xy_6' - y_5 - (x+1)y_6 = 0 \\
xy_4' + y_5' + y_6' - y_5 - y_6 = 0 \\
y_4' - y_3 = 0 \\
y_1 - y_5 - y_6 = 0 \\
y_1 + xy_2 = 0
\end{cases}
\tag{4.7}
$$

Система (4.7) может быть представлена в виде (1.12), где

$$
A_1 = \begin{bmatrix} -1 & 0 & x & -x & 0 & 0 \\ 0 & 0 & 0 & (x+1)x & 0 & x \\ 0 & 0 & 0 & x & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad A_0 = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 & -(x+1) \\ 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 1 \\ 1 & x & 0 & 0 & 0 & 0 \end{bmatrix},
$$

$y = (y_1, y_2, y_3, y_4, y_5, y_6)^T$. Пусть множество выделенных неизвестных содержит лишь неизвестную y_1, т. е. $s = \{y_1\}$. Как будет показано далее, компоненты решений системы (4.7), соответствующие неизвестной y_1, принадлежат расширению $\overline{\mathbb{Q}}(x, e^x)$ и могут быть представлены в виде рядов Лорана по степеням x. Другие компоненты решения, вообще говоря, этим свойством не обладают. Определяя линейно сателлитные неизвестные для s, мы можем построить частичные решения этой системы, используя процедуру построения Лорановых решений (например, процедуру `LinearFunctionalSystems:-LaurentSolution` из Maple).

Как видно из последнего уравнения (4.7), неизвестная y_2 является линейно сателлитной для $\{y_1\}$. Покажем, как этот факт может быть установлен алгоритмически. Алгоритм Extract, будучи применённым к системе (4.7) по отношению к множеству выделенных неизвестных $\{y_1, y_2\}$, даёт в результате следующую пару систем:

$$
S_d\colon \tilde{y}' = \begin{bmatrix} 1 - 1/x & 1 & 0 & 0 \\ 0 & -1/x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & x+1 & 0 & 1 \end{bmatrix} \tilde{y}, \qquad S_a\colon y_1 = -xy_2,
$$

где $\tilde{y} = (y_2, y_3, y_4, y_6)^T$. Поскольку неизвестная y_2 присутствует в алгебраической системе S_a, это означает, что она является линейно сателлитной для $\{y_1\}$, и соответствующие компоненты решений (4.7) можно представить в виде рядов Лорана по x.

Проверим, что неизвестная y_3 также является линейно сателлитной для $\{y_1\}$. Результатом применения алгоритма Extract к системе (4.7) по отношению к множеству выделенных неизвестных $\tilde{s} = \{y_1, y_3\}$ будет только нормальная дифференциальная система S_d, поскольку все неизвестные из $\tilde{s}$ войдут в $\tilde{y}$ (алгебраическая система S_a в этом случае является пустой):

$$\tilde{y}' = \begin{bmatrix} 1 & -x & 0 & 0 \\ 0 & -1/x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1/x & 1 & 0 & 1 \end{bmatrix} \tilde{y}, \quad \tilde{y} = (y_1, y_3, y_4, y_5)^T. \tag{4.8}$$

Остаётся проверить, что y_3 линейно сателлитная для $\{y_1\}$ в (4.8). АВ-алгоритм позволяет получить матрицы

$$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \quad \text{и} \quad \begin{bmatrix} 1 & -x \\ 0 & -1/x \end{bmatrix}$$

систем $(S_d)_s^{\mathrm{AB}}$ и $(S_d)_{\tilde{s}}^{\mathrm{AB}}$ соответственно. Из чего убеждаемся, что y_3 действительно является линейно сателлитной для $\{y_1\}$. Системы $(S_d)_{\{y_1\}}^{\mathrm{AB}}$ и $(S_d)_{\{y_1,y_3\}}^{\mathrm{AB}}$ могут быть легко решены, и мы получаем общее решение для компонент, соответствующих неизвестным y_1, y_2 и y_3, в следующем виде:

$$y_1 = C_1 e^x + C_2, \quad y_2 = -(C_1 e^x + C_2)/x, \quad y_3 = C_2/x, \tag{4.9}$$

где C_1, C_2 произвольные постоянные. Применение процедуры построение решений в виде рядов непосредственно к системе (4.7) или (4.8) не позволит получить выражения (4.9) в виду присутствия в системе неизвестных y_4, y_5, y_6, поскольку им соответствующие компоненты решений не могут быть представлены в виде рядов Лорана в общем случае.

4.3.2. Частичная устойчивость автономных систем

Рассмотрим классическую постановку задачи устойчивости по части переменных (см. [5]): пусть имеется система обыкновенных дифференциальных уравнений:

$$\dot{\mathbf{x}} = \mathbf{X}(t, \mathbf{x}), \quad \mathbf{X}(t, \mathbf{0}) \equiv \mathbf{0}, \tag{4.10}$$

где $\mathbf{x} = (y_1, \ldots, y_k, z_1, \ldots, z_p)^T = (\mathbf{y}^T, \mathbf{z}^T)^T$, $m > 0$, $p \geq 0$, $m = k + p$.

В системе (4.10) переменные, входящие в фазовый вектор $\mathbf{x}$, разбиты на две группы:

– переменные $y_1, \ldots, y_k$, по отношению к которым исследуется устойчивость положения равновесия (*невозмущённого движения*) $\mathbf{x} = \mathbf{0}$;

– оставшиеся переменные $z_1, \ldots, z_p$.

Конкретный выбор такого разбиения может быть различным (зависит от задачи), но, как правило, предполагается, что выбор $\mathbf{y}$-переменных *уже сделан*. Переменные $z_1, \ldots, z_p$ часто называют *«неконтролируемыми»* переменными; при изучении задачи частичной устойчивости поведение этих переменных системы (4.10) исследователя в принципе не интересует.

Обозначим $\mathbf{x}(t) = \mathbf{x}(t; t_0, \mathbf{x}_0)$ решение системы (4.10), определённое начальными условиями $\mathbf{x}_0 = \mathbf{x}(t_0; t_0, \mathbf{x}_0)$.

В теории устойчивости по отношению к части переменных обычно делаются следующие предположения:

а) Правые части системы (4.10) в области

$$t \geq 0, \quad \|\mathbf{y}\| \leq h, \quad \|\mathbf{z}\| < \infty,$$

$$\|\mathbf{y}\| = \left(\sum_{i=1}^{k} y_i^2 \right)^{1/2}, \quad \|\mathbf{z}\| = \left(\sum_{j=1}^{p} z_j^2 \right)^{1/2}, \quad \|\mathbf{x}\| = (\|\mathbf{y}\|^2 + \|\mathbf{z}\|^2)^{1/2}$$

непрерывны и удовлетворяют *условиям единственности* решений (например, локальному условию Липшица).

б) Решения системы (4.10) $\mathbf{z}$-*продолжимы*. Это значит, что любое решение $\mathbf{x}(t)$ определено при всех $t \geq 0$, при которых $\|\mathbf{y}(t)\| \leq h$.

Определение 16 ([38, 39])**.** Невозмущённое движение $\mathbf{x} = \mathbf{0}$ системы (4.10) называется:

– *устойчивым по отношению к* $y_1, \ldots, y_k$ (кратко $\mathbf{y}$-устойчивым), если для любых чисел $\varepsilon > 0$, $t_0 \geq 0$ найдётся число $\delta(\varepsilon, t_0) > 0$ такое, что из $\|\mathbf{x}_0\| < \delta$ следует $\|\mathbf{y}(t; t_0, \mathbf{x}_0)\| < \varepsilon$ при всех $t \geq t_0$;

– *асимптотически* $\mathbf{y}$*-устойчивым*, если оно $\mathbf{y}$-устойчиво и, кроме того, для каждого $t_0 \geq 0$ существует число $\Delta(t_0) > 0$ такое, что решение $\mathbf{x}(t; t_0, \mathbf{x}_0)$ с $\|\mathbf{x}_0\| < \Delta$ удовлетворяет условию

$$\lim_{t\to\infty} \|\mathbf{y}(t; t_0, \mathbf{x}_0)\| = 0.$$

При этом область $\|\mathbf{x}_0\| < \Delta$ лежит в *области* $\mathbf{y}$*-притяжения* невозмущённого движения $\mathbf{x} = \mathbf{0}$ для начального момента t_0.

Будем рассматривать наиболее простой случай, когда система (4.10) является линейной и автономной. Тогда её можно переписать в виде

$$\dot{\mathbf{x}} = A\mathbf{x}, \tag{4.11}$$

где A — постоянная матрица размера $m \times m$, т. е. получаем нормальную дифференциальную систему вида (1.1).

Возвращаясь к обозначениям, использованным в начале главы, задачу частичной устойчивости для линейных автономных дифференциальных систем можно переформулировать в следующем виде. Рассматривается нормальная дифференциальная система (1.1), неизвестные которой являются функциями времени ($y = y(t) = \big(y_1(t), \ldots, y_m(t)\big)^T$), и начальное условие $y(0) = (0, \ldots, 0)^T$. Некоторые неизвестные, образующие множество s, являются выделенными. Для простоты будем считать, что выделены первые k неизвестных: $s = \{y_1, \ldots, y_k\}$. Неизвестные из s играют роль той части неизвестных, по отношению к которым исследуется устойчивость положения равновесия невозмущённого движения $y = (0, \ldots, 0)^T$. Невыделенные неизвестные являются неконтролируемыми.

Предложение 13. *Пусть невозмущённое движение* $y = (0, \ldots, 0)^T$ *автономной системы (1.1) устойчиво (асимптотически устойчиво) по отношению* *к части неизвестных s и пусть невыделенная неизвестная $y_j \notin s$ является линейно сателлитной для s. Тогда невозмущённое движение $y = (0, \ldots, 0)^T$ будет также устойчиво (асимптотически устойчиво) по отношению к части неизвестных $s \cup \{y_j\}$.*

Доказательство. Поскольку система является автономной, каждая компонента любого решения представима в виде

$$y_i(t) = \sum_{n=1}^{m_i} p_n(t) e^{\lambda_n t}, \tag{4.12}$$

где $p_n(t)$ — ненулевые многочлены переменной t, $\lambda_n \in \mathbb{C}$. Ввиду частичной устойчивости, для каждой компоненты из s все λ_n, входящие в представление (4.12), удовлетворяют условиям

$$\begin{aligned} &\operatorname{Re}\lambda_n \leq 0, \ \ \text{если} \ \ \deg p_n(t) = 0, \\ &\operatorname{Re}\lambda_n < 0, \ \ \text{если} \ \ \deg p_n(t) > 0 \end{aligned} \tag{4.13}$$

(в случае асимптотической устойчивости $\operatorname{Re}\lambda_n < 0$ не зависимо от $\deg p_j$). Поскольку y_j является линейно сателлитной для s, то для любого решения его j-я компонента представима в виде линейной комбинации (с постоянными коэффициентами) компонент этого решения, соответствующих неизвестным из s, и их производным. Из представления (4.12) следует, что

$$y_i'(t) = \sum_{n=1}^{m_i} \bigl(p_n'(t) + \lambda_n p_n(t)\bigr) e^{\lambda_n t} = \sum_{n=1}^{m_i} \tilde{p}_n(t) e^{\lambda_n t},$$

т. е. показатели экспонент любой производной совпадают с показателями экспонент самой компоненты. Таким образом, условия (4.13) выполняются для компоненты y_j, и невозмущённое движение $y = (0, \ldots, 0)^T$ устойчиво относительно y_j. $\square$

ГЛАВА 5. ПРОГРАММНЫЙ КОМПЛЕКС СИМВОЛЬНЫХ ВЫЧИСЛЕНИЙ

Разработанные и представленные в предыдущих главах алгоритмы работы с линейными дифференциальными системами с выделенными неизвестными были реализованы на языке системы компьютерной алгебры Maple в виде программного комплекса из двух компонент: 1) программный модуль, содержащий процедуру `Extract`, реализующую одноимённый алгоритм и 2) пакет `Satellite`, предоставляющий процедуры распознавания сателлитных и линейно сателлитных неизвестных.

5.1. ПРОЦЕДУРА EXTRACT

Разработка процедуры `Extract` была ориентирована на расширение области применения процедуры `ReducedSystem` (реализация AB-алгоритма) из пакета `Consequences` стандартного пакета `OreTools`. Процедура `ReducedSystem` позволяет работать лишь с нормальными линейными дифференциальными системами, т. е. с системами вида (1.1). Разработанная процедура `Extract` позволяет обеспечить обработку общего случая линейных дифференциальных систем произвольного порядка, т. е. систем вида (1.9). Интерфейс процедуры `Extract`, а именно входные и выходные данные, согласованы с интерфейсом процедуры `ReducedSystem`. Исходный код процедуры `Extract` вместе с примерами работы и описанием доступны в Интернете по адресу `http://www.ccas.ru/ca/extract`.

Реализация процедуры `Extract` опирается на возможности, предоставляемые пакетом `OreTools`, при этом этапы самого алгоритма (исключения неизвестных) сводятся к умножению операторной матрицы исходной системы на унимодулярные матрицы вида (2.2), как это было описано в разделе 2.1. Дифференциальные операторы задаются с помощью специальных структур *OrePoly*, соответствующих внутреннему представлению некоммутативных полиномов Оре. Пакет `OreTools` предоставляет все необходимые операции над структурами *OrePoly*.

На вход процедуре `Extract` поступает произвольная система вида (1.9), задаваемая матричными коэффициентами, и множество индексов выделенных неизвестных. Если исходная система имеет полный ранг, результатом работы процедуры будет матрица нормальной дифференциальной системы для части выделенных неизвестных и матрица, с помощью которой значения остальных выделенных неизвестных могут быть получены по значениям выделенных неизвестных, попавших в нормальную дифференциальную систему. В противном случае работа процедуры завершается выдачей сообщения об ошибке.

Алгоритм Extract, описанный ранее, работает с системами вида (2.1). Если исходная система, поступающая на вход процедуре `Extract` имеет порядок выше первого, то на предварительном этапе она преобразуется к системе первого порядка (1.12). Далее при необходимости выполняется приведение системы по строкам с помощью алгоритма Row-Reduction, реализованного для случая дифференциальных систем первого порядка в виде вспомогательной процедуры `RowReduction`, входящей в программный модуль вместе с процедурой `Extract`.

Процедура `Extract` может вызываться со следующими входными параметрами:

```
Extract(A1::Matrix, A0::Matrix, s::set, R)
Extract(M::list(Matrix), s::set, R)
```

`A1`, `A0` — матрицы исходной системы, `s` — множество индексов выделенных неизвестных, `R` — кольцо полиномов Оре, соответствующее дифференциальному случаю и выбираемое с помощью процедуры `SetOreRing` из пакета `OreTools`. Вторая форма вызова позволяет работать с системами (1.9) произвольного порядка, при этом система задаётся списком матричных коэффициентов: $[A_r, A_{r-1}, \dots, A_1, A_0]$.

Результатом работы процедуры `Extract` является список из четырех элементов:

1) матрица результирующей нормальной дифференциальной системы;

2) множество пар, устанавливающих соответствие между индексами выделенных неизвестных исходной системы и индексами неизвестных результирующей дифференциальной системы;

3) матрица линейного преобразования (матрица T, строящаяся на третьем этапе алгоритма), с помощью которой выделенные неизвестные исходной системы, не вошедшие в новую дифференциальную систему, могут быть вычислены по значениям выделенных неизвестных, вошедших в новую дифференциальную систему;

4) множество пар, устанавливающих соответствие между индексами выделенных неизвестных исходной системы и строками матрицы линейного преобразования.

В случае, если в результирующую систему попадают все выделенные неизвестные, элементы 3 и 4 отсутствуют.

Пример 20. Пусть $K = \overline{\mathbb{Q}}(x)$, $\partial = \frac{d}{dx}$, $y = (y_1, y_2, y_3, y_4)^T$. Для системы

$$\begin{bmatrix} 0 & 0 & x & 0 \\ x & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} y' + \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & -x^3 & 0 \\ 0 & 0 & -1 & 0 \\ x & -1 & 0 & 0 \end{bmatrix} y = 0 \tag{5.1}$$

будем интересоваться рациональными решениями для компонент y_1 и y_2.

Перед обращением к `Extract` необходимо ввести матрицы исходной системы (5.1) и произвести ряд настроек:

> `with(OreTools):`

> `R := SetOreRing(`x`, 'differential'):`

> `A1 :=` $\begin{bmatrix} 0 & 0 & x & 0 \\ x & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$:

> `A0 :=` $\begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & -x^3 & 0 \\ 0 & 0 & -1 & 0 \\ x & -1 & 0 & 0 \end{bmatrix}$:

```
> Extract(A1, A0, {1,2}, R)
```

$$\left[\left[\begin{array}{ccc} 2/x & x^3 & 0 \\ 0 & -1/x & 0 \\ 0 & 1 & 0 \end{array}\right], \{[2,1]\}, \left[\begin{array}{c} 1/x \end{array}\right], \{[1,1]\}\right]$$

Результаты процедуры `Extract` имеют следующую трактовку: по исходной системе (5.1) построена новая нормальная дифференциальная система

$$\tilde{y}' = \left[\begin{array}{ccc} 2/x & x^3 & 0 \\ 0 & -1/x & 0 \\ 0 & 1 & 0 \end{array}\right] \tilde{y}, \tag{5.2}$$

в которой выделенная неизвестная y_2 стала первой компонентой $\tilde{y}$. Матрица линейного преобразования T в данном случае имеет размер 1×1, поэтому выделенная неизвестная y_1, не вошедшая в $\tilde{y}$, линейно выражается через y_2 следующим образом:

$$y_1 = T y_2 = \left[\begin{array}{c} 1/x \end{array}\right] y_2 = \frac{y_2}{x}.$$

Далее, если мы интересуемся рациональными y_2, то к системе (5.2) нужно применить процедуру `ReducedSystem` (реализация AB-алгоритма), доступную из подпакета `Consequences`:

```
> Consequences:-ReducedSystem(%[1], {1}, R)
```

$$\left[\left[\begin{array}{cc} 0 & 1 \\ -6/x^2 & 4/x \end{array}\right], \{[1,1]\}\right]$$

Тем самым получаем новую систему

$$z' = \left[\begin{array}{cc} 0 & 1 \\ -6/x^2 & 4/x \end{array}\right] z,$$

неизвестные которой $z = (y_2, y_2')^T$.

Применяя процедуру поиска рациональных решений (процедура `RationalSolution` из пакета `LinearFunctionalSystems`) к последней системе, находим $y_2 = C_1 x^2 + C_2 x^3$, где C_1, C_2 — произвольные постоянные. Используя найденное значение y_2, для y_1 получаем

$$y_1 = \frac{y_2}{x} = C_1 x + C_2 x^2.$$

Заметим, что применение процедуры поиска рациональных решений непосредственно к системе (5.1), позволит найти $y_1 = C_1 x$, $y_2 = C_1 x^2$, т. е. слагаемые с C_2 найдены не будут.

Рассмотрим работу процедуры `Extract` с системами высоких порядков. Как было отмечено в разделе 1.3, любую систему произвольного порядка $r > 1$ можно свести к дифференциальной системе первого порядка, при этом вектор неизвестных будет содержать mr компонент, где наряду с исходными неизвестными $y_1, \ldots, y_m$ будут также присутствовать их производные до порядка $(r-1)$ включительно. При этом, после перехода к новому вектору неизвестных, множество выделенных неизвестных не меняется, т. е. выделенными становятся компоненты нового вектора неизвестных Y, соответствующие выделенным компонентам вектора y.

Пример 21. Рассмотрим следующую дифференциальную систему второго порядка:

$$A_2 y'' + A_1 y' + A_0 y = 0 \tag{5.3}$$

где

$$A_2 = \begin{bmatrix} x^2 & 0 \\ 0 & 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} x & -2x^3 \\ 0 & x \end{bmatrix}, \quad A_0 = \begin{bmatrix} -1 & -2x^2 \\ 0 & 1 \end{bmatrix}.$$

Здесь вектор неизвестных y состоит из двух компонент: $y = (y_1, y_2)^T$. Пусть выделена первая компонента вектора неизвестных — y_1. При переходе к новому вектору неизвестных $Y = (y_1, y_2, y_1', y_2')^T$, система (5.3) перепишется в виде:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & x^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} Y' + \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & -2x^2 & x & -2x^3 \\ 0 & 1 & 0 & x \end{bmatrix} Y = 0,$$

где по-прежнему выделенной будет первая компонента вектора неизвестных. Обращение к процедуре `Extract` будет осуществляться со следующими параметрами (множество индексов выделенных неизвестных состоит из одного элемента):

```
> Extract([A2,A1,A0], {1}, R)
```

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & -2/x & 0 \\ 1/x^2 & 0 & -1/x \end{bmatrix}, \{[1,1]\}$$

В этом случае удалось построить нормальную дифференциальную систему, куда вошла единственная выделенная неизвестная, поэтому матрица T отсутствует.

5.1.1. Оценка быстродействия

Для оценки быстродействия реализации процедуры `Extract` была написана специальная процедура `GenDAE1(m)` для генерации линейных дифференциально-алгебраических систем вида (2.1) в виде двух полиномиальных матриц A_1 и A_0 размера $m \times m$. Элементы матриц генерируются с помощью встроенной процедуры генерации случайных полиномов `randpoly` с указанным ограничением на степени — не больше третьей:

```
Matrix(m, (i,j)->randpoly([x],degree=rand(0..3)()))
```

Для ведущей матрицы A_1 проверяется условие вырожденности

```
LinearAlgebra:-Rank(A1) < m
```

и если оно не выполняется, случайная строка матрицы заменяется линейной комбинацией других строк матрицы, что гарантирует вырожденность получаемой матрицы.

Замер времени осуществлялся с использованием встроенной в Maple процедуры `time()` по заранее подготовленному массиву `data` тестовых данных с помощью следующих операторов:

```
start:=time();
for sys in data do
  Extract(sys[1], sys[2], s, R)
end do;
evalf((time()-start)/nops(data))
```

Полученные средние значения времён[1] для различных значений m и различных множеств выделенных неизвестных s представлены в таблице 1.

Таблица 1 – Время в секундах работы процедуры `Extract`

m	$s = \{1\}$	$s = \{1, 2\}$	$s = \{1..m\}$
3	0.034	0.035	0.027
4	0.055	0.055	0.040
5	0.097	0.101	0.084
6	0.165	0.170	0.120
7	0.240	0.210	0.190
8	0.340	0.400	0.320
9	0.640	0.630	0.490
10	0.860	0.920	0.690

Полученные результаты согласуются с теоретической оценкой $O(m^3)$, полученной в главе 2. Меньшее время, требующееся для обработки случая $s = \{1..m\}$ объясняется тем, что для этого случая первый этап алгоритма (исключение невыделенных неизвестных) не требуются.

5.2. ПАКЕТ SATELLITE

Пакет `Satellite`, исходный код которого вместе с примерами и описанием доступны в Интернете по адресу `http://www.ccas.ru/ca/satellite`, содержит 10 процедур, из которых экспортируются две процедуры, реализующие частичные алгоритмы распознавания сателлитных неизвестных, и две процедуры распознавания линейно сателлитных неизвестных.

5.2.1. Реализация частичных алгоритмов распознавания сателлитных неизвестных

Частичный алгоритм распознавания сателлитных неизвестных (алгоритм 4) для нормальных дифференциальных систем был реализован в виде про-

[1] Вычисления проводились в Maple 17, Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz, 4GB ОЗУ.

цедуры `Satellite:-Testing(A, s, v)` пакета `Satellite`. Эта процедура принимает на вход матрицу `A` линейной дифференциальной системы вида (1.1), множество индексов выделенных неизвестных `s` и индекс тестируемой невыделенной неизвестной `v`. Результатом её работы является значение `true`, если удалось определить, что тестируемая неизвестная является сателлитной для указанного множества выделенных неизвестных, или `FAIL`, если не удалось.

Процедура `Testing` в своей работе использует обращения к процедуре `ReducedSystem` пакета `OreTools:-Consequences`).

Наиболее сложное место алгоритма (и реализации) — проверка эквивалентности систем (шаг 5 алгоритма 4). Она реализована в виде специальной процедуры и осуществляется путём построения рациональной матрицы преобразования решений (матрица T из определения 5) и проверки ее обратимости. Для построения матрицы преобразования решений используется процедура поиска рациональных решений `RationalSolution` пакета `LinearFunctionalSystems`, входящего в стандартную поставку Maple.

На основе процедуры `Testing` реализована процедура `Satellite:-Determination(M, s)`, которая может быть использована для построения множества сателлитных неизвестных. Реализация процедуры `Determination`, основывается на алгоритме 3, в котором на шаге 4 используется частичный алгоритм распознавания сателлитных неизвестных (процедура `Testing`). Тем самым процедура `Determination` обеспечивает лишь частичное решение задачи.

Входными параметрами процедуры являются:

- `M` — матрица нормальной дифференциальной системы вида (1.1) или список $[A_r, A_{r-1}, \dots, A_0]$, элементами которого являются матрицы линейной однородной дифференциальной системы вида (1.9);
- `s` — множество индексов выделенных неизвестных.

Результатом работы `Determination` является список из трёх множеств:

- первое множество содержит индексы невыделенных неизвестных, которые являются сателлитными,
- второе множество содержит индексы невыделенных неизвестных, для которых не удалось определить, являются ли они сателлитными или нет;

- третье множество содержит индексы невыделенных неизвестных, которые не являются сателлитными (условие шага 6 алгоритма 3 используется в качестве критерия и позволяет распознавать случаи, когда неизвестная не является сателлитной).

Процедура использует сведение дифференциальной системы высокого порядка к дифференциальной системе первого порядка, а также процедуру `Extract` для работы с дифференциально-алгебраическими системами.

Пример 22. Рассмотрим дифференциальную систему из примера 10. Используя реализацию описанного частичного алгоритма определения множества сателлитных неизвестных, можно получить следующие результаты:

`> A:=` $\begin{bmatrix} 1/x & 0 & 0 \\ 3 & -1/x & 0 \\ 0 & 0 & 1 \end{bmatrix}$:

```
> Satellite:-Determination(A, {1})
```

$$[\{2\}, \{3\}, \{\}]$$

Тем самым, удалось определить, что неизвестная с индексом 2 (y_2) является сателлитной для множества выделенных неизвестных $\{y_1\}$. Для y_3 используемый метод не позволил определить, является она сателлитной для $\{y_1\}$ или нет.

```
> Satellite:-Determination(A, {3})
```

$$[\{1, 2\}, \{\}, \{\}]$$

При указании множества выделенных неизвестных $\{y_3\}$, процедуре удалось определить, что и y_1, и y_2 являются сателлитными.

Пример 23. Для демонстрации работы процедуры `Determination` с линейными однородными дифференциальными системами высоких порядков, рассмотрим следующую систему:

$$\begin{bmatrix} x & 0 \\ 0 & 0 \end{bmatrix} y'' + \begin{bmatrix} 1 & x \\ 1 & 0 \end{bmatrix} y' + \begin{bmatrix} 0 & 1 \\ x & x \end{bmatrix} y = 0, \tag{5.4}$$

в которой $y = (y_1, y_2)^T$. Будем предполагать, что множество выделенных неизвестных состоит из одной неизвестной y_1.

`> A2:=` $\begin{bmatrix} x & 0 \\ 0 & 0 \end{bmatrix}$:

```
> A1:=
```
$$\begin{bmatrix} 1 & x \\ 1 & 0 \end{bmatrix}:$$

```
> A0:=
```
$$\begin{bmatrix} 0 & 1 \\ x & x \end{bmatrix}:$$

```
> Satellite:-Determination([A2,A1,A0],{1})
```

$$[\{2\}, \{\}, \{\}]$$

И тем самым определено, что неизвестная y_2 в системе (5.4) является сателлитной для множества выделенных неизвестных $\{y_1\}$.

5.2.2. Реализация алгоритмов распознавания линейно сателлитных неизвестных

Процедуры, реализующие алгоритмы работы с линейно сателлитными неизвестными, являются частью пакета `Satellite`.

Алгоритм 5 реализован на основе имеющейся в Maple реализации AB-алгоритма (процедура `OreTools:-Consequences:-ReducedSystem`) в виде процедуры `LinSatTesting`. Вместо результатов «ДА» и «НЕТ» используются логические значения `true` и `false` соответственно. Процедура принимает на вход следующие параметры:

```
LinSatTesting(A::Matrix, s::set, v::posint),
```

где `A` — матрица нормальной дифференциальной системы, `s` — множество положительных целых чисел — индексы выделенных неизвестных и `v` — индекс тестируемой неизвестной. Индексами неизвестных являются целые числа от 1 до m, где m — размер системы. При этом индекс тестируемой неизвестной `v` не должен входить в `s`.

Процедура `LinSatTesting` имеет ещё одну форму вызова:

```
LinSatTesting(A::Matrix, s::set, v::posint, 'T'),
```

где четвёртый параметр является выходным (допустимо передавать в качестве него только имена переменных), в котором, в случае, если неизвестная с индексом `v` является линейно сателлитной для множества выделенных неизвестных `s`, возвращается матрица коэффициентов линейного выражения тестируемой неизвестной через выделенные неизвестные и их производные. Строки матрицы соответствуют выделенным неизвестным, столбцы — порядкам производных (первый столбец — нулевой порядок, т. е. содержит коэффициенты при самих неизвестных, второй столбец — коэффициенты при производных первого порядка и т. д.).

Построение матрицы коэффициентов линейного выражения осуществляется локальной для модуля `Satellite` процедурой `LinExpr`. Для этого на основе результата работы `ReducedSystem` формируется матрица, столбцы которой являются базисом подпространства линейных форм от неизвестных системы (подпространства в P), содержащего линейно сателлитную неизвестную. После чего искомые коэффициенты получаются в результате решения системы линейных алгебраических уравнений (процедура `LinearSolve` пакета `LinearAlgebra`).

Пример 24. Рассмотрим систему из примера 14. Как было показано, неизвестная y_2 является линейно сателлитной для $\{y_3\}$. Это означает, что вторая компонента любого решения может быть выражена через третью компоненту и её производные. Процедура `LinSatTesting` позволяет получить это выражение.

```
> A := Matrix([[2,1,-2], [4,6,-4], [2,3,-2]])/(2*x)
```

$$\begin{bmatrix} 1/x & 1/(2x) & -1/x \\ 2/x & 3/x & -2/x \\ 1/x & 3/(2x) & -1/x \end{bmatrix}$$

```
> Satellite:-LinSatTesting(A, {3}, 2, 'B')
```

$$true$$

```
> B
```

$$\begin{bmatrix} 0 & 2x & -x^2 \end{bmatrix}$$

Матрица B имеет размер 1×3, что даёт следующее выражение для второй компоненты решений:

$$y_2 = B \cdot \begin{bmatrix} y_3 \\ y_3' \\ y_3'' \end{bmatrix} = 2x\, y_3' - x^2\, y_3''.$$

Также с помощью `LinSatTesting` можно проверить, что неизвестная y_3 является линейно сателлитной для $\{y_1, y_2\}$.

```
> Satellite:-LinSatTesting(A, {1,2}, 3, 'B')
```

$$true$$

```
> B
```

$$\begin{bmatrix} 1 & -x \\ 1/2 & 0 \end{bmatrix}$$

Для любого решения системы (4.1) его третья компонента может быть выражена через первую и вторую компоненты и их производные:

$$y_3 = B_{1,*} \cdot \begin{bmatrix} y_1 \\ y_1' \end{bmatrix} + B_{2,*} \cdot \begin{bmatrix} y_2 \\ y_2' \end{bmatrix} = y_1 - x\, y_1' + y_2/2.$$

На основе процедуры `LinSatTesting`, которая позволяет работать лишь с нормальными дифференциальными системами, реализована процедура `LinearlySatellite(M, s::set)`, также входящая в пакет `Satellite` и позволяющая строить множество всех линейно сателлитных неизвестных для указанного множества выделенных неизвестных. Процедура `LinearlySatellite` использует в своей работе реализацию алгоритма Extract и позволяет работать не только с нормальными системами, но и с дифференциальными системами вида (1.9).

Параметрами процедуры `LinearlySatellite` являются:

- `M` — матрица нормальной дифференциальной системы вида (1.1) или список $[A_r, A_{r-1}, \ldots, A_0]$, элементами которого являются матрицы линейной однородной дифференциальной системы вида (1.9);

- `s` — множество индексов выделенных неизвестных.

Результатом работы процедуры `LinearlySatellite` является множество индексов невыделенных неизвестных, которые являются линейно сателлитными для выделенных неизвестных с индексами из `s`.

Пример 25. Продемонстрируем работу процедуры `Satellite:-Linearly-Satellite` на дифференциальной системе из примера 19. Система задаётся своими матрицами:

```
> A1 := Matrix([[-1, 0, x, -x, 0, 0],
               [0, 0, 0, (x+1)*x, 0, x],
               [0, 0, 0, x, 1, 1],
               [0, 0, 0, 1, 0, 0],
               [0, 0, 0, 0, 0, 0],
               [0, 0, 0, 0, 0, 0]]):
> A0 := Matrix([[0, 0, 1, 0, 1, 1],
               [0, 0, 0, 0, -1, -(x+1)],
               [0, 0, 0, 0, -1, -1],
               [0, 0, -1, 0, 0, 0],
               [-1, 0, 0, 0, 1, 1],
               [1, x, 0, 0, 0, 0]]):
```

Вызов процедуры `LinearlySatellite` даёт следующий результат:

```
> Satellite:-LinearlySatellite([A1,A0], {1})
                                   {2,3}
```

что означает, что неизвестные y_2 и y_3 являются линейно сателлитными для $s = \{y_1\}$, как это было установлено в примере 19.

5.2.3. Оценка быстродействия

Для получения оценок времени работы процедуры `LinearlySatellite` была задействована та же процедура `GenDAE1` для генерации тестовых линейных дифференциально-алгебраических систем первого порядка, что использовалась для генерации тестовых данных для оценки быстродействия процедуры `Extract`. Работа с дифференциально-алгебраическими системами гарантированно потребует обращение к процедуре `Extract`.

Замер времени осуществлялся с использованием встроенной в Maple процедуры `time()` по заранее подготовленному массиву `data` тестовых данных с помощью следующих операторов:

```
mm:=0;
for sys in data do
  t:=time();
  Satellite:-LinearSatellite([sys[1], sys[2]], s)
  t:=time()-t;
  if mm<t then mm:=t end if;
end do;
evalf(mm,3)
```

Полученные максимальные значения времён[2] для различных значений m и различных множеств выделенных неизвестных s представлены в таблице 2.

Таблица 2 – Время в секундах работы процедуры `LinearlySatellite`

m	$s=\{1\}$	$s=\{1,2\}$	$s=\{1,2,3\}$
3	0,14	0,08	—
4	0,31	0,20	0,09
5	1,03	0,70	0,53
6	1,92	1,67	1,17
7	6,94	4,38	2,96
8	23,90	19,70	9,75
9	58,30	39,70	27,70
10	136	101	69

Поскольку алгоритм распознавания линейно сателлитных неизвестных заключается в не более чем двукратном применении AB-алгоритма, основное время работы процедуры `LinearlySatellite` тратиться на обращение к процедуре `ReducedSystem`. Перебор всех невыделенных неизвестных, производимый процедурой `LinearlySatellite`, даёт оценку сложности $O(m^5)$, что согласуется с полученными результатами.

[2]Вычисления проводились в Maple 2015, Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz, 6GB ОЗУ.

ЗАКЛЮЧЕНИЕ

Основные результаты диссертационной работы:

- Разработан алгоритм Extract, позволивший обобщить AB-алгоритм на случай линейных однородных дифференциальных систем произвольного порядка.

- Разработаны алгоритмы распознавания сателлитных неизвестных в линейных дифференциальных системах с выделенными неизвестными.

- На основе предложенных алгоритмов разработан программный комплекс символьных вычислений в среде компьютерной алгебры Maple. Исходный код программ доступен по адресу `http://www.ccas.ru/ca`.

Дальнейшее развитие работы может вестись по разным направлениям, среди которых можно выделить совершенствование реализации представленных и используемых алгоритмов в части обработки коэффициентов более широкого класса. Также возможно дальнейшее развитие в части приложений понятия сателлитных неизвестных — возможность использования алгоритмов распознавания сателлитных неизвестных для повышения эффективности известных алгоритмов обработки систем дифференциальных уравнений.

Список литературы

[1] С. А. Абрамов. Элементы компьютерной алгебры линейных обыкновенных дифференциальных, разностных и q-разностных операторов. М.: МЦНМО, 2012, 127 с.

[2] В. М. Глушков, В. Г. Боднарчук, Т. А. Гринченко, А. А. Дородницина, В. П. Клименко, А. А. Летичевский, С. Б. Погребинский, А. А. Стогний, Ю. С. Фишман. АНАЛИТИК — алгоритмический язык для описания процессов с использованием аналитических преобразований // Кибернетика. 1971, № 3, с. 102–134.

[3] А. П. Будник, Е. В. Гай, Н. С. Работнов, А. В. Климов, В. Ф. Турчин, И. Б. Щенков. Базисные волновые функции и матрицы операторов в коллективной модели ядра // Ядерная физика. 1971, т. 14, вып. 2, с. 304–314.

[4] Е. В. Панкратьев. Элементы компьютерной алгебры: учебное пособие. М.: Интернет-Ун-т информ. технологий: БИНОМ. Лаборатория знаний, 2010, 247 с.

[5] В. И. Воротников, В. В. Румянцев. Устойчивость и управление по части координат фазового вектора динамических систем: теория, методы и приложения. М.: Научный мир, 2001, 320 с.

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

[7] А. А. Панферов. Системы дифференциальных уравнений с выделенной частью неизвестных // Программирование. 2015, № 2, с. 26–36.

[8] А. А. Панфёров. О разбиениях множества выделенных неизвестных в линейных дифференциально-алгебраических системах // Программирование. 2016, № 2, с. 41–48.

[9] A. A. Panferov. Selected and satellite unknowns in linear differential systems // Advances in Applied Mathematics. 2017, vol. 85, p. 1–11.

[10] А. А. Панфёров. Частичные алгоритмы определения сателлитных неизвестных // Программирование. 2016, № 2, с. 72–80.

[11] А. А. Панфёров. Сателлитные неизвестные в неприводимых дифференциальных системах // Программирование. 2018, № 2, с. 42–50.

[12] A. A. Panferov. On determination of satellite unknowns in linear differential systems. // *Компьютерная алгебра. Сборник научных статей. Труды международной конференции «Компьютерная алгебра». Москва, 29 июня – 2 июля 2016 г. Под ред. С. А. Абрамова и Л. А. Севастьянова. М.: ФИЦ ИУ РАН*, с. 78–80, 2016.

[13] А. А. Панфёров. Символьный алгоритм распознавания сателлитных неизвестных в линейных дифференциальных системах с выделенными. // *Ломоносовские чтения: Научная конференция, Москва, факультет ВМК МГУ имени М. В. Ломоносова, 17–26 апреля 2017 г. Тезисы докладов. М: МАКС Пресс*, с. 122–122, 2017.

[14] A. A. Panferov. Irreducible differential systems and satellite unknowns. // *Компьютерная алгебра : материалы Международной конференции. Москва, 30 октября – 3 ноября 2017 г. / отв. ред. С. А. Абрамов, Т. М. Садыков. – Москва : ФГБОУ ВО «РЭУ им. Г. В. Плеханова»*, с. 144–150, 2017.

[15] A. A. Panferov. Linearly satellite unknowns in linear differential systems // In: Schneider C., Zima E. (eds) Advances in Computer Algebra. Springer Proceedings in Mathematics & Statistics. 2018, vol. 226, p. 215–227.

[16] А. А. Панфёров. Линейно сателлитные неизвестные в задаче частичной устойчивости линейных автономных дифференциальных систем. // *Ломоносовские чтения: Научная конференция, Москва, факультет ВМК МГУ имени М. В. Ломоносова. Тезисы докладов. М: МАКС Пресс*, с. 94–95, 2018.

[17] S. A. Abramov, M. A. Barkatou. Computable infinite power series in the role of

coefficients of linear differential systems. In *Proc. of CASC 2014, LNCS 8660, Springer, Cham*, p. 1–12, 2014.

[18] M. van der Put, M. F. Singer. Galois theory of linear differential equations. Springer, Berlin, 2003.

[19] M. Beckermann, H. Cheng, G. Labahn. Fraction-free row reduction of matrices of ore polynomials // J. of Symbolic Computation. 2006, vol. 41(5), p. 513–543.

[20] Moulay A. Barkatou, Carole El Bachaa, George Labahnb, Eckhard Pflügel. On simultaneous row and column reduction of higher-order linear differential systems // J. of Symbolic Computation. 2013, vol. 49, p. 45–64.

[21] W. A. Harris, Y. Sibuya, L. Weinberg. A reduction algorithm for linear differential systems // Funkcialaj Ekvacioj. 1968, vol. 11, p. 59–67.

[22] S. A. Abramov, M. A. Barkatou. On the dimension of solution spaces of full rank linear differential systems. In *Proc. of CASC 2013, LNCS 8136, Springer, Heidelberg*, p. 1–9, 2013.

[23] A. Minchenko, A. Ovchinnikov, M. F. Singer. Reductive linear differential algebraic groups and the Galois groups of parameterized linear differential equations // International Mathematics Research Notices. 2015, vol. 215(7), p. 1733–1793.

[24] E. Hrushovski. Computing the Galois group of a linear differential equation // Banach Center Publications. 2002, vol. 58, p. 97–138.

[25] Jerald J. Kovacic. An algorithm for solving second order linear homogeneous differential equations // Journal of Symbolic Computation. 1986, vol. 2(1), p. 3–43.

[26] Elie Compoint, Michael F. Singer. Computing Galois groups of completely reducible differential equations // Journal of Symbolic Computation. 1999, vol. 28(4–5), p. 473–494.

[27] M. A. Barkatou, E. Pflüegel. On the equivalence problem of linear differential systems and its application for factoring completely reducible systems. In *Proc. of ISSAC'98, Rostock, Germany*, p. 268–275, 1998.

[28] T. Cluzeau. Factorization of differential systems in characteristic p. In *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation ISSAC'03, Philadelphia, Pennsylvania, USA*, p. 58–65, 2003.

[29] А. А. Болибрух. Обратные задачи монодромии в аналитической теории дифференциальных уравнений. М: МЦНМО, 2009, 200 с.

[30] E. Beke. Die irreducibilität der homogenen linearen differentialgleichungen // Mathematische Annalen. 1894, vol. 45, p. 185–195.

[31] F. Schwarz. A factorization algorithm for linear ordinary differential equations. In *Proceedings of the ACM-SIGSAM 1989 International Symposium on Symbolic and Algebraic Computation ISSAC'89, Portland, Oregon, USA*, p. 17–25, 1989.

[32] M. Bronstein. On solutions of linear differential equations in their coefficient field // Journal of Symbolic Computation. 1992, vol. 13, p. 413–439.

[33] D. Yu. Grigoriev. Complexity of factoring and calculating the gcd of linear ordinary differential operators // Journal of Symbolic Computation. 1990, vol. 10, p. 7–37.

[34] M. F. Singer. Testing reducibility of linear differential operators: A group theory perspective // Applicable Algebra in Engineering, Communication and Computing. 1996, vol. 7(2), p. 77–104.

[35] S. P. Tsarev. An algorithm for complete enumeration of all factorizations of a linear ordinary differential operator. In *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation ISSAC'96, Zurich, Switzerland*, p. 226–231, 1996.

[36] M. van Hoeij. Factorization of differential operators with rational functions coefficients // Journal of Symbolic Computation. 1997, vol. 24, p. 537–561.

[37] D. Yu. Grigoriev. Complexity of irreducibility testing for a system of linear ordinary differential equations. In *Proceedings of the 1990 International Symposium on Symbolic and Algebraic Computation ISSAC'90, Tokyo, Japan*, p. 225–230, 1990.

[38] А. М. Ляпунов. Исследование одного из особенных случаев задачи об устойчивости движения. с. 272–331, 1956.

[39] В. В. Румянцев. Об устойчивости движения по отношению к части переменных // Вестн. МГУ. Сер. Мат., Механ., Физ., Астрон., Хим. 1957, № 4, с. 9–16.