James Harold Davenport

ON THE INTEGRATION OF ALGEBRAIC FUNCTIONS

Springer-Verlag Berlin Heidelberg New York 1981

Дж. Дэвенпорт

ИНТЕГРИРОВАНИЕ АЛГЕБРАИЧЕСКИХ ФУНКЦИЙ

Перевод с английского
Г. Е. МИНЦА

под редакцией
А. О. СЛИСЕНКО

Москва «Мир» 1985

ББК 22.162 + 22.19
Д 94
УДК 517.312

Дэвенпорт Дж.

Д 94 Интегрирование алгебраических функций: Пер. с англ. — М.: Мир, 1985. — 192 с.

Книга английского математика содержит замкнутое и подробное описание новых алгоритмов аналитического интегрирования. В частности, в ней, представлены известные результаты Риша. Излагаются теоретические основы таких алгоритмов, включающие большой объем сведений из алгебраической геометрии, приведены новые результаты.

Для программистов и математиков, для студентов и аспирантов математических специальностей.

$Д\frac{1702070000-275}{041(01)-85}$ 29—85, ч. 1

ББК 22.162 + 22.19
517.2 + 518

Редакция литературы по математическим наукам

ПРЕДИСЛОВИЕ РЕДАКТОРА ПЕРЕВОДА

Предлагаемая читателю книга является первой в мировой литературе монографией, в которой описаны общие алгоритмы символьного интегрирования или, другими словами, нахождения первообразных в «явном виде». Интерес к вопросам символьного интегрирования или, общее, символьного решения дифференциальных уравнений значительно возрос за последние годы. Это связано прежде всего с тем, что компьютерные системы символьных вычислений оказались мощным средством повышения производительности труда физиков, математиков, инженеров и других специалистов. Сказанное касается как теоретических исследований, так и в особенности прикладных работ, включая промышленные. С помощью таких систем получены как яркие теоретические результаты, так и решение огромного числа прикладных задач, которые вряд ли удалось бы решить достаточно быстро без применения компьютерных систем символьных вычислений.

Компьютерные интеграторы, т. е. комплексы программ для нахождения интегралов в символьном виде, являются, как правило, частью более широких систем символьных вычислений. На Западе из таких систем очень распространены системы MACSYMA и различные версии системы REDUCE. В книге Дэвенпорта описаны эксперименты, выполненные им на базе очень популярной системы REDUCE-2. Она достаточно хорошо известна и в СССР. Однако не нужно думать, что знакомство с системой REDUCE-2 сколько-нибудь существенно для понимания предлагаемой книги или разработки интеграторов. Книга Дэвенпорта посвящена прежде всего алгоритмам — как описанию собственно процедур, так и краткому изложению тех математических знаний, на которых эти алгоритмы основаны.

Как же ставится задача символьного интегрирования? Обычно ее формулируют следующим образом. Пусть задан класс функциональных выражений или, используя термин, принятый в математической логике, термов. Например, задан

класс выражений, которые можно построить с помощью суперпозиции из переменной x, рациональных чисел, числа π, операций сложения, умножения, взятия абсолютной величины, функции sin и какой-нибудь функции, неинтегрируемой в элементарных функциях. Нужно построить алгоритм, который бы по любому терму из рассматриваемого класса давал ответ, имеет ли этот терм первообразную в этом же классе и если да, то находил бы для первообразной соответствующее выражение. Из известной теоремы Д. Ричардсона следует, что для описанного класса термов такой алгоритм невозможен. Таким образом, задача символьного интегрирования в общей постановке неразрешима. Это относится и к задаче нахождения определенных интегралов в «явном виде», и к различным модификациям описанной выше задачи.

Однако наибольший интерес для приложений — а приложениями в данном случае являются компьютерные интеграторы, — представляют частные классы функций. Хорошо известно, как искать неопределенный интеграл от рациональной функции — его всегда можно построить. Если расширить поле рациональных функций конечным числом алгебраически независимых экспонент, то, хотя мы получим функции, вообще говоря, не интегрируемые в этом классе, задача о существовании первообразной и ее построении, когда она существует, останется алгоритмически разрешимой. Более того, имеется достаточно эффективный алгоритм (алгоритм Риша) для ее решения. Он кратко описан в гл. 4.

Основным результатом Дэвенпорта является разрешающий алгоритм для другого в некотором смысле крайнего случая — случая конечного алгебраического расширения поля рациональных функций. Им же описан ряд полезных соображений о дальнейшем расширении применимости соответствующей методики. Алгоритм Дэвенпорта, конечно, не столь эффективен как алгоритм Риша уже на уровне теоретических оценок вычислительной сложности. Оценки его эффективности упираются в весьма трудные вопросы алгебраической геометрии, которая вообще играет в вопросах символьного интегрирования ключевую роль.

Разумеется, практические интеграторы наряду с общими процедурами типа алгоритмов Риша и Дэвенпорта содержат в том или ином числе различные эвристические приемы интегрирования — например те, которые в изобилии можно найти в учебниках по классическому математическому анализу или соответствующих справочниках. Роль последних может оказаться существенной для ускорения решения простых или очень специальных задач. Однако более принципиальное значение для построения перспективных интегралов играют общие алгоритмы. Имеющаяся в настоящее время алгоритмика

позволяет строить довольно хорошие практические интеграторы.

В последние годы весьма интенсивно ведутся исследования в области вычислительных аспектов алгебраической геометрии. Это касается как базисных алгоритмов типа разложения многочленов на неприводимые, так и задач типа вычисления оценки кручения дивизоров на алгебраической кривой. Алгоритмы разложения многочленов на неприводимые являются составной частью всех известных общих алгоритмов интегрирования. Недавно в разработке эффективных алгоритмов для этой задачи был достигнут, по крайней мере в теоретическом плане, существенный прогресс [1]).

Это полезно иметь в виду, поскольку Дэвенпорт ссылается на более ранние работы, а алгоритмы разложения многочленов на неприводимые во многом определяют эффективность процедур, в которые они входят.

Интересную информацию об алгоритмах символьных вычислений, относящихся к интегрированию, содержит сборник Computer Algebra. Symbolic and Algebraic Computation. — 2nd edition. Eds. B. Buchberger, G. E. Collins, R. Loos, Springer Verlag, 1983. [2]) Систематическим источником информации об алгоритмах и системах символьных вычислений является ACM SIGSAM Bulletin, а также соответствующие тома серии Lecture Notes in Computer Science, издаваемой издательством Springer. Информация о советских работах в этой области по большей части содержится в трудах конференций по системам символьных вычислений.

При чтении книги Дэвенпорта нужно иметь в виду, что она написана «на скорую руку», по горячим следам разработки и реализации разработанного автором алгоритма интегрирования. Такой стиль изложения имеет свои плюсы и минусы. С одной стороны, можно весьма быстро получить довольно цельное представление об алгоритмах и относящихся к ним вопросах теории. С другой стороны, такое изложение требует значительных усилий при восстановлении деталей, особенно касающихся обоснования правильности алгоритмов. Внимательный читатель найдет широкое поле для работы по усовершенствованию многих из описанных

---

[1]) См. Lenstra A. K., Lenstra H. W., Lovasz L. Factoring polynomials with rational coefficients. — Preprint Math. Centrum Amsterdam IW 195/82, 1982; Chistov A. L., Grigoryev D. Yu. Polynomial-time factoring of the multivariable polynomials over a global field. — LOMI, Preprint E—5—82, Leningrad 1982.

[2]) В издательстве «Мир» готовится русский перевод.

алгоритмов. Хочется надеяться, что эта книга послужит хорошим стимулом для разработки и построения систем символических вычислений и вообще для исследований в области вычислительных аспектов алгебраической геометрии.

При переводе книги в качестве основного источника терминологии, касающейся алгебры и теории алгебраических функций, использовалась книга: Ван-дер-Варден. Алгебра. Пер. с англ. — М.: Наука, 1976.

А. О. Слисенко

ОТ АВТОРА

Я хотел бы выразить свою благодарность всем тем, кто сделал возможной эту работу. Мне не удалось бы справиться с программированием, не пользуясь советами опытных специалистов д-ров Дж. П. Ффитча и А. Ч. Нормана по вопросам их Лисп-системы, а также советами проф. А. Херна, д-ра А. Нормана и м-ра Р. Холла относительно системы REDUCE-2. Я благодарен также миссис П. Мур за любезное разрешение ввести некоторые из принадлежавших ей изменений в систему REDUCE-2.

Я благодарен директору и сотрудникам вычислительного центра Кембриджского университета, где были выполнены почти все описанные здесь исследования, за обеспечение моих казавшихся бесконечными вычислений и за советы. Особенно ценной была помощь М. Гая и Ч. Томсона. Реализация системы интегрирования в Исследовательском центре IBM имени Томаса Дж. Уотсона опиралась на советы Дж. Гарри, а реализация MTS была осуществлена с помощью м-ра У. Доджа и миссис Дж. Кавинесс из Политехнического института Ренсселара и м-ра М. Александера из Мичиганского университета. Многочисленные улучшения в программу разложения на множители были введены д-ром Д. Дамом (Барроуз корпорейшн), который обнаружил и несколько ошибок. Многие затруднения с интерфейсом Лисп-системы были обнаружены и устранены Д. Моррисоном (Университет штата Юта).

Я весьма благодарен многим математикам, советами которых я пользовался, особенно профессору сэру Питеру Суиннертон-Дайеру, д-ру Б. Берчу, проф. Д. Льюису, проф. С. Маклейну, д-ру Р. Ришу, проф. А. Шинцелю и проф. М. Зингеру. Своим первоначальным знакомством с вычислительными машинами и алгебраической геометрией я обязан д-ру Н. Стефенсу. Обсуждение специальных значений параметров в гл. 6 основано на беседах с д-ром П. Нойманом и участниками семинара Оксфордского университета по теории чисел.

Я хотел бы также выразить признательность за помощь, которую оказали мне многочисленные беседы с д-ром Дж. Ффитчем, проф. Б. Кавинессом, участниками семинара в Политехническом институте Ренсселара, д-ром Р. Джексом

и д-ром Д. Яном из Исследовательского центра IBM имени Томаса Дж. Уотсона, а также с членами группы MACSYMA в Массачусетском Технологическом институте (особенно с Б. Трейгером).

Я очень благодарен проф. П. Суиннертон-Дайеру и проф. Херну, которые прочли предварительный вариант этой книги, за их многочисленные комментарии и предложения. Профессор М. Зингер также прочел один из вариантов рукописи и сделал много ценных замечаний. Первые варианты этой книги были подготовлены Вычислительным центром Кембриджского университета, и я благодарен д-ру А. Герберту и М. Джонсону за их советы по поводу обработки текстов. Нынешний вариант был подготовлен в Исследовательском центре IBM имени Томаса Дж. Уотсона с помощью Йорктаунского языка макетирования, и я благодарен консультантам по обработке текстов мисс К. Кини и м-ру Ч. Томсону за их советы.

В заключение я хотел бы выразить благодарность моей матери за ее помощь (особенно при составлении списка литературы) и д-ру А. Норману, моему руководителю, за неустанные советы и ободрение.

Дж. Дэвенпорт

Глава 1

ВВЕДЕНИЕ

В этой работе рассматривается следующий вопрос: *когда алгебраическая функция интегрируема?* Мы можем придать этому вопросу другой вид, проясняющий наше понимание интегрируемости: когда по данной алгебраической функции мы можем найти выражение, составленное из алгебраических функций, логарифмов и экспонент, производная которого равна данной функции, и каково это выражение?

Этот вопрос можно рассматривать чисто математически как проблему алгоритмической разрешимости, но наш интерес носит более практический характер и его источник — требования компьютерной алгебры. Итак, наша цель — *написать программу, которая по данной алгебраической функции* либо выдаст выражение для ее интеграла, составленное из алгебраических функций, логарифмов и экспонент, либо докажет, что такого выражения нет.

КОМПЬЮТЕРНОЕ ИНТЕГРИРОВАНИЕ ВООБЩЕ

В этом разделе[1]) я кратко рассмотрю всю проблему нахождения неопределенного интеграла с помощью вычислительной машины и разъясню соотношение между поставленной выше задачей и компьютерной алгеброй вообще.

Простейший класс функций, которые мы могли бы пожелать проинтегрировать,— это многочлены. Здесь решение чрезвычайно просто: все многочлены интегрируемы и алгоритм состоит в простой замене aX^n на $aX^{n+1}/(n+1)$.

Следующий класс — рациональные функции, и снова любая рациональная функция интегрируема. Однако решение уже не столь непосредственно, как в предыдущем абзаце. Прежде всего интеграл от рациональной функции уже не обязан быть рациональной функцией, а может содержать логарифмы. Во-вторых, для того чтобы интегрировать рациональные функции, нам, возможно, придется расширить поле констант (простой пример этой ситуации приведен в работе

---

[1]) Этот раздел основан на недавнем обзоре компьютерного интегрирования (Norman & Davenport, 1979). Я благодарен д-ру Норману за многочисленные полезные обсуждения.

(Risch, 1969, предложение 1.1)). Поэтому, несмотря на то, что алгоритмы интегрирования рациональных функций существуют со времен Лиувилля и Эрмита, здесь еще есть место для усовершенствований, и последние успехи в этой области были достигнуты в работах (Trager, 1976) и (Yun, 1977). Эта проблематика стимулировала также интенсивные исследования по оперированию с алгебраическими числами (Trager, 1976) и (Zippel, 1977).

Очевидным расширением класса рациональных функций является класс алгебраических функций — см. ниже разд. «Краткий обзор предшествующих работ». Это первый случай, когда появляются неинтегрируемые функции, что в свою очередь служит главной причиной значительно большей трудности алгебраического случая по сравнению с рациональным.

Следующее расширение класса рациональных функций приводит к чисто трансцендентным функциям, т. е. к функциям, лежащим в чисто трансцендентной башне полей, основанной на $K(x)$ (где K — поле констант, x — переменная интегрирования). Здесь каждое расширение порождается чисто трансцендентно [1]) над предыдущим полем с помощью логарифма или экспоненты некоторого элемента этого поля. Этот случай проблемы интегрирования был полностью рассмотрен Ришем (Risch, 1969), алгоритмы которого в основном реализованы в программе SIN (Moses, 1971). Другой подход был использован в работах (Risch & Norman, 1977) и (Rothstein, 1977).

Результаты, описанные в этой монографии, которые дают полный алгоритм для алгебраического случая, показывают, что теперь имеются алгоритмы интегрирования как для чисто алгебраических, так и для чисто трансцендентных функций. Мы обсудим в гл. 9 возможность интегрирования функций, не принадлежащих ни одному из этих классов (т. е. являющихся трансцендентными, но не чисто трансцендентными).

ПЛАН КНИГИ (1)

Создание машинной программы, которая ищет интегралы от алгебраических функций или доказывает их неинтегрируемость, затрагивает как чистую математику, так и информатику. Мы можем построить частичный алгоритм [2]) для задачи интегрирования алгебраических функций без использо-

[1]) Например, нельзя строить $\exp(\log(1+x)/2)$, так как это выражение равно $(1+x)^{1/2}$ и потому является алгебраическим, а не трансцендентным.

[2]) То есть алгоритм, который заканчивает работу и выдает корректный интеграл, если подынтегральная функция интегрируема в элементарных функциях, но может работать вечно в противном случае (вместо того, чтобы сказать, что функция неинтегрируема).

вания слишком богатого набора сложных понятий (гл. 2—4), однако перевод в полный алгоритм требует существенно больше математических средств (гл. 5—8).

Прежде чем можно будет конкретнее описать материал, охватываемый этой монографией, мы должны лучше понять ее общую структуру. По существу, это порочный круг, но его можно разорвать, если обращаться с рациональной функцией, как если бы она была алгебраической; это и сделано в следующем разделе.

ПРИМЕР

Методы, которые мы используем для интегрирования алгебраических функций, основаны на алгебраической геометрии и теории чисел, так что трудно описать пример до того, как будет изложена лежащая в основе теория. Поэтому мы будем рассматривать рациональную функцию и интегрировать ее методами, которые потом будут использованы для алгебраических функций. Преимущество здесь в том, что многие вспомогательные алгоритмы становятся тривиальными.

Рассмотрим задачу интегрирования выражения $dx/(x-1)(x-2)^2$. Если $y = x-1$, то мы можем записать это выражение в виде $y^{-1}(1+2y+3y^2+\ldots)\,dy$, и почленное интегрирование показывает, что ответ содержит член $\log y$. Полагая $z = x-2$, мы получим подынтегральное выражение $z^{-2}(1-z+z^2+\ldots)\,dz$, и почленное интегрирование дает члены $\log z$ и $-1/z$.

Следовательно, интеграл содержит $\log(x-1) - \log(x-2) - 1/(x-2)$ и в действительности равен этому выражению.

Мы обошли молчанием одно важное обстоятельство: как быть со значением $x =$ бесконечность. Не может ли нам понадобиться член, происходящий от разложения в ряд в окрестности этой точки? Полагая $z = 1/x$, мы хотели бы расписать подынтегральное выражение в виде ряда по z:

$$\frac{dx}{(x-1)(x-2)^2} = \frac{z^3\,dx}{(1-z)(1-2z)^2} = \frac{z^3(-dz/z^2)}{(1-z)(1-2z)^2} = \frac{-z\,dz}{(1-z)(1-2z)^2},$$

так что не будет логарифмического члена, происходящего от значения $x =$ бесконечности.

ПЛАН КНИГИ (2)

В предыдущем примере мы разложили подынтегральную функцию в степенной ряд, получили логарифмические члены из вычетов (т. е. коэффициентов при z^{-1} в разложении в сте-

пенной ряд по z) и получили интеграл из логарифмической части и оставшейся части разложений в степенной ряд. Эти три шага соответствуют трем главам (2, 3, 4), дающим частичный алгоритм интегрирования алгебраических функций.

В гл. 2 рассматриваются представление алгебраических функций в компьютерной алгебре, вопрос о *плейсах* в алгебраической геометрии[1]) и вопрос о разложениях Пюизо (именно их мы используем в качестве корректных аналогов разложений в степенные ряды). В гл. 3 изучается алгоритм Коутса, при помощи которого находится функция с заданными полюсами и нулями. Он оказывается нужным не только для нахождения аргументов логарифмов, но и для получения алгебраической части, а также для решения многих других задач в алгоритме интегрирования. В гл. 4 сформулирована и доказана теорема Риша, которая (в описанной выше ситуации) сводит задачу интегрирования алгебраических функций к *задаче о дивизорах с кручением*. В частности, если допустить, что все дивизоры — с кручением, мы получаем упомянутый выше частичный алгоритм.

В гл. 5—8 рассматривается задача о дивизорах с кручением, которую можно (приблизительно) переформулировать так: *найдется ли для данного списка плейсов, в которых мы требуем, чтобы аргумент логарифма имел определенные полюсы и нули, такая функция, которая имеет эти полюсы и нули нужных порядков с точностью до постоянного множителя?* Например, мы, возможно, не сумеем найти функцию с полюсами и нулями порядка 1, но сможем найти функцию с полюсами и нулями порядка 3. В этом случае кубический корень из этой функции будет в действительности иметь полюсы и нули порядка 1. Так как $\log(f(x)^{1/3}) = \log(f(x))/3$, то нам останется только подобрать коэффициент при логарифме.

В гл. 5 рассматривается общая теория этой задачи и дается ответ в некоторых частных случаях. В гл. 6 мы рассматриваем эту задачу при наличии независимого трансцендентного параметра. В этом случае имеется полное решение, позволяющее нам либо сказать, обладает ли дивизор кручением, не находя степени кручения, либо свести задачу о кручении к задаче, не содержащей параметра (обе эти последние задачи рассматриваются в дальнейших главах). Если интеграл неэлементарен, мы можем задать дальнейший вопрос: *при каких значениях параметра этот интеграл элементарен?* Этот вопрос рассматривается в гл. 6.

В гл. 7 обсуждается вторая возможность (т. е. отсутствие трансцендентного параметра, когда задача определена над

[1]) Плейс — это надлежащее обобщение понятия «значение x», которое мы использовали при разложении в окрестности $x = 1$ или $x = 2$ (или даже около бесконечности).

полем алгебраических чисел) в случае эллиптических кривых. В этом случае мы можем, применяя хорошо развитую математическую теорию эллиптических кривых, определить, обладает ли дивизор кручением и какова степень кручения. В гл. 8 обсуждается оставшийся случай (т. е. поля алгебраических чисел, когда кривые неэллиптические) и описывается алгоритм для нахождения границы для кручения дивизора. После этого можно применять алгоритм Коутса к каждому кратному дивизора вплоть до этой границы, и мы можем объявить, что исходная функция не интегрируема элементарно, если мы не нашли множителя, для которого алгоритм Коутса даст соответствующую функцию.

ТЕОРЕТИЧЕСКИЕ ОГРАНИЧЕНИЯ

Одна из серьезных трудностей при такой формулировке проблемы — многозначность «логарифмической» функции. Следуя Ришу (Risch, 1969), мы можем устранить эту трудность, подойдя к задаче чисто алгебраически, т. е. определяя $\log u$ как такую функцию f, что $f' = u'/u$, и понимая это определение в духе дифференциальной алгебры (см. (Kolchin, 1973), а также (Ritt, 1948)).

В постановке этой задачи имеется более фундаментальная трудность. Так как существует и неинтегрируемая алгебраическая функция $f(x)$ (см. пример 2 из приложения 2), мы знаем, что $cf(x)$ интегрируема тогда и только тогда, когда $c = 0$. Однако известно (Richardson, 1968), что невозможен алгоритм, узнающий, равно ли нулю выражение, порожденное из целых чисел, $\log 2$ и π с помощью операций сложения, вычитания, умножения, деления и взятия экспонент, логарифмов и абсолютных величин. Соединение этих результатов показывает, что в некотором смысле проблема интегрируемости [1]) формально неразрешима. Ясно, однако, что установленная выше неразрешимость не совсем настоящая, так как она зависит от свойств чисел, а не интегралов.

Мы обойдем это ограничение, потребовав, чтобы были явно сформулированы все алгебраические зависимости между константами из подынтегрального выражения. Эти константы должны быть рациональными числами, алгебраически независимыми трансцендентными числами или выражениями с заданной явно алгебраической зависимостью от таких чисел.

---

[1]) Заметим, что мы не можем непосредственно воспользоваться теоремой (Richardson, 1968) о неразрешимости проблемы интегрируемости, так как она основана на возможности брать логарифмы и абсолютные значения в поле функций, а мы допускаем эти операции только в области констант.

Практической иллюстрацией необходимости выявления всех алгебраических зависимостей может служить наша реализация результатов Манина (гл. 6), где нам нужно дифференцировать по трансцендентной константе. Этот прием может сработать, только если мы точно знаем, какие из остальных констант алгебраически зависят от рассматриваемой.

Как указано в гл. 6, наше требование, чтобы зависимости были явными, означает, что мы не можем гарантировать правильных результатов [1]), когда работаем с интегралами, содержащими явно как e, так и π, ибо мы не знаем, являются ли эти константы алгебраически независимыми.

КРАТКИЙ ОБЗОР ПРЕДШЕСТВУЮЩИХ РАБОТ

Первой существенной попыткой интегрирования алгебраических функций с помощью программы для вычислительной машины была система SAINT (Slagle, 1961). Это была эвристическая система для интегрирования функций, основанная на ряде преобразований и упрощений. Ей сопутствовал замечательный успех в решении простых задач, но все ее эвристики были нацелены лишь на отыскание интегралов — она вообще не обращалась к проблеме установления неинтегрируемости выражений.

Следующим важным этапом в интегрировании алгебраических функций (и в компьютерном интегрировании вообще) была система SIN (Moses, 1967, 1971). Она включала алгоритм Риша (Risch, 1969) для чисто трансцендентных функций, а также ряд специальных правил преобразования и упрощения. Это дало программу, которая весьма полезна на практике и может интегрировать большинство «обычных» алгебраических функций, хотя она не использует сколько-нибудь глубокой теории и ее нельзя обобщить так, чтобы охватить более трудные интегралы (например, примеры 3—5 из приложения 2).

Хотя принятая в этой программе трактовка алгебраических функций не была алгоритмической, она могла в определенных случаях сводить алгебраическую функцию к чисто трансцендентной, где можно применить алгоритм (например, она выполнила бы подстановку $y = \mathrm{ch}(x)$ в $\sqrt{x^2 - 1}$, чтобы сделать интегрируемую функцию трансцендентной). Однако эта форма рационализации возможна обычно только для подынтегральных выражений над кривыми рода 0 (см. гл. 2),

[1]) Заметим, однако, что если мы построим интеграл в предположении алгебраической независимости этих констант, то он будет правильным даже если они в действительности алгебраически зависимы. Не пройти может только доказательство неинтегрируемости.

а все эти функции интегрируемы, так что частично алгоритмическая природа этой программы не столь полезна, как может показаться, ибо главное практическое преимущество алгоритмических методов состоит в том, что с их помощью можно доказывать неинтегрируемость.

Еще один значительный шаг в этой области сделал Нг (Ng, 1974), опиравшийся на математические результаты Карлсона. Используя теорию R-функций, Нг смог полностью решить вопрос об эллиптических и гиперэллиптических интегралах (т. е. интегралах, содержащих ровно один квадратный корень).

В последнее время в этой области работал Трейгер (Trager, 1978). Его (неопубликованные) результаты обобщают более ранние результаты (Trager, 1976) о поле констант поля определения для рациональных интегралов. Однако в настоящее время нет реализации этих существенных результатов. Совсем недавно (Trager, 1979) он работал над специальным случаем *простых радикальных расширений*, который используется, когда единственная алгебраическая величина, зависящая от переменной интегрирования, — это корень n-й степени (для некоторого n) из некоторой рациональной величины, и построил весьма эффективные алгоритмы для нахождения алгебраических частей интегралов в таких расширениях, реализующие, по-видимому, некоторые идеи Чебышева (Чебышев, 1853).

УЧЕТ МАШИННОГО ВРЕМЕНИ

Мы часто будем приводить конкретные примеры интегралов и сообщать, сколько времени затратила программа, реализующая алгоритм, на нахождение интеграла или установление неинтегрируемости. Если не оговорено противное, все эти данные относятся к реализации на установленной в Кембриджском университете машине IBM 370/165 [1]) с модифицированной версией операционной системы OS/MVT, выпуск 21.6. Алгоритм интегрирования реализован как часть системы REDUCE-2 (Hearn, 1973) версией от 6 марта 1978 г. (с несколькими модификациями: см. приложение 1, где описаны наиболее существенные из них) и работает на кембриджской версии Лиспа (ffitch & Norman, 1977). Приводится время центрального процессора минус время сборки мусо-

[1]) Эта машина была снабжена более быстрой основной памятью, чем обычно. Суммарный эффект этого изменения трудно оценить, но для программ, которые мы прогоняем, время центрального процессора оказывается примерно на 10 % меньше, чем на немодифицированной IBM 370/165. Система ДПА (динамического преобразования адресов, т. е. виртуальная память) не была установлена.

ра[1]). Не следует придавать слишком большого значения мелким вариациям в распределении времени, так как в нем заложена внутренняя ошибка порядка 15 % из-за вариаций.

Эти данные о времени решения годятся лишь для сравнения с аналогичными интегралами, обработанными той же версией программы, так как они сильно зависят от реализации вспомогательных алгоритмов — как описанных в этой монографии, так и основных алгоритмов компьютерной алгебры, доставляемых системой REDUCE-2. Например, модификация программы нахождения наибольшего общего делителя, описанная в п. 3 приложения 1, сократила время нахождения некоторых интегралов по меньшей мере в 10 раз. Кроме того, время, нужное для обработки интеграла, может сильно меняться из-за факторов, не связанных со сложностью самого интеграла; это иллюстрируется примером 1 в приложении 2. Приводимые времена не дают вполне корректного представления о скорости программы на сравнительно простых примерах, так как они прогонялись на неокончательной версии программы, которая содержала дополнительные проверки и печати, но не содержала многих искусственных приемов, весьма важных практически, но не имеющих теоретического значения. Пример 6 из приложения 2 иллюстрирует выигрыш, который может быть достигнут использованием версии, которая была сделана для предоставления пользователям.

[1]) Оно составляет обычно около 30 % остального времени, если программа работает с достаточным объемом памяти (примерно 700K байт для большинства примеров). Это время включает также загрузку транслированных лисповских функций с диска, что может составить до 2 с для сложных случаев.

Глава 2

АЛГЕБРАИЧЕСКИЕ ВЫЧИСЛЕНИЯ

АЛГЕБРАИЧЕСКИЕ СООТНОШЕНИЯ

Алгебраические соотношения между переменными или выражениями обычны в компьютерной алгебре. Они часто встречаются не только в явном виде, например SQRT(X**2+1), но и в неявном, когда возникают хорошо известные трудности, такие как в случае sin(x)**2+cos(x)**2=1 (Stoutemeyer, 1977). Несмотря на нашу привычку к алгебраическим соотношениям, нелегко вычислять, учитывая их. В этой главе обсуждаются проблемы, возникающие при таких вычислениях, и здесь мы попадаем в область алгебраической геометрии, которая естественно выросла из попыток выполнять такие вычисления столь же быстро, как и при отсутствии дополнительных соотношений.

Простейший случай вычислений, когда имеются алгебраические соотношения, возникает в поле $\{K(X, Y) \mid F(X, Y)=0\}$, где F — рациональная функция (которую можно свести к полиному путем умножения на все встречающиеся знаменатели), действительно зависящая от X и Y, а K — поле характеристики 0[1]). В качестве простого примера я рассмотрю алгебраическое соотношение, вводимое функцией $F(X, Y)=Y^2-X^2+1$.

Одной из основных идей эффективной компьютерной алгебры является понятие *канонической формы*. Под этим понимается средство, которое обеспечивает эквивалентность структур данных, представляющих эквивалентные выражения, независимо от того, как они были вычислены. Преимущество такого подхода в том, что случай неравных выражений обычно выявляется очень быстро при любой операции сравнения. Так как наши вычисления вполне могут оказаться на границе возможностей даже самых больших вычислительных машин, мы не можем позволить себе игнорировать это обстоятельство.

Первая проблема, возникающая в нашем примере, состоит в том, что Y^2 и X^2-1 — одна и та же функция в силу соотношения между X и Y. Чтобы сохранить каноническое

[1]) Обычно (за исключением гл. 8) мы не будем интересоваться полями конечной характеристики, и термин «поле» следует понимать (если не оговорено иное) как «поле характеристики 0».

представление наших алгебраических выражений, мы должны сделать что-то вроде замены выражения Y^2 на X^2-1 везде, где это выражение встречается. Это следует делать на очень низком уровне во всех частях системы, иначе наша система алгебраических преобразований решит, что матрица

$$\begin{pmatrix} X-1 & Y \\ Y & X+1 \end{pmatrix}$$

имеет определитель $(X^2-1)-Y^2=0$, но ранг 2, если при вычислении ранга это соотношение не было применено на надлежащем шаге. Из-за этого очень трудно добавлять алгебраические соотношения к системе, не предусматривающей хорошего оперирования с алгебраическими выражениями и соотношениями, которые могут из них возникнуть. В частности, обычный метод введения квадратных корней в систему REDUCE-2 (Hearn, 1973) с помощью правил вида

FOR ALL X LET SQRT(X)**2=X

не приведет к успеху в аналогичных, но более сложных случаях, так как нужное преобразование не применяется в нужных местах.

Замена высоких степеней переменной Y (в нашем примере Y^2) абсолютно во всех вхождениях может в некоторых случаях создать новые проблемы. Рассмотрим попытку вычисления н. о. д. полиномов $X^2-X*Y+1$ и $X*Y$ с помощью стандартного алгоритма Евклида, если X считается старшей переменной, т. е. в кольце $K[Y][X]$. В очевидных обозначениях вычисление проходит следующим образом:

Пояснение	U	V
Начальное состояние	$X**2-Y*X+1$	$X*Y$
$U:=U*Y-V*X$	$-X*Y**2+Y$	$X*Y$
	$=-X**3+X+Y$	$X*Y$
$U:=U*Y=V*X**2$	$X*Y+Y**2$	$X*Y$
	$=X**2+X*Y-1$	$X*Y$
$U:=U*Y-V*X$	$X*Y**2-Y$	$X*Y$
	$=X**3-X-Y$	$X*Y$
Продолжающийся цикл	...	...

Можно придумать более сложные примеры, показывающие, что объявление Y старшей (по сравнению с X) переменной не всегда позволяет обойти эти трудности. В действительности $K[X, Y]$ с соотношением $F(X, Y)=0$ не является областью с однозначным разложением на множители, так как $X^2-1=Y^2=(X-1)*(X+1)$, и все множители Y, $X-1$, $X+1$ неприводимы. Поэтому мы не можем корректно поставить вопрос о нахождении н. о. д. в области $\{K[X, Y] \mid F(X, Y)=0\}$. Любое представление, основанное на н. о. д., будет при-

водить к неудаче, пока мы четко не уясним себе, что именно мы понимаем под н. о. д. и какие алгебраические зависимости мы должны учитывать.

Это еще более затрудняет разработку канонического представления выражений в $\{K[X, Y] \mid F(X, Y) = 0\}$. В моих программах алгебро-геометрических вычислений для системы REDUCE-2 [1]) я применил следующее представление, основанное на допущении, что F — полином от X и Y со старшим коэффициентом, равным единице по Y. Если это допущение не выполнено, т. е. старший коэффициент G полинома F не является единицей в $K[X]$, то мы можем ввести $Y' = G*Y$, а тогда Y' удовлетворяет уравнению $F'(X, Y') = 0$ со старшим коэффициентом, равным единице, и $\{K(X, Y) \mid F(X, Y) = 0\}$ изоморфно $\{K(X, Y') \mid F'(X, Y') = 0\}$. Далее, в большинстве систем, предназначенных для алгебраических вычислений (и уж наверняка в REDUCE-2), элемент поля K представляется частным двух элементов некоторой области целостности L (например, рациональное число — это частное двух целых чисел, рациональная функция — частное двух полиномов). Тогда мы требуем (из чисто практических соображений), чтобы все коэффициенты полинома $F(X, Y)$ лежали в L (если это не так, достаточно умножить исходный полином на произведение всех знаменателей, а затем обеспечить равенство единице старшего коэффициента указанным выше способом). Это приводит к тому, что $L[X, Y]^2 \subseteq L[X, Y]$. Методика расширения с помощью ненормированных полиномов (т. е. полиномов с произвольным старшим коэффициентом) применялась в работе (Cohen & Yun, 1979), чтобы ввести некоторые рациональные дроби в областях целостности, но нам она не потребуется, так как мы всегда предполагаем наличие поля, в котором можно работать.

Если эти предположения выполнены, то элемент области $\{K(X, Y) \mid F(X, Y) = 0\}$ можно представить в виде A/B, где B принадлежит $K[X]$, а A принадлежит $\{K[X, Y] \mid F(X, Y) = 0\}$, причем A и B взаимно просты как элементы $K[X, Y]$ (т. е. без учета соотношения между X и Y), а выражение B в некотором смысле нормализовано (например, старший коэффициент B положителен). Последнее требование нужно, чтобы предотвратить появление различных представлений одного и того же элемента области $\{K(X, Y) \mid F(X, Y) = 0\}$, когда мы умножаем A и B на элемент из K.

ЕДИНСТВЕННОСТЬ АЛГЕБРАИЧЕСКИХ ВЫРАЖЕНИЙ

Однако описанный выше способ оперирования с алгебраическими выражениями недостаточен, если мы не имеем га-

[1]) Дополнительные сведения имеются в приложении 1.

рантии, что они правильно определены (в том смысле, что $f(X, Y)$, рассматриваемый как полином от Y, не имеет корней в $K(X)$). Действительно, в противном случае не гарантировано каноническое представление, даже если используется описанная в предыдущем разделе система. Хотя $F(X, Y) = Y^2 - X^2 + 2X - 1$ есть вполне разумное алгебраическое соотношение, определяющее алгебраическую кривую, однако эта кривая не является неприводимой, а алгебраическое расширение $K(X)$, определяемое этим уравнением, в действительности совпадает с $K(X)$, так как уравнение раскладывается на линейные множители. Аналогичные соображения включены в алгоритм интегрирования (Risch, 1970), где мы должны выражать некоторые вычеты как элементы $\mathbb{Z}$-модуля, а потому приходится распознавать, что $\sqrt{-1}$ и $\sqrt{-4}$ линейно зависимы над множеством целых чисел [1]). Аналогичным образом корректной алгебро-геометрической моделью для интегрирования выражения

$$\sqrt{X^2 - 1} + \frac{1}{\sqrt{1 - 1/X^2}}$$

является не $\{K(X, Y, Z) \mid Y^2 - X^2 + 1 = 0,\ Z^2 - 1 + 1/X^2 = 0\}$ с подынтегральным выражением $Y + 1/Z$, а $\{K(X, Y) \mid Y^2 - X^2 + 1\}$ с подынтегральным выражением $Y + X/Y$.

Чтобы проверить, что алгебраическое выражение правильно построено, нам нужно потребовать неприводимость [2]) полинома $F(X, Y)$ (точнее, наличие только одного множителя, содержащего Y; любой множитель, не содержащий Y, не играет роли). Если K есть $\mathbb{Q}$ (или чисто трансцендентное расширение поля $\mathbb{Q}$), то это — задача о разложении на множители многочленов от нескольких переменных над целыми числами, разрешимость которой известна (Wang & Rotschild, 1975), (Yun, 1976). Если K не является чисто трансцендентным расширением, то можно использовать недавно разработанные алгоритмы (см. Trager, 1976) разложения над алге-

[1]) Это особенно ярко проявилось при рассмотрении интеграла Чебышева (приложение 2, пример 5), где, обнаружив, что SQRT(2) — независимая алгебраическая величина над $\mathbb{Q}$, мы должны были рассмотреть SQRT(4SQRT(2)+9). Эта величина уже зависима над $\mathbb{Q}$(SQRT(2)), так как она равна 2 SQRT(2) + 1.

[2]) Здесь мы не отметили одно обстоятельство. Нам нужна *абсолютная неприводимость* (т. е. неприводимость над алгебраическим замыканием поля констант). Теперь, если $F(X, Y)$ — такой многочлен, выберем значение y переменной Y, такое, что $F(X, y)$ свободен от квадратов (что возможно, если $F(X, Y)$ таков). Тогда, если $F(X, Y)$ абсолютно неприводим, то он неприводим над полем разложения многочлена $F(X, y)$, так как любое разложение многочлена $F(X, y)$ соответствующее некоторому разложению многочлена $F(X, Y)$, может быть «поднято» до этого разложения без расширения области констант (см. любой p-адический метод разложения, например (Wang, 1978) или (Yun, 1973, 1976)).

браическими полями. Если бы у нас встречались только радикалы, а не алгебраические выражения общего вида, то мы могли бы использовать алгоритмы из статьи (Zippel, 1977).

ПРЕДСТАВЛЕНИЕ АЛГЕБРАИЧЕСКИХ ВЫРАЖЕНИЙ

Если мы хотим работать с несколькими (для простоты — с двумя) алгебраическими выражениями, то появляется выбор: мы можем работать с ними обоими непосредственно в *представлении с многими переменными* или выразить их оба через (более сложное) алгебраическое выражение, а именно через примитивный [1]) элемент для этого поля (во втором случае мы будем говорить о *примитивном представлении*). Математическая теория почти всегда оформляется в терминах примитивных элементов, так как это упрощает обозначения и описания. Однако использование примитивных элементов имеет и свои недостатки: в этом случае пришлось бы представлять $\sqrt{2}+\sqrt{3}+\sqrt{5}+\sqrt{7}+\sqrt{11}$ полиномом степени 31 от примитивного элемента, удовлетворяющего уравнению степени 32. Это представление может оказаться чрезвычайно трудоемким, а, кроме того, его все равно придется преобразовывать ко второму из рассматриваемых видов, если мы хотим, чтобы оно было понятно пользователю. Кроме того, может оказаться, что задача, поставленная в терминах двух алгебраических выражений (например, SQRT(*X***2 — 1) и SQRT(*X***2 — 2)), разлагается на две подзадачи, каждая из которых затрагивает только одно из этих алгебраических выражений. В системе, где два выражения хранятся раздельно, очень легко воспользоваться преимуществами этого разложения, а система с использованием примитивного элемента могла бы столкнуться здесь с большими трудностями. По этим причинам я использую представление с многими переменными, а не представление через примитивный элемент, хотя такой выбор — в основном дело вкуса. Этот вопрос рассматривался также и в работе (Caviness & Rothstein, 1976), которые пришли к тому же заключению, но тоже без твердых теоретических доводов. Интересно было бы найти такие доводы. Стоит заметить, что комментарии к описываемому в гл. 7 алгоритму ЗНАМЕНАТЕЛЬ-АЛГЕБРАИЧЕСКОГО, который строит определяющее уравнение для алгебраического числа с целью определить его знаменатель, не дают весомых доводов ни в какую сторону, так как та же конструкция была бы необходима и в системе, основанной на

[1]) См. (van der Waerden, 1976), § 46 русского перевода. Там применяется установленный в § 45 факт, что поля характеристики 0 совершенны, а потому имеют только сепарабельные расширения.

примитивном представлении, хотя ее было бы чуть легче программировать.

Если выбрано представление с многими переменными, то в общем случае поле имеет примерно следующий вид: $\{K(X, Y, Z) \mid F(X, Y) = 0,\ G(X, Y, Z) = 0\}$, где F фактически [1]) содержит X и Y, а G действительно содержит Z и одну из переменных X, Y (но не обязательно обе); такие представления рассматриваются в работе (Shtokhammer, 1977). Примером является $\{K(X, Y, Z) \mid Y^2 - 1 - X^2 = 0,\ Z^2 - 2 - X^2 = 0\}$. Если мы выберем это представление, то нам нужен способ, позволяющий узнать, независимо ли алгебраическое выражение от предыдущих, т. е. механизм разложения полиномов над алгебраическими полями от нескольких переменных. Его можно получить, используя работу (Trager, 1976), где описан алгоритм ALG_FACTOR (см. также подстрочное примечание к алгоритму АЛГ_ФАКТОРИЗАЦИЯ_2 в приложении 3), сводящий проблему разложения над $\{K(a) \mid f(a) = 0\}$ к проблеме разложения над K. Этот алгоритм можно итерировать, чтобы свести дело к задаче разложения над $K(X)$, а она была решена выше.

Сделанные выше замечания могут создать впечатление, что система, хорошо оперирующая с алгебраическими объектами, уже достаточна для алгебро-геометрической системы. Ошибочность такого вывода показывает следующий пример поведения алгебраических выражений при преобразованиях обычно применяемого типа: $\sqrt{x^2 - 1} \to$ (при подстановке $x \to 1/x$) $\sqrt{1/x^2 - 1} = \sqrt{1 - x^2}/x = i\sqrt{x^2 - 1}/x \to$ (при подстановке $x \to 1/x$, которая делается, чтобы аннулировать действие предыдущего преобразования) $-\sqrt{x^2 - 1}$; тем самым мы доказали, что $1 = -1$. Решение этой проблемы состоит в том, чтобы иметь отдельный базис для алгебраических объектов в системе для каждого «значения» переменной x, чтобы в приведенном примере мы не стали бы выражать $-\sqrt{1 - x^2}$ в виде $i\sqrt{x^2 - 1}$, так как у нас нет оснований делать это. Таким образом, нужна какая-то форма контроля программиста над системой, оперирующей с алгебраическими объектами. Это требование, однако, не совсем разрушительно, так как современные алгоритмы определения зависимостей между алгебраическими выражениями (Trager, 1976) имеют по меньшей мере экспоненциальную (по числу рассматриваемых алгебраических выражений) сложность. Поэтому

[1]) Заметим, что мы не разрешаем, чтобы F вообще содержал Z. Мы требуем, чтобы каждое алгебраическое выражение было определено через предыдущие (а значит, через X) ровно одним уравнением. Чтобы ослабить это ограничение, нам пришлось бы рассмотреть всю проблему базисов Грёбнера (см. обсуждение в статье (Buchberger, 1979)).

тот факт, что нам не нужно рассматривать алгебраические выражения, связанные с одним значением x, когда мы работаем с другим значением, может привести к радикальному уменьшению сложности.

СООБРАЖЕНИЯ, СВЯЗАННЫЕ С РЕАЛИЗАЦИЕЙ

Здесь мы рассмотрим практические последствия соображений, высказанных в предыдущих разделах, и опишем, как описанные процессы реализованы в программе (см. также приложение 2, где в более технических терминах обсуждаются изменения, сделанные в программе REDUCE-2).

Самое важное ограничение состоит в том, что программа работает только с квадратными корнями, т. е. уравнение $F(X, Y)$ должно иметь вид $Y^2 - G(X)$. Так как мы приняли представление с многими переменными для алгебраических выражений, мы можем представить несколько квадратных корней одновременно, а также и корни из корней. Решение ввести эти ограничения было принято из чисто практических соображений: все проблемы интегрирования можно проиллюстрировать на примерах, где участвуют только квадратные корни. С другой стороны, программы, работающие только с квадратными корнями, гораздо короче[1]) и функционируют существенно быстрее. Так как разработка программы потребовала напряженного использования имевшихся вычислительных ресурсов, эти соображения оказались чрезвычайно важными.

Для каждого «базисного плейса» (этот термин разъясняется в следующем разделе, но пока его можно понимать как значение X с соответствующими значениями $Y, \ldots$) мы храним отдельный список алгебраических выражений, определяющих поле $\{K(X, Y, \ldots) \mid F(X, Y, \ldots) = 0, \ldots\}$, причем каждое такое выражение определяется уравнением, неприводимым над расширением, определяемым всеми предшественниками данного выражения. Всякий раз когда в рассмотрение входит новое алгебраическое выражение, оно проверяется на независимость от списка, соответствующего надлежащему базисному плейсу. Этот механизм позволяет избежать трудностей, описанных в конце предыдущего раздела, и предотвращает проверку алгебраических выражений на независимость в тех случаях, когда это не нужно.

В действительности мы работаем не в переменных $X, Y, \ldots$, а только в переменной X. Переменная Y (когда

[1]) Главная причина этого в том, что у квадратного корня ровно один сопряженный, который можно получить, изменив знак у данного выражения, в то время как алгебраическое выражение общего вида даже не обязательно определено над полем, где лежит исходное выражение.

задано соотношение $F(X, Y) = Y^2 - G(X) = 0$) представляется выражением SQRT(G(X)), которое входит в ядро системы REDUCE-2 (Hearn, 1973), т. е. с ним можно обращаться как с отдельной переменной при построении полиномов и рациональных функций. В программе используется (с помощью описания в системе REDUCE-2) правило упрощения

FOR ALL Z LET SQRT(Z)**2 = Z,

а необходимые вызовы стандартной процедуры упрощения вписаны в текст программы в нужных местах. (В действительности программа была оптимизирована — подробности см. в п. 4 приложения 1.) Этот процесс, к сожалению, чреват появлением ошибок в программе, и многие из них были вызваны лишними упрощениями (т. е. работой в $\{K(X, Y) \mid F(X, Y) = 0\}$, когда нужно было $K(X, Y)$) или, наоборот, их отсутствием.

То, что процедуру упрощения нужно вызывать лишь «иногда»,— одна из самых тяжелых проблем при использовании системы REDUCE-2 для алгебраических преобразований; те же проблемы возникают и у многих других систем компьютерной алгебры. Одна из целей системы SCRATCHPAD/370, разрабатываемой в настоящее время в отделении IBM в Йорктаун-Хайтсе (Jenks, 1979) и (Davenport & Jenks, 1980),— устранение этой коллизии путем расширения запаса возможных областей вычисления.

АЛГЕБРАИЧЕСКАЯ ГЕОМЕТРИЯ

Мы можем рассматривать соотношение $F(X, Y) = 0$ как равенство, определяющее алгебраическую кривую в пространстве $K(X, Y)$, и это оказывается весьма полезной точкой зрения во всей оставшейся части нашей книги. Обзор алгебраических кривых и алгебраической геометрии на нужном нам уровне имеется в книге (Fulton, 1969) или (Seidenberg, 1968). Чисто алгебраический подход можно найти у (Chevalley, 1951), а смешанный подход содержится в книге (Eichler, 1966). Алгебраическую кривую можно рассматривать как риманову поверхность с ветвями, но в действительности алгебраические геометры обычно говорят о «плейсах». Плейс — это то же самое, что ветвь, хотя стандартное определение формулируется на языке локальных колец нормирований. Говоря о плейсе, *лежащем над* значением A переменной X (где A может быть и бесконечностью) или *центрированном* этим значением, мы имеем в виду, что функция $X - A$ (или $1/X$ в случае бесконечности) принимает значение 0 в этом плейсе.

Рассматриваемая алгебраическая кривая может иметь кратные точки, но для каждой алгебраической кривой (и для любого конечно порожденного алгебраического расширения

поля $K(X)$) имеется «неособая модель» этой кривой (см. (Fulton, 1969), гл. 7, теорема 3 и ее следствие), которая получается из исходной кривой путем «раздувания» всех ее особенностей. Эта образная терминология означает, что точка плоскости, которая была кратной точкой, заменена линией, а кратная точка на кривой заменена несколькими более простыми точками, и если повторить этот процесс достаточное число раз, то у нас останутся только простые точки. К сожалению, эта неособая модель уже не является кривой на плоскости X, Y, она трудно обозрима и с ней нелегко оперировать непосредственно. Поэтому теорема о существовании неособых моделей не слишком полезна в вычислительном аспекте.

С геометрической точки зрения возможны два типа поведения кривой в окрестности кратной точки в зависимости от наличия кратных касательных. Кривая $Y^2 = X^2 - X^3$ имеет в начале координат точку кратности 2 с двумя отдельными касательными, а кривая $Y^2 = X^3$ — точку кратности 2 с единственной касательной. Разумеется, в данной точке для данной кривой могут реализоваться оба эти типа поведения (например, может существовать точка кратности 7 с тремя касательными), но эта ситуация не сложнее по существу. Точка кратности n, в которой имеется n различных касательных к данной кривой, называется *обычной* кратной точкой. Намного полезнее преобразования, описанного в предыдущем абзаце, оказывается преобразование плоской кривой в бирационально эквивалентную плоскую кривую, все кратные точки которой обычные (см. (Fulton, 1969), гл. 7). На практике, как правило, достаточно производить эту элиминацию особых точек «локально», т. е. устранять необычную кратную точку, если она мешает, а не пытаться найти все кратные точки и произвести преобразование для всех них. Эти приемы плюс распознавание обычных кратных точек позволяют нам производить вычисления на алгебраических кривых, не пугаясь потенциальных особенностей, хотя программа должна всегда предусматривать возможность появления особенности и производить необходимые преобразования.

Еще один вид преобразований, который часто может потребоваться, это *бирациональные преобразования*, т. е. такие отображения кривой в пространстве $(x_1, \ldots, x_n)$ в кривую в пространстве $(y_1, \ldots, y_m)$, что все x_i — рациональные функции переменных y_j и наоборот. Если имеется бирациональное преобразование A в B, то мы говорим, что они *бирационально эквивалентны*.

Лемма[1]) **1.** *Две кривые бирационально эквивалентны тогда и только тогда, когда их поля функций изоморфны.*

[1]) (Fulton, 1969), предложение 12, с. 155.

РАЗЛОЖЕНИЯ ПЮИЗО

Важный инструмент алгебраической геометрии — теория *разложений Пюизо*. Разложение Пюизо в данном плейсе — аналог разложения в ряд Лорана в окрестности данной точки, хорошо знакомого из обычной теории функций. Точно так же, как в обычной теории функций, мы не можем выразить $\sqrt{X}$ в виде ряда Лорана (с целыми показателями) по степеням X, мы не можем в поле $\{K(X, Y) \mid Y^2 - X = 0\}$ выразить Y в виде разложения Пюизо по X в точке $X = 0$, хотя мы можем выразить X через Y в той же точке. Однако для любого плейса, соответствующего элементу A (или с центром в A), всегда имеется некоторая дробная степень выражения $(X - A)$ (или $1/X$ в случае бесконечности), такая что все функции из $\{K(X, Y) \mid F(X, Y) = 0\}$ можно выразить в виде разложения Пюизо по этой степени (см. (Chevalley, 1951), гл. 1, теорема 2). Если все функции могут быть выражены в виде разложений Пюизо по некоторой фиксированной функции относительно данного плейса, а никакая целая степень этой функции недостаточна, то эта функция называется *униформизующей* (*переменной*) или *локальным параметром*. Нам придется часто вспоминать о том, что $(X - A)$ или $1/X$ не обязательно является локальным параметром. Плейс, в котором это выражение не является локальным параметром, называется *разветвленным*, а степень, в которую нужно возвести локальный параметр, чтобы получить $(X - A)$ или $1/X$, называется *индексом ветвления* [1]) (так что для неразветвленного плейса индекс ветвления равен 1).

Мы определяем *порядок* функции в плейсе как частное от деления индекса первого ненулевого члена ее разложения Пюизо в этом плейсе на индекс ветвления этого локального параметра. Очевидно, что эта величина не зависит от выбранного локального параметра.

СТРУКТУРЫ ДАННЫХ ДЛЯ ПЛЕЙСОВ

Нам требуется какая-то структура данных, чтобы представить в нашей программе математическое понятие плейса. Избранный мной общий формат этой структуры данных — подстановочный список языка Лисп, т. е. список, состоящий из пар (старое значение, новое значение).

Первая пара списка определяет значение переменной интегрирования (во всех примерах — это переменная X), над

[1]) Термины «разветвленный» и «индекс ветвления» используются также и в алгебраической геометрии в несколько ином смысле, и в этом смысле они нам тоже понадобятся в гл. 8 (где и будут определены). Из контекста всегда будет ясно, какое понимание мы в данный момент имеем в виду — геометрическое или теоретико-числовое.

которым лежит рассматриваемый плейс. Если плейс лежит над $X = 0$, то первая пара есть (X.X); если плейс лежит над бесконечно удаленной точкой, то первая пара (X QUOTIENT 1 X); если он лежит над точкой $X = a$, то первая пара (X PLUS X — a). После того как такая подстановка выполнена, переменная X принимает значение 0 в точке, над которой лежит плейс.

Однако X не обязан быть локальным параметром. Второй элемент списка — это подстановочная пара, подобранная так, чтобы справиться с этой проблемой (если она имеется). Он имеет вид (X EXPT X n), где $X^{1/n}$ — локальный параметр рассматриваемого плейса. Если уже сам X был локальным параметром, то эта компонента отсутствует. Описанные две компоненты списка (вторая может отсутствовать) образуют *базисный плейс*, т. е. такое выражение локального параметра в данной точке, что для его преобразования в описание плейса нужен лишь некоторый способ выделения того плейса над этой точкой, который мы имеем в виду.

Оставшаяся часть списка предназначена для того, чтобы отличать друг от друга различные плейсы, лежащие над одной и той же точкой. Это может быть гораздо труднее, чем кажется, так как плейсы в некотором смысле неразличимы: все они — корни алгебраического уравнения, определяющего рассматриваемую кривую. Можно принять совсем произвольное решение, например обозначить один из корней [1]) уравнения $Y^2 = X^2 - 1$ через SQRT(X**2 — 1), а другой — через —SQRT(X**2 — 1). Тогда рассматриваемая часть списка содержит подстановочные пары вида (Y.Y) или (Y MINUS Y), где Y — выражение вида SQRT (полином от X); Y представлен в виде выражения, которое получилось бы, если бы преобразование, закодированное в первой и второй частях списка, уже было проведено. Например, два плейса кривой $Y^2 = X^4 + X^3 + 1$, которые лежат над бесконечной точкой, — это ((X QUOTIENT 1 X)(expr.expr)) и (X QUOTIENT 1 X)(expr MINUS expr), где expr есть SQRT(1 + X + X**4).

Казалось бы, что при такой структуре данных можно непосредственно вычислять значения функций в плейсах: достаточно подставить $X = 0$ в результат подстановки выражения для плейса в выражение для функции. К сожалению, это часто приводит к ответу 0/0.

На это естественно ответить использованием правила Лопиталя для раскрытия неопределенностей, т. е. равенства $\lim F(X)/G(X) = \lim F'(X)/G'(X)$, если $G(0) = 0$. Хотя это

[1]) В случае когда $F(X, Y)$ не имеет вида $Y^2 - G(X)$, а его степень по Y равна n, мы будем иметь n возможностей, соответствующих сопряженным величинам Y.

правило чрезвычайно полезно для математика, нет гарантии, что будет легче вычислить значение $F'(X)/G'(X)$, чем $F(X)/G(X)$. Это вполне может оказаться труднее — читатель, быть может, не сочтет за труд рассмотреть выражение

$$\frac{iX-\sqrt{2}\sqrt{A-\sqrt{A^2+X}}}{A+\sqrt{X^2+A^2}}$$

где F' и G' сложнее, чем F и G. В действительности ответ на возникшую проблему таков: вычислять такие значения, как коэффициенты при X^0 в разложении Пюизо. Харрингтон в своей работе (Harrington, 1979a) о вычислении пределов приводит многообразные применения правила Лопиталя, но в конце концов признает, что ряды Тейлора (эквивалентные разложениям Пюизо) могут оказаться необходимыми для вычисления некоторых пределов.

ВЫЧИСЛЕНИЕ РАЗЛОЖЕНИЙ ПЮИЗО

Вычисление разложений Пюизо — задача, весьма похожая на вычисление рядов Тейлора (или, точнее, Лорана). Прежде чем отмахнуться от этой задачи как совсем тривиальной, стоит рассмотреть вопрос: чему равен коэффициент при X^0 в разложении $(1+Y)/(1-Y)$, где $Y^2=X^2+1$?[1]). Вычисление «в лоб» дает нам ответ 2/0, а для получения правильного ответа нам пришлось бы найти коэффициенты при X^2 в разложении $1+Y$ и $1-Y$. Кроме того, вычислив степенной ряд, скажем вплоть до члена с X^4, мы можем обнаружить, что нам нужен член с X^5, и было бы расточительно перевычислять весь ряд снова для того, чтобы получить следующий член.

Наилучшая техника такого разворачивания рядов — это «неполный, но точный» подход Нормана (Norman, 1975). Норман резюмирует свой замысел, говоря, что при использовании его пакета «создается впечатление, что работаешь скорее с полными, чем с урезанными, степенными рядами. Вычисления выполняются только тогда, когда делается попытка вывести или использовать результаты, так что никогда не происходит бесполезной работы».

Норман сумел использовать системы правил и оценок SCRATCHPAD, чтобы создать чрезвычайно тонкую реализацию. У нас в системе REDUCE-2 нет этого механизма, но мы можем выполнять эквивалентные функции. Представление разложения Пюизо — это структура данных, содержащая две компоненты: список уже вычисленных членов и метод вычис-

[1]) В действительности ответ равен -2.

ления их значений, т. е. оператор (+, —, *, /, SQRT), и список аргументов, которые сами являются разложениями Пюизо компонент исходного выражения.

ДИВИЗОРЫ

Мы можем построить свободную абелеву группу, образующими которой являются все плейсы алгебраической кривой, т. е. множество всех конечных наборов плейсов с целыми кратностями. Элемент этой группы называется *дивизором*. Эту группу можно записывать мультипликативно или аддитивно, и в литературе используются оба способа. Обычно нам будет удобно записывать ее мультипликативно. Сумма всех кратностей называется *степенью* дивизора. Можно считать, что кратности в дивизоре показывают, является ли данный плейс корнем (кратности, указанной в дивизоре) или полюсом (кратности, указанной в дивизоре, но без учета знака минус) определенной функции. В этом случае дивизор D называется *дивизором* этой *функции* f, и мы пишем $D=(f)$. Любая функция, отличная от нулевой, имеет дивизор, и отображение мультипликативной группы ненулевых функций в группу дивизоров является в действительности групповым гомоморфизмом. Ядро этого гомоморфизма — в точности множество постоянных функций (т. е. ненулевых элементов K). Дивизор функции всегда имеет степень 0 ((Chevalley, 1951), гл. 1, теорема 5), но обратное верно не всегда. В частном случае $K(X)$ это верно, так как мы перемножаем $(X-A)^N$ для всех конечных плейсов, лежащих над A с кратностью N, а это дает правильную кратность над бесконечностью. Дивизор без отрицательных кратностей называется *эффективным*.

Для дивизора степени 0 имеются, вообще говоря, всего 3 возможности. В первом случае он является дивизором некоторой функции из $\{K(X, Y) \mid F(X, Y)=0\}$, и тогда мы говорим, что этот дивизор является *главным* или что он *линейно эквивалентен нулю*. Во втором случае некоторая его степень линейно эквивалентна нулю, и тогда мы говорим, что этот дивизор *рационально эквивалентен нулю* или является *дивизором с кручением*. В третьем случае никакая его степень не является дивизором функции, и тогда мы говорим, что этот дивизор (рационально) *не эквивалентен нулю*. Если дивизор рационально эквивалентен нулю, то его *порядком* называется степень, в которую его нужно возвести, чтобы он стал линейно эквивалентен нулю.

В случае поля $K(X)$, как замечено выше, можно тривиальным образом найти для любого дивизора степени 0 соответствующую ему функцию, и потому такой дивизор ли-

нейно эквивалентен 0. В общем случае это не так, и прогресс в проблеме нахождения функции, соответствующей дивизору который линейно эквивалентен 0, был достигнут в работе (Coates, 1970), содержание которой излагается в следующей главе.

ДИФФЕРЕНЦИАЛЫ

Дифференциал — еще одно полезное понятие из алгебраической геометрии. Его можно представлять себе как выражение вида $f(X)\,dX$, хотя строгое определение должно быть гораздо более абстрактным (см. (Fulton, 1969), p. 203—205, или (Shafarevich, 1972)). Мы отождествляем дифференциалы, равные в силу обычных правил элементарного математического анализа, полагая, например, $d(X^2) = 2X\,dX$ (однако снова см. точное описание в книгах (Fulton, 1969) или (Shafarevich, 1972). Мы можем очевидным образом превратить множество дифференциалов в модуль над полем $\{K(X, Y) \mid F(X, Y) = 0\}$, полагая, например, $c\,dX = d(cX)$ для элементов c поля K. Пространство всех дифференциалов — 1-мерное векторное пространство над $\{K(X, Y) \mid F(X, Y) = 0\}$ и dX — его базис.

Если t — униформизующий параметр плейса P, то любой дифференциал можно записать в виде $f\,dt$ для некоторого f из $\{K(X, Y) \mid F(X, Y) = 0\}$. Положим по определению, что *порядок* дифференциала в плейсе P есть порядок f в P (это — корректное определение, см. (Fulton, 1969), гл. 8 предложение 7). Следовательно, мы можем определить *дивизор дифференциала* точно так же, как мы раньше определили дивизор функции. Отметим существенное различие между этими двумя понятиями: дивизор функции $1/X$ — это полюс порядка 1 в нуле и нуль порядка 1 на бесконечности, в то время как дивизор дифференциала $1/X\,dX$ — это полюс порядка 1 в нуле и полюс порядка 1 на бесконечности (так как $t = 1/X$ является там униформизующим параметром и $1/X\,dX = t(dX/dt)\,dt = -1/t\,dt$.

Говорят, что данный дифференциал является дифференциалом *первого рода*, если он не имеет полюсов. Примером является $1/Y\,dX$, если $F(X, Y) = Y^2 - (X^3 + 1)$. Дифференциалы первого рода образуют векторное пространство над K, а размерность этого пространства, часто обозначаемая через g, называется *родом* или *дефектом* кривой. Род — весьма важный параметр кривой в алгебраической геометрии. В частности, кривую рода 0 можно преобразовать в прямую линию с помощью *рационализирующей подстановки*. Это обосновывает использование тригонометрических подстановок при интегрировании функций, определенных над кривыми рода 0 (простейший пример — кривая $F(X, Y) = Y^2 - (1 - X^2)$), так

как если X есть $\cos t$, то Y есть $\sin t$ и обе они — рациональные функции от $\operatorname{tg}(t/2)$, а это и есть настоящий рационализирующий параметр. *Кривая* рода 1 называется *эллиптической*. Мы подробно рассмотрим такие кривые в гл. 5 и 7.

Как и в классической теории функций, мы можем определить *вычет* дифференциала в данном плейсе. Имеются разнообразные абстрактные определения, но нам достаточно сказать, что вычет — это коэффициент при t^{-1} в разложении дифференциала по униформизующему параметру. Если вычет дифференциала отличен от нуля, то рассматриваемый плейс должен быть полюсом этого дифференциала, поэтому дифференциал может иметь лишь конечное число ненулевых вычетов. Говорят, что дифференциал является дифференциалом *второго рода*, если все его вычеты равны нулю. Очевидно, что всякий дифференциал первого рода является и дифференциалом второго рода. Дифференциалы второго рода образуют $2g$-мерное векторное пространство над пространством точных дифференциалов. Говорят, что дифференциал является дифференциалом третьего рода, если все его полюса имеют порядок 1. Дифференциал второго и третьего рода одновременно необходимо является и дифференциалом первого рода.

Представление дифференциала в точности такое же, как и представление функции, но правила его преобразования и вычисления значений иные. Поскольку дифференциал имеет вид $f\mathrm{d}t$, где f — некоторая функция и t — униформизующий параметр, то для пересчета дифференциала в новом плейсе мы должны преобразовать функцию f (как описано выше в разд. «Структура данных для плейсов»), а затем умножить на $\mathrm{d}t/\mathrm{d}t'$, где t' — униформизующий параметр в этом новом плейсе.

Глава 8

АЛГОРИТМ КОУТСА

ВВЕДЕНИЕ

В этой главе мы рассматриваем проблему нахождения функции с данным множеством полюсов. Что эта проблема нетривиальна в случае алгебраических функций (хотя она и тривиальна в случае рациональных функций), видно хотя бы из того, что такая функция не обязательно существует Например, на кривой, определенной выражением $\sqrt{X^3+1}$ не существует функции, имеющей нуль порядка 1 в плейсе лежащем над точкой $X=0$, и полюс порядка 1 на бесконечности и не имеющей никаких других нулей и полюсов, хотя имеется функция с дивизором, кратность которого в 3 раза больше (т. е. дивизор имеет порядок 3). На кривой, определенной уравнением $Y^2=X^3-3X^2+X+1$, нет функции, имеющей нуль в одном плейсе, лежащем над $X=0$, полюс — в другом (причем оба — одного и того же порядка) и не имеющей никаких других нулей и полюсов (см. приложение 2, пример 7).

Коутс опубликовал[1]) в 1970 г. алгоритм (Coates, 1970) нахождений функций на алгебраических кривых над полями алгебраических чисел. На точные формулировки результатов, которые он формулировал и доказывал, сильно повлияло то обстоятельство, что алгоритм был частью более широкого исследования диофантовых уравнений (Baker & Coates 1970). Ниже мы рассмотрим несколько иную формулировку алгоритма в несколько ином контексте. В частности, в свете замечаний, сделанных в предыдущей главе, мы можем считать, что плейсы, в которых мы будем производить вычисления, неразветвленные, хотя теория требует только того, что бы это выполнялось на бесконечности (но если даже и этого нет, то теорема 2 статьи (Coates, 1970) показывает, что все еще верен несколько модифицированный результат). Для

---

[1]) Коутс сформулировал свой алгоритм только для полей алгебраических чисел и стремился лишь показать, что он эффективен в математическом смысле слова. Распространение на произвольное основное поле по-видимому, новый (хотя и не очень глубокий) результат. Я думаю также, что в этой книге описана первая машинная реализация алгоритм Коутса.

упрощения вычислений мы часто будем использовать неразветвленное представление даже тогда, когда теория этого не требует.

ОПИСАНИЕ АЛГОРИТМА

Пусть D — дивизор (состоящий только из) полюсов (т. е. все кратности в нем отрицательные) и все его плейсы конечны (т. е. лежат над конечными точками). Тогда алгоритм дает по D базис векторного пространства (над K) всех функций f, таких, что $(f) \geqslant D$, т. е. пространства функций, полюса которых не хуже, чем D. Этот алгоритм использует два вспомогательных: РЕДУКЦИЯ_К_ЦЕЛОМУ_БАЗИСУ (INTEGRAL_BASIS_REDUCTION) и РЕДУКЦИЯ_К_НОРМАЛЬНОМУ_БАЗИСУ (NORMAL_BASIS_REDUCTION), которые в соответствии с требованиями программирования сверху вниз будут описаны после основного алгоритма. Эти алгоритмы описаны в упрощенной форме без учета усложнений, вызванных необходимостью рассматривать представления нашей кривой, где некоторые плейсы разветвленные. Мы не описываем также несколько приемов, использованных для повышения эффективности, так как, несмотря на их практическую важность, они слишком усложняют формулировки.

Алгоритм можно резюмировать следующим образом. Мы начинаем с любого списка функций, не имеющих конечных полюсов, постепенно добавляем все конечные полюса, указанные в дивизоре D (используя РЕДУКЦИЮ_К_ЦЕЛОМУ_БАЗИСУ), а затем находим те из результирующих функций, которые не имеют полюсов на бесконечности (используя РЕДУКЦИЮ_К_НОРМАЛЬНОМУ_БАЗИСУ).

Для полной формулировки алгоритма нам нужно несколько вспомогательных определений. Пусть V — модуль над (коммутативным) кольцом R. Мы говорим, что $(v_1, \ldots, v_k)$ есть *базис* (для) V над R, если все v_i лежат в V и для каждого элемента v из V имеется представление вида $rv = \sum_{i=1}^{k} r_i v_i$, где r и r_i принадлежат R и r отличен от нуля, причем такое представление единственно с точностью до умножения на элементы кольца R. *Базис* называется *целым*, если во всех таких представлениях можно взять $r = 1$, т. е. каждый элемент модуля V может быть представлен в виде линейной комбинации элементов v_i с коэффициентами из R.

КОУТС

Вход:

F(X, Y): уравнение кривой (возможно, в представлении с многими переменными, например, $F(X, Y) = 0$, $G(X, Y, Z) = 0$).

D: Дивизор конечных полюсов на этой кривой

Выход:
V: Базис K-пространства функций f, таких, что $(f) \geqslant D$.

[1] V := (1, Y, Y**2, Y**(степень F по Y — 1)).

В случае представления с многими переменными мы берем в качестве V множество всевозможных произведений алгебраических переменных со всевозможными показателями, меньшими, чем их степень.

Теперь V — базис (но не целый базис в смысле приведенного выше определения) для $K[X]$-модуля функций, порядки полюсов которых в конечных точках не выше, чем D.

[2] Для каждой точки P, над которой лежат плейсы из D, V := РЕДУКЦИЯ_К_ЦЕЛОМУ_БАЗИСУ (V, P, та часть D, которая лежит над P).

Каждое исполнение этого шага придает V нужные кратности в тех плейсах из D, которые лежат над P. Теперь V — базис для $K[X]$-модуля функций с конечными полюсами не хуже, чем D, причем этот базис — «квазицелый» в следующем смысле: любой элемент рассматриваемого модуля может быть представлен как линейная комбинация элементов базиса V с коэффициентами, являющимися частными элементов $K[X]$, где знаменатели не имеют нулей в конечных плейсах, входящих в D.

[3] Теперь добиваемся того, чтобы этот базис стал целым.

[3.1] Z := произвольный элемент поля K (например, GENSYM).

[3.2] A := определитель матрицы A(I, J), где A(I, J) = коэффициент при X**0 в разложении V(I) в J-м плейсе над Z.

[3.3] Для всех корней (в Z) числителя или знаменателя A V := РЕДУКЦИЯ_К_ЦЕЛОМУ_БАЗИСУ (V, Z, 0-дивизор над Z).

[4] V := РЕДУКЦИЯ_К_НОРМАЛЬНОМУ_БАЗИСУ (V, плейсы кривой, лежащие над бесконечно удаленной точкой).

После этого шага элементы целого базиса имеют максимальный возможный порядок (т. е. минимальный возможный полюс) на бесконечности. Элементы этого базиса V, умноженные на подходящие степени X, составляют искомый базис.

[5] V := Для каждого V(I), имеющего порядок N $\geqslant$ 0 на бесконечности

Собрать (V(I), V(I)*X, ..., V(I)*X**N).

В действительности N должен быть минимальным порядком элемента V(I) по всем плейсам над бесконечностью.

Я еще не провел анализа времени работы этого алгоритма, но на практике шаг 3 почти всегда оказывается самым трудоемким. Он нужен, чтобы мы не ввели в самом начале или в процессе вычислений зависимостей между элементами базиса V в какой-либо точке Z, не встречающейся в дивизоре D. Для многих конкретных формулировок задачи можно доказать, что это и не может случиться, и я предполагаю, что имеется широкий класс задач, для которого это верно и к которому можно свести все задачи. Однако я не достиг больших успехов в этой области.

Если не учитывать шаг 3, то может показаться, что время работы алгоритма пропорционально сумме кратностей в дивизоре, так как делается именно столько с точностью до сложных граничных эффектов шагов РЕДУКЦИЯ_К_ЦЕЛОМУ_БАЗИСУ. Ситуация, однако, не столь проста, ибо такие шаги, как правило, увеличивают размер элементов базиса V. Я в действительности думаю, что время экспоненциально зависит от суммы кратностей, но не проводил строгого анализа. Зависимость времени от степени многочленов $F(X, Y)$ очевидным образом намного хуже, но и здесь у меня нет точной формулы. Одна из нежелательных особенностей этого алгоритма состоит в том, что он начинает с довольно простого базиса на шаге 1, затем шаг 2 дает целый базис с огромными выражениями, а шаг 4 радикально сокращает их величину зачастую до той же сложности, что и в первоначальном базисе. Поэтому кажется, что гораздо лучшая формулировка этого фундаментального алгоритма еще ждет своего открытия. Иногда упомянутое выше разбухание можно предотвратить, выполняя шаги редукции на бесконечности (что можно делать между шагами редукции в других точках), но зачастую это оказывается пустой тратой времени (так как редукцию на бесконечности с тем же успехом можно сделать в самом конце) и вообще не уменьшает величину выражений.

РЕДУКЦИЯ_К_ЦЕЛОМУ_БАЗИСУ

Вход:

V: базис $K[X]$-модуля.

P: значение X, в котором базис нужно сделать целым.

D: дивизор полюсов, полностью лежащий над P.

Выход:

V: модифицированный базис, целый над P.

[1] Пусть A(I, J) — коэффициент при (X — P)**N(J) в разложении V(I) в J-м плейсе из D, где N(J) — кратность, с которой этот плейс входит в D.

Могут возникнуть трудности, если (X — P) не является локальным параметром, так что в действительности мы должны рассмотреть некоторую дробную степень $(X - P_э)$. Однако алгоритм легче описать без этого усложнения.

[2] Пока матрица A сингулярна, делать:

[2.1] Пусть {B(I)} — набор элементов поля K, такой, что SUM(B(I)*A(I, J)) = 0 для всех J, но не все B(I) равны нулю. Считаем, в частности, что B(K) не равен нулю.

[2.2] V(K) := SUM(B(I)*V(I))/(X — P)

После этого шага элементы $V(I)$ все еще линейно независимы над $K(X)$ и имеют в P полюса не хуже, чем D.

[2.3] Вычислить заново A(K, J) так же, как на шаге [1].

[3] Выдать значение V.

РЕДУКЦИЯ_К_НОРМАЛЬНОМУ_БАЗИСУ

Вход:

V: Целый базис для $K[X]$-модуля.

D: Плейсы кривой, лежащие на бесконечности.

Выход:

V: Нормальный целый базис для $K[X]$-модуля.

[1] Упорядочить элементы базиса V(I) по убыванию минимального порядка по всем плейсам на бесконечности.

[2] Пусть A(I, J) — коэффициент при (1/X)**0 в разложении V(I) по степеням 1/X в J-м плейсе из D.

Если 1/X не является локальным параметром, то могут возникнуть трудности, так что в действительности мы должны рассмотреть некоторую дробную степень 1/X. Однако алгоритм легче описать без этого усложнения.

[3] Пока A сингулярна, делать:

[3.1] Пусть B(I) — набор элементов поля K, такой что SUM(B(I)*A(I, J)) = 0 для всех J, но не все B(I) равны нулю. Считаем, в частности, что B(K) не равен нулю.

Среди всех таких B[I] мы хотим найти такой, который входят элементы V(I) наибольшего порядка т. е. первый, который будет найден, если применять к матрице A алгоритм исключения Гаусса без перестановок.

[3.2] V(K) := SUM(B(I)*V(I))/X

Тогда элементы V(I) все еще линейно независимы над $K(X)$ и имеют в P полюса не хуже D.

[3.3] Упорядочить элементы V(I) так же, как на шаге [1].
[3.4] Вычислить заново A(K, J) так же, как на шаге [2].
[4] Выдать значение V.

ДОКАЗАТЕЛЬСТВО КОРРЕКТНОСТИ АЛГОРИТМА. ШАГИ [1] — [3]

Введем некоторые обозначения. Допустим, что K — некоторое поле алгебраических чисел и L — его алгебраическое замыкание. Пусть $F(X, Y) = 0$ — определяющее уравнение нашей алгебраической кривой [1]), причем его коэффициенты лежат в K, а его степень по y равна $n \geqslant 1$. Будем предполагать, что полином F неприводим над L. Пусть, кроме того, он имеет n различных неразветвленных плейсов Q_i, лежащих на бесконечности. Пусть R — поле $\{L(x, y) \mid F(x, y) = 0\}$. Пусть дано $s \geqslant 1$ различных элементов Q_h $(1 \leqslant h \leqslant s)$ поля L. Будем обозначать через A_{hl} $(1 \leqslant l \leqslant r_h)$ плейсы [2]) поля R, лежащие над (конечными) точками $x = a_h$, а через e_{hl} соответствующие индексы ветвления, так что $\sum_{l=1}^{l=r_h} e_{hl} = n$. Наконец, мы предположим, что для каждого A_{hl} дано положительное целое число ν_{hl}, и обозначим через M множество всех функций g, имеющих порядок $\geqslant -\nu_{hl}$ в A_{hl} и порядок $\geqslant 0$ в любом другом плейсе поля R (включая плейсы на бесконечности). Таким образом, мы рассматриваем набор A_{hl} и ν_{hl} как дивизор, а M как множество функций, дивизор которых больше, чем рассматриваемый дивизор полюсов.

Если a — любой элемент поля L и A_i $(1 \leqslant i \leqslant r)$ — плейсы поля R, лежащие над точкой $x = a$ (и имеющие в этих плейсах индексы ветвления соответственно e_i), то мы следующим образом сопоставим плейсам A_i $(1 \leqslant i \leqslant r)$ целые числа ν_i $(1 \leqslant i \leqslant r)$. Если a есть одна из точек a_h, упомянутых в предыдущем абзаце, то $\nu_i = \nu_{hl}$, в противном случае полагаем $\nu_i = 0$. Положим $\sum_{i=1}^{r} \nu_i = V$. Пусть теперь M' — множество всех функций g в R, порядки которых в A_i больше или равны $-\nu_i$ для всех элементов a поля L. Мы видим, что M' есть $L[x]$-модуль размерности n. Первым шагом построения базиса множества M над L будет построение целого базиса множества M' над $L[x]$. (Говорят, что элементы $w_1, \ldots, w_n$

---

[1]) Это не препятствует использованию в вычислениях многомерных представлений; просто мы предпочли формулировать и доказывать эти результаты об алгоритме Коутса в представлении с помощью примитивного элемента. При реализации алгоритма Коутса я использую представление с несколькими переменными.

[2]) Коутс называет их нормированиями в своей статье (Coates, 1970), но для нас эти два термина равнозначны.

множества M' образуют *целый базис* множества M' над $L[x]$, если они линейно независимы над $L[x]$ и любой элемент множества M' можно выразить в виде их линейной комбинации с коэффициентами из $L[x]$.)

Пусть $w_1, \ldots, w_n$ лежат в M'. Мы можем записать разложение w_j в плейсе A_i в виде $w_j = X^{-v} \sum w_{jik} x_k$, где $X^e = (x-a)$, и потому X — локальный параметр в плейсе A с индексом ветвления e. Пусть $D(a)$ — определитель, j-я строка которого состоит из лексикографически упорядоченных по индексам элементов w_{jik}, таких, что $1 \leqslant i \leqslant r$ и $0 \leqslant k \leqslant e_i$.

Лемма 1 (эквивалентная лемме 6 из работы (Coates, 1970)). *Элементы $w_1, \ldots, w_n$ образуют целый базис множества M' тогда и только тогда, когда $D(a)$ отличен от нуля для любого a из L.*

Доказательство. Чтобы установить необходимость, допустим, что $D(a) = 0$ для некоторого a из L. Тогда строки матрицы должны быть линейно зависимы над L и, значит, существуют элементы $x_1, \ldots, x_n$ поля L, не все равные нулю, такие, что $\sum_{j=0}^{n} w_{jik} x_j = 0$ для $1 \leqslant i \leqslant r$ и $1 \leqslant k \leqslant e_i$. Но тогда

$$w = (x-a)^{-1} \sum_{j=1}^{j=n} x_j w_j$$

принадлежит M' и показывает, что первоначальные элементы w_j не составляли целого базиса.

Допустим теперь, что $D(a)$ всегда отличен от нуля, и докажем достаточность. Запишем в виде $w_i^{(1)}, \ldots, w_i^{(n)}$ полный список элементов, сопряженных с w_i и принадлежащих расширению поля $L(x)$, в котором $F(x, y)$ раскладывается на линейные множители (считаем, что $w_i^{(j)}$ упорядочены соответствующим образом). Пусть $d(w_1, \ldots, w_n)$ — определитель[1]) i, j-й элемент которого есть $w_i^{(j)}$. Тогда $d(w_1, \ldots, w_n)$ — элемент поля $L(x)$, не равный тождественно нулю. Действительно, если мы возьмем точку a, над которой лежит n неразветвленных плейсов, то мы имеем $d(w_1, \ldots, w_n) = D(a)(x-a)^v +$ более высокие степени $(x-a)$ (так как разложение w_i в A_j — это формальный степенной ряд по $(x-a)$, представляющий $w_i^{(j)}$) и $D(a)$ отличен от нуля. По-

[1]) В действительности d определен лишь с точностью до множителя ± 1, так как порядок строк и столбцов определителя безразличен, но это не доставляет нам никаких хлопот. В лемме 6 из работы (Coates, 1970) d определяется как квадрат нашего определителя, чтобы избавиться от этой неопределенности, но мы не будем этого делать. Я благодарен д-ру Норману, который указал на это упрощение.

этому w_i образуют базис множества M' и любой элемент w этого множества можно единственным образом представить в виде $wq(x) = \sum_{j=1}^{j=n} q_j(x) w_j$, где $q(x)$ и $q_j(x)$ — полиномы от x, не имеющие общего множителя. Если мы сможем показать, что $q(x)$ — константа, то мы получаем целое выражение w через w_j, и лемма будет доказана. Поэтому предположим для приведения к противоречию, что $q(x)$ имеет множитель $x - b$. Положим $q(x) = (x-b)q'(x)$ и рассмотрим

$$z = \frac{\sum_{i=1}^{i=r} q_i(b) w_i}{(x-b)} = q'(x) \frac{\sum_{i=1}^{i=r} q_i(x) - q_i(b)}{(x-b)}.$$

Каждый член последнего выражения принадлежит множеству M'; значит, и z принадлежит M'. Но константы $q_i(b)$ не все равны нулю (так как иначе q и все q_i имели бы общий множитель $(x-b)$ вопреки предположению) и $D(b)$ отличен от нуля; поэтому имеется пара (i, k) с $k < e_i$, такая, что $\sum_{j=0}^{j=n} q_j(b) w_{jik}$ отлична от нуля. Поэтому z не может лежать в M', что и дает искомое противоречие.

Эта лемма подсказывает идею *редукционного шага*, который состоит в замене одного из элементов w_i первоначального базиса на элемент w, вычисленный в первой половине леммы (равенство (1)), что дает новый базис $w_1', \ldots, w_n'$, «более близкий» к целому базису, чем исходный. Описанный выше алгоритм РЕДУКЦИЯ_К_ЦЕЛОМУ_БАЗИСУ делает нужное число таких шагов над точкой P, причем мы избрали для удобства вычислений модель кривой, над которой P неразветвленная.

Лемма 2. *Описанный процесс завершается через конечное число шагов.*

Доказательство. Заметим сначала, что после редукционного шага, в котором участвует точка a поля L, мы имеем $d(w_1, \ldots, w_n) = d(w_1', \ldots, w_n')(x-a)$ с точностью до постоянного множителя (т. е. элемента из L), так как мы заменили один из w_i на элемент, имеющий лишний множитель $(x-a)$ в знаменателе. Положим теперь $A(x) = \prod_{h=1}^{h=s} (x-a_h)^{v_h} v_h$, где $v_h = \sum_{i=1}^{i=r_h} v_{hi}$. Тогда элементы множества $A(x)M'$ вообще не имеют конечных полюсов, и $d(w_1, \ldots, w_n)A(x)$ в действительности лежит в $L[x]$, так как каждое слагае-

мое суммы, которой равен определитель, содержит по одному члену из каждого класса сопряженных, а потому может содержать лишь один вклад каждого v_{hi} для каждого h. Тогда процесс должен завершиться, так как d убывает на каждом редукционном шаге и ограничен снизу.

ДОКАЗАТЕЛЬСТВО КОРРЕКТНОСТИ АЛГОРИТМА. ШАГИ [4] — [5]

К этому моменту построен целый базис, скажем $(w_1, \ldots, w_n)$ пространства M'. Пусть разложения элементов w_j в плейсах Q_i (над бесконечностью) заданы равенствами $w_j = X^{-l_j} \sum_{k=0}^{\infty} w_{ijk} X^k$, где $X = 1/x$ — локальный параметр в Q_i (так как этот плейс неразветвленный), а l_j — целое число, такое, что коэффициент w_{ij0} отличен от нуля для некоторого плейса Q_i (но не обязательно для всех таких плейсов). Базис называется *нормальным целым базисом*, если определитель D с элементами w_{ij0} отличен от нуля. Будем предполагать, что элементы w_i упорядочены таким образом, что $l_i \geqslant l_{i+1}$ $(1 \leqslant i < n)$ и числа l_i неотрицательны для $i \leqslant l$ (может, разумеется, оказаться, что $l = 0$).

Это подсказывает идею *редукционного шага на бесконечности*, который делается, когда $D = 0$. В этом случае имеются $x_1, \ldots, x_n$, такие, что $\sum_{j=1}^{j=n} x_j w_{ij0} = 0$ $(1 \leqslant i \leqslant n)$, и, обозначая через x_h последний ненулевой x_i, мы заменяем базис $(w_1, \ldots, w_n)$ на новый базис, в котором w_h заменен на $w_h' = \sum_{j=1}^{n} x_j x^{l_j - l_h} w_j$ (где степень переменной x неотрицательна в силу упорядочения элементов w_i).

Лемма 3. *Если w_i образуют целый базис пространства M', то редукционный шаг на бесконечности снова дает целый базис.*

Доказательство. Очевидно, что новые элементы w_i лежат в M', поэтому единственный вопрос заключается в том, образуют ли они целый базис. Но ведь редукционный шаг на бесконечности изменяет определитель $d(w_1, \ldots, w_n)$ разве лишь на постоянный множитель, так как w_h заменяется на линейную комбинацию строк, в которую w_h входит с постоянным коэффициентом.

Лемма 4. *Мы можем сделать лишь конечное число редукционных шагов на бесконечности.*

Доказательство. После редукционного шага на бесконечности l_h увеличивается не менее чем на 1, а остальные l_i не

меняются (так как не меняются соответствующие w_l). Поэтому $\sum_{l=1}^{i=n} l_l$ увеличивается не менее чем на 1 при каждом редукционном шаге на бесконечности. С другой стороны, $\sum_{l=1}^{i=n} l_l$ не превосходит порядка $d(w_1, \ldots, w_n)$ на бесконечности (так как $w_j^{(i)}$ имеет порядок не меньше l_j в любом из Q_k), и, как отмечено в предыдущей лемме, $d(w_1, \ldots, w_n)$ почти не меняется при редукционном шаге. Поэтому $\sum_{i=1}^{i=n} l_i$ ограничена сверху.

Лемма 5 (эквивалентная лемме 9 из статьи (Coates, 1970)). *Если* w_j $(1 \leqslant j \leqslant n)$ *— нормальный целый базис пространства* M', *то* $\{x^h w_j\colon 1 \leqslant j \leqslant l,\ 0 \leqslant h \leqslant l_j\}$ *— L-базис множества M.*

Доказательство. Очевидно, что элементы рассматриваемого множества линейно независимы над L и лежат в M, поэтому остается только доказать, что любая функция g из M может быть выражена в виде L-линейной комбинации этих элементов. Так как g лежит в M' и w_j $(1 \leqslant j \leqslant n)$ образуют целый базис M', мы можем записать $g = \sum_{j=1}^{j=n} q_j(x) w_j$, где $q_j(x)$ лежит в $L[x]$. Пусть m_j — степень q_j как многочлена от x, а m есть $\min(l_j - m_j)$ по всем j, для которых $q_j(x)$ не равен тождественно нулю. Тогда мы можем записать $q_j(x)$ в виде $c_j(1/x)^{m-l_j} +$ (более высокие степени $(1/x)$). Следовательно, g в плейсе Q_i имеет вид $(1/x)^m \sum_{j=1}^{j=n} c_j w_{ij0} +$ (члены более высокого порядка по $1/x$). Так как D отличен от нуля, хотя бы одна из сумм $\sum c_j w_{ij0}$ также отлична от нуля. Тогда порядок функции g в этом плейсе Q_i равен m. Так как g принадлежит M, мы получаем отсюда, что $m \geqslant 0$, а, значит, $m_j \leqslant l_j$, а это показывает, что g лежит в пространстве, порожденном рассматриваемым множеством.

ОБОБЩЕНИЯ

Описанный выше вариант алгоритма Коутса работает только для дивизоров, целиком лежащих над конечными точками. Очевидно, что это ограничение преодолимо, и можно указать много способов его обхода. Наиболее очевидный метод — преобразовать кривую и пространство, в котором она лежит, таким образом, чтобы бесконечно удаленная прямая

(или в общем случае гиперплоскость) не проходила больше ни через какие плейсы рассматриваемого дивизора. Здесь возникают две неприятности. Во-первых, такое преобразование весьма затруднительно, если у нас больше одной алгебраической величины, а во-вторых, оно вводит совершенно иллюзорное множество алгебраических сущностей. Даже если учесть сделанное в разд. «Представление алгебраических величин» предыдущей главы замечание о том, что эти величины можно (и нужно) хранить отдельно от только что упомянутых, нам все же пришлось бы поработать, чтобы установить неприводимость и вычислить разложения Пюизо.

Прием, реализованный в программме, позволяет избежать обеих этих трудностей. Пусть N — наибольший (т. е. самый отрицательный) порядок на бесконечности. Добавим N ко всем порядкам на бесконечности (с учетом всех ветвлений, которые могут иметь место) и вычтем N из всех порядков, соответствующих плейсам, лежащим над точкой 0 [1]). Теперь все наши полюса лежат над конечными плейсами, и мы можем применить алгоритм Коутса в описанном выше виде. Мы умножаем затем все ответы на X^N и имеем базис, удовлетворяющий всем требованиям с единственной оговоркой; это базис для функций с полюсами порядка N над всеми бесконечными плейсами. Для редукции этих порядков до нужных нам следует лишь применить некоторые линейные ограничения, так что легко получается базис для функций с полюсами не хуже нужных.

До сих пор мы рассматривали только дивизоры полюсов. Однако алгоритм Коутса используется в основном для того, чтобы узнать, является ли данный дивизор дивизором некоторой функции, а для этого мы должны найти базис (который будет одномерным, если он вообще существует) пространства функций, имеющих те же нули и полюса, что и данный дивизор. Очевидный способ, которым я и воспользовался, состоит в том, чтобы взять (найденный алгоритмом Коутс) базис множества функций, имеющих полюса не хуже, чем у рассматриваемого дивизора, а затем считать нули последовательностью линейных ограничений [2]) и решать получающееся множество линейных уравнений. Можно подумать, что этот прием не обязательно сработает. Действительно, не все функции, выданные алгоритмом Коутса, имеют все рас-

[1]) Если в дивизоре не было плейсов, лежащих над $X = 0$, то мы должны ввести (с кратностью 0) все плейсы на кривой, лежащие над $X = 0$. Возможно, эффективность увеличится, если выбрать значение, отличное от $X = 0$, чтобы избежать необходимости проводить редукцию над дополнительным значением X, но я этого не делал.

[2]) Это — ограничения того же типа, что и введенные в предыдущем абзаце для перемещения полюсов из бесконечности, так что мы можем решить одну систему линейных уравнений, а не две.

сматриваемые полюса, и не видно, почему их линейная комбинация, имеющая нужные нули, будет иметь и все полюса. На самом деле, если дивизор имеет степень 0 (а иначе он не был бы дивизором функции), то любая из выданных алгоритмом Коутса функций, имеющих все нули, должна иметь и все полюса, так как она должна иметь именно это количество полюсов и не может иметь других, ибо все элементы базиса не имеют других полюсов.

Это решение очень хорошо работает в большинстве случаев, но иногда возникают трудности практического характера. Рассмотрим в $K(X)$ задачу нахождения функции с нулем порядка 1000 в $X-1$ и полюсом порядка 1000 в X (решением, очевидно, является $\left(\frac{X-1}{X}\right)^{1000}$). Нашим базисом для функций с полюсами порядка $\leqslant 1000$ в X, не имеющих иных полюсов, может быть последовательность $1, 1/X, 1/X^2, \ldots, 1/X^{1000}$. Тогда мы получаем систему линейных уравнений с матрицей порядка 1000×1000, которую мы должны решить, чтобы найти подходящую функцию. Хотя метод получения этих уравнений подсказывает, что они должны обладать весьма значительными структурными особенностями, непосредственное вычисление ликвидирует их разреженность, так как решение — это строка, i-м элементом которой является биномиальный коэффициент $\binom{1000}{i}$, а ненулевых элементов нет) и делает нахождение решения практически невозможным. Это конкретное затруднение можно преодолеть, если приспособить (очевидным образом) алгоритм Коутса к случаю, когда заданы и полюса, и нули, а затем начать с функции $(X-1)^{1000}$, но это решение не работает для алгебраических задач общего типа. Рассмотрим кривую, определяемую многочленом $F(X, Y) = Y^2 - (X^2 + 1)$, и попытаемся найти функцию с полюсом порядка 1000 в одном плейсе над X и нуль порядка 1000 в другом (ответом, как читатель может проверить, будет $\left(\frac{X}{Y-1}\right)^{1000}$). Здесь нашей первоначальной догадкой была бы функция X^{1000} и нам пришлось бы сделать 2000 шагов редукции на X и промежуточный базис содержал бы плотные (т. е. неразреженные) полиномы порядка 2000 по X. Эти примеры с большими числами приведены, чтобы проиллюстрировать, что этот алгоритм и основанные на нем исследования сильно страдают от экспоненциального роста.

Другой приводимый ниже метод [1]) (который реализован в последней версии системы интегрирования) устраняет все

[1]) Я чрезвычайно благодарен проф. сэру Суиннертон-Дайеру, который предложил этот подход.

эти трудности. Пусть мы хотим найти функцию с полюсами в $P_1, \ldots, P_n$ и нулями в $Q_1, \ldots, Q_n$. Пусть f — функция с полюсами в P_1 и P_2 и нулем в Q_1. Такая функция [1]) легко может быть найдена с помощью алгоритма Коутса. Пусть теперь P' — другой нуль функции f, а f' — функция, которая (находится рекурсивно и) имеет полюса в $P', P_3, \ldots P_n$ и нули в $Q_2, \ldots, Q_n$. Тогда ответом в первоначальной задаче будет функция $f*f'$.

Мы приводим ниже формулировку этого алгоритма. Заметим, что для фиксированной кривой и фиксированных плейсов, над которыми лежит дивизор, время работы (приблизительно) пропорционально произведению (число плейсов)* * $\log_2$(наибольшая кратность).

ДИВИЗОР_В_ФУНКЦИЮ

Вход: F(X, Y): Уравнение алгебраической кривой (возможно, в представлении с несколькими переменными).

D: дивизор степени 0 на этой алгебрической кривой.

Выход: FUN: функция на этой кривой, у которой дивизор нулей и полюсов совпадает с D, или НЕУДАЧА, если такой функции нет.

[1.1] POLES := Все полюса из D с соответствующими кратностями.
ZEROS := Все нули из D с соответствующими кратностями.

[1.2] FUN := 1.
В FUN мы накапливаем ответ по мере редукции дивизора D.

[2] Пока (количество нулей) $\geqslant$ 1, делать:

[2.1] P1 := Любой элемент списка POLES.
P2 := Любой элемент списка (POLES — P1).
Заметим, что P1 и P2 могут совпасть, если полюс входит в POLES с кратностью $>$ 1. Теоретически безразлично, какой элемент выбирается из POLES, но на практике время работы уменьшается, если P1 и P2 выбираются таким образом, чтобы дивизор P1 + P2 как можно чаще был делителем дивизоров из POLES. Этого можно достичь, если взять в качестве P1 элемент, входящий в POLES с наибольшей кратностью, а в качестве P2 элемент, входящий с наибольшей кратностью в POLES′, где POLES′ получается из POLES уменьшением вдвое (и отбрасыванием дробной части) кратности элемента P1.

---

[1]) Если она существует — см. шаги 2.2.1 и 2.4.1 ниже и обсуждение после описания алгоритма.

[2.2] V := КОУТС(F(X, Y),— P1 — P2).
[2.2.1] Если V = НЕУДАЧА,
то перейти к шагу [4].
[2.3] Z := некоторый элемент списка ZEROS.
Здесь снова теоретически безразлично, какой элемент мы выбираем, но время работы программы уменьшается, если выбирать элемент наибольшей кратности.
[2.4] V := Та линейная комбинация элементов базиса V, которая имеет нуль в Z.
[2.4.1] Если V = НЕУДАЧА,
то перейти к шагу [4].
[2.5] Z′ := Другой нуль функции V.
[2.6] Пока (P1 + P2)|POLES и Z|ZEROS, делать
[2.6.1] POLES := POLES — P1 — P2.
[2.6.2] ZEROS := ZEROS — Z.
[2.6.3] Если Z′|ZEROS
То ZEROS := ZEROS — Z′.
Иначе POLES := POLES + Z′.
[2.6.4] FUN := FUN*V.
[3] Если список ZEROS пуст,
то выдать окончательный результат FUN.
[4] (Завершающий шаг)
Мы приходим сюда, если в списке POLES еще остались полюса, с которыми мы не можем разделаться попарно с помощью действий из [2.6]. Это может произойти либо потому, что в POLES остался всего один элемент (обычный случай), либо потому, что один из шагов [2.2], [2.4] выдал значение НЕУДАЧА (исключительный случай). Исключительный случай рассматривается ниже более подробно.
[4.1] V := КОУТС(F(X, Y),— POLES).
[4.2] Если ZEROS = нули функции V,
то выдать результат FUN*V,
Иначе выдать НЕУДАЧА.

Как уже сказано в комментарии к шагу [4], промежуточные функции FUN не обязаны существовать, даже если окончательная задача разрешима [1]). Это проблема возникает

[1]) Это показывает следующий пример (за который я признателен проф. М. Ф. Зингеру). Пусть C — кривая рода $g \geqslant 3$, не являющаяся гиперэллиптической (т. е. которую нельзя представить в виде 2-листного накрытия проективной прямой, а это, по существу, означает, что она не имеет вида $y^2 = p(x)$). Пусть f — рациональная (отличная от константы) функция на C с минимальным числом k нулей и полюсов. Тогда $k > 2$, так как C негиперэллиптическая (см. любую стандартную книгу об алгебраических кривых). Следовательно, на C нет рациональных функций, имеющих только два полюса, и применение алгоритма ДИВИЗОР_В_ФУНКЦИЮ к (f) даст ответ НЕУДАЧА.

в основном потому, что может вообще не быть функций, имеющих только 2 полюса (хотя это не может случиться на кривых рода 1). Мы можем продолжить, используя обычную процедуру КОУТС, как указано в описании шагов [4.1] и [4.2]. Однако здесь, быть может, есть место для более изощренного процесса добавления полюсов по одному до тех пор, пока пространство функций не станет 2-мерным, использования линейной комбинации этих функций для удаления нуля из дивизора и итерации, — хотя я представления не имею, насколько хорошо такой алгоритм будет вести себя на практике. Совершенно очевидно, что вся эта глава об алгоритме Коутса лишь слегка поцарапала поверхность, которая должна стать областью плодотворных исследований для специалистов по компьютерной алгебре.

АЛГОРИТМ КОУТСА И ДИФФЕРЕНЦИАЛЫ

Мы можем применять алгоритм Коутса не только к функциям, но и к дифференциалам. Мы можем записать дифференциал в виде $F(X, Y)\,\mathrm{d}X$ для некоторой функции F и должны затем рассмотреть наши локальные параметры в каждом плейсе на кривой. Для точки, над которой лежит наш плейс, имеются следующие три возможности.

1) Плейс может лежать на бесконечности.

В этом случае наш локальный параметр есть $t = (1/X)^{1/k}$ для некоторого натурального числа k, поэтому $\mathrm{d}t/\mathrm{d}X = t^{-(k+1)}$, так что порядок нашего дифференциала на бесконечности оказывается на $k+1$ меньше, чем порядок функции F.

2) Плейс может лежать над точкой, являющейся корнем Y.

В этом случае наша локальная переменная есть $t = (X-a)^{1/k}$ для некоторого натурального числа k. Это дает $\mathrm{d}X/\mathrm{d}t = t^{k-1}$, так что порядок дифференциала на $k-1$ больше, чем порядок функции.

3) Точка не обладает ни одним из этих свойств.

В этом случае $X-a$ является локальным параметром и дифференциал имеет тот же порядок, что и функция.

Таким образом, для нахождения базиса K-пространства дифференциалов с полюсами не хуже данного дивизора нам нужно только изменить кратности в дивизоре для (конечного числа) плейсов, удовлетворяющих сформулированным только что условиям 1) и 2), а затем применить алгоритм Коутса для нахождения базиса функций $F(X, Y)$.

В частности, мы можем найти род алгебраической кривой, так как он совпадает с размерностью пространства диффе-

ренциалов первого рода (т. е. не имеющих полюсов). Дивизор, полученный из нулевого с помощью преобразований из пунктов 1) и 2), называется *каноническим дивизором* кривой.

РЕАЛИЗАЦИЯ

Так как алгоритм Коутса будет краеугольным камнем наших алгоритмов интегрирования, имеет смысл потратить время на внимательное рассмотрение его реализации. Один из важных технических приемов — минимизация количества разложений Пюизо, вычисляемых при работе алгоритма Коутса. В частности, если мы уже знаем разложения Пюизо для A и B, то нам не нужно заново вычислять разложение Пюизо для $A+B$ с самого начала. Мы можем воспользоваться тождеством, которое показывает, что разложение Пюизо коммутирует со сложением. Во время работы алгоритма РЕДУКЦИЯ_К_ЦЕЛОМУ_БАЗИСУ может оказаться, что целая строка матрицы $A(I, J)$ нулевая. Если I-я строка равна нулю, то мы можем немедленно заключить, что $V(I)$ нужно заменить на $V(I)/(X-P)$, и мы можем это сделать и заново вычислить I-ю строку матрицы. Далее мы не должны снова разлагать эту новую функцию в различных плейсах, а можем взять первоначальные разложения Пюизо и получить разложение нового $V(I)$, просто сдвинув степенные ряды на один член. Этот прием особенно хорошо согласован с нашим методом получения разложений Пюизо, описанным в гл. 2, так как новые члены можно вычислять на любом шаге, используя в то же время полученные ранее частичные результаты.

Применив алгоритм Коутса к дивизору kD, мы можем захотеть применить его к $(k+1)D$. Наивное применение описанного выше алгоритма КОУТС (не использующее алгоритма ДИВИЗОР_В_ФУНКЦИЮ) начиналось бы каждый раз с одного и того же исходного базиса на шаге [1] алгоритма КОУТС, игнорируя таким образом при вычислении $(k+1)D$ всю информацию, накопленную при вычислении kD. В действительности мы можем начинать шаг [2] с нормального целого базиса, полученного на шаге [4] обработки дивизора kD, так как этот базис состоит из n линейно независимых элементов множества M'. Преимущество такого начального приближения состоит в том, что его элементы гораздо легче превратить в функции, имеющие полюса из $(k+1)D$, чем элементы, которые были бы вычислены на шаге [1]; в частности, они имеют полюса, входящие в kD. Если D имеет общую степень d, то количество полюсов (с учетом степени), которые нужно ввести в базис при вычислении согласно алгоритму Коутса для $D, 2D, \ldots, kD$ равно

kd вместо $k(k+1)d/2$ при использовании очевидного алгоритма. Соответствующий пример приводится в гл. 5, где в разд. «Метод Кэли» показано, что, применяя этот прием, можно (если использовать алгоритм КОУТС, а не ДИВИЗОР_В_ФУНКЦИЮ) сэкономить 5.3 с. из общего времени 38.2 с.

Если мы используем алгоритм ДИВИЗОР_В_ФУНКЦИЮ, то можем начинать итерации снова на шаге [3] и добавлять полюса и нули дивизора D, который представляет собой разность между $(k+1)D$ (для которого мы должны вычислить функцию) и kD (для которого мы почти закончили вычисление).

ВЫВОДЫ

Имеются две основных области применения той алгебро-геометрической трактовки выражений, которая намечена в этой и предыдущей главе. Первая, составляющая основной предмет нашей книги, — это интегрирование алгебраических функций. Вторая — нахождение целых (или рациональных) решений систем алгебраических уравнений. В этой области была проделана большая работа (см. обзор (Swinnerton-Dyer, 1976)), но до сих пор здесь применялись специально подобранные арифметические приемы, причем вся алгебраическая геометрия вручную переводилась на какой-нибудь алгоритмический язык, например на Фортран.

Мы поставили задачу о нахождении всех функций с заданным множеством полюсов, которая тривиальна для рациональных функций, и описали алгоритм Коутса, дающий ее решение. Он позволяет затем дать ответы на следующие вопросы: соответствует ли данный дивизор какой-либо функции, каков род кривой и каковы дифференциалы первого рода? Позднее мы увидим, что как раз эти вопросы возникают в процессе интегрирования алгебраических функций.

Глава 4

ТЕОРЕМА РИША

ВВЕДЕНИЕ

Среди различных областей компьютерной алгебры интегрирование демонстрирует, пожалуй, наибольшее несоответствие между ожиданиями пользователей и реальными характеристиками имеющихся систем. Хотя эвристические приемы интегрирования хорошо разработаны (Moses, 1967), хотелось бы иметь разрешающую процедуру, способную, например, доказать, что данное выражение неинтегрируемо (Moses, 1971).

В этой главе излагается теория, лежащая в основе отыскания (или доказательства несуществования) элементарных интегралов от алгебраических функций. При этом функция называется *алгебраической*, если ее можно породить из переменной интегрирования и констант с помощью арифметических операций и нахождения корней алгебраических уравнений [1]), причем допускается вложенность (это понятие уточняется ниже). (См. гл. 2, где подробно рассматриваются такие выражения и выбранное нами для них представление.) *Элементарность* означает возможность порождения из переменной интегрирования и констант с помощью арифметических операций, использования корней, экспонент и логарифмов, причем допускается вложенность. Частным случаем этой проблемы является вопрос о том, когда интеграл, представляющийся на первый взгляд эллиптическим, может быть в действительности выражен через элементарные выражения. Эту задачу долго считали неразрешимой, и Харди сформулировал классическую точку зрения следующим образом:

> «Не было придумано метода, с помощью которого мы всегда могли бы определить за конечное число шагов, является ли данный эллиптический интеграл псевдоэллиптическим, и проинтегрировать его в этом случае; есть основания думать, что такого метода и нельзя дать» (Hardy, 1916, с. 47—48).

Мы изложим новый алгоритм для этой задачи, использующий аппарат алгебраической геометрии (см. предыдущие

[1]) Теория не требует, чтобы эти корни обязательно выражались радикалами. Как упомянуто в гл. 2, имеющаяся в настоящее время реализация ограничивается алгебраическими величинами, выразимыми через квадратные корни.

главы), обсудим его реализацию и основные открытые проблемы, оставшиеся в этой области. Мы применяем без пояснений терминологию и идеи предыдущих глав. Отметим, что наш подход отличается от подхода Нга (Ng, 1974), где целью было нахождение канонических форм для неэлементарных эллиптических и гиперэллиптических интегралов. Соотношение между этими двумя подходами к родственным, но различным задачам еще не вполне ясно.

Можно подумать, что легко поставить задачу «интегрируема ли данная функция?», но это не так. В частности, предположим, что у нас есть алгоритм, распознающий неразрешимость алгебраических функций над K, где K — поле, порожденное из целых чисел, $\log 2$ и π с помощью операций сложения, вычитания, умножения, деления и функций $x \Rightarrow \log x$, $x \Rightarrow \exp x$, $x \Rightarrow |x|$. Пусть теперь $f(x)$ — какая-нибудь неинтегрируемая функция, например $\sqrt{(1-x^2)(1-kx^2)}$, где k отлично от 0 и 1. Тогда для выражений A, не зависящих от x, вопрос об интегрируемости $Af(x)$ эквивалентен вопросу о равенстве $A = 0$, а последняя проблема алгоритмически неразрешима в силу результата Ричардсона (Richardson, 1968)[1]). Эти соображения аналогичны приведенным у Риша (Risch, 1969, предложение 2.2), и мы отсылаем читателя к его статье за дальнейшими разъяснениями по поводу вычислимости. Результаты Риша показывают, что наше рассмотрение интегрируемости нуждается в прочной теоретической основе, а ее дает дифференциальная алгебра, к описанию которой мы и переходим.

ДИФФЕРЕНЦИАЛЬНАЯ АЛГЕБРА

Перед тем как рассматривать задачу интегрирования, нам нужно определить, что мы понимаем под интегрированием и интегралом. Очевидный ответ и принимаемое нами определение состоят в следующем: $F(x)$ есть интеграл от $G(x)\,dx$ тогда и только тогда, когда $G(x)$ есть производная по x от $F(x)$. Таким образом, определение интегрирования просто сводится к определению дифференцирования. Изучение дифференцирования как алгебраической (а не аналитической) операции — компетенция *дифференциальной алгебры* — области математики, основанной Дж. Риттом, хотя ее простейшие результаты восходят к Лиувиллю и Лапласу. Чтобы ввести терминологию, нужную для исследования дифференцирования и интегрирования, мы приводим здесь очень краткое резюме элементарной дифференциальной алгебры. Дальнейшие све-

[1]) Эта фраза уточняет формулировку, приведенную в оригинале. — *Прим. перев.*

дения можно найти в книге Капланского (Kaplansky, 1957), дающей введение в предмет, обзоре Ритта (Ritt, 1950) или монографии Колчина (Kolchin, 1973), содержащей полное изложение дифференциальной алгебры.

Под *дифференциальным полем* мы будем понимать поле[1]) вместе с семейством D_i одноместных операторов (записываемых в виде $a \to Da$) на этом поле (дифференцирований поля), удовлетворяющих следующим условиям:

$$D(a+b) = Da + Db,$$
$$D(ab) = aDb + bDa$$

для любых элементов a, b поля и любого дифференцирования D. Поле K является *дифференциальным расширением* поля L, если K — расширение L как поля и любое дифференцирование поля K можно считать ограничением некоторого дифференцирования поля L. Мы будем иногда писать a' вместо Da, если имеется только одно дифференцирование.

Элемент, имеющий образ 0 при любом дифференцировании, называется константой поля. Мы сразу получаем $D0=0$, а затем $D1 = 0$, так что любое целое число является константой в только что определенном смысле. Всюду, кроме гл. 6, мы будем рассматривать только одно дифференцирование, а именно операцию, обратную к интегрированию, которое мы хотим выполнить. Если X — переменная, по которой мы хотим интегрировать (переменная интегрирования), то мы можем превратить $\{K(X, Y) \mid F(X, Y) = 0\}$ в дифференциальное поле, полагая по определению $a' = 0$ для всех a из K, $X' = 1$ и $Y' = -(F_X/F_Y)$, где F_X обозначает формальную частную производную от F по X (т. е. производная от cX^n равна cnX^{n-1}, и это частное дифференцирование является аддитивной операцией). В более общей ситуации, например в случае представлений с многими переменными, если $F(X, Y, Z, \ldots, W) = 0$, то $0 = F' = X'F_X + Y'F_Y + Z'F_Z + \ldots + W'F_W$, и это определяет W' через $X, Y, Z, \ldots, W$. Это очевидным образом превращает поле в дифференциальное поле, а интегрирование определяется как операция, обратная к дифференцированию, всюду, где она определена.

Если K — дифференциальное поле, x, y принадлежат K, причем y отличен от нуля и $Dx = (Dy)/y$ для любого дифференцирования D поля K, то мы говорим, что x есть *логарифм* y или что y есть *экспонента* (от) x. Отметим, что это согла-

[1]) Если не оговорено противное, считается, что характеристика поля равна 0. «Сам Ритт не видел никакого проку в полях ненулевой характеристики и называл их „дурацкими полями"» (Kolchin, 1973, с. xiii). Я надеюсь показать ниже (см. в частности гл. 8), что эти поля могут быть полезны в теории интегрирования.

суется со стандартными (аналитическими) определениями, так как уравнение $Dx = (Dy/y)$ обычно принимается в качестве определения экспоненциальной функции в классическом дифференциальном исчислении, а логарифм определяется как обращение экспоненты. Дифференциальное расширение L поля K называется *элементарным расширением* K, если оно имеет вид $K(t_1, \ldots, t_n)$, где для каждого $1 \leqslant i \leqslant n$ выполняется одно из следующих условий:

t_i — логарифм некоторого элемента поля $K(t_1, \ldots, t_{i-1})$;
t_i — экспонента некоторого элемента поля $K(t_1, \ldots, t_{i-1})$;
t_i — алгебраический элемент над $K(t_1, \ldots, t_{i-1})$.

С помощью этого определения уточняется понятие элементарной функции: это функция, которая может быть представлена в виде элемента некоторого элементарного расширения поля констант.

ТЕОРЕМА РИША

Интеграл от рациональной функции является суммой рациональной функции и логарифмов с постоянными коэффициентами. Точно так же интеграл от алгебраической функции, если он элементарен, является суммой алгебраической функции и взятых с постоянными коэффициентами логарифмов алгебраических функций. Мы скажем, что такой интеграл состоит из *алгебраической части* и *логарифмической части*. Интеграл, не имеющий логарифмической части, называется *чисто алгебраическим*, а интеграл без алгебраической части — *чисто логарифмическим*.

Первый важный результат о виде этих интегралов — принцип Лапласа (Hardy, 1916, с. 9—10), утверждающий, что интеграл от алгебраической функции можно преобразовать так, чтобы он содержал лишь те алгебраические величины, которые встречаются в подынтегральном выражении. Этот результат можно доказать «элементарными» средствами, и Харди намечает такое доказательство (с. 36—42 и 46 упомянутой работы). Однако мы получим это из наших общих результатов — см. следствие 2 ниже.

При интегрировании рациональной функции логарифмическая часть происходит в точности от интегрирования членов вида $1/(X - k)$ в разложении интегрируемой функции на простейшие дроби, а алгебраическая часть — от интегрирования всех остальных членов. Обобщением этого замечания на алгебраические функции является следующее утверждение, сформулированное Ришем (Risch, 1970).

Теорема. *Пусть* w *— дифференциал в поле* $\{K(X, Y) \mid F(X, Y) = 0\}$. *Пусть* $r_1, \ldots, r_k$ *— базис* Z*-модуля, порожденного вычетами дифференциала* w, *так что в каждом* K*-плей-*

се P дифференциал w имеет вычет $\sum_{i=1}^{k} a_{iP} r_i$, *где* a_{iP} *лежат в Z. Пусть дивизор* d_i *задан кратностями* a_{iP} *в плейсах P. Тогда, если* $\int w$ *элементарен, то найдутся элементы* $v_0, \ldots, v_k$ *поля* $K(X, Y)$ *и целые числа* $j_1, \ldots, j_k$, *такие, что* d_i *в степени*[1]) j_i *есть дивизор функции* v_i *и*

$$w = dv_0 + \sum_{i=1}^{i=k} \frac{r_i}{j_i} \frac{\mathrm{d}\, v_i}{v_i},$$

т. е.

$$\int w = v_0 + \sum_{i=1}^{i=k} \frac{r_i}{j_i} \log v_i.$$

В этой теореме v_0 — алгебраическая часть ответа, а остальное — логарифмическая часть, зависящая только от вычетов интегрируемой функции. Это — непосредственное обобщение стандартного метода интегрирования рациональных функций, однако может оказаться, что дивизоры рационально (а не линейно) эквивалентны нулю. В этом последнем случае j_i и есть тот целый множитель, который превращает d_i в дивизор функции. Эта теорема непосредственно подсказывает следующий алгоритм интегрирования алгебраических функций.

РИШ_АЛГЕБРАИЧЕСКИЙ

Вход:
F(X, Y): уравнение алгебраической кривой.
f(X, Y): функция от X и Y
(w будет тогда равен $f(X, Y)\,\mathrm{d}X$)

Выход:
I: интеграл от $f(X, Y)\,\mathrm{d}X$
(или НЕ_ЭЛЕМЕНТАРЕН, если не существует элементарного интеграла).

[1] ПОТЕНЦИАЛЬНЫЕ_ПОЛЮСА:=«БЕСКОНЕЧНОСТЬ» вместе со всеми множителями знаменателя функции $f(X, Y)$ (которая является многочленом по X в силу замечаний о представлении алгебраических величин, сделанных в гл. 2).

[2] ВЫЧЕТЫ:= для каждого U из списка ПОТЕНЦИАЛЬНЫЕ_ПОЛЮСА
для каждого плейса V, лежащего над U
Создать совокупность (V, вычет функции f(X, Y) в плейсе V).

[1]) В смысле умножения дивизоров, введенного в гл. 3: все кратности умножаются на j_i, а плейсы сохраняются.

[3] Z_БАЗИС := базис **Z**-модуля вычетов, созданного на шаге [2].
[4] Для каждого Ri из списка Z_БАЗИС делать:
[4.1] Di := для каждого U из списка ВЫЧЕТЫ создать совокупность (Плейс. коэффициент при Ri в данном вычете).
Это соответствует дивизору d_i в теореме Риша.
[4.2] Ji := НАХОЖДЕНИЕ_ПОРЯДКА Di
По теореме Риша интеграл может существовать только в случае, когда Di — дивизор конечного порядка.
Этот шаг дает нам порядок дивизора Di, т. е. наименьшую степень, в которой он линейно эквивалентен нулю. Если такого порядка нет, дается ответ БЕСКОНЕЧНОСТЬ.
[4.3] Если Ji = БЕСКОНЕЧНОСТЬ, то выдать ответ НЕ_ЭЛЕМЕНТАРЕН.
[4.4] ИНТЕГРАЛ := ИНТЕГРАЛ + (Ri/Ji)*log(ДИВИЗОР_В_ФУНКЦИЮ (F(X, Y), Di**Ji))
Мы нашли теперь все логарифмические части интеграла и можем удалить их производные из подынтегральной функции, чтобы найти алгебраическую часть.
[5] Теперь находим алгебраическую часть ответа.
[5.1] f(X, Y) := f(X, Y) — производная функции ИНТЕГРАЛ.
[5.2] АЛГ_ЧАСТЬ := НАХОЖДЕНИЕ_АЛГЕБРАИЧЕСКОЙ_ЧАСТИ $f(X, Y)$.
[5.3] Если АЛГ_ЧАСТЬ = НЕ_АЛГЕБРАИЧЕСКАЯ, то выдать ответ НЕ_ЭЛЕМЕНТАРНАЯ.
В этом случае мы можем найти все логарифмические части интеграла, но не его алгебраическую часть.
[6] Выдать ответ ИНТЕГРАЛ + АЛГ_ЧАСТЬ.

Этот алгоритм легко реализовать, как только определены вспомогательные алгоритмы. Многие из них — хорошо известные алгоритмы компьютерной алгебры, хотя это может быть не сразу видно из их описания. Разложение на множители на шаге [1] производится только в $K[X]$ в силу сделанного в гл. 2 замечания о том, что мы можем представлять элемент поля $\{K(X, Y) \mid F(X, Y) = 0\}$ как частное, числитель которого — многочлен от X и Y, а знаменатель — многочлен от X.

Задачу о построении базиса **Z**-модуля на шаге [3] можно переформулировать как вопрос о нахождении линейно независимого подмножества строк матрицы. Ответ находится очевидным способом, который, однако, не эффективен для больших матриц. Правда, этот вопрос не должен нас слишком заботить: ведь если наша матрица велика, то породившее ее подынтегральное выражение почти наверняка и само слищ-

ком велико для того, чтобы его можно было проинтегрировать.

Проблема нахождения вычета в данном плейсе требует вычисления члена с t^{-1} в разложении Пюизо (где t — локальный параметр в рассматриваемом плейсе). Как уже сказано в гл. 2, разложения Пюизо — это, по существу, разложения в ряд Лорана, и их можно вычислять описанным в гл. 2 методом, который основан на приемах из работы (Norman, 1975). Хотя программа, выполняющая эти разложения правильно и эффективно, получается довольно длинной, возникающие трудности носят скорее технический, чем математический характер.

Алгоритм ДИВИЗОР_В_ФУНКЦИЮ описан в гл. 3. Алгоритм НАХОЖДЕНИЕ_ПОРЯДКА сложнее, и мы вернемся к нему в гл. 6.

ДОКАЗАТЕЛЬСТВО ТЕОРЕМЫ РИША

Риш получил упомянутую выше теорему как следствие своей основной теоремы об интегрировании (доказательство которой не было включено в его опубликованные работы, хотя и может быть получено из работы (Risch, 1969, с. 171 и сл.)). Поэтому необходимо доказать эту теорему прежде, чем мы сможем положить ее в основу построения теории интегрирования. Теорема сформулирована в виде эквивалентности, но если w имеет указанный вид, то она очевидным образом элементарна. Поэтому основная задача — доказать, что все элементарные дифференциалы имеют такой вид. Мы не будем приспосабливать первоначальное доказательство Риша (в котором речь шла главным образом о трансцендентных функциях), а дадим новое и более короткое доказательство, опирающееся на один недавний результат (Rosenlicht, 1976).

Воспользуемся интуитивным наблюдением: дифференцирование не устраняет ни экспонент, ни логарифмов, кроме случая, когда последние встречаются в сумме $\ldots + c \log f(x)$. Кроме того, дифференцирование не устраняет алгебраических выражений. Поэтому с интуитивной точки зрения естественно ожидать, что интеграл от алгебраической функции, если он элементарен, может быть представлен в виде $u(x) + \sum c_i \log u_i(x)$, где c_i — константы. Это замечание было сделано Лапласом и впервые доказано Лиувиллем (Liouville, 1833) с помощью изобретательной индукции спуска. Это доказательство можно найти в книге Ритта, где описаны результаты Лиувилля (Ritt, 1948, с. 20). Наш подход будет основан на более современных методах абстрактной алгебры, так как представляется правдоподобным, что именно они должны служить источником полезных обобщений.

Лемма 1. *Для функции $f(X)$ из $L=\{K(X, Y, Z, \ldots) \mid F(X, Y)=0,\ G(X, Y, Z)=0 \ldots\}$, где K алгебраически замкнуто, дифференциал $f\mathrm{d}X$ элементарен (в том смысле, что имеется элементарное расширение поля L, содержащее элемент g, такой, что $f=g'$) тогда и только тогда, когда имеются константы $c_1, \ldots, c_n$, линейно независимые над целыми числами в L, и элементы $u_1, \ldots, u_n, v$ поля L, такие, что $f=v'+\sum c_i \frac{u_i'}{u_i}$.*

Доказательство. Если не требовать линейной независимости констант c_i, то это — ограничение теоремы 3 из работы (Rosenlicht, 1976) на случай одного дифференцирования специального вида. А если c_i линейно зависимы над целыми числами, то мы можем добиться линейной независимости, объединяя их и пользуясь тождеством $D(u^k)/(u^k)=kDu/u$.

Следствие 2. *Элементарный интеграл от алгебраической функции не может содержать иных алгебраических величин, зависящих от переменной интегрирования, кроме тех, которые встречаются в подынтегральной функции* [1]).

Доказательство. Если интеграл элементарен, то интегрируемая функция имеет вид, указанный в предыдущей лемме, а ее интеграл равен $v+\sum c_i \log u_i$, где v и u_i принадлежат дифференциальному полю, в котором лежит исходная подынтегральная функция, а потому представимы через те же алгебраические величины.

Отметим важность фразы «зависящие от переменной интегрирования» в приведенной выше формулировке принципа Лапласа. Без этого ограничения принцип неверен. Например, Риш (Risch, 1969, предложение 1.1) рассматривает интеграл от $1/(X^2-2)$, который обязательно содержит $\sqrt{2}$.

Лемма 3. *Поле $K((t))$ степенных рядов от одной переменной — это дифференциальное поле, где $\left(\sum a_i t^i\right)'=\sum i a_i t^{i-1}$.*

Доказательство. Соотношение для сложения очевидно, соотношение для умножения получается сравнением левой и правой части.

[1]) Это — принцип Лапласа. Харди (Hardy(1916, с. 9—10) цитирует Лапласа следующим образом: «L'intégrale d'une fonction différentielle (algébrique) ne peut contenir d'autres radicaux (sic) que selles qui entrent dans cette fonction». Я не смог найти в точности эту формулировку в какой-либо работе Лапласа, но обнаружил следующую: «L'intégrale d'une fonction différentielle ne peut renfermer d'autres quantités exponentielles et radicales que celles qui sont contenues dans fonction» (Laplace, 1820, т. 1, с. 5). Заметим, что Лаплас не утверждает в явной форме, что это относится только к алгебраическим выражениям, зависящим от переменной интегрирования. Другое доказательство можно найти у Ритта (Ritt, 1948, с. 28—31).

Лемма 4. *Если* x *— переменная дифференцирования и* $dz = y\,dt$, *то* $y = z'$, *где* $'$ *— дифференцирование «в смысле степенных рядов», описанное в предыдущей лемме.*

Доказательство. Это теорема 8 книги (Lang, 1957), с. 247.

Следствие 5. *Если* t *— локальная переменная, то разложение Пюизо функции* $f'\,dx/dt$ *есть производная (в смысле леммы 3) разложения Пюизо функции* f.

Следствие 6. *Если* F *имеет чисто алгебраический интеграл и разложение Пюизо функции* F *имеет вид* $\sum a_i t^i$, *то разложение Пюизо ее интеграла имеет вид* $\sum \frac{a_i t^{i+1}}{i+1}$.

Лемма 7. *Разложение Пюизо для производной от* $\log F(X)$ *совпадает с разложением* $F'(X)/F(X)$. *В частности, если* $F(X)$ *имеет нуль (или полюс) порядка* n *в плейсе* P, *то* $(\log F(X))'$ *имеет в этом плейсе вычет* n *(соответственно* $-n$*).*

Доказательство. Первая часть — это в точности данная ранее характеризация логарифмов. Вторая часть получается путем рассмотрения разложений Пюизо. Если F имеет разложение $a_n t^n + \ldots$, то F' имеет разложение $n a_n t^{n-1} + \ldots$ и частное имеет вид $nt^{-1} + \ldots$, что и требовалось.

Доказательство теоремы. Если w — выражение указанного вида, то оно очевидным образом интегрируемо. Для доказательства обратного утверждения предположим, что w интегрируемо. Тогда по лемме 1 мы можем записать w в виде $v\,dx + \sum c_i (\log u_i)'\,dx$. Можно считать, что ни одна из u_i не является константой, так как иначе она не могла бы внести вклада в производную интеграла. Выберем выражение указанного вида с минимальным n. В силу леммы 7 вычет w в плейсе P равен $\sum c_i b_{iP}$, где b_{iP} — порядок u_i в плейсе P. Далее, так как каждая u_i имеет хотя бы один полюс и c_i линейно независимы, каждый из коэффициентов c_i входит хотя бы в один вычет выражения w. Следовательно, c_i образуют базис для множества вычетов выражения w (или, возможно, для большего модуля, который в любом случае включает только дробные кратные вычетов). Поэтому, если $\{r_i\}$ — наш базис для множества вычетов выражения w, рассматриваемого как Z-модуль, то имеются целые числа j_i, такие, что множество $\{r_i/j_i\}$ есть базис множества $\{c_i\}$, и мы можем записать c_i в виде $\sum \frac{n_{ij} r_j}{j_j}$. Теперь мы можем переписать сумму логарифмов так, чтобы коэффициентами оказались r_i/j_i, и теорема доказана.

Следствие 8. *Если интегрируемое выражение не имеет вычетов, то его интеграл не может иметь логарифмической части.*

АЛГЕБРАИЧЕСКАЯ ЧАСТЬ

На шаге 5.2 обрисованного выше алгоритма интегрирования мы получаем функцию, интеграл которой, если он вообще элементарен, является чисто алгебраическим. Эта задача намного проще, чем общий случай; она была, по существу, решена более 150 лет назад (Liouville, 1833a, 1833b). Мы опишем метод решения этой задачи, который хорошо взаимодействует с остальным материалом и часто оказывается очень эффективным, так как выясняется, что многие вычисления уже проделаны в процессе нахождения логарифмической части интеграла. Имеется несколько путей решения этой задачи, и описываемый здесь не обязательно самый эффективный[1]); просто он ближе всего к нашей алгебро-геометрической точке зрения на задачу интегрирования.

Этот алгоритм основан на тесной связи между полюсами алгебраической функции $f(X, Y)$ и ее производной $f'(X, Y)$ (см. ниже следствие 6). В действительности, если f' имеет где-то полюс порядка n, то f имеет там полюс порядка $n-1$ (или порядка $n+1$, если плейс лежит на бесконечности). Это утверждение — обобщение обычного правила интегрирования степеней переменной X, которое легко можно доказать, рассматривая разложения Пюизо. Мы можем сказать немногое о нулях функции f, так как разложение Пюизо для f может иметь постоянный член, пропадающий при дифференцировании.

В этом процессе есть одно трудное место. Мы не можем полностью найти интеграл с помощью этого метода, так как интеграл определен лишь с точностью до постоянной интегрирования. Вначале я экспериментировал с различными стратегиями, позволяющими узнать, возникает ли неопределенность в линейной комбинации функций с данными полюсами из-за этой константы интегрирования или по иной причине, но не смог найти разумного способа делать это. Поэтому я принял следующую стратегию: выбирается точка P (не являющаяся полюсом подынтегральной функции) и требуется, чтобы алгебраическая часть интеграла принимала там

[1]) В случае простых расширений с помощью радикалов теория сильно упрощается и принадлежит, по существу, Чебышеву (Chebyshev, 1853). В работе (Trager, 1979) недавно опубликован алгоритм нахождения алгебраической части в этом случае, который кажется очень эффективным, так как он использует только полиномиальные алгоритмы; однако я не смог обнаружить его реализации и провести сравнение продолжительности работы.

значение 0, т. е. выбирается конкретное значение постоянной интегрирования. Это приводит иногда к любопытным значениям интегралов, но оставляет очевидную возможность эвристического второго захода с целью выбора более «соблазнительных» значений константы интегрирования после того, как уже найден весь интеграл.

НАХОЖДЕНИЕ_АЛГЕБРАИЧЕСКОЙ_ЧАСТИ

Этот алгоритм описан для простых частных случаев. Хотя возникают осложнения, когда $(X-a)$ не является локальным параметром в точке $X=a$, общий принцип тот же самый.

Вход:
F(X, Y): уравнение алгебраической кривой
f(X, Y): функция от X и Y
(тогда w будет обозначать $f(X, Y)\,dX$)

Выход: ФУНКЦИИ: Интеграл от $f(X, Y)\,dX$
(или НЕ_АЛГЕБРАИЧЕСКИЙ, если нет алгебраического интеграла).

[1] ПОТЕНЦИАЛЬНЫЕ_ПОЛЮСА:=«БЕСКОНЕЧНОСТЬ» и все множители знаменателя функции f(X, Y) (который является многочленом по X).

[2] ДИВИЗОР := для каждого U из ПОТЕНЦИАЛЬНЫЕ_ПОЛЮСА
для каждого плейса V, лежащего над U
Создать совокупность (V, порядок функции $f(X, Y)$ в V)).
Этот порядок вычисляется с помощью разложения Пюизо функции f(X, Y) в V.

[3] Для любого U из ДИВИЗОР
ПОРЯДОК := минимум (0, ПОРЯДОК + 1),
или ПОРЯДОК–1, если плейс лежит над бесконечностью. Знаки + и — поменялись местами по сравнению с приведенным выше описанием теории, так как порядок полюса всегда отрицателен.

[4] ФУНКЦИИ := КОУТС (F(X, Y), ДИВИЗОР).
В действительности здесь нам нужна модификация алгоритма КОУТС, которая может работать с плейсами на бесконечности, но, как объяснено в гл. 3, ее не очень трудно построить.

[5] Для P = 1, 2, ... делать:
если никакой элемент совокупности ДИВИЗОР не лежит над P, то перейти к шагу 6.
Этот цикл должен в конце концов закончиться, так как ДИВИЗОР — конечное множество полюсов. Тогда мы можем потребовать, чтобы интеграл был равен 0 в точке P, и найти постоянную интегрирования.

[6] Пусть A[i] — значение функции ФУНКЦИИ[i] в точке P.

Устраняем один из элементов множества ФУНКЦИИ, используя ограничение $\sum A[i] = 0$.

[7] Для каждого U из ДИВИЗОР, имеющего ПОРЯДОК $\neg = 0$, проделываем следующее.

Для каждого члена At^{-k} разложения Пюизо функции $f(X, Y)$ добиваемся, чтобы интеграл имел член $-At^{-k'}/k'$, где k' есть $k+1$ для конечных плейсов и $k-1$, когда U лежит на бесконечности. Это достигается путем применения линейного ограничения к списку ФУНКЦИИ, построенному на шаге 4, и на практике эти линейные ограничения разрежены для положительных k. Мы не можем сделать этого для $k' = 0$.

[8] Когда список ФУНКЦИИ сведется к единственному элементу, продифференцировать его. Если мы получили $f(X, Y)$, то найден интеграл, а если нет — то $f(X, Y)$ вообще не имеет алгебраического интеграла, и ответом будет НЕ_АЛГЕБРАИЧЕСКИЙ.

Этот алгоритм способен установить неинтегрируемость стандартных функций, например функции $1/\sqrt{(x^2-1)(x^2-k)}$, которая не имеет полюсов.

ЧАСТИЧНЫЙ АЛГОРИТМ

Даже при отсутствии эффективной реализации алгоритма НАХОЖДЕНИЕ_ПОРЯДКА описанные выше методы дают нам частичный алгоритм интегрирования алгебраических функций. Если ему предъявлена интегрируемая алгебраическая функция, то он завершит работу и выдаст интеграл, если же предъявленная функция не интегрируема, то он может выдать ответ НЕ_ЭЛЕМЕНТАРНА, но может и не кончить работу.

Мы напомним, что порядок дивизора на алгебраической кривой определен как наименьшая его степень, которой соответствует некоторая функция, т. е. для которой алгоритм Коутса применяется к нему с положительным ответом. Следовательно, мы можем заменить процедуру НАХОЖДЕНИЕ_ПОРЯДКА другой, которая перебирает по поряку все степени дивизора Di, пока не найдется нужная. Эта процедура, разумеется, не кончит работу, если дивизор не является рационально эквивалентным нулю.

Такой подход показывает, что нам нужен не точный порядок дивизора, а лишь его граница сверху. Если мы допустим, что все дивизоры имеют порядок 1, то получим правильный ответ для кривых рода 0, т. е. для интегралов, которые можно взять тригонометрическими подстановками. Более изощренные методы нахождения порядка, рассматриваемые ниже, отражают трудность распознавания элементарных ин-

тегралов среди тех, которые кажутся эллиптическими или гиперэллиптическими.

Заметим, что этот частичный алгоритм полон для интегралов, не содержащих логарифмической части, так как их подынтегральные функции не имеют вычетов.

СООБРАЖЕНИЯ ЭФФЕКТИВНОСТИ

Описанный алгоритм может оказаться не очень эффективным даже в случае интегралов, соответствующих дивизорам порядка 1. Рассмотрим, например, интегрирование функции

$$\frac{1}{\sqrt{x^2+1}} + \frac{100}{\sqrt{x^2+10\,000}}.$$

Она имеет вычеты, равные 99, 101, —99, —101 в четырех плейсах, лежащих на бесконечности. Вызов процедуры ДИВИЗОР_В_ФУНКЦИЮ на шаге 4.4 нашего алгоритма даст ответ

$$\frac{x^{101}}{(\sqrt{x^2+1}-1)(\sqrt{x^2+10\,000}-100)^{100}},$$

а в предыдущей главе уже говорилось, что нахождение этой функции наивными методами требовало бы решения неразреженной системы линейных уравнений порядка 200×200. Даже используя метод, описанный в алгоритме ДИВИЗОР_В_ФУНКЦИЮ, мы должны вычислить окончательный результат, а его знаменатель — это четный неразреженный многочлен степени 100 по X. К счастью, эта функция имеет специальную структуру, и ее можно получить в неразвернутой форме, так что проблема не столь трудна, как может показаться. Некоторые случаи такого рода приведены в примере 8 из приложения 2. Так как интегрирование каждого члена в отдельности почти не добавляет работы, ясно, что мы должны это делать, как только представляется возможность. Таким образом мы уменьшим среднее время работы алгоритма, даже если максимальное время работы не понизится. Иными словами, имеется большой простор для эвристик, улучшающих функционирование системы, даже если они и не увеличивают область разрешимых задач.

Одна из предшествующих версий программы, описываемой в этой монографии, а также пакет трансцендентного интегрирования (Norman & Moore, 1977) были включены (Harrington, 1977, 1979b) вместе с распознавателем образцов и набором правил для интегралов, заданных пользователем, в алгебраическую систему REDUCE. Методы Харрингтона, использующие по мере необходимости только что упомянутые методы интегрирования, дают, по-видимому, наилучший подход к практической системе интегрирования, так как они сочетают изящество и четкость алгоритмического подхода со скоростью, которую дает полуэвристический подход.

Глава 5

ЗАДАЧА О ДИВИЗОРАХ С КРУЧЕНИЕМ

ВВЕДЕНИЕ

Основное содержание этой и трех последующих глав — теоретические и практические соображения, относящиеся к процедуре НАХОЖДЕНИЕ_ПОРЯДКА, которая, как мы видели в предыдущей главе, является неотъемлемой частью нашего алгоритма интегрирования и которая оказывается наиболее трудной в математическом отношении. В этой главе мы обрисуем общую природу проблемы, уделив особое внимание простейшему нетривиальному случаю — задачам, где встречается квадратный корень из единственной кубики или квартики [1]) и все константы — рациональные числа.

В самом простом случае имеется лишь один корень квадратный из рассматриваемого линейного или квадратичного выражения. В этом случае [2]) процедура НАХОЖДЕНИЕ_ПОРЯДКА всегда может выдавать ответ 1, так как не все дивизоры степени 0 рационально эквивалентны нулю. Это замечание эквивалентно утверждению о том, что такую кривую всегда можно преобразовать в рациональную, так что задача сводится к интегрированию рациональной функции. Это показывает, что если в нашем алгоритме заменить процедуру НАХОЖДЕНИЕ_ПОРЯДКА на любую процедуру, выдающую число, не меньшее 1 (хотя этот ответ не обязательно правильный), то он справится со всеми «очевидными случаями», доступными эвристическому алгоритму.

К сожалению, создание полной процедуры НАХОЖДЕНИЕ_ПОРЯДКА — гораздо более трудная задача, чем любая из тех, с которыми мы имели дело до сих пор, и ее решение требует гораздо больше математики. Эту ситуацию иллюстрирует простой пример 3 из приложения 2, где интегрирование функции, содержащей SQRT(X^3+1), требует рассмотрения дивизора порядка 3, а это кажется посложнее, чем случай функции SQRT(X^2+1). Чтобы лучше понять задачу, нам придется использовать значительный объем материала из ал-

---

[1]) Это (в общем случае) кривые рода 1 или эллиптические кривые, а интегралы, определенные над ними, — это эллиптические интегралы, дающие простейшие алгебраические примеры неинтегрируемости.

[2]) Когда кривая имеет род 0. (Кривые рода 0 могут иметь и более сложный вид, но их всегда можно свести к рассматриваемому виду.)

гебраической геометрии и теории чисел и ряд недавних результатов.

Говоря неформально, трудность проистекает от того, что существуют дивизоры, которые не являются дивизорами функций, но имеют кратные, являющиеся дивизорами функций. С помощью алгоритма Коутса (гл. 3) мы можем узнать, является ли данный дивизор дивизором функции или нет, поэтому мы будем отождествлять два дивизора, если их разность является дивизором функции. Тогда наша задача сводится к распознаванию того, имеет ли данный дивизор (степени 0) кратное, которое отождествляется с 0. Такой дивизор называют *дивизором с кручением.*

Нам придется рассмотреть все виды дивизоров на алгебраических кривых, так как все они имеют отношение к задаче интегрирования [1]).

Сформулируем теперь более формально необходимые математические сведения. Дивизоры, линейно эквивалентные 0, образуют подгруппу группы всех дивизоров, и алгоритм Коутса дает нам процедуру, узнающую, лежит ли дивизор в этой подгруппе. Рассмотрим теперь факторгруппу группы всех дивизоров степени 0 по этой подгруппе. Результат, который мы назовем *якобиевой группой дивизоров*, является [2]) конечно порожденной абелевой группой. Поэтому ее можно рассматривать как прямое произведение некоторого числа бесконечных циклических групп и конечной абелевой группы (которую называют *группой кручения* исходной группы). Дивизор рационально эквивалентен нулю тогда и только тогда, когда его образ лежит в этой группе кручения, и в этом случае порядок дивизора равен его теоретико-групповому порядку как элемента этой конечной абелевой группы. Величина этой конечной части якобиевой группы дивизоров известна под названием *кручения* группы или кривой. Поэтому порядок любого дивизора, рационально эквивалентного нулю, должен быть делителем кручения кривой.

ЭЛЛИПТИЧЕСКИЕ КРИВЫЕ

В этом разделе применим сделанные выше замечания к частному случаю эллиптических кривых, т. е. кривых рода 1. Мы делаем это не только потому, что этот случай, поглощающий эллиптические интегралы, важен сам по себе, но и потому, что он позволяет на простых примерах проиллюстри-

[1]) Для любого дивизора имеется дифференциал третьего рода, среди вычетов которого имеется данный дивизор (Lang, 1972, с. 44).

[2]) В силу теоремы Морделла — Вейля (Mordell, 1922), (Weil, 1928) и (Lang & Neron, 1959) в применении к якобиеву многообразию кривой (см. ниже).

ровать проблемы, которые встанут перед нами позднее при рассмотрении кривых произвольного рода. Большая часть этих результатов описана у Касселса (Cassels, 1966); более современное изложение имеется у Тейта (Tate, 1974). Вот еще два важных соображения: во-первых, на эллиптических кривых лежат многие задачи, возникающие на практике, в частности те, которые содержат два квадратных корня из квартик или один квадратный корень из кубики или квартики. Во-вторых, имеется много специальных приемов, применимых в этом случае, но не к кривым более высокого рода (главным образом потому, что эллиптические кривые изучались в математике гораздо интенсивнее, чем кривые более высокого рода).

Наше исследование использует ряд результатов из алгебраической геометрии и теории эллиптических кривых, которые излагаются ниже. Первый шаг показывает, как снабдить рассматриваемую группу конкретной геометрической моделью.

Теорема 1 (Риман — Рох) [1]. *Если D — произвольный дивизор кривой рода 1 и $l(D)$ есть размерность K-пространства $L(D)$ функций f, таких, что $D_+(f)$ эффективен, то $l(D)$ равно 0, 1 или степени D в зависимости от выполнения условия: степень $D < 0$, $= 0$ или > 0.*

Лемма 2 [2]. *На эллиптической кривой с выделенной точкой O линейно неэквивалентные дивизоры степени 0 находятся во взаимно однозначном соответствии с точками кривой. При этом точка O соответствует нулевому дивизору.*

Поэтому мы можем задать сложение точек кривой, превратив это соответствие в групповой изоморфизм. Тогда $A+B=C$ означает, что дивизор $A+B$ линейно эквивалентен дивизору $C+O$ или что $(A-O)+(B-O)$ линейно эквивалентен $(C-O)$.

Теорема 3 [3]. *Эллиптическая кривая E с выделенной точкой O бирационально эквивалентна* [4] *кривой $Y^2 = X^3 - AX - B$ для некоторых A и B (это так называемая каноническая форма Вейерштрасса), причем точка O отображается*

---

[1]) См. (Fulton, 1969, с. 210) или (Chevalley, 1951) или (Wan der Waerden, 1949), или любую другую книгу по алгебраической геометрии. (Cassels, 1966) принимает это в качестве определения алгебраической кривой.

[2]) Этот результат доказан в работе (Cassels, 1966, с. 211).

[3]) (Cassels, 1966), теорема 7.1 на с. 211, лемма 7.1 и обсуждение на с. 213—214.

[4]) То есть можно переходить туда и обратно с помощью рациональных преобразований. (Cassels, 1966) отмечает, что традиционная терминология неестественна, так как рациональное преобразование не обязательно имеет рациональные коэффициенты.

в бесконечно удаленную точку. Далее, на этой эллиптической кривой равенство $P + Q + R = 0$ *выполняется тогда и только тогда, когда* P, Q *и* R *коллинеарны. Поэтому* (X_1, Y_1) *и* (X_2, Y_2) *имеют сумму с координатами*

$$X = \frac{((-A + X_1X_2)(X_1 + X_2) - 2B + 2Y_1Y_2}{(X_1 - X_2)^2},$$

$$Y = \frac{(X(Y_1 - Y_2) + X_1Y_2 - X_2Y_1)}{(X_1 - X_2)}.$$

При $X_1 = X_2$ эти формулы теряют смысл. В действительности при $Y_1 \neq Y_2$ точка (X, Y) лежит на бесконечности, а при $Y_1 = Y_2$ имеем

$$X = \frac{X_1^4 + 2AX_1^2 + 8BX_1 + A^2}{4Y_1^2},$$

$$Y = \frac{Y_1 + (X - X_1)(3X_1^2 - A)}{2Y_1}.$$

Мы можем рассматривать задачи о линейно неэквивалентных дивизорах степени 0 как задачи об этой группе точек. Мы можем преобразовать любой дивизор в единственную точку эллиптической кривой Вейерштрасса, а затем задавать вопросы об этой точке. Для того чтобы применять приведенную выше теорему, нам нужен ее эффективный вариант, который обеспечивается следующим алгоритмом (из работы (Baker & Coates, 1970), см. также (Baker, 1975)).

ФОРМА_ВЕЙЕРШТРАССА

Вход:

F(X, Y): уравнение, определяющее кривую рода 1.
В действительности не обязательно, чтобы это уравнение было задано через примитивный элемент. Так как алгоритм КОУТС будет работать с представлением со многими переменными, то и наш алгоритм сможет работать с ним.

D: дивизор на данной кривой.

Выход:

F′(X′, Y′): уравнение вида $Y'^2 = X'^3 + aX' + b$, бирационально эквивалентное уравнению F,

D′: дивизор на F′, соответствующий дивизору D на F при бирациональной эквивалентности F и F′.

[1] Пусть Q — плейс, лежащий над бесконечно удаленной точкой на кривой F(X, Y) = 0.

[2] С помощью алгоритма Коутса (гл. 3) найдем функцию X_1, имеющую полюс в точности порядка 2 в плейсе Q и не имеющую никаких других полюсов на F(X, Y) = 0.

[3] Аналогичным образом найдем функцию X_2, имеющую полюс порядка 3.

[4] Пусть (a, b, c, d, e, f, g) — ненулевое решение (в K) уравнения

$$a + bX_1 + cX_2 + dX_1^2 + eX_2^2 + fX_1^3 + gX_1X_2 = 0.$$

Такое решение существует, так как в уравнении участвуют 7 функций, каждая из которых имеет дивизор $\geqslant -6Q$, так что они не могут (в силу теоремы Римана — Роха) быть линейно независимы.
Здесь $e \neq 0$, так как

$$eX_2^2 + (cX_2 + gX_1X_2) + (a + bX_1 + dX_1^2 + fX_1^3) = 0.$$

и слагаемое в первых скобках обращается в нуль или имеет полюс нечетного порядка в Q, а слагаемое во вторых скобках обращается в нуль или имеет полюс четного порядка в Q. Тогда и $f \neq 0$, так как только e и f соответствуют слагаемым, имеющим в Q полюс порядка 6.

[5] $X' := X_1$
$Y' := 2eX_2 + gX_1 + c$
$A := -4ef$
$B := g^2 - 4de$
$C := 2cd - 4be$
$D := c^2 - 4ae.$

Это — эллиптическое уравнение (так как $A \neq 0$ и оно бирационально эквивалентно исходной кривой рода 1), которое выполнено в силу равенства из [4].

[6] $Y' := AY'$
$X' := AX'$
$C := AC$
$D := A^2D.$

Это — описание преобразования, возникающего при умножении всего уравнения на A^2 и замене AY' на Y' и AX' на X'. Коэффициент при X'^3 обращается тогда в 1.

[7] $K := B/3$
$X' := X' + K$
$C := C - 3K^2$
$D := D - K^3 - CK.$

Это — описание замены X' на $(X' + K)$ в предыдущем уравнении. Теперь мы имеем настоящую каноническую форму Вейерштрасса и должны только преобразовать дивизор соответствующим образом.

[8] D' := для каждого (N, P) из D создать совокупность (N, X'(P))

РЕЗУЛЬТАТ МАЗУРА

Этот математический аппарат позволяет нам немедленно улучшить частичный алгоритм, описанный в предыдущей главе. Известно ((Mazur, 1978), теорема 2; см. также (Mazur, 1977), теорема 1), что в случае эллиптической кривой (т. е. кривой рода 1, определенной над рациональными числами [1]), кручение ограничено числом 16. В действительности можно сделать более сильное утверждение: максимальный порядок любого элемента в группе кручения якобиевой группы дивизоров эллиптической кривой, определенной над рациональными числами (в смысле последнего подстрочного примечания), равен 12. Эта оценка кручения дает полный алгоритм интегрирования для кривых рода 1 над рациональными числами и тем самым решает проблему, поставленную Харди (Hardy, 1916, с. 47—48) и сформулированную во введении к настоящей главе, для случая эллиптических интегралов, определенных над рациональными числами.

Согласно давней гипотезе (которую Касселс относит к «фольклору» (Cassels, 1966, с. 264)), для любого конечно порожденного [2]) поля K имеется оценка кручения, не зависящая от кривой $\{K(X, Y) \mid F(X, Y) = 0\}$, однако математики не достигли большого прогресса в этом направлении в общем случае. По общему мнению, результаты (Демьяненко, 1971) не вполне обоснованны, и Касселс пишет в реферате этой работы: «К сожалению, изложение настолько неясно, что референту еще нужно встретить кого-нибудь, кто поручится за корректность доказательства. Но, с другой стороны, ему еще нужно показать ошибку, которая недвусмысленно и безвозвратно опровергает рассуждение».

Предложенная выше схема состояла в нахождении верхней границы для кручения и последующем переборе всех кратных дивизора вплоть до этой границы, пока алгоритм Коутса не выдаст функцию по одному из них. Эта схема практически неудовлетворительна, хотя она и дает полное решение задачи интегрирования. В следующем разделе приводятся некоторые другие методы, которые могут дополнить

---

[1]) При этом требуется, чтобы не только уравнение кривой имело чисто рациональные коэффициенты, но и плейсы, участвующие в рассматриваемом дивизоре, имели рациональные коэффициенты.

[2]) Может показаться, что здесь есть противоречие, так как в гл. 2 мы сказали, что будем рассматривать только алгебраически замкнутые поля K. Однако любая конкретная задача содержит лишь конечное число алгебраических выражений и потому может рассматриваться в конечно порожденном поле. Это замечание требует обоснования, но по существу оно очевидно, так как если дивизор имеет бесконечный порядок над некоторым полем, то он имеет бесконечный порядок и над любым алгебраическим расширением этого поля, и вся задача (и ответ) может быть задана конечным выражением.

эту общую схему и сильно увеличить эффективность системы в соответствующих случаях.

Очень простое усиление, применимое только в случае эллиптических кривых над рациональными числами, это теорема Лютца — Нагеля (см. подробности в гл. 7 или (Lang, 1978), с. 55, теорема 2.2). Она утверждает, что рациональная точка на эллиптической кривой в стандартной форме Вейерштрасса с целыми коэффициентами является точкой кручения, только если она в действительности целая [1]) (т. е. X, а потому и Y являются целыми числами). Ясно, что это очень легкая проверка, если кривая уже приведена к канонической форме.

ПРИЛОЖЕНИЕ ИССЛЕДОВАНИЙ МАЗУРА [2])

В этом разделе рассматриваются приложения результатов Мазура, описанных выше. Найденную Мазуром границу 12 для кручения любой конкретной точки (или дивизора) на эллиптической кривой, определенной над рациональными числами, можно использовать (в подходящих случаях), чтобы узнать, интегрируема ли функция. Здесь мы наметим альтернативный метод проверки интегрируемости, который использует эту границу, но не опирается на какой-либо метод построения интеграла, если последний существует. Хотя этот результат не связан непосредственно с основным направлением наших работ, он дает метод установления неинтегрируемости алгебраических выражений и опирается на многие приемы, упомянутые в наших доказательствах, но не участвующие в реализации.

Мы рассматриваем задачу об интегрировании выражения $(x-A)/f(x)$ (для подходящих A), где $f^2(x)=a_0^2x^4+a_1x^3+a_2x^2+a_3x+a_4$, причем a_i — целые числа и полином в правой части этого равенства свободен от квадратов. К этому виду может быть преобразован интеграл от любой квартики.

Лемма 4. *Если цепная дробь для $f(x)$ периодична, то это происходит немедленно.*

Доказательство. Это утверждение следует из работы (Schinzel, 1962, абзац 2).

---

[1]) Я благодарен д-ру Н. М. Стефенсу, который привлек мое внимание к этому тесту в связи с группами Морделла — Вейля эллиптических кривых.

[2]) Я благодарен профессору А. Шинцелю, который привлек мое внимание к этим исследованиям. См., однако, докторскую диссертацию Г. Золотарева и его статьи в журнале Лиувилля (Journal de Math. Pure et Appl.), (2^{me} Ser.) XIX (1874) и (3^{me} Ser.) VI (1879). Я благодарен проф. Шинцелю за указание на эти источники.

Не умаляя общности, мы можем считать, что $f(x)^2$ имеет вид $x^4 + 6bx^2 + 4cx + d$. Тогда следующий результат следует из работы (Schinzel, 1962), где делается ссылка на книгу (Halphen, 1886, vol. 1, с. 120; vol. 2, с. 591).

Лемма 5. *Точка* $(-b, c)$ *имеет порядок* r *на эллиптической кривой* $Y^2 = 4X^3 - (3b^2 + d)X - (bd - b^3 - c^2)$ *тогда и только тогда, когда цепная дробь для* $f(x)$ *имеет период* $r - 1$ *или* $2(r - 1)$.

Теорема 6. *Если* $f(x)^2$ *— квартика с рациональными коэффициентами, то функция* $(x - A)/f(x)$ *для подходящих* A *интегрируема тогда и только тогда, когда цепная дробь для* $f(x)$ *периодична с периодом длины* < 23.

Доказательство. В силу результата Мазура точка на эллиптической кривой над рациональными числами имеет порядок, не превосходящий 12, поэтому в силу предыдущей леммы имеем $r - 1 < 12$ и $2(r - 1) < 23$. Это доказывает нужный результат и дает финитный тест интегрируемости для функции $(x - A)/f(x)$.

Дополнительные результаты о взаимосвязях цепных дробей и интегрируемости можно найти у Чебышева (Chebyshev, 1857, 1860), и, по-видимому, многое еще можно сделать для того, чтобы связать цепные дроби с алгоритмами интегрирования. Так как методы работы (Trager, 1979) позволяют нам вычислять алгебраическую часть интегралов в расширении единственным радикалом и так как работы (Chebyshev, 1857, 1860) также применимы к этому случаю, ясно, что есть место для их совместного использования. Мы должны отметить, что эти цепные дроби можно эффективно вычислять с помощью подходящего обобщения «цепного» метода из работы (Churchhouse, 1976).

МЕТОД КЭЛИ

В этом разделе мы рассмотрим метод, определяющий по данному дивизору на эллиптической кривой, является ли он линейно эквивалентным нулю, не используя алгоритм Коутса. Этот метод основан на недавней статье (Griffits & Harris, 1978), в которой переформулированы некоторые результаты Кэли (Caley, 1853, 1861). Отметим сначала, что разд. «Эллиптические кривые» этой главы показывает, что нам нужно рассматривать лишь простые точки на нашей эллиптической кривой.

Теорема 7 [1]**.** *Пусть* p *— точка на эллиптической кривой* E. *Мы можем переписать* E *в виде* $y^2 = (x - a)(x - b)(x - c)$,

[1]) (Griffits & Harris, 1978), уравнение (10), являющееся переформулировкой теоремы на с. 376 работы (Cayley, 1853).

где x имеет значение 0 в p. Запишем разложение Пюизо для y в виде $\sum_{i=0}^{\infty} A_k x^k$. В этих условиях p имеет конечный порядок n тогда и только тогда, когда обращается в нуль соответствующий определитель, выписанный ниже:

$$\begin{vmatrix} A_2 & A_3 & \dots & A_{m+1} \\ A_3 & A_4 & \dots & A_{m+2} \\ \dots & \dots & \dots & \dots \\ A_{m+1} & A_{m+2} & \dots & A_{2m} \end{vmatrix} \quad n = 2m+1,$$

$$\begin{vmatrix} A_3 & A_4 & \dots & A_{m+1} \\ A_4 & A_5 & \dots & A_{m+2} \\ \dots & \dots & \dots & \dots \\ A_{m+1} & A_{m+2} & \dots & A_{2m-1} \end{vmatrix} \quad n = 2m.$$

Таким образом, мы можем применить этот тест до алгоритма Коутса к кривым рода 1, так как вычисление определителя гораздо проще, чем применение алгоритма Коутса [1]). Хотя это не обязательно понижает вычислительную сложность нашего алгоритма интегрирования для кривых рода 1, мы определенно достигаем большого улучшения для тех случаев, которые требуют рассмотрения процедуры НАХОЖДЕНИЕ_ПОРЯДКА.

Пример. Рассмотрим этот улучшенный метод в применении к кривой Тейта с $D = 2$ (см. пример 4 из приложения 2). Это дает нам кривую $y^2 = 4x^3 - 15x^2 + 8x + 16$, на которой мы хотим интегрировать функцию $P(x, y)/Q(x, y)$, где

$$P(x, y) = (-14yx^5 - 30yx^4 + 251yx^3 + 60yx^2 - 688x - 488y + 80x^6 - 320x^5 - 341x^4 + 1992x^3 + 448x^2 - 3200x - 1792)$$

и

$$Q(x, y) = 7x(4yx^5 + 5yx^4 - 51yx^3 - 4yx^2 + 112x + 64y - 20x^6 + 79x^5 + 41x^4 - 368x^3 - 32x^2 + 512x + 256),$$

что порождает дивизор порядка 7. При попытке интегрировать без использования метода Кэли мы имеем 7 применений алгоритма Коутса, занявших соответственно 1.9 с., 3.5 с., 6.5 с., 9.6 с., 16.5 с., 25.8 с. и 32.9 с. времени центрального процессора. Таким образом, полное время вычислений по алгоритму Коутса (исключая вычисление рода кривой) составляет 96.7 с из общего затраченного времени 107 с. Повторный

[1]) В действительности мы можем вычислить этот определитель еще быстрее, чем в общем случае, так как он имеет специальный (теплицев) вид (Yun & Gustavson, 1979).

запуск этого примера с использованием теста Кэли для нахождения порядка точек дает два применения алгоритма Коутса (так как мы проверяем, равен ли единице порядок дивизора раньше, чем пытаемся использовать процедуру НАХОЖДЕНИЕ_ПОРЯДКА), занимающих соответственно 1.9 и 38.2 с.[1]) из общего времени 48.3 с.

ЯКОБИЕВЫ МНОГООБРАЗИЯ

Мы можем обобщить некоторые из упомянутых понятий, введя якобиево многообразие алгебраической кривой. Нам нужно при этом понятие (проективного) *алгебраического многообразия*, которое мы определяем (Shafarevich, 1972, с. 58) как подмножество проективного n-мерного пространства[2]), заданное одновременным обращением в нуль нескольких однородных многочленов[3]). Если V и W — проективные многообразия, то мы можем превратить множество $V \times W$ в многообразие. Для проективных многообразий это неочевидно и использует возможность отобразить $P^n \times P^m$ в $P^{(n+1)(m+1)-1}$ (см. Shafarevich. 1972), с. 65—67). Функция из одного многообразия в другое называется регулярной, если в некоторой окрестности любой точки она может быть выражена с помощью однородной рациональной функции степени 0.

Многообразие V называется *алгебраической группой*, если на точках многообразия имеется групповой закон, такой, что функция f: $V \to V$, такая, что $f(x) = x^{-1}$, и такая функция g: $V \times V \to V$, что $g(x, y) = xy$, являются регулярными. Проективная неприводимая алгебраическая группа называется *абелевым многообразием.* Абелево многообразие всегда коммутативно (Shafarevich, 1972, с. 210, теорема 3). В силу теоремы 3, сформулированной выше, эллиптическая кривая с

---

[1]) Отметим, что, хотя мы рассматриваем один и тот же дивизор (с кратностями 7 и —7), время 38.2 с, затраченное на него в этом случае, отлично от 32.9 с для случая, когда метод Кэли не применялся. Это происходит потому, что в последнем случае мы уже вычислили базис пространства функций с полюсами и нулями порядка 6, который можно использовать в качестве начального приближения для алгоритма Коутса, в то время как в рассматриваемом случае мы можем начать лишь с базиса функций с полюсами и нулями порядка 1 и должны поэтому выполнить больше шагов редукции. Эти вопросы обсуждались в разд. «Реализация» гл. 3.

[2]) Проективное n-мерное пространство определяется как множество $\{(a_0, \ldots, a_n),\ a_i \in K$, не все a_i одновременно равны 0$\}$ с отношением эквивалентности $(a_0, \ldots, a_n) = (b_0, \ldots, b_n)$, если имеется $c \in K$, такой, что $a_i = b_i c$ для всех i. Мы будем обозначать проективное n-мерное пространство через P^n, когда это будет удобно.

[3]) Заметим, что не требуется, чтобы количество многочленов было равно $n - 1$ или чтобы многочлены были независимы.

выделенной точкой является абелевым многообразием, так как приведенные там формулы регулярны.

В более общей постановке мы можем на любой алгебраической кривой построить группу линейно не эквивалентных дивизоров степени 0 Ее можно превратить в многообразие, и она становится таким образом абелевым многообразием, которое называется *якобиевым многообразием* данной кривой C и обозначается через $\mathrm{Jac}(C)$. Тогда теорему 3 можно переформулировать в виде утверждения о том, что эллиптическая кривая бирационально эквивалентна своему якобиевому многообразию, и в большинстве ситуаций мы их можем отождествлять.

Глава 6

ОПЕРАТОРЫ ГАУССА — МАНИНА

ВВЕДЕНИЕ

Эта глава посвящена случаю, когда подынтегральная функция содержит кроме переменной интегрирования также и трансцендентный параметр, так что мы можем считать, что имеем дело с задачей интегрирования функции в поле $\{K(x,y)\,|\,F(u,x,y)=0\}$, где K — алгебраическое расширение поля $k(u)$, где k — некоторое поле, а u трансцендентен над k. Мы будем применять эти обозначения, считая u независимым трансцендентным элементом и применяя префикс D для обозначения дифференцирования [1]) по u и суффикс $'$ для обозначения дифференцирования по x. Этот случай зачастую легче, чем случай, когда нет трансцендентного параметра, ибо интегрирование по x коммутирует с дифференцированием по u, так что интегрируемость $G(u,x,y)$ влечет интегрируемость $DG(u,x,y)$, $D^2G(u,x,y)$ и так далее.

В этом случае мы иногда можем узнать, что дивизор не имеет конечного порядка, что влечет неинтегрируемость функции. Если мы не можем сделать этого, то мы можем найти величину u_0 из k для параметра u, такую, что дивизор $P(u)$ на $F(u,x,y)=0$ имеет конечный порядок тогда и только тогда, когда дивизор $P(u_0)$ имеет конечный порядок на $F(u_0,x,y)=0$. Иными словами, мы можем свести нашу задачу к задаче, не содержащей u. Итерируя этот процесс, мы

---

[1]) Мы собираемся дифференцировать по u, хотя u является постоянным трансцендентным параметром (таким, как e или π). Это возможно, так как из-за трансцендентности параметра он не может удовлетворять алгебраическим уравнениям, связывающим его с другими константами из подынтегральной функции, так что его точная величина неважна.

Здесь приобретает значение сделанное в гл. 1 (в разд. «Теоретические ограничения») замечание о том, что мы должны знать все зависимости между нашими константами. Например, поскольку неизвестно, являются ли числа e и π алгебраически независимыми, мы не можем рассматривать интеграл, куда они оба входят, ибо мы не знаем, как выразить dF/de через $\partial F/\partial e$ и $\partial f/\partial\pi$. Разумеется, мы могли бы допустить, что они независимы, и получить результат вида «если e и π алгебраически независимы, то F не имеет элементарного интеграла».

$'$ будет пониматься как полное дифференцирование по x с учетом зависимости y от u, вызванной функциональным соотношением $F(u,x,y)=0$ (в предположении, что u действительно входит в это соотношение). Мы имеем тогда $dy/du = -\partial F/\partial u/\partial F/\partial y$, так что $DG = \partial G/\partial u + (dy/du)\partial G/\partial y$.

можем свести дело к задаче, не содержащей трансцендентных параметров. Мы не будем рассматривать решение таких задач в этой главе, а отложим этот вопрос до следующих двух глав. Если $P(u_0)$ имеет бесконечный порядок, то, как мы знаем, то же верно для $P(u)$, и задача решена. В противном случае, обозначая через n порядок дивизора $P(u_0)$ и рассматривая по очереди $nP(u)$, $2nP(u)$, мы в конце концов узнаем, каков порядок $P(u)$, хотя понятия не имеем, когда это произойдет. В случае кривых рода 1 мы можем применить для уменьшения затрат труда тест Кэли, описанный в конце предыдущей главы.

Этот подход основан на двух статьях (Manin, 1958, 1963) советского математика Ю. И. Манина, и читатель может найти в них изложение в полной общности и доказательства; мы будем только формулировать результаты. Читать эти статьи, однако, нелегко, причем и русский, и английский варианты содержат много опечаток. Кроме того, изложение Манина осложнено его желанием работать сразу с n параметрами (и, следовательно, с n дифференцированиями) вместо одного. Мы можем, разумеется, тоже иметь n параметров $u_1, \ldots, u_n$, но мы будем элиминировать их не сразу все, а по одному, используя методы этой главы итеративно. Я не смог провести каких-либо экспериментов с кривыми с двумя параметрами и потому не составил твердого мнения о том, какой из двух подходов (параллельный или итеративный) лучше. Однако интуиция подсказывает, что желательно по возможности уменьшить размер задачи, устраняя параметры, как только это становится возможно, а не работать со всеми ними до самого конца.

ПРИМЕР

Перед тем как излагать общую теорию, мы рассмотрим развернутый пример ее применения, взятый, по существу, из статьи (Manin, 1963, с. 1397—1400). Рассмотрим «общую» эллиптическую кривую $y^2 = x(x-1)(x-u)$ и возьмем в качестве основного поля K конечное расширение поля $k(u)$ для некоторого поля k. Пусть $w = y^{-1}dx$ — дифференциал первого рода. Так как наша кривая эллиптическая, все остальные дифференциалы получаются умножением этой формы на константу, т. е. величину, не зависящую от x и y. Тогда, если C — любая замкнутая кривая на поверхности $\{K(x, y) \mid F(u, x, y) = 0\}$, то интеграл $e = \int_C w$ бесконечнозначен (и аналитичен как функция от u), причем в действительности пространство этих значений порождается любой парой незави-

симых периодов[1]) e_1, e_2. Эти функции являются решениями линейного уравнения Гаусса $4u(u-1)D^2e - 4(1-2u)De + e = 0$.

С другой стороны, функции вида $\int_O^P w$, где O — бесконечно удаленная точка, чрезвычайно важны для исследования геометрии кривой. Они удовлетворяют уравнению $I(P) + I(Q) = I(P+Q)$, где $P+Q$ — сумма в группе (см. предыдущую главу), и определены лишь с точностью до периода, так как мы можем выбрать любой путь, чтобы попасть из O в P. Мы можем устранить эту неопределенность, подействовав на обе части уравнения оператором Гаусса, который разрушает такие периоды, и получить функцию $J(P) = (4u(u-1)D^2 - 4u(1-2u)D + 1)I(P)$, которую можно отождествить с некоторым элементом поля $\{K(x,y) \mid F(u,x,y) = 0\}$. Ввиду линейности оператора Гаусса и соотношения между $I(P)$, $I(Q)$ и $I(P+Q)$ формула для J определяет гомоморфизм группы точек кривой в аддитивную группу рассматриваемого поля функций.

Предыдущим соотношениям можно придать следующий более явный вид. Заметим прежде всего, что

$$[4u(u-1)D^2 - 4(1-2u)D + 1]\, y^{-1}dx = -2d(y(x-u)^2).$$

Если мы теперь проинтегрируем по замкнутой кривой, то справа окажется интеграл от полного дифференциала, который обратится в нуль, а в левой части интегрирование коммутирует с дифференцированием по u, так что мы получаем

$$4u(u-1)D^2 - 4(1-2u)D + 1]\int_C y^{-1}dx = [4u(u-1)D^2 - 4(1-2u)D + 1]\, e(t)$$

что и дает дифференциальное уравнение Гаусса. Тогда

$$J(P) = [4u(u-1)D^2 - 4(1-2u)D + 1]\int_O^P \frac{dx}{y},$$

но на этот раз мы не можем переставить интегрирование и оператор D, так как P зависит от u. Запишем P в виде точки $(X(u), Y(u))$. Тогда, если G — алгебраическая функция от x и u, рациональная по x и y, то мы можем записать

$$D\int_O^P G(x,u)\,dx = (DX(u))\,G(X(u),u) + \int_O^P DG(x,u).$$

[1]) *Период* есть по определению интеграл дифференциала первого рода по замкнутому пути на римановой поверхности данной кривой.

Мы можем продифференцировать еще раз, чтобы получить

$$D^2 \int_O^P G(x, u)\, \mathrm{d}x =$$

$$= (D^2X(u))\, G(X, u) + (DX(u))\,[(DG(X(u), u)) + \partial G(X(u), u)/\partial u] +$$

$$+ \int_O^P D^2G(x, u).$$

В рассматриваемом частном случае это дает нам уравнение [1])

$$J(P) = -2Y(X-u)^{-2} + D\,\frac{2u(u-1)(DX)}{y} + 2u(u-1)\,D(XDY).$$

УРАВНЕНИЕ ПИКАРА — ФУКСА

Пусть C — кривая рода g, определенная уравнением $F(x, y) = 0$, где коэффициенты F лежат в алгебраическом расширении K поля $k(u)$ и действительно содержат u. Тогда мы можем найти $2g$ замкнутых кривых на $C(u)$, таких, что любая замкнутия кривая на $C(u)$ непрерывно деформируема в сумму этих $2g$ кривых [2]). Пусть $c_1, \ldots, c_{2g}$ — эти кривые, а $w_1, \ldots, w_g$ — линейно независимые дифференциалы первого рода на $C(u)$. Положим $L_{a,b} = \int_{c_b} w_a$, так что это — период кривой.

Лемма 1. *Для любого a периоды $L_{a,b}$ удовлетворяют линейному дифференциальному уравнению порядка $2g$ (или меньше в вырожденных случаях):*

$$p_{a,2g}(u)\, D^{2g}L_{a,b} + \ldots + p_{a,1}DL_{a,b} + p_{a,0}L_{a,b} = 0$$

(где само уравнение не зависит от b).

Доказательство. Если w — дифференциал без вычетов (т. е. второго рода), то Dw — тоже дифференциал без вычетов (Manin, 1963, следствие 2, с. 1404). Тогда w_a, Dw_a, ..., $D^{2g}w_a$ — $2g+1$ дифференциала второго рода, и поэтому

[1]) Отметим различие между двумя слагаемыми суммы в квадратных скобках. Первое содержит полную производную от $G(X(u), u)$ по u, а второе — только частную производную и является в действительности результатом подстановки $X(u)$ вместо x в $DG(x, u)$. Различие между ними оказывается слишком тонким для многих алгебраических систем, и это увеличивает трудности непосредственной реализации описываемых идей (см. разд. 2 приложения 1).

[2]) Точная формулировка: мы можем найти $2g$ кривых $c_1, \ldots, c_{2g}$, образующих базис 1-мерной гомологии на римановой поверхности.

между ними должна существовать линейная зависимость

$$p_{a,\,2g}D^{2g}w_a + \ldots + p_{a,\,1}Dw_a + p_{a,\,0}w_a = \mathrm{d}f$$

для некоторой функции f. Теперь мы просто интегрируем по замкнутым кривым, чтобы получить периоды, а интеграл от $\mathrm{d}f$ по любой замкнутой кривой равен 0.

Следуя Манину, мы называем полученные соотношения *уравнениями Пикара — Фукса.* Дифференциальный оператор $L = \sum p_{a,\,i}D^i$ называется *оператором Гаусса — Манина.* Мы можем снабдить пространство таких уравнений вида $L\int\limits_c w = 0$ структурой модуля, если разрешим более общее уравнение [1]) $\sum L_i \int\limits_c w = 0$. Если теперь D — дивизор вида $\sum n_i P_i$, где n_i — целые числа, то положим $\int\limits^D f(x)\,\mathrm{d}x = \sum n_i \int\limits_O^{P_i} f(x)\,\mathrm{d}x$, где нижний предел интегрирования — некоторая фиксированная точка O. Его значение очевидным образом не зависит от выбора точки O для дивизоров степени 0.

Если J — такое уравнение Пикара — Фукса, то $\sum L_i w$ есть полный дифференциал, который мы обозначим через $\mathrm{d}z$. Тогда, если P и Q — две точки на абелевом многообразии A, то мы можем определить [2]) $J(P, Q)$ как $z(Q) - z(P)$, и это определение корректно (Manin, 1963, с. 1408, теорема 1). В действительности $\sum L_i \int\limits_Q^P w_i$ и не зависит от контура интегрирования, так как интеграл от полного дифференциала по любой замкнутой кривой (т. е. разности двух контуров) равен 0.

---

[1]) Согласно более формальному определению (Manin, 1963, с. 1405), уравнение Пикара—Фукса — это любое соотношение вида J: $\sum L_i \int\limits_c w_i = 0$, где w_i — дифференциалы первого рода, а L_i — линейные дифференциальные операторы. Такие уравнения могут быть определены на любом абелевом многообразии A, но нас будет интересовать лишь случай $A = \mathrm{Jac}(C)$.

[2]) Мы используем обозначение J как для уравнения Пикара — Фукса, так и для оператора (называемого *оператором Пикара — Фукса*), который переводит (P, Q) в $z(Q) - z(P)$. Это, однако, не вызовет никаких недоразумений.

Далее, $J(P, Q) = J(P, R) + J(R, Q)$, так как мы можем выбрать контур, идущий от P к Q через R, и затем разбить интеграл в точке R (это часть (b) теоремы 1 из статьи (Manin, 1963, с. 1408)).

ОПЕРАТОРЫ ПИКАРА — ФУКСА КАК ГОМОМОРФИЗМЫ

Теперь мы выпишем фундаментальное соотношение, связывающее оператор J Пикара — Фукса со сложением точек нашего абелева многообразия:

$$J(P + R, Q + R) = J(P, Q)$$

Это следует из инвариантности дифференциалов первого рода при сдвигах нашего многообразия. Точное изложение дано в статье (Manin, 1963, с. 1413, лемма 12).

Мы определим теперь $J(P)$ как $J(O, P)$, где O — это нуль группового умножения на нашем абелевом многообразии. Это восстанавливает связь с обозначениями, которые мы использовали в примере ранее.

Лемма 2. *J — это гомоморфизм точек абелева многообразия (как аддитивной группы) в основное поле.*

Доказательство. Достаточно установить, что $J(P + Q) = J(P) + J(Q)$, так как $J(O) = 0$ и O — нуль аддитивной группы на абелевом многообразии.

$$J(P + Q) = J(0, P, Q) =$$
$$= J(O, P) + J(P, P + Q) = J(O, P) + J(O, Q) = J(P) + J(Q)$$

в силу предыдущего замечания

Следствие 3. *Для любого уравнения Пикара — Фукса J множество точек конечного порядка из абелева многообразия A лежит в ядре соответствующего оператора Пикара — Фукса J.*

Доказательство. Пользуемся тем, что основное поле не имеет кручения и гомоморфизмы отображают подгруппу кручения в подгруппу кручения.

Следствие 3 показывает, что точки конечного порядка из A лежат в пересечении ядер всех операторов Пикара — Фикса. Было бы замечательно получить и обратное включение, но это невозможно. Чтобы установить это, возьмем кривую C над $\mathbb{Q}$ с точкой P бесконечного порядка. Рассмотрим теперь C и P над $\mathbb{Q}(u)$. Тогда точка $P(u)$, разумеется, все еще имеет бесконечный порядок, но мы не можем сказать, что она не лежит в ядре оператора J, так как J должен переводить ее и точку O в одну и ту же величину, ибо обе зависят от u одинаково (а именно не зависят совсем). Однако

наше предположение почти истинно в том смысле, что может нарушаться лишь только что описанным способом.

Чтобы пояснить это точнее, нам нужны дополнительные обозначения. Мы берем их у Ленга из книги (Lang, 1959, с. 213), но достигаем некоторого упрощения, так как рассматриваем только случай характеристики 0. В этом абзаце K обозначает любое надполе $\bar{k}$ (хотя в приложениях K будет алгебраическим расширением $\bar{k}(u)$). Пусть A — многообразие над K. Пара (A', τ) называется *K/k-следом* многообразия A, если A' — абелево многообразие над k, и τ — гомоморфизм из A' в A, имеющий конечное ядро и такой, что для любого абелева многообразия B, определенного над k, и гомоморфизма $\beta\colon B \to A$, определенного над K, имеется гомоморфизм $\beta'\colon B \to A'$, определенный над k и такой, что $\tau\beta' = \beta$. Это определение может показаться (и действительно является) абстрактным, но оно определяет след как часть многообразия A, которая, по существу, не зависит от K/k (в нашем случае от u). Гомоморфизм τ играет в некотором смысле техническую роль: проблема в том, что в A' может быть определено несколько больше точек, чем нам хотелось бы, — они соответствуют точкам, определенным над алгебраическими расширениями поля K.

Теорема [1]) **4.** *Если P — точка многообразия A и $J(P) = 0$ для всех операторов Пикара — Фукса J на A, то имеется целое число n, такое, что точка nP (в смысле операции сложения на A) лежит в образе $k(u)/k$-следа A. Обратно, если такое n существует, то $J(P) = 0$ для всех операторов Пикара — Фукса J.*

Для нас это означает, что если $J(P) = 0$ для всех J, то P по существу не зависит от u. Разумеется, существует бесконечно много таких операторов Пикара — Фукса J, но нам нужно рассмотреть только базис пространства операторов Пикара — Фукса, а не все эти операторы.

Размерность этого базиса в случае $A = \mathrm{Jac}(C)$ равна роду алгебраической кривой C.

ДИВИЗОРЫ КОНЕЧНОГО ПОРЯДКА

Пусть теперь D — дивизор на алгебраической кривой C. Тогда D соответствует точке D' на $\mathrm{Jac}(C)$ и D рационально эквивалентен 0 тогда и только тогда, когда D' имеет конечный порядок на $\mathrm{Jac}(C)$. Если D' действительно зависит от u, то он, наверняка, имеет бесконечный порядок, а если D' не

[1]) (Manin, 1963, с. 1414, теорема 2). Я хотел бы поблагодарить профессора М. Ф. Зингера за неоценимую помощь в связи с этой теоремой.

зависит от u, то задача сводится к более простой. Сведéние происходит путем подстановки вместо u некоторого значения из k, так что задача ставится над k, а не над $k(u)$. Для u годится не любое значение — рассмотрите подстановку $u=0$ в $y^2=ux^3-1$. Выбор подходящих значений для u называется вопросом о «хорошей редукции» и рассматривается более подробно в гл. 8 (см. разд. «Критерии хорошей редукции» и особенно теорему 8), где он играет гораздо более важную роль в рассуждениях. Здесь достаточно сказать, что имеется лишь конечное число значений u, не дающих хорошей редукции (т. е. неподходящих), и имеется простой априорный тест[1]), позволяющий определить, дает ли данное значение u хорошую редукцию.

НАХОЖДЕНИЕ_ПОРЯДКА_ПО_МАНИНУ

Вход:

F(X, Y): уравнение алгебраической кривой.

В действительности не обязательно, чтобы оно было представлено через примитивный элемент, а не со многими переменными. F используется непосредственно только в вычислении дифференциалов первого рода.

D: дивизор на данной кривой, записанный в виде $\sum_{I=1}^{M} n_I P_I$

Мы будем иногда записывать P_I в виде (X_I, Y_I).

U: параметр, над которым определена кривая.

Выход:

БЕСКОНЕЧНЫЙ или целое число N в зависимости от того, какой порядок имеет D — бесконечный или конечный. Целое значение N указывает, что образ имел порядок N, так что нам следует рассмотреть ND, $2ND$, ..., при поиске порядка D.

[1] DIFF_1 :=

Линейно независимый базис для дифференциалов первого рода на кривой $F(X, Y)=0$.

Пусть G — длина списка $DIFF\_1$, т. е. род кривой $F(X, Y)=0$.

Этот шаг можно сделать с помощью простой модификации алгоритма Коутса. Подробности см. в гл. 3.

[2] Для каждого W из списка DIFF_1 делать:

[2.1] Пусть $A_{2G-1}, \ldots, A_0$ — неопределенные постоянные; положим

$$A_{2G}=1-\sum_{I=0}^{2G-1} A_I, \quad \text{так что сумма всех } A_I \text{ равна 1.}$$

(Это сделано потому, что операторы Пикара — Фукса

[1]) Значение u_0 из k для u дает хорошую редукцию, если $F(u_0, x, y)$ абсолютно неприводим над k и имеет тот же род, что и $F(u, x, y)$.

определены лишь с точностью до постоянных множителей.)

[2.2] Из уравнения $\sum A_i \mathrm{d}^i W/\mathrm{d}U^i = \mathrm{d}R(X, Y)/\mathrm{d}X$ находим неизвестные A_i и рациональную функцию $R(X, Y)$.

Знаменатель функции $R(X, Y)$ можно взять равным наименьшему общему кратному знаменателей в левой части. После освобождения от знаменателя наше уравнение распадается на несколько линейных уравнений относительно A_i и коэффициентов функции $R(X, Y)$, причем все они зависят от U, но не от X, Y. Далее, степень функции R не более чем на единицу выше максимальной степени, встречающейся в левой части уравнения.

Наш оператор Гаусса — Манина J, соответствующий дифференциалу первого рода W, имеет теперь вид $\sum A_i \mathrm{d}^i/\mathrm{d}U^i$, а $JW = \mathrm{d}R(X, Y)/\mathrm{d}X$.

Мы должны теперь вычислить $\dfrac{\mathrm{d}^i}{\mathrm{d}U^i}\int\limits_O^P W\,\mathrm{d}X$ для $1 \leqslant i \leqslant 2G$. Это выражение можно переписать в виде $\int\limits_O^P \mathrm{d}^i W/\mathrm{d}U^i + B_i(X, U)$, где B — вклад всех остальных членов от повторного дифференцирования.

[2.3] SUM := 0.

Здесь мы будем накапливать $J(D)$.

[2.4] Для $j = 1 \ldots M$ делать:

[2.4.1] $B_0 := 0$.

[2.4.2] Для $i = 1 \ldots 2G$ делать:

$$B_i := \frac{\mathrm{d}B_{i-1}}{\mathrm{d}U} + \frac{\mathrm{d}X_j}{\mathrm{d}U}\frac{\mathrm{d}^{i-1}W}{\mathrm{d}U^{i-1}}\Bigg|_P$$

(где $|_P$ означает значение, вычисленное для величин X, Y в точке P_j).

Первое слагаемое в этом выражении происходит от B-члена для предыдущего i, а второе возникает из соотношения

$$\frac{\mathrm{d}}{\mathrm{d}U}\int\limits_O^P f(x)\,\mathrm{d}X = \int\limits_O^P \frac{\mathrm{d}f(x)}{\mathrm{d}U}\,\mathrm{d}X + \frac{\mathrm{d}P}{\mathrm{d}U} f(P) \text{ (так как } \mathrm{d}O/\mathrm{d}U = 0).$$

Теперь $J(P)$ имеет вид $\sum A_i \dfrac{\mathrm{d}^i}{\mathrm{d}U^i}\int\limits_O^P W\,\mathrm{d}X$, или после перестановок,

$$\sum B_i(P, U) A_i + \int\limits_O^P A_i \frac{\mathrm{d}^i W}{\mathrm{d}U^i}\,\mathrm{d}x.$$

[2.4.3] SUM := SUM + $n_j \sum B_i A_i + R(P_j)$.

Добавка к SUM — это предыдущая формула, умноженная на n_j; она представляет собой вклад $n_j P_j$ в $J(D)$.

[2.5] Если SUM не нуль, то выдать результат БЕСКОНЕЧНЫЙ.

[3] Для U0 = 0, 1, 2, ...

Если ХОРОШАЯ_РЕДУКЦИЯ (U0, F, K).

То делать:

Этот метод выбора значения U не всегда самый лучший. Допустим, например, что уравнение F зависит от U и $\sqrt{U^2+1}$ и что 0 не дает хорошей редукции. Тогда выбор $U0 = 1$ даст нам кривую, определенную над $Q(\sqrt{2})$, в то время как выбор $U0 = 3$ даст кривую, определенную над $\mathbb{Q}$. Как ни привлекателен такой интуитивный выбор, трудно построить программу для нахождения «хороших» в этом смысле значений U.

[3.1] D := Результат подстановки U0 вместо U в D.

[3.2] F := Результат подстановки U0 вместо U в F.

[3.3] Выдать НАХОЖДЕНИЕ_ПОРЯДКА (F, D).

НАХОЖДЕНИЕ_ПОРЯДКА здесь означает одну из процедур НАХОЖДЕНИЕ_ПОРЯДКА_ПО_МАНИНУ (в случае если после подстановки U0 вместо U все еще остался трансцендентный параметр), КОНЕЧНЫЙ_ПОРЯДОК_ЭЛЛИПТИЧЕСКИЙ (см. гл. 7) или ГРАНИЦА_КРУЧЕНИЯ (см. гл. 8).

РЕАЛИЗАЦИЯ

Реализация этого алгоритма сталкивается лишь с немногими математическими проблемами, но оказывается довольно трудной в техническом отношении. Уже раньше при рассмотрении примера было отмечено, что одним из главных источников затруднений является наличие в одном и том же выражении дифференцирования по U и по X, а также необходимость различать частные производные по U (встречающиеся на шаге 2.2) и полные производные (в случае когда вместо (X, Y) подставлена точка P_j, которая вполне может зависеть от U). Я обнаружил, что для преодоления этих трудностей легче всего написать свои собственные специальные процедуры высокого уровня для дифференцирования, а не пытаться использовать процедуры системы REDUCE, ибо структуры данных для производных, которые имеются в этой системе, не способны различать полные и частные производные. Я все-таки применяю модифицированные версии некоторых процедур дифференцирования из системы REDUCE для выполнения более простых частей моей задачи; подробности этой модификации даны в приложении 1.

Другой важный источник трудностей — это необходимость эффективной реализации, при которой повторялось бы как можно меньше вычислений. Так как все встречающиеся выражения зависят от нескольких переменных (в них входит *параметр Гаусса — Манина* и переменная интегрирования), то даже «простые» операции вроде вычисления наибольшего общего делителя могут поглощать очень много времени. Поэтому требуются разнообразные вспомогательные таблицы, содержащие, например, $\mathrm{d}^i W/\mathrm{d}U^i$ или (частные) производные $\partial^i Y/\partial U^i$, которые приходится создавать и стирать по мере надобности. Имеются и другие соображения, связанные с эффективностью. Например, система линейных уравнений, возникающая на шаге [2.2], может быть частично разреженной [1]), и этим необходимо воспользоваться для получения реализации с разумной степенью эффективности. Несмотря на этот и другие специальные приемы, этот алгоритм все еще может быть очень дорогостоящим из-за размера преобразуемых выражений, в особенности на шаге [2.2]. При попытке выяснить, конечен ли порядок дивизора на кривой Тейта (приложение 1, пример 4), состоящего из точки $(d(d-1), d(d-1)^3)$ с кратностью 1 и точки на бесконечности с кратностью -1 (в действительности этот дивизор имеет порядок 7), тщательно запрограммированная экспериментальная реализация этого алгоритма заняла примерно 15 мин на машине IBM 370/168 в Исследовательском центре ИБМ им. Томаса Дж. Уотсона в Йорктаун-Хайтс [2]).

Хотя оператор Гаусса — Манина имеет в общем случае порядок $2G$ (где G — род кривой), есть много случаев, когда он вырождается и имеет меньшую степень. Например, хотя оператор Гаусса — Манина общей эллиптической кривой имеет степень 2, он имеет степень 1 в следующих специальных случаях (Manin, 1958, с. 755): $Y^2 = X^3 + aX$, $Y^2 = X^3 + a$, $Y^2 = X^3 + a^2X + ba^3$, где b — любая константа (т. е. не зависит от X, Y и a), отличная от 27/4 (в противном случае кривая перестает быть эллиптической). Кроме того, в этих слу-

---

[1]) Дополнительная информация о разреженности и пример того, как она может значительно упростить решение уравнений, приведены в приложении 2, пример 4. В гл. 9 имеется общее рассмотрение линейных уравнений.

[2]) К сожалению, не удалось получить более точной цифры из-за ошибки в CMS-Batch — симуляторе OS, установленном в Йорктаун-Хайтсе, которая привела к тому, что таймер системы ЛИСП не работал, и время центрального процессора пришлось вычислять, умножая общее время на «сервисный процент» (т. е. долю центрального процессора в общем времени), который давала операционная система. Однако эта цифра, вероятно, верна с точностью до 25 %. Машина IBM 370/168 примерно на 5 % быстрее, чем машина 370/165, установленная в Кембридже, на которой замерялись все остальные времена, упоминаемые в этой монографии

чаях операторы Пикара — Фукса особенно просты: они имеют вид

$\frac{XDa-2aDX}{2aY}$, $\frac{XDa-3aDX}{3aY}$, $\frac{XDa-aDX}{aY}$ соответственно.

Поэтому в интересах эффективности важно распознавать эти вырожденные случаи возможно раньше, а это нелегко.

Еще один частный случай возникает, когда X_I не зависит от U. В этом случае все B_i равны 0, и вычисление значительно упрощается. Этот случай часто возникает на практике, поэтому следует производить соответствующую проверку.

СПЕЦИАЛЬНЫЕ ЗНАЧЕНИЯ ПАРАМЕТРОВ

В этом разделе мы предполагаем, что наше подынтегральное выражение $f(x,u)\,dx$ зависит от u алгебраически. Это предположение в действительности не ограничивает общности, так как если подынтегральное выражение зависит от u трансценденто, то мы можем заменить трансцендентную функцию от u новым трансцендентным параметром u', не меняя задачи по существу. Ведь мы знаем, что если $f(x,u)$ интегрируема, то ее интеграл определен над *алгебраическим замыканием* исходного основного поля, т. е. никакие новые трансцендентности не вводятся.

Предложение 5. *Если $f(x,u)$ зависит от u алгебраически, то вычеты дифференциала $f(x,u)\,dx$ — алгебраические функции от u (так как они лежат в алгебраическом замыкании поля констант).*

Если алгоритм НАХОЖДЕНИЕ_ПОРЯДКА_ПО_МАНИНУ дает ответ БЕСКОНЕЧНЫЙ из-за того, что одно из значений SUM оказалось ненулевым, мы можем заключить, что интеграл невыразим в элементарном виде. По-видимому, имеет смысл рассматривать те специальные значения параметра u, для которых SUM оказывается равной нулю, и выяснять, элементарен ли интеграл в этом частном случае. Это приводит к следующему более общему вопросу:

«Для каких значений u интеграл $\int f(x,u)\,dx$ — элементарная функция?»

К сожалению, значениями u, для которых SUM обращается в нуль, не исчерпываются значения, которые могут сделать интеграл элементарным, так как имеется несколько других способов редукции задачи при подстановке вместо u специальных значений.

Рассмотрим теперь различные ситуации, в которых подстановка специального значения вместо параметра u может изменить природу проблемы интегрирования.

1) Может измениться род кривой. Это может произойти только для конечного числа значений u, и мы можем найти эти значения, рассмотрев канонический дивизор и возможности его вырождения.
2) Могут измениться плейсы, в которых имеются вычеты интегрируемого выражения. Их только конечное число, и мы можем найти их, рассмотрев все значения u, для которых совпадают множители числителя и знаменателя или множители знаменателя совпадают друг с другом (или, наконец, числитель или знаменатель меняют степень – для случая множителя x — бесконечность).
3) Может уменьшиться размерность пространства вычетов. Это — особенно коварный случай, и мы отложим его исчерпывающее обсуждение.
4) Дивизор может оказаться дивизором с кручением для конкретного значения u, но не в общем случае. Именно с этого случая мы начали это обсуждение, и такие значения (их конечное число) могут быть обнаружены просмотром всех корней (по u) функций SUM в алгоритме НАХОЖДЕНИЕ_ПОРЯДКА_ПО_МАНИНУ.
5) Может оказаться, что алгебраическая часть интегрируема для частного значения u, но не в общем случае. Эти значения можно обнаружить, рассматривая уравнения, порождаемые при работе алгоритма НАХОЖДЕНИЕ_АЛГЕБРАИЧЕСКОЙ_ЧАСТИ, и выясняя, когда вырождается уравнение, которое доказывает, что данная функция не интегрируема.

Таким образом мы показали, что в случаях 1, 2, 4 и 5 количество «исключительных» значений параметра u конечно (и эти значения могут быть эффективно вычислены).

Случай 3 существенно труднее. Например, мы можем иметь бесконечно много значений u, для которых уменьшается размерность Z-модуля вычетов.

Рассмотрим подынтегральное выражение, имеющее 4 вычета: 1, —1, u, $-u$; например,

$$\frac{1}{x\sqrt{x^2+1}}+\frac{1}{x\sqrt{x^2+u^2}}\,\mathrm{d}x.$$

Тогда для любого рационального значения u пространство вычетов становится одномерным, так что все эти случаи следует рассматривать как потенциально исключительные.

Лемма 6. *Пусть* Z*-модуль вычетов* $(r_1, \ldots, r_k)$ *выражения* $f(x, u)\,\mathrm{d}x$ *имеет размерность* k. *Допустим, что имеются значения* $u_1, \ldots, u_k$ *параметра* u, *такие что интеграл от* $f(x, u_i)\,\mathrm{d}x$ *имеет элементарную логарифмическую часть* (*но не подпадает под случаи* 1, 2, 4, *описанные выше*) *для* $1 \leqslant$

$\leqslant i \leqslant k$, *причем множество векторов* $\{(r_i(u_a)\,1 \leqslant i \leqslant k)\,1 \leqslant a \leqslant k\}$ *имеет размерность* k. *Тогда* $f(x, u)\,dx$ *имеет элементарную логарифмическую часть.*

Доказательство. Имеется целое число n, такое, что вектор $(n, 0, \ldots, 0)$ может быть выражен в виде линейной комбинации векторов вычетов $(r_i(u_a))$ с целыми коэффициентами. Тогда дивизор d_1, соответствующий вычету r_1, должен быть дивизором с кручением, так как он выражен в виде корня n-й степени из суммы дивизоров с кручением. Аналогичным образом и остальные дивизоры являются дивизорами с кручением; следовательно, должна существовать логарифмическая часть общего интеграла.

Теорема 7. *Если* $f(x, u)\,dx$ *не интегрируема в элементарных функциях, то имеется лишь конечное число значений* u_i *параметра* u, *для которых* $f(x, u_i)\,dx$ *интегрируема в элементарных функциях.*

Доказательство. Из случаев 1—5, описанных выше, в особом рассмотрении нуждается только случай 3, так как мы показали (и наши рассуждения можно сделать совершенно строгими), что имеется лишь конечное число значений, соответствующих случаям 1, 2, 4, 5. Допустим, что имеется бесконечно много значений, соответствующих случаю 3, но не подпадающих ни под один из пунктов 1, 2 или 4. Тогда по только что доказанной лемме 6 $\mathbf{Z}$-модуль, натянутый на векторы вычетов $(r_i(u_a))$, имеет размерность, меньшую, чем k, и потому может быть вложен в пространство размерности $k - 1$. Тогда имеется линейная зависимость между $r_1(u)$, ..., $r_k(u)$, которая неверна в общем случае, но верна для бесконечного множества частных значений u. Так как в силу предложения 5 векторы $r_i(u)$ алгебраичны по u, мы получаем алгебраическое выражение, неравное нулю тождественно, но имеющее бесконечно много корней, что и дает искомое противоречие.

Заметим, что эта теорема не вполне конструктивна, так как мы не указали никакого способа нахождения конечного множества значений в случае 3. Эта задача не вполне тривиальна, как показывает следующий пример.

Пусть E — эллиптическая кривая над $\mathbb{Q}$, причем на ней имеются точка бесконечного порядка и точка конечного порядка, обозначаемые соответственно через P и Q. (Иными словами, бесконечная часть группы Морделла — Вейля должна иметь ранг не меньше 1, а подгруппа кручения должна быть нетривиальной. Такие кривые существуют, как показывают таблицы из работ (Birch & Swinnerton-Dyer, 1963) и (Swinnerton-Dyer, 1974).) Пусть D — дивизор, линейно экви-

валентный $3P$, а D' — дивизор, линейно эквивалентный $5P - Q$. Пусть $f(x)$ — функция на нашей кривой с дивизором вычетов D, а $f'(x)$ — функция с дивизором вычетов D'. Рассмотрим тогда функцию $f(x) + uf'(x)$, пространство вычетов которой двумерно для иррациональных u и одномерно для рациональных u. Когда значение u иррационально, логарифмическая часть не может быть найдена, а когда u рационально (и равно, скажем, m/n), дивизор равен $n(3P) + m(5P - Q)$ и поэтому является дивизором с кручением, только если коэффициент при P равен нулю, т. е. для $u = -3/5$. Этот пример показывает необходимость ограничения $f(x, u)$ функциями, зависящими алгебраически от u, так как если бы мы рассмотрели $f(x) + \sin uf'(x)$, то было бы уже бесконечно много решений, а именно все корни функции $\arcsin(-3/5)$.

Глава 7

ЭЛЛИПТИЧЕСКИЕ ИНТЕГРАЛЫ. ОКОНЧАНИЕ

В предыдущей главе (включая алгоритм НАХОЖДЕНИЕ_ПОРЯДКА_ПО_МАНИНУ) полностью решена проблема дивизоров кручения над основными полями, содержащими трансцендентный параметр. У нас остался лишь случай, когда все элементы основного поля алгебраичны над рациональными числами. Именно эту задачу мы будем рассматривать в этой главе (для эллиптических кривых) и в следующей главе. Далее определение любой конкретной кривой или дивизора может содержать лишь конечное число алгебраических величин, поэтому мы можем ограничиться полями, которые порождены из поля рациональных чисел путем расширения конечным числом алгебраических величин, т. е. *полями алгебраических чисел*. Перед тем как исследовать проблему дивизоров с кручением над такими полями, мы должны узнать побольше об их структуре и возможных машинных представлениях. Именно это служит предметом рассмотрения в следующем разделе, дополняющем рассмотрение произвольных алгебраических выражений в гл. 2.

ПОЛЯ АЛГЕБРАИЧЕСКИХ ЧИСЕЛ

Теория полей алгебраических чисел чрезвычайно обширна, и нам будет нужна лишь небольшая ее часть. Обычно мы будем приводить лишь формулировки необходимых результатов, а доказательства и дальнейшие подробности читатель может найти в книгах (Borevich & Shafarevich, 1964) или (Weiss, 1963). Имеются некоторые поверхностные несогласованности между терминологией алгебраической геометрии и алгебраической теории чисел, однако из контекста всегда будет вполне очевидно, которое из возможных значений имеется в виду.

Произвольное поле алгебраических чисел можно записать в виде $\mathbb{Q}(l_1, \ldots, l_k)$, где l_i алгебраично над $\mathbb{Q}(l_1, \ldots, l_{i-1})$. Как уже сказано в гл. 2 (см. также (van der Waerden, 1949, т. I, с. 126—127)), мы можем представить его в виде $\mathbb{Q}(l)$, где l алгебраично над $\mathbb{Q}$. В действительности l — корень неприводимого многочлена с целыми коэффициентами (гл. 2),

и элементы $l_1, \ldots, l_k$ можно представить как многочлены из $\mathbf{Q}[l]$. Если K — поле алгебраических чисел, то мы обозначаем через K_Z множество: $\{a \in K \mid a$ удовлетворяет уравнению с коэффициентами в $\mathbf{Z}$ и старшим коэффициентом единица$\}$, т. е. множество *целых элементов поля* K. Мы отметим, что K_Z — подкольцо поля K (van der Waerden, 1949, т. II, с. 76) и потому содержит кольцо $\mathbf{Z}[l]$. Эти кольца не обязательно совпадают, как показывает случай $l^2 = 5$, когда $\frac{l-1}{2}$ не принадлежит $\mathbf{Z}[l]$, но удовлетворяет уравнению $x^2 + x - 1 = 0$. С нашей вычислительной точки зрения относительно трудно определить [1]), принадлежит ли кольцу K_Z элемент поля K; поэтому мы часто (когда это законно) будем заменять такую проверку соответствующим тестом на принадлежность кольцу $\mathbf{Z}[l]$. K_Z — область целостности, но не обязательно факториальное кольцо [2]). Идеалы кольца K_Z образуют полугруппу по умножению, и мы можем сопоставить каждому элементу $n \in K_Z$ идеал $\{n*a \mid a \in K_Z\}$, который будем обозначать через (n). Отображение $n \to (n)$ является полугрупповым гомоморфизмом из K_Z (как мультипликативной полугруппы) в мультипликативную полугруппу идеалов, причем его ядро — это множество единиц кольца K_Z (которое отображается на $(1) = K_Z$). Если A и B — идеалы K_Z, то мы будем обозначать через $A + B$ множество $\{a + b \mid a \in A,\ b \in B\}$. Это — тоже идеал кольца K_Z. Мы обозначаем $(m) + (n)$ через (m, n).

Теорема 1 (Weiss, 1963, с. 132). *Полугруппа идеалов K_Z обладает единственностью разложения в том смысле, что любой идеал B можно единственным образом (с точностью до порядка сомножителей) записать в виде $\prod P_i^{n_i}$ где n_i — положительные целые числа, P_i — простые идеалы кольца K_Z, т. е. не могут быть записаны в виде произведения нетривиальных идеалов.*

Пример. Мы можем теперь рассмотреть пример из подстрочного примечания о неединственности разложения в K_Z в свете этой теоремы. Пусть A — идеал $(3, 2 + l)$, а B — идеал $(3, 2 - l)$. Тогда $B^2 = (2 - l)$, $A^2 = (2 + l)$ и $A*B = (3)$. Следовательно, $(9) = A^2B^2$, и в действительности A и B — простые идеалы кольца K_Z (см. (Weiss, 1963, с. 134—135)).

Пусть p — *рациональное простое число* (т. е. простое в $\mathbf{Z}$ в обычном смысле слова). Тогда p можно разложить в про-

---

[1]) Алгоритм решения этой задачи изложен в разделе о реализации в конце настоящей главы.

[2]) Рассмотрим случай $l^2 = -5$. Здесь мы имеем $9 = 3*3 = (2 + l)*(2 - l)$ и ни один из элементов 3, $2 + l$, $2 - l$ не имеет собственных множителей (Weiss, 1963, с. 134).

изведение простых идеалов P_i кольца K_Z, и показатели этого разложения называются *индексами ветвления* простых идеалов. Мы говорим, что P_i *разветвленный*, если соответствующий показатель больше 1, и *неразветвленный*, если показатель в точности равен 1. Говорят, что число p *неразветвленное* (для расширения K поля $\mathbb{Q}$), если все показатели равны 1. Мы говорим, что идеалы P_i *лежат над* p. При этом простой идеал может лежать над не более чем одним рациональным простым числом (это видно из рассмотрения нормы и того, что два рациональных простых числа не могут иметь общих множителей).

Теорема 2 (Дедекинд. См. (Weiss, 1963, с. 157)). *Для любого l (т. е. для любого конечного расширения K поля $\mathbb{Q}$) имеется лишь конечное число разветвленных простых чисел.*

Если p' — простой идеал кольца K_Z, то мы можем построить факторкольцо K_Z/p', где факторизация производится по отношению $a \urcorner b$, которое имеет место тогда и только тогда, когда $a - b$ лежит в p'. Это эквивалентность, так что результат факторизации — действительно кольцо, называемое *кольцом классов вычетов*.

Теорема 3 (Borevich & Shafarevich, 1964, с. 208, теорема 2). *Кольцо классов вычетов K_Z/p' — конечное поле характеристики p.*

ЭЛЛИПТИЧЕСКИЕ КРИВЫЕ

Имеется весьма развитая теория точек кручения на эллиптических кривых, затронутая до некоторой степени в гл. 5 в рамках общего рассмотрения кручения. К сожалению, эта область остается очень трудной, а многие из ее результатов частичны или неконструктивны, несмотря на внимание, которое ей уделяли некоторые из величайших математиков. В качестве примера приведем следующее утверждение.

Теорема 4 (Manin, 1969)[1]). *Для любого поля алгебраических чисел K и простого числа p имеется целое число $n(K,p)$, такое, что для любой эллиптической кривой E, определенной над K, верно $|E_{\mathrm{tors}}(K)^p| \leqslant p^{n(K,p)}$, где $E_{\mathrm{tors}}(K)$ — группа всех точек кручения кривой E, определенных над K, а верхний индекс p обозначает p-часть этой группы, т. е. множество элементов, порядок которых является степенью p.*

К сожалению, мы знаем очень мало о числах $n(K,p)$, кроме случая $K = \mathbb{Q}$, который рассмотрен в гл. 5 в разд.

[1]) Имеется изложение этого вопроса в статье (Serre, 1971).

«Результат Мазура». Один из немногих результатов заключается в следующем (Kenku, 1979).

Теорема 5. *Если K — квадратичное числовое поле, то $n(K, 2) \leqslant 5$, где граница 5 неулучшаема.*

В действительности Кенку показывает также, что никакая точка на эллиптической кривой над квадратичным числовым полем не может иметь кручение 32.

Это показывает, какие трудности стоят на пути отыскания общих оценок кручения на эллиптических кривых. Гораздо больше можно сделать, если мы рассмотрим конкретную точку вместо того, чтобы искать границы для всевозможных точек кручения. Этим мы и займемся в следующем разделе.

ТЕОРИЯ ЛЮТЦА — НАГЕЛЯ

В этом разделе мы рассмотрим эллиптическую кривую $Y^2 = X^3 + aX + b$, где a и b — целые элементы поля алгебраических чисел $K = Q[l]$. Точка кривой называется *целой*, если и X, и Y являются целыми элементами поля K (хотя если X есть целый элемент поля K и Y лежит в K, то он должен быть целым элементом K).

Теорема 6. *На любой данной эллиптической кривой может лежать лишь конечное число целых точек.*

Доказательство. Этот результат был первоначаально доказан в статье (Siegel, 1929); см. также (Lang, 1960) и (Siegel, 1969). Однако это доказательство не дает никакой информации о числе целых точек или о их величине. Сейчас известно, что во многих случаях целые точки удовлетворяют неравенству (Baker, 1975, с.45) $|X|, |Y| < \exp((10^6 H)^{1\,000\,000})$ где H — максимум высот[1]) коэффициентов кривой.

Следствие 7. *Для фиксированного d конечно количество точек на данной эллиптической кривой, для которых dX — целая, т. е.* $\mathrm{den}(X) \mid d$.

Доказательство. Рассмотрим кривую $Y^2 = X^3 + d^4aX + d^6b$, точки которой — это в точности (d^2X, d^3Y), где (X, Y) — точка исходной кривой.

В действительности граница нужна нам лишь в той степени, в какой она затрагивает время работы алгоритма, а

[1]) *Высота* алгебраического числа определяется как максимум абсолютных величин коэффициентов его минимального многочлена. Поэтому высота рационального числа — это его абсолютная величина.

оценки Бейкера внушают порой неоправданный[1]) пессимизм.

Теорема 8. *Пусть* (X, Y) *— точка, имеющая порядок в точности* n *на эллиптической кривой* $Y^2 = X^3 + aX^2 + bX + c$, *определенной над полем алгебраических чисел* K.

(1) *Если* n *не является степенью рационального простого числа, то* X *— целое алгебраическое число.*

(2) *Если* n *— степень рационального простого числа* p, *то* pX *— целое алгебраическое число.*

Доказательство. Эта теорема почти совпадает с теоремой 2.1 книги (Lang, 1978) плюс замечание, сопровождающее теорему 2.2. Единственное отличие имеется в случае (2), где Ленг утверждает, что знаменатель координаты X является делителем произведения степеней $P_i^{r(i)}$, где P_i — простые идеалы кольца K_Z, делящие p, а $r(i)$ — рациональные целые числа, не превосходящие $e(i)/2$, где $e(i)$ — индекс ветвления идеала P_i как делителя числа p. Однако это произведение, наверняка, является делителем $\prod P_i^{e_i} = p$, так что теорема доказана.

Отметим, что для нетривиальности случая (2) необходимо, чтобы простое число p имело индекс ветвления не меньше 2 (иначе $r(i)$ были бы равны 0), а это случается очень редко. (В действительности при $a = 0$ мы можем заменить 2 на 4, а эта степень ветвления чрезвычайно необычна.) Заметим также, что мы можем рассматривать уравнения кривых, чуть более общие, чем каноническая форма Вейерштрасса. Такой вид обобщенной теоремы Лютца — Нагеля — Касселса рассмотрен в статье (Zimmer et al., 1979).

Следствие 9 (теорема Лютца — Нагеля)[2]). *Любая точка кручения эллиптической кривой* $Y^2 = X^3 + aX + b$, *определенной над рациональными числами, с целыми* a, b *должна иметь целые коэффициенты.*

---

[1]) В качестве примера рассмотрим эллиптическую кривую $20A$ (в терминологии статьи (Swinnerton-Dyer, 1975), таблица 1). Эту кривую можно представить уравнением $Y^2 = X^3 - 108X + 297$, и в силу неравенства Бейкера на ней нет целых точек (а потому и рациональных точек целого порядка) с $|X|$ или $|Y|$ большим, чем $E(e, 10, 10, 6.92)$, где обозначения $E(a, b) = a^b$ и $E(a, b, c, \ldots) = a^{E(b, c, d, \ldots)}$ служат для записи многоэтажных экспонент. В действительности, однако, вычисления Дэвенпорта и Стефенса (частично использующие алгоритм из статьи (Birch & Swinnerton-Dyer, 1963) показывают, что самая большая целая точка на этой кривой — это $X = 12$, $Y = \pm 27$, и эта точка в действительности имеет конечный порядок. Насколько мы можем судить, это «типичный» пример подлинной величины целых точек конечного порядка.

[2]) Этот результат был сформулирован и в гл. 5 как часть специальной методики для эллиптических кривых над рациональными числами, но в математическом отношении он принадлежит настоящей главе.

ЛЮТЦ_НАГЕЛЬ [1])

Вход:

F(X, Y): Эллиптическая кривая вида $Y^2 = X^3 + aX + b$, определенная над некоторым полем алгебраических чисел, где a, b — целые элементы этого поля.

ТОЧКА: Целая точка (X, Y) на этой кривой, также определенная над некоторым полем алгебраических чисел.

Упомянутые два поля алгебраических чисел не обязаны совпадать, так как нас в действительности не интересует, что представляют собой эти алгебраические поля, если только мы можем вычислять над ними.

Выход:

БЕСКОНЕЧНЫЙ, если точка имеет бесконечный порядок, N, если порядок точки равен N.

[1] Объявляем массивы X СТАРЫЙ [0: бесконечность], Y СТАРЫЙ [0: бесконечность].

В действительности их, вероятно, лучше всего реализовать в виде списков.

[2] XСТАРЫЙ[0] := X
YСТАРЫЙ[0] := Y
N := 0.

[3.1] Если знаменатель (X) не равен 1, то перейти к [4.1].

[3.2] Для M := 0 : N — 1 делать
Если X = XСТАРЫЙ[M], то делать:

[3.2.1] Если Y = YСТАРЫЙ[M], то N := $(2^N) - 2^M$
иначе, если Y = — YСТАРЫЙ[M], то N := $(2^N) + 2^M$,
иначе ОШИБКА.

Теперь мы знаем, что ТОЧКА имеет порядок, который делит $2^N \pm 2^M$, но мы хотели бы иметь гораздо меньшее число (предпочтительно настоящий порядок) перед тем, как будет вызвана процедура ДИВИЗОР_В_ФУНКЦИЮ. Мы можем также утверждать, что множитель 2^M необходим, так как иначе мы нашли бы нуль для меньших M и N.

[3.2.2] Для каждого множителя L числа N (в порядке возрастания) делать:
если ТОЧКА**$(L*2^M) = 0$, то выдать ответ $L*2^M$.

[3.2.3] ОШИБКА

[3.3] (X, Y) := 2*(X, Y) в смысле сложения точек на кривой.

[3.4] N := N + 1
XСТАРЫЙ[N] := X

---

[1]) Этот алгоритм был подсказан одной процедурой из работы (Stephens, 1970). Я очень благодарен д-ру Стефенсу за многочисленные обсуждения вопроса о вычислениях на эллиптических кривых над рациональными числами.

YСТАРЫЙ[N] := Y.

[3.5] Если X = бесконечность,
то выдать ответ 2^N.

[3.6] Перейти к [3.1].

Мы можем пройти по этому циклу лишь конечное число раз, так как мы знаем, что на кривой есть лишь конечное число целых точек.

[4.1] P := знаменатель (X).

Как сказано в гл. 2, наша структура данных для алгебраических величин помещает все алгебраическое поведение в числитель, а знаменатель остается свободным, так что P будет целым рациональным числом.

[4.2] Если P — не простое,

[4.3] X := XСТАРЫЙ[0]
Y := YСТАРЫЙ[0]
N := 1.

Мы знаем, что если ТОЧКА вообще имеет конечный порядок, то этот порядок должен быть степенью числа P, так что мы сразу рассматриваем выражения этого вида.

[4.4] (X, Y) := P*(X, Y)
в смысле сложения точек на кривой
N := P*N.

[4.5] Если X = БЕСКОНЕЧНЫЙ,
то выдать ответ N.

X = Y = БЕСКОНЕЧНЫЙ — это нуль нашей группы.

[4.6] Если знаменатель (X) | P,
то перейти к [4.4].

[4.7] Выдать ответ БЕСКОНЕЧНЫЙ.

ТОЧКА ЗРЕНИЯ ХАРДИ

Простейший нетривиальный случай общей задачи интегрирования алгебраических функций в конечном виде — это интегрирование функций, определенных на кривых рода 1. Эту задачу долго считали неразрешимой (см. начало гл. 4) и Харди сформулировал классическую точку зрения следующим образом:

> «Не было придумано метода, с помощью которого мы всегда могли бы определить за конечное число шагов, является ли данный эллиптический интеграл псевдоэллиптическим, и проинтегрировать его в этом случае; есть основания думать, что такого метода и нельзя дать» (Hardy, 1916, с. 47—48).

Заметим, что предшествующие исследования по эллиптическим интегралам (Ng, 1974) и (Carlson, 1965) не решили эту задачу полностью, так как они требуют, чтобы подынте-

гральное выражение уже имело явно эллиптический вид (т. е. было произведением рациональной функции от x на квадратный корень из кубики или квартики от x). Приведение к этому виду делает наш алгоритм ФОРМА_ВЕЙЕРШТРАССА, интенсивно использующий алгоритм Коутса.

Алгоритм и методика, которые мы описали к этому моменту, позволяют нам решить эту задачу, т. е. определить, является ли данный эллиптический интеграл псевдоэллиптическим, и взять этот интеграл. Перед тем как рассмотреть (в следующей главе) кривые произвольного рода, мы поясним, как соотносятся эти алгоритмы, поскольку алгоритмы для кривых рода 1 сильно специализированы применительно к этому случаю, и все они были реализованы в отличие от некоторых из алгоритмов для кривых более высокого рода, которые будут описаны ниже.

В гл. 4 задача интегрирования сведена к задаче определения для данного дивизора, обладает ли он кручением, и нахождения порядка этого дивизора в случае положительного ответа. Поэтому достаточно описать процедуру, которая будет решать эту задачу для эллиптических кривых над произвольными основными полями (характеристики 0).

КОНЕЧНЫЙ_ПОРЯДОК_ЭЛЛИПТИЧЕСКИЙ

Вход:

F(X, Y): Уравнение кривой рода 1.
Оно может быть задано как представлением с несколькими переменными, так и через примитивный элемент, так как его можно привести к канонической форме, когда это потребуется.

D: Дивизор на заданной кривой.
Мы считаем, что этот дивизор неглавный (т. е. не порядка 1), так как мы проверяем это прежде, чем начинаем применять процедуру НАХОЖДЕНИЕ_ПОРЯДКА.

Выход:

БЕСКОНЕЧНЫЙ, если рассматриваемый дивизор не обладает кручением;
порядок дивизора в противном случае.

[1] Если поле констант кривой F содержит трансцендентный параметр U, то делать:

[1.1] Если НАХОЖДЕНИЕ_ПОРЯДКА_ПО_МАНИНУ (F, D, U) = БЕСКОНЕЧНЫЙ,
то выдать ответ БЕСКОНЕЧНЫЙ.

[1.2] Для I = 2, 3, ... делать:
Если ДИВИЗОР_ФУНКЦИИ (F, I*D) выдает функцию, то выдать ответ I.
Здесь мы можем применить тест Кэли (см. гл. 5).

[2] Если F(X, Y) и D не содержат алгебраических чисел, то делать:
[2.1] Для I = 2, 3, 4, 5, 6, 7, 8, 9, 10, 12 делать:
Если ДИВИЗОР_В_ФУНКЦИЮ (I * D) выдает функцию, то выдать ответ I.
[2.2] Выдать ответ БЕСКОНЕЧНЫЙ.
[3] Если F не находится в каноническом виде с целыми алгебраическими коэффициентами,
то делать:
[3.1] F′D′ = ФОРМА_ВЕЙЕРШТРАССА (F, D).
[3.2] Запишем F′(X, Y) в виде $Y^2 = X^3 + aX + b$.
[3.3] Если не оба числа a, b целые,
то делать:
[3.3.1] D := н. о. к. (знаменатель a, знаменатель b).
[3.3.2] b := b*D^6.
[3.3.3] a := a*D^4.
[3.3.4] Для каждого (X, Y) из D′ делать:
$X := X * D^2$;
$Y := Y * D^3$.
[4] ТОЧКА := нулевая точка кривой F′, т. е. (БЕСКОНЕЧНОСТЬ, БЕСКОНЕЧНОСТЬ).
[5] Для каждого P^N из D′ делать:

ТОЧКА := ТОЧКА + $N*P$.

Эти действия производятся как сложение точек на кривой F' в канонической форме Вейерштрасса согласно теории, изложенной в гл. 5.
[6] Если Знаменатель (ТОЧКА) не равен [1]) 1, то делать:
[6.1] A := A * Знаменатель^4.
B := B * Знаменатель^6.
Мы умножили уравнение F', имеющее по предположению вид $Y^2 = X^3 + A*X + B$, на Знаменатель^6, и должны теперь соответствующим образом преобразовать знаменатель элемента ТОЧКА.
[6.2] Y := Y * Знаменатель^3.
X := X * Знаменатель^2.
[7] Выдать ответ ЛЮТЦ_НАГЕЛЬ(F′, ТОЧКА).

РЕАЛИЗАЦИЯ

Алгоритмы, описанные в этой главе, довольно легко реализуются, если уже реализованы алгоритмы для полей алгебраических чисел. Основная проблема при работе на эллиптических кривых состоит в том, что нам нужно выражение

[1]) Этот знаменатель вычисляется как наименьшее общее кратное знаменателей X и Y. В действительности можно действовать лучше — см. ниже раздел о реализации.

для нулевой точки кривой, для которой $x = y =$ бесконечность. Наше решение этой проблемы предусматривает введение специального ядра [1]) «бесконечность» и правильную работу с ним процедур для сложения точек на алгебраических кривых. Этого нетрудно добиться, так как верны следующие утверждения.

Предложение 10. *Бесконечность может войти в представление точек на эллиптической кривой, приведенной к каноническому виду, только через представление бесконечно удаленной точки.*

Предложение 11. *Бесконечно удаленная точка есть нуль группового закона на кривой.*

Предложение 12. *Сумма двух точек на кривой может быть точкой, содержащей бесконечность, только в том случае, если они имеют одну и ту же X-координату, а их Y-координаты в сумме дают* 0.

Описанные выше алгоритмы неоптимальны в нескольких маловажных отношениях. Основная причина этого в том, что при избавлении от знаменателей мы нередко умножали на некоторую степень знаменателя, хотя зачастую хватило бы меньшего выражения. Например, превращая точку (X, Y) в целую, мы умножали все на Знаменатель (ТОЧКА)^6, хотя в действительности нам была нужна просто наименьшая 6-я степень, большая чем Знаменатель $(X)^2$ и Знаменатель $(Y)^3$.

Основные трудности заключены в используемых алгоритмах для полей алгебраических чисел и целых алгебраических чисел. Как уже говорилось, не всегда просто определить, является ли данное алгебраическое выражение целым алгебраическим числом. Мы приводим здесь алгоритм, который не только решает эту задачу, но и определяет в общем случае знаменатель алгебраического числа.

ЗНАМЕНАТЕЛЬ_АЛГЕБРАИЧЕСКОГО

Вход:
X: алгебраическое число
Выход:
наименьшее целое число, такое, что $N * X$ — алгебраическое целое.
[1] Если Знаменатель $(X) = 1$
то выдать ответ 1.

---

[1]) В смысле системы REDUCE-2. Подробности см. в приложении 1.

Это — просто вспомогательная проверка, при которой знаменатель числа X рассматривается как чисто формальное выражение, причем алгебраическим выражениям, входящим в него, не приписывается никакого смысла. Она срабатывает, так как мы знаем, что все алгебраические выражения, порождаемые системой REDUCE (согласно стратегии, описанной в гл. 2), оказываются целыми алгебраическими.

[2] МАТРИЦА[0] := (1, 0, ..., 0)

МАТРИЦА будет списком строк матрицы, причем строка$_i$ ($i = 0, 1, \ldots$) будет состоять из коэффициентов всех алгебраических выражений из X^i.

[3] K := 0.

[4] Пока МАТРИЦА особенная, делать:

[4.1] K := K + 1.

[4.2] МАТРИЦА [K] := Коэффициенты алгебраических выражений из X^K.

Эта операция собирания коэффициентов всех алгебраических величин из некоторого выражения всегда должна давать вектор, состоящий из рациональных чисел, расположенных в одном и том же порядке, где первый член — неалгебраическая часть.

[5] Пусть V[0] ... V[K] таковы, что V*МАТРИЦА = 0 и V[K] = 1.

Тогда минимальное уравнение с рациональными коэффициентами и старшим коэффициентом 1, которому удовлетворяет X, — это $\sum_{i=0}^{i=k} V[i] X^i = 0$.

[6] N := 1.

[7] Для I := 0, 1, ..., K — 1 делать:

[7.1] Для любого простого делителя P числа $V[I]$ делать:

[7.1.1] Для J := 0, 1, ..., K делать:

V[J] := V[J]*P^{K-J}.

В результате этого X заменяется на X*P и V[J] ренормализуется, чтобы сохранить единичный старший коэффициент уравнения.

[7.1.2] N := N*P.

[7.1.3] Если P — делитель V[J],
то перейти к [7.1.1].

[8] Выдать ответ N.

Глава 8

КРИВЫЕ НАД ПОЛЯМИ АЛГЕБРАИЧЕСКИХ ЧИСЕЛ

ПРОИЗВОЛЬНЫЙ РОД

Случай кривых произвольного рода намного более труден, чем случай кривых рода 1, и для этого случая нет хорошо разработанных алгоритмов. Я не смог написать ни одной сколько-нибудь значительной программы для этого случая, так как у меня не было программ для большого числа вспомогательных алгоритмов, хотя такие программы уже были написаны в других местах или их легко было бы написать. Поэтому здесь изложены наброски методов, с помощью которых можно было бы найти границы кручения кривых произвольного рода над полями алгебраических чисел.

Дело осложняется тем, что бóльшая часть необходимого математического аппарата сформулирована на очень абстрактном языке, а многое содержится лишь в «фольклоре» алгебраической геометрии. Поэтому приводимые результаты будут менее подробными, чем в остальной части монографии, а ссылки на литературу не всегда полными. Описываемая ниже теория справедлива для кривых рода 1 точно так же, как и для кривых более высокого рода. В действительности она лежит в основе многих теоретических положений предыдущей главы. Поэтому мы сможем проиллюстрировать [1]) бóльшую часть этих методов на примерах [2]) кривых рода 1, что и будем обычно делать для простоты, хотя основные приложения этих исследований будут к кривым рода > 1.

Основная идея аналогична многим приемам, часто применяемым в алгебраической геометрии: мы сводим задачу к случаю конечных полей, где можем вычислять непосредственно и делать полный перебор, а затем комбинируем информацию, полученную для нескольких конечных полей, чтобы решить исходную задачу. Говоря языком компьютерной алгебры, мы применяем *модулярный* подход.

---

[1]) Превосходная иллюстрация работы этих методов имеется в статье (Mazur & Swinnerton-Dyer, 1974, с. 21, лемма 1).

[2]) Кроме того, имеется намного больше вычислений и таблиц, относящихся к эллиптическим кривым (например (Swinnerton-Dyer, 1974)), так что для них легче найти подходящие примеры и объяснить поведение.

ХОРОШАЯ РЕДУКЦИЯ

Пусть K — общее поле определения кривой C и дивизора D, так что D соответствует элементу якобиева многообразия кривой C, определенному над K. Пусть p' — простой идеал поля K, лежащий над рациональным простым числом p. Пусть K' — *поле классов вычетов* поля $K \bmod p'$, т. е. поле, порожденное целыми числами из K по модулю идеала p'. Пусть A — абелево многообразие над K (обычно мы считаем его якобиевым многообразием кривой C, но общая теория не требует этого). Пусть A' — многообразие над K', определяемое теми же уравнениями, которые определяют A над K (т. е. A' есть *специализация* A в смысле книги (Mumford, 1965)). Заметим, что A' определено над конечным полем и потому может иметь лишь конечное число элементов, и все они должны обладать кручением.

Если p — любое рациональное простое число и G — абелева группа (обычно это будет якобиева группа дивизоров, но это не обязательно), то определим *p-часть* группы G как множество тех ее элементов, порядок которых является степенью p. Это очевидным образом подгруппа группы G, и ее порядок есть степень p. Определим *не-p-часть* группы G как множество тех ее элементов, (конечный) порядок которых взаимно прост с p. Это — тоже подгруппа G. Если любой элемент группы G имеет конечный порядок, то для любого простого числа p она равна прямому произведению своей p-части и не-p-части. Для нужных нам преобразований с участием p-частей потребуется следующий алгоритм.

МАКСИМАЛЬНАЯ_СТЕПЕНЬ

Вход:
P: положительное целое число, часто простое.
N: положительное целое число.
Выход:
Q: наибольшая целая степень числа P, не превосходящая N.

Мы не будем утруждать себя подробным описанием столь простого алгоритма.

Следуя работе (Serre & Tate, 1968), мы скажем, что A *обладает хорошей редукцией*[1]) в p', если $A = A' \times K$, где операция «$\times$» обозначает взятие тензорных произведений над кольцом нормирования идеала p'. Следующий результат, по существу, хорошо известен.

[1]) Грубо говоря, это условие означает, что A можно восстановить, зная A' и K. Это понятие аналогично понятиям хорошей оценки или счастливого простого числа (см., например, Yun, 1973).

Теорема 1. *Если A обладает хорошей редукцией в идеале p′, который лежит над p, то не-p-часть подгруппы кручения многообразия A инъективно вкладывается в не-p-часть подгруппы кручения многообразия A′.*

Следствие 2. *Размер не-p-части группы кручения многообразия A делит размер многообразия A′ (рассматриваемого как группа).*

Мы можем также сформулировать следующий результат (из статьи (Serre & Tate, 1968) или (Shimura & Taniyama, 1961)) применительно к нашей ситуации и обозначениям.

Теорема 3. *Для фиксированных K и A имеется лишь конечное множество простых чисел, обладающих плохой (т. е. не обладающих хорошей) редукцией.*

Из теоремы 2 гл. 7 следует, что это утверждение остается справедливым, если мы ограничимся неразветвленными простыми числами, что мы и делаем. Это означает, что мы можем использовать следующий алгоритм для редукции общей задачи к задаче определения границ кручения с точностью до простых идеалов.

ГРАНИЦА_КРУЧЕНИЯ

(Вариант 1)

Вход:
K: поле алгебраических чисел.
F(X, Y): уравнение кривой, определенной над K.
Выход:
N: граница кручения кривой F над K.
[1] Для $P = 2, 3, 5, \ldots$ делать:
Для любого простого идеала P′ поля K, такого, что P′|P, делать:
если ХОРОШАЯ_РЕДУКЦИЯ (F, K, P'),
то делать:
[1.1] НЕ_P_ЧАСТЬ := КОНЕЧНАЯ_ГРАНИЦА (F, K, P′).
Алгоритм КОНЕЧНАЯ_ГРАНИЦА должен давать границу кручения по модулю простого идеала P′.
См. следующий раздел настоящей главы.
[1.2] Перейти к [2].
[2] Для Q = простое, большее P, ..., делать:
Для каждого простого идеала Q′ поля K, такого, что Q′|Q, делать:
если ХОРОШАЯ_РЕДУКЦИЯ (F, K, Q′),
то делать:
[2.1] НЕ_Q_ЧАСТЬ := КОНЕЧНАЯ_ГРАНИЦА (F, K, Q′).
[2.2] Перейти к [3].

[3] ОТВЕТ1 := НЕ_Q_ЧАСТЬ*МАКСИМАЛЬНАЯ_СТЕПЕНЬ (Q, НЕ_P_ЧАСТЬ).
ОТВЕТ2 := НЕ_P_ЧАСТЬ*МАКСИМАЛЬНАЯ_СТЕПЕНЬ (P, НЕ_Q_ЧАСТЬ).
Выдать ответ minimum (ОТВЕТ1, ОТВЕТ2).

ОТВЕТ1 разбивает группу кручения на ее не-q-часть и ее q-часть, в то время как ОТВЕТ2 производит «двойственное» разбиение.

Приведенный алгоритм — это не единственный способ использования разработанного нами механизма нахождения границы кручения эллиптической кривой. Можно, например, взять 3 простых числа с хорошей редукцией, а не 2, а затем заметить, что p-кручение для любого p появится в не-q-частях для 2 простых чисел q, так что кручение ограничено корнем квадратным из произведения трех не-q-кручений.

Позднее в этой главе мы увидим, что можно действовать значительно лучше, если мы точно знаем, чему равно кручение над конечным полем.

КРУЧЕНИЕ НАД КОНЕЧНЫМИ ПОЛЯМИ

Цель этого раздела — описание различных методов нахождения (или хотя бы оценки) размера якобиевой группы дивизоров над конечным полем, т. е. реализации алгоритма КОНЕЧНАЯ_ГРАНИЦА. Бóльшая часть этого материала заимствована из книги (Lang, 1959, гл. V, особенно разд. 3)[1]. Нам понадобится одно обозначение. Если A — абелево многообразие, определенное над полем K, то $|A|_K$ обозначает число точек многообразия A, определенных над K. Наше первое замечание состоит в том, что если C — кривая рода g, то $\mathrm{Jac}(C)$ — абелево многообразие размерности g (см. обсуждение в конце гл. 5). Следовательно, мы хотим найти $|\mathrm{Jac}(C)|_K$ для исследуемой кривой C. К сожалению, это число плохо связано с $|C|_K$. Однако верно следующее утверждение, доказательство которого дано в книге (Lang, 1959, с. 139—140).

Лемма 4. *Пусть A — абелево многообразие размерности r над конечным полем K с q элементами. Пусть φ: $A \to A$ — эндоморфизм Фробениуса, индуцированный некоторым эндоморфизмом Фробениуса*[2] *на K. Тогда* $|A|_K = \prod_{j=1}^{j=2r} (w_j - 1)$, *где w_j — характеристические корни φ.*

---

[1]) Я благодарен проф. Суиннертон-Дайеру, который привлек мое внимание к этой работе и исправил многие ошибки, сделанные мной при ее изучении.

[2]) Эндоморфизм Фробениуса определяется соотношением $x \to x^q$. Дальнейшие подробности см. в книге (Eichler, 1966, с. 249).

Лемма 5 (Lang, 1958, с. 138, лемма 2 и гл. IV, разд. 3). *В обозначениях леммы 4 имеем* $|w_l| = q^{1/2}$.

Теорема 6. $|\mathrm{Jac}(C)|_K \leqslant (q^{1/2}+1)^{2g}$.

Доказательство получается ссылкой на леммы 4 и 5.

КРИТЕРИЙ ХОРОШЕЙ РЕДУКЦИИ

Теорема 7. *Если C обладает хорошей редукцией, то и* $\mathrm{Jac}(C)$ *— тоже.*

Обратное утверждение не всегда верно, но, по-видимому, не стоит и пытаться получить выгоду, используя те простые числа, для которых $\mathrm{Jac}(C)$ обладает хорошей редукцией, а C не обладает. Я сумел избежать явного построения якобиевых многообразий в процессе программирования, и мне кажется, что возникающие осложнения перебесили бы полученный небольшой выигрыш.

Очевидно, что хорошей редукции не получится, если род кривой C' отличен от рода C, так как тогда якобиевы многообразия имели бы различные размерности. В действительности следующий критерий необходим и достаточен для наличия хорошей редукции у кривой C, а потому и у $\mathrm{Jac}(C)$.

Теорема 8. *Если C и C' имеют одинаковый род, а F' абсолютно неприводима (т. е. неприводима во всех алгебраических расширениях поля K'), то мы имеем хорошую редукцию.*

Рассмотрим отдельно две половины этого теста, начав с условия сохранения рода. Род кривой C' не может стать больше, чем род кривой C, поэтому мы должны выявить случаи, в которых род понижается. Это может произойти одним из двух способов: некоторый дифференциал первого рода на C может перестать быть дифференциалом первого рода на C' или пространство дифференциалов первого рода может «сжаться». В качестве примера первой ситуации рассмотрим кривую $Y^2 = (X-1)(X+1)(X+2)$, которая имеет род 1 над $\mathbb{Q}$ с единственным дифференциалом первого рода, а именно dX/Y. Однако по модулю 3 эта функция имеет полюс в точке $X = 1 = -2$, так как $X-1$ является там локальным параметром и $1/Y$ ведет себя как $1/(X-1)\sqrt{X+1}$. В действительности эта кривая, разумеется, имеет род 0 по модулю 3. Так как мы знаем дифференциалы первого рода в результате вычисления рода кривой, этот случай легко проверяется. Вторая возможность заключается в том, что дифференциалы первого рода перестают быть независимыми. В качестве примера рассмотрим кривую в пространстве, заданную уравнениями $Y^2 = X^3 - 1$ и $Z^2 = X^3 + 2$. Здесь dX/Y

и dX/Z — независимые дифференциалы первого рода над рациональными числами, но по модулю 3 они уже зависимы, как и следовало ожидать.

Отметим еще возможность того, что мы отвергнем некоторые простые числа в качестве кандидатов, могущих дать хорошую редукцию (хотя в действительности они дают ее), из-за того, что нечаянно получили выражение для дифференциала, делящееся на рассматриваемое простое число. Особенно вероятно, что это произойдет над полями алгебраических чисел, когда мы имеем дело с простыми идеалами вместо настоящих простых чисел, так как в этом случае мы не можем устранить этот простой множитель путем деления. Однако это может произойти лишь конечное число раз, так что у нас все еще остается бесконечно много простых чисел, дающих хорошую редукцию. Это соображение может оказаться обескураживающим с вычислительной точки зрения, но оно не влияет на теорию.

ПРИМЕР

Мы рассмотрим кривую Тейта из примера 4 приложения 2 при $d=2$ и будем работать над рациональными числами. Это не обязательно, так как хватило бы границы Мазура или подхода Лютца — Нагеля, но на этом примере относительно легко объяснить работу алгоритмов. Уравнение имеет вид $y^2 = 4x^3 - 15x^2 + 8x + 16$. Очевидно, что оно не обладает хорошей редукцией mod 2, так как сводится к уравнению $y^2=x^2$, которое разложимо, и сверх того дифференциал первого рода $1/y$ не остается дифференциалом первого рода.

По модулю 3 это уравнение сводится к уравнению $y^2 = x^3 + 2x + 1$, которое неприводимо и сохраняет дифференциалы первого рода. Поэтому не-3-часть кручения не превосходит $(3^{1/2}+1)^2 = 7$ (после округления с недостатком до целого числа). Рассматриваемая кривая обладает хорошей редукцией и по модулю 5, так что не-5-часть не превосходит $(5^{1/2}+1)^2 = 10$ (после округления с недостатком до целого числа). Поэтому эта кривая имеет кручение, не превосходящее [1]) minimum $(7*9, 10*5) = 50$.

---

[1]) В действительности можно получить лучшую оценку, если рассмотреть различные случаи отдельно.

Случай 1. Кривая имеет 7-кручение. В этом случае группа кручения должна иметь ровно 7 элементов (что и верно на самом деле).

Случай 2. Кривая имеет 3-кручение и 5-кручение. В этом случае 3-кручение ограничено числом 9, а 5-кручение — числом 5, так что кручение в целом ограничено числом 45 (так как введение 2-кручения понижает итог). В действительности это можно исключить, если рассмотреть редукцию по модулю 11, что дает 18 (а значит, и 15, если убедиться в том, что структура группы сохраняется) в качестве границы для не-11-части, а значит, и для кручения в целом.

УЛУЧШЕННЫЙ АЛГОРИТМ

В этом разделе мы опишем метод использования точного количества точек на якобиевом многообразии данной кривой над нашим конечным полем. Аналогично тому, как в предыдущем случае нам нужен был для извлечения p-частей тривиальный алгоритм МАКСИМАЛЬНАЯ_СТЕПЕНЬ, здесь мы имеем для той же цели алгоритм ВХОЖДЕНИЯ.

ВХОЖДЕНИЯ

Вход:

P: положительное целое число, часто простое.

N: положительное целое число.

Выход:

Q: наибольшая целая степень числа P, на которую делится N.

Мы не будем утруждать себя подробным описанием столь простого алгоритма.

Нахождение количества точек на якобиевом многообразии кривой — нелегкая задача, и я не могу предложить алгоритм для ее решения. В случае кривой рода 1 она сама и является якобиевым многообразием, так что проблем не возникает (кроме необходимости проявлять осторожность при подсчете кратных точек и при различении обычных кратных точек и точек ветвления). Тем не менее мы излагаем этот алгоритм, так как он (при нынешнем состоянии наших знаний) представляет собой разумный подход к задаче о кручении дивизоров в случае полей алгебраических чисел.

ГРАНИЦА_КРУЧЕНИЯ

(Вариант 2)

Вход:

K: поле алгебраических чисел.

F(X, Y): уравнение кривой, определенной над K.

Случай 3. Кривая имеет 3-кручение (но не имеет ни-5-, ни-7-кручения). Тогда максимальная величина группы кручения равна 10.

Случай 4. Кривая имеет 5-кручение (но не 3-кручение). В этом случае не может быть и 2-кручения (рассмотрите не-3-часть), поэтому группа кручения должна иметь порядок 5.

Случай 5. Кривая имеет только 2-кручение. В этом случае кручение ограничено числом 4.

К сожалению, я не нашел хорошего способа механизировать интуицию такого рода, поэтому нам остается граница

minimum (НЕ_Q_ЧАСТЬ*МАКСИМАЛЬНАЯ_СТЕПЕНЬ (Q, НЕ_P_ЧАСТЬ))
(НЕ_P_ЧАСТЬ*МАКСИМАЛЬНАЯ_СТЕПЕНЬ (P, НЕ_Q_ЧАСТЬ)).

Выход:
N: граница кручения кривой F над K,
такая что настоящее кручение — делитель числа N (в отличие от варианта 1, где мы знали только, что кручение не превосходит N).
[1] Для P = 2, 3, 5, ... делать:
Для каждого простого идеала P′ поля K, такого, что P′|P, делать:
Если ХОРОШАЯ_РЕДУКЦИЯ (F, K, P′),
то делать:
[1.1] НЕ_P_ЧАСТЬ := КОНЕЧНОЕ_КРУЧЕНИЕ (F, K, P′).
Алгоритм КОНЕЧНОЕ_КРУЧЕНИЕ должен давать кручение по модулю простого идеала P′.
[1.2] Перейти к [2].
[2] Для Q = простое число, большее P, ..., делать:
Для каждого простого идеала Q′ поля K, такого, что Q′|Q, делать:
Если ХОРОШАЯ_РЕДУКЦИЯ (F, K, Q′),
то делать:
[2.1] НЕ_Q_ЧАСТЬ := КОНЕЧНОЕ_КРУЧЕНИЕ (F, K, Q′).
[2.2] Перейти к [3].
[3] ОТВЕТ1 := ВХОЖДЕНИЯ (Q, НЕ_P_ЧАСТЬ)*НЕ_Q_ЧАСТЬ

ВХОЖДЕНИЯ (Q, НЕ_Q_ЧАСТЬ)
ОТВЕТ2 := ВХОЖДЕНИЯ (P, НЕ_Q_ЧАСТЬ)*НЕ_P_ЧАСТЬ

ВХОЖДЕНИЯ (P, НЕ_P_ЧАСТЬ)
Выдать ответ НОД (ОТВЕТ1, ОТВЕТ2).

ВЫЧИСЛИТЕЛЬНЫЕ СООБРАЖЕНИЯ

В этом разделе мы опишем некоторые проблемы, возникающие при реализации этого «модулярного» алгоритма, и наметим некоторые решения. Так как у меня нет ничего даже отдаленно напоминающего полную реализацию алгоритмов, описанных в этой главе, настоящий раздел вполне может оказаться неполным.

Мы знаем, что поле классов вычетов конечно и имеет характеристику p. В действительности оно получено из множества целых элементов поля $\mathbb{Q}(l)$ отождествлением тех элементов, разность которых лежит в соответствующем простом идеале. Так как p лежит в этом простом идеале, мы должны рассматривать лишь те элементы $\mathbf{Z}[l]$, целые коэффициенты которых лежат между 0 и p, т. е. некоторое конечное множество. Следовательно, мы можем построить поле классов вычетов путем прямого перечисления, хотя это и не всегда эф-

фективно. Так как это поле имеет характеристику p, оно содержит подполе, изоморфное полю целых чисел по модулю p. Вычисления над такими полями, являющимися расширениями поля целых вычетов по модулю p, изучались в статье (Mignotte, 1976).

Для проверки на хорошую редукцию мы должны определить не только неприводимость, но и абсолютную неприводимость в поле классов вычетов. Мы можем с помощью алгоритма Берлекампа (Zimmer, 1972) факторизовать многочлены от одной переменной в подполе, которое соответствует целым числам по модулю p, но этого недостаточно. Например, в поле вычетов по модулю 5 поля $\mathbb{Q}(\sqrt{2})$ (см. подробнее рассмотрение этого примера в следующем разделе) многочлен X^2+2 раскладывается на множители, хотя в поле целых чисел по модулю 5 он не раскладывается. Следовательно, даже для многочлена, определенного над этим подполем, редукция к этому подполю неадекватна как стратегия факторизации.

В статье (Berlekamp, 1970) изложен алгоритм сведения задачи о разложении многочленов от одной переменной над полем порядка p^k для простого p к задаче разложения над простым полем, однако это нелегкий процесс[1]), и я его еще не реализовал. Мы можем вместо этого использовать методы Трейгера (Trager, 1976), обобщенные (там, где они применимы) на конечные поля и представление алгебраических выражений со многими переменными (подробности и алгоритмы см. в приложении 3). С их помощью мы можем свести нашу задачу к задаче разложения гораздо большего многочлена над полем из p элементов, которую можно затем решить непосредственно. Как только мы получим алгоритм факторизации для одной переменной, мы можем почти во всех случаях[2]) достроить его до алгоритма для многих переменных, используя, по существу, те же приемы, что и в работах по p-адическому разложению (Wang, 1978) или (Yun, 1973, 1976)).

ПРИМЕР НАД АЛГЕБРАИЧЕСКИМИ ПОЛЯМИ

Рассмотрим теперь кривую $Y^2=X^3+8$ с дифференциалом $1/Y$. Она не обладает хорошей редукцией ни по модулю 2 (так как превращается в кривую $Y^2=X^3$, род которой

[1]) В статье (Mignotte, 1976) этот процесс назван «сложным, но эффективным».

[2]) При факторизации для многих переменных может оказаться, что все значения остальных переменных «несчастливые» в том смысле, что получающиеся многочлены уже не свободны от квадратов. Однако это может произойти лишь для конечного множества простых чисел, так что в худшем случае мы можем позволить себе счесть и это случаем плохой редукции и попробовать другое простое число.

уже не равен 1), ни по модулю 3 (так как превращается в кривую $Y^2=(X-1)^3$, род которой также не равен 1). Она обладает хорошей редукцией как по модулю 5, так и по модулю 7, и не-p-части кручения равны соответственно 6 и 12. Следовательно, кручение не превосходит 6. (В действительности эта кривая не имеет кручения над $\mathbb{Q}$, но имеет одну образующую бесконечного порядка (Birch & Swinnerton-Dyer, 1963), которая над конечным полем должна отображаться в дивизор с кручением.)

Над полем $\mathbb{Q}(\sqrt{2})$ ситуация несколько иная. 2 и 3 простые числа и все еще дают плохую редукцию (и вообще расширение основного поля не избавляет нас от простых чисел, дающих плохую редукцию). 5 — простое число этого поля, и соответствующее поле классов вычетов имеет 25 элементов, которые можно записать в виде $\{i+j\sqrt{2},\ 0<i,\ j\leqslant 4\}$. Рассматриваемая кривая имеет 36 точек конечного порядка в этом поле классов вычетов.

В $\mathbb{Q}(\sqrt{2})$ рациональное простое число 7 разлагается в произведение двух простых идеалов: $\langle 7,\ 4+\sqrt{2}\rangle$ и $\langle 7,\ 3+\sqrt{2}\rangle$. Первый из них дает хорошую редукцию, и поле классов вычетов имеет 7 элементов, которые мы можем приравнять числам от 0 до 6 по модулю 7. Наша кривая имеет над этим полем 12 точек конечного порядка (точно так же, как в случае поля $\mathbb{Q}$, ибо оба поля классов вычетов изоморфны и изоморфизм сохраняет кривую). Таким образом, оценки 36 для не-5-кручения и 12 для не-7-кручения приводят к границе 12 для кручения кривой над полем $\mathbb{Q}(\sqrt{2})$. Хотя мне не известен никакой простой[1]) способ нахождения кручения эллиптической кривой над полями, отличными от поля рациональных чисел, эта кривая имеет точку порядка 6 над полем $\mathbb{Q}(\sqrt{2})$, а именно $X=4$, $Y=6\sqrt{2}$, причем эта удвоенная точка (имеющая поэтому порядок 3) проникает в некоторые сравнительно простые интегралы (см. подстрочное примечание в примере 3 приложения 2).

Заметим, что, хотя наша точка имеет порядок 3 (или, возможно, 6), мы получаем лишь границу 12 с помощью второго метода и лишь 252 с помощью первого метода. Таким образом эти методы, хотя они, без сомнения, пригодны для реализации и эффективны в математическом смысле слова, имеют ограниченную ценность для практических вы-

[1]) Насколько мне известно, алгоритмы из статьи (Birch & Swinnerton-Dyer, 1963) не были реализованы над полями, отличными от поля рациональных чисел.

числений. Одна из причин этого — возможность дивизоров бесконечного порядка, которые склонны отображаться в точки высокого (но конечного) порядка в группе кручения, соответствующей хорошей редукции, порождая таким образом завышенные оценки.

Это особенно неприятно с вычислительной точки зрения, так как оценка нужна нам только тогда, когда имеются дивизоры бесконечного порядка, ибо если дивизор имеет конечный порядок, то мы найдем этот порядок независимо от того, дана ли нам оценка.

Глава 9

ВЫВОДЫ

В этой главе мы резюмируем как теоретические, так и практические результаты этой монографии. Затем мы укажем, в каких местах теория допускает усовершенствование, а реализация неполна или не столь эффективна, сколь это возможно. Последняя ситуация распадается на два класса — задачи, связанные с проблемой интегрирования, и общие проблемы компьютерной алгебры. Затем мы рассмотрим важнейший открытый вопрос: до какой степени эти методы могут быть приспособлены к работе с трансцендентными функциями наряду с алгебраическими?

СОСТОЯНИЕ ТЕОРИИ

В гл. 4 мы сформулировали и доказали теорему Риша (Risch, 1970), которая сводит задачу интегрирования (для алгебраических функций) к задаче о дивизорах с кручением. В гл. 6 мы решаем задачу о дивизорах с кручением для основных полей, которые трансцендентны над полем рациональных чисел, используя работы (Manin, 1957, 1963). В гл. 8 мы используем метод «хорошей редукции», чтобы решить задачу о дивизорах с кручением для основных полей, алгебраических над полем рациональных чисел. Поэтому мы можем решить задачу о дивизорах с кручением для всех основных полей, и тем самым задача интегрирования алгебраических функций полностью решена.

Таким образом, теория завершена, хотя она не всегда так хороша, как хотелось бы. Есть много областей, где она могла бы быть поизящнее, и некоторые из них описаны ниже в разд. «Дальнейшие теоретические исследования».

СОСТОЯНИЕ РЕАЛИЗАЦИИ

Хотя теория полна, имеющаяся реализация на базе REDUCE-2 неполна в нескольких областях. Только одна из них относится к задаче интегрирования, описываемой в настоящей монографии, — это результаты гл. 8, которые не были реализованы в сколько-нибудь значительной степени.

Реализация неполна также и в том отношении, что не все вспомогательные алгоритмы надлежащим образом подключены к основной системе. Мне пришлось разрабатывать многие из них отдельно либо потому, что этому препятствовала слишком высокая стоимость прогонки основной программы до того места, где вызывается вспомогательный алгоритм, либо из-за трудностей с подбором подходящих тестовых примеров.

Другие важные части, нуждающиеся в завершении, таковы:

1. Работа с алгебраическими выражениями, не представимыми через квадратные корни.
2. Нахождение границ кручения с помощью более тонких методов (см. различные комментарии в гл. 8).
3. Реализация теста Кэли (см. гл. 5).
4. Использование работ Лютца — Нагеля и Мазура (см. пояснения ниже и пример 7 из приложения 2).

Те из них, которые не рассмотрены в других местах, кратко упоминаются ниже.

Алгебраические выражения. На пути реализации оперирования с алгебраическими выражениями общего вида нет больших теоретических трудностей. Тем не менее по приводимым ниже причинам требуются значительные программистские усилия.

a) Многочлен теперь уже не обязательно полностью разлагается в расширении, порожденном одним из его корней.

b) По соображениям эффективности желательно учесть обобщения недавних результатов (Trager, 1976, 1978) о поле определения интеграла, в частности заметить, что если знаменатель подынтегрального выражения имеет неприводимый множитель $f(x)$, то это выражение либо имеет вычеты во всех множителях $f(x)$, либо вообще не имеет вычетов в этих множителях.

c) И вообще можно достичь очень большой экономии, если подойти к рассматриваемым задачам с точки зрения теории Галуа.

Рассматриваемая программа в действительности объединена с программой трансцендентного интегрирования (Norman & Mooge, 1977), так как они используют один и тот же аппарат разложения на множители (см. разд. 5 приложения 1) и имеют один и тот же интерфейс пользователя, а именно оператор INT, первым аргументом которого является интегрируемая функция, а вторым — переменная интегрирования. Вместе эти программы занимают примерно 13 500 строк на объектном языке системы REDUCE-2 и около 285 000 байтов объектного LISP-кода после компиляции (ffitch & Norman, 1977). Обе эти цифры не включают основ-

ную часть системы REDUCE-2. Эта программа интегрирования распространяется теперь с исследовательскими целями вместе с системой REDUCE-2 и операционной системой MTS.

Лютц — Нагель. В настоящий момент процедура для эллиптических кривых (НАЙТИ_ПОРЯДОК_ЭЛЛИПТИЧЕСКИЙ, описанная в гл. 7) не использует теорему Лютца — Нагеля (см. гл. 7), когда кривая и дивизор эллиптические и определены над рациональными числами, а применяет вместо этого абсолютную оценку 12, найденую Мазуром (см. гл. 5). Причина этого историческая: оценка Мазура была реализована гораздо раньше. В действительности, как показывает пример 7 из приложения 2, использование теоремы Лютца — Нагеля (благодаря которой мы можем определить, есть ли у нас дивизор с кручением) намного более эффективно в случае, когда у нас нет дивизора с кручением, т. е. в случае логарифмической неинтегрируемости.

НЕОБХОДИМЫЕ ДОПОЛНИТЕЛЬНЫЕ ПРОЦЕДУРЫ

Чтобы превратить имеющуюся сейчас программу в полную практически применимую систему интегрирования алгебраических функций, нужно много дополнительных процедур, в том числе следующие:

1. Преобразование комплексных логарифмов в обратные тригонометрические функции.
2. Эффективное вычисление наибольших общих делителей (см. разд. 3 приложения 1).
3. Факторизация (см. разд. 5 приложения 1).
4. Эффективное вычисление результантов.
5. Более обозримая распечатка сложных результатов, включая повторяющиеся длинные ядра (см. разд. 1 приложения 1).
6. Рационализация оперирования с линейными уравнениями.
7. Эвристическое преобразование подынтегрального выражения.
8. Факторизация над полями алгебраических чисел.

Мы опишем ниже некоторые из них чуть подробнее.

Обратные тригонометрические функции. Комплексные логарифмы кажутся неестественными многим людям, тем более что обычные таблицы интегралов почти всегда дают свои ответы в терминах обратных тригонометрических и гиперболических функций. Поэтому было бы приятно, если бы логарифмическую часть ответа можно было выразить в таком виде, хотя в ряде случаев ее почти наверняка лучше оставлять в логарифмической форме (см. примеры 4 и 5 из при-

ложения 2). Один из случаев, когда обратные гиперболические функции выглядят привычнее,— это пример 1 из приложения 2, где интеграл от 1/SQRT(X²—1) дан в виде LOG(SQRT(X²—1)+X), в то время как многие таблицы интегралов дают Arch(X).

Результанты. В процессе разложения на множители над алгебраическими расширениями полей нам нужно вычислять результанты (Trager, 1976, алгоритм БЕСКВ_НОРМА); см. также алгоритм БЕСКВ_НОРМА в приложении 3, а также гл. 2 и разд. 5 в приложении 1. Алгоритм, который есть у меня сейчас, чрезвычайно груб. Он использует определение результанта как определителя матрицы (van der Waerden, 1975, с. 84) и алгоритмы (Lipson, 1969) вычисления определителей, имеющиеся в системе REDUCE (Hearn, 1973). Как отмечено в работе (Norman, 1978b), имеются гораздо лучшие алгоритмы (Collins, 1971). Коллинз показывает, что два алгоритма (оба намного лучше нашего) могут различаться более чем в 10 раз по затрачиваемому времени. Это — область компьютерной алгебры, в которой все еще наблюдается существеный прогресс, и имеется новый алгоритм, основанный на работе (Zippel, 1979), который, видимо, стоит того, чтобы исследовать его.

Линейные уравнения. Нынешняя версия программы содержит не менее[1]) 5 наборов процедур для решения линейных уравнений.

(a) Набор для нахождения линейных зависимостей в алгоритмах РЕДУКЦИЯ_К_ЦЕЛОМУ_БАЗИСУ и РЕДУКЦИЯ_К_НОРМАЛЬНОМУ_БАЗИСУ.

(b) Набор, нужный в алгоритме КОУТС для нахождения функции, имеющей нужные нули.

(c) Набор, позволяющий разложить вычеты по базису **Z**-модуля, который они порождают.

(d) Набор для использования в алгоритме АЛГЕБРАИЧЕСКАЯ_ЧАСТЬ.

---

[1]) Кроме того, алгоритм трансцендентного интегрирования (Norman & Moore, 1977) содержит еще 2, и программа разложения на множители обычно включает еще одну (Berlenkamp, 1967). Сама система REDUCE (Hearn, 1973) содержит такую процедуру, основанную на методе Барейсса (Lipson, 1969), но она, к сожалению, бесполезна при работе с алгебраическими выражениями. Как было отмечено в гл. 2, мы должны уменьшать степени алгебраических выражений в процессе преобразования матриц и линейных уравнений, однако реализованный метод имеет общие черты с алгоритмом Евклида, который, как показано в гл. 2, может не сработать, если уменьшаются степени алгебраических выражений, так как во входной информации некоторого шага могут присутствовать члены, которые могут неожиданно исчезнуть на выходе.

(e) Набор для решения линейных уравнений, нужных для точного нахождения уравнений Пикара — Фукса (см. гл. 6).

Все они имеют тонкие отличия друг от друга. Например, (c) должен считать, что SQRT(A) не имеет ничего общего с A, в то время как остальные должны отождествлять $(\mathrm{SQRT}(A))^2$ с A. Далее, (c) работает с целыми числами, в то время как остальные процедуры работают с рациональными функциями или многочленами. Дополнительное осложнение состоит в том, что (a) должен выдавать первую линейную зависимость, в то время как остальные процедуры обычно составлены в предположении, что можно переставлять строки как угодно. Далее, система в (c) часто имеет (почти) верхнюю треугольную матрицу и потому, наверно, требует специального алгоритма.

Рассматриваемые задачи решения линейных уравнений могут иметь различный размер и большой разброс в степени разреженности. Поэтому процедуры должны эффективно работать с небольшими системами (иначе тривиальные задачи потребуют несоразмерного времени), работать с разреженными системами с помощью разреженных алгоритмов (иначе некоторые задачи окажутся практически неразрешимыми — см. подстрочное примечание к примеру 4 из приложения 2) и справляться за разумное время с неразреженными системами среднего размера (большие неразреженные системы, вероятно, невозможны в любом случае). Это требует, вероятно, какого-то «метаалгоритма», который обозревает задачу и выбирает в зависимости от нее используемый алгоритм: непосредственный метод исключения Гаусса для небольших систем, разреженный алгоритм (возможно, правило Крамера и алгоритм для разреженных определителей (Smit, 1976) с анализом, позволяющим избежать многократного вычисления одного и того же минора) или адаптированный бездробный метод (Lipson, 1969). Возможно, нужен еще специальный модуль для работы с разреженными строками неразреженной матрицы (см. следующее подстрочное примечание).

Но и это еще не все: требуются различные копии процедуры для многочленов, целых чисел и рациональных функций, а также в зависимости от того, производится или нет упрощение степеней алгебраических выражений. Так как имеется лишь конечное число возможных случаев (не более чем 3×2, да и они не все нужны на самом деле), то теоритически достаточно использовать стратегию MODE_REDUCE (Hearn, 1976), при которой один и тот же участок исходной программы компилировался бы 6 раз. Однако я думаю, что действительно динамическая система вроде SCRATCHPAD/370, предложенной в статье (Jenks, 1979), была бы более эффективной и обеспечила бы бо́льшую гибкость.

Программа, написанная в настоящее время для задачи (b), часто тратит очень много времени на распознавание неразрешимости системы, хотя она и содержит некоторые простые улучшения [1]) по сравнению со стандартным исключением по Гауссу, предназначенные для ускорения ее работы на разреженных или частично разреженных системах уравнений. Так как неразрешимость системы, рассматриваемой в пункте (b), эквивалентна неудачному завершению алгоритма Коутса, это может происходить довольно часто. По этой причине я вставил модулярную проверку, чтобы убедиться в разрешимости уравнений по модулю некоторого простого числа до того, как делается попытка решить его в настоящем основном поле. Это, очевидно, нужно было бы сделать и для всех систем, решающих линейные уравнения, однако усилия, которые пришлось бы затратить на дополнительную 3-кратную реализацию, по-видимому, перевешивают возможный выигрыш. С другой стороны, если бы одна реализация могла справиться со всеми задачами решения линейных уравнений, то эта простая проверка была бы доступна для всех них.

Эта модулярная проверка в настоящее время используется лишь для получения ответа «да — нет», но из ее результатов можно извлечь больше информации. Например, используя некоторые идеи из теории вероятностных алгоритмов (Zippel, 1979), мы можем сказать, что нулевые компоненты в модулярном решении, вероятно, представляют нули в настоящем решении. Поэтому перед тем, как решать общую задачу, нам следует посмотреть, нет ли решения, в котором все эти элементы действительно равны нулю. К сожалению, если такого решения нет, то мы не можем утверждать, что исходная система неразрешима, а должны попытаться решить эту большую систему. Тем не менее экономия, вероятно, была бы значительной.

Эвристические преобразования подынтегрального выражения. Имеется несколько способов преобразовать входные данные программы интегрирования, чтобы ускорить выполнение этой программы. Простейшим из них является, вероятно, указанная в конце примера 2 из приложения 6 замена A на B^2, которая ускоряет выполнение программы примерно на 10 %.

Более тонкий пример — интегрирование функции 1/SQRT(X**2 + 1) + 100/SQRT(X**2 + 10 000). Она имеет выче-

[1]) Основное из них — предварительный просмотр с целью выбора самой разреженной строки для ближайшего исключения. Это означает, в частности, что строки, имеющие единственный ненулевой элемент, автоматически исключаются первыми и поэтому не страдают от «вставок» (Smit, 1976).

ты 99, 101, —99, —101 в четырех плейсах, лежащих на бесконечности. Вычисления в похожих случаях были рассмотрены в примере 8 из приложения 2. Если бы мы попытались проинтегрировать сначала одно из слагаемых, то смогли бы это сделать за 868 мс, а интегрирование остатка заняло бы примерно столько же времени, что дает суммарное время менее 2 с. Мы, однако, не можем произвольно разбивать подынтегральное выражение на слагаемые, так как если $f(x)$ и $g(x)$ не интегрируемы, то сумма $f+g$ все еще может быть интегрируема.

Поэтому хорошая программа интегрирования искала бы в подынтегральном выражении интегрируемые частичные слагаемые. Найдя, она убирала бы их и пыталась бы решить оставшуюся меньшую задачу.

Разложение на множители над полями алгебраических чисел. Наш подход к разложению был основан на работе (Trager, 1976) (см. гл. 2), и, насколько мне известно, не существует иного алгоритма интегрирования над алгебраическими расширениями общего вида. Для разложения над полями алгебраических чисел имеется обобщение (Wang, 1976) схемы разложения над целыми числами (Wang & Rotschild, 1975). См. также статью (Weinberger & Rotschild, 1976). Если существует подходящее небольшое простое число, этот метод не намного медленнее, чем разложение того же многочлена от многих переменных над целыми числами. Если же такого простого числа не существует, то метод Вана намного менее эффективен, и кажется (Trager, 1976), что он теперь реализовал для этого случая алгоритм Трейгера.

Поэтому в целях эффективности мы должны были бы использовать метод Вана там, где он непосредственно применим (см. также комментарии к алгоритму АЛГ_МНОЖИТЕЛЬ_2 в приложении 3). Теперешняя схема разложения (см. разд. 5 приложения 1) не подходит для такой адаптации, так как она не использует p-адические методы работы (Wang & Rotschild, 1975). Версия, разрабатываемая Норманом и Муром (Norman, 1978b), основана на принадлежащем Вану (Wang, 1978) усовершенствовании алгоритма Вана и Ротшильда, и мне не вполне ясно, как можно приспособить эти усовершенствования к случаю полей алгебраических чисел. Дальнейшая работа в этой области, наверняка, существенно ускорит процесс интегрирования во многих важных случаях.

ВЫВОДЫ ДЛЯ СИСТЕМ АЛГЕБРАИЧЕСКИХ ВЫЧИСЛЕНИЙ

В этом разделе мы хотим собрать вместе все высказанные в этой монографии соображения, относящиеся к проектированию и реализации систем компьютерной алгебры, и сде-

лать несколько замечаний, которыми должен руководствоваться создатель любой такой системы в будущем. Здесь имеются два аспекта: соображения, специфические для интегрирования (особенно в применении к алгебраическим функциям), и соображения, которые, по моему мнению, имеют отношение к реализации любого большого алгоритма.

Одно из общих соображений, относящихся к алгебраическим системам и возникших в результате реализации описываемой здесь системы интегрирования, заключается в необходимости «изощренных алгоритмов», таких, как p-адические методы (Yun, 1976) или методы работы с разреженными матрицами (Smit, 1976). Можно написать весьма результативную систему, используемую математиками и научными работниками (например, REDUCE-2 (Hearn, 1973)) и не использующую упомянутых методов, однако такая система не подходит для наших целей (см. примеры в разд. 3 и 5 приложения 1). Кроме того, эти алгоритмы в целом весьма длинны [1]) и трудны.

Одно из возникающих специальных соображений касается алгебраических выражений. Оказалось, что REDUCE-2 трудно приспособить к оперированию с ними, и кажется (в той степени, в какой можно указать одну конкретную причину), что это происходит потому, что о них не думали при проектировании системы. Поэтому любая будущая алгебраическая система должна, по-моему, учитывать проблемы оперирования с алгебраическими выражениями на стадии проектирования, чтобы они вписывались в общую структуру этой системы.

Как уже упомянуто раньше (например, выше при рассмотрении линейных уравнений), кажется, что системы компьютерной алгебры недостаточно гибки: проектировщик создает некоторый фиксированный список основных типов [2]), а всю работу по добавлению нового типа приходится проделывать самому пользователю. В частности, наличие решателя систем линейных уравнений не означает, что его можно применять для многочленов или для элементов конечных полей. Это ве-

[1]) Хотя число строк исходной программы — вряд ли хорошая мера сложности, трудно представить себе лучшую меру. Поэтому мы приведем некоторые цифры в подтверждение этого соображения. Система REDUCE-2 (Hearn, 1973) содержит около 2000 строк на языке препроцессора и около 5500 строк алгебраической системы, 1500 из которых принадлежат «пакету программ для физики высоких энергий», который не имеет отношения к нашей задаче. Черновая версия системы для p-адического разложения и нахождения наибольшего общего делителя (Norman, 1978b) занимает сейчас около 3500 строк и, вероятно, станет длиннее, когда будет закончена.

[2]) В случае системы REDUCE-2 он состоит из типов ARRAY, MATRIX, STANDARD FORM (т. е. многочлен), STANDARD QUOTIENT (т. е. рациональная функция) и PREFIX FORM (служит для представления неупрощенных выражений).

дет к дублированию и делает реализацию нового алгоритма, использующего новые типы данных, более утомительной, чем необходимо.

Сейчас разрабатываются два больших проекта, связанных с этой задачей, — MODE_REDUCE (Hearn, 1976) и SCRATCHPAD/370 (Jenks, 1979) (Jenks & Davenport, 1980). Оба эти подхода, которые можно грубо охарактеризовать как основанные соответственно на статическом и динамическом анализе, должны привести к большой экономии в этой задаче. Важной исследовательской задачей была бы реализация системы интегрирования на одной из этих систем. Это показало бы, оправдывают ли эти системы возлагаемые на них надежды, и послужило бы орудием будущих исследований интегрирования.

БУДУЩИЕ ТЕОРЕТИЧЕСКИЕ ИССЛЕДОВАНИЯ

Важная область будущих исследований — алгоритм Коутса. Очень часто (особенно если обрабатываемый дивизор имеет большие порядки в нескольких плейсах) случается, что промежуточные выражения в алгоритме Коутса намного больше, чем окончательный ответ. Это может проявляться в том, что большинство функций из нормального целого базиса (т. е. списка функций с полюсами не хуже, чем у рассматриваемого дивизора) может иметь длиннейшие выражения, а ответ может быть чуть ли не самой маленькой из этих функций. По-видимому, отсюда следует, что алгоритм Коутса — не обязательно наилучший метод нахождения функции с нужными нулями и полюсами, но я не представляю, как можно было бы делать лучше.

С предыдущим тесно связан вопрос: Как долго работает алгоритм Коутса? У меня нет сейчас временно́го анализа этого алгоритма главным образом потому, что нет очевидного метода, предсказывающего по входу число шагов редукции ни для конечных плейсов, ни на бесконечности.

Еще один вопрос в связи с алгоритмом Коутса сформулирован в гл. 3: нужен ли шаг 3? Очевидно, что алгоритм стал бы аккуратнее в теоретическом отношении и работал бы существенно быстрее, если бы удалось показать, что этот шаг не нужен при условиях, выполнения которых мы всегда можем добиться. Я несколько продвинулся в этом направлении, но мне еще предстоит много сделать.

Если судить по гл. 8, то имеется мало хороших методов работы над полями классов вычетов. В качестве ответа там предложено бесхитростное перечисление, но очевидно, что в этой области нужны дальнейшие исследования.

В гл. 5 сказано о том, что теория интегрирования имеет связи с теорией непрерывных дробей (Chebyshev, 1857, 1860).

Она имеет современные приложения (Schinzel, 1962), которые можно использовать вместе с результатом Мазура (Maur, 1977) для построения совершенно нового, хотя и несколько специализированного, алгоритма интегрирования. Очевидно, что предстоят дальнейшие исследования связей между двумя областями.

Важная область, где предложенный алгоритм не слишком удовлетворителен, — это вопрос о кривых рода больше 1 над полями алгебраических чисел (включая поле рациональных чисел). В случае кривых над трансцендентными полями у нас есть исследования Манина (см. гл. 6), которые позволяют нам узнать, имеет ли дивизор конечный порядок. В случае кривых рода 1 над полями алгебраических чисел мы можем использовать алгоритм ЛЮТЦ_НАГЕЛЬ (гл. 7), чтобы определить, имеет ли дивизор конечный порядок, и найти этот порядок. Но в остальных случаях нам приходится вычислять границу (зачастую намного большую, чем нужно) и затем проверять все дивизоры вплоть до этой границы. Ситуация была бы гораздо лучше, если бы у нас был механизм, проверяющий без такого перебора, имеет ли такой дивизор конечный порядок.

РАСШИРЕНИЕ НА ТРАНСЦЕНДЕНТНЫЕ ФУНКЦИИ

Как уже сказано в гл. 1, проблема интегрирования чисто трансцендентных функций была решена теоретически более 10 лет назад (Risch, 1969); см. также (Moses, 1967, 1971), (Norman & Moore, 1977) и обзор в работе (Norman & Davenport, 1979). В настоящей монографии показано, как решается задача интегрирования чисто алгебраических функций, и большая часть необходимого программирования уже проделана.

Таким образом, возникает вопрос о смешанных функциях, которые трансцендентны, но не чисто трансцендентны, например $\sqrt{1+\log x}$ или [1])

$$\log\left(\sqrt{a}+\sqrt{b-x}\right)/\sqrt{(1-x^2)\,(b-x)\,(a-x)}$$

((Caviness, 1978), приведено в статье (Davenport, 1979c)).

---

[1]) Заметим, что нелегко решить, имеет ли функция такой вид. Можно, например, подумать, что функция $\sin x+\sqrt{1-\sin^2 x}$ не является чисто трансцендентной, но ее можно записать в виде $\sin x+\cos x$, а затем выразить через $\operatorname{tg}(x/2)$. Это составляет целую область «структурных теорем» (Epstein, 1979) или (Rothstein & Caviness, 1979), которая тесно связана с задачей интегрирования.

Теорема 1 (Лиувилль[1])). *Пусть F — дифференциальное поле с алгебраически замкнутым полем констант K. Пусть f лежит в F и g элементарен над F, причем $g' = f$. Тогда имеются $v_0, v_1, \ldots, v_k$ из F и $c_1, \ldots, c_k$ из K, такие, что* $f = v_0' + \sum c_i \frac{v_i'}{v_i}$, *т. е.* $g = v_0 + \sum c_i \log v_i$.

Мы можем назвать v_0 *рациональной частью* интеграла, а остальную сумму *логарифмической частью.* (Эта терминология несколько иная, чем в гл. 4, где v_0 была названа алгебраической частью. Причина в том, что v_0 рационально относительно одночленов, входящих в f, которые в гл. 4 всегда были алгебраическими, а теперь могут быть и трансцендентными.)

Если мы знаем логарифмическую часть, то рациональную часть можно найти с помощью обобщения методов Риша[2]) (Risch, 1969), реализованного Мозесом (Moses, 1967, 1971); см. также (Norman & Moore, 1977)). Как и в гл. 4, главная проблема — нахождение логарифмической части.

Логарифмическая часть определяется вычетами подынтегрального выражения точно так же, как и раньше, но это подынтегральное выражение определено уже не над кривой, а над произвольным многообразием. Например, выражение

$$\log\left(1 + \sqrt{1 + x^2}\right) + \frac{\sqrt{1 + \log x}}{\log x}$$

следует рассматривать как $Z + V/W$ в поле $\{K(X, Y, Z, V, W) \mid Y^2 = 1 + X^2,\ V^2 = 1 + W\}$. Теория дивизоров над произвольными многообразиями гораздо менее разработана, чем над кривыми, и я не вижу никакого подхода к обобщению алгоритмов из этой монографии на произвольные многообразия.

Однако часто случается, что многообразие можно рассматривать как произведение кривой и гиперплоскости, а плейсы, в которых определены дивизоры, можно рассматривать как произведения точек на кривой и той же самой гиперплоскости. В этом случае нам достаточно работать просто на кривой, и рассуждения, приведенные в этой монографии, все еще годятся.

---

[1]) В формулировке Риша (Risch, 1970). Основная теорема, на которой основана вся эта работа (см. гл. 4) получена Ришем в качестве следствия этой теоремы.

[2]) В специальном случае, когда алгебраическое расширение — это просто присоединение радикала, может показаться, что методы Трейгера (Trager, 1979) без усилий обобщаются на случай смешанных интегралов. Однако пока что эти методы обеспечивают только алгебраическую часть, и я считаю, что логарифмическая часть гораздо труднее.

Пример 1. $\log(\sqrt{a}+\sqrt{b-x})/\sqrt{(1-x^2)(b-x)(a-x)}$ (Caviness, 1978). Положим $c^2=A$ и $y^2=b-x$, так что подынтегральное выражение принимает вид $\log(c+y)\times$ $\times(\sqrt{(1-(b-y^2)^2)(c^2-b+y^2)})^{-1}$. Оно не имеет вычетов. (Это нетривиально, если использовать чисто вычислительный подход, но можно применить к корню квадратному механизм разложения Пюизо.) Поэтому подынтегральное выражение чисто рационально (как функция от y, логарифма и квадратного корня) и в действительности неинтегрируемо.

Пример 2. Рассмотрим функцию

$$\frac{\log x\,(x^4+yx^3-x^2)+y\,(x^3+x^2-x)+x^4+x^3-2x^2-x+1}{x\,(y+x)\,(x^2-1)}$$

где $y^2=x^2-1$. Эта функция имеет вычеты ± 1 в двух гиперплоскостях, лежащих над $x=$ бесконечность. Таким образом (ср. пример 1 из приложения 2), эта функция имеет логарифмическую часть $\log(y+x)$. Тогда рациональная часть равна $\log(x)*y$, и вся функция интегрируема. В действительности моя программа, работая вместе с программой Нормана и Мура (Norman & Moore, 1977), способна проинтегрировать ее.

Пример 3. Рассмотрим подынтегральную функцию

$$\frac{\log^2 x\,yx^2+(\log x+1)\,(yx^2-y+x^4-x^2)}{x\,(\log x+y)\,(x^2-1)}$$

где, как и раньше, $y^2=x^2-1$. Подынтегральная функция имеет вычеты ± 1 на двух гиперплоскостях $y=\pm\log(x)$. Мы можем тогда заключить, что логарифмическая часть должна быть равна $\log(\log(x)+y)$, даже несмотря на то что любая попытка получить это путем вычислений должна была бы, вероятно, опираться на эвристики, например на замену $\log(x)$ выражением $z=y+\log(x)$. Мы можем затем получить, что алгебраическая часть должна быть равна $y\log(x)$.

Пример 4. К предыдущим примерам смешанных интегралов можно было подойти с помощью стратегии, описанной выше, и методов настоящей монографии. Я не вижу способов решить таким образом этот пример. Рассмотрим $\log(y+$ $+\log^2 x)$, где $y^2=x^2-1$, как и в двух предыдущих примерах. Производная этой функции равна

$$\frac{2\log x\,(x^2-1)+yx^2}{x\,(x^2-1)\,(\log^2 x-y)}.$$

Рассмотрим задачу ее интегрирования. С интуитивной точки зрения, быть может, и резонно рассмотреть член вида

$\log(\log^2 x + y)$, но мы пытаемся найти алгоритмические, а не эвристические методы. Подынтегральная функция имеет вычет на подмногообразии $\log^2(x) = y$ многообразия $y^2 = x^2 - 1$. Вычет равен ± 1 в зависимости от того, на какой части многообразия мы находимся, со специальными значениями в точках $x = \pm 1$. Хотя можно изучить эту ситуацию и заключить, что $\log^2(x) + y$ имеет нужные нули и полюса, я не вижу в настоящий момент, как можно было бы сделать этот шаг алгоритмически.

ВИДЫ НЕИНТЕГРИРУЕМОСТИ

Имеется два различных вида неинтегрируемости, которые могут выявиться при попытке интегрирования алгебраических функций. Первый, который мы назовем *логарифмической неинтегрируемостью*, возникает, когда мы имеем дивизор бесконечного порядка, так что вычеты подынтегральной функции нельзя учесть в интеграле с помощью логарифма. Второй, называемый *алгебраической неинтегрируемостью*, возникает, когда мы можем найти логарифмическую часть, но функция все-таки не интегрируема.

Простой пример алгебраической неинтегрируемости дает пример 2 из приложения 2, где подынтегральное выражение является в действительности дифференциалом первого рода на своей определяющей алгебраической кривой. В общем случае, когда функция алгебраически неинтегрируема, мы можем выбрать функцию, имеющую как можно больше нужных отрицательных коэффициентов Пюизо (см. алгоритм НАХОЖДЕНИЕ_АЛГЕБРАИЧЕСКОЙ_ЧАСТИ в гл. 4), и тогда остаток будет минимальной в некотором смысле алгебраической частью. В частном случае эллиптических кривых имеется разработанная теория (Whittaker & Watson, 1927) таких *эллиптических интегралов*, которой можно будет воспользоваться, если преобразовать кривую к стандартному виду (алгоритм ФОРМА_ВЕЙЕРШТРАССА в гл. 5). Следовательно, в этом частном случае мы можем свести неэлементарную алгебраическую часть к полезной стандартной форме. Интересно было бы знать, насколько это наблюдение поддается обобщению и какова его связь с работой (Ng, 1974).

В примере 7 из приложения 2 имеется пример логарифмической неинтегрируемости. Этот интеграл чрезвычайно интересен, ибо он ведет себя в точности, как логарифм (в том смысле, что его производная есть дифференциал второго рода), но логарифмом не является. Очевидно, что такого рода функции заслуживают дальнейшего исследования.

РЕЗЮМЕ

В настоящий момент существует алгоритм интегрирования алгебраических функций и программа, реализующая существенную его часть, достаточную для того, чтобы опровергнуть гипотезу Харди (см. начало гл. 4). До полной реализации еще не близко, но трудности, по-видимому, имеют скорее практический, чем теоретический характер. В двух предыдущих разделах мы видели, что можно расширять алгоритм в направлении трансцендентных функций, и есть место для обобщения нашего определения элементарности, включающего эллиптические интегралы.

Приложение 1

ИЗМЕНЕНИЯ, ВНЕСЕННЫЕ В СИСТЕМУ REDUCE-2

В этом приложении описаны различные изменения, которые потребовалось внести в систему REDUCE-2 (версия марта 1978 г.), чтобы она поддерживала систему интегрирования, описанную в нашей монографии. Они описаны здесь не для того, чтобы привлечь внимание к каким-то недостаткам системы REDUCE-2, а для того, чтобы выявить трудности, ожидающие того, кто будет реализовать алгоритмы интегрирования, описанные в этой монографии. Стоит отметить, что попытки автора реализовать эти алгоритмы с помощью системы SCRATCHPAD (Griesmer et al., 1975) пали жертвой гораздо худших проблем.

Важное понятие системы REDUCE-2 — понятие ядра (Hearn, 1973, гл. 5), которое для наших целей мы можем отождествить с переменной или упрощенным[1]) применением оператора (т. е. структурой вроде SQRT(2) или LOG(X)).

1. РАСПЕЧАТКА

В процедуру печати выражений системы REDUCE-2 были внесены изменения, чтобы структура (SQRT-1) печаталась в виде i, а не как SQRT(— 1). Это существенно сократило многие выражения и облегчило их чтение. Это не нужно в стандартной версии REDUCE-2, так как там с I работают как с независимой переменной, удовлетворяющей равенству $I^2 = -1$ (см. ниже разд. 4).

Даже после внесения этого изменеия распечатка многих выражений занимала по несколько страниц, занятых в большой степени повторяющейся распечаткой одного и того же сложного квадратного корня. В действительности первая по-

---

[1]) Термин «упростить» (simplify) имеет в системе REDUCE-2 (как и в большей части компьютерной алгебры) специальный смысл. В REDUCE-2 упрощение — это специфическая операция, целью которой является преобразование выражений в некоторую стандартную форму, которая должна быть канонической. В частности, с каждым оператором (например, SQRT, LOG) связана своя процедура упрощения, которая должна преобразовывать любое применение этого оператора к стандартной форме. Ниже, в разд. 4, мы подробно рассматриваем упрощающую процедуру для SQRT.

пытка найти функцию $X(7)$ (см. пример 4 из приложения 2) потребовала более 5 с времени центрального процессора только на то, чтобы напечатать промежуточное выражение, более простое, чем окончательный ответ. Тогда процедуры печати системы REDUCE-2 подверглись дальнейшей модификации, после которой выражение вида SQRT(...), равное предшествующему напечатанному (как часть той же формулы) выражению такого вида, заменяется на символ «lastsqrt». Пример 4 из приложения 2, где встречается $\mathrm{SQRT}(A^2+X^2)$, содержит убедительную иллюстрацию полезности этой модификации.

На самом деле это решение не слишком хорошо в ситуации, когда в выражении присутствует несколько квадратных корней. Например, выражение

$$\mathrm{SQRT}(A^2+X^2)*K+\mathrm{SQRT}(A^2+X^2)*X+\mathrm{SQRT}(A)*A+\\ +\mathrm{SQRT}(A)+\mathrm{SQRT}(A^2+X^2)$$

было бы напечатано в виде

$$\mathrm{SQRT}(A^2+X^2)*A+\mathrm{lastsqrt}*X+\mathrm{SQRT}(A)*A+\\ +\mathrm{lastsqrt}+\mathrm{SQRT}(A^2+X^2).$$

Намного предпочтительнее (но и гораздо труднее[1]) для реализации в существующей структуре системы REDUCE) другой формат, когда каждое выражение вида SQRT(...), встречающееся более 1 раза, заменяется символом вроде «ans1», а в конце выражения печатается список таких обозначений. Тогда приведенный выше пример был бы напечатан в виде

$$\mathrm{ans1}*A+\mathrm{ans1}*X+\mathrm{ans2}*A+\mathrm{ans2}+\mathrm{ans1}$$

где $\mathrm{ans1}=\mathrm{SQRT}(A^2+X^2)$

$\mathrm{ans2}=\mathrm{SQRT}(A)$.

2. ДИФФЕРЕНЦИРОВАНИЕ

Процедуры дифференцирования, имеющиеся в системе REDUCE-2, поддерживают[2]) список пар, состоящих из ядра (отличного от атома) и его производной, чтобы избежать многократного вычисления производной одного и того же ядра (которое может быть весьма сложным выражением).

[1]) Тем не менее проф. Хёрн (Hearn) реализовал этот подход в версии системы REDUCE от 15 апрля 1979 г. (непосредственно следующей за версией, которая в настоящий момент используется в системе интегрирования). Я имел возможность обработать некоторые интегралы с помощью этой новой версии системы REDUCE, и пример 5 из приложения 2 иллюстрирует результат.

[2]) По-моему эта черта была устранена из последней версии системы REDUCE-2 как раз по тем причинам, которые указаны ниже.

Первый недостаток такого подхода, с нашей точки зрения, связан с тем, что алгоритм интегрирования требует большого числа дифференцирований, зачастую в различных плейсах, так что одно и то же выражение будет представлено по-разному (например, в виде SQRT(X — 2) при X = 0, но SQRT(1 — 2X)/SQRT(X) на бесконечности). Поэтому экономия не столь велика, как можно было надеяться, в то время как для хранения списка нужен большой объем памяти (в одном случае потребовалось до 40 кбайт, что означало бы на машине IBM 370/165, установленной в Кембриджском университете, увеличение арендной платы на 12 %). Гораздо более серьезный недостаток состоит в том, что при дифференцировании данного ядра всегда выдается один и тот же ответ, хотя в промежутке могли измениться зависимости между переменными. Это может быть очень неудобно при вычислении операторов Гаусса — Манина (описанном в гл. 6), так как при этом x иногда зависит от u, а иногда нет. В силу этих причин упомянутый список пар был удален из пакета программ дифференцирования в системе REDUCE-2.

Пакет программ для работы с оператором Гаусса — Манина порождает еще одно требование, которое повело к изменению в пакете программ дифференцирования в системе REDUCE-2. Если x — переменная интегрирования и u — «параметр Манина», то иногда мы хотим вычислять производные с учетом зависимости x от u. Если x — переменная (а не функция), то это невозможно в стандартной системе REDUCE-2. Добавление этой возможности потребовало введения дополнительных проверок в пакет программ дифференцирования, и это было сделано.

3. НАИБОЛЬШИЕ ОБЩИЕ ДЕЛИТЕЛИ

Пакет программ для нахождения наибольшего общего делителя, реализованный в системе REDUCE-2 в настоящее время, использует модифицированный метод полиномиальных остатков (Hearn, 1979). Хотя это почти наверное наилучший алгоритм для широкого круга задач, он все же может потребовать необозримого времени, чтобы распознать взаимную простоту двух многочленов (особенно от нескольких переменных).

Как было замечено (Norman, 1978b), имеются гораздо более эффективные методы решения этой задачи, и я применил первую стратегию Нормана — введение предварительной модулярной проверки[1]) на взаимную простоту до обращения к процедурам нахождения н. о. д. в системе REDUCE-2. В си-

[1]) Я благодарен д-ру Норману за разрешение использовать его программу для этой цели и за помощь в ее реализации.

стеме, имеющейся в Кембридже, вычисление по модулю, меньшему 2^{24}, выполняется особенно дешево, и «маловероятно», чтобы столь большое простое число оказалось неудачным (Zippel, 1979). Я не реализовал второй метод Нормана (полностью p-адический наибольший общий делитель, т. е. EZGCD (Yun, 1973)), так как в данный момент [1]) в REDUCE-2 нет реализации с многими переменными. Уже первый метод позволяет свести до 1 с время вычислений, которые в стандартной системе REDUCE не заканчиваются за 30 с (и конца им не видно).

4. АЛГЕБРАИЧЕСКИЕ ВЫРАЖЕНИЯ

Как уже сказано в гл. 2, обработка алгебраических выражений в стандартной системе REDUCE-2 недостаточна для наших целей, так как нам (почти) все время нужны канонические выражения. Добавленные мной методы работы с алгебраическими выражениями полны лишь для выражений, порожденных с помощью операций SQRT (вложения квадратных корней допускаются), но использованные принципы допускают обобщение на алгебраические выражения общего вида.

Для работы с выражениями, содержащими алгебраические величины, в нашей системе интегрирования имеются аналоги стандартных REDUCE-2-функций для обработки выражений (например, процедура ADDSQ сложения двух рациональных функций, процедура MULTF перемножения двух многочленов), обеспечивающие каноническую форму результата (так что никакая алгебраическая величина не встречается в степени, большей или равной ее порядку, и ни в каком выражении знаменатель не содержит собственно алгебраических величин, т. е. величин, определяемых как корни уравнений). Логическая структура этих процедур (имена которых образуются путем приписывания * спереди к стандартным именам системы REDUCE) состоит из вызова стандартной REDUCE-функции с последующим применением функций SUBS2Q (или SUBS2F) и SQRT2TOP. В системе REDUCE-2 функции SUBS2Q и SUBS2F осуществляют правила подстановки вроде

FOR ALL X LET SQRT(X)** 2= X,

уничтожая таким образом все вхождения квадратных корней, возводимых в степени. SQRT2TOP — это локальная процедура, которая устраняет собственно алгебраические выражения путем умножения числителя и знаменателя на выражение, сопряженное знаменателю.

[1]) Однако такая реализация пишется в Кембриджском университете (Norman, 1978b), и я надеюсь, что смогу получить ее в ближайшем будущем.

В действительности эти процедуры работают не совсем таким образом, а пытаются справиться со степенями алгебраических величин по мере их возникновения, вместо того чтобы накапливать их и пытаться избавиться от них потом. Следует отметить, что этот несколько сомнительный участок программы делает работу, эквивалентную описанной в предыдущем абзаце, а используется исключительно из соображений эффективности. Такая реализация алгебраических величин вместо прямолинейного способа, описанного в предыдущей главе, часто может сэкономить до 50 % рабочего времени и даже больше в случаях, где имеется вложенность (пример 6 из приложения 2). Это объясняется тем, что в некоторых случаях прямолинейная реализация должна обрабатывать выражения несколько раз, чтобы устранить те степени внутренних квадратных корней, которые возникают из-за устранения внешних квадратных корней.

5. РАЗЛОЖЕНИЕ НА МНОЖИТЕЛИ

Разложение многочленов на множители важно для интегрирования и как составная часть работы с алгебраическими выражениями (см. разд. 6 ниже), и для определения полюсов и нулей функций. В последнем случае мы понимаем разложение на множители несколько необычно, так как следует предусмотреть расширение основного поля, чтобы получить все нули и полюса. Например, x^2-2 не раскладывается на множители над целыми числами, но $1/(x^2-2)$ имеет полюса в точках $\pm\sqrt{2}$.

Очевидное решение состоит в том, чтобы взять стандартную программу разложения многочленов от многих переменных над целыми числами (Wang, 1978), добавить метод Трейгера, чтобы распространить разложение на случай алгебраических величин ((Trager, 1976), см. также гл. 2), и предусмотреть расширение основного поля корнями любого многочлена, который не разлагается в текущем основном поле.

К сожалению, в REDUCE-2 нет пакета программ для разложения на множители. К счастью, я смог позаимствовать пакет, разработанный д-ром А. Ч. Норманом и миссис П. М. А. Мур в рамках их работы по интегрированию трансцендентных функций (Norman & Moore, 1977)[1]) и использовать его в своей программе интегрирования. Этот пакет не всегда раскладывает многочлены степени, большей чем 4, но всегда раскладывает многочлены от одной переменной. В тех

[1]) Я благодарен д-ру Норману и миссис Мур за разрешение использовать их модуль и помощь в его подключении к моим программам. Я признателен также д-ру Д. Даму из «Барроуз Корпорейшн», который внес несколько усовершенствований в этот пакет программ, и проф. А. Ч. Хёрну, который сообщил мне о них.

редких случаях, когда мне нужно было разложение многочленов более высокой степени, я либо раскладывал их на множители вручную, либо использовал систему MACSYMA (Bogen et al., 1977), чтобы получить разложение, а затем вставлял его вручную в программу интегрирования, основанную на системе REDUCE-2.

Я надеюсь, что скоро смогу использовать новый пакет программ разложения на множители, разрабатываемый теми же авторами (Norman, 1978b), который сможет разложить любые многочлены и будет во многих случаях работать значительно быстрее, чем нынешняя реализация.

6. ЕДИНСТВЕННОСТЬ АЛГЕБРАИЧЕСКИХ ВЫРАЖЕНИЙ

В гл. 2 объяснено, как важно, чтобы новые алгебраические выражения создавались только тогда, когда для рассматриваемой величины нет выражения в старых терминах, — в противном случае мы уже не будем иметь канонических выражений. Например, если SQRT(2) уже существует, то нам не разрешается создать SQRT(8), так как эту величину можно записать в виде 2*SQRT(2). Так как нынешняя реализация работает только с алгебраическими величинами, выразимыми через квадратные корни, то нужная программа оформлена в виде процедуры (SIMPSQRT), которая обеспечивает выполнение требуемых условий для любого SQRT-выражения.

К сожалению, задача усложняется из-за отмеченной в гл. 2 необходимости делать это «локально» (т. е. отдельно в каждом плейсе).

Для этого процедуре SIMPSQRT приходится поддерживать список всех SQRT, определенных в каждом плейсе, и при каждом вызове находить нужный ей список. Фактические вычисления в большой степени соответствуют описанию, данному в гл. 2 (некоторые из алгоритмов приведены в приложении 3), — это алгоритмы Трейгера (Trager, 1976), за которыми следует разложение на множители (см. выше). Так как эти операции могут оказаться чрезвычайно трудоемкими[1]), в пакете программ интегрирования имеется возможность отключить бóльшую часть этих проверок. Очевидно, что если программа выдает интеграл, не делая проверок, то полученное выражение (которое может содержать линейно зависимые алгебраические выражения) действительно является интегралом. Поэтому полная проверка необходима лишь в случае, когда программа считает, что без нее функция не интегрируема.

[1]) Проверка разложимости квадратичного выражения над полем, содержащим 4 независимых квадратных корня, включает разложение многочлена степени 32.

Приложение 2

ПРИМЕРЫ

Мы собрали здесь несколько примеров работы различных алгоритмов, описанных в этой монографии. Хотя все методы изложены во вполне стандартной математической форме, следует подчеркнуть, что с этими примерами справились машинные программы, работающие, по существу, в полном соответствии с описаниями, данными в этой монографии.

ПРИМЕР 1. ПРОСТОЕ ЛОГАРИФМИЧЕСКОЕ ВЫРАЖЕНИЕ

Рассмотрим задачу интегрирования функции $1/\sqrt{X^2-4}$. Подынтегральное выражение может иметь вычеты в точках ± 2 или на бесконечности. В то время как рассматриваемая функция от X имеет полюса в точках ± 2, подынтегральное выражение (рассматриваемое как дифференциал) их не имеет. Действительно, $X-2$ не является локальным параметром при $X=2$, и мы должны выразить дифференциал через $\sqrt{X-2}$ (обозначим его через Y), что дает $dY/\sqrt{Y^2+4}$, так как $dX = YdY$. В плейсах, лежащих над бесконечностью, ситуация обратная: функция не имеет полюса, а дифференциал его имеет. Действительно, X не является там локальным параметром, а $1/X$ (обозначим его через Z) является, и после замены дифференциал принимает вид $-dZ/Z*\sqrt{1-4Z^2}$ и имеет вычеты ± 1 в двух рассматриваемых плейсах.

Эти вычеты образуют 1-мерный **Z**-модуль, поэтому мы хотим найти функцию с нулями порядка ± 1 в двух плейсах, лежащих над бесконечностью. Так как алгоритм Коутса не решит эту задачу непосредственно, мы делим всю задачу на X и ищем функции с нулями порядка -1 в двух плейсах, лежащих над точкой 0, и нуль порядка 2 в одном из плейсов, лежащих над бесконечностью (и нуль порядка 0 в другом).

Исходный базис — это $\{1, \sqrt{X^2-4}\}$. Затем мы должны разделить каждый элемент базиса на X, чтобы получить целый базис, и в действительности — это нормальный целый базис. Тогда базис множества функций, имеющих полюса не хуже указанных (т. е. -1 в обоих плейсах, лежащих над нулем), есть $\{1, 1/X, \sqrt{X^2-4}/X\}$. Мы хотим найти ту линейную комбинацию этих функций, которая имеет нуль порядка 2

в одном из плейсов, лежащих над бесконечностью. Это — функция $(\sqrt{X^2-4}+X)/X$. Затем мы должны умножить это выражение на X (так как делили на X раньше) и получаем функцию с нужными нулями и полюсами. Это вычисление заняло 403 мс, из которых 80 мс было затрачено на умножение и деление на X для приведения задачи к виду, подходящему для применения алгоритма Коутса.

Мы имеем, таким образом, логарифмическую часть $\log(X+\sqrt{X^2-4})$, а ее производная равна исходной подынтегральной функции, так что задача решена. Общее время, затраченное на это интегрирование, равнялось 868 мс, включая все время, затраченное на алгоритм Коутса. Повторные запуски примерно в одно и то же время дня [1]) дают разницу ±3,5 % по затраченному времени центрального процессора. При замене числа 4 на другие полные квадраты затраченное время, по существу, не менялось, но замена 4 на 2 привела к увеличению времени примерно на 12 %. Это обстоятельство можно объяснить тем фактом, что программа должна учесть возможность вычетов в $\pm\sqrt{2}$, так что она должна построить алгебраическое выражение $\sqrt{2}$ и работать с ним. Это объяснение подтверждает и тот факт, что на шаги алгоритма Коутса было затрачено почти в точности то же время.

Если мы рассмотрим теперь $1/\sqrt{2X^2-1}$, то ход вычисления будет несколько иным, несмотря на то что интеграл имеет, по существу, тот же вид. Здесь вычеты равны $\pm\sqrt{2}/2$ и все разложения Пюизо содержат $\sqrt{2}$. Это усложнение привело к тому, что шаг алгоритма Коутса занял теперь 530 мс, а все интегрирование заняло 1590 мс, т. е. вдвое больше.

На стр. 134—136 мы приводим текст, выданный нашей программой (точнее, версии, которая печатает большое количество промежуточных результатов и наблюдений).

ПРИМЕР 2. 1/SQRT ((X**2 — 1)*(X**2 — K**2))

Это — типичный эллиптический интеграл, который, как хорошо известно, неинтегрируем в элементарных функциях, если $K \neq \pm 1$. Дифференциал $dX/\sqrt{(X^2-1)(X^2-K^2)}$ мог бы иметь полюса только в плейсах, лежащих над бесконечностью,

[1]) И тем самым, как мы надеемся, при примерно одинаковой загрузке машины. Это существенно для оценок времени, так как они получены из статистики операционной системы, а применяемые при этом методы подсчета времени иногда относят время обработки прерывания за счет задачи, которая выполнялась, когда произошло прерывание. Поэтому упомянутые оценки времени зависят до некоторой степени от загрузки машины.

```
INT(1/SQRT(X**2-4),X);
WITH 'NEW' FUNCTIONS :
((SQRT (PLUS (EXPT X 2) (MINUS 4))) X)
ПЛЕЙСЫ, В КОТОРЫХ МОГУТ БЫТЬ ПОЛЮСА
((QUOTIENT 1 X) (PLUS X 2) (PLUS X (MINUS 2)))
ДИФФЕРЕНЦИАЛ ПОСЛЕ ПЕРВОЙ ПОДСТАНОВКИ РАВЕН

                    2
( - 1)/(SQRT( - 4*X  + 1)*X)

ВЫЧЕТЫ В   ((X QUOTIENT 1 X))
РАВНЫ
((((X QUOTIENT 1 X)
     ((SQRT (PLUS (MINUS (TIMES 4 (EXPT X 2))) 1))
        SQRT
        (PLUS (MINUS (TIMES 4 (EXPT X 2))) 1)))
    -1
    . 1)
   (((X QUOTIENT 1 X)
       ((SQRT (PLUS (MINUS (TIMES 4 (EXPT X 2))) 1))
          MINUS
          (SQRT (PLUS (MINUS (TIMES 4 (EXPT X 2))) 1)))))
      1
      . 1))
ДИФФЕРЕНЦИАЛ ПОСЛЕ ПЕРВОЙ ПОДСТАНОВКИ РАВЕН

1/(SQRT(X - 4)*SQRT(X))

ДИФФЕРЕНЦИАЛ ПОСЛЕ ПЕРВОЙ ПОДСТАНОВКИ РАВЕН

1/(SQRT(X + 4)*SQRT(X))

НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА  :(-1 1)
 В
(((X QUOTIENT 1 X)
    ((SQRT (PLUS (MINUS (TIMES 4 (EXPT X 2))) 1))
       SQRT
       (PLUS (MINUS (TIMES 4 (EXPT X 2))) 1)))
   ((X QUOTIENT 1 X)
      ((SQRT (PLUS (MINUS (TIMES 4 (EXPT X 2))) 1))
         MINUS
         (SQRT (PLUS (MINUS (TIMES 4 (EXPT X 2))) 1)))))
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА :(-1 -1 2)
 В
(((X . X)
```

```
    ((SQRT (PLUS (EXPT X 2) (MINUS 4)))
       SQRT
       (PLUS (EXPT X 2) (MINUS 4))))
   ((X . X)
      ((SQRT (PLUS (EXPT X 2) (MINUS 4)))
         MINUS
         (SQRT (PLUS (EXPT X 2) (MINUS 4)))))
   ((X QUOTIENT 1 X)
      ((SQRT (PLUS (MINUS (TIMES 4 (EXPT X 2))) 1))
         MINUS
         (SQRT (PLUS (MINUS (TIMES 4 (EXPT X 2))) 1)))))
КОРНИ НА ЭТОЙ КРИВОЙ:
((SQRT (PLUS (EXPT X 2) (MINUS 4)))
   (SQRT (PLUS (MINUS (TIMES 4 (EXPT X 2))) 1)))
ИСХОДНЫЙ БАЗИС ПРОСТРАНСТВА M(X)

      2
SQRT(X  - 4)

1

РЕДУКЦИЯ К ЦЕЛОМУ БАЗИСУ В
((X . X))
МАТРИЦА ПЕРЕД ШАГОМ РЕДУКЦИИ:
%<(((((SQRT -1) . 1) . 2)) . 1),(((((SQRT -1) . 1) . -2)) . 1)%>
%<(1 . 1),(1 . 1)%>
РЕДУКЦИЯ К НОРМАЛЬНОМУ ЦЕЛОМУ БАЗИСУ
 СО СЛЕДУЮЩИМИ КОРНЯМИ ЛЕЖАЩИМИ НА БЕСКОНЕЧНОСТИ:
((SQRT (PLUS (MINUS (TIMES 4 (EXPT X 2))) 1)))
ВЫЧИСЛЕНИЕ
            2
SQRT( - 4*X  + 1)

%<0,0%>
%<(1 . 1),(-1 . 1)%>
ВЫЧИСЛЕНИЕ
X

%<1,1%>
%<(1 . 1),(1 . 1)%>
МАТРИЦА ПЕРЕД ШАГОМ РЕДУКЦИИ НА БЕСКОНЕЧНОСТИ РАВНА:
%<(1 . 1),(1 . 1)%>
%<(1 . 1),(-1 . 1)%>
БАЗИС ФУНКЦИИ, ИМЕЮЩИХ В ТОЧНОСТИ НУЖНЫЕ ПОЛЮСА

      2
SQRT(X  - 4)/X
```

```
1

1/X

НУЖНО РЕШИТЬ УРАВНЕНИЯ:
СТРОКА НОМЕР 0
 - 1

1

0

СТРОКА НОМЕР 1
0

0

1

СТРОКА НОМЕР 2
1

1

1

ОТВЕТ, ПОЛУЧЕННЫЙ ПОСЛЕ РЕШЕНИЯ ЛИНЕЙНЫХ УРАВНЕНИЙ

        2
(SQRT(X  - 4) + X)/X

(ВРЕМЯ РАБОТЫ АЛГОРИТМА КОУТСА 396 МИЛЛИСЕКУНД)
ЛОГАРИФМ ПОСЛЕ РАСШИРЕНИЯ РАВЕН

       2
SQRT(X  - 4) + X

ВНУТРЕННЯЯ РАБОТА ДАЕТ

          2
LOG(SQRT(X  - 4) + X)

С ПРОИЗВОДНОЙ

       2          2                  2                   3
(SQRT(X  - 4)*X + X  - 4)/(lastsqrt*X  - 4*lastsqrt + X  - 4*X)

(ЗАТРАЧЕННОЕ ВРЕМЯ 868 МИЛЛИСЕКУНД)

          2
LOG(SQRT(X  - 4) + X)
```

± 1 и $\pm K$. Рассматривая сначала плейс над бесконечностью, положим $Y = 1/X$, так что Y — локальный параметр на бесконечности. Тогда подынтегральное выражение принимает вид $(-dY/Y^2)/\sqrt{(Y^{-2}-1)(Y^{-2}-K^2)} = -dY/\sqrt{(1-Y^2)(1-K^2Y^2)}$, а это выражение очевидным образом конечно при $Y = 0$ (т. е. при бесконечном X). В каждом из остальных плейсов (т. е. корней радикала) обозначим X-корень через Y. Тогда подынтегральное выражение равно $dY/\sqrt{Y\Pi}$, где Π — произведение остальных 3 членов. Π не имеет ни нулей, ни полюсов, при $Y = 0$, так как все корни различны (чем и объясняется то, что $K = \pm 1$ — особый случай). Положим теперь $Z^2 = Y$, так что Z — локальный параметр. (Y не будет локальным параметром, так как в наше выражение входит $\sqrt{Y}$.) Подынтегральное выражение примет тогда вид $(2Z\,dZ)/(Z\sqrt{\text{произведение}}) = 2dZ\sqrt{\text{произведение}}$ и тем самым не имеет полюса в $Z = 0$.

Так как подынтегральное выражение не имеет полюсов, то оно не имеет и вычетов, а потому интеграл (если он элементарен) не имеет логарифмической части, т. е. должен быть чисто алгебраическим. Эта алгебраическая функция не должна иметь конечных полюсов (иначе их имела бы и ее производная, а мы только что показали, что это не так) и должна иметь полюс порядка не выше 1 на бесконечности. Следовательно, интеграл (если он элементарен) имеет вид $a + bX$, и, дифференцируя, легко усмотреть, что этого не может быть. Поэтому рассматриваемая функция не имеет элементарного интеграла.

ПРИМЕР 3. ПРИМЕР С КРУЧЕНИЕМ

Рассмотрим функцию $3YX^2/2(Y+1)(X^3+1)$, где $Y = \sqrt{X^3+1}$. Она — производная от $\mathrm{LOG}(1+Y)$ и была выбрана, чтобы показать, что тривиальные на вид интегралы могут потребовать изощренного аппарата.

Подынтегральное выражение могло бы иметь вычеты на бесконечности, в 0 и в любом из корней уравнения $Y^2 = X^3 + 1 = 0$. В действительности оно имеет вычет -3 на бесконечности, которая является точкой ветвления с индексом ветвления 2, и имеет вычет 3 в одном из плейсов, лежащих над 0, но не в другом. На локально несингулярной модели, где бесконечность не является точкой ветвления, имеются два плейса, лежащие над бесконечностью, в одном из которых подынтегральное выражение имеет вычет $-3/2$. Следовательно, мы можем рассмотреть r_1, равное $3/2$, и имеем дивизор D, содержащий один из плейсов, лежащих

над 0 (и обозначаемый в дальнейшем через 0+), с кратностью 2 и один разветвленный плейс на бесконечности с кратностью —1. В терминах несингулярной модели (на кривой $Y^2 = X^6 + 1$) мы имеем 0+ с кратностью 2 и два плейса, лежащих над бесконечностью (обозначим их через $I+$ и $I-$), с кратностью —1 каждый.

В действительности дивизор D не является линейно эквивалентным нулю в силу алгоритма Коутса, так как единственная функция, имеющая полюса порядка —1 в $I+$ и $I-$, но не имеющая никаких других полюсов, — это X, а ее, очевидно, нельзя заставить иметь нуль порядка 2 в 0+. Мы не можем использовать алгоритм Коутса непосредственно, так как ищем функции с полюсами на бесконечности. Мы должны сначала рассмотреть задачу нахождения функций с полюсом порядка 1 в 0— и нулем порядка 1 в 0+, а затем умножить эти функции на X, чтобы снова получить функции, имеющие те же нули и полюса, что и дивизор D.

Так как $Y^2 = X^3 + 1$ — эллиптическая кривая (она имеет один линейно независимый дифференциал первого рода, а именно $1/Y$), определенная над рациональными числами, и так как D определен над рациональными числами, мы можем использовать границу Мазура [1]) на кручение эллиптических кривых над рациональными числами (см. гл. 5). Следовательно, мы должны перебирать $2D$, $3D$, ..., $10D$, $12D$, пока не наткнемся на кратное, являющееся дивизором некоторой функции. $2D$ не годится, но $3D$ действительно является дивизором функции $2Y + X^3 + 2$.

Следовательно, логарифмическая часть равна

$$\frac{3}{2}\,\frac{1}{3}\log(2Y + X^3 + 2) = \frac{\log(2Y + X^3 + 2)}{2}.$$

Если мы продифференцируем это выражение, то обнаружим, что результат равен исходной интегрируемой функции, а значит, алгебраической части нет. Заметим, что в конце у нас получилось выражение, не вполне совпадающее с функцией, которую мы дифференцировали в начале, но в действительности эти два выражения представляют одну и ту же функцию. Это вычисление заняло 5191 мс на кембриджской

[1]) Заметим, что для аналогичной задачи с $Y^2 = X^3 + 8$, которую легко можно преобразовать в рассматриваемую, не выполнены условия применимости границы Мазура, так как в этом случае дивизор уже не будет определен над полем рациональных чисел, хотя кривая все еще определена над этим полем. Нам пришлось бы расширить наше поле констант квадратным корнем из 2. Такую задачу можно решить с помощью алгоритма ЛЮТЦ-НАГЕЛЬ (см. гл. 7), или мы можем попытаться найти оценку кручения этой эллиптической кривой над расширенным полем. Этот частный случай рассмотрен в гл. 8 как пример работы над полями алгебраических чисел.

машине IBM 370/165 (плюс 3.97 с на сборку мусора), из которых на 3 шага алгоритма Коутса приходится соответственно 400, 403 и 530 мс.

ПРИМЕР 4. МОДУЛЯРНАЯ КРИВАЯ $X(7)$

Как отмечено в работе (Tate, 1974, с. 198), точка (0, 0) имеет порядок 7 на кривой $y^2+(1+d-d^2)xy+(d^2-d^3)y=x^3+(d^2-d^3)x^2$, которая будет эллиптической, если ее дискриминант $d^7(d-1)^7(d^3-8d^2+5d+1)$ отличен от нуля. Мы можем преобразовать ее к более стандартному виду, и тогда получим утверждение, что точка $P=(0,\ d^3-d^2)$ имеет порядок 7 на кривой $y^2=x^3+x^2(1+2d+3d^2-6d^3+d^4)+4xd^2(1-4d^2+2d^3)+16d^4(1-d)^2$, которая почти находится в канонической форме Вейерштрасса. Точка $-2P$ есть $(d(d-1),\ d(d-1)^3)$, и, как показывают элементарные теоретико-групповые рассуждения, она тоже имеет порядок 7. Дивизор, состоящий из этой точки с кратностью -7 и бесконечности с кратностью 7, является поэтому главным и соответствует функции

$$G(x,y)=x^{-7}$$
$$\{yd^2+2y\,dx+yx^2+yx+d^5+d^4(3x-1)+d^3(3x^2-2x)+$$
$$+d^2(x^3-4x^2-2x)+d(-3x^3-3x^2)-3x^3-x^2)\}$$

что можно заметить, если применить к этой точке алгоритм Коутса [1]).

Мы поэтому хотим рассмотреть интегрирование производной от $\log G$, которая (вычислена системой [2]) REDUCE-2) равна

---

[1]) Это — еще один пример значительных вычислительных трудностей, окружающих эту область. Применение алгоритма Коутса для нахождения функций, имеющих нужные полюса, потребовало 417 с (+184 с на сборку мусора) при оперативной памяти 650K на кембриджской IBM 370/165. Решение системы (линейных уравнений 14 на 7) для получения еще и нужных нулей не только не закончилось за следующие 517 с, но и заполнило всю имевшуюся память числами порядка 10^{2400} — коэффициентами промежуточных выражений. Переписанная версия программы решения линейных уравнений, которая сначала искала разреженные столбцы и специальным образом обрабатывала их, смогла решить линейные уравнения, по существу, за время 0. Это наблюдение связано с рассуждениями из статьи (Wang & Minamikawa, 1976), где сравнивались бездробные методы типа Барейса (Lipson, 1969), аналогичные предыдущей реализации, и методы, которые ищут разреженную структуру.

[2]) Версия системы REDUCE-2, на которой основана описываемая здесь система интегрирования, имеет локальную модификацию (см. приложение 1), в результате которой выражение «lastsqrt» печатается вместо любого квадратного корня SQRT, который совпадает с непосредственно предшествующим SQRT в том же самом выражении. Пусть читатель представит себе, как выглядело бы приводимое ниже выражение, если бы эта модификация не была введена.

```
           6      5         5    4  2      4       4       3  2
( - 7*SQRT(D  + 2*D *X - 2*D  + D *X  - 4*D *X + D  - 6*D *X

      2  2      2           2       3    2   8
 + 3*D *X  + 2*D *X + 2*D*X  + 4*X  + X )*D  - 25*lastsqrt*

  7                   7                  6  2                   6
D *X + 14*lastsqrt*D  - 33*lastsqrt*D *X  + 44*lastsqrt*D *X

                 6                   5  3                  5  2
 - 7*lastsqrt*D  - 19*lastsqrt*D *X  + 79*lastsqrt*D *X  - 4

           4  4                  4  3              4  2
*lastsqrt*D *X  + 73*lastsqrt*D *X  - lastsqrt*D *X  - 19*

          4                    3  4                 3  2
lastsqrt*D *X + 24*lastsqrt*D *X  - 34*lastsqrt*D *X  - 12*

          2  4                 2  3                 2  2
lastsqrt*D *X  - 66*lastsqrt*D *X  - 17*lastsqrt*D *X  - 44*
```

```
            4                     3                   5
lastsqrt*D*X  - 20*lastsqrt*D*X  - 14*lastsqrt*X  - 22*

          4                 3       11        10          10
lastsqrt*X  - 5*lastsqrt*X  - 7*D   - 32*D  *X + 21*D   - 58

  9  2       9          9       8  3        8  2       8
*D *X  + 90*D *X - 21*D  - 52*D *X  + 195*D *X  - 58*D *X +

   8       7  4        7  3        7  2       7        6  5
7*D  - 23*D *X  + 240*D *X  - 127*D *X  - 26*D *X - 4*D *X

        6  4        6  3       6  2       6         5  5
 + 150*D *X  - 180*D *X  - 82*D *X  + 26*D *X + 36*D *X  -

     5  4        5  3       5  2       4  5        4  4
174*D *X  - 192*D *X  + 36*D *X  - 72*D *X  - 183*D *X  +

     4  3       4  2        3  5        3  4       3  3
102*D *X  + 36*D *X  - 104*D *X  + 112*D *X  + 66*D *X  - 16

  2  6        2  5        2  4       2  3         6
*D *X  + 136*D *X  + 137*D *X  + 22*D *X  + 48*D*X  + 96*D*
```

```
 5          4         6         5         4                    8
X  + 25*D*X   + 48*X   + 32*X   + 5*X )/(X*(lastsqrt*D   + 4*

              7                   7                  6  2
      lastsqrt*D *X - 2*lastsqrt*D  + 6*lastsqrt*D *X  - 7*

              6                   6                  5  3
      lastsqrt*D *X + lastsqrt*D  + 4*lastsqrt*D *X  - 14*

            5  2                4  4                  4  3
      lastsqrt*D *X  + lastsqrt*D *X  - 15*lastsqrt*D *X  + 3

             4                   3  4                  3  2
      *lastsqrt*D *X - 6*lastsqrt*D *X  + 6*lastsqrt*D *X  +

               2  4                  2  3                  2  2
      3*lastsqrt*D *X  + 13*lastsqrt*D *X  +. 3*lastsqrt*D *X

                        4                   3                   5
       + 10*lastsqrt*D*X  + 4*lastsqrt*D*X  + 4*lastsqrt*X

                     4              3     11       10          10
       + 5*lastsqrt*X  + lastsqrt*X  + D   + 5*D  *X - 3*D

           9  2        9          9        8  3        8  2
     + 10*D *X  - 14*D *X + 3*D  + 10*D *X  - 33*D *X  + 9*

    8       8       7  4        7  3        7  2       7       6
   D *X - D   + 5*D *X  - 45*D *X  + 21*D *X  + 4*D *X + D

     5        6  4        6  3        6  2       6         5  5
   *X  - 32*D *X  + 33*D *X  + 14*D *X  - 4*D *X - 9*D *X

           5  4        5  3        5  2        4  5        4  4
     + 36*D *X  + 35*D *X  - 6*D *X  + 18*D *X  + 37*D *X

           4  3        4  2        3  5        3  4        3  3
     - 18*D *X  - 6*D *X  + 23*D *X  - 22*D *X  - 12*D *X

           2  6        2  5        2  4        2  3         6
     + 4*D *X  - 30*D *X  - 27*D *X  - 4*D *X  - 12*D*X  -

          5         4        6        5     4
    21*D*X  - 5*D*X  - 12*X  - 7*X  - X ))
```

К сожалению, это огромное подынтегральное выражение потребовало бóльших ресурсов, чем было доступно. Вычисление вычетов[1]) заняло 34 с, начальный шаг по алгоритму

[1]) Все эти замеры времени были сделаны на установленнной в Кембридже машине IBM 370/165 с оперативной памятью 700K. Сюда не включено время сборки мусора, которое дает в среднем дополнительно 20 %.

Коутса (т. е. проверка, является ли дивизор главным) занял 6 с, вычисление дифференциалов первого рода заняло 2 с[1]), второй шаг по алгоритму Коутса (т. е. обработка дивизора $2D$) занял 18 с и третий шаг алгоритма Коутса исчерпал все отведенное время через 200 с.

В частном случае $d = 2$ рассматривается точка (0, 4), имеющая порядок 7 на кривой $y^2 = 4x^3 - 15x^2 + 8x + 16$. Следовательно, дивизор, состоящий из плейса, отвечающего этой точке P, с кратностью -1 и плейса над бесконечно удаленной точкой O с кратностью 1 — это дивизор порядка 7. Мы применяем алгоритм Коутса, чтобы найти функцию, соответствующую этому дивизору, умноженному на 7, — это функция $G(x, y) = (-yx^2 - 5yx - 4y + 5x^3 - x^2 - 24x - 16)/x^7$, имеющая полюс порядка 7 в точке P и нуль порядка 7 в этом плейсе над бесконечно удаленной точкой (один разветвленный плейс имеется над бесконечностью).

Затем мы можем продифференцировать эту функцию и попытаться снова проинтегрировать ее, чтобы проверить правильность нашей алгебраической геометрии хотя бы для кривых рода 1. Отметим, что кривая эллиптична над рациональными числами, так что мы можем применить оценку Мазура на кручение. В действительности мы можем также применить тест Кэли, использующий определители, и, как уже сказано в гл. 5, это приводит к существенной экономии общего времени на вычисление этого интеграла, которое все еще составляет весомые 40 с центрального процессора (плюс 20 с на сборку мусора при работе в оперативной памяти 550K байт на IBM 370/165).

ПРИМЕР 5. ИНТЕГРАЛ ЧЕБЫШЕВА

В своей статье (Chebyshev, 1857) о цепных дробях и интегрировании псевдоэллиптических функций Чебышев рассмотрел следующий интеграл:

$$\int \frac{2X^6 + 4X^5 + 7X^4 - 3X^3 - X^2 - 8X - 8}{(2X^2 - 1)^2 \sqrt{X^4 + 4X^3 + 2X^2 + 1}}\, \mathrm{d}X$$

Подынтегральное выражение имеет 4 вычета, равные 5/2 (но при наивном вычислении получится выражение SQRT(4 * SQRT(2) + 9) * (10 * SQRT(2) — 5)/14, если не заметить, что 4 * SQRT(2) + 9 является полным квадратом в $Q(\sqrt{2})$; подробности этих задач см. в гл. 2). Оказывается,

[1]) При этом запуске задачи не исполнялся алгоритм НАЙТИ ПОРЯДОК—ПО—МАНИНУ, так как вычисление было проведено раньше (см. гл. 6) и иначе сделало бы вычисление в целом недопустимо дорогостоящим.

что нужный дивизор имеет порядок 5, и $5D$ — это дивизор функции

$$\frac{A(X)\,Y - B(X)}{C(X)}$$

где $A(X) = 1023X^8 + 4104X^7 + 5084X^6 + 2182X^5 + 805X^4 + 624X^3 + 10X^2 + 28X$,

$B(X) = 1025X^{10} + 6138X^9 + 12\,307X^8 + 10\,188X^7 + 4503X^6 + 3134X^5 + 1589X^4 + 140X^3 + 176X^2 + 2$,

$C(X) = 32X^{10} - 80X^8 + 80X^6 - 40X^4 + 10X^2 - 1$, и

$$Y = \sqrt{X^4 + 4X^3 + 2X^2 + 1}$$

Хотя кривая, определенная координатами X и Y, является эллиптической и определяющее уравнение имеет рациональные коэффициенты, мы не можем применить оценку Мазура, так как вычеты подынтегрального выражения (а значит, и рассматриваемый дивизор) лежат над $X = 2^{-1/2}$ и $X = -2^{-1/2}$, а значит, дивизор не определен над рациональными числами. Когда теоретические положения гл. 7 еще не были реализованы, еще не было программы, находящей границу кручения, поэтому нам пришлось удовлетвориться частичным вычислением, которое оканчивается и дает интеграл, если функция интегрируема, но может никогда не закончиться, если она не интегрируема.

Поэтому мы перебирали по очереди всевозможные порядки дивизора, и вычисления показали, что D, $2D$, $3D$, $4D$ не являются линейно эквивалентными нулю, а $5D$ является. Весь этот процесс (не использовавший метода Кэли, описанного в гл. 5) занял 75.6с (+38.5с сборки мусора) на кембриджской IBM 370/165 в оперативной памяти 650K байт. Интересно отметить, что 30 с из этого времени было затрачено на решение линейных уравнений (см. также подстрочное примечание к предыдущему примеру), определяющих линейную комбинацию элементов нашего нормального целого базиса, имеющую нужные нули.

Это дает нам решение в виде

$$\frac{(2X+1)\,Y}{2(2X^2-1)} + \frac{\log\left(\dfrac{A(X)\,Y - B(X)}{C(X)}\right)}{2}$$

где A, B, C определены выше и $Y^2 = X^4 + 4X^3 + 2X^2 + 1$. Заметим, что решение самого Чебышева имеет ту же алгебраическую часть, что и наше, но его логарифмическая часть

имеет коэффициент 1/4, и, следовательно, аргумент логарифма равен, по существу, квадрату нашего аргумента.

Когда был реализован алгоритм ЛЮТЦ_НАГЕЛЬ из гл. 7, было быстро обнаружено, что точка, которая была суммой членов дивизора, имела порядок 5 (так как $4P = -P$), и поэтому нам нужно было проверять только $5D$. Интересно отметить, что это вычисление заняло значительно больше времени, чем до включения процедуры ЛЮТЦ_НАГЕЛЬ, хотя и непонятно, до какой степени это просто отражает различие в объеме настройки, которая потребовалась для этих двух программ. Разумеется, добавление процедуры ЛЮТЦ_НАГЕЛЬ дает уверенность в том, что если дивизор имеет бесконечный порядок, то мы об этом узнаем.

Этот пример был снова пропущен на версии системы REDUCE от 15 апреля 1979 г., которая позволяет печатать сокращенные обозначения подвыражений, и мы прилагаем текст, отпечатанный при этом запуске[1]), чтобы продемонстрировать работу программы в сложном случае. Здесь был установлен более низкий уровень печати сообщений, чем в примере 1, приведенном выше, так как иначе результаты были бы еще длиннее, чем теперь. Мы приводим сначала текст, напечатанный при запуске программы, в которой шаг ДИВИЗОР_В_ФУНКЦИЮ был реализован в виде вызова алгоритма Коутса с заданными полюсами, за которыми следовала реализация всех нулей в виде линейных ограничений.

```
INT((2*X**6+4*X**5+7*X**4-3*X**3-X**2-8*X-8)*(2*X**2-1)**(-2)/
    SQRT(X**4+4*X**3+2*X**2+1),X);
ПЛЮСЫ, В КОТОРЫХ МОГУТ БЫТЬ ПОЛЮСА
X=БЕСКОНЕЧНОСТЬ
X=-1
X=ЛЮБОЙ КОРЕНЬ ФУНКЦИИ
    3      2
 - X  - 3*X  + 2*X - 1

X=
( - SQRT(2))/2

X=
SQRT(2)/2

НОВОЕ МНОЖЕСТВО ВЫЧЕТОВ СОСТОИТ ИЗ
( - 5)/2

5/2

( - 5)/2
```

[1]) С использованием 700K байтов оперативной памяти он занял 105 с плюс дополнительные 75 с. на сборку мусора и загрузку программы.

```
5/2
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА    ;(1 -1 1 -1)
ОКАЗАЛОСЬ НЕВОЗМОЖНЫМ103
(ВРЕМЯ АЛГОРИТМА КОУТСА 4020 МИЛЛИСЕКУНД)
НАЙТИ ДИФФЕРЕНЦИАЛЫ ПЕРВОГО РОДА НА КРИВОЙ, ОПРЕДЕЛЕННОЙ МНОГОЧЛЕНОМ
 4       3       2
X  + 4*X  + 2*X  + 1
ДИФФЕРЕНЦИАЛЫ РАВНЫ:
1/ANS1
  ГДЕ
                    4       3       2
     ANS1 := SQRT(X  + 4*X  + 2*X  + 1)
НАЙТИ ФОРМУ ВЕЙЕРШТРАССА ДЛЯ ЭЛЛИПТИЧЕСКОЙ КРИВОЙ, ОПРЕДЕЛЕННОЙ МНОГОЧЛЕНОМ:
ANS1
  ГДЕ
                    4       3       2
     ANS1 := SQRT(X  + 4*X  + 2*X  + 1)
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА : (-3)
ФУНКЦИЯ РАВНА
                     3       2   3
(2*ANS1*X + ANS1 + 2*X  + 5*X )/X
  ГДЕ
                    4       3       2
     ANS1 := SQRT(X  + 4*X  + 2*X  + 1)
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА : (-2)
ФУНЦИЯ РАВНА
       2         2
(ANS1 + X  + 2*X)/X
  ГДЕ
                    4       3       2
     ANS1 := SQRT(X  + 4*X  + 2*X  + 1)
СТАНДАРТНАЯ ФОРМА ИМЕЕТ ВИД Y**2 =
 3
X  - 12*X + 38
ТОЧКА, ИССЛЕДУЕМАЯ НА КРУЧЕНИЕ,
(-2)

 - 3*SQRT(2)*SQRT(3)
ТОЧКА ИМЕЕТ ПОРЯДОК - ДЕЛИТЕЛЬ ЧИСЛА 5
ТОЧКА В ДЕЙСТВИТЕЛЬНОСТИ ИМЕЕТ ПОРЯДОК 5
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА : (5 -5 5 -5)
(ВРЕМЯ АЛГОРИТМА КОУТСА 72940 МИЛЛИСЕКУНД)
ДИВИЗОР ИМЕЕТ ПОРЯДОК 5
ЛОГАРИФМ РАСШИРЕНИЯ РАВЕН
AN61
  ГДЕ
                                        4       3       2       8
     ANS1 := NTHROOT((1023*SQRT(X  + 4*X  + 2*X  + 1)*X  + 4104*
        4       3       2       7                4       3       2
   SQRT(X  + 4*X  + 2*X  + 1)*X  + 5084*SQRT(X  + 4*X  + 2*X  + 1)*
    6               4       3       2       5               4       3
   X  + 2182*SQRT(X  + 4*X  + 2*X  + 1)*X  + 805*SQRT(X  + 4*X  + 2
     2       4               4       3       2       3               4
   *X  + 1)*X  + 624*SQRT(X  + 4*X  + 2*X  + 1)*X  + 10*SQRT(X  + 4
     3       2       2               4       3       2                10
   *X  + 2*X  + 1)*X  + 28*SQRT(X  + 4*X  + 2*X  + 1)*X - 1025*X
```

```
          9          8          7          6          5          4
   - 6138*X  - 12307*X  - 10188*X  - 4503*X  - 3134*X  - 1589*X
         3        2          10        8        6        4        2
   - 140*X  - 176*X  - 2)/(32*X   - 80*X  + 80*X  - 40*X  + 10*X
   - 1),5)
```

ВНУТРЕННЯЯ РАБОТА ДАЕТ

```
( - ANS1 + ANS2)/2
```

ГДЕ

```
                                  4       3       2       8                4
     ANS2 := LOG(1023*SQRT(X  + 4*X  + 2*X  + 1)*X  + 4104*SQRT(X
     3       2       7                 4       3       2       6
+ 4*X  + 2*X  + 1)*X  + 5084*SQRT(X  + 4*X  + 2*X  + 1)*X  + 2182*
     4       3       2       5                 4       3       2       4
SQRT(X  + 4*X  + 2*X  + 1)*X  + 805*SQRT(X  + 4*X  + 2*X  + 1)*X
          4       3       2       3                 4       3       2
 + 624*SQRT(X  + 4*X  + 2*X  + 1)*X  + 10*SQRT(X  + 4*X  + 2*X  + 1
   2          4       3       2                 10          9
)*X  + 28*SQRT(X  + 4*X  + 2*X  + 1)*X - 1025*X   - 6138*X  - 12307
  8         7         6         5         4         3         2
*X  - 10188*X  - 4503*X  - 3134*X  - 1589*X  - 140*X  - 176*X  - 2)
                    10        8        6        4        2
     ANS1 := LOG(32*X   - 80*X  + 80*X  - 40*X  + 10*X  - 1)
```

С ПРОИЗВОДНОЙ

```
                12                  11                  10
( - 6150*ANS1*X   - 41953*ANS1*X   - 111707*ANS1*X   - 165629*ANS1*
  9                 8                 7                 6
 X  - 164107*ANS1*X  - 112635*ANS1*X  - 56725*ANS1*X  - 30723*ANS1*
  5                 4                 3                 2
 X  - 12879*ANS1*X  - 1860*ANS1*X  - 1244*ANS1*X  - 10*ANS1*X - 14*
              14          13          12          11          10
 ANS1 + 6138*X   + 54291*X   + 189417*X   + 359458*X   + 442796*X
           9         8         7         6         5
  + 392574*X  + 261828*X  + 153172*X  + 87512*X  + 33251*X  + 10019
    4         3         2                   14                  13
 *X  + 4978*X  + 210*X  + 196*X)/(2046*ANS1*X   + 16392*ANS1*X   +
                12                  11                  10                  9
   46069*ANS1*X   + 53256*ANS1*X   + 17902*ANS1*X   - 6102*ANS1*X
                 8                 7                 6                 5
    - 2324*ANS1*X  - 5316*ANS1*X  - 7326*ANS1*X  - 2138*ANS1*X  -
              4                 3                 2                          16
   917*ANS1*X - 624*ANS1*X  - 10*ANS1*X  - 28*ANS1*X - 2050*X   -
         15          14          13          12          11
   20476*X   - 76793*X   - 133146*X   - 102879*X   - 23628*X   + 18
     10        9         8         7         6         5
   *X   + 1756*X  + 18604*X  + 14716*X  + 4531*X  + 3822*X  + 1583*
    4         3         2
   X  + 148*X  + 176*X  + 2)
```

```
ГДЕ
                    4      3      2
     ANS1 := SQRT(X  + 4*X  + 2*X  + 1)
ОБНАРУЖЕННЫЕ ПОТЕНЦИАЛЬНЫЕ СОКРАЩЕНИЯ
(ЗАТРАЧЕННОЕ ВРЕМЯ 108473 МИЛЛИСЕКУНД)
              10       8       6       4       2       2
( - 2*LOG(32*X   - 80*X  + 80*X  - 40*X  + 10*X  - 1)*X  + LOG(32*
  10       8       6       4       2                   4
X   - 80*X  + 80*X  - 40*X  + 10*X  - 1) + 2*LOG(1023*SQRT(X  + 4*
  3       2       8                   4       3       2       7
X  + 2*X  + 1)*X  + 4104*SQRT(X  + 4*X  + 2*X  + 1)*X  + 5084*SQRT
    4       3       2       6                   4       3       2       5
 (X  + 4*X  + 2*X  + 1)*X  + 2182*SQRT(X  + 4*X  + 2*X  + 1)*X  +
             4       3       2       4                   4       3       2
 805*SQRT(X  + 4*X  + 2*X  + 1)*X  + 624*SQRT(X  + 4*X  + 2*X  + 1)
   3             4       3       2       2             4       3       2
 *X  + 10*SQRT(X  + 4*X  + 2*X  + 1)*X  + 28*SQRT(X  + 4*X  + 2*X
                  10          9          8          7          6
  + 1)*X - 1025*X   - 6138*X  - 12307*X  - 10188*X  - 4503*X  -
       5          4          3          2          2                   4
 3134*X  - 1589*X  - 140*X  - 176*X  - 2)*X  - LOG(1023*SQRT(X  + 4
    3       2       8                   4       3       2       7
 *X  + 2*X  + 1)*X  + 4104*SQRT(X  + 4*X  + 2*X  + 1)*X  + 5084*
       4       3       2       6                   4       3       2       5
 SQRT(X  + 4*X  + 2*X  + 1)*X  + 2182*SQRT(X  + 4*X  + 2*X  + 1)*X
              4       3       2       4                   4       3       2
  + 805*SQRT(X  + 4*X  + 2*X  + 1)*X  + 624*SQRT(X  + 4*X  + 2*X
         3             4       3       2       2             4       3
  + 1)*X  + 10*SQRT(X  + 4*X  + 2*X  + 1)*X  + 28*SQRT(X  + 4*X  +
    2                   10          9          8          7          6
 2*X  + 1)*X - 1025*X   - 6138*X  - 12307*X  - 10188*X  - 4503*X
          5          4          3          2                   4       3
  - 3134*X  - 1589*X  - 140*X  - 176*X  - 2) + 2*SQRT(X  + 4*X  + 2
    2                   4       3       2                   2
 *X  + 1)*X + SQRT(X  + 4*X  + 2*X  + 1))/(2*(2*X  - 1))
```

Когда этот пример пропустили с использованием алгоритма ДИВИЗОР_В_ФУНКЦИЮ, описанного в гл. 3, были получены ответы совсем другого вида. Текст, отпечатанный при этом запуске, приведен ниже, причем заметно, что, хотя время работы алгоритма Коутса над дивизором 5*D* сильно уменьшилось (примерно до 37 % старой величины), общее время увеличилось примерно на 12,3 %, что можно с некоторой натяжкой объяснить случайными флуктуациями. Это вы-

звано более сложным видом логарифмической части, из-за чего ее дифференцирование заняло 10 с, а ее вычитание из первоначального интеграла (включавшее приведение к общему знаменателю) заняло 20 с. Вид этой логарифмической части очень интересен, хотя и неясно, лучше ли он, чем один громадный логарифм, полученный от предыдущего запуска.

```
Y:=(2*X**6+4*X**5+7*X**4-3*X**3-X*X-8*X-8)/
    ((2*X**2-1)**2*SQRT(X**4+4*X**3+2*X**2+1));
          6      5      4      3     2                          4       3
Y := (2*X  + 4*X  + 7*X  - 3*X  - X  - 8*X - 8)/(SQRT(X  + 4*X  + 2
           2           4       2
         *X  + 1)*(4*X  - 4*X  + 1))
INT(Y,X);
ПЛЕЙСЫ, В КОТОРЫХ МОГУТ БЫТЬ ПОЛЮСА
X=бесконечность
X=-1
X=любой корень функции
    3      2
 - X  - 3*X  + 2*X - 1

X=
( - SQRT(2))/2

X=
SQRT(2)/2

НОВОЕ МНОЖЕСТВО ВЫЧЕТОВ СОСТОИТ ИЗ
( - 5)/2

5/2

( - 5)/2

5/2
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА :(1 1 -1 -1)
ОКАЗАЛОСЬ НЕВОЗМОЖНЫМ103
(ВРЕМЯ АЛГОРИТМА КОУТСА 4403 МИЛЛИСЕКУНДЫ)
НАЙТИ ДИФФЕРЕНЦИАЛЫ ПЕРВОГО РОДА НА КРИВОЙ, ОПРЕДЕЛЕННОЙ МНОГОЧЛЕНОМ:
 4      3      2
X  + 4*X  + 2*X  + 1
ДИФФЕРЕНЦИАЛЫ РАВНЫ:
1/ANS1
    ГДЕ
                    4      3      2
      ANS1 := SQRT(X  + 4*X  + 2*X  + 1)
НАЙТИ ФОРМУ ВЕЙЕРШТРАССА ДЛЯ ЭЛЛИПТИЧЕСКОЙ КРИВОЙ, ОПРЕДЕЛЕННОЙ МНОГОЧЛЕНОМ
```

```
ANS1
   ГДЕ
                        4       3       2
      ANS1 := SQRT(X  + 4*X  + 2*X  + 1)
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА  :(-3)
ФУНКЦИЯ РАВНА
                         3       2   3
(2*ANS1*X + ANS1 + 2*X  + 5*X )/X
     ГДЕ
                        4       3       2
      ANS1 := SQRT(X  + 4*X  + 2*X  + 1)
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА    :(-2)
ФУНКЦИЯ РАВНА
         2          2
(ANS1 + X  + 2*X)/X
   ГДЕ
                        4       3       2
      ANS1 := SQRT(X  + 4*X  + 2*X  + 1)
СТАНДАРТНАЯ ФОРМА ИМЕЕТ ВИД Y**2 =
 3
X  - 12*X + 38
ТОЧКА,ИССЛЕДУЕМАЯ НА КРУЧЕНИЕ,
(-2)

 - 3*SQRT(2)*SQRT(3)
ТОЧКА ИМЕЕТ ПОРЯДОК-ДЕЛИТЕЛЬ ЧИСЛА 5
ТОЧКА В ДЕЙСТВИТЕЛЬНОСТИ ИМЕЕТ ПОРЯДОК 5
РАБОТА НА ДИВИЗОРЕ  :(5 5 -5 -5)
  В
X=
SQRT(2)/2
В ПЛЕЙСЕ +
X=
( - SQRT(2))/2
В ПЛЕЙСЕ +
X=
SQRT(2)/2
В ПЛЕЙСЕ -
X=
( - SQRT(2))/2
В ПЛЕЙСЕ -
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА :(-1 -1 1)
ЗАМЕНЕНО ПОЛЮСОМ
X=
 - SQRT(2) + 2
В ПЛЕЙСЕ     -
5 РАЗ
```

```
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА: (-2 1)
ЗАМЕНЕНО ПОЛЮСОМ
X=
SQRT(2) + 2
В ПЛЕЙСЕ +
2 РАЗА
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА: (-1 -1 1)
ЗАМЕНЕНО ПОЛЮСОМ
X=
SQRT(2)/2
В ПЛЕЙСЕ -
1 РАЗ
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА: (2 -1 -1)
(ВРЕМЯ АЛГОРИТМА КОУТСА 27197 МИЛЛИСЕКУНД)
ДИВИЗОР ИМЕЕТ ПОРЯДОК 5
ЛОГАРИФМ РАСШИРЕНИЯ РАВЕН
ANS1
   ГДЕ
                                                  4       3       2
      ANS1 := NTHROOT(( - 731*SQRT(X  + 4*X  + 2*X  + 1)*SQRT(2)*X
                     4       3       2                              4
    - 71492*SQRT(X  + 4*X  + 2*X  + 1)*SQRT(2) + 70030*SQRT(X  + 4*
     3       2                          4       3       2
    X  + 2*X  + 1)*X + 141522*SQRT(X  + 4*X  + 2*X  + 1) + 40957*
              3                       2
    SQRT(2)*X  + 174250*SQRT(2)*X  + 122871*SQRT(2)*X - 50648*SQRT(2
              3              2                                2
    ) - 90874*X  - 403722*X  - 272622*X + 61070)/(SQRT(2)*X  + 4*
                                  3       2                       4       3
    SQRT(2)*X + 2*SQRT(2) + 2*X  + 8*X  + 4*X)*( - 2*SQRT(X  + 4*X
           2                                 2
     + 2*X  + 1) + SQRT(2)*X + 2*SQRT(2))/(X  + 4*X + 2)*
           4       3       2                                4       3       2
(10*SQRT(X  + 4*X  + 2*X  + 1)*SQRT(2) + 17*SQRT(X  + 4*X  + 2*X
                      2                                  2
  + 1) + 4*SQRT(2)*X  + 16*SQRT(2)*X - 2*SQRT(2) - 11*X  - 44*X -
                                    2           2
 39)/(2*SQRT(2)*X + 4*SQRT(2) - X  - 4*X - 6) *
             4       3       2                                4       3       2
( - 4*SQRT(X  + 4*X  + 2*X  + 1)*SQRT(2) - 19*SQRT(X  + 4*X  + 2*X
                      2                                  2
  + 1) + 16*SQRT(2)*X  + 8*SQRT(2)*X - 6*SQRT(2) + 29*X  + 38*X - 5
        2     5
 )/(2*X  - 1) ,5)
ВНУТРЕННЯЯ РАБОТА ДАЕТ
(6*LOG((-1)) - 2*ANS1 - ANS2 + ANS3 - 5*ANS4 - 2*ANS5 + 5*ANS6 + 2*
 ANS7 + ANS8)/2
```

```
   ГДЕ
                                                4      3      2
     ANS8 := LOG(731*SQRT(2)*SQRT(X  + 4*X  + 2*X  + 1)*X + 71492*
                4      3      2                          3
SQRT(2)*SQRT(X  + 4*X  + 2*X  + 1) - 40957*SQRT(2)*X  - 174250*SQRT
    2                                                          4      3
(2)*X  - 122871*SQRT(2)*X + 50648*SQRT(2) - 70030*SQRT(X  + 4*X  +
   2                                 4      3      2                  3
2*X  + 1)*X - 141522*SQRT(X  + 4*X  + 2*X  + 1) + 90874*X  + 403722
  2
*X  + 272622*X - 61070)
                                               4      3      2
     ANS7 := LOG(10*SQRT(2)*SQRT(X  + 4*X  + 2*X  + 1) + 4*SQRT(2)
  2                                           4      3      2
*X  + 16*SQRT(2)*X - 2*SQRT(2) + 17*SQRT(X  + 4*X  + 2*X  + 1) - 11
  2
*X  - 44*X - 39)
                                            4      3      2
     ANS6 := LOG(4*SQRT(2)*SQRT(X  + 4*X  + 2*X  + 1) - 16*SQRT(2)
  2                                        4      3      2
*X  - 8*SQRT(2)*X + 6*SQRT(2) + 19*SQRT(X  + 4*X  + 2*X  + 1) - 29*
 2
X  - 38*X + 5)
                                                  2
     ANS5 := LOG(2*SQRT(2)*X + 4*SQRT(2) - X  - 4*X - 6)
                  2
     ANS4 := LOG(2*X  - 1)
                                                        4      3      2
     ANS3 := LOG(SQRT(2)*X + 2*SQRT(2) - 2*SQRT(X  + 4*X  + 2*X
 + 1))
     ANS2 := LOG(SQRT(2) + 2*X)
                2
     ANS1 := LOG(X  + 4*X + 2)
С ПРОИЗВОДНОЙ
         2              6      5      4      3
(ANS1*(6*X  + 5*X + 7))/(2*X  + 8*X  + 3*X  - 4*X  - 1)
   ГДЕ
                   4      3      2
     ANS1 := SQRT(X  + 4*X  + 2*X  + 1)
ОБНАРУЖЕННЫЕ ПОТЕНЦИАЛЬНЫЕ СОКРАЩЕНИЯ
(ЗАТРАЧЕННОЕ ВРЕМЯ 121778 МИЛЛИСЕКУНД)
          2              2              2
( - 4*LOG(X  + 4*X + 2)*X  + 2*LOG(X  + 4*X + 2) - 2*LOG(SQRT(2) +
       2
 2*X)*X  + LOG(SQRT(2) + 2*X) + 2*LOG(SQRT(2)*X + 2*SQRT(2) - 2*
       4      3      2          2
 SQRT(X  + 4*X  + 2*X  + 1))*X  - LOG(SQRT(2)*X + 2*SQRT(2) - 2*
```

```
      4      3      2                        2       2              2
SQRT(X  + 4*X  + 2*X  + 1)) - 10*LOG(2*X  - 1)*X  + 5*LOG(2*X  - 1
                                        2            2
) - 4*LOG(2*SQRT(2)*X + 4*SQRT(2) - X  - 4*X - 6)*X  + 2*LOG(2*
                                    2                          4
SQRT(2)*X + 4*SQRT(2) - X  - 4*X - 6) + 10*LOG(4*SQRT(2)*SQRT(X
      3      2                        2
 + 4*X  + 2*X  + 1) - 16*SQRT(2)*X  - 8*SQRT(2)*X + 6*SQRT(2) + 19
       4      3      2             2              2
*SQRT(X  + 4*X  + 2*X  + 1) - 29*X  - 38*X + 5)*X  - 5*LOG(4*SQRT(
         4      3      2                      2
2)*SQRT(X  + 4*X  + 2*X  + 1) - 16*SQRT(2)*X  - 8*SQRT(2)*X + 6*
                    4      3      2             2
SQRT(2) + 19*SQRT(X  + 4*X  + 2*X  + 1) - 29*X  - 38*X + 5) + 4*
                      4      3      2                      2
LOG(10*SQRT(2)*SQRT(X  + 4*X  + 2*X  + 1) + 4*SQRT(2)*X  + 16*SQRT
                                  4      3      2             2
(2)*X - 2*SQRT(2) + 17*SQRT(X  + 4*X  + 2*X  + 1) - 11*X  - 44*X
        2                             4      3      2
 - 39)*X  - 2*LOG(10*SQRT(2)*SQRT(X  + 4*X  + 2*X  + 1) + 4*SQRT(2
   2                                    4      3      2
)*X  + 16*SQRT(2)*X - 2*SQRT(2) + 17*SQRT(X  + 4*X  + 2*X  + 1) -
     2                                   4      3      2
11*X  - 44*X - 39) + 2*LOG(731*SQRT(2)*SQRT(X  + 4*X  + 2*X  + 1)*
                          4      3      2                      3
X + 71492*SQRT(2)*SQRT(X  + 4*X  + 2*X  + 1) - 40957*SQRT(2)*X  -
                 2
174250*SQRT(2)*X  - 122871*SQRT(2)*X + 50648*SQRT(2) - 70030*SQRT(
 4      3      2                             4      3      2
X  + 4*X  + 2*X  + 1)*X - 141522*SQRT(X  + 4*X  + 2*X  + 1) +
       3             2                             2
90874*X  + 403722*X  + 272622*X - 61070)*X  - LOG(731*SQRT(2)*SQRT
  4      3      2                               4      3      2
(X  + 4*X  + 2*X  + 1)*X + 71492*SQRT(2)*SQRT(X  + 4*X  + 2*X  + 1
                    3                        2
) - 40957*SQRT(2)*X  - 174250*SQRT(2)*X  - 122871*SQRT(2)*X +
                                  4      3      2
50648*SQRT(2) - 70030*SQRT(X  + 4*X  + 2*X  + 1)*X - 141522*SQRT(
 4      3      2                        3             2
X  + 4*X  + 2*X  + 1) + 90874*X  + 403722*X  + 272622*X - 61070)
                   2                                4      3      2
 + 12*LOG((-1))*X  - 6*LOG((-1)) + 2*SQRT(X  + 4*X  + 2*X  + 1)*X
        4      3      2                    2
 + SQRT(X  + 4*X  + 2*X  + 1))/(2*(2*X  - 1))
```

ПРИМЕР 6. ВЛОЖЕННЫЕ ВЫРАЖЕНИЯ

Следующий пример взят из распространенной таблицы интегралов (Bois, 1961), и поэтому можно сказать, что он более «реалистичен», чем предыдущие примеры, построенные для иллюстрации различных конкретных положений. Интегрируемая функция имеет вид $\sqrt{X+\sqrt{A^2+X^2}}/X$, т. е. содержит вложенные квадратные корни (в своем нынешнем представлении, с которым, вероятно, легче всего работать и которое, несомненно, доступнее для понимания, чем представление через примитивный элемент с помощью минимального многочлена). Представление с несколькими переменными для этой функции будет иметь вид Z/X, где $Z^2=X+Y$ и $Y^2=A^2+X^2$, в то время как примитивное представление для Z имеет вид $Z^4-2XZ^2-A^2=0$.

Подынтегральное выражение имеет вычеты $\sqrt{A}$, $i\sqrt{A}$ $-\sqrt{A}$, $-i\sqrt{A}$ в четырех плейсах, лежащих над 0, и не имеет больше никаких вычетов. Эти вычеты образуют 2-мерный **Z**-модуль, поэтому нужно рассмотреть два различных дивизора. Оба они имеют на самом деле порядок 1, и соответствующие им функции (найденные алгоритмом Коутса, соответственно, за 3.52 и 3.66 с.) таковы:

$$(-YZ\sqrt{A}+YA-ZA\sqrt{A}+Z\sqrt{A}\,X+A^2)/X \text{ и}$$
$$(-YZi\sqrt{A}-YA+Zi\sqrt{A}+Zi\sqrt{A}\,X+A^2)/X$$

(вычисление по алгоритму Коутса заняло 3.52 и 3.66 с.).

После того как эти два логарифма вычтены из интеграла, от подынтегральной функции остается

$$\frac{Z(YX+A^2+X^2)}{Y*A^2+Y*X^2+A^2*X+X^3}$$

а интеграл этой функции равен

$$a(1+i)\log(-yz\sqrt{a}+ya-za\sqrt{a}+z\sqrt{a}\,x+a^2)+$$
$$+a(1+i)\log(-yzi\sqrt{a}-ya+zi\sqrt{a}\,a+zi\sqrt{a}\,x+a^2)-$$
$$-\sqrt{a}\,(1+i)\log x+2z.$$

Это выражение для интеграла сильно отличается от выражения в стандартной таблице (Bois, 1961), так как последнее выражено через обратные тригонометрические функции. Еще интереснее, что эти выражения не эквивалентны (даже если учесть, что производная константы равна 0). В действительности в стандартной таблице помещен неверный ответ — интеграл от функции Z/Y.

Весь процесс интегрирования занял 12 с., включая два шага алгоритма Коутса и время, которое потребовалось для того, чтобы обработать все алгебраические выражения и проверить их на независимость. Кроме того, версия программы, поставляемая вместе с системой REDUCE-2 (и не предназначенная для исследований по проблеме интегрирования алгебраических выражений), содержит различные эвристические разделы, которые зачастую дают ей возможность «угадать» правильный ответ, не делая всей работы. Из этой версии удалены многие «диагностические» сообщения, и она делает не так много проверок на внутреннюю правильность. Важно отметить, что эти изменения не влияют на полноту результата работы программы и не означают, что она может объявить какую-то интегрируемую функцию неинтегрируемой; они просто отражают разницу между программами, приспособленными для исследовательских работ и для практической эксплуатации. Поставляемая версия программы взяла этот интеграл за 1.55 с.

В эксплуатационной версии программы следует, вероятно, сделать еще одно изменение — вставить проверку существования менее «дорогих» выражений для алгебраических величин в тех случаях, когда их легко можно найти. Например, в описанном выше примере нам часто встречается квадратный корень из A, и работать с этим алгебраическим выражением приходится весьма осторожно (см. гл. 2). Если бы мы всюду заменили A на B^2, то эта проблема не возникла бы, и можно было бы ожидать, что программа будет работать быстрее. В нашем случае можно было бы ожидать, что экономия составит около 10 % от общего времени.

ПРИМЕР 7. ЛОГАРИФМИЧЕСКИ НЕИНТЕГРИРУЕМАЯ ФУНКЦИЯ

Рассмотрим задачу интегрирования функции $1/Y*X$, где $Y^2 = X^3 - 3X^2 + X + 1$.. Подынтегральное выражение есть дифференциал третьего рода и имеет вычеты ± 1 в двух плейсах, лежащих над $X = 0$. В действительности плейсы, лежащие над $X = 0$, имеют бесконечный порядок на эллиптической кривой, и программа обнаруживает это, пытаясь применить алгоритм Коутса к дивизорам $D, D^2, \ldots, D^{10}, D^{12}$. Так как ни один из этих дивизоров не главный, мы можем

использовать оценку Мазура и заключить, что никакое ненулевое кратное этого дивизора не является главным дивизором, и следовательно функция не интегрируема. Заметим, что имеется отличие от приведенного выше примера 2, где не было вычетов. Здесь мы имеем вычеты, но не можем найти логарифм, который породил бы их после дифференцирования.

Интегрирование заняло 57,4 секунды, из которых 11 применений алгоритма Коутса заняло 0.5, 1.1, 1.3, 1.8, 2.3, 3.3, 4.0, 5.4, 6.5, 9.4 и 15.4 с. соответственно. Последнее применение алгоритма Коутса (к дивизору $12D$) породило следующее множество функций в качестве базиса пространства функций, имеющих нужные полюсы.

```
                                                    3      2           5
(64587461128523013562365623381754908388O*SQRT(X  - 3*X  + X + 1)*X

                                                      4
 + 126138337392550826553734563041284369888*lastsqrt*X  -

                                                   3
171649619884877543290007948893663863859 2*lastsqrt*X  -

                                                  2
264366112672565270849916395223433129984*lastsqrt*X  +

105900251808836156512187623923268171980 8*lastsqrt*X +

383047054317859708337698142489458630656*lastsqrt +

                                            7
267088026983715424094921823683103840001*X  +

                                            6
465162773628884682455314935369884569800*X  -

                                           5
225361068275637900238104147393158438464 8*X  -

                                           4
330309532751911649696040040039618069408*X  +

                                           3
306680908828645474445485776328404854553 6*X  +

                                           2
357316316894906514337737757152462544896*X  -

125052604524729141929072531047741103513 6*X -
```

```
                                           7
383047054317859708337698142489458630656)/X

                                                3        2            5
(645874611285230135623656233817549083880*SQRT(X  - 3*X  + X + 1)*X

                                                        4
 + 126138337392550826553734563041284369888*lastsqrt*X  -

                                                       3
171649619884877543290007948893663863592*lastsqrt*X  -

                                                      2
264366112672565270849916395223433129984*lastsqrt*X  +

105900251808836156512187623923268179808*lastsqrt*X +

383047054317859708337698142489458630656*lastsqrt +

                                           7
267088026983715424094921823683103840001*X  +

                                           6
465162773628884682455314935369884569800*X  -

                                            5
225361068275637900238104147393158438464 8*X  -

                                           4
330309532751911649696040040039618069408*X  +

                                            3
306680908828645474445485776328404854553 6*X  +

                                           2
357316316894906514337737757152462544896*X  -

125052604524729141929072531047741103513 6*X -

                                            8
383047054317859708337698142489458630656)/X

                                                 3        2            5
(645874611285230135623656233817549083880*SQRT(X  - 3*X  + X + 1)*X

                                                         4
 + 126138337392550826553734563041284369888*lastsqrt*X  -
```

```
                                                                  3
1716496198848775432900079488936638638592*lastsqrt*X  -

                                                                2
264366112672565270849916395223433129984*lastsqrt*X  +

10590025180883615651218762392326817198 08*lastsqrt*X +

38304705431785970833769814248945863 0656*lastsqrt +

                                                          7
267088026983715424094921823683103840001*X  +

                                                          6
4651627736288846824553149353698845698 00*X  -

                                                          5
2253610682756379002381041473931584384648*X  -

                                                          4
3303095327519116496960400400396180694 08*X  +

                                                          3
3066809088286454744454857763284048545536*X  +

                                                          2
3573163168949065143377377571524625448 96*X  -

12505260452472914192907253104774110351 36*X -

                                                          9
3830470543178597083376981424894586306 56)/X

                                                                3       2              5
(64587461128523013562365623381754908388 0*SQRT(X  - 3*X  + X + 1)*X

                                                                  4
 + 126138337392550826553734563041284369888*lastsqrt*X  -

                                                                3
1716496198848775432900079488936638638592*lastsqrt*X  -

                                                              2
264366112672565270849916395223433129984*lastsqrt*X  +

10590025180883615651218762392326817198 08*lastsqrt*X +
```

```
38304705431785970833769814248945863065 6*lastsqrt +

                                                 7
267088026983715424094921823683103840001*X  +

                                                 6
465162773628884682455314935369884569800*X  -

                                                 5
225361068275637900238104147393158438464 8*X  -

                                                 4
330309532751911649696040040039618069408*X  +

                                                 3
306680908828645474444548577632840485455 36*X  +

                                                 2
357316316894906514337737757152462544896*X  -

125052604524729141929072531047741103513 6*X -

                                                  10
38304705431785970833769814248945863065 6)/X

                                                      3       2
(64587461128523013562365623381754908388 0*SQRT(X  - 3*X  + X + 1)*X

                                                          4
 + 12613833739255082655373456304128436988 8*lastsqrt*X  -

                                                        3
171649619884877543290007948893663863859 2*lastsqrt*X  -

                                                      2
264366112672565270849916395223433129984*lastsqrt*X  +

105900251808836156512187623923268171980 8*lastsqrt*X +

38304705431785970833769814248945863065 6*lastsqrt +

                                                 7
267088026983715424094921823683103840001*X  +

                                                 6
465162773628884682455314935369884569800*X  -
```

```
                                                                  5
225361068275637900238104147393158438464 8*X  -

                                                                4
330309532751911649696040040039618069408*X  +

                                                                3
306680908828645474445485776328404854553 6*X  +

                                                                2
357316316894906514337737757152462544896*X  -

125052604524729141929072531047741103513 6*X -

                                                               11
383047054317859708337698142489458630656)/X

                                                                    3       2              5
(64587461128523013562365623381754908388 0*SQRT(X  - 3*X  + X + 1)*X

                                                                   4
 + 126138337392550826553734563041284369888*lastsqrt*X  -

                                                                  3
171649619884877543290007948893663863859 2*lastsqrt*X  -

                                                                 2
264366112672565270849916395223433129984*lastsqrt*X  +

105900251808836156512187623923268171980 8*lastsqrt*X +

383047054317859708337698142489458630656*lastsqrt +

                                                                7
267088026983715424094921823683103840001*X  +

                                                                6
465162773628884682455314935369884569800*X  -

                                                                 5
225361068275637900238104147393158438464 8*X  -

                                                                4
330309532751911649696040040039618069408*X  +

                                                                 3
306680908828645474445485776328404854553 6*X  +
```

```
                                                    2
3573163168949065143377377571524625448 96*X  -

1250526045247291419290725310477411035136*X -

                                                   12
38304705431785970833769814248945863 0656)/X

                                  3      2           5
( - 2316186053639990532*SQRT(X  - 3*X  + X + 1)*X  -

                                    4
3548648083863035028 0*lastsqrt*X  - 342870252002846026 24*lastsqrt*

 3                                      2
X  + 4733070901227714201 6*lastsqrt*X  + 59834813484366611456*

lastsqrt*X + 1568408874531272499 2*lastsqrt - 16342827998351920001*

 6                          5                              4
X  - 28462807885868569800*X  + 8200978454614901282 4*X  +

                         3                                 2
872678761280188236 80*X  - 5176147154332726963 2*X  -

                                                  7
67676857857022973952*X - 1568408874531272499 2)/X

                                  3      2           5
( - 2316186053639990532*SQRT(X  - 3*X  + X + 1)*X  -

                                    4
3548648083863035028 0*lastsqrt*X  - 342870252002846026 24*lastsqrt*

 3                                      2
X  + 4733070901227714201 6*lastsqrt*X  + 59834813484366611456*

lastsqrt*X + 1568408874531272499 2*lastsqrt - 16342827998351920001*

 6                          5                              4
X  - 28462807885868569800*X  + 8200978454614901282 4*X  +

                         3                                 2
872678761280188236 80*X  - 5176147154332726963 2*X  -

                                                  8
67676857857022973952*X - 1568408874531272499 2)/X
```

```
                                  3       2           5
( - 2316186053639990532*SQRT(X  - 3*X  + X + 1)*X  -

                                    4
3548648083863035028O*lastsqrt*X  - 342870252002846026Z4*lastsqrt*

 3                                  2
X  + 4733070901227714201G*lastsqrt*X  + 59834813484366611456*

lastsqrt*X + 1568408874531272499Z*lastsqrt - 16342827998351920001*

 6                    5                              4
X  - 2846280788586856980O*X  + 8200978454614901282Z*X  +

                     3                              2
8726787612801882368O*X  - 5176147154332726963Z*X  -

                                                    9
67676857857022973952*X - 1568408874531272499Z)/X

                                  3       2           5
( - 2316186053639990532*SQRT(X  - 3*X  + X + 1)*X  -

                                    4
3548648083863035028O*lastsqrt*X  - 342870252002846026Z4*lastsqrt*

 3                                  2
X  + 4733070901227714201G*lastsqrt*X  + 59834813484366611456*

lastsqrt*X + 1568408874531272499Z*lastsqrt - 16342827998351920001*

 6                    5                              4
X  - 2846280788586856980O*X  + 8200978454614901282Z*X  +

                     3                              2
8726787612801882368O*X  - 5176147154332726963Z*X  -

                                                   10
67676857857022973952*X - 1568408874531272499Z)/X

                                  3       2           5
( - 2316186053639990532*SQRT(X  - 3*X  + X + 1)*X  -

                                    4
3548648083863035028O*lastsqrt*X  - 342870252002846026Z4*lastsqrt*
```

```
 3                                    2
X  + 473307090122771420l6*lastsqrt*X  + 59834813484366611456*

lastsqrt*X + 156840887453127249l2*lastsqrt - 163428279983519200l*

 6                      5                      4
X  - 2846280788586856980l*X  + 820097845461490128l4*X  +

                       3                                2
872678761280188236l0*X  - 5176147154332726963l*X  -

                                                       11
67676857857022973952*X - 1568408874531272499l)/X

                                  3      2           5
( - 231618605363999053l*SQRT(X  - 3*X  + X + 1)*X  -

                                4
3548648083863035028l*lastsqrt*X  - 34287025200284602624*lastsqrt*

 3                                    2
X  + 473307090122771420l6*lastsqrt*X  + 59834813484366611456*

lastsqrt*X + 156840887453127249l2*lastsqrt - 163428279983519200l*

 6                      5                                4
X  - 2846280788586856980l*X  + 820097845461490128l4*X  +

                       3                                2
872678761280188236l0*X  - 5176147154332726963l*X  -

                                                       12
67676857857022973952*X - 1568408874531272499l)/X
```

Не существует линейной комбинации этих функций, имеющей нужный нуль кратности 12, поэтому этот дивизор действительно неглавный.

Размер этих выражений в сравнении с исходной подынтегральной функцией показывает, как сильно этот алгоритм страдает от «разбухания промежуточных выражений». Стоит заметить, что если бы мы использовали тест Лютца — Нагеля (см. гл. 7), то немедленно узнали бы, что рассматриваемый дивизор не имеет кручения. Эту возможность повышения эффективности вероятно следует использовать при реализации.

ПРИМЕР 8. СУММА ДВУХ ФУНКЦИЙ

Рассмотрим интеграл от $1/f + k/g$, где k — константа, $f = \sqrt{X^2 - 1}$, $g = \sqrt{X^2 - 4}$. Подынтегральное выражение имеет вычеты $\pm 1 \pm k$ в 4 плейсах, лежащих на бесконечности. Поэтому, если число k не рационально, то пространство вычетов двумерно и логарифмическая часть является суммой двух членов, один из которых соответствует $1/f$, другой $1/g$.

Если k рационально (т. е. равно p/q, где p, q — целые), то пространство вычетов одномерно с базисом $1/q$, а коэффициенты вычетов равны $\pm(p \pm q)$. Если p и q нечетны, то $p \pm q$ всегда четно и в качестве базиса можно взять элемент $2/q$. В действительности случаи $1/q$ и $2/q$ различны, и мы рассмотрим их отдельно.

Рассмотрим случай $1/q$ и приведем результаты запуска с непосредственным использованием алгоритма Коутса, затем — с использованием алгоритма ДИВИЗОР_В_ФУНКЦИЮ, чтобы показать, как в некоторых случаях применение этого улучшенного метода может ускорить работу и дать более осмысленный результат.

```
INT(1/SQRT(X**2-1)+10/SQRT(X**2-4),X);
ПЛЕЙСЫ, В КОТОРЫХ МОГУТ БЫТЬ ПОЛЮСА
X=БЕСКОНЕЧНОСТИ
X=-1
X=1
X=2
X=-2
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА :(-11 9 -9 11)
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА :(-11 -11 -11 -11 20 2 22)
(ВРЕМЯ АЛГОРИТМА КОУТСА 42316 МИЛЛИСЕКУНД)
ЛОГАРИФМ РАСШИРЕНИЯ РАВЕН
                9                   7                   5
 - ANS1*ANS2*X  + 3*ANS1*ANS2*X  - 21*ANS1*ANS2*X  + 20*ANS1*ANS2
 3                          10           8           6
X  - 5*ANS1*ANS2*X - ANS1*X   + 8*ANS1*X  - 21*ANS1*X  + 20*ANS1*
           2          10            8           6           4
 - 5*ANS1*X  - ANS2*X   + 10*ANS2*X  - 35*ANS2*X  + 50*ANS2*X  -
      2              11       9       7       5       3
*ANS2*X  + 2*ANS2 - X   + 10*X  - 35*X  + 50*X  - 25*X  + 2*X
   ГДЕ
```

```
                    2
      ANS2 := SQRT(X  - 1)
                    2
      ANS1 := SQRT(X  - 4)
ВНУТРЕННЯЯ РАБОТА ДАЕТ
LOG((-1)) + ANS1 + ANS2
  ГДЕ
                          2        9           2        7            2
      ANS2 := LOG(SQRT(X  - 4)*X  - 8*SQRT(X  - 4)*X  + 21*SQRT(X
       5           2          3           2           10       8
 - 4)*X  - 20*SQRT(X  - 4)*X  + 5*SQRT(X  - 4)*X + X   - 10*X  + 35
  6       4        2
*X  - 50*X  + 25*X  - 2)
                       2
      ANS1 := LOG(SQRT(X  - 1) + X)
С ПРОИЗВОДНОЙ
                 12                    10                   8
(11*ANS1*ANS2*X   - 122*ANS1*ANS2*X   + 503*ANS1*ANS2*X  - 954*ANS1
        6                    4                   2
 *ANS2*X  + 835*ANS1*ANS2*X  - 290*ANS1*ANS2*X  + 20*ANS1*ANS2 + 11
        13              11              9              7
 *ANS1*X   - 123*ANS1*X   + 515*ANS1*X  - 1007*ANS1*X  + 939*ANS1*
  5              3                          13              11
 X  - 375*ANS1*X  + 40*ANS1*X + 11*ANS2*X   - 144*ANS2*X   + 725*
        9              7              5              3
 ANS2*X  - 1760*ANS2*X  + 2115*ANS2*X  - 1152*ANS2*X  + 208*ANS2*X
        14         12         10         8         6         4
  + 11*X   - 145*X   + 739*X   - 1835*X  + 2305*X  - 1377*X  + 310*
  2                    13                   11                   9
 X  - 8)/(ANS1*ANS2*X   - 13*ANS1*ANS2*X   + 65*ANS1*ANS2*X  - 157*
                     7                    5                   3
           ANS1*ANS2*X  + 189*ANS1*ANS2*X  - 105*ANS1*ANS2*X  + 20*
                                14                  12                  10
           ANS1*ANS2*X + ANS1*X   - 13*ANS1*X   + 65*ANS1*X   - 157*
                 8                   6                   4                   2
           ANS1*X  + 189*ANS1*X  - 105*ANS1*X  + 20*ANS1*X  + ANS2*
            14                  12                  10                  8
           X   - 15*ANS2*X   + 89*ANS2*X   - 265*ANS2*X  + 415*ANS2*
            6                   4                   2                  15        13
          .X  - 327*ANS2*X  + 110*ANS2*X  - 8*ANS2 + X   - 15*X   +
              11         9         7         5         3
           89*X   - 265*X  + 415*X  - 327*X  + 110*X  - 8*X)
  ГДЕ
                    2
      ANS2 := SQRT(X  - 1)
                    2
      ANS1 := SQRT(X  - 4)
```

(ЗАТРАЧЕННОЕ ВРЕМЯ 47144 МИЛЛИСЕКУНДЫ)

```
          2                                2          9              2        7
LOG(SQRT(X  - 1) + X) + LOG(SQRT(X  - 4)*X  - 8*SQRT(X  - 4)*X  +
         2        5              2         3           2            10
21*SQRT(X  - 4)*X  - 20*SQRT(X  - 4)*X  + 5*SQRT(X  - 4)*X + X
    8        6         4         2
10*X  + 35*X  - 50*X  + 25*X  - 2) + LOG((-1))
```

Теперь используем алгоритм ДИВИЗОР_ФУНКЦИИ

```
INT(1/SQRT(X**2-1)+10/SQRT(X**2-4),X);
ПЛЕЙСЫ, В КОТОРЫХ МОГУТ БЫТЬ ПОЛЮСА
X= бесконечности
X=-1
X=1
X=2
X=-2
РАБОТА С ДИВИЗОРОМ: (11 9 -9 -11)
 B
X= бесконечность --
X= бесконечность +-
X= бесконечность -+
X= бесконечность ++
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА    : (-1 -1 1)
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА    : (-1 -1 -1 -1 2 1)
ЗАМЕНЕНО НА ПОЛЮС
X= бесконечность +-
9 РАЗ
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА    : (2 -2)
НАЙТИ ФУНКЦИЮ С НУЛЯМИ ПОРЯДКА    : (-2 -2 -2 -2 4 2 2)
(ВРЕМЯ АЛГОРИТМА КОУТСА 2948 МИЛЛИСЕКУНД)
ЛОГАРИФМ РАСШИРЕНИЯ РАВЕН
ANS1
  ГДЕ
                    2              2              2                2
     ANS1 := SQRT(X  - 1)*SQRT(X  - 4) + SQRT(X  - 1)*X + SQRT(X
                   2        2          9
           - 4)*X + X *SQRT(X  - 4) + X
ВНУТРЕННЯЯ РАБОТА ДАЁТ
ANS1 + 10*ANS2
  ГДЕ
                      2
     ANS2 := LOG(SQRT(X  - 4) + X)
                     2
     ANS1 := LOG(SQRT(X  - 1) + X)
С ПРОИЗВОДНОЙ
```

```
                3                                   4             2
(11*ANS1*ANS2*X  - 14*ANS1*ANS2*X + 11*ANS1*X  - 15*ANS1*X  + 4*
              4                2                    5        3
 ANS1 + 11*ANS2*X  - 54*ANS2*X  + 40*ANS2 + 11*X  - 55*X  + 44*X)/(
              4                    2                          5
   ANS1*ANS2*X  - 5*ANS1*ANS2*X  + 4*ANS1*ANS2 + ANS1*X  - 5*ANS1*
    3                    5             3                 6       4
   X  + 4*ANS1*X + ANS2*X  - 5*ANS2*X  + 4*ANS2*X + X  - 5*X  + 4*
    2
   X )
   где
                    2
      ANS2 := SQRT(X  - 1)
                    2
      ANS1 := SQRT(X  - 4)
(затраченное время 6107 миллисекунд)
         2                                  2
LOG(SQRT(X  - 1) + X) + 10*LOG(SQRT(X  - 4) + X)
```

Случай «$2/q$» сложнее, так как дивизор, заданный вычетами, имеет порядок 2, т. е. D — неглавный, а $2D$ главный. Однако анализ дивизора $2D$ совпадает с приведенным выше анализом дивизора D. В частности, он заканчивается гораздо быстрее, если используется алгоритм ДИВИЗОР_В_ФУНКЦИЮ.

АЛГОРИТМЫ РАБОТЫ С АЛГЕБРАИЧЕСКИМИ ВЫРАЖЕНИЯМИ

Это приложение содержит различные алгоритмы, предназначенные для работы с алгебраическими выражениями общего вида. Многие из них можно найти (правда, в вариантах, рассчитанных на представление через примитивный элемент) в статье (Trager, 1976).

Следует отметить, что многие из них не всегда дают нужный результат во всех полях конечной характеристики. Очевидно, что алгебраические алгоритмы над конечными полями требуют дальнейших интенсивных исследований.

БЕСКВ_НОРМА [1]

Вход: K: поле характеристики p (возможно, 0).
A: алгебраический элемент над K, определяемый многочленом P_A.
F(X, A): многочлен над K(A), не содержащий квадратов.

Выход: Либо ответ НЕУДАЧА (см. подстрочное примечание), либо
S: положительное целое число.
G(X, A) := F(X — SA, A).
R(X) := NORM(G(X, A)): бесквадратный многочлен над K.
Норма берется относительно расширения K(A)/K.

[1] S := 0
G(X, A) := F(X, A).

[2] R(X) := Результант (P_A(Y), G(X, Y), Y).
Это — результат G и минимального многочлена для A, взятый относительно фиктивной переменной Y.

---

[1]) Составлен по образцу алгоритма SQFR_NORM из работы (Trager, 1976). Разница в том, что мы хотим применять его в случае, когда основное поле имеет конечную характеристику. В этом случае алгоритм не обязательно завершит работу успешно, и может иногда выдать ответ НЕУДАЧА. Тогда вычисления приходится прекратить, но это не очень заботит нас, так как может произойти лишь для конечного множества простых чисел. Мы отсылаем читателя к статье Трейгера за подробностями доказательства корректности этого алгоритма.

[3] Если R(X) = РАЗЛОЖЕНИЕ_БЕСКВАДРАТНОЕ (R(X)), то выдать ответ (S, G, R).

[4] S := S + 1
G(X, A) := G(X — A, A).

[5] Если S отлично от нуля
то перейти к [2].
Это может не выполниться только в поле ненулевой характеристики, так как S — положительное целое число.

[6] Выдать ответ НЕУДАЧА.

АЛГ_ФАКТОРИЗАЦИЯ

Вход F(X): многочлен иад $K(A_1, \ldots, A_n)$.
K: чисто трансцендентное поле.
$A_1, \ldots, A_n$: список алгебраических выражений, где A_i имеет неприводимый минимальный многочлен P_i над $K(A_1, \ldots, A_{i-1})$.

Выход: список множителей многочлена F с их кратностями. Если поле имеет конечную характеристику, то процедура может выдать ответ НЕУДАЧА, показывающий, что один из внутренних алгоритмов не смог работать в этом конечном поле.

[1] L := РАЗЛОЖЕНИЕ_БЕСКВАДРАТНОЕ (F)
[2] ОТВЕТ := NIL.
[3] Для I = 1, ..., длина L делать
[3.1] A := АЛГ_ФАКТОРИЗАЦИЯ_2(L(I), K, A_1, ..., A_n).
[3.2] Если A = НЕУДАЧА,
то выдать ответ НЕУДАЧА
[3.3] Для каждого B из A делать:
[3.3.1] ОТВЕТ := (B**I)·ОТВЕТ.
Возведение в степень здесь чисто символическое; оно показывает, что множитель B входит в F с кратностью I.
[4] Выдать ответ ОТВЕТ.

АЛГ_ФАКТОРИЗАЦИЯ 2 [1])

Вход: F(X): бесквадратный многочлен над $K(A_1, \ldots, A_n)$.
K: Чисто транцендентное поле.
$A_1, \ldots, A_n$: Список алгебраич с их выражений, где A_i имеет неприводимый минимальный многочлен P_i над $K(A_1, \ldots, A_{i-1})$.

Выход: Список множителей многочлена F.

[1] (S, G, R):= БЕСКВ_НОРМА ($K(A_1, \ldots, A_{n-1})$, A_n, F). Этот шаг может (но лишь в случае, когда K имеет конечную характеристику) дать ответ НЕУДАЧА. В этом случае и алгоритм в целом дает ответ НЕУДАЧА.

[2] Если n = 1
то L:= ФАКТОР (R(X))
Иначе L:= АЛГ_ФАКТОРИЗАЦИЯ_2 ($R(X), K, A_1, \ldots, A_{n-1}$).
Это — рекурсивный шаг процесса факторизации: алгоритм ФАКТОР используется тогда, когда нет алгебраических зависимостей, которые нужно разрешать [2]).

[3] Если Длина (L) = 1
то выдать ответ F.

[4] Для каждого H из L делать

[4.1] $H(X, A_n) := \text{н.о.д.}\ (H(X), G(X, A_n))$.

[4.2] $G(X, A_n) := G(X, A_n)/H(X, A_n)$.

[4.3] $H(X, A_n) := H(X + SA_n, A_n)$.

[5] Выдать ответ L.

ПРИМИТИВНЫЙ_ЭЛЕМЕНТ [3])

Вход: K: поле
$A_1, \ldots, A_n$: множество элементов алгебраического замыкания поля K, где A_i определяется многочленом $P_i(X)$ над полем $K(A_1, \ldots, A_{i-1})$.

---

[1]) Эта процедура аналогична алгоритму ALG_FACTOR из статьи (Trager, 1976), но она рекурсивна, так как предназначена для работы с представлением алгебраических выражений со многими переменными. Она вызывает алгоритм ФАКТОР, который предназначен для разложения своего аргумента на множители над любым неалгебраическим основным полем любой степени или характеристики. Такие алгоритмы построены с помощью *p*-адических методов (Wang, 1976, 1978).

[2]) В случае, когда K имеет характеристику 0 и все A_i — алгебраические целые числа, возможно улучшение; мы можем использовать какой-нибудь алгоритм разложения над полями алгебраических чисел, например тот, который приведен в статье (Wang & Rotschild, 1975).

[3]) Этот алгоритм аналогичен алгоритму PRIMITIVE_ELEMENT из статьи (Trager, 1976), но он предназначен для работы в представлении со многими переменными; он также выдает выражение для выбранного примитивного элемента в терминах исходного представления. Он годится и для полей конечной характеристики, и заканчивает работу, если срабатывает описанный выше алгоритм БЕСКВ_НОРМА; последнее может не произойти лишь для конечного множества простых чисел.

Выход: $R(X)$: минимальный многочлен для примитивного элемента X поля $K(A_1, \ldots, A_n)$ над K

X: представление (со многими переменными) этого элемента как элемента поля $K(A_1, \ldots, A_n)$

$Q_i(X)$: представление элемента A_i через примитивный элемент. Это — рациональные функции от X над K.

[1] Если $n = 1$
то выдать ответ $(P_1(X), A_1, P_1(X))$.

[2] $(R', X', (Q_i'(X'))) :=$ ПРИМИТИВНЫЙ ЭЛЕМЕНТ $(K, A_1, \ldots, A_{n-1})$.

Это дает нам примитивное представление для поля, порожденного элементами $A_1, \ldots, A_{n-1}$. Эти ответы не будут непосредственно включены в качестве компонент окончательного ответа, так как при добавлении элемента A_n примитивный элемент X изменится и значит придется изменить многочлены $Q_i(X)$.

Если характеристика поля K отлична от нуля, то этот шаг может выдать ответ НЕУДАЧА. В этом случае и ответ вычисления в целом будет НЕУДАЧА.

[3] $(S, G, R) :=$ БЕСКВ_НОРМА (K, X', P_n).

Заметим, что если в нашем представлении со многими переменными встречались вложенные выражения, то в P_n могут входить некоторые из предшествующих A_i. В этом случае их следует заменить представлениями через X', т. е. многочленами $P_i'(X')$.

Заметим также, что этот шаг может дать ответ НЕУДАЧА, если характеристика поля K отлична от нуля, и если это произойдет, то мы не сможем продолжить это вычисление (даже если примитивный элемент в действительности существует).

[4] $Z := B/A$, где $AX' - B =$ н. о. д. $(G(X', X), R'(X'))$.

Это вычисление выполняется в $K(X)$, где X — корень многочлена R, а X' рассматривается как переменная многочленов, для которых находится наибольший общий делитель.

Z — это представление X' через X.

[5] $Q_n(X) := X - S*Z$

Это отражает тот факт, что $X = A_n + S*X'$.

[6] Для $i = 1, \ldots, n-1$ делать:

[6.1] $Q_i(X) :=$ Подставить $(Q_i'(X'), X' \to Z)$.

Этот цикл порождает все многочлены, определяющие A_i через X.

[7] Выдать ответ $(R(X), A_n + S*X', (Q_i))$.

РАЗЛОЖЕНИЕ_БЕСКВАДРАТНОЕ [1])

Вход: R(X): многочлен над некоторым основным полем K.
Выход: $(R_1(X), \ldots, R_n(X))$

где

$$R(X) = \prod_{i=1}^{i=n} R_i(X)^i$$

и R_i взаимно просты и не содержат квадратов.

[1] Если степень (R) = 1, то выдать ответ R.

[2] R′ := dR/dX

как формальная производная многочленов

[3] Если R′ = 0, то

Это может произойти лишь в том случае, когда каждый одночлен в R является p-й степенью по X, где p — характеристика поля K, так как R имеет положительную степень. В частности, это не может произойти, если K имеет характерику 0.

[3.1] $R'' := (R(X))^{1/p}$

Полагаем $R'' = \sum_{i=0}^{i=k} a_i X^i$, где $R = \sum_{i=0}^{i=k} a_i X^{ip}$

и p — характеристика поля K.

[3.2] L := РАЗЛОЖЕНИЕ_БЕСКВАДРАТНОЕ (R″)

[3.3] Выдать ответ L′,

где L' получается из L вставкой $(p-1)$ нулей перед каждым элементом L в качестве указания на то, что мы хотим разложить $R = R'^p$.

[4] R″ = н. о. д. (R, R′).

[5] Если R″ — элемент поля K

то выдать ответ R.

В этом случае R бесквадратный и мы просто выдаем его в качестве ответа.

[6.1] M[1] := (R/R″)/н. о. д. (R″, R/R″).

Ответ накапливается в векторе M. Заметим, что хотя элемент $M[1]$ вычисляется как частное, в действительности он является многочленом по X над K.

[6.2] R := R/M[1]

Мы исключаем множители из R по мере их нахождения.

[7] L := РАЗЛОЖЕНИЕ_БЕСКВАДРАТНОЕ (R″).

---

[1]) Этот алгоритм разложения многочленов на множители, не содержащие квадратов, почти наверняка не оригинален. Единственная особенность, которую я не смог найти в элементарных текстах, это работа с p-ми степенями по модулю простого числа p. Я уверен, что не я первый это придумал, но не знаю другого описания.

Это — не самый эффективный алгоритм для полей характеристики 0 (такой см. в статье (Yun, 1977b)), но более эффективный алгоритм не обобщается так легко на поля простой характеристики.

[8] Для I = 2, ... делать:
До тех пор, пока не исчерпаются ненулевые элементы списка L.
[8.1] Если p | I,
то M[I] := L[I]
Иначе M[I + 1] := L[I]
[8.2] Если p | I,
то R := R/L[I]**I
Иначе R := R/L[I]**(I + 1).
[9] Если R — элемент поля K,
То выдать ответ M.
[10] Для I = p, 2p, ..., делать:
Этот цикл, который выполняется только в случае ненулевой характеристики, нужен для преодоления недоразумений, вызванных смешением множителей Y^p и Y^{p+1}.
[10.1] R′ := н. о. д. (R, M(I)).
[10.2] M[I + 1] := R′.
[10.3] M[I] := M[I]/R′.
[10.4] R := R/R′.
[11] Выдать ответ M.

Лемма 1. *Описанный выше алгоритм завершает работу.*

Доказательство. Единственный способ, которым этот алгоритм может зациклиться — это бесконечная рекурсия. Но если он вызывает сам себя на шаге 3.2, то он уменьшает степень своего аргумента не менее чем в p раз, а если он вызывает себя на шаге 7, то мы имеем $\deg(R'') = \deg(\text{н. о. д.}(R, R')) \leqslant \deg(R') < \deg(R)$, так что степень — строго убывающая целочисленная функция. Следовательно рекурсия конечна, и алгоритм завершает работу.

Теорема 2. *Алгоритм выдает правильный ответ.*

Доказательство. Пусть $R(X) = \prod_{i=1}^{n} R_i(X)^i$. Мы можем считать, что степень многочлена $R(X)$ больше 1, так как иначе задача тривиальна. Используя предыдущую лемму, будем рассуждать с помощью индукции по степени многочлена R. Если $R(X)$ есть p-я степень (где p — характеристика основного поля), то $R'(X) = 0$ и алгоритм переходит к шагу 3. Тогда R'' есть корень p-й степени из R и алгоритм работает правильно.

Если R не есть p-я степень, то

$$R'(X) = \sum_{i=1}^{n} iR_i(X)' \frac{R(X)}{R_i(X)},$$

где $R_i(X)'$ есть производная многочлена $R_i(X)$. Пусть $S = \prod_{i=1}^{n} R_i(X)^{i-1}$, $T = \prod_{i=1}^{n} R_i(X)$. Тогда $R'' = \text{н. о. д.}(R, R') =$

$$S \text{ н. о. д.} \left(T, \sum_{i=1}^{i=n} iR_i(X)' \frac{T}{R_i(X)} \right).$$

Из того что R_i взаимно просты, мы получаем, что единственно возможные множители последнего наибольшего общего делителя — это те R, для которых $i = 0 \pmod p$. Следовательно, $R'' = \prod_{i=1}^{n} R_i(X)^{i'}$, где

$$i' = \begin{cases} i, \text{ если } i \mid p, \\ i - 1 \text{ в противном случае.} \end{cases}$$

Тогда R/R'' есть $\prod_{i=1}^{i=n} R_i(X)$ (p не является делителем i), и $M[1]$ есть тогда $R_1(X)$.

Затем мы можем получить свободное от квадратов разложение R'' — именно это и делается на шаге [7]. К сожалению, члены вида $R_{kp}(X)^{kp}$ и $R_{kp+1}(X)^{kp+1}$ были смешаны в R'', так как и те и другие имеют теперь показатель kp. В разложение R'' они оба войдут с показателем kp, и цикл на шаге [10] произведет необходимый отбор, так как $R_{kp+1}(X)$ даст остаточный множитель, а $R_{kp}(X)$ не даст. Это доказывает корректность другой ветви алгоритма, а значит и алгоритма в целом.

ЛИТЕРАТУРА

Сокращения

Proc. SYMSAC 76.
Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation. ACM Inc., New York, 1976.

Proc. EUROSAM 79. Proceedings of the 1979 European Symposium on Symbolic and Algebraic Computation. Springer-Verlag Lecture Notes in Computer Science 72, Berlin-Heidelberg-New York 1979.

Proc. 1977 MACSYMA Users' Conference. NASA publication CP-2012, National Technical Information Service, Springfield, Virginia.

(Baker,1975) Baker,A., Transcendental Number Theory. Cambridge University Press, 1975.

(Baker & Coates,1970) Baker,A. & Coates,J., Integer Points on Curves of genus 1. Proc. Cam. Phil. Soc. 67(1970) pp. 595 et seq.

(Baldassari & Dwork,1979) Baldassari,R., & Dwork,B., On second order linear differential equations with algebraic solutions. Amer. J. Math. 101(1979) pp. 42-76.

(Berlekamp,1967) Berlekamp,E.R., Factoring Polynomials over Finite Fields. Bell System Tech. J. 46(1967) pp. 1853-1859.

(Berlekamp,1970) Berlekamp,E.R., Factoring Polynomials over Large Finite Fields. Math. Comp. 24 (1970) pp. 713-735.

(Birch & Swinnerton-Dyer,1963) Birch,B.J. & Swinnerton-Dyer, H.P.F., Notes on Elliptic Curves I, J. Reine u. Angew. Math. 212(1963) pp. 7-23.

(Bogen et al,1977) Bogen,R.A, et al. MACSYMA Reference Manual. MIT Laboratory for Computer Science, Cambridge, Mass.

(Bois,1961) Bois, G. Petit, A Table of Indefinite Integrals. Dover 1861.

(Borevich & Shafarevich,1966) Borevich,Z.I. & Shafarevich,I.R., Number Theory. Academic Press, New York, 1966 (Translated from Teoria Chisel, Moscow, 1964).

(Buchberger,1979) Buchberger,B. A Criterion for Detecting Unnecessary Reductions in the Construction of Groebner Bases. Proc. EUROSAM 79 pp. 3-21.

(Carlson,1965) Carlson,B.C., On Computing Elliptic Integrals & Functions. J. Math. Phys. 44(1965) pp. 36-51.

(Cassels,1966) Cassels,J.W.S., Diophantine Equations with Special Reference to Elliptic Curves. J. L.M.S. 41(1966) pp. 193-291.

(Caviness,1978) Caviness,B.F., Private Communication, July 1978.

(Caviness & Fateman,1976) Caviness,B.F. & Fateman,R.J., Simplification of Radical Expressions. Proc SYMSAC 76, pp. 329-338.

(Cayley,1853) Cayley,A., Note on the Porism of the in-and-circumscribed Polygon. Philosophical Magazine (4th. ser) VI(1853) pp. 99-103 & 376-7.

(Cayley,1861) Cayley,A., On the Porism of the in-and-circumscribed Polygon. Phil. Trans. Roy. Soc. CLI(1861) pp. 225-239.

(Chebyshev,1821-1894) Chebyshev(Tchebichef),P.L., Oeuvres de. 2 vols., St Pétersbourg. Reprinted Chelsea Publishing Co., New York.

(Chebyshev,1853) Chebyshev(Tchebichef),P.L., Sur l'intégration des différentielles irrationelles. Journal de Maths. Pures et Appls. XVIII(1853) pp. 87-111. Oeuvres (vide supra) vol. I pp. 147-168.

(Chebyshev,1857) Chebyshev(Tchebichef),P.L., Sur l'intégration des différentielles qui contiennent une racine carrée d'un polynôme du troisième ou du quatrième degré. Journal de Maths. Pures et Appl. (2nd. ser) II(1857) pp. 1-42 (taken from Memoires de l'Academie Impériale des Sciences de Saint-Pétersbourg (6th. ser) Sciences Math. et Phys., vol. VI pp. 203-232). Oeuvres (vide supra) vol. I pp. 171-200.

(Chebyshev,1860) Chebyshev(Tchebichef),P.L., Sur l'intégration de la différentielle $(x + A)dx/\sqrt{x^4 + ax^3 + bx^2 + cx + d}$. Bulletin de l'Académie Impériale de Saint-Pétersbourg III(1861) pp. 1-12. (Summarised in Comptes Rendus de l'Académie des Sciences LI(1860) p.46-48) (Reprinted with Summary in Journal de Maths. Pures et Appl. (2nd. ser) 9(1864) pp. 225-246). Oeuvres (vide supra) vol. I pp. 517-530.

(Chevalley,1951) Chevalley,C., Introduction to the Theory of Algebraic Functions of one Variable, A.M.S. Surveys VI, 1951.

(Churchhouse,1976) Churchhouse,R.F., Efficient Computation of Algebraic Continued Fractions. Astérisque 38-39 (1976) pp. 23-32.

(Coates,1970) Coates,J., Construction of Rational Functions on a Curve. Proc. Cam. Phil. Soc. 68(1970) pp. 105-123.

(Cohen & Yun,1979) Cohen,J.D., & Yun,D.Y.Y., Algebraic Extensions of Arbitrary Integral Domains. Proc EUROSAM 79 pp. 134-139.

(Collins,1971) Collins,G.E., The Calculation of Multivariate Polynomial Resultants. JACM 18(1971) pp. 515-532.

(Davenport,1979a) Davenport,J.H., The Computerisation of Algebraic Geometry. Proc. EUROSAM 79 pp. 119-133.

(Davenport,1979b) Davenport,J.H., Algorithms for the Integration of Algebraic Functions. Proc EUROSAM 79 pp. 415-425.

(Davenport,1979c) Davenport,J.H., Anatomy of an Integral. SIGSAM Bulletin, November 1979.

(Davenport & Jenks,1980) Davenport,J.H. & Jenks,R.D., MODLISP - an Introduction. Proc. LISP80, The LISP Company, Stanford, California, 1980.

(Demjanenko,1971) Demjanenko,V.A., On the Torsion of Elliptic Curves. Izv. Akad. Nauk. SSSR Ser. Mat. 35(1971) pp. 280-307 (MR 44(1972) #2755).

(Eichler,1966) Eichler,M., Introduction to the Theory of Algebraic Numbers and Functions. Academic Press, London, 1966.

(Epstein,1979) Epstein,H.I., A Natural Structure Theorem for Complex Fields. SIAM J. Computing 8(1979) pp. 320-325.

(ffitch & Norman,1977) ffitch,J.P. & Norman,A.C., Implementing LISP in a High-level Language. Software - Practice and Experience, 7(1977) pp. 713-725.

(Fulton,1969) Fulton,W., Algebraic Curves, An Introduction to Algebraic Geometry. W.A. Benjamin Inc, 1969.

(Griesmer et al.,1975) Griesmer,J.H., Jenks,R.D. & Yun,D.Y.Y., SCRATCHPAD User's Manual. IBM Research Publication RA70, June 1975.

(Griffiths & Harris,1978) Griffiths,P. & Harris,J., On Cayley's Explicit Solution to Poncelet's Porism. L'Enseignement Mathématique (2nd. Series) 24(1974) pp. 31-40.

(Halphen,1886) Halphen,G.H., Traité des fonctions elliptiques et de leurs applications. Paris, 1886-1891.

(Hardy,1916) Hardy,G.H., The Integration of Functions of a Single Variable (2nd. ed.). Cambridge Tract 2, C.U.P.,1916.

(Harrington,1977) Harrington,S.J., A new Symbolic Integration System in Reduce. Utah Computational Physics Report 57, University of Utah, Nov. 1977 (revised May 1978).

(Harrington,1979a) Harrington,S.J., A Symbolic Limit Evaluation Program in REDUCE. SIGSAM Bulletin 49 (Feb. 1979) pp.27-31.

(Harrington,1979b) Harrington,S.J., A new Symbolic Integration System in REDUCE. Computer Journal 22(1979) 2 pp. 127-131.

(Hearn,1973) Hearn,A.C., REDUCE-2 User's Manual. Computing Physics Group, University of Utah, 1973.

(Hearn,1976) Hearn,A.C., A New Reduce Model for Algebraic Simplification. Proc. SYMSAC 76, pp.46-51.

(Hearn,1979) Hearn,A.C., Non-Modular Computation of Polynomial Gcd Using Trial Division. Proc. EUROSAM 79 pp.227-239.

(Jenks,1979) Jenks.R.D., MODLISP. Proc. EUROSAM 79 pp. 466-480.

(Kaplansky,1957) Kaplansky,I., An Introduction to Differential Algebra, Publications de l'Institut de Mathématique de l'Université de Nancago V, Hermann, Paris, 1957 (2nd. ed. 1976).

(Kenku,1979) Kenku,M.A., Certain Torsion Points on Elliptic Curves defined over Quadratic Fields. J. L.M.S. 2nd. ser. 19(1979) pp. 233-240.

(Kolchin,1973) Kolchin,E.R., Differential Algebra and Algebraic Groups, Academic Press, London, 1973.

(Lang,1957) Lang,S. Introduction to Algebraic Geometry. Interscience, New York, 1957.

(Lang,1959) Lang,S., Abelian Varieties. Interscience, New York, 1959.

(Lang,1960) Lang,S., Integral Points on Curves, Publ. Math. IHES 6(1960), pp. 319-335.

(Lang,1972) Lang,S., Introduction to Algebraic and Abelian Functions. Addison-Wesley, 1972.

(Lang,1978) Lang,S., Elliptic Curves Diophantine Analysis. Springer-Verlag, Berlin-Heidelberg-New York, 1978.

(Lang & Neron,1959) Lang,S. & Neron,A., Rational Points of Abelian Varieties over Function Fields. American J. of Math. 81(1959) pp. 95-118.

(Laplace,1820) Laplace,(P.S.) Marquis de, Théorie Analytique des Probabilités, 3rd. ed., Courcier, Paris, 1820. In: Laplace, Oeuvres complètes du Marquis de, Gauthier-Villars, Paris, 1886, vol. 7 (of 14); or Laplace, Oeuvres de, Imprimerie Royale, Paris, 1847, vol. 7 (of 7).

(Liouville,1833a) Liouville,J., Premier Mémoire sur la Détermination des Intégrales dont la Valeur est Algébrique. Journal de l'Ecole Polytechnique 14(1833) cahier 22, pp. 124-148.

(Liouville,1833b) Liouville,J., Second Mémoire sur la Detérmination des Intégrales dont la Valeur est Algébrique. Journal de l'Ecole Polytechnique 14(1833) cahier 22, pp. 149-193.

(Liouville,1833c) Liouville,J., Mémoire sur les Transcendentes Elliptiques de Première et Seconde Espèce, Considerées comme Fonctions de leur Amplitude, Journal de l'Ecole Polytechnique 14(1833) cahier 23, pp. 57-83.

(Lipson,1969) Lipson,J.D., Symbolic Methods for the Computer Solution of Linear Equations with Applications to Flow-Graphs. Proc. 1968 Summer Institute on Symbolic Mathematical Computation. IBM, Yorktown Heights, 1969, pp. 233-303.

(Manin,1958) Manin,Ju.I., Algebraic Curves over Fields with Differentiation. Izv. Akad. Nauk. SSSR Ser. Mat. 22(1958) pp. 737-756 (translated in AMS Trans. Ser. 2 37(1964) pp. 59-78).

(Manin,1963) Manin,Ju.I., Rational Points of Algebraic Curves over Function Fields. Izv. Akad. Nauk. SSSR Ser. Mat. 27(1963) pp. 1395-1440 (translated in AMS Trans. Ser. 2 50(1966) pp. 189-234).

(Manin,1969) Manin,Ju.I., Uniform bounds for p-torsion on elliptic curves. Izv. Akad. Nauk. SSSR 33(1969) pp. 459-465. (MR 42 #7667.) See also Serre,1971.

(Mazur,1977) Mazur,B., Rational Points on Modular Curves, in Modular Functions of One Variable V, Springer Lecture Notes in Mathematics 601, Berlin-Heidelberg-New York 1977 (Proceedings International Conference on Modular Functions, Bonn 1976) pp. 107-148.

(Mazur,1978) Mazur,B., Rational Isogenies of Prime Degree. Inventiones Math. 44(1978) pp. 129-162.

(Mazur & Swinnerton-Dyer,1974) Mazur,B. & Swinnerton-Dyer, H.P.F., Arithmetic of Weil Curves. Inventiones Math. 25 (1974) pp. 1-61.

(Mignotte,1976) Mignotte,M., Factorisation des Polynômes sur un Corps Fini. Astérisque 38-39 (1976) pp. 149-157

(Mordell,1922) Mordell,L.J., On the Rational Solutions of the Indeterminate Equations of the Third and Fourth Degrees. Proc. Cam. Phil. Soc. 21(1922) pp. 179-192.

(Moses,1967) Moses,J., Symbolic Integration. Project MAC report 47, M.I.T., 1967.

(Moses,1971) Moses,J., Symbolic Integration, the stormy decade. Communications ACM 14(1971) pp. 548-560.

(Mumford,1965) Mumford,D., Geometric Invariant Theory. Springer-Verlag, Berlin-Heidelberg-New York, 1965.

(Ng,1974) Ng E.W., Symbolic Integration of a Class of Algebraic Functions. NASA Technical Memorandum 33-713 (1974).

(Norman,1975) Norman,A.C., Computing with Formal Power Series. ACM Transactions on Mathematical Software 1(1975) pp. 346-356.

(Norman,1978a) Norman,A.C., Symbolic and Algebraic Modes in REDUCE. REDUCE Newsletter 3 (July 1978) pp. 5-9.

(Norman,1978b) Norman,A.C., Towards a REDUCE Solution to SIGSAM Problem 7. SISGAM Bulletin 48 (Nov. 1978) pp. 14-18.

(Norman & Davenport,1979) Norman,A.C., & Davenport,J.H., Symbolic Integration - the Dust Settles? Proc. EUROSAM 79 pp. 398-407.

(Norman & Moore,1977) Norman,A.C. and Moore,P.M.A., Implementing the New Risch Integration Algorithm. Proc. 4th. Int. Colloquium on Advanced Computing Methods in Theoretical Physics, Marseilles, 1977.

(Richardson,1968) Richardson,D., Some Unsolvable Problems Involving Elementary Functions of a Real Variable. Journal of Symbolic Logic 33(1968), pp. 511-520.

(Risch,1969) Risch,R.H., The Problem of Integration in Finite Terms. Trans. A.M.S. 139(1969) pp. 167-189 (MR 38 #5759).

(Risch,1970) Risch,R.H., The Solution of the Problem of Integration in Finite Terms. Bulletin AMS 76(1970) pp. 605-608.

(Risch,1974) Risch,R.H., A Generalization and Geometric Interpretation of Liouville's Theorem on Integration in Finite Terms. IBM Research Report RC 4834 (6 May 1974).

(Ritt,1948) Ritt,J.F., Integration in Finite Terms, Liouville's Theory of Elementary Methods. Columbia University Press, New York, 1948.

(Ritt,1950) Ritt,J.F., Differential Algebra. American Mathematical Society Colloquium Proceedings vol. XXXIII, Providence R.I., 1950 (reprinted Dover, New York, 1966).

(Rosenlicht,1976) Rosenlicht,M., On Liouville's Theory of Elementary Functions. Pacific J. Math 65(1976), pp. 485-492.

(Rothstein,1977) Rothstein,M., A New Algorithm for the Integration of Exponential and Logarithmic Functions. Proc. 1977 MACSYMA Users' Conference, pp. 263-274.

(Rothstein & Caviness,1979) Rothstein,M. & Caviness,B.F., A Structure Theorem for Exponential and Primitive Functions. SIAM J. Computing (to appear).

(Schinzel,1962) Schinzel,A., On Some Problems of the Arithmetical Theory oF Continued fractions II. Acta Arithmetica VII (1962) pp. 287-298.

(Seidenberg,1968) Seidenberg,A., Elements of the Theory of Algebraic Curves. Addison-Wesley, 1968.

(Serre,1971) Serre,J.-P., p-torsion des courbes elliptiques (d'après Y. Manin). Séminaire Bourbaki 69/70 no. 380. Springer Lecture Notes in Mathematics 180 (Berlin-Heidelberg-New York, 1971).

(Serre & Tate,1968) Serre,J.-P. & Tate,J.T., Good Reduction of Abelian Varieties. Annals of Mathematics 88 (1968) pp. 492-517.

(Shafarevich,1972) Shafarevich,I.R., Osnovy Algebraicheskoi Geometrii, Nauka, Moscow, 1972. English translation: Basic Algebraic Geometry, Springer-Verlag, Berlin-Heidelberg-New York,1974.

(Shimura & Taniyama,1961) Shimura,G. & Taniyama,A. Complex Multiplication of Abelian Varieties and its Applications to Number Theory. Publ. Math. Soc. Japan 6(1961).

(Shtokhamer,1977) Shtokhamer,R., Attempts in Local Simplification of Non-Nested Radicals. SIGSAM Bulletin 41 (Feb. 1977) pp. 20-21.

(Siegel,1929) Siegel,C.L., Ueber einige Anwendungen Diophantischer Approximationen. Abh. Preuss. Akad. Wiss. (1929) pp. 41-69.

(Siegel,1969) Siegel,C.L., Abschaetzung von Einheiten. Nachr. Akad. Wiss. Goettingen (1969) pp. 71-86.

(Slagle,1961) Slagle,J., A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus. Ph.D. Dissertation, Harvard U., Cambridge, Mass. May 1961.

(Smit,1976) Smit,J., The Efficient Calculation of Symbolic Determinants. Proc SYMSAC 76, pp. 105-113.

(Smit,1979) Smit,J., New Recursive Minor Expansion Algorithms, a Presentation in a Comparative Context. Proc. EUROSAM 79, pp. 74-87

(Stephens,1970) Stephens,N.M., Notes on the Algorithm of Birch & Swinnerton-Dyer. Unpublished, 1970.

(Stoutemyer,1977) Stoutemyer,D.R., sin(x)**2 + cos(x)**2 = 1. Proc 1977 MACSYMA Users' Conference, pp. 425-433.

(Swinnerton-Dyer,1975) Swinnerton-Dyer,H.P.F., Numerical Tables on Elliptic Curves. In: Modular Functions of One Variable IV (Proceedings Antwerp 1972), Springer Lecture Notes in Mathematics 476(Berlin-Heidelberg-New York, 1975).

(Tate,1974) Tate,J.T., The Arithmetic of Elliptic Curves. Inventiones Math. 23(1974) pp. 179-206.

Tchebichef - see Chebyshev.

(Trager,1976) Trager,B.M., Algebraic Factoring and Rational Function Integration. Proc. SYMSAC 76, pp. 219-226.

(Trager,1978) Trager,B.M., IBM Yorktown Heights Integration Workshop, 28-9 Aug. 1978. (Tape recording available from Dr. D.Y.Y. Yun)

(Trager,1979) Trager,B.M., Integration of Simple Radical Extensions. Proc. EUROSAM 79 pp. 408-414.

(van der Waerden,1949) van der Waerden,B.L., Modern Algebra, Frederick Ungar, New York, 1949 (Translated from Moderne Algebra 2nd. ed.).

(Wang,1976) Wang,P.S., Factoring Multivariate Polynomials over Algebraic Number Fields. Math. Comp. 30(1976) pp. 324-336.

(Wang,1978) Wang,P.S., An Improved Multivariable Polynomial Factorising Algorithm. Math. Comp. 32(1978) pp. 1215-1231.

(Wang & Minamikawa,1976) Wang,P.S. & Minamikawa,T., Taking Advantage of Zero Entries in the Exact Inverse of Sparse Matrices. Proc. SYMSAC 76, pp. 346-350.

(Wang & Rothschild,1975) Wang,P.S. & Rothschild,L.P., Factoring Multi-Variate Polynomials over the Integers. Math. Comp. 29(1975) pp. 935-950.

(Weil,1928) Weil,A., L'Arithmétique sur les Courbes Algébriques. Acta. Math. 52(1928) pp. 281-315.

(Wienberger & Rothschild,1976) Factoring Polynomials over Algebraic Number Fields. ACM Transactions on Mathematical Software 2(1976) pp. 335-350.

(Weiss,1963) Weiss,E., Algebraic Number Theory. McGraw-Hill, New York, 1963.

(Whittaker & Watson,1927) Whittaker,E.T., & Watson,G.N., A Course of Modern Analysis. C.U.P. 4th. ed 1927.

(Yun,1973) Yun,D.Y.Y., The Hensel Lemma in Algebraic Manipulation. M.I.T. Thesis MAC TR-138, 1973.

(Yun,1976) Yun,D.Y.Y., Algebraic Algorithms using *p*-adic Techniques. Proc. SYMSAC 76, pp. 248-259.

(Yun,1977a) Yun,D.Y.Y., Fast Algorithm for Rational Function Integration. IBM Research Report RC 6563 (6 Jan. 1977). Proceedings IFIP 77, Toronto, Canada.

(Yun,1977b) Yun,D.Y.Y., On the Equivalence of Polynomial Gcd and and Squarefree Factorization Algorithms. Proc. 1977 MACSYMA Users' Conference, pp. 65-70.

(Yun & Gustavson,1979) Yun,D.Y.Y., & Gustavson,F., Fast Computation of the Rational Hermite Interpolant and Solving Toeplitz Systems of Equations via the Extended Euclidean Algorithm. Proc. EUROSAM 79, pp. 58-65.

(Zimmer,1972) Zimmer,H.G., Computational Problems, Methods, and Results in Algebraic Number Theory. Lecture Notes in Mathematics 262, Springer-Verlag, Berlin-Heidelberg-New York, 1972.

(Zimmer et al.,1979) Bartz,H., Fischer,K., Folz,H. and Zimmer,H.G., Some Computations relating to Torsion Points of Elliptic Curves over Algebraic Number Fields. Proc. EUROSAM 79 pp. 108-118.

(Zippel,1977) Zippel, R.E.B., Radical Simplification Made Easy, Proc. 1977 MACSYMA Users' Conference, pp. 361-367.

(Zippel,1979) Zippel,R.E.B., Probabilistic Algorithms for Sparse Polynomials. Proc. EUROSAM 79, pp. 216-226.

Список книг, имеющихся на русском языке

Боревич З. И., Шафаревич И. Р. Теория чисел. 2-е изд. — М.: Наука, 1972.
ван дер Варден Б. Л. Алгебра. — М.: Наука, 1976.
Демьяненко В. А. О кручении эллиптических кривых. — Изв. АН СССР, сер. мат., т. 35, 1971, с. 280—307.
Капланский И. Введение в дифференциальную алгебру. — М.: ИЛ, 1959.
Ленг С. Введение в алгебраические и абелевы функции. — М.: Мир, 1976.
Мамфорд Д. Геометрическая теория инвариантов. — В кн.: Дьёдонне Ж., Керрол Дж., Мамфорд Д. Геометрическая теория инвариантов. — М.: Мир, 1974.
Манин Ю. И. Алгебраические кривые над полями с дифференцированием. — Изв. АН СССР, сер. мат., т. 22, 1958, с. 737—756.
Манин Ю. И. Рациональные точки алгебраических кривых над функциональными полями. — Изв. АН СССР, сер. мат., т. 27, 1963, с. 1395—1440.
Манин Ю. И. p-кручение эллиптических кривых равномерно ограничено. — Изв. АН СССР, сер. мат., 1969, с. 459—465.
Уиттекер Э. Т., Ватсон Г. Н. Курс современного анализа. — М.: ГТТИ, ч. I, 1933; ч. II, 1934.
Чебышев П. Л. Об интегрировании иррациональных дифференциалов. — В кн.: Чебышев П. Л. Избранные математические труды. — М.: ГОНТИ, 1946.
Шафаревич И.Р. Основы алгебраической геометрии. — М.: Наука, 1972.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Абелева группа свободная 31
Абелево многообразие 73
Абсолютная неприводимость 22
Алгебраическая геометрия 26
— группа 73
— кривая 26
— функция 51
— часть интеграла 54
Алгебраические выражения 19, 20, 113
— — единственность 21, 22
— — представление *см.* Представление алгебраических выражений
— соотношения 19
Алгебраическое замыкание 86
— многообразие 73
Алгоритм АЛГ_ФАКТОРИЗАЦИЯ 168
— АЛГ_ФАКТОРИЗАЦИЯ 2 169
— БЕСКВ_НОРМА 167—168
— ВХОЖДЕНИЯ 107
— ГРАНИЦА_КРУЧЕНИЯ (вариант 1) 103, 104
— ГРАНИЦА_КРУЧЕНИЯ (вариант 2) 107, 108
— ДИВИЗОР_В_ФУНКЦИЮ 46, 47
— ЗНАМЕНАТЕЛЬ_АЛГЕБРАИЧЕСКОГО 99, 100
— КОНЕЧНЫЙ_ПОРЯДОК_ЭЛЛИПТИЧЕСКИЙ 97, 98
— КОУТС 35, 36
— КОУТСА 14, 34, 35, 120, 121
— ЛЮТЦ_НАГЕЛЬ 95, 96
— МАКСИМАЛЬНАЯ_СТЕПЕНЬ 102
— НАХОЖДЕНИЕ_АЛГЕБРАИЧЕСКОЙ_ЧАСТИ 61, 62
— НАХОЖДЕНИЕ_ПОРЯДКА_ПО_МАНИНУ 82—84
— ПРИМИТИВНЫЙ_ЭЛЕМЕНТ 169—170
— РАЗЛОЖЕНИЕ_БЕСКВАДРАТНОЕ 171, 172
— РЕДУКЦИЯ_К_НОРМАЛЬНОМУ_БАЗИСУ 38, 39
— РЕДУКЦИЯ_К_ЦЕЛОМУ_БАЗИСУ 37, 38
— РИШ_АЛГЕБРАИЧЕСКИЙ 55, 56
— ФОРМА_ВЕЙЕРШТРАССА 67, 68

Базис 35
— нормальный целый 42
— целый 35, 40
Бирациональные преобразования 27

Вложенные выражения 153
Высота алгебраического числа 93
Вычет дифференциала 33

Граница *Мазура* 70, 138
Группа кручения 65

Дефект кривой *см.* Род кривой
Дивизор 31
— главный 31
— дифференциала 32

— линейно эквивалентный нулю 31
— не эквивалентный нулю 31
— рационально эквивалентный нулю 31
— с кручением 31, 65
— функции 31
— эффективный 31
Дифференциал 32
— второго рода 33
— первого рода 32
Дифференциальная алгебра 52
Дифференциальное поле 53
— расширение поля 53

Идеал неразветвленный 92
— разветвленный 92
Индекс ветвления 28, 39, 92
Интеграл *Чебышева* (пример) 142—143
— чисто алгебраический 54
— элементарный 51

Каноническая форма 19
Кольцо классов вычетов 92
Компьютерная алгебра 11
Компьютерное интегрирование 11, 12
Кратная точка 27
— — обычная 27
Кривые бирационально эквивалентные 27
Кручение 65

Логарифм 53
Логарифмическая часть интеграла 54, 122
Логарифмически неинтегрируемая функция 154, 155
Локальный параметр 28

Метод *Кэли* 71, 72
Модуль 35
Модульный подход 101

Не-*p*-часть группы 102
Неинтегрируемость алгебраическая 124
— логарифмическая 124
Ненормированные полиномы 21
Неособая модель кривой 27

Оператор *Гаусса — Манина* 79
— *Пикара — Фукса* 79

Параметр *Гаусса — Манина* 85
Плейс 14, 26
— базисный 25, 29
— лежащий над 26
— неразветвленный 39
— разветвленный 28
— центрированный 26
Поле алгебраических чисел 39, 90
— классов вычетов 102
— характеристики 0 53
Полюс 35
Порядок дифференциала 32
— функции в плейсе 28
Представление алгебраических выражений 23
— — — примитивность 23
— — — с многими переменными 23, 24
Преобразование кривых 27
— — бирациональное 27
Преобразования подынтегрального выражения 117
Примитивный элемент 23, 39
Принцип *Лапласа* 58
Простое радикальное расширение 17

Разложение *Пюизо* 14, 28, 30
— — вычисление 30
Расширение на трансцендентные функции 121, 124

Рационализирующая подстановка 32
Рациональная часть интеграла 122
Рациональное простое число 91
Редукционный шаг 41
— — на бесконечности 42
Результат *Мазура* 69
Решение линейных уравнений 115—117
Род кривой 32
Ряд *Лорана* 28

Система REDUCE-2 17, 18
— — изменения 126—131
— SAINT 16
— SIN 16
Специализация 102
Список 28, 29
Степень дивизора 31
Структура данных 29
Сумма двух функций (пример) 163

Теорема *Лютца — Нагеля* 94, 114
— *Риша* 54, 55

Униформизующая переменная *см.* Локальный параметр
Уравнение *Пикара — Фукса* 79

Целая точка кривой 93
Целые элементы поля 91

Хорошая редукция 102

Экспонента 53
Элементарное расширение 54
Эллиптическая кривая 33, 65, 66
Эллиптические интегралы 124

Якобиева группа дивизоров 65
Якобиево многообразие 74

IBM 730/165 17
IBM 730/168 85
K/*k*-след многообразия 81
p-часть группы 102

ОГЛАВЛЕНИЕ

Предисловие редактора перевода 5
От автора 9

Глава 1. Введение 11

Компьютерное интегрирование вообще 11
План книги (1) 12
Пример 13
План книги (2) 13
Теоретические ограничения 15
Краткий обзор предшествующих работ 16
Учет машинного времени 17

Глава 2. Алгебраические вычисления 19

Алгебраические соотношения 19
Единственность алгебраических выражений 21
Представление алгебраических выражений 23
Соображения, связанные с реализацией 25
Алгебраическая геометрия 26
Разложения Пюизо 28
Структуры данных для плейсов 28
Вычисление разложений Пюизо 30
Дивизоры 31
Дифференциалы 32

Глава 3. Алгоритм Коутса 34

Введение 34
Описание алгоритма 35
Доказательство корректности алгоритма. Шаги [1]—[3] . . . 39
Доказательство корректности алгоритма. Шаги [4]—[5] . . . 42
Обобщения 43
Алгоритм Коутса и дифференциалы 43
Реализация 49
Выводы 50

Глава 4. Теорема Рнша 51

Введение 51
Дифференциальная алгебра 52
Теорема Риша 54
Доказательство теоремы Риша 57
Алгебраическая часть 60

Частичный алгоритм 62
Соображения эффективности 63

Глава 5. Задача о дивизорах с кручением 64

Введение 64
Эллиптические кривые 65
Результат Мазура 69
Приложение исследований Мазура 70
Метод Кэли 71
Якобиевы многообразия 73

Глава 6. Операторы Гаусса — Манина 75

Введение 75
Пример 76
Уравнения Пикара — Фукса 78
Операторы Пикара — Фукса как гомоморфизмы 80
Дивизоры конечного порядка 81
Реализация 84
Специальные значения параметров 86

Глава 7. Эллиптические интегралы. Окончание 90

Поля алгебраических чисел 90
Эллиптические кривые 92
Теория Лютца — Нагеля 93
Точка зрения Харди 96
Реализация 98

Глава 8. Кривые над полями алгебраических чисел 101

Произвольный род 101
Хорошая редукция 102
Кручение над конечными полями 104
Пример 106
Вычислительные соображения 108
Пример над алгебраическими полями 109

Глава 9. Выводы 112

Состояние теории 112
Состояние реализации 112
Необходимые дополнительные процедуры 114
Выводы для системы алгебраических вычислений 118
Будущие теоретические исследования 120
Расширение на трансцендентные функции 121
Виды неинтегрируемости 124
Резюме 125

Приложение 1. Изменения, внесенные в систему REDUCE-2 126

1. Распечатка 126
2. Дифференцирование 127
3. Наибольшие общие делители 128
4. Алгебраические выражения 129

5. Разложение на множители 130
6. Единственность алгебраических выражений 131

Приложение 2. Примеры 132

Пример 1. Простое логарифмическое выражение 132
Пример 2. 1/SQRT ((X**2 — 1)*(X**2 — K**2)) 133
Пример 3. Пример с кручением 137
Пример 4. Модулярная кривая 139
Пример 5. Интеграл Чебышева 142
Пример 6. Вложенные выражения 153
Пример 7. Логарифмически неинтегрируемая функция 154
Пример 8. Сумма двух функций 163

Приложение 3. Алгоритмы работы с алгебраическими выражениями 167

Литература 174

Предметный указатель 185

УВАЖАЕМЫЙ ЧИТАТЕЛЬ!

Ваши замечания о содержании книги, ее оформлении, качестве перевода и другие просим присылать по адресу: 129820, Москва, И-110, ГСП, 1-й Рижский пер., д. 2, издательство «Мир».