ISSN 1810-0198. Вестник Тамбовского университета. Серия Естественные и технические науки

Том 23, № 121 2018

УДК
DOI: 10.20310/1810-0198-2018-23-121-xx-xx

Алгоритмы символьного интегрирования в системе MathPartner

© В. А. Корабельников

ФГБОУ ВО «Тамбовский государственный университет им. Г.Р. Державина»
392000, Российская Федерация, г. Тамбов, ул. Интернациональная, 33
E-mail: korabelnikov.va@gmail.com

Аннотация. Начало созданию библиотеки процедур для символьного интегрирования положила теорема Риша, опубликованная в 1969 году. Однако за прошедшие почти 50 лет такая библиотека все еще не создана. Известно только несколько попыток создания подобных библиотек, но не одна из них не была завершена. В системе компьютерной математики MathPartner строится новая библиотека процедур для символьного интегрирования, в основе которой лежит теорема Риша. Мы даем подробное описание основных процедур, составляющих эту библиотеку, и роль каждой из них в алгоритме символьного интегрирования. Мы приводим уточненную процедурную блок-схему всего алгоритма и примеры вычисленных интегралов.
Ключевые слова: алгоритм Риша; символьное интегрирование; неопределенный интеграл; система MathPartner; дифференциальное поле; элементарные функции

Введение

Алгоритм вычисления производной для произвольной композиции элементарных функций хорошо известен, а соответствующие программы давно вошли во все библиотеки, которые поддерживают символьные вычисления. Для обратной операции, операции вычисления первообразной функции по ее производной, ситуация значительно более сложная. Только для некоторых видов функций имеются подробные алгоритмы аналитических преобразований, позволяющих найти неопределенный интеграл. Причем каждому виду функций соответствует своя последовательность таких преобразований. Такие алгоритмы очень громоздки и применимы для небольшого числа классов

Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект No 16-07-00420.)

функций. В большинстве случаев указанные недостатки делают такой алгоритм непригодным для практического применения. [2]

Можно выделить один важный класс — класс дробно-рациональных функций. Алгоритм интегрирования дробно-рациональных функций был известен еще во времена Ньютона, Лейбница и Бернулли. Так как интеграл от дробно-рациональной функции всегда существует, то этот алгоритм позволяет проинтегрировать любую дробно-рациональную функцию. Главный недостаток этого алгоритма — это необходимость разложения знаменателя подинтегральной функции на линейные множители. В случае, если знаменатель подинтегрального выражения не раскладывается на линейные множители над кольцом $Q(x)$, то потребуется добавлять к этому кольцу константы из R или C, даже если интеграл содержится в кольце $Q(x)$. В 19 веке Эрмит улучшил этот алгоритм, уменьшив количество добавляемых констант. [4]

Систематическое изучение случаев, когда интеграл может быть выражен в конечном виде, начались в начале 19 века. В 1820 году Лаплас заметил, что интеграл функции не содержит радикалов, кроме тех, что находятся в подинтегральном выражении. Около десятилетия спустя Лиувилль сформулировал и доказал более строгую и более точную теорему, которая, грубо говоря, утверждает что если интеграл элементарной функции является элементарным, то он может быть выражен с использованием только функций которые появляются в подынтегральном выражении и линейной комбинацией логарифмов таких функций. Эта теорема теперь известна как теорема Лиувилля, или принцип Лиувилля.

В 1969 году Риш использовал теорему Лиувилля чтобы описать алгоритм, который находит элементарное выражение для интеграла от элементарной функции если оно существует. Если же не удалось вычислить первообразную, то это должно означать, что первообразная не является композицией элементарных функций. [11]

Алгоритм Риша являлся на время своего опубликования алгоритмом интегрирования лишь чисто трансцендентных функций. Для алгебраических составляющих Риш привёл в общих чертах доказательство существования алгоритма, но чтобы сделать из него алгоритм, потребовался ещё не один десяток лет. [10]

Первая программа по алгоритму Риша была написана Джоэлом Мозесом в 1971 г. Программа под названием SIN интегрировала чисто трансцендентные функции. Дэвенпорт в 1981 г., основываясь на работе Риша, разработал алгоритм интегрирования чисто алгебраических функций. Программы Мозеса и Дэвенпорта служили для решения частных случаев, но для общего случая — интегрирования произвольных элементарных функций — были одинаково непригодны. [10]

Алгоритм Дэвенпорта обладал большой вычислительной сложностью. Барри Трагер в 1984 г. внёс серьёзные улучшения в алгоритм Дэвенпорта. Обновлённый алгоритм обладал гораздо большей практической ценностью и был реализован в системах компьютерной алгебры Axiom и Maple. [10]

Решающий шаг к практическому решению вопроса сделал Мануэль Бронштейн в 1990 г, обобщив алгоритм Трагера на произвольные элементарные функции. В 1998 г. Бронштейн написал монографию по символьному интегрированию, в которой понятным языком изложил лучшие достижения в этой области, начиная с теоремы Лиувилля и заканчивая собственными результатами. [10]

Полная реализация алгоритма Риша-Трагера-Бронштейна на данное время не завершена. К тому же в теории символьного интегрирования остались открытые вопросы.

Во первых алгоритм Риша (даже для чисто трансцендентных функций) не является "честным"алгоритмом. Дело в том, что в процессе работы алгоритм Риша выстраивает "башню"элементарных расширений поля рациональных функций, с тем чтобы "верхушка башни"содержала подынтегральное выражение. При этом не всегда понятно, готова ли уже "башня"или нужно добавить в неё ещё один логарифм, экспоненту, или алгебраическую функцию. Например, если в подынтегральном выражении встретилась экспонента $\exp(x)$, она добавляется в поле (происходит элементарное расширение поля функцией $\exp(x)$). Если при дальнейшем анализе в подынтегральном выражении встречается $\exp(x^2)$, то эту функцию также следует добавить в поле, вырастив таким образом "башню"до двух этажей. Но если после этого попадётся функция $\exp(x+x^2)$, то алгоритм должен определить, что она принадлежит сформированному на данный момент элементарному расширению, и не наращивать "башню". В общем случае эта задача очень сложна и нынешние реализации алгоритма Риша используют лишь эвристики для её решения, которые по определению ненадёжны. [10]

Как указывают Дэвенпорт и Бронштейн, при реализации алгоритма можно предположить, что нам известен так называемый базис трансцендентности для поля констант, то есть, например, считать e и $\sqrt{2}$ алгебраически независимыми. Это значит, что мы принимаем на веру, что нет никакого полинома от e и $\sqrt{2}$, который был бы равен 0. Такое решение нужно принимать для каждой новой константе, появляющейся в ходе работы алгоритма, которую эвристика не позволяет связать алгебраическим выражением с ранее учтёнными константами. Если полученный базис трансцендентности окажется неверным, это может отразиться на корректности выданного ответа в том случае, если ответ гласит, что "интеграл не является элементарной функцией". Если же интеграл найден, его можно считать правильным. [10]

Известным частным случаем задачи об алгебраической независимости является определение того, тождественно ли нулю некоторое математическое выражение. Задача тривиально решается для многочленов, но для более широкого класса функций, как показал Даниэль Ричардсон в 1968 г., эта задача неразрешима. [10]

Целью данной работы является описание алгоритмов символьного интегрирования, которые сегодня применяются в системе компьютерной алгебры MathPartner. Это еще один шаг на пути к созданию доказательного алгоритма вычисления первообразной, без эвристик. Под "доказательным"понимается такой алгоритм, который либо вычисляет интеграл, как композицию элементарных функций, либо возвращает ответ - "функция не интегрируема в элементарных функциях". Данная работа описывает текущее состояние создаваемой библиотеки процедур символьного интегрирования на пути решения это трудной фундаментальной задачи.

1. Основные понятия

Сначала введем определения, необходимые для описания алгоритма Риша. При этом мы будем придерживаться такого варианта изложения, который был принят в работе [3].

Определение 1.1. Дифференциальным полем называется поле K, на котором действует оператор дифференцирования d, удовлетворяющий следующим условиям:

$$d(ab) = d(a)b + ad(b)$$

$$d(a+b) = d(a) + d(b)$$

Элемент $d(a)$ называется производной элемента $a \in K$ и обычно обозначается a'.

Определение 1.2. Пусть K — дифференциальное кольцо (поле). Множество $C = \{a \in K : d(a) = 0\}$ образует подкольцо (подполе) кольца (поля) K, называемое подкольцом (подполем) констант.

Определение 1.3. Пусть K — дифференциальное поле с дифференцированием d. Расширение E поля K называется дифференциальным расширением дифференциального поля K, если на E определено дифференцирование d_1, ограничение которого на K совпадает с d. Дифференцирование d_1 называется продолжением дифференцирования d и обозначается, если это не приводит к двусмысленности, тем же символом d.

Определение 1.4. Пусть E — дифференциальное расширение дифференциального поля K. Элемент $\theta \in E$ называется логарифмом над K, если θ удовлетворяет дифференциальному уравнению $f\theta' = f'$ для некоторого ненулевого элемента $f \in K$ (обозначается $\theta = log(f)$).

Определение 1.5. Пусть E — дифференциальное расширение дифференциального поля K. Элемент $\theta \in E$ называется экспонентой над K, если θ удовлетворяет дифференциальному уравнению $\theta' = f'\theta$ для некоторого ненулевого элемента $f \in K$ (обозначается $\theta = exp(f)$).

Теорема 1.1 (принцип Лиувилля)**.** *Пусть K — некоторое дифференциальное поле, C — его поле констант и $f \in K$. Пусть $g(x)$ — элементарная над K функция, удовлетворяющая уравнению $g'(x) = f(x)$. Тогда $g(x)$ можно представить в виде*

$$g(x) = v_0 + \sum_i c_i \log(v_i(x)),$$

где c_i — константы из алгебраического замыкания поля C, $v_0(x) \in K$, $v_i(x) \in K_0$, где K_0 — некоторое расширение поля K, получающееся присоединением к нему конечного числа констант, алгебраических над K.

Определение 1.6. Элемент θ является регулярным мономом над дифференциальным полем K, если θ трансцендентен над K, и является либо логарифмом, либо экспонентой. Последовательность $\theta_1, \theta_2, \dots, \theta_n$ называется последовательностью регулярных мономов, если каждый ее элемент θ_i является регулярным мономом над $K(x, \theta_1, \theta_2, \dots, \theta_{i-1})$, $i = 1, \dots, n$.

Нам нужно уметь конструировать наименьшее дифференциальное поле, содержащее подинтегральную функцию. Для решения этой задачи используется структурная теорема.

Теорема 1.2. *Пусть* K *— поле констант,* $\theta_1, \ldots, \theta_{k-1}$ $(k \geq 0)$ *— последовательность регулярных мономов,* E *— множество индексов* $1 \leq i \leq k-1$*, таких, что* θ_i *является экспонентой* $\theta_i = \exp(f_i)$*, а* L *— множество индексов* $1 \leq i \leq k-1$*, таких, что* θ_i *является логарифмом* $\theta_i = \log(f_i)$*.*

1. Пусть $\theta_k = \exp(f_k)$ *— экспонента над дифференциальным полем* $F_{k-1} = K(x, \theta_1, \ldots, \theta_{k-1})$*,* $f_k \in F_{k-1}$*. Если элемент* θ_k *алгебраичен над* F_{k-1}*, то* f_k *представляется в виде линейной комбинации с рациональными коэффициентами*

$$f_k = c + \sum_{i \in E} n_i f_i + \sum_{j \in L} m_j \theta_j, n_i, m_j \in Q,$$

где c *— некоторая константа.*

2. Пусть $\theta_k = \log(f_k)$ *— логарифм над дифференциальным полем* $F_{k-1} = K(x, \theta_1, \ldots, \theta_{k-1})$*,* $f_k \in F_{k-1}$*. Если элемент* θ_k *алгебраичен над* F_{k-1}*, то* f_k *представляется в виде произведения рациональных степеней*

$$f_k = c \prod_{i \in E} \theta_i^{n_i} \times \prod_{j \in L} f_j^{m_j}, n_i, m_j \in Q,$$

где c *— некоторая константа.*

2. Алгоритм Риша

Пусть x — независимая переменная над полем констант K, $\theta_1, \theta_2, \ldots, \theta_n$ — последовательность регулярных мономов, $K(x, \theta_1, \theta_2, \ldots, \theta_n)$ — дифференциальное поле, и $f \in K(x, \theta_1, \theta_2, \ldots, \theta_n)$. Опишем алгоритм, позволяющий найти неопределенный интеграл функции f, если он является элементарной функцией, или доказать, что в элементарных функциях f неинтегрируема.

Пусть f — подынтегральная функция. Представим эту функцию в виде суммы $f = p + q/r$, где p, q, r имеют следующий вид:

1. Если $f \in K(x)$, то $p, q, r \in K[x]$, и $deg(r) > deg(q)$.

2. Если $f \in K(x, \theta_1, \theta_2, \ldots, \theta_n)$, $\theta_n = \log(u)$ и $u \in K(x, \theta_1, \ldots, \theta_{n-1})$, то p, q, r — полиномы от переменной θ_n с коэффициентами из кольца $K(x, \theta_1, \theta_2, \ldots, \theta_{n-1})$, и $deg(r) > deg(q)$.

3. Если $f \in K(x, \theta_1, \theta_2, \ldots, \theta_n)$, $\theta_n = \exp(u)$ и $u \in K(x, \theta_1, \ldots, \theta_{n-1})$, то $p = \sum_{i=-k_1}^{l_1} p_i \theta_n^i$, $q = \sum_{i=-k_2}^{l_2} a_i \theta_n^i$, $r = \sum_{i=-k_3}^{l_3} b_i \theta_n^i$, где $p_i, a_i, b_i \in K(x, \theta_1, \ldots, \theta_{n-1})$ и выполняется условие $l_3 > l_2$.

2.1. Интегрирование полиномиальной части

Алгоритм интегрирования полиномиальной части сводит вычисление интеграла в кольце $K(x, \theta_1, \theta_2, \ldots, \theta_n)$ к вычислению одного или более интегралов в кольце $K(x, \theta_1, \theta_2, \ldots, \theta_{n-1})$.

Каждый полученный интеграл рекурсивно вычисляется при помощи алгоритма Риша. [6]

Рассмотрим интегрирование обобщенного полинома p. По теореме Лиувилля

$$p = v_0' + \sum_{i=1}^{d} c_i \frac{v_i'}{v_i}$$

Рассмотрим два случая: θ_n — логарифм, и θ_n — экспонента.

1. Пусть $\theta_n = \log(u)$, $u \in K(x, \theta_1, \dots, \theta_{n-1})$. Пусть $p = \sum\limits_{i=0}^{m} p_i \theta_n^i$, где $p_i \in K(x, \theta_1, \dots, \theta_{n-1})$, $i = 0, 1, \dots, m$. Пусть $v_0 = \sum\limits_{i=0}^{m+1} t_i \theta_n^i$, где $t_i \in K(x, \theta_1, \dots, \theta_{n-1})$, $i = 0, 1, \dots, m+1$. Получаем следующее выражение:

$$\sum_{i=0}^{m} p_i \theta_n^i = \sum_{i=0}^{m+1} t_i' \theta_n^i + \sum_{i=0}^{m} (i+1) t_{i+1} \theta_n^i \theta_n' + \sum_{i=1}^{d} c_i \frac{v_i'}{v_i} \quad [6]$$

Старший коэффициент t_{m+1} является константой, обозначим ее a_{m+1}. Для нахождения остальных коэффициентов $t_i, i = m, m-1, \dots, 0$ мы получаем, приравнивая коэффициенты при одинаковых степенях θ_n, систему дифференциальных уравнений:

$$p_i = t_i' + (i+1) t_{i+1} \theta_n' \quad (1)$$

Предположим, что элемент t_{i+1} уже определен с точностью до аддитивной константы a_{i+1}. Для определения константы a_{i+1} и элемента t_i рассмотрим подробнее уравнение (1). Перепишем это уравнение в виде:

$$t_i = \int (p_i - (i+1)(t_{i+1} + a_{i+1})\theta_n') = -(i+1) a_{i+1} \theta_n + \int (p_i - (i+1) t_{i+1} \theta_n')$$

Элемент $p_i - (i+1) t_{i+1} \theta_n'$ принадлежит кольцу $K(x, \theta_1, \dots, \theta_{n-1})$. Воспользуемся рекурсией и вычислим этот интеграл при помощи алгоритма Риша. Необходимым условием интегрируемости исходной функции в классе элементарных функций является то, что $\int (p_i - (i+1) t_{i+1} \theta_n') = \alpha \theta_n + \beta$, где α — константа, а $\beta \in K(x, \theta_1, \dots, \theta_{n-1})$.

Если это условие выполнено, то мы получаем значение константы $a_{i+1} = -\alpha/(i+1)$ и значение коэффициента t_i. Интегрируя уравнение (1) при $i = 0$, т. е. вычисляя t_0, нужно отказаться от условия $\beta \in K(x, \theta_1, \dots, \theta_{n-1})$, достаточно, чтобы существовал элементарный интеграл $\int (p_0 - t_1 \theta')$. Этот интеграл определяется с точностью до аддитивной константы, которая является константой интегрирования, и не может быть определена при рассматриваемой постановке задачи.

2. Пусть $\theta_n = \exp(u)$, $u \in K(x, \theta_1, \dots, \theta_{n-1})$. Пусть $p = \sum\limits_{i=-k}^{l} p_i \theta_n^i$, где $p_i \in K(x, \theta_1, \dots, \theta_{n-1})$, $i = -k, \dots, l$. Пусть $v_0 = \sum\limits_{i=-k}^{l} t_i \theta_n^i$, где $t_i \in K(x, \theta_1, \dots, \theta_{n-1})$, $i = -k, \dots, l$. Получаем следующее выражение:

$$\sum_{i=-k}^{l} p_i \theta_n^i = \sum_{i=-k}^{l} t_i' \theta^i + \sum_{i=-k}^{l} i t_i \theta^i u' + \sum_{i=1}^{d} c_i \frac{v_i'}{v_i}$$

Приравнивая коэффициенты при одинаковых степенях θ_n получаем систему уравнений (дифференциальных уравнений Риша):

$$\begin{cases} p_i &= t_i' + it_iu', i \neq 0 \\ p_0 &= t_0' + \sum\limits_{i=1}^{d} c_i\frac{v_i'}{v_i}. \end{cases}$$

Пусть i имеет фиксированное значение, $i \neq 0$. Рассмотрим уравнение $p_i = t_i' + it_iu'$. Найдем t_i. Сначала избавимся от знаменателей. Пусть p_i — полином. Тогда и t_i — тоже полином.

Если и u' — полином, то оставляем уравнение без изменений, а если $u' = \frac{v}{w}$, то домножаем все уравнение на w.

Пусть $p_i = \frac{r}{g}$. Тогда t_i будет иметь вид $t_i = \frac{a}{b}$. Если u' — полином, то получаем уравнение:

$$\frac{r}{g} = \frac{a'b - ab'}{b^2} + \frac{iu'a}{b}$$

Следовательно, $b = \sqrt{g}$, $r = a'b - ab' + iu'ab = ba' + (-b' + iu'b)a$

Если $u' = \frac{v}{w}$, то получаем уравнение:

$$\frac{r}{g} = \frac{a'b - ab'}{b^2} + \frac{iva}{bw}$$

Следовательно, $b = \sqrt{\frac{g}{w}}$, $r = a'bw - ab'w + ivab = bwa' + (ivb - b'w)a$

После того, как мы избавились от знаменателя, мы получили уравнение вида:

$$P = c_1a' + c_2a,$$

где P, c_1, c_2, a — полиномы от переменной θ_{n-1} с коэффициентами из $F(x, \theta_1, \ldots, \theta_{n-2})$. [6] Пусть $deg(a) = h$.

Рассмотрим три случая:

1. $P, c_1, c_2, a \in F(x)$. Очевидно, что $m = max\{deg(c_1) + h - 1, deg(c_2) + h\} = h + max\{deg(c_1) - 1, deg(c_2)\}$. Из последнего равентсва мы можем найти h: $h = m - max\{deg(c_1) - 1, deg(c_2)\}$. Если $h < 0$, то исходная функция f не интегрируема в элементарном виде. Если $h = 0$, то $a = \frac{P}{c_2}$. Если $h > 0$, то получаем формулу:

$$P = c_1\left(a_0 + \sum_{i=1}^{h} ia_ix^{i-1}\right) + c_2\sum_{i=0}^{h} a_ix^i. \quad [6]$$

2. $P, c_1, c_2, a \in F(x, \theta_1, \ldots, \theta_{n-1})$, и $\theta_{n-1} = \log(u)$. В этом случае $m = h + max\{deg(c_1), deg(c_2)\}$. Следовательно $h = m - max\{deg(c_1), deg(c_2)\}$. Если $h < 0$, то то исходная функция f не интегрируема в элементарном виде. Если $h \geq 0$, то получаем систему дифференциальных уравнений:

$$P = c_1\left(\sum_{i=0}^{h} a_i'\theta_{n-1}^i + \sum_{i=1}^{h} ia_i\theta_{n-1}'\theta_{n-1}^{i-1}\right) + c_2\sum_{i=0}^{h} a_i\theta_{n-1}^i \quad [6]$$

3. $P, c_1, c_2, a \in F(x, \theta_1, \ldots, \theta_{n-1})$, $\theta_{n-1} = \exp(u)$. В этом случае в полиномиальной части содержатся отрицательные степени θ_{n-1}. Поэтому помимо положительной степени полинома a нужно оценить ещё и отрицательную. Обозначим через h_1 старшую степень полинома a, а через h_2 обозначим младшую степень полинома a, через k_1 и k_2 старшую и младшую степени полинома c_1 соответственно, через g_1 и g_2 старшую и младшую степени полинома c_2, через m_1 и m_2 старшую и младшую степени полинома P. Тогда $h_1 = m_1 - max\{k_1, g_1\}$ и $h_2 = m_2 - max\{k_2, g_2\}$. Получаем систему дифференциальных уравнений:

$$P = c_1 \sum_{i=h_2}^{h_1} (a_i' + i a_i u')\theta_{n-1}^i + c_2 \sum_{i=h_2}^{h_1} a_i \theta_{n-1}^i. \quad [6]$$

2.2. Интегрирование дробной части

Теперь рассмотрим интегрирование q/r, $deg(q) < deg(r)$. Разложим r на свободные от квадратов множители и представим q/r в виде суммы простых дробей $q/r = \sum_i q_i/r_i^i$, где r_i — множитель r, свободный от квадратов.

В случае, если $\theta_n = \exp(\xi)$, то перед разложением r на свободные от квадратов множители нужно в обобщенном полиноме r вынести в качестве отдельного множителя переменную θ_n в максимальной степени. Пусть $r = \theta_n^t r_t$, где r_t не делится на θ_n. Полином r_t раскладываем на свободные от квадратов множители. Другими словами, представляем обобщенный полином r в виде $r = \theta_n^t \prod_i r_i^i$, где r_i — множитель r, свободный от квадратов; r_i не делится на θ_n. Раскладываем дробную функцию q/r в сумму простых дробей $q/r = a/\theta_n^t + \sum_i q_i/r_i^i$, где r_i — множитель r, свободный от квадратов. Интеграл от функции a/θ_n^t вычисляется при помощи алгоритмов интегрирования полиномиальной части.

Нам нужно проинтегрировать каждую полученную дробь.

Теорема 2.1. *Пусть h — свободный от квадратов полином, старший коэффициент которого равен еденице, $h \in K(x, \theta_1, \theta_2, \ldots, \theta_n)$. Пусть h не делится на θ_n в случае, когда θ_n — экспонента. Тогда $gcd(h, h') = 1$. [11]*

Из этой теоремы, а также из того факта, что $K(x, \theta_1, \theta_2, \ldots, \theta_n)$ является Евклидовым кольцом следует, что существуют такие $a, b \in K(x, \theta_1, \ldots, \theta_n)$, что $r_i a + r_i' b = 1$.

Воспользуемся алгоритмом Эрмита:

$$\int \frac{q_k}{r_k^k} = -\frac{q_k b}{(k-1)r_k^{k-1}} + \int \frac{(k-1)q_k a + (q_k b)'}{(k-1)r_k^{k-1}},$$

где $a, b \in K(x, \theta_1, \ldots, \theta_n)$, $r_i a + r_i' b = 1$; и $k > 1$. В результате применения этой формулы степень знаменателя уменьшилась, таким образом мы должны повторно применять эту формулу пока $k \neq 1$. [11]

Нам осталось проинтегрировать выражение вида: u/v, где v — свободный от квадратов полином. Для этого воспользуемся алгоритмом Ротштейна - Трагера. Рассмотрим три случая:

1. $u/v \in K(x)$. Обозначим $R(y) = resultant_x(u - yv', v)$. Корни полинома $R(y)$ обозначим $a_1, a_2, \ldots, a_s$. Тогда справедлива формула:

$$\int \frac{u}{v} = \sum_{i=1}^{s} a_i \ln(\text{НОД}(u - a_i v', v))$$

2. $u/v \in K(x, \theta_1, \theta_2, \ldots, \theta_n)$, где θ_n — логарифм. Обозначим $R(y) = resultant_{\theta_n}(u - yv', v)$. Корни полинома $R(y)$ обозначим $a_1, a_2, \ldots, a_s$. Если хотя-бы один корень $R(y)$ не является константой, то функция f не имеет элементарного интеграла. Если $a_1, a_2, \ldots, a_s$ — константы, то справедлива формула:

$$\int \frac{u}{v} = \sum_{i=1}^{s} a_i \ln(\text{НОД}(u - a_i v', v))$$

3. $\frac{u}{v} \in K(x, \theta_1, \theta_2, \ldots, \theta_n)$, где θ_n — экспонента. Обозначим $R(y) = resultant_{\theta_n}(u - y(v' - N\xi' v), v)$, где N — степень полинома v, ξ — аргумент функции θ_n. Обозначим $a_1, a_2, \ldots, a_s$ корни полинома $R(y)$. Если хотя-бы один корень $R(y)$ не является константой, то функция f не имеет элементарного интеграла. Если $a_1, a_2, \ldots, a_s$ — константы, то справедлива формула:

$$\int \frac{u}{v} = \sum_{i=1}^{s} a_i (\ln(\text{НОД}(u - a_i (v' - N\xi' v), v)) - n_i \xi)$$

где n_i — степень полинома $\text{НОД}(u - a_i(v' - N\xi' v), v)$ [2]

3. Библиотека процедур для символьного интегрирования в системе MathPartner.

Все процедуры библиотеки расположены в четырёх пакетах:

- StrucTheorem — содержит алгоритмы для проверки трансцендентности функций.
- IntPolynom — содержит алгоритмы интегрирования обобщенной полиномиальной части интеграла.
- IntegrateFractions — содержит алгоритмы интегрирования дробно-рациональной части интеграла.
- Integrate — содержит алгоритмы построения последовательности регулярных мономов.

3.1. Пакет процедур StructTheorem

Содержит 18 процедур, основной среди которых является процедура `makeRegular-Monomials-Sequence`. Эта процедура составляет последовательность регулярных мономов на основе списка всех логарифмов и экспонент, содержащихся в подынтегральном выражении.

Этот список отсортирован следующим образом. Чем проще аргумент логарифма, тем ближе к началу списка находится этот логарифм. Чем проще аргумент экспоненты, тем ближе к началу списка находится эта экспонента.

В процедуре `makeRegularMonomialsSequence` выполняются следующие действия. Мы проходим последовательно все элементы списка логарифмов и экспонент, начиная со второго элемента. Берем очередной элемент списка логарифмов и экспонент. Применяем к нему структурную теорему. Если он оказывается трансцендентным, то рассматриваем следующий элемент. В противном случае выражаем его через предыдущие элементы списка, и в каждом следующем элементе списка меняем текущий элемент его выражением через предыдущие элементы, и удаляем текущий элемент из списка. В итоге получаем последовательность регулярных мономов.

Выражаем логарифмы и экспоненты, содержащиеся в подынтегральной функции, через элементы последовательности регулярных мономов.

3.2. Пакет процедур IntPolynom

Содержит 20 процедур, основными среди которых являются:

`integPol`: Интегрирование полинома от переменной x.

`integPolLn`: Интегрирование обобщенного полинома от переменной $ln(u)$.

`integPolExp`: Интегрирование обобщенного полинома от переменной $exp(u)$.

3.3. Пакет процедур IntegrateFractions

Содержит 17 процедур, основными среди которых являются:

`Hermite`: Интегрирование дробной функции по методу Эрмита.

`LogarithmicPart`: Вычисление интеграла от дробной функции со свободным от квадратов знаменателем.

`partialFraction`: Разложение дробной функции в сумму простейших.

3.4. Пакет процедур Integrate

Содержит 11 процедур, основным среди которых являются процедуры:

`integrate`: Преобразует функцию f в комплексный вид. То есть, заменяем тригонометрические, обратные тригонометрические гиперболические, обратные гиперболические выражения комбинацией логарифмов и экспонент. Затем вызывает процедуру `main`, которая возвращает неопределенный интеграл функции f в комплексном виде. Затем, при помощи процедур из пакета обработки функций MathPartner, полученная функция упрощается и приводится к вещественному виду (комбинации логарифмов и экспонент заменяются на тригонометрические, обратные тригонометрические гиперболические, обратные гиперболические).

`main`: Основная процедура интегрирования функции. Возвращает первообразную от входного выражения. Выделяет в подынтегральном выражении последовательность логарифмов и экспонент $f_0, \ldots, f_n$. С помощью структурной теоремы (Пакет процедур StrucTheorem) составляет из последовательности $f_0, \ldots, f_n$ последовательность регулярных мономов. Отделяет полиномиальную и дробно-рациональю части интеграла и интегрирует отдельно полиномиальную часть (Пакет процедур IntPolynom) и отдельно дробно-рациональную часть интеграла (Пакет процедур IntegrateFraction).

3.5. Общая схема алгоритма.

Входные данные : f – подынтегральная функция, x — переменная интегрирования. Сначала преобразуем функцию f в комплексный вид. То есть, заменяем тригонометрические, обратные тригонометрические, гиперболические, обратные гиперболические выражения комбинацией логарифмов и экспонент. Составляем список логарифмов и экспонент, содержащихся в функции f.

При помощи процедуры `makeRegularMonomialsSequence` составляем последовательность регулярных мономов. Выражаем функцию f через найденные регулярные мономы. Раскладываем функцию f в виде суммы полиномиальной и дробной части $f = p + q/r$.

Если q/r не равно нулю, то используя процедуру `FactorPol_SquareFree` из пакета полиномиальных вычислений MathPartner раскладываем r на произведение свободных от квадратов множителей $r = \sum_i r_i^i$. Затем вызываем процедуры из пакета `integrateFractiions`. Процедура `partialFraction` выражает q/r в виде суммы простейших дробей $q/r = \sum_i q_i/r_i^i$. К каждой дроби из полученной суммы применяем алгоритм Эрмита (процедура `Hermit`), и затем алгоритм Ротштейна-Трагера (процедура `LogarithmicPart`). Если интеграл от очередной дроби не выражается через элементарные функции, то интеграл от функция f так же не выражается через элементарные функции.

Если p не равно нулю, то вызываем процедуры из пакета `intPolynom`. Возможны три случая:

Если p – полином переменной x, то процедура `integPol` вычисляет интеграл от p.

Если p – полином от переменной $exp(\xi)$, то процедура `integPolExp` составляет систему дифференциальных уравнений, описанных в разделе 2.2. Решение каждого уравнения выражается через интеграл, не содержащий последний регулярный моном. Каждый такой интеграл рекурсивно вычисляем алгоритмом Риша. Если очередное уравнение не имеет решений, то интеграл от функция f не выражается через элементарные функции.

Если p – полином переменной $ln(\xi)$, то процедура `integPolLn` составляет и решает систему дифференциальных уравнений, описанных в разделе 2.1. Решение каждого уравнения выражается через интеграл, не содержащий последний регулярный моном. Каждый такой интеграл рекурсивно вычисляем алгоритмом Риша. Если в системе находится уравнение, которое не имеет решений, то интеграл от функция f не выражается через элементарные функции.

Полученный ответ упрощается и приводится к вещественному виду (комбинации логприфмов и экспонент заменяются на тригонометрические, обратные тригонометрические гиперболические, обратные гиперболические).

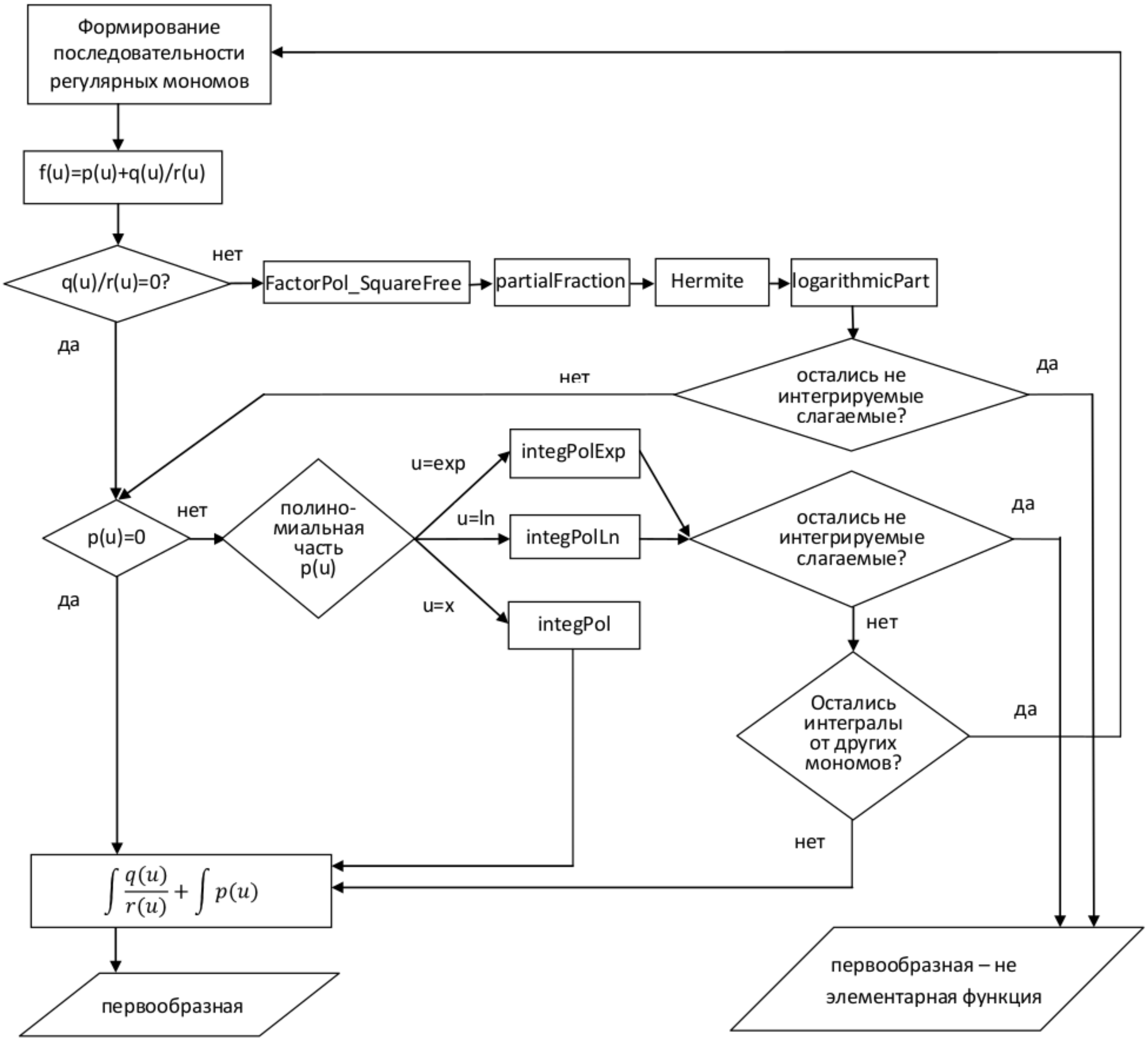

Рис.1 Общая схема алгоритма.

4. Примеры вычисления первообразных

Мы приводим три таблицы, содержащие результаты интегрирования функций разных видов.

функция	**результат**
$3/(x+2)$	$3 \cdot \ln(abs(x+2))$
$3x/(x+2)$	$(3x - 6 \cdot \ln(abs(x+2)))$
$(3x+4)/(x+2)$	$(3x - 2 \cdot \ln(abs(x+2)))$
$1/(x^2+x+1)$	$2/3 \cdot 3^{(1/2)} \cdot \operatorname{arctg}(2/3x \cdot 3^{(1/2)} + 1/3 \cdot 3^{(1/2)})$
$(4x+5)/(x^2+2x+3)$	$(2 \cdot \ln(abs(1/2x^2 + x + 3/2)) + 1/2 \cdot 2^{(1/2)} \cdot \operatorname{arctg}(1/2x \cdot 2^{(1/2)} + 1/2 \cdot 2^{(1/2)}))$
$(4x^2+5x+6)/(x^2+2x+3)$	$(4x - (((-3/2) \cdot 2^{(1/2)} \cdot (-1) \cdot \operatorname{arctg}(1/2x \cdot 2^{(1/2)} + 1/2 \cdot 2^{(1/2)})) + 3/2 \cdot \ln(abs(9/2x^2 + 9x + 27/2))))$
$1/(x^3+x)$	$\ln(abs(x)) - 1/2 \cdot \ln(abs(x^2+1))$
$1/(x^2+1)$	$\operatorname{arctg}(x)$

Табл.1 Дробно-рациональные функции.

функция	**результат**
$\sin(2x+3)$	$(-1/2) \cdot \cos(2x+3)$
$\cos(2x+3)$	$1/2 \cdot \sin(2x+3)$
$\sin(x)^2 \cos(x)^2$	$1/8x - 1/32 \cdot \sin(4x)$
$\operatorname{tg}(2x+3)$	$(-1/2) \cdot \ln(abs(\cos(2x+3)))$
$\operatorname{ctg}(2x+3)$	$1/2 \cdot \ln(abs(\sin(2x+3)))$
$1/\cos(x)^2$	$\operatorname{tg}(x)$
$1/\sin(x)^2$	$-\operatorname{ctg}(x)$
$1/(1-\cos(x))$	$-\operatorname{ctg}(x/2)$
$\operatorname{tg}(x)^2$	$(\sin(x) - x \cdot \cos(x) - i \cdot \cos(x))/\cos(x)$
$\sin(x) \cdot \operatorname{tg}(\cos(x))$	$\ln(abs(\cos(\cos(x))))$

Табл.2 Тригонометрические функции.

функция	**результат**
$\sin(x)\exp(\cos(x))$	$-1 \cdot \exp(\cos(x))$
$\exp(x) \cdot \sin(x)$	$1/2 \cdot \exp(x) \cdot \sin(x) - 1/2 \cdot \exp(x) \cdot \cos(x)$
$(y^2+1) \cdot \exp(y)$	$((y^2 \cdot \exp(y) + 3 \cdot \exp(y)) - 2y \cdot \exp(y))$
$(y^2+1) \cdot \ln(y)$	$1/3y \cdot \ln(y^3) + 1/3y^3 \cdot \ln(y) + (-1/9)y^3 + (-y)$
$1/x \ln(x)$	$\ln(x)^2/2$
$(x+1/x)\ln(x)$	$(1/2 \cdot (\ln(x))^2 + 1/2x^2 \cdot \ln(x)) - 1/4x^2$
$(x+\exp(x))^2$	$((1/2 \cdot \exp(2x) + 2x \cdot \exp(x) + 1/3x^3) - 2 \cdot \exp(x))$
$2 \cdot \exp(x)/(\exp(4x)-1)$	$-1 \cdot (\operatorname{arctg}(\exp(x)) - 1/2 \cdot \ln(abs(\operatorname{sh}(1/2x))) + 1/2 \cdot \ln(abs(\operatorname{ch}(1/2x))))$
$(\ln(x)+1) \cdot x^x$	x^x
$\log(\pi, x)$	$(x \cdot \ln(x) - x)/\ln(\pi)$
π^x	$(\pi^x)/\ln(\pi)$

Табл.3 Другие типы функций.

5. Заключение

Задача вычисления первообразной для композиции элементарных функций, когда результат может быть записан через элементарные функции и алгебраические и трансцендентные постоянные, привлекает внимание математиков с давних времен. И не смотря на то, что общие подходы достаточно понятны, эта задача в виде библиотеки программ все еще ждет своего решения. Мы надеемся, что настоящее описание текущего состояния библиотеки программ в системе MathPartner будет полезным и позволит приблизить полное решение этой классической математической задачи.

6. Благодарности

Автор благодарит своего научного руководителя Г. И. Малашонка за постановку задачи и многочисленные обсуждения алгоритмов символьного интегрирования. Он выражает глубокую признательность всем коллегам по лаборатории алгебраических вычислений и участникам семинара по компьютерной алгебре ТГУ за плодотворные критические обсуждения.

СПИСОК ЛИТЕРАТУРЫ

1. *Risch R.* The problem of integration in finite terms. // Trans. Amer. Math. Soc. 1969. P. 167-189.

2. *Дэвенпорт Дж., Сирэ И., Турнье Э.* Компьютерная алгебра. Системы и алгоритмы алгебраических вычислений. М.: Мир, 1991, 351 с.

3. *Панкратьев Е. В.* Элементы компьютерной алгебры (Конспекты спецкурса). М.: МГУ, 2007, 243 с.

4. *Bronstein M.* Symbolic Integration Tutorial // ISSAC'98, Rostock, 35с.

5. *Bronstein M.* Symbolic Integration. I: Transcendental functions, volume 1 of Algorithms and Computation in Mathematics. Berlin: Springer-Verlag, 2005, Second edition.

6. *Тарарова С. М.* К проблеме построения алгоритма символьного интегрирования. // Вестник Тамбовского университета. Сер. Естественные и технические науки. 2012. Том 17. Выпуск 2. С. 607-617.

7. *Малашонок Г. И.* Руководство по языку "Mathpar". Тамбов: Издательство Тамбовского университета, 2013.

8. *Malaschonok G. I.* MathPartner Computer Algebra ISSN 0361-7688, Programming and Computer Software, 2017, Vol. 43, No. 2, pp. 112-118.

9. *Малашонок Г.И.* Компьютерная математика для вычислительной сети // Вестник Тамбовского университета. Сер. Естественные и технические науки. Тамбов, 2010. Том 15. Вып. 1. С. 322-327.

10. *Павлов Д. А.* Символьное интегрирование. // Компьютерные инструменты в образовании. Москва, 2010. Вып. 2. С. 38-43.

11. *Terelius B.* Symbolic integration. Master of science thesis. Stockholm, 2009.

Поступила в редакцию 00 месяц 2018 г.

Прошла рецензирование 00 месяц 2018 г.
Принята в печать 00 месяц 2018 г.
Конфликт интересов отсутствует.

Корабельников Вячеслав Алексеевич, Тамбовский государственный университет им. Г.Р. Державина, г. Тамбов, Российская Федерация, аспирант кафедры функционального анализа, e-mail: korabelnikov.va@gmail.com

Для цитирования: *В. А. Корабельников* Алгоритмы символьного интегрирования в системе MathPartner // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2018. Т. 23. № 121. С. xx–xx. DOI: 10.20310/1810-0198-2018-23-121-xx-xx

DOI: 10.20310/1810-0198-2018-23-121-xx-xx

Symbolic integration algorithms in CAS MathPartner

© V. A. Korabelnikov

Tambov State University named after G.R. Derzhavin
33 Internatsionalnaya St., Tambov, Russian Federation, 392000
E-mail: korabelnikov.va@gmail.com

Abstract. Risch theorem, published in 1969, gave beginning to creation of procedure library for symbolic integration. But such library, for past almost 50 years, still not been created. Some attempts of creation such libraries is known, but not one of them not finished. In computer algebra system MathPartner a new procedure library for symbolic integration, based on Risch theorem, is creating. We give detailed description of basic procedures contained in this library, and role of each procedure in symbolic integration algorithm. We represent procedural block diagram of whole algorithm and examples of computed integrals.

Key words: Risch algorithm; symbolic integration; indefinite integral; CAS MathPartner; differential field; elementary functions

REFERENCES

1. Risch R. The problem of integration in finite terms. // Trans. Amer. Math. Soc. 1969. P. 167-189.

2. Davenport J., Siret Y., Tournier E. *Komp'yuternaya algebra. Sistemy i algoritmy algebraicheskikh vychisleniy.* [Computer algebra. Systems and algorithms of algebraic computation.] Moscow, Mir, 1991, 351 p. (In Russian).

3. Pankrat'yev E. V. *Elementy komp'yuternoy algebry (Konspekty spetskursa).* [Elements of computer algebra (special course notes)] Moscow, MSU, 2007, 243 p. (In Russian).

4. Bronstein M. Symbolic Integration Tutorial // ISSAC'98, Rostock, 35c.

5. Bronstein M. Symbolic Integration. I: Transcendental functions, volume 1 of Algorithms and Computation in Mathematics. Berlin: Springer-Verlag, 2005, Second edition.

6. Tararova S.M. *K probleme postroyeniya algoritma simvol'nogo integrirovaniya.* [To the problem of constructing an algorithm for symbolic integration.] *Vestnik Tambovskogo universiteta. Seriya Estestvennye i tekhnicheskie nauki - Tambov University Reports. Series: Natural and Technical Sciences*, 2012, vol. 17, no. 2, pp. 607-617. (In Russian).

7. Malaschonok G. I. *Rukovodstvo po yazyku "Mathpar".* [Language guide "Mathpar"] Nambov, Publishing house of Tambov university, 2013.

8. Malaschonok G. I. MathPartner Computer Algebra ISSN 0361-7688, Programming and Computer Software, 2017, Vol. 43, No. 2, pp. 112-118.

9. Malaschonok G. I. *Komp'yuternaya matematika dlya vychislitel'noy seti.* [Computer mathematics for computer network.] *Vestnik Tambovskogo universiteta. Seriya Estestvennye i tekhnicheskie nauki*

The work is partially supported by the Russian Fund for Basic Research (project No 16-07-00420)

- *Tambov University Reports. Series: Natural and Technical Sciences*, 2010, vol. 15, no. 1, pp. 322-327. (In Russian).

10. Pavlov D. A. *Simvol'noye integrirovaniye.* [Symbolic integration] *Komp'yuternyye instrumenty v obrazovanii - Computer tools in education*, Moscow, 2010. Vol. 2. p. 38-43.

11. Terelius B. Symbolic integration. Master of science thesis. Stockholm, 2009.

Received 00 Month 2018
Reviewed 00 Month 2018
Accepted for press 00 Month 2018
There is no conflict of interests.

Korabelnikov Vyacheslav Alekseevich, Tambov State University named after G.R. Derzhavin, Tambov, the Russian Federation, Post-Graduate Student of the Functional Analysis Department, e-mail: korabelnikov.va@gmail.com

For citation: Korabelnikov V.A. Algoritmy simvol'nogo integrirovaniya v sisteme MathPartner [Symbolic integration algorithms in CAS MathPartner]. *Vestnik Tambovskogo universiteta. Seriya Estestvennye i tekhnicheskie nauki – Tambov University Reports. Series: Natural and Technical Sciences*, 2018, vol. 23, no. 121, pp. xx–xx. DOI: 10.20310/1810-0198-2018-23-121-xx-xx(In Russian, Abstr. in Engl.).