

© Корабельников В.А., 2019

DOI 10.20310/1810-0198-2019-24-125-75-89

УДК 517.312, 004.421.6

Алгоритмы символьного интегрирования в системе MathPartner

Вячеслав Алексеевич КОРАБЕЛЬНИКОВ

ФГБОУ ВО «Тамбовский государственный университет им. Г.Р. Державина»

392000, Российская Федерация, г. Тамбов, ул. Интернациональная, 33

ORCID: <https://orcid.org/0000-0002-3920-8373>, e-mail: korabelnikov.va@gmail.com

Symbolic integration algorithms in CAS MathPartner

Vyacheslav A. KORABELNIKOV

Tambov State University named after G.R. Derzhavin

33 Internatsionalnaya St., Tambov 392000, Russian Federation

ORCID: <https://orcid.org/0000-0002-3920-8373>, 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

Благодарности: Работа выполнена при поддержке РФФИ (проект № 16-07-00420).

Для цитирования: *Корабельников В. А.* Алгоритмы символьного интегрирования в системе MathPartner // Вестник Тамбовского университета. Серия: естественные и технические науки. Тамбов, 2019. Т. 24. № 125. С. 75–89. DOI 10.20310/1810-0198-2019-24-125-75-89

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.

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

Acknowledgements: The work is partially supported by the Russian Foundation for Basic Research (project no. 16-07-00420).

For citation: Korabelnikov V. A. Algoritmy simvol'nogo integrirvaniya 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*, 2019, vol. 24, no. 125, pp. 75–89. DOI 10.20310/1810-0198-2019-24-125-75-89 (In Russian, Abstr. in Engl.)

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. В отличие от дифференцирования, в интегрировании есть только одно общее правило $\int(f + g) = \int f + \int g$. Но и оно применимо далеко не всегда (см. [1, с. 218]).

Алгоритм интегрирования дробно-рациональных функций был известен еще во времена И. Ньютона, Г.В. Лейбница и Д. 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 подынтегральной функции на линейные множители. В случае, если знаменатель подынтегрального выражения не раскладывается на линейные множители над кольцом Q , то потребуется добавлять к этому кольцу элементы из R или C . В 19 веке Ш. Эрмит улучшил этот алгоритм (см. [2]).

Систематическое изучение случаев, когда интеграл может быть выражен в конечном виде, началось в начале 19 века. В 1820 году П.-С. Лаплас заметил, что интеграл функции не содержит радикалов, кроме тех, что находятся в подынтегральном выражении. Около десятилетия спустя Ж. Лиувилль сформулировал и доказал более строгую и более точную теорему, которая означает, 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 Р. Риш [3] 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 (см. [4]). Стоит отметить, что алгоритм Риша позволяет производить интегрирование лишь чисто трансцендентных функций. Для алгебраических функций Р. Риш доказал существование алгоритма, но алгоритм им не был получен (см. [5]).

В 1971 году Дж. Мозесом на основе алгоритма Риша была написана программа под названием SIN, которая интегрировала чисто трансцендентные функции. В 1981 г. Дж. Дэвенпорт, также основываясь на алгоритме Риша, разработал алгоритм интегрирования чисто алгебраических функций. Для интегрирования произвольных элементарных функций программы Мозеса и Дэвенпорта были непригодны (см. [5]).

Алгоритм Дэвенпорта обладал большой вычислительной сложностью. Б. Трагер в 1984 г. внёс серьёзные улучшения в алгоритм Дэвенпорта. Обновлённый алгоритм был реализован в системах компьютерной алгебры Axiom и Maple (см. [5]). М. Бронштейн в 1990 г. обобщил алгоритм Трагера на произвольные элементарные функции. В 1998 г. М. Бронштейн написал монографию [2] по символьному интегрированию, в которой достаточно популярно изложил лучшие достижения в этой области, начиная с теоремы Лиувилля и заканчивая собственными результатами.

Полная программная реализация алгоритма Риша–Трагера–Бронштейна на данное время не завершена (см. [5]).

Целью данной работы является описание алгоритмов символьного интегрирования, которые сегодня применяются в системе компьютерной алгебры MathPartner [6]. 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 сообщает, что подынтегральное выражение не интегрируемо в элементарных функциях. Данная работа описывает текущее состояние создаваемой библиотеки процедур символьного интегрирования на пути решения этой сложной фундаментальной задачи.

1. Основные понятия

Сначала введем определения, необходимые для описания алгоритма Риша. Будем придерживаться терминологии, принятой в [7, с. 213–229].

О п р е д е л е н и е 1.1. Поле K называется дифференциальным полем, если на нем определен оператор дифференцирования d , удовлетворяющий двум условиям: $d(ab) = d(a)b + ad(b)$, $d(a + b) = d(a) + d(b)$. Элемент $d(a)$ называется производной элемента $a \in K$, производную также обозначают a' .

О п р е д е л е н и е 1.2. Пусть K — дифференциальное поле. Тогда множество $C = \{a \in K : d(a) = 0\}$ является подполем K . Элементы этого подполя называют константами.

О п р е д е л е н и е 1.3. Пусть K — дифференциальное поле с оператором дифференцирования d . Расширение E поля K называется дифференциальным расширением K , если на E определен оператор дифференцирования d_1 , такой, что $d_1(a) = d(a)$, $\forall a \in K$. Оператор d_1 называется продолжением оператора d и часто обозначается тем же символом d .

Отметим, что если E — алгебраическое расширение поля K , то дифференцирование поля K однозначно продолжается до дифференцирования поля E (см. [7, с. 217]).

О п р е д е л е н и е 1.4. Пусть E — дифференциальное расширение дифференциального поля K . Если $\theta \in E$ удовлетворяет соотношению $f\theta' = f'$ для некоторого ненулевого элемента $f \in K$, то θ называют логарифмом элемента f над K , и обозначают $\theta = \log(f)$.

О п р е д е л е н и е 1.5. Пусть E — дифференциальное расширение дифференциального поля K . Если $\theta \in E$ удовлетворяет соотношению $\theta' = f'\theta$ для некоторого ненулевого элемента $f \in K$, то θ называют экспонентой элемента f над K , и обозначают $\theta = \exp(f)$.

О п р е д е л е н и е 1.6. Пусть $K(x)$ — дифференциальное поле. Будем обозначать $K(x, \theta)$ дифференциальное расширение поля $K(x)$, которое получается добавлением в поле $K(x)$ элемента θ .

О п р е д е л е н и е 1.7. Элемент θ является регулярным мономом над дифференциальным полем K , если θ трансцендентен над K , и является либо логарифмом, либо экспонентой. Набор $\theta_1, \theta_2, \dots, \theta_n$ называется набором регулярных мономов, если каждый ее элемент θ_i является регулярным мономом над $K(x, \theta_1, \theta_2, \dots, \theta_{i-1})$, $i = 1, \dots, n$.

О п р е д е л е н и е 1.8. (см. [1, с. 225]) Пусть K — функциональное поле. Расширение $K(\theta_1, \theta_2, \dots, \theta_n)$ поля K называется полем элементарных функций над K , если каждая функция θ_i либо является регулярным мономом над K , либо алгебраична над K . Функция называется элементарной над K , если она принадлежит некоторому полю элементарных функций над K .

Сформулируем принцип Лиувилля (см. [1, с. 226]).

Теорема 1.1. Пусть K — дифференциальное поле, $f(x) \in K$, $g(x)$ — элементарная над K функция, такая, что $g'(x) = f(x)$. Тогда $g(x)$ можно представить в виде

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

где $v_0(x) \in K$; $v_i(x)$ принадлежат расширению K_0 поля K , получающемуся присоединением к K конечного числа констант, алгебраических над K ; c_i — константы из поля K_0 .

Нам требуется конструировать наименьшее дифференциальное поле, содержащее подынтегральную функцию. Для решения этой задачи используется следующее утверждение, приведенное в [7, с. 225].

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

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

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

где c — некоторая константа.

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

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

где c — некоторая константа.

2. Алгоритм Риша

Пусть x — независимая переменная над полем констант K , $\theta_1, \theta_2, \dots, \theta_n$ — набор регулярных мономов, $K(x, \theta_1, \theta_2, \dots, \theta_n)$ — дифференциальное поле и $f \in K(x, \theta_1, \theta_2, \dots, \theta_n)$.

Ниже описывается алгоритм, позволяющий вычислить неопределенный интеграл от f тогда и только тогда, когда этот интеграл есть элементарная функция. Таким образом, алгоритм еще и позволяет определить ситуацию, когда интеграл не выражается через элементарные функции.

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 $p, q, r \in K[x]$, $\deg(r) > \deg(q)$.

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

3. Если $f \in K(x, \theta_1, \theta_2, \dots, \theta_n)$, $\theta_n = \exp(u)$ и $u \in K(x, \theta_1, \dots, \theta_{n-1})$, то $p = \sum_{i=-k}^l p_i \theta_n^i$, а q и r — полиномы от переменной θ_n с коэффициентами из поля $K(x, \theta_1, \theta_2, \dots, \theta_{n-1})$, $\deg(r) > \deg(q)$.

2.1. Интегрирование полиномиальной части

Алгоритм интегрирования полиномиальной части сводит вычисление интеграла в поле $K(x, \theta_1, \theta_2, \dots, \theta_n)$ к вычислению одного или более интегралов в поле $K(x, \theta_1, \theta_2, \dots, \theta_{n-1})$. Каждый полученный интеграл рекурсивно вычисляется при помощи алгоритма Риша (подробнее см. [7, с. 229–236], [8]).

Рассмотрим интегрирование полинома p . По теореме Лиувилля

$$p = v'_0 + \sum_{i=1}^d c_i \frac{v'_i}{v_i}.$$

Возможны следующие два случая.

1. Пусть $\theta_n = \log(u)$, $u \in K(x, \theta_1, \dots, \theta_{n-1})$. Для $p = \sum_{i=0}^m p_i \theta_n^i$, $v_0 = \sum_{j=0}^{m+1} t_j \theta_n^j$, где $p_i \in K(x, \theta_1, \dots, \theta_{n-1})$, $i = 0, 1, \dots, m$, $t_j \in K(x, \theta_1, \dots, \theta_{n-1})$, $j = 0, 1, \dots, m+1$, получаем (см. [8]) следующее выражение

$$\sum_{i=0}^m p_i \theta_n^i = \sum_{i=0}^{m+1} t'_i \theta_n^i + \sum_{i=0}^m (i+1) t_{i+1} \theta_n^i \theta'_n + \sum_{i=1}^d c_i \frac{v'_i}{v_i}.$$

Здесь старший коэффициент t_{m+1} является константой, обозначим ее a_{m+1} . Для нахождения остальных коэффициентов t_i , $i = m, m-1, \dots, 0$ приравниваем коэффициенты при одинаковых степенях θ_n . Получаем систему дифференциальных уравнений

$$p_i = t'_i + (i+1) t_{i+1} \theta'_n. \quad (2.1)$$

Для того чтобы определить константу a_{i+1} и коэффициент t_i (с точностью до аддитивной константы), перепишем уравнение 2.1 в виде

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

Элемент $p_i - (i+1) t_{i+1} \theta'_n$ принадлежит полю $K(x, \theta_1, \dots, \theta_{n-1})$. Воспользуемся рекурсией и вычислим этот интеграл при помощи алгоритма Риша. Необходимым условием интегрируемости f в классе элементарных функций является выполнение равенства $\int (p_i - (i+1) t_{i+1} \theta'_n) = \alpha \theta_n + \beta$, где α — константа, а $\beta \in K(x, \theta_1, \dots, \theta_{n-1})$. При его выполнении получаем $a_{i+1} = -\alpha/(i+1)$ и находим значение коэффициента t_i . Вычисляя t_0 , нужно отказаться от условия $\beta \in K(x, \theta_1, \dots, \theta_{n-1})$. Достаточно, чтобы существовал элементарный интеграл $\int (p_0 - t_1 \theta'_n)$ (этот интеграл определяется с точностью до аддитивной константы).

2. Пусть $\theta_n = \exp(u)$, $u \in K(x, \theta_1, \dots, \theta_{n-1})$. Для $p = \sum_{i=-k}^l p_i \theta_n^i$, $v_0 = \sum_{j=-k}^l t_j \theta_n^j$, где $p_i \in K(x, \theta_1, \dots, \theta_{n-1})$, $i = -k, \dots, l$, $t_j \in K(x, \theta_1, \dots, \theta_{n-1})$, $j = -k, \dots, l$, получаем следующее выражение

$$\sum_{i=-k}^l p_i \theta_n^i = \sum_{i=-k}^l t'_i \theta_n^i + \sum_{i=-k}^l i t_i \theta_n^i u' + \sum_{i=1}^d c_i \frac{v'_i}{v_i}.$$

Приравнивая коэффициенты при одинаковых степенях θ_n получаем следующую систему уравнений

$$\begin{cases} p_i = t'_i + i t_i u', & i \neq 0 \\ p_0 = t'_0 + \sum_{j=1}^d c_j \frac{v'_j}{v_j}. \end{cases} \quad (2.2)$$

Зафиксируем $i \neq 0$ и рассмотрим уравнение $p_i = t'_i + i t_i u'$ относительно t_i . Пусть p_i — полином. Тогда t_i — тоже полином. Если u' — полином, то оставляем уравнение

без изменений, а если $u' = v/w$, то чтобы «избавиться от знаменателей», домножаем уравнение на w .

Пусть $p_i = r/g$. Тогда t_i будет иметь вид $t_i = a/b$. 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 $u' = v/w$, то из соотношения

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

получаем: $b = \sqrt{\frac{g}{w}}$, $r = a'bw - ab'w + iwab = bwa' + (ivb - b'w)a$.

Во всех рассмотренных выше случаях было получено уравнение вида

$$P = c_1a' + c_2a, \tag{2.3}$$

где P, c_1, c_2, a — полиномы от переменной θ_{n-1} с коэффициентами из $K(x, \theta_1, \dots, \theta_{n-2})$.

Для определения полинома a из уравнения 2.3 возникает три ситуации.

1. $P, c_1, c_2, a \in K[x]$. Пусть $a = \sum_{i=0}^h a_i x^i$. $\deg(P) = \max\{\deg(c_1)+h-1, \deg(c_2)+h\} = h + \max\{\deg(c_1) - 1, \deg(c_2)\}$. Из последнего равенства находим h : $h = \deg(P) - \max\{\deg(c_1) - 1, \deg(c_2)\}$. Если $h < 0$, то интеграл от функции f не является элементарной функцией. Если $h = 0$, то $a = P/c_2$. Если $h > 0$, то получаем формулу:

$$P = c_1 \left(a_0 + \sum_{i=1}^h i a_i x^{i-1} \right) + c_2 \sum_{i=0}^h a_i x^i.$$

Приравнивая коэффициенты при одинаковых степенях x , получаем систему линейных уравнений относительно неизвестных a_i .

2. $\theta_{n-1} = \log(u)$. Пусть $a = \sum_{i=0}^h a_i \theta_{n-1}^i$. Тогда $\deg(P) = h + \max\{\deg(c_1), \deg(c_2)\}$. Следовательно, $h = \deg(P) - \max\{\deg(c_1), \deg(c_2)\}$. Если $h < 0$, то интеграл от функции f не является элементарной функцией. Если $h \geq 0$, то получаем формулу

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

Приравнивая коэффициенты при одинаковых степенях θ_{n-1} , получаем систему дифференциальных уравнений относительно неизвестных a_i . Все выражения, входящие в эту систему, принадлежат $K(x, \theta_1, \dots, \theta_{n-2})$. Таким образом, мы свели решение дифференциального уравнения в $K(x, \theta_1, \dots, \theta_{n-1})$ к решению системы дифференциальных уравнений в $K(x, \theta_1, \dots, \theta_{n-2})$.

3. $\theta_{n-1} = \exp(u)$. В этом случае в полиномах P, c_1, c_2, a содержатся отрицательные степени θ_{n-1} . Пусть $a = \sum_{i=-h_1}^{h_2} a_i \theta_{n-1}^i$. Обозначим через k_1 старшую степень полинома

c_1 , а через k_2 обозначим младшую степень полинома c_1 , через g_1 и g_2 старшую и младшую степени полинома c_2 соответственно, 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 формулу

$$P = c_1 \sum_{i=h_2}^{h_1} (a'_i + ia_i u') \theta_{n-1}^i + c_2 \sum_{i=h_2}^{h_1} a_i \theta_{n-1}^i.$$

Приравнивая коэффициенты при одинаковых степенях θ_{n-1} , получаем систему дифференциальных уравнений относительно неизвестных a_i . Все выражения, входящие в эту систему, принадлежат $K(x, \theta_1, \dots, \theta_{n-2})$. Таким образом, мы свели решение дифференциального уравнения в $K(x, \theta_1, \dots, \theta_{n-1})$ к решению системы дифференциальных уравнений в $K(x, \theta_1, \dots, \theta_{n-2})$.

2.2. Интегрирование дробной части

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 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 (см. [4]). Другими словами, представляем 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. Используем следующее утверждение из [4].

Теорема 2.1. Пусть h — свободный от квадратов полином от переменной θ_n с коэффициентами из $K(x, \theta_1, \theta_2, \dots, \theta_{n-1})$, со старшим коэффициентом, равным единице. Тогда, если h не делится на θ_n в случае, когда θ_n — экспонента, то $\text{НОД}(h, h') = 1$.

Из этой теоремы, а также из того факта, что $K(x, \theta_1, \theta_2, \dots, \theta_n)$ является Евклидовым кольцом, следует, что существуют такие $a, b \in K(x, \theta_1, \dots, \theta_n)$, 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 $a, b \in K(x, \theta_1, \dots, \theta_n)$, $r_i a + r_i' b = 1$; и $k > 1$. В результате применения этой формулы степень знаменателя уменьшилась, таким образом мы должны повторно применять эту формулу, пока $k \neq 1$ (подробнее см. [4]).

Остается проинтегрировать выражение u/v , где v — свободный от квадратов полином. Рассмотрим три случая (подробнее см. [1, с. 222–236]):

1. $u/v \in K(x)$. Определим $R(y) = \text{resultant}_x(u - yv', v)$. Корни полинома $R(y)$ обозначим a_1, a_2, \dots, a_s . Справедлива формула

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

2. $u/v \in K(x, \theta_1, \theta_2, \dots, \theta_n)$, где θ_n — логарифм. Пусть $R(y) = \text{resultant}_{\theta_n}(u - yv', v)$. Если хотя-бы один корень a_1, a_2, \dots, a_s полинома $R(y)$ не является константой, то функция f не имеет элементарного интеграла. Если его корни a_1, a_2, \dots, a_s — константы, то

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

3. $u/v \in K(x, \theta_1, \theta_2, \dots, \theta_n)$, где θ_n — экспонента. В этом случае положим $R(y) = \text{resultant}_{\theta_n}(u - y(v' - N\xi'v), v)$, где N — степень полинома v , ξ — аргумент функции θ_n . Если хотя бы один корень a_1, a_2, \dots, a_s полинома $R(y)$ не является константой, то функция f не имеет элементарного интеграла. Если его корни a_1, a_2, \dots, a_s — константы, то

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

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

3. Библиотека процедур для символьного интегрирования в системе MathPartner

Все процедуры библиотеки расположены в четырёх пакетах:

- Integrate — содержит алгоритмы построения набора регулярных мономов.
- StrucTheorem — содержит алгоритмы для проверки трансцендентности функций.
- IntPolynom — содержит алгоритмы интегрирования полиномиальной части интеграла.
- IntegrateFractions — содержит алгоритмы интегрирования дробной части интеграла.

3.1. Пакет процедур Integrate

Содержит 11 процедур, основными являются процедуры:

integrate: Заменяет в подынтегральном выражении f тригонометрические, обратные тригонометрические, гиперболические, обратные гиперболические и показательные выражения комбинацией логарифмов и экспонент. Вызывает процедуру **mainProcOfInteg**, которая возвращает первообразную функции f . Затем при помощи

процедур из пакета обработки функций `MathPartner`, полученная функция упрощается. Комбинации логарифмов и экспонент заменяются на тригонометрические, обратные тригонометрические гиперболические, обратные гиперболические функции.

`mainProcOfInteg`: Основная процедура интегрирования функции. Возвращает первообразную от входного выражения. Выделяет в подынтегральном выражении список логарифмов и экспонент f_0, \dots, f_n . Этот список отсортирован следующим образом. Чем глубже в аргументах находятся логарифм или экспонента, тем ближе они находятся к началу списка. С помощью процедуры `makeRegularMonomialsSequence` (пакет процедур `StructTheorem`) составляет из списка f_0, \dots, f_n набор регулярных мономов. Отделяет полиномиальную и дробную части интеграла. Интегрирует отдельно полиномиальную часть (пакет процедур `IntPolynom`) и отдельно дробную часть интеграла (пакет процедур `IntegrateFraction`).

3.2. Пакет процедур `StructTheorem`

Содержит 18 процедур, процедура `makeRegularMonomialsSequence` является основной. Она составляет набор регулярных мономов на основе списка всех логарифмов и экспонент, содержащихся в подынтегральном выражении.

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. 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 набор регулярных мономов.

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

3.3. Пакет процедур `IntPolynom`

Содержит 20 процедур, основными являются:

`integPol`: Интегрирование полинома от переменной x .

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

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

3.4. Пакет процедур `IntegrateFractions`

Содержит 17 процедур, основными 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 дробей.

3.5. Общая схема алгоритма.

Входные данные: f — подинтегральная функция, x — переменная интегрирования. Заменяем в подинтегральной функции f тригонометрические, обратные тригонометрические, гиперболические, обратные гиперболические и показательные выражения комбинацией логарифмов и экспонент. Составляем список логарифмов и экспонент, содержащихся в функции f .

При помощи процедуры `makeRegularMonomialsSequence` составляем набор регулярных мономов. Выражаем функцию f через найденные регулярные мономы. Раскладываем функцию f в виде суммы полинома и правильной дроби $f = p + q/r$.

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 `integrateFractions`. 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 `Hermite`), и затем производим вычисление интеграла от дробной функции со свободным от квадратов знаменателем (процедура `LogarithmicPart`). Если интеграл от очередной дроби не выражается через элементарные функции, то интеграл от функции 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 этой системы уравнений выражается через интегралы, не содержащие последний регулярный моном. Каждый такой интеграл рекурсивно вычисляем алгоритмом Риша. Если система уравнений не имеет решений, то интеграл от функции 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 этой системы уравнений выражается через интегралы, не содержащие последний регулярный моном. Каждый такой интеграл рекурсивно вычисляем алгоритмом Риша. Если система уравнений не имеет решений, то интеграл от функции f не выражается через элементарные функции.

Полученный ответ упрощается. Комбинации логарифмов и экспонент заменяются на тригонометрические, обратные тригонометрические, гиперболические, обратные гиперболические функции.

Общая схема алгоритма изображена на рис. 1.

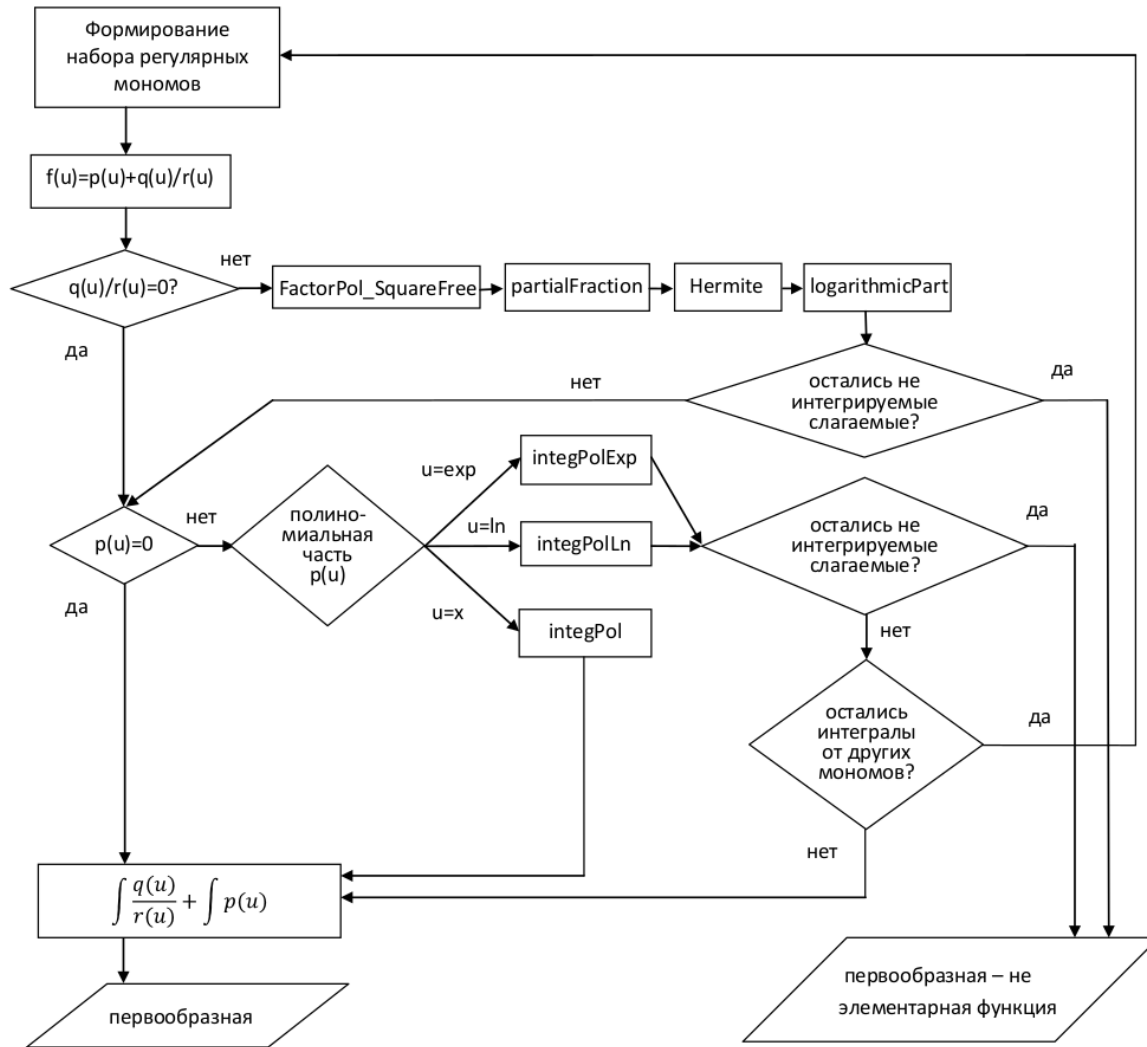


Рис. 1. Общая схема алгоритма

4. Примеры вычисления первообразных

В следующих трех таблицах показаны результаты интегрирования функций разных видов, полученных с помощью созданной в MathPartner библиотеки процедур символьного интегрирования.

Таблица 1
Дробно-рациональные функции

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

Таблица 2
Тригонометрические функции

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

Таблица 3
Другие типы функций

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

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 несмотря 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 профессора Г.И. Малашонка за постановку задачи и многочисленные обсуждения, а также выражает признательность участникам семинара по компьютерной алгебре ТГУ за критику.

Список литературы

- [1] Дж. Дэвенпорт, И. Сирэ, Э. Турнье, *Компьютерная алгебра. Системы и алгоритмы алгебраических вычислений*, Мир, М., 1991.
- [2] М. Bronstein, *Symbolic Integration Tutorial*, ISSAC'98 and Differential Algebra Workshop (Rostock, August 1998; Rutgers, November 2000).
- [3] R. Risch, "The problem of integration in finite terms", *Trans. Amer. Math. Soc.*, 1969, 167–189.
- [4] В. Terelius, *Symbolic Integration*, Master of science thesis, Stockholm, 2009.
- [5] Д. А. Павлов, "Символьное интегрирование", *Компьютерные инструменты в образовании*, 2010, № 2, 38–43.

- [6] Г. И. Малашенок, *Руководство по языку "Mathpar"*, Издательство Тамбовского университета, Тамбов, 2013.
- [7] Е. В. Панкратьев, *Элементы компьютерной алгебры*, МГУ, М., 2007.
- [8] С. М. Тарарова, “К проблеме построения алгоритма символьного интегрирования”, *Вестник Тамбовского университета. Серия: естественные и технические науки*, **17:2** (2012), 607-617.

References

- [1] J. Davenport, Y. Siret, E. Tournier, *Computer Algebra. Systems and Algorithms of Algebraic Computation*, Mir, Moscow, 1991 (In Russian).
- [2] M. Bronstein, *Symbolic Integration Tutorial*, ISSAC'98 and Differential Algebra Workshop (Rostock, August 1998; Rutgers, November 2000).
- [3] R. Risch, “The problem of integration in finite terms”, *Trans. Amer. Math. Soc.*, 1969, 167–189.
- [4] B. Terelius, *Symbolic Integration*, Master of science thesis, Stockholm, 2009.
- [5] D. A. Pavlov, “Symbolic integration”, *Computer Tools in Education*, 2010, №2, 38–43 (In Russian).
- [6] G. I. Malaschonok, *Language Guide "Mathpar"*, Publishing House of Tambov University, Tambov, 2013 (In Russian).
- [7] E. V. Pankrat'yev, *Elements of Computer Algebra*, MSU, Moscow, 2007 (In Russian).
- [8] S. M. Tararova, “To the problem of constructing an algorithm for symbolic integration”, *Tambov University Reports. Series: Natural and Technical Sciences*, **17:2** (2012), 607-617 (In Russian).

Информация об авторе

Корабельников Вячеслав Алексеевич, аспирант, кафедра функционального анализа. Тамбовский государственный университет им. Г.Р. Державина, г. Тамбов, Российская Федерация. E-mail: korabelnikov.va@gmail.com
ORCID: <https://orcid.org/0000-0002-3920-8373>

Поступила в редакцию 23.01.2019 г.
 Поступила после рецензирования 27.02.2019 г.
 Принята к публикации 28.03.2019 г.

Information about the author

Vyacheslav A. Korabelnikov, Post-Graduate Student, Functional Analysis Department. Tambov State University named after G.R. Derzhavin, Tambov, the Russian Federation. E-mail: korabelnikov.va@gmail.com
ORCID: <https://orcid.org/0000-0002-3920-8373>

Received 23 January 2019
 Reviewed 27 February 2019
 Accepted for press 28 March 2019