

АЛГОРИТМ ДЛЯ РЕШЕНИЯ СЕМЕЙСТВА ДИОФАНТОВЫХ УРАВНЕНИЙ
ЧЕТВЕРТОЙ СТЕПЕНИ, УДОВЛЕТВОРЯЮЩИХ УСЛОВИЮ РУНГЕ© 2021 г. Н. Н. Осипов^{а,*}, А. А. Кытманов^{а,**}^а Институт космических и информационных технологий
Сибирского федерального университета
660074 Красноярск, ул. Киренского, 26, Россия

*E-mail: nnosipov@gmail.com

**E-mail: aakytm@gmail.com

Поступила в редакцию 19.07.2020 г.

После доработки 01.09.2020 г.

Принята к публикации 12.09.2020 г.

В статье предлагается алгоритмическая реализация элементарной версии метода Рунге для семейства диофантовых уравнений 4-й степени с двумя неизвестными. К уравнениям рассматриваемого типа сводится любое диофантово уравнение 4-й степени, старшая однородная часть которого разлагается в произведение линейного и кубического многочленов. Компьютерную реализацию алгоритма решения (в его оптимизированном виде) предполагается осуществить в системе компьютерной алгебры PARI/GP.

DOI: 10.31857/S0132347421010076

1. ВВЕДЕНИЕ

В современных системах компьютерной алгебры (таких, как Mathematica, Maple, PARI/GP, SageMath и др.) реализовано только небольшое число известных алгоритмов для решения диофантовых уравнений в целых числах. Как правило, речь идет о системах линейных уравнений с произвольным числом неизвестных, квадратных уравнениях с двумя неизвестными, а также кубических уравнений Туэ (но при некоторых дополнительных условиях, см. [1]). Между тем, имеется довольно широкий класс диофантовых уравнений с двумя неизвестными

$$f(x, y) = 0, \quad (1)$$

для которого можно предложить эффективный (предоставляющий явные верхние границы для целочисленных решений) метод решения — так называемый метод Рунге.

Краткое описание упрощенной версии метода Рунге можно найти в хорошо известных монографиях [2–4]. Необходимо отметить, что оригинальная версия метода Рунге является более общей (см. старую статью Рунге [5] или, в современном изложении, статью [6]). Несмотря на то, что метод Рунге известен уже более ста лет, его реализация в системах компьютерной алгебры весьма

ограничена. Лишь в очень небольшом числе публикаций (см. [7–9] и, особенно, [10]) обсуждаются алгоритмические аспекты реализации этого метода, которые в целом представляются нетривиальными. Последнее частично объясняется следующим обстоятельством.

Пусть многочлен $f(x, y) \in \mathbb{Z}[x, y]$ неприводим. Положим

$$d_0 = \max\{\deg_x f(x, y), \deg_y f(x, y)\}.$$

Можно доказать, что если $f(x, y)$ удовлетворяет так называемому стандартному условию Рунге (см. ниже), то справедлива оценка

$$\max\{|x|, |y|\} < (2d_0)^{18d_0^7} h^{12d_0^6} \quad (2)$$

для всех решений $(x, y) \in \mathbb{Z}^2$ уравнения (1) (см. [6]). Здесь h — высота многочлена $f(x, y)$ (максимум модулей его коэффициентов). В частности, этот результат показывает, что тривиальная реализация — полный перебор в предписанных границах — является практически бессмысленной даже в случае довольно малого d_0 .

Метод Рунге основан на конструктивном способе доказательства теоремы Рунге о конечности

множества решений уравнения (1) в целых числах. В литературе широко известна упрощенная версия этой теоремы, которая и приводится ниже (полную версию, помимо первоисточника, можно найти, например, в [6]).

Пусть $d = \deg f(x, y)$ и $f_d(x, y)$ — старшая однородная часть многочлена $f(x, y)$.

Теорема Рунге. Предположим, что $f_d(x, y)$ разложима в произведение двух взаимно простых многочленов из $\mathbb{Z}[x, y]$ положительных степеней. Тогда уравнение (1) имеет конечное множество решений $(x, y) \in \mathbb{Z}^2$.

Далее для краткости условие теоремы Рунге будем называть *стандартным условием Рунге*. В случае кубических уравнений ($d = 3$) при выполнении стандартного условия Рунге в работе [9] был предложен практически работающий алгоритм решения уравнения (1). Этот алгоритм основан на так называемой элементарной версии метода Рунге для диофантовых уравнений степени не выше четвертой (см. [11]). Алгоритмическая реализация элементарной версии метода Рунге при $d = 4$ пока получена только в некоторых частных случаях (см. [12, 13]). Следует также упомянуть хорошо известный элементарный алгоритм решения для наиболее простого типа уравнений 4-й степени, к которым применим метод Рунге (этот алгоритм подробно описан в [7]). Как отмечено в [10], имеет смысл “избегать разложений в ряды Пюизо и алгебраических коэффициентов”, поскольку это обычно приводит к плохим оценкам типа (2). Собственно, данное обстоятельство и составляет основную трудность в алгоритмической реализации метода Рунге.

Элементарная версия метода Рунге для диофантовых уравнений малой степени базируется на специальных параметризациях, которые позволяют перечислить возможные целочисленные решения. Как результат, решение данного диофантова уравнения может быть сведено к решению конечного множества уравнений (как правило, квадратных или кубических) с одним неизвестным в целых числах. Эта идея была воплощена в работах [9, 12, 13]. Особенно хорошо метод работает в случае кубических диофантовых уравнений, где необходимо решать только квадратные уравнения (и, следовательно, все сводится к алгоритмически простой задаче извлечения квадратных корней).

Пример 1. а) Для решения диофантова уравнения

$$x(y^2 - 2x^2) + Hx + y + 1 = 0$$

(см. пример 9 из [9]) достаточно решить порядка $|H|^{1/4}$ квадратных уравнений с одним неизвестным. В частности, это позволяет за приемлемое время решить уравнение для каждого H в диапазоне $-10^{10} \leq H < 0$.

б) Аналогично, диофантово уравнение

$$xy^2 = 2Hx^2 + y$$

сводится к решению порядка $|H|^{1/3}$ квадратных уравнений. Это не только теоретически проще, но и гораздо быстрее, чем нахождение целых точек на подходящей кривой Морделла (как предлагалось в [14], см. раздел 6.1).

в) Диофантово уравнение

$$z = y^4 + 8Hy^3 - 12y^2 + 4$$

(см. [15]) заменой

$$z = x + y^2 + 4Hy - 8H^2 - 6$$

приводится к виду (1), где

$$f(x, y) = 2xy^2 + \dots$$

(см. в других обозначениях пример 1.1 из [13]). Оптимизированный алгоритм из [9] сводит решение уравнения (1) с такой левой частью к решению порядка $|H|^2$ квадратных уравнений. Но и этот результат можно улучшить, понизив порядок числа решаемых квадратных уравнений до $|H|^{3/2}$. Для сравнения отметим, что стандартный алгоритм решения (см. [7]) потребовал бы решения порядка $|H|^3$ квадратных уравнений.

В настоящей статье рассматривается новое семейство диофантовых уравнений (1) 4-й степени с левой частью вида

$$f(x, y) = x(ax^3 + bx^2y + cxy^2 + dy^3) + xg(x, y) + h(y), \quad (3)$$

где, в свою очередь,

$$g(x, y) = p_0x^2 + (p_1y + p_2)x + p_3y^2 + p_4y + p_5, \\ h(y) = Ay^2 + By + C.$$

Коэффициент d предполагается ненулевым, что и гарантирует применимость метода Рунге. Как по-

казано в [11], произвольное уравнение со старшей однородной частью

$$f_4(x, y) = (a_1x + b_1y)(a_2x^3 + b_2x^2y + c_2xy^2 + d_2y^3)$$

сводится к уравнению с левой частью (3).

Статья организована следующим образом.

В разделе 2 предлагается алгоритм решения уравнения (1) с левой частью (3). Корректность его работы гарантируется теоремой 1. Технически данный алгоритм несколько отличается от подобных алгоритмов (см. [9, 12]), так как требует решения некоторого количества уравнений 4-й степени с одним неизвестным в целых числах. Это необходимо принимать во внимание, если мы хотим корректно оценить сложность алгоритма. Как и в работе [13], с этой целью вводится дополнительный параметр – *весовой коэффициент*, который зависит от той системы компьютерной алгебры, в которой мы планируем реализовывать алгоритм (PARI/GP, см. [1]). Далее можно оптимизировать алгоритм уже известным способом (см. [9, 13]). К сожалению, многочисленные технические детали на этом пути пока не преодолены, поэтому мы приводим только один пример такой оптимизации.

В разделе 3 делаются некоторые замечания относительно полученных результатов. Кроме того, обсуждаются дальнейшие перспективы применения элементарной версии метода Рунге для решения оставшихся типов диофантовых уравнений 4-й степени.

2. АЛГОРИТМ

Будем считать $A \neq 0$ (в случае $A = 0$ предлагаемый метод можно упростить, см. раздел 3). Напомним, что $d \neq 0$.

Предполагая $x \neq 0$, рассмотрим число

$$l = \frac{h(y)}{x}.$$

Очевидно, что l должно быть целым для каждого решения $(x, y) \in \mathbb{Z}^2$ с $x \neq 0$. Поделив на x обе части уравнения, получим

$$ax^3 + bx^2y + cxy^2 + dy^3 + g(x, y) + l = 0.$$

Это равенство влечет сравнение

$$dy^3 + p_3y^2 + p_4y + p_5 + l \equiv 0 \pmod{x}$$

в кольце $\mathbb{Z}$. Далее имеем

$$A^2(dy^3 + p_3y^2 + p_4y + p_5) \equiv B_1y + C_1 \pmod{h(y)}$$

для некоторых целых чисел B_1 и C_1 (это сравнение в кольце $\mathbb{Z}[y]$). А именно:

$$\begin{aligned} B_1 &= p_4A^2 - (p_3B + dC)A + dB^2, \\ C_1 &= p_5A^2 - p_3AC + dBC. \end{aligned} \quad (4)$$

Ввиду того, что $h(y) \equiv 0 \pmod{x}$, получим еще одно сравнение $A^2l + B_1y + C_1 \equiv 0 \pmod{x}$ (оба сравнения в кольце $\mathbb{Z}$). Наконец, положим

$$k = \frac{A^2l + B_1y + C_1}{x} = \frac{A^3y^2 + B_1xy + A^2By + C_1x + A^2C}{x^2}.$$

Ясно, что k также должно быть целым числом. Таким образом, приходим к следующему результату.

Теорема 1. Пусть $(x, y) \in \mathbb{Z}^2$ – произвольное решение уравнения с $x \neq 0$. Тогда число

$$k = \frac{A^3y^2 + B_1xy + A^2By + C_1x + A^2C}{x^2} \quad (5)$$

является целым. Здесь B_1 и C_1 определяются равенствами (4).

Отметим, что формальное доказательство теоремы 1 легко получить, опираясь на символьные вычисления в какой-нибудь системе компьютерной алгебры. Выразим из уравнения коэффициент C :

$$C = -x(ax^3 + bx^2y + cxy^2 + dy^3) - xg(x, y) - Ay^2 - By.$$

Далее подставим это выражение вместо C в правую часть (5). После сокращения на x^2 мы получим явное (но довольно громоздкое) выражение для k в виде многочлена из $\mathbb{Z}[x, y]$. Теперь очевидно, что значение k должно быть целым, поскольку $(x, y) \in \mathbb{Z}^2$. Отметим, что подобные “механические” рассуждения очень просты логически и актуальны всякий раз, когда мы хотим верифицировать сложное “синтетическое” доказательство или быстро удостовериться в справедливости гипотезы (см., например, [16]).

Пример 2. Для уравнения с левой частью

$$f(x, y) = x(y^3 - 2x^3) + x^3 + y^2 + 1$$

описанная выше процедура приводит к следующему результату:

$$k = \frac{y^2 - xy + 1}{x^2} = y^4 - (2x^3 - x^2)y + 2x^2 - x.$$

Фактически речь идет о равенстве

$$y^2 - xy + 1 = x^2(y^4 - (2x^3 - x^2)y + 2x^2 - x)$$

в кольце классов вычетов $\mathbb{Z}[x, y]/\langle f(x, y) \rangle$.

Дальнейшие рассуждения базируются на следующей идее. Предположим, что многочлен

$$\phi(t) = a + bt + ct^2 + dt^3 \in \mathbb{Z}[t]$$

неприводим. Пусть α — любой вещественный корень $\phi(t)$. Для соответствующей ветви $y = \Psi(x)$ алгебраической функции, определяемой уравнением, имеет место оценка

$$\Psi(x) \sim \alpha x, \quad x \rightarrow \infty.$$

Как следствие, функция

$$R(x) = \frac{A^3\Psi^2(x) + B_1x\Psi(x) + A^2B\Psi(x) + C_1x + A^2C}{x^2}$$

при $x \rightarrow \infty$ имеет пределом число

$$L(\alpha) = A^3\alpha^2 + B_1\alpha.$$

Таким образом, для любого числа $m > 0$ найдется такое число $Q_\alpha(m) > 0$, что

$$|R(x) - L(\alpha)| < m$$

для всех x с условием $|x| > Q_\alpha(m)$. Теперь можно предложить следующий алгоритм для решения уравнения (1) с левой частью (3).

Алгоритм решения. Для всех вещественных корней α многочлена $\phi(t)$ сделать:

1. Выбрать $m > 0$ и найти $Q_\alpha(m)$.
2. Для всех целых чисел x с условием

$$|x| \leq Q_\alpha(m)$$

решить уравнение $f(x, y) = 0$ как кубическое относительно y в целых числах. Найденные пары $(x, y) \in \mathbb{Z}^2$ добавить в множество решений.

3. Для всех целых чисел k с условием

$$|k - L(\alpha)| < m$$

найти все пары $(x, y) \in \mathbb{Z}^2$, удовлетворяющие системе уравнений

$$\begin{aligned} f(x, y) &= 0, \\ A^3y^2 + B_1xy + A^2By + C_1x + A^2C &= kx^2, \end{aligned} \quad (6)$$

и добавить их в множество решений.

Первая проблема в реализации нашего алгоритма состоит в том, что необходимо задать $Q_\alpha(m)$ как явную функцию *управляющего параметра* m . Эта проблема технически довольно сложна и пока полностью не решена (мы кратко обсудим это в разделе 3).

Вторую проблему можно сформулировать следующим образом: как выбрать оптимальное значение m ? Более формально, для каждого фиксированного α мы хотим минимизировать *функцию затрат* вида

$$\text{cost}(m) = 2qm + 2Q_\alpha(m), \quad (7)$$

где весовой коэффициент q предлагается определять экспериментально — в зависимости от той системы компьютерной алгебры, в которой предполагается реализовывать алгоритм (PARI/GP). Более точно, в качестве q следует взять отношение сложности решения систем уравнений вида (6) к сложности решения кубических уравнений (в обоих случаях — в целых числах).

Рассмотрим систему уравнений (6) более подробно. Исключив y , мы получим уравнение 4-й степени относительно x вида

$$K_0x^4 + K_1x^3 + K_2x^2 + K_3x + K_4 = 0, \quad (8)$$

где $K_j = K_j(k)$ — многочлены от k с целыми коэффициентами (мы не приводим их явных и довольно громоздких выражений, отметим лишь, что $\deg K_0(k) = 3$ и $\deg K_j(k) \leq 2$ для остальных j). Таким образом, необходимо определить, насколько более сложной является проблема решения уравнений четвертой степени в целых числах по сравнению с аналогичной проблемой для кубических уравнений. В PARI/GP мы намерены решать обе проблемы с помощью функции `nfroots`, которая позволяет, в частности, находить все рациональные корни.

нальные корни многочлена от одной переменной с целыми коэффициентами. Как итог, весовой коэффициент q может быть выбран с помощью компьютерных экспериментов и предварительного анализа возможной высоты многочлена в левой части равенства (8).

К сожалению, ожидаемое аналитическое выражение для $Q_\alpha(m)$ (см. раздел 3) не позволяет минимизировать функцию затрат (7) символьными методами. Обозначим через m^* значение m , доставляющее глобальный минимум $\text{cost}(m)$. Имеет смысл сосредоточиться на “разумных” оценках для m^* и $\text{cost}(m^*)$ — тогда мы сможем оценить теоретическую сложность так называемого *оптимизированного алгоритма* (алгоритма, в котором $m = m^*$). Практически мы можем найти m^* с помощью любого приближенного метода (при этом желательно решить проблему локализации для m^* аналитически).

Приведем один пример с целью продемонстрировать все трудности, которые необходимо преодолеть в общем случае.

Пример 3. Сконструируем “вручную” оптимизированный алгоритм для семейства уравнений

$$x(y^3 - 2x^3) + x^3 + y^2 + H = 0, \quad (9)$$

где $H > 0$. Единственный вещественный корень $\phi(t) = t^3 - 2$ есть $\alpha = 2^{1/3}$. Тогда

$$L(\alpha) = 2^{2/3} - 2^{1/3}H = l(H).$$

Кроме того, мы имеем

$$k = \frac{y^2 - Hxy + H}{x^2}.$$

Коэффициенты уравнения (8) таковы:

$$\begin{aligned} K_0 &= k^3 + 6Hk + 2H^3 - 4, \\ K_1 &= -3Hk - H^3 + 4, \\ K_2 &= -4Hk^2 + 4k - 4H^2 - 1, \\ K_3 &= -2k + 2H^2, \\ K_4 &= -k^2 + 2H^2k - H^4. \end{aligned}$$

Для $y = \Psi(x)$ можно доказать следующее: если $|x| > H^{1/2}$, то справедлива оценка

$$\left| \frac{y}{x} - 2^{1/3} \right| < \frac{1}{|x|}. \quad (10)$$

С помощью (10) нетрудно получить требуемую оценку для $|k - l(H)|$. А именно,

$$|k - l(H)| < H^{1/2} + 6$$

для тех x , для которых $|x| > H^{1/2}$. Поэтому, положив $m = H^{1/2}$, получим

$$\text{cost}(m) \asymp H^{1/2}, \quad H \rightarrow \infty.$$

Поскольку при $x = H^{1/2}$ имеем

$$|k - l(H)| \asymp H^{1/2}, \quad H \rightarrow \infty,$$

оценка для $\text{cost}(m^*)$ имеет такой же порядок по H . Тем самым оптимизированный алгоритм для семейства уравнений (9) построен.

3. ЗАМЕЧАНИЯ

При $A = 0$ предложенный алгоритм будет работать быстрее, чем в общем случае. Действительно, тогда второе уравнение в (6) сводится к линейному, так что уравнение (8) становится кубическим по x (но по-прежнему остается кубическим по k).

Конечно, узкое место нашего алгоритма — это получение явного символьного выражения для $Q_\alpha(m)$. Очевидный подход (успешно примененный в [9]) состоит в том, чтобы решить уравнение (8) как кубическое относительно k по формуле Кардано. Однако из-за громоздких выражений для коэффициентов такой подход представляется непродуктивным. Более реалистичный путь — решить исходное уравнение (1) как кубическое по y и далее явно оценить разность $y/x - \alpha$ при $x \rightarrow \infty$. Иными словами, мы можем конкретизировать теоретическую оценку

$$\frac{y}{x} - \alpha = \frac{\Psi(x)}{x} - \alpha = O\left(\frac{1}{x}\right), \quad x \rightarrow \infty,$$

подобно тому, как это было сделано в примере 3 (т. е. получить оценку типа (10) для всех достаточно больших x , причем с указанием явной границы для этих x). С этой целью можно рассмотреть несколько первых членов степенного разложения для функции $\Psi(x)/x$ при $x \rightarrow \infty$. В общем случае, помимо (очень осторожного) обращения к рядам Пуизо, можно воспользоваться алгоритмами степенной геометрии, хорошо себя зарекомендовавшими при разрешении алгебраических особенностей (см. [17, 18]).

Чтобы проиллюстрировать возможные трудности, вновь обратимся к примеру 3. Имеем

$$\frac{\Psi(x)}{x} - 2^{1/3} = -\frac{2^{1/3}}{6x} + O\left(\frac{1}{x^2}\right), \quad x \rightarrow \infty.$$

Здесь может показаться, что существует такая абсолютная константа M , что для всякого $x \neq 0$ верна оценка

$$\left| \frac{y}{x} - 2^{1/3} \right| < \frac{M}{|x|}. \quad (11)$$

Однако это не так — из-за того, что коэффициенты разложения зависят от параметра H , который может быть сколь угодно большим. В самом деле, если $x = 1$, то, как можно убедиться, $y \sim -H^{1/3}$ при $H \rightarrow \infty$. Таким образом, в общем случае нам следует указать явную нижнюю границу для $|x|$, которая гарантировала бы оценку типа (11) для $y/x - \alpha$. В конкретной ситуации (пример 3) такую границу указать легко.

В заключение скажем несколько слов о дальнейших перспективах элементарной версии метода Рунге, предложенной в [11].

По-видимому, самым нетривиальным в плане алгоритмической реализации является последний неисследованный случай, когда старшая однородная часть уравнения (1) допускает разложение вида

$$f_4(x, y) = (a_1x^2 + b_1xy + c_1y^2)(a_2x^2 + b_2xy + c_2y^2).$$

Обнадёживающим моментом можно считать то, что принципиально новых проблем в реализации здесь не предвидится, а те проблемы, что описаны выше, могут быть решены.

Работа поддержана Красноярским математическим центром, финансируемым Минобрнауки РФ в рамках мероприятий по созданию и развитию региональных НОМЦ (Соглашение 075-02-2020-1534/1).

СПИСОК ЛИТЕРАТУРЫ

1. <https://pari.math.u-bordeaux.fr>
2. *Mordell L.J.* Diophantine equations. London: Academic Press Inc., 1969.
3. *Спринджук В.Г.* Классические диофантовы уравнения от двух неизвестных. М.: Наука, 1982.
4. *Masser D.W.* Auxiliary Polynomials in Number Theory. Cambridge University Press, 2016.
5. *Runge C.* Ueber ganzzahlige Lösungen von Gleichungen zwischen zwei Veränderlichen // J. reine und angew. Math. 1887. V. 100. P. 425–435.
6. *Walsh P.G.* A quantitative version of Runge's theorem on diophantine equations // Acta Arith. 1992. V. 62. P. 157–172.
7. *Poulakis D.* A simple method for solving the diophantine equation $Y^2 = X^4 + aX^3 + bX^2 + cX + d$ // Elem. Math. 1999. V. 54. P. 32–36.
8. *Tengely Sz.* On the Diophantine equation $F(x) = G(y)$ // Acta Arith. 2003. V. 110. P. 185–200.
9. *Osipov N.N., Gulnova B.V.* An algorithmic implementation of Runge's method for cubic diophantine equations // J. Sib. Fed. Univ. Math. Phys. 2018. V. 11. 2. P. 137–147.
10. *Beukers F., Tengely Sz.* An implementation of Runge's method for diophantine equations // Preprint <https://arxiv.org/abs/math/0512418v1>
11. *Осипов Н.Н.* Метод Рунге для уравнений 4-й степени: элементарный подход // Математическое просвещение. Сер. 3. Вып. 19. М.: МЦНМО, 2015. С. 178–198.
12. *Osipov N.N., Medvedeva M.I.* An elementary algorithm for solving a diophantine equation of degree four with Runge's condition // J. Sib. Fed. Univ. Math. Phys. 2019. V. 12. № 3. P. 331–341.
13. *Osipov N.N., Dalinkevich S.D.* An algorithm for solving a quartic diophantine equation satisfying Runge's condition // Lecture Notes in Computer Science. V. 11661. Springer, 2019. P. 377–392.
14. *Stroeker R.J., de Weger B.M.M.* Solving elliptic diophantine equations: the general cubic case // Acta Arith. 1999. V. 87. P. 339–365.
15. *Masser D.W.* Polynomial Bounds for Diophantine Equations // Amer. Math. Monthly. 1986. V. 93. P. 486–488.
16. *Осипов Н.Н.* О механическом доказательстве планиметрических теорем рационального типа // Программирование. 2014. № 2. С. 41–50.
17. *Брюно А.Д.* Степенная геометрия в алгебраических и дифференциальных уравнениях. М.: Физматлит, 1998.
18. *Брюно А.Д., Бахтин А.Б.* Разрешение алгебраической сингулярности алгоритмами степенной геометрии // Программирование. 2012. № 2. С. 11–28.