ISSN 0361-7688, Programming and Computer Software, 2021, Vol. 47, No. 1, pp. 29–33. © Pleiades Publishing, Ltd., 2021.
Russian Text © The Author(s), 2021, published in Programmirovanie, 2021, Vol. 47, No. 1.

An Algorithm for Solving a Family of Fourth-Degree Diophantine Equations that Satisfy Runge's Condition

N. N. Osipov[a],* and A. A. Kytmanov[a],**

[a]*School of Space and Information Technology, Siberian Federal University, ul. Kirenskogo 26, Krasnoyarsk, 660074 Russia*

**e-mail: nnosipov@gmail.com*

***e-mail: aakytm@gmail.com*

Received July 19, 2020; revised September 1, 2020; accepted September 12, 2020

Abstract—This paper proposes an algorithmic implementation of the elementary version of Runge's method for a family of fourth-degree Diophantine equations in two unknowns. Any Diophantine equation of the fourth degree the leading homogeneous part of which is decomposed into a product of linear and cubic polynomials can be reduced to equations of the type considered in this paper. The corresponding algorithm (in its optimized version) is implemented in the PARI/GP computer algebra system.

DOI: 10.1134/S0361768821010060

1. INTRODUCTION

Modern computer algebra systems (e.g., Mathematica, Maple, PARI/GP, SageMath, etc.) include only a small number of algorithms for solving Diophantine equations in integers. Generally, these are systems of linear equations in an arbitrary number of unknowns, quadratic equations in two unknowns, and cubic Thue equations (under some additional conditions, see [1]). Meanwhile, there is a fairly large class of Diophantine equations in two unknowns

$$f(x, y) = 0, \tag{1}$$

for which an effective method (which yields explicit upper bounds for integer solutions), the so-called Runge's method, can be applied.

A brief description of a simplified version of Runge's method can be found in the well-known monographs [2–4]. It should be noted that the original version of Runge's method is more general (see Runge's old paper [5] or a modern paper [6]). Even though Runge's method has been known for over a hundred years, its implementation in computer algebra systems is quite limited. Only a few publications (see [7–9] and especially [10]) discuss algorithmic aspects of its implementation that, by and large, seem nontrivial. This is partially explained by the following.

Suppose that the polynomial $f(x, y) \in \mathbb{Z}[x, y]$ is irreducible and

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

It can be proved that, if $f(x, y)$ satisfies Runge's standard condition (see below), then the estimate

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

holds for all solutions $(x, y) \in \mathbb{Z}^2$ of Eq. (1) (see [6]). Here, h is the height of the polynomial $f(x, y)$ (the maximum of the absolute values of its coefficients). This result suggests that the trivial implementation using brute force within the bounds mentioned above makes almost no sense even for fairly small d_0.

Runge's method is based on a constructive proof of Runge's theorem on the finiteness of a set of solutions to Eq. (1) in integers. A simplified version of this theorem (cited below) is well known; its full version can be found in the original paper and, e.g., in [6].

Suppose that $d = \deg f(x, y)$ and $f_d(x, y)$ is the leading homogeneous part of the polynomial $f(x, y)$.

Runge's theorem. Suppose that $f_d(x, y)$ is decomposable into a product of two relatively prime polynomials in $\mathbb{Z}[x, y]$ of positive degrees. Then, Eq. (1) has a finite set of solutions $(x, y) \in \mathbb{Z}^2$.

For brevity, we hereinafter refer to the condition of Runge's theorem as *Runge's standard condition*. For cubic equations ($d = 3$), under Runge's standard condition, a realistic algorithm for solving Eq. (1) was proposed in [9]. This algorithm is based on the so-called elementary version of Runge's method for Diophantine equations of degrees not higher than four (see [11]). Currently, the elementary version of Runge's method for $d = 4$ has been algorithmically implemented only in some special cases [12, 13]. We should also mention the well-known elementary algorithm for solving fourth-degree equations of the simplest

type to which Runge's method can be applied (this algorithm was described in detail in [7]). As noted in [10], it makes sense to avoid the use of Puiseux series and algebraic coefficients because they usually lead to bad estimates of type (2). In fact, this is the main challenge in algorithmic implementation of Runge's method.

The elementary version of Runge's method for Diophantine equations of low degrees is based on special parametrizations used to enumerate possible integer solutions. Hence, the solution of a Diophantine equation can be reduced to solving a finite set of (as a rule, quadratic or cubic) equations in one unknown over the integers. This idea was implemented in [9, 12, 13]. The method works especially well in the case of cubic Diophantine equations, where only quadratic equations need to be solved (therefore, the problem is reduced to algorithmically simple extraction of square roots).

Example 1. (a) To solve the Diophantine equation

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

(see Example 9 from [9]), it is sufficient to solve approximately $|H|^{1/4}$ quadratic equations in one unknown. In particular, this allows one to solve the equation for each H in the range of $-10^{10} \le H < 0$ within a reasonable time.

(b) Similarly, the Diophantine equation

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

is reduced to solving approximately $|H|^{1/3}$ quadratic equations. In addition to being theoretically simpler, this approach is also much faster than finding integer points on a Mordell curve (as was proposed in [14, Section 6.1]).

(c) The Diophantine equation

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

(see [15]) is reduced by the substitution

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

to type (1), where

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

(see Example 1.1 in other notation from [13]). The optimized algorithm from [9] reduces the solution of Eq. (1) with this left-hand side to the solution of approximately $|H|^2$ quadratic equations. However, this result can be further improved by lowering the order of the number of quadratic equations to be solved to $|H|^{3/2}$. For comparison, the standard solution algorithm [7] would require solving approximately $|H|^3$ quadratic equations.

In this paper, we consider a new family of Diophantine equations (1) of the fourth degree with the left-hand side of the form

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

where, in turn,

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

The coefficient d is assumed to be nonzero, which guarantees the applicability of Runge's method. As shown in [11], an arbitrary equation with a leading homogeneous part

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

is reduced to an equation with left-hand side (3).

This paper is organized as follows. Section 2 proposes an algorithm for solving Eq. (1) with left-hand side (3). Its correctness is guaranteed by Theorem 1. Technically, this algorithm somewhat differs from similar algorithms (see [9, 12]) as it requires solving a number of fourth-degree equations in one unknown over the integers. This must be taken into account to correctly estimate the complexity of the algorithm. As in [13], for this purpose, we introduce an additional parameter—*weight coefficient*—that depends on a computer algebra system used to implement the algorithm (in our case, it is PARI/GP; see [1]). Then, the algorithm can be optimized in the well-known way (see [9, 13]). Unfortunately, numerous technical difficulties associated with its optimization have not yet been overcome, which is why we confine ourselves only to a single example of such optimization.

In Section 3, we make some remarks on the results. In particular, we discuss further application of the elementary version of Runge's method to solve fourth-degree Diophantine equations of the remaining types.

2. ALGORITHM

Suppose that $A \ne 0$ (in the case of $A = 0$, the proposed method can be simplified, see Section 3). Recall that $d \ne 0$.

Assuming that $x \ne 0$, we consider

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

Obviously, l must be integer for each solution $(x, y) \in \mathbb{Z}^2$ with $x \ne 0$. Having divided both sides of the equation by x, we obtain

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

This equality implies a congruence

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

in the ring $\mathbb{Z}$. Next, we have

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

for some integers B_1 and C_1 (which is a congruence in the ring $\mathbb{Z}[y]$). In particular,

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

With $h(y) \equiv 0 \pmod{x}$, we obtain another congruence $A^2l + B_1y + C_1 \equiv 0 \pmod{x}$ (both congruences are in the ring $\mathbb{Z}$). Finally, we assume that

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

Clearly, k must also be an integer. Thus, we arrive at the following result.

Theorem 1. Suppose that $(x, y) \in \mathbb{Z}^2$ is an arbitrary solution of the equation with $x \neq 0$. Then, the number

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

is an integer. Here, B_1 and C_1 are determined by equalities (4).

Note that Theorem 1 is easy to formally prove based on symbolic computations in some computer algebra system. Let us express the coefficient C as

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

Then, this expression is substituted for C in the right-hand side of (5). Upon canceling by x^2, we obtain an explicit (yet rather cumbersome) expression for k in the form of a polynomial from $\mathbb{Z}[x, y]$. Now, it is obvious that k must be an integer because $(x, y) \in \mathbb{Z}^2$. It should be noted that this "mechanical" reasoning is very straightforward in terms of its logic and is applicable whenever we want to verify a complex "synthetic" proof or quickly validate a hypothesis (see, for example, [16]).

Example 2. For the equation with a left-hand side

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

the procedure described above yields

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

In fact, we have

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

in the residue-class ring $\mathbb{Z}[x, y]/\langle f(x, y)\rangle$.

Further reasoning is based on the following idea. Suppose that the polynomial

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

is irreducible. Suppose also that α is any real root of $\phi(t)$. For the corresponding branch $y = \Psi(x)$ of the algebraic function defined by the equation, the following estimate holds:

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

Hence, for $x \to \infty$, the function

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

has the limit

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

Thus, for any number $m > 0$, there is a number $Q_\alpha(m) > 0$ such that

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

for all x that satisfy the condition $|x| > Q_\alpha(m)$. Now, we can proceed to the algorithm for solving Eq. (1) with left-hand side (3).

Solution algorithm. For any real root α of the polynomial $\phi(t)$, do

1. choose $m > 0$ and compute $Q_\alpha(m)$;
2. for any integer x satisfying

$$|x| \le Q_\alpha(m),$$

solve $f(x, y) = 0$ as a cubic equation in y over the integers and add the found pairs $(x, y) \in \mathbb{Z}^2$ into a set of solutions;

3. for any integer k with

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

find all pairs $(x, y) \in \mathbb{Z}^2$ that satisfy the system of equations

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

and add them to the set of solutions.

The first problem in the implementation of our algorithm is associated with setting $Q_\alpha(m)$ as an explicit function of the *control parameter* m. This problem is technically complex and has not yet been fully resolved (we briefly discuss this in Section 3).

The second problem can be formulated as follows: how to choose the optimal value of m? More formally,

for each fixed α, we aim at minimizing a *cost function* of the form

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

where the weight coefficient q can be determined by experiment depending on the computer algebra system used to implement the algorithm (in our case, PARI/GP). More precisely, as q, one should take a ratio between the complexity of solving systems of form (6) and the complexity of solving cubic equations (in both cases, over the integers).

Let us consider the system of equations (6) in more detail. Having eliminated y, we obtain a fourth-degree equation in x of the form

$$K_0x^4 + K_1x^3 + K_2x^2 + K_3x + K_4 = 0, \tag{8}$$

where $K_j = K_j(k)$ are polynomials in k with integer coefficients. Here, we do not cite their explicit and rather cumbersome expressions; note only that $\deg K_0(k) = 3$ and $\deg K_j(k) \le 2$ for the other j. Thus, it is required to determine to what extent the problem of solving fourth-degree equations over the integers is more complex than the same problem for cubic equations. In PARI/GP, we solve both the problems by using the I nfroots function, which allows us to find all rational roots of a polynomial in one variable with integer coefficients. Thus, the weight coefficient q can be determined by computer experiments and preliminary analysis of the possible height of the polynomial on the left-hand side of Eq. (8).

Unfortunately, the expected analytical expression for $Q_\alpha(m)$ (see Section 3) does not allow cost function (7) to be minimized using symbolic methods. Let us denote the value of m that delivers the global minimum of $\mathrm{cost}(m)$ by m^*. It makes sense to focus on "reasonable" estimates for m^* and $\mathrm{cost}(m^*)$; then, we can estimate the theoretical complexity of the so-called *optimized algorithm* (an algorithm in which $m = m^*$). In practice, we can find m^* using any approximate method (in this case, it is desirable to solve the localization problem for m^* analytically).

To illustrate the difficulties that must be overcome in the general case, we consider the following example.

Example 3. Let us "manually" construct an optimized algorithm for a family of equations

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

where $H > 0$. The only real root of $\phi(t) = t^3 - 2$ is $\alpha = 2^{1/3}$. Then,

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

In addition, we have

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

The coefficients of Eq. (8) are as follows:

$$K_0 = k^3 + 6Hk + 2H^3 - 4,$$
$$K_1 = -3Hk - H^3 + 4,$$
$$K_2 = -4Hk^2 + 4k - 4H^2 - 1,$$
$$K_3 = -2k + 2H^2,$$
$$K_4 = -k^2 + 2H^2k - H^4.$$

For $y = \Psi(x)$, we can prove that, if $|x| > H^{1/2}$, then the following estimate holds:

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

Using (10), it is easy to obtain the desired estimate for $|k - l(H)|$, more specifically,

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

for those x that satisfy $|x| > H^{1/2}$. Hence, assuming that $m = H^{1/2}$, we obtain

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

Since, for $x = H^{1/2}$, we have

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

the estimate for $\mathrm{cost}(m^*)$ has the same order in H. Thus, the optimized algorithm for the family of equations (9) is constructed.

3. REMARKS

For $A = 0$, the proposed algorithm runs faster than in the general case. Indeed, the second equation in (6) is reduced to a linear one, and Eq. (8) becomes cubic in x (while remaining cubic in k).

Clearly, a bottleneck of our algorithm is the derivation of an explicit symbolic expression for $Q_\alpha(m)$. An obvious approach (successfully applied in [9]) is to solve Eq. (8) as a cubic equation in k by using the Cardano formula. However, because of the cumbersome expressions for the coefficients, this approach seems counterproductive. A more realistic way is to solve original equation (1) as a cubic one in y and, then, explicitly estimate the difference $y/x - \alpha$ for $x \to \infty$. In other words, we can concretize the theoretical estimate

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

in the same way as in Example 3 (i.e., obtain estimates of type (10) for all sufficiently large x while specifying an explicit bound for these x). For this purpose, we can consider several first terms of the power series expansion for the function $\Psi(x)/x$ with $x \to \infty$. In the general case, in addition to the (very careful) use of Puiseux series, we can employ algorithms of power geom-

etry that proved useful in resolving algebraic singularities (see [17, 18]).

To illustrate possible difficulties, let us return to Example 3. We have

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

Here, it may seem that there is an absolute constant M such that the following estimate holds for any $x \neq 0$:

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

However, this is not the case because the expansion coefficients depend on the parameter H, which can be arbitrarily large. Indeed, if $x = 1$, then $y \sim -H^{1/3}$ for $H \to \infty$. Thus, in the general case, we should specify an explicit lower bound for $|x|$ that would guarantee an estimate of type (11) for $y/x - \alpha$. In this particular case (Example 3), this bound is easy to set.

In conclusion, we outline future prospects for the elementary version of Runge's method proposed in [11].

Apparently, in terms of algorithmic implementation, the most nontrivial case is the one where the leading homogeneous part of Eq. (1) admits an expansion of the form

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

With that said, we expect that radically new difficulties in the implementation will not occur, which is rather encouraging, whereas the problems described above can be resolved.

FUNDING

This work was supported by the Krasnoyarsk Mathematical Center and financed by the Ministry of Science and Higher Education of the Russian Federation in the framework of the establishment and development of regional Centers for Mathematics Research and Education (agreement no. 075-02-2020-1534/1).

REFERENCES

1. PARI/GP homepage. https://pari.math.u-bordeaux.fr.
2. Mordell, L.J., *Diophantine Equations,* London: Academic, 1969.
3. Sprindzhuk, V.G., *Klassicheskie diofantovy uravneniya ot dvukh neizvestnykh* (Classical Diophantine Equations in Two Unknowns), Moscow: Nauka, 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 Angew. Math.,* 1887, vol. 100, pp. 425–435.
6. Walsh, P.G., A quantitative version of Runge's theorem on Diophantine equations, *Acta Arith.,* 1992, vol. 62, pp. 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, vol. 54, pp. 32–36.
8. Tengely, Sz., On the Diophantine equation $F(x) = G(y)$, *Acta Arith.,* 2003, vol. 110, pp. 185–200.
9. Osipov, N.N. and Gulnova, B.V., An algorithmic implementation of Runge's method for cubic Diophantine equations, *J. Sib. Fed. Univ. Math. Phys.,* 2018, vol. 11, pp. 137–147.
10. Beukers, F. and Tengely, Sz., An implementation of Runge's method for Diophantine equations. https://arxiv.org/abs/math/0512418v1.
11. Osipov, N.N., Runge's method for equations of degree four: An elementary approach, *Mat. Prosveshchenie,* 2015, vol. 3, no. 19, pp. 178–198.
12. Osipov, N.N. and 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, vol. 12, pp. 331–341.
13. Osipov, N.N. and Dalinkevich, S.D., An algorithm for solving a quartic Diophantine equation satisfying Runge's condition, *Lect. Notes Comput. Sci.,* 2019, vol. 11661, pp. 377–392.
14. Stroeker, R.J. and de Weger, B.M.M., Solving elliptic Diophantine equations: The general cubic case, *Acta Arith.,* 1999, vol. 87, pp. 339–365.
15. Masser, D.W., Polynomial bounds for Diophantine equations, *Am. Math. Monthly,* 1986, vol. 93, pp. 486–488.
16. Osipov, N.N., Mechanical proof of planimetric theorems of rational type, *Program. Comput. Software,* 2014, vol. 40, pp. 71–78.
17. Bryuno, A.D., *Stepennaya geometriya v algebraicheskikh i differentsial'nykh uravneniyakh* (Power Geometry in Algebraic and Differential Equations), Moscow: Fizmatlit, 1998.
18. Bryuno, A.D. and Bakhtin, A.B., Resolution of an algebraic singularity by power geometry algorithms, *Program. Comput. Software,* 2012, vol. 38, no. 2, pp. 57–72.

Translated by Yu. Kornienko