An Algorithm for Solving a Quartic Diophantine Equation Satisfying Runge's Condition

N. N. Osipov, S. D. Dalinkevich
Siberian Federal University (Krasnoyarsk)
e-mail: nnosipov@gmail.com

Abstract

In this paper we suggest an implementation of elementary version of Runge's method for solving a family of diophantine equations of degree four. Moreover, the corresponding solving algorithm (in its optimized version) is implemented in the computer algebra system PARI/GP.

Keywords: *diophantine equations*, *Runge's method*, *computer algebra systems*.

Contents

Introduction **1**

1 Solving Algorithm **3**
1.1 The Equation $z^2 = P(x)$ 3
1.2 The Main Case 4

2 Estimates for Integer Solutions **10**

3 Concluding Remarks **13**

References **14**

Introduction

There is a wide class of diophantine equations in two variables

$$f(x, y) = 0 \tag{0.1}$$

for which one can propose an *effective* solving method (that provides explicit upper bounds for the size of integer solutions), the so-called *Runge's method*. A description of the standard version of Runge's method can be found in the well-known monographs [4] and [10] (for more detailed proof, see [3, Ch. 4]). The original version (see old Runge's paper [9] or a modern paper [12]) is more general, below we give main theoretical result (so-called *Runge's theorem*). Despite the fact that Runge's method has been known for more than 100 years, its implementation in *computer algebra systems* (CAS) is very limited. At the same time, there is a small number of publications (see [8, 11, 6] and, especially, [1]) which refer to algorithmic aspects of implementation of this method (at least for some special cases) in CAS.

Assume that the polynomial $f(x, y) \in \mathbb{Z}[x, y]$ is irreducible over $\mathbb{Q}$ and let $d_0 = \max\{m, n\}$ where $m = \deg_x f(x, y)$ and $n = \deg_y f(x, y)$. If $f(x, y)$ satisfies Runge's condition (see below), then the estimate

$$\max\{|x|, |y|\} < (2d_0)^{18d_0^7} h^{12d_0^6} \tag{0.2}$$

holds for all integer solutions (x, y) of the equation (0.1) (see [12]). As usually, h denotes the *height* of given polynomial. This general result shows that the trivial implementation (brute force in the mentioned bounds) makes no sense in terms of the time required even in the case of d_0 small enough.

Let

$$f(x,y) = \sum_{i=0}^{m} \sum_{j=0}^{n} a_{ij} x^i y^j \tag{0.3}$$

be an irreducible polynomial in $\mathbb{Z}[x, y]$.

Runge's theorem. Assume that the equation (0.1) has infinitely many solutions $(x, y) \in \mathbb{Z}^2$. Then each of the following conditions holds:

(a) $a_{in} = a_{mj} = 0$ for all $i > 0$ and $j > 0$,

(b) $a_{ij} = 0$ for all pairs (i, j) satisfying $ni + mj > mn$,

(c) the leading part

$$f_L(x,y) = \sum_{ni+mj=mn} a_{ij} x^i y^j$$

is a constant multiple of a power of an irreducible polynomial in $\mathbb{Z}[x, y]$,

(d) the algebraic function $y = \Psi(x)$ defined by (0.1) has only one class of conjugate Puiseux expansions.

We say that the polynomial (0.3) satisfies *Runge's condition*, if at least one of the conditions (a), (b), (c) or (d) does not hold. Runge's theorem can be reformulated in the following equivalent form: if $f(x, y)$ satisfies Runge's condition, then the equation (0.1) has a finite set of integer solutions. In the literature, the following simplified version of this theorem is widely known. Denote by $f_d(x, y)$ the *leading homogeneous part* of the polynomial (0.3), $d = \deg f(x, y)$.

Corollary. If $f_d(x, y)$ can be decomposed into a product of non-constant relatively prime polynomials in $\mathbb{Z}[x, y]$, then the equation (0.1) has a finite set of integer solutions.

Below, the condition of Corollary will be called the *standard Runge's condition*. Under standard Runge's condition, in the case $d = 3$, a realistic (practically working) solving algorithm was proposed in [6]. This algorithm is based on the *elementary version of Runge's method* for diophantine equations of degree $d \leqslant 4$ (see [5]). In the case $d = 4$ an algorithmic implementation of elementary version of Runge's method is obtained only in some particular cases (see [8] and more recent paper [7]). It is necessary to refer to the preprint [1] where it is proposed to avoid "the use of Puiseux series and algebraic coefficients" which leads to "bad" estimates (i.e., estimates of the type (0.2)) for integer solutions.

The elementary version of Runge's method for diophantine equations of small degree is based on a convenient parametrization (by means of a special integer parameter) which provides enumerating possible integer solutions. As a result, the resolution of diophantine equation can be reduced to solving finitely many equations in one variable (usually, of degree two) over the integers. This idea for algorithmic implementation of Runge's method was applied in [6, 7].

In our paper, we consider a family of diophantine equations (0.1) with the left hand side

$$f(x,y) = (a_1x + b_1)y^2 + (a_2x^2 + b_2x + c_2)y + Ax^4 + Bx^3 + Cx^2 + Dx + E. \tag{0.4}$$

By default we assume this polynomial to be irreducible in $\mathbb{Z}[x, y]$. In general case both coefficients a_1 and A are non-zero and Runge's method can be applied because the condition (a) of Runge's theorem is violated.

In Section 1, we propose solving algorithm in the main case $a_1 = 1$ and $b_1 = 0$ (i.e., for the equation (1.3), see below). This algorithm is inspired by Theorem 1.1. Technically, this algorithm differs from the similar algorithms introduced in [6, 7] since it requires to resolve a number of equations in one variable of degree three. This fact must be taken into account if we want to estimate correctly the complexity of an algorithm. Therefore, we introduce an additional parameter (the so-called weight coefficient) for correct estimation of computational complexity. The weight coefficient depends on the CAS in which we plan to implement our algorithm (PARI/GP, see [13]). Further, we optimize the proposed algorithm in the same way as in [6]. The final result is represented in Theorem 1.2. At the moment, we do not know any other implementations of algorithms for solving diophantine equations of the specified type.

In Section 2, we give a few examples of estimating of integer solutions to several diophantine equations of small degree. In the case $d = 4$, the used method does not allow to the "reasonable" estimates (i.e., estimates which are close to realistic) for integer solutions, therefore we do not give any general theorems (we refer to [6] where the reader can find relevant examples of such theorems).

In Section 3, we give some remarks on the obtained results. In particular, we consider different ways to construct solving algorithm for the equation (0.1) with $f(x, y)$ of the general form (0.4). Also, we discuss a further application of the elementary version of Runge's method for diophantine equations of degree four.

1 Solving Algorithm

We begin with the case $a_1 = 0$ which is trivial in certain sense. In this case we can improve the well-known solving algorithm (see, e.g., [8]).

1.1 The Equation $z^2 = P(x)$

In the case $a_1 = 0$ and $b_1 \neq 0$, the equation (0.1) with the polynomial (0.4) can be reduced to the equation

$$z^2 = P(x) \tag{1.1}$$

with the polynomial $P(x) \in \mathbb{Z}[x]$ which satisfies $\deg P(x) \leqslant 4$. Runge's method works for the equation (1.1) in the case when $\deg P(x) = 4$ and the leading coefficient of $P(x)$ is a perfect square in $\mathbb{Z}$ (here, we can assume, without loss of generality, that $P(x)$ is monic). Otherwise, we need to refer to more complicated methods (see, for instance, [10]; of course, with the exception of the case $\deg P(x) \leqslant 2$ which is well studied).

We now consider the equation (1.1) with the polynomial

$$P(x) = x^4 + ax^3 + bx^2 + cx + d.$$

A well-known algorithm for solving (1.1) with this $P(x)$ was described in [8]. Below, we refer to this algorithm as the *standard algorithm* (or *method*). Here, we propose the following alternative approach. First, we reduce the equation (1.1) to a certain cubic diophantine equation. Next, we resolve the corresponding cubic equation using the technique from [6]. Sometimes, this approach is more effective than the standard method (for details, see Section 3). We demonstrate this phenomenon in the following example.

Example 1.1. Consider the equation

$$z^2 = x^4 + 8Hx^3 - 12x^2 + 4, \tag{1.2}$$

where the coefficient $H \geqslant 1$ is supposed to be rather large. Note that the equation (1.2) was firstly mentioned in the short note [2]. This equation has a solution $(x, z) \in \mathbb{Z}^2$ with

$$x = 4H^3 - 2H$$

that is quite large with respect to H. At the same time, it was proved (see ibid) that the upper bound for the integer solutions (x, z) of (1.1) is

$$|x| < 26h^3$$

where h is the height of $P(x)$. Thus, the equation (1.2) with $h = 8H$ has the biggest solution (up to a constant factor) with respect to the upper bound mentioned above. The direct computation shows that the standard solving algorithm (see [8]) needs $\approx 64H^3$ operations of taking square root for the integers with the maximal value $O(H^{12})$, and it is unexpected that the equation (1.2) can be solved faster.

Namely, for $P(x) = x^4 + 8Hx^3 - 12x^2 + 4$, we determine

$$R(x) = x^2 + 4Hx - 8H^2 - 6 = \sqrt{P(x)} + O\left(\frac{1}{x}\right), \quad x \to \infty.$$

Next, we introduce the new variable $w = z - R(x)$ and rewrite (1.2) in the form $F(w, x) = 0$ with the cubic polynomial

$$F(w, x) = (R(x) + w)^2 - P(x) =$$
$$= 2wx^2 + w^2 + 8Hwx + (-16H^2 - 12)w + (-64H^3 - 48H)x + 64H^4 + 96H^2 + 32.$$

Omitting technical details, we can formulate the final result as follows. The solving algorithm from [6] which is optimized for the equation $F(w, x) = 0$ "by hands" (i.e., analytically) requires only $\approx 96H^2$ operations of taking square root for the integers with the maximal value $O(H^6)$. This may appear surprising, especially because the height of $F(w, x)$ is much larger than the height of $P(x)$.

It is easy to see that the right hand side of (1.2) is a perfect square for

$$x \in \{0, H, 4H^3 - 2H\}.$$

Thus, the equation (1.2) has at least 6 integer solutions (x, y). In Table 1 we represent a certain statistical information on the number of additional (non-trivial) solutions for H taking values in the range $1 \leqslant H \leqslant 500$.

1.2 The Main Case

Suppose that $a_1 \neq 0$. In the case $A = 0$, we obtain a cubic diophantine equation with the leading homogenous part $x(Bx^2 + a_2xy + a_1y^2)$ satisfied the standard Runge's condition. Thus, we can use the algorithmic implementation of elementary version of Runge's method proposed in [6]. Therefore, we can suppose that $A \neq 0$.

$\#(x,z)$	$\#H$
0	393
2	85
4	20
6	1
8	1

Table 1: Distribution of the number of non-trivial solutions of the equation (1.2) in the range $1 \leqslant H \leqslant 500$.

For simplicity, here we consider in details only the particular case $a_1 = 1$, $b_1 = 0$ (the general case will be discussed briefly in Section 3). Then, the equation can be written as

$$xy^2 + (ax^2 + bx + c)y + Ax^4 + Bx^3 + Cx^2 + Dx + E = 0 \tag{1.3}$$

(we use simplified notation for convenience). Also, we can suppose that $c \neq 0$ (otherwise, the possible integer values of x must be in the set of divisors of E which can be found). Assuming $x \neq 0$, consider the number

$$l = \frac{cy + E}{x}.$$

Clearly, the value of l must be integer for all the solutions $(x, y) \in \mathbb{Z}^2$ of the equation (1.3) with $x \neq 0$. Dividing by x, we obtain

$$y^2 + (ax + b)y + Ax^3 + Bx^2 + Cx + D + l = 0.$$

This equality implies the congruence

$$y^2 + by + D + l \equiv 0 \pmod{x}$$

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

$$c^2(y^2 + by + D) \equiv c^2D + E^2 - bcE \pmod{cy + E}$$

(here we mean the congruence in the polynomial ring $\mathbb{Z}[y]$). Taking into account that

$$cy + E \equiv 0 \pmod{x},$$

we arrive at another congruence

$$c^2l + c^2D + E^2 - bcE \equiv 0 \pmod{x}$$

(both congruences are in the ring $\mathbb{Z}$). Finally, we set

$$k = \frac{c^2l + c^2D + E^2 - bcE}{x} = \frac{c^3y + (c^2D + E^2 - bcE)x + c^2E}{x^2}.$$

If (x, y) is an arbitrary integer solution of the equation (1.3) then the value of k must be integer as well as the value of l. Thus, we obtain the following result.

Theorem 1.1. Let $(x, y) \in \mathbb{Z}^2$ be a solution of the equation (1.3) with $x \neq 0$. Then, the number

$$k = \frac{c^3y + (c^2D + E^2 - bcE)x + c^2E}{x^2} \tag{1.4}$$

is integer.

One can propose the following straightforward and shorter proof of Theorem 1.1 which can be obtained by computer algebra methods (i.e., using symbolic computations in a computer algebra system). Using the equation (1.3), we find the expression for the coefficient E:

$$E = -xy^2 - (ax^2 + bx + c)y - Ax^4 - Bx^3 - Cx^2 - Dx.$$

Next, we plug it into the right hand side of (1.4). After dividing the numerator of the fraction in (1.4) by x^2, we obtain the explicit (but rather large) expression for k as a polynomial in the ring $\mathbb{Z}[x, y]$. Hence, the value of k must be integer. In order to illustrate this method, consider the equation

$$xy^2 + (x^2 + 1)y + x^4 + 1 = 0$$

with the polynomial $f(x, y) = xy^2 + (x^2 + 1)y + x^4 + 1$. We want to prove that the number

$$k = \frac{y + x + 1}{x^2}$$

is integer for each solution $(x, y) \in \mathbb{Z}^2$ with $x \neq 0$. Indeed, using the method described above we obtain

$$\frac{y + x + 1}{x^2} = xy^4 + (2x^2 + 2)y^3 + (2x^4 + x^3 + 2x)y^2 + (2x^5 + 2x^3 - 1)y + x^7 - x^2$$

which can be viewed as an equality in the residue class ring of $\mathbb{Z}[x, y]$ modulo $f(x, y)$. Note that this representation can be simplified:

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

Our further reasoning is based on the following idea. It is easy to check that both explicit real solutions $y = \Psi_i(x)$ $(i = 1, 2)$ of the equation (1.3) admit the estimate

$$\Psi_i(x) = O(|x|^{3/2}), \quad x \to \infty.$$

Hence, we have

$$\frac{c^3\Psi_i(x) + (c^2D + E^2 - bcE)x + c^2E}{x^2} \to 0, \quad x \to \infty.$$

As a corollary, for any $m \geqslant 1$, there exists a number $Q = Q(m) > 0$ such that

$$\left|\frac{c^3\Psi_i(x) + (c^2D + E^2 - bcE)x + c^2E}{x^2}\right| < Q(m)$$

for any x satisfying $|x| > m$ (of course, here we can use only those values of x for which $\Psi_i(x)$ are defined). Using this assertion, we can propose the following algorithm for solving the equation (1.3) over the integers.

Solving algorithm.

1. Choose $m \geqslant 1$ and compute the number $Q(m)$.
2. For all integers x satisfying $|x| \leqslant m$, solve the equation (1.3) (as a quadratic equation in y) over the integers.
3. For all integers k with $|k| < Q(m)$, solve the system of equations

$$\begin{cases} xy^2 + (ax^2 + bx + c)y + Ax^4 + Bx^3 + Cx^2 + Dx + E = 0, \\ c^3y + (c^2D + E^2 - bcE)x + c^2E - kx^2 = 0 \end{cases} \tag{1.5}$$

over the integers.

Let us consider an example in order to illustrate the proposed method.

Example 1.2. We show in details how the equation

$$x^4 - x^2y - xy^2 - y^2 + 1 = 0$$

can be solved over the integers (the resolution of this equation is outlined in [5]). Substituting $x - 1$ for x, we get the equation

$$xy^2 + (x^2 - 2x + 1)y - x^4 + 4x^3 - 6x^2 + 4x - 2 = 0 \tag{1.6}$$

of the form (1.3). By Theorem 1.1, the number

$$k = \frac{y + 4x - 2}{x^2}$$

must be integer for any solution $(x, y) \in \mathbb{Z}^2$ with $x \neq 0$. Eliminating y, we obtain an explicit expression for k, namely:

$$k = \frac{7x^2 - 2x - 1 \pm \sqrt{4x^5 - 15x^4 + 20x^3 - 10x^2 + 4x + 1}}{2x^3}.$$

Thus, if x satisfies $|x| > m$ then we certainly get $|k| < Q(m)$ with

$$Q(m) = \frac{7m^2 + 2m + 1 + \sqrt{4m^5 + 15m^4 + 20m^3 + 10m^2 + 4m + 1}}{2m^3}.$$

Further, we can proceed in various ways.

1) Firstly, we can determine m_0 so that the number $Q(m_0)$ is close to 1 (which is due to the fact that $Q(m) \to 0$ as $m \to \infty$). This is reasonable since when $m = m_0$ we need to solve (mainly) only quadratic equations (1.3) in y over the integers. For example, taking $m_0 = 8$, we obtain $Q(m_0) < 1$. Thus, it is necessary to solve: (a) for $x \in \{0, \pm 1, \ldots, \pm 8\}$, the equation (1.6) and, (b) for $k = 0$, the system (1.5), namely

$$\begin{cases} xy^2 + (x^2 - 2x + 1)y - x^4 + 4x^3 - 6x^2 + 4x - 2 = 0, \\ y + 4x - 2 = 0. \end{cases}$$

It is easy to see that this system can be reduced to the (again) quadratic equation

$$x^2 - 16x + 12 = 0.$$

Finally, we obtain that all the solutions of the equation (1.6) are

$$(x, y) \in \{(0, 2), (1, -1), (1, 1)\}.$$

2) Secondly, we can find m^* such that the total number of equations needed to resolve happens to be minimal (possibly, close to being minimal) when $m = m^*$. For instance, we can take $m^* = 4$ which provides $Q(m^*) < 2$. This is somewhat better than using the previous tactics.

The first issue of the proposed method is the following: we need to determine the number $Q(m)$ as an explicit function of the so-called *control parameter* m. This can be overcome by Lemma 1.1 (see below). The second issue can be formulated as follows: how to choose the optimal value of m? More precisely, we want to minimize the *cost-function* of the form

$$\text{cost}\,(m) = 2m + 2qQ(m), \tag{1.7}$$

where the *weight coefficient* $q > 1$ can be determined by experiments in a given CAS (in our case, PARI/GP). Here, for q, we take the ratio of the complexity of resolution of algebraic system of the form (1.5) and the complexity of resolution of quadratic equations (in both cases over the integers).

Now, consider the system (1.5) in details. Eliminating y, we obtain just a cubic (with the exception of the case $k = 0$) equation with respect to x, namely

$$k^2x^3 + K_1x^2 + K_2x + K_3 = 0. \tag{1.8}$$

Here, the coefficients K_j given as follows:

$$\begin{aligned} K_1 &= (-2c^2D - 2E^2 + 2bcE + ac^3)k + c^6A, \\ K_2 &= c^2(-2E + bc)k + c^6B + c^4D^2 + 2c^2DE^2 - 2bc^3DE - ac^5D + \\ &\quad + E^4 - 2bcE^3 - ac^3E^2 + b^2c^2E^2 + abc^4E, \\ K_3 &= c^4k + c^6C + 2c^4DE - bc^5D + 2c^2E^3 - 3bc^3E^2 + c^4(b^2 - ac)E. \end{aligned} \tag{1.9}$$

Therefore, we need to determine how much harder is the problem of solving cubic equations over the integers compared to that for quadratic equations. In PARI/GP, we intend to solve both problems via the function `nfroots` which provides, in particular, finding all rational roots of a univariate polynomial with integer coefficients. Preliminary computer experiments with the quadratic and cubic polynomials of moderate height (up to 10^{20}) have shown that, for this purpose, one can take $q = 2$. In Section 3, we discuss the method of choosing q in details.

Note that, although we can use the value $m = m_0$ with $Q(m_0)$ close to 1 (the motivation for this can be found in Example 1.2) in the algorithm, this can be disadvantageous due to the fact that m_0 may happen to be too large.

Example 1.3. Consider the equation

$$xy^2 + (x^2 + 1)y + x^4 + H = 0 \tag{1.10}$$

where the coefficient H is supposed to be rather large. The direct computation of $Q(m)$ based on Lemma 1.1 (see below) shows that the inequality

$$Q(m) > \frac{|H|^2}{m}$$

holds. Hence, if $Q(m_0) = 1$ then $m_0 > |H|^2$. On the other hand, taking $m^* = |H|$, we obtain $Q(m^*) \sim |H|$ as $H \to \infty$. Obviously, for the equation (1.10), the proposed algorithm with $m = m^*$ works faster than that with $m = m_0$.

For every H, the equation (1.10) has the trivial solution $(x, y) = (0, -H)$. A statistical information on the number of non-trivial solutions in the range $1 \leqslant H \leqslant 10^4$ is represented in Table 2.

For convenience purposes, let us introduce the notation:

$$\begin{aligned} Q_1 &= 2c^2D + 2E^2 - 2bcE - ac^3, & Q_6 &= -4C + 2ab, \\ Q_2 &= 2c^2E - bc^3, & Q_7 &= -4D + 2ac + b^2, \\ Q_3 &= -c^4, & Q_8 &= -4E + 2bc, \\ Q_4 &= -4A, & Q_9 &= c^2. \\ Q_5 &= -4B + a^2, \end{aligned} \tag{1.11}$$

The following technical result is necessary for an algorithmic implementation of the described method.

$\#(x,y)$	$\#H$
0	9200
1	639
2	133
3	26
4	1
5	1

Table 2: Distribution of the number of non-trivial solutions of the equation (1.10) in the range $1 \leqslant H \leqslant 10^4$.

Lemma 1.1. For any $m \geqslant 1$, the number $Q(m)$ can be defined as follows:

$$Q(m) = \frac{1}{2}\sum_{i=1}^{3}\frac{|Q_i|}{m^i} + \frac{|c|^3}{2}\left(\sum_{i=1}^{6}\frac{|Q_{i+3}|}{m^i}\right)^{1/2}, \tag{1.12}$$

where the coefficients $Q_1, \ldots, Q_9$ are given by (1.11).

Proof. The formulas (1.9) for the coefficients K_j show that the equation (1.8) is quadratic in k. Dividing by the leading coefficient x^3 and resolving with respect to k, we obtain

$$k = \frac{1}{2}\sum_{i=1}^{3}\frac{Q_i}{x^i} \pm \frac{c^3}{2}\left(\sum_{i=1}^{6}\frac{Q_{i+3}}{x^i}\right)^{1/2}.$$

Obviously, the condition $|x| > m$ implies the required estimate $|k| < Q(m)$ with $Q(m)$ given by (1.12). □

Unfortunately, the analytic expression for $Q(m)$ provided by Lemma 1.1 is too complicated to minimize the cost-function (1.7) by means of symbolic methods. Therefore, we need to focus on the reasonable estimates for cost (m^*) where m^* is a such value of m that it delivers the global minimum of cost (m). Further, the proposed solving algorithm with $m = m^*$ will be called the *optimized algorithm*. Denote by H the height of the left hand side of the equation (1.3).

Theorem 1.2. For the optimized algorithm, the estimate

$$\text{cost}\,(m^*) \leqslant C_1|c|^{4/3}H \tag{1.13}$$

holds. Here $C_1 > 0$ is a constant which depends only on q.

Proof. Let $m_1 = 4|c|^{4/3}H$. Since

$$\text{cost}\,(m^*) \leqslant \text{cost}\,(m_1) = 2m_1 + 2qQ(m_1) = 8|c|^{4/3}H + 2qQ(m_1),$$

it is sufficient to estimate the number $Q(m_1)$. We can perform this in a straightforward manner (i.e., by estimating each of the fractions $|Q_i|/m^i$, $|Q_{i+3}|/m^i$ at $m = m_1$ in the right hand side of (1.12); also, we use the obvious inequality $\sqrt{\alpha_1 + \ldots + \alpha_n} \leqslant \sqrt{\alpha_1} + \ldots + \sqrt{\alpha_n}$). The extremal case is the following:

$$|c|^3\sqrt{\frac{|Q_4|}{m_1}} \leqslant |c|^3\sqrt{\frac{4H}{4|c|^{4/3}H}} = |c|^{7/3} = |c|^{4/3}\cdot|c| \leqslant |c|^{4/3}H.$$

As a result, we arrive at the inequality $Q(m_1) \leqslant 2|c|^{4/3}H$. Thus,

$$\operatorname{cost}(m_1) \leqslant (8+4q)|c|^{4/3}H,$$

and we can set $C_1 = 8+4q$. □

The estimate (1.13) of complexity of the optimized algorithm in some cases occurs to be accurate (of course, up to a constant factor). For example, this is true for the equation (1.10) because $m^* \asymp m_1 \asymp H$ and $\operatorname{cost}(m^*) \asymp H$ as $H \to \infty$. On the other hand, it happens that sometimes the general estimate (1.13) can be improved.

Example 1.4. For the equation

$$xy^2 + (Hx^2+1)y + x^4 + 1 = 0 \tag{1.14}$$

we have $m^* \asymp |H|^{1/2}$ and, consequently, $\operatorname{cost}(m^*) \asymp |H|^{1/2}$ as $H \to \infty$. Using the optimized algorithm, we can check that for $1 \leqslant H \leqslant 10^5$ the equation (1.14) has no solutions $(x,y) \neq (0,-1)$, with the exception of $H=2$ and $H=8$ (see Example 2.5 below).

In general, the minimization of the cost-function (1.7) can be performed by a numerical method (for instance, we can use the well-known *golden-section search*). The starting (and, probably, rough) approximation $m^* \approx m_1$ proposed in the proof of Theorem 1.2 can be used as follows. Let us introduce $m_2 = tm_1$ where a constant factor $t > 1$ will be determined later. Earlier, we showed that the inequality $Q(m_1) \leqslant m_1/2$ holds. Hence, we have

$$\begin{gathered}\operatorname{cost}(m_2) = 2tm_1 + 2qQ(m_2) > 2tm_1 = 2m_1 + 2(t-1)m_1 \geqslant \\ \geqslant 2m_1 + 4(t-1)Q(m_1) \geqslant 2m_1 + 2qQ(m_1) = \operatorname{cost}(m_1)\end{gathered}$$

whenever $4(t-1) \geqslant 2q$. Therefore, setting $t = q/2+1$, we localize m^* in the interval $[1, m_2]$. It remains to apply a numerical search algorithm in the given interval. Heuristically, this additional procedure of optimization has a small (negligible) contribution to the total computational complexity.

2 Estimates for Integer Solutions

In this section, we give a few examples of explicit bounds for integer solutions of diophantine equations of small degree satisfying Runge's condition. Usually, these bounds are supposed to be used in order to find the solutions themselves, but the method (based on the elementary version of Runge's method) provides some estimates for solutions as an additional result (for more information, see [6]).

We start with three examples of cubic diophantine equations in order to demonstrate that the result entirely depends on the specifics of an equation.

Let H be a positive integer, C_2, C_3, etc. denote some positive absolute constants.

Example 2.1. For all the solutions (x,y) of the equation

$$x(y^2 - x^2) = Hy + 1 \tag{2.1}$$

in positive integers, we have the estimate

$$y \leqslant (H+3)/2$$

(the elementary proof can be obtained via the technique proposed in [6]). The upper bound is achieved for any odd H since the pair $(x,y) = ((H+1)/2, (H+3)/3)$ satisfies (2.1).

Example 2.2. For all the solutions (x, y) of the equation

$$x(y^2 - x^2) = Hy \tag{2.2}$$

in positive integers, we can propose the estimate

$$y \leqslant (H+1)^{3/4}$$

(the proof is also elementary, yet it requires some effort). The upper bound is achieved for infinitely many H since the pair $(x, y) = ((H+1)^{1/4}, (H+1)^{3/4})$ satisfies (2.2). This improves the expected estimate $y < C_2 H$ (see Exercise 4.15 [3]).

Example 2.3. For all the solutions (x, y) of the equation

$$x(y^2 - 2x^2) = Hy$$

in integers, the estimate

$$|x| < C_3 H^{3/2}$$

holds (for details, see [6]). There are no proved results on the accuracy of this estimate (apparently, it is achieved for infinitely many H).

For diophantine equations of degree four, the problem of estimating of integer solutions is much harder. In the case of (1.3), we can hope to obtain an estimate for integer solutions (x, y) by rewriting the auxiliary equation (1.8) as

$$1 + \frac{K_1}{k^2 x} + \frac{K_2}{k^2 x^2} + \frac{K_3}{k^2 x^3} = 0$$

and showing that $|x|$ cannot be too large. However, this method leads to quite rough estimates which are overvalued (not achieved in reality). In order to illustrate this fact, we consider the following three examples.

Example 2.4. For integer solutions (x, y) of the equation (1.10), we have the estimate

$$|x| < C_4 H^2$$

which can be obtained by the above-mentioned technique. Using the optimized algorithm, we can see that this estimate is unrealistic for $1 \leqslant H \leqslant 10^4$. On the other hand, for $H = t^3 + t^2$, the pair $(x, y) = (-t^2 - t, -t^3 - t^2)$ satisfies (1.10) and for this solution we have $|x| \sim H^{2/3}$ as $H \to \infty$. The hypothetical estimate

$$|x| < C_5 H^{2/3}$$

for non-trivial integer solutions $(x, y) \neq (0, -H)$ is confirmed by computer experiments. This estimate seems more realistic, but it is not clear how to prove it.

Example 2.5. Similarly, for integer solutions (x, y) of the equation (1.14) we can give the estimate

$$|x| < C_5 H.$$

At the same time, computer experiments (see Example 1.4) suggest the following conjecture: the equation (1.14) has integer solutions $(x, y) \neq (0, -1)$ if and only if $H \in \{2, 8\}$.

This conjecture is actually true, and we now outline the proof. Rewrite the equation (1.14) in the form

$$H = -\frac{xy^2 + y + x^4 + 1}{x^2 y}.$$

From this, one can conclude that the number

$$l = \frac{y + x^4 + 1}{xy}$$

must be in $\mathbb{Z}$. The last equality can be rewritten as

$$y = \frac{x^4 + 1}{lx - 1}.$$

Since $y \in \mathbb{Z}$, the number

$$d = \frac{x^2 + l^2}{lx - 1} \tag{2.3}$$

is also in $\mathbb{Z}$. Next, eliminating l, we get the equation

$$y^2 - (dx^2 - 2)y + x^4 + 1 = 0$$

which implies

$$y = \frac{dx^2 - 2 \pm xz}{2}, \quad z = \sqrt{(d^2 - 4)x^2 - 4d} \geqslant 0.$$

Since $x \neq 0$, it follows that $z \in \mathbb{Z}$. Finally, eliminating y, we obtain

$$2H = -d(x + 1) \mp z + \frac{2 \pm z}{x}.$$

Since $H \in \mathbb{Z}$, we have $2 \pm z \equiv 0 \pmod{x}$ that yields

$$4d + 4 \equiv 0 \pmod{x}. \tag{2.4}$$

It remains to prove that the congruence (2.4) and the condition $z \in \mathbb{Z}$ can be simultaneously held for finitely many pairs (x, d) at most. Thus, there are only finitely many possible values of H. More precisely, in the case of an arbitrary integer H, we conclude that

$$H \in \{-14, -9, -5, -4, -2, 0, 2, 8\}.$$

Using *Pell's equations*, we can somewhat simplify the proof. Namely, we can use the well-known result: if a triple (x, l, d) of integers satisfies (2.3) then $d = 5$ or $d = -t$ where t is a perfect square.

Example 2.6. For integer solutions (x, y) of the equation

$$xy^2 + (Hx + 1)y + x^4 + 1 = 0 \tag{2.5}$$

we have the same rough estimate as in Example 2.5. However, the equation (2.5) unlike the equation (1.14) is solvable for infinitely many H. For instance, the triple

$$x = \pm\sqrt{t}(t^2 - 1), \quad y = -t^4 + t^2 - 1, \quad H = t^4 - t^2 + 1 \pm \sqrt{t}(t^3 - 2t)$$

satisfies (2.5) and $|x| \sim H^{5/8}$ as $H \to \infty$.

Note that the equation (2.5) can be studied in the same way as the equation (1.14). The final description of the set of all integer solutions (x, y, H) use the *Chebyshev polynomials* of the second kind.

The last two examples may look artificial, but they vividly illustrate that, in general, obtaining exact bounds for integer solutions can be very difficult.

3 Concluding Remarks

In conclusion, we comment on some obtained results and discuss further applications of the elementary version of Runge's method.

In view of Example 1.1, it is worth discussing a strategy for solving the equation (1.1). The following seems to be reasonable. If the height of $P(x)$ is determined by the coefficient of x^3 (i.e., the other coefficients are small compared to it) then it is recommended to reduce given equation to the corresponding cubic equation (similarly to the case of the equation (1.2)). Otherwise, we recommend to use the standard method since this trick does not give a significant advantage (at least, the case of one-parametric equations of the type (1.2) confirms this).

Now, let's get back to the general case. Given the polynomial (0.4), we can use the linear substitution $a_1x + b_1 \to x$ that reduces the problem to solving the equation (1.3). However, this may lead to a significant increase in the height of the polynomial $f(x, y)$ as well as in the case of cubic diophantine equations (see [6]).

It seems that a more successful way is to generalize the already available solving algorithm for the equation (1.3) (we mean such generalization that is based on the direct analogue of Theorem 1.1). The expected estimate for complexity of the generalized algorithm (which is similar to the estimate (1.13), see Theorem 1.2) will be worse than that in the case of $a_1 = 1$, $b_1 = 0$.

For optimization of solving algorithm we need to choose the weight coefficient q correctly. Now, we describe how to do this in the case when H (the height of the left hand side of (1.3)) is moderate enough (up to 10^5) and $|c| \ll H$. Let $\widetilde{H}$ be the height of the left hand side of (1.8). Due to (1.9) and Theorem 1.2, we can assume $\widetilde{H} \approx H^4$ to be moderate (up to 10^{20}). Then,

$$q = \frac{\texttt{time}\,(\text{cubic}, \widetilde{H}, M)}{\texttt{time}\,(\text{quadratic}, H, M)},$$

where $\texttt{time}\,(\cdot)$ is the running time for solving $M = 10^6$ randomly chosen equations of the given type. For $H = 10^5$ (and $\widetilde{H} = 10^{20}$, respectively), using the function `nfroots` for finding rational roots in PARI/GP CAS, we obtain $q \approx 2$. However, in the case $c \asymp H$, we have $\widetilde{H} \gg H$, so that we recommend to increase q up to 6. In this case, the running time of the optimized algorithm will be reasonable for H up to at least 10^2.

Clearly, the results of computer experiments represented in Tables 1 and 2 should be developed further. At the moment, the running time for obtaining Table 2 is $t_1 \approx 13.5$ min and the similar table for the range $1 \leqslant H \leqslant 10^5$ requires $t_2 \approx 100t_1$ min (by using the processor `AMD Ryzen 7 2700x 3.7 GHs` and `16gb RAM`). Obviously, the running time can be decreased by implementing a parallel version of the proposed algorithm. Namely, the procedure of finding integer roots of a collection of univariate polynomials can be distributed between CPU threads that allows to use computer resources more efficiently, since PARI/GP CAS supports parallel programming.

It seems that the elementary version of Runge's method for $d = 4$ proposed in [5] can be implemented in the same way—at least for the polynomial $f(x, y)$ with the leading homogenous part of the form

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

We expect considerably more technical aspects in such an implementation. In particular, the corresponding auxiliary equation (as an analog of (1.8)) will be more complicated, although we hope that this is not crucial.

References

[1] *Beukers F., Tengely Sz.* An implementation of Runge's method for diophantine equations // arXiv:math/0512418 [math.NT]

[2] *Masser D.W.* Polynomial Bounds for Diophantine Equations // American Mathematical Monthly. 1986. Vol. 93. P. 486 — 488.

[3] *Masser D.W.* Auxiliary Polynomials in Number Theory. Cambridge University Press, 2016.

[4] *Mordell L.J.* Diophantine equations, London, Academic Press Inc., 1969.

[5] *Osipov N.N.* Runge's method for the equations of fourth degree: an elementary approach // Matematicheskoe Prosveshchenie, Ser. 3. Vol. 19. Moscow: MCCME, 2015. P. 178 — 198. (In Russian)

[6] *Osipov N.N., Gulnova B.V.* An algorithmic implementation of Runge's method for cubic diophantine equations // J. Sib. Fed. Univ. Math. Phys. 2018. Vol. 11(2). P. 137 — 147.

[7] *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. Vol. 12(3). P. 331 — 341.

[8] *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. P. 32 — 36.

[9] *Runge C.* Ueber ganzzahlige Lösungen von Gleichungen zwischen zwei Veränderlichen // J. reine und angew. Math. 1887. Vol. 100. P. 425 — 435.

[10] *Sprindžuk V.G.* Classical Diophantine Equations. New York: Springer-Verlag, 1993.

[11] *Tengely Sz.* On the Diophantine equation $F(x) = G(y)$ // Acta Arithmetica. 2003. Vol. 110. P. 185 — 200.

[12] *Walsh P.G.* A quantitative version of Runge's theorem on diophantine equations // Acta Arithmetica. 1992. Vol. 62. P. 157 — 172.

[13] https://pari.math.u-bordeaux.fr