Annales Mathematicae et Informaticae
***40** (2012) pp. 125–134*
http://ami.ektf.hu

Parallel algorithm for determining the "small solutions" of Thue equations*

Gergő Szekrényesi

University of Miskolc, Department of Applied Mathematics
gergosz5@gmail.com

Submitted July 17, 2011 — Accepted March 2, 2012

Abstract

A typical research field in number theory is determining the solutions of Diophantine equations. One of the earliest topic amongst these are the topics of Thue equations and inequalities. These equations are bivariate homogenous forms, with integer coeffictients.

The aim of the article is to create a parallel program, that can solve arbitrary Thue equations in finite time, and also gives good running times with the classical families. Naturally with the computational packages such as Maple, Magma or Kant a Thue equation can be solved, but in practice they can't handle problems given with relatively large coefficients. The parallel version of the algorithm provides an alternative to this problem.

Another application of the parallel program lies in determining the solutions of simultaneos Pellian equations. According to the work of L. Szalay the solutions of the system of equations can be traced back to the solutions of Thue equations, which can be found with the method given.

Keywords: Thue equations, parallel algorithm, simultaneous Pellian equations

MSC: 11Y40; 11D75

*This research was carried out as part of the TAMOP-4.2.1.B-10/2/KONV-2010-0001 project with support by the European Union, co-financed by the European Social Fund.

1. Basic terms

1.1. Thue equations

Let $F(x, y)$ be a bivariate, homogenous, over $\mathbb{Q}[x, y]$ irreducible polinomial with integer coefficients of at least third degree. Explicitly:

$$F(x, y) = a_n x^n + a_{n-1} x^{(n-1)} y + \ldots + a_0 y^n \in \mathbb{Z}[x, y]$$
$$n \geq 3 \quad \text{and} \quad m \in \mathbb{Z}, m \neq 0.$$

Then the

$$F(x, y) = m \tag{1.1}$$

equation is called Thue equation, of which's $x, y \in \mathbb{Z}$ are sought. Thue [10] proved that, (1.1) has only finitely many solutions for all m. Later, Baker [1] showed, that there exists an effectively computable constant based only on m and the coefficients of F which serves as an upper bound for the solutions.

Thue equations can be classified into parametric families. The first parametric family was introduced by Thue himself. In 1990 Thomas was the first to investigate families of fixed degrees, specifically a family of degree three. Furthermore several authors investigated certain families E.g. A. Pethő, Gaál, Lettl, Heuberger, Togbé and Ziegler.

There are a number of algorithms which provide the solutions of Thue equations. Unfortunately Baker's bound is too large, and does not make it possible to handle the problem by simple enumeration. For a similar problem Baker and Davenport created a method to reduce this bound drastically. A. Pethő and Schulenberg used a method of continued fraction reduction, and later the LLL-algorithm.

Among these algorithms a prominent one is the method described by A. Pethő, which provides the "small solutions" of such equations. An advantage of this algorithm is that it can be relatively easily implemented (it is based on the continued fraction algorithm). Furthermore it can benefit from many parallelization techniques. The most important definitions and theorems like the approximation theorems and their uses regarding the solutions of Thue equations are part of the article.

A method to find all the solutions of (1.1) for $m = 1$ and arbitrary n also exists. Finally Bilu and Hanrot [3] gave a much more efficient continued fraction method to reduce the bound, and they were able to solve equations up to the degree of 1000.

1.2. Simultaneous Pell equations

Simultaneous Pell equations are defined as follows:

$$a_1 x^2 + b_1 y^2 = c_1, \tag{1.2}$$

$$a_2x^2 + b_2z^2 = c_2, \tag{1.3}$$

where the x, y and z are nonnegative integers, and the coefficients satisfy the following conditions:

$$a_1b_1 < 0, \quad a_2b_2 < 0, \quad c_1c_2 \neq 0, \quad a_1c_2 - a_2c_1 \neq 0.$$

There is a method due to L. Szalay [9] with which one can trace back the solutions of such equations to the solving finitely many Thue equations.

1.3. Balancing numbers

Definition 1.1. An n positive integer is called a balancing number, if

$$1 + 2 + \cdots + (n-1) = (n+1) + (n+2) + \cdots + (n+r)$$

holds, for a suitable positive integer r.

Liptai published some results [5, 6] on special kinds of balancing numbers. For example a balancing number is called a Fibonacci balancing number if it is a Fibonacci number as well. He also proved that such balancing numbers are solutions of the equation

$$x^2 - 5y^2 = \pm 4.$$

If we are to find such common numbers in the Fibonacci sequence and the balancing numbers we have to provide an additional equation regarding the current balancing number, and solve the system. Another special type of balancing numbers are (a, b)-type balancing numbers.

Definition 1.2. Let $a, b \in \mathbb{N}$. The $an + b$ natural number is called an (a, b)-type balancing number if the following equation holds for a suitable $r \in \mathbb{N}$:

$$(a+b) + (2a+b) + \cdots + (a(n-1)+b) = (a(n+1)+b) + \cdots + (a(n+r)+b)$$

Kovács, Liptai and Olajos published about (a, b)-type balancing numbers, see [4]. Consider the following simultaneous Pellian equation system:

$$x^2 - 5y^2 = 4 \tag{1.4}$$

$$8x^2 - z^2 = 167. \tag{1.5}$$

The solutions of this system provide the common elements of the Fibonacci series and a (a, b)-type balancing numbers.

2. Theory of Thue equations

2.1. Continued Fraction Algorithm

The continued fractions play an important role in finding the solutions of Thue equations and inequalities.

Input: $\alpha \in \mathbb{R}, \alpha \neq 0, A \in \mathbb{Z}$
Output: $a_0 \in \mathbb{Z}, a_1, \ldots a_n \in \mathbb{N}, p_n, q_n \in \mathbb{Z}$
1. $i \leftarrow 0,\ \alpha_0 \leftarrow \alpha,\ p_{-2} \leftarrow 0,\ p_{-1} \leftarrow 1,\ q_{-2} \leftarrow 1,\ q_{-1} \leftarrow 0$
2. DO
3. $\quad a_i \leftarrow \lfloor \alpha_i \rfloor$
4. $\quad p_i \leftarrow a_i p_{i-1} + p_{i-2}$
5. $\quad q_i \leftarrow a_i q_{i-1} + q_{i-2}$
6. $\quad$ IF $\alpha_i - a_i = 0$ THEN STOP
7. $\quad i \leftarrow i + 1$
8. $\quad \alpha_i \leftarrow 1/(\alpha_{i-1} - a_{i-1})$
9. WHILE $q_i \leq A$

The algorithm provides the continued fraction expansion of the α real number, the values a_i, and the convergents, the $\frac{p_i}{q_i}$. It is easy to see, that the algorithm stops at the 6th step if and only if α is rational. Furthermore, for the convergents the following inequality holds for every $n \geq 0$:

$$\frac{1}{(a_{n+1}+2)q_n^2} \leq \left|\alpha - \frac{p_n}{q_n}\right| < \frac{1}{q_n^2} \tag{2.1}$$

If α is irrational, then (2.1) implies $\lim_{n\to\infty} \frac{p_n}{q_n} = \alpha$.

The right side of (2.1) characterizes the continued fractions. The following lemma is due to Legendre

Lemma 2.1. (Legendre) *Let $\alpha \in \mathbb{R}$ and $x, y \in \mathbb{Z}, y \neq 0$. If*

$$\left|\alpha - \frac{x}{y}\right| < \frac{1}{2y^2}$$

holds, then $\dfrac{x}{y}$ is a convergent to α.

2.2. Fast algorithm for finding the "small solutions"

Consider the following Thue inequality

$$|F(x,y)| \leq m. \tag{2.2}$$

The aim is to find the solutions of (2.2) where $|y| < C$ and $(x,y) = 1$ (coprime integers).

Remark 2.2. Using the coprime solutions it is easy to determine all the other solutions:
Let $|F(x,y)| \leq m$ and $(x,y) = d : x = x_0 d, y = y_0 d, (x_0, y_0) = 1$. The method computes (x_0, y_0) for which $|F(x_0, y_0)| \leq m$ and $|F(x_0 d, y_0 d)| \leq m \Rightarrow d^n |F(x_0, y_0)| \leq m \Rightarrow d^n \leq \dfrac{m}{|F(x_0, y_0)|}$, thus the possible values of d can be checked individually.

Let α be the root of $F(x,1) = 0$. Its conjugates: $\alpha^{(1)}, \ldots, \alpha^{(n)}$. To find the solutions we need to calculate some constants and bounds from the roots of $F(x,1) = 0$, the $\alpha^{(i)}$ numbers. These are as follows:

$$c_1 = \frac{2|m|^{\frac{1}{n}}}{|\alpha^{(j)} - \alpha^{(i)}|},$$
$$c_2 = \frac{|m|2^{n-1}}{\prod\limits_{j \neq i} |\alpha^{(j)} - \alpha^{(i)}|}$$

If $\alpha^{(i)} = a + bI$ is a complex number, then

$$c_3 = \left(\frac{c_2}{|b|}\right)^{\frac{1}{n}}.$$

Furthermore

$$c_4 = (2c_2)^{\frac{1}{n-2}}.$$

Lemma 2.3. *(Thue) The Thue equation $F(x,y) = m$ has only finitely many $(x,y) \in \mathbb{Z}^2$ solutions.*

Remark 2.4. This lemma does not give any method on how to find these solutions. Experience shows, that Thue equations usually only have few number and small in absolute value solutions.

The following lemma gives described by A. Pethő provides a method to find the solutions of a given Thue equation or inequality, up to a certain prescribed bound.

Lemma 2.5. *(Pethő) For all (x,y) solutions of the* (2.2) *Thue-inequality either $|y| < \max(c_1, c_3, c_4)$ or $\frac{x}{y}$ is a convergent in the continued fraction expansion of one of the real conjugates of the α roots of $F(x,1) = 0$, for which*

$$|y| < (c_2(A+2))^{\frac{1}{n-2}}, \text{ where}$$

$A = \max\limits_{j \leq j_0} a_j$, *where j_0 is the index for which the denominator of the convergent of $\alpha^{(i)}$ $q_{j_0} < C$, $q_{j_0+1} > C$ holds.*

The main point of the algorithm is, that if an $(x,y) \in \mathbb{Z}^2$ pair is a solution of (2.2) then $\frac{x}{y}$ is a good approximation of the roots of $F(x,1)$. We calculate the "medium large" solutions with the simple continued fraction expansion of the roots of $F(x,1)$, then we find the smaller solutions by elementary bounds and equations.

Remark 2.6.

- The constant C is is taken to be considerably large, for example 10^{500}.
- The lemma provides the $|y| < C$ solutions fast in this case as well, but does not give any proof that no other solutions exist in the range $|y| > C$.

- In the case of $C = 10^{500}$ the $\alpha^{(i)}$ number must be at least 1000 digits precise for the continued fraction algorithm to work accurately.
- The value of A is generally small, as the continued fraction digits are small.
- The algorithm provies a way to reduce the C upper bound, by substituting it with $(c_2(A+2))^{\frac{1}{n-2}}$, and reapplying the calculation. This can be done as long as it yields new values for the upper bound, or until we get a value below $\max(c_1, c_3, c_4)$.

3. Main result: the Algorithm

3.1. Steps of the Algorithm

1. Calculate the α_i roots of the polinomial.
2. For all roots perform the followings (parallelly):
   (a) Calculate the constants c_1, c_2, c_3, c_4.
   (b) For only the real α_i-s calculate the continued fraction expansion and the convergents, using the above described algorithm, and reduce the upper bound as much as possible.
   (c) Up to the newly calculated upper bound subsitute the convergents p_k, q_k to see if they satisfy the $|F(p_k, q_k)| \leq m$ inequality.
   (d) If yes, add (p_k, q_k) to the set of solutions.
   (e) In the interval $|y| < \max(c_1, c_3, c_4)$ find the solutions using an exhaustive search.

Remark 3.1. The algorithm can be used to solve equations as well besides inequalites. In this case replace the $\leq$ operator with equality. The case is similar with the absolute value too.

3.2. Parallelization

The implementation of the above described algorithm is rather inefficient in terms of time, especially in the case of equations with large coefficients. Based on the steps described above, the algorithm can be parallellized.

After calculating the roots of $F(x, 1) = 0$, the α_i numbers, it is easy to see that the operations that must be performed with the current α_i root are independent and such a dedicated process can be started for every α_i root. This way the continued fraction algorithm can be run parallelly, reducing the running time of the algorithm. This gives the best speed-up if we have n processors for the n different roots.

For each of the roots the exhaustive search must be performed in the interval $|y| < \max(c_1, c_3, c_4)$. On the other hand if we calculate this interval for every root,

and then calculate their maximum we get the interval where we must perform the search regardless of which root are we processing. Furthermore this search interval can be divided to smaller parts as well, exactly as many parts as many processes we have. By using this method not only we speed-up the algorithm, but eliminate some otherwise redundant calculation.

It is obvious that the data parallellism of the algorithm is beneficial from many aspects: not only the time to find the solutions is reduced, but the strain on the individual processes is much less than in the sequential case. Furthermore the practical implementation of the algorithm can be done relatively easily, as the stages are totally independent and synchronization is only needed at the end of the algorithm, to collect the solutions found by the individual processes.

4. Simultaneous Pell equations

Consider again the system given with the equations (1.2) and (1.3):

$$a_1x^2 + b_1y^2 = c_1,$$
$$a_2x^2 + b_2z^2 = c_2,$$

satisfiying the natural conditions $a_1b_1 < 0$, $a_2b_2 < 0$, $c_1c_2 \neq 0$, $a_1c_2 - a_2c_1 \neq 0$. The algorithm to solve such equations can be derived from a combination of (1.2) and (1.3), which leads to the solution of Thue equations of degree four. These types of equations can be solved for instance with the method described above. Unfortunately the number of Thue equations needed to solve increases as the coefficients, the $a_i, b_i, c_i (i = 1, 2)$ numbers get larger.

Introduce the following equation by combining (1.2) and (1.3):

$$a_3x^2 + b_3y^2 + c_3y^2 = 0, \tag{4.1}$$

where a_3, b_3 and c_3 are non-zero integers. For this equation the following lemma holds.

Lemma 4.1. *Assume that* (x_0, y_0, z_0) *is a solution of* (4.1) *with* $z_0 \neq 0$. *Then every integer* (x, y, z) $z \neq 0$ *solution of* (4.1) *can be parametrized in the following way:*

$$x = \pm\frac{D}{d}(-ax_0s^2 - 2by_0rs + bx_0r^2),$$
$$y = \pm\frac{D}{d}(-ay_0s^2 - 2ax_0rs + by_0r^2),$$
$$z = \pm\frac{D}{d}(-az_0s^2 + bz_0r^2),$$

where r *and* $s > 0$ *are coprime integers,* D *is a non negative integer and* $d \mid 2a^2bcz_0^3$ *is a non negative integer.*

Using this lemma we gain the following equation from the simultaneous Pell equation system:

$$a_1c_y^2(\alpha_{i_1}s^2+\beta_{i_1}sr+\gamma_{i_1}r^2)^2+b_1c_x^2(\alpha_{i_2}s^2+\beta_{i_2}sr+\gamma_{i_2}r^2)^2=c_1c_x^2c_y^2\Big(\frac{d}{D}\Big)^2. \quad (4.2)$$

Expanding this equation we gain Thue equations of degree four. The number of Thue equations we need to solve is exactly the number of divisors of the right hand side of the equation. The solutions of these Thue equations are used to determine the solutions of the Pellian system, using the given parametrization.

Furthermore, this step of the algorithm can be optimized. Only one Thue inequality is needed to be solved instead of a series of Thue equations, as the coefficients of the equations remain the same, only the right hand changes. By this, the right hand side of the inequality has to be chosen so it corresponds to the greatest of the original Thue equations. After calculating the solutions of the inequality the solutions have to be tested to filter out the solutions that provide different right sides than the original ones.

5. Running times

To demonstrate the program we have tested it on several examples. To compare the results we ran the same examples using Maple 12. The examples were run with the following configuration: Intel Pentium 4 3.2GHz dual core processor, 1GB RAM. The speed-up is defined as the quotient of the parallel running time and the sequential running time.

Comparison:

	Maple 12	Own Program	Speed-up
$\lvert f(x,y)\rvert \le 200$	1671 ms	34 ms	49.14
$\lvert g(x,y)\rvert \le 27$	843 ms	82 ms	10.28

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

$$g(x,y)=x^6+20000x^5y-50015x^4y^2- \\ -20x^3y^3+50000x^2y^4+ \\ +20006xy^5+y^6.$$

Considering another case, the $\left|x^{19}+2y^{19}\right|=2$ Thue equation was solved with both a program written in C programming language using the PARI/GP computational package and the program created in this essay. The results are the following:

	PARI/GP	Own Program	Speed-up
$\left\lvert x^{19}+2y^{19}\right\rvert =2$	11.7 sec	63.3 sec	0.18

The difference is due to the program using the PARI/GP package uses the method of Bilu and Hanrot, thus calculated the set of fundamental units of the algebraic number field, and got better running times, but the order of magnitude is the same.

We tested the program with the following simultaneous Pell equation system as well.

$$x^2 - 5y^2 = \pm 4,$$
$$z^2 - 8x^2 = 1.$$

From this system we gain the

$$\left|s^4 - 30s^3r + 195s^2r^2 - 150sr^3 + 25r^4\right| \leq 96100$$

Thue inequality. The running time of Maple 12 was 573.921 seconds, while our program solved the task in 395 milliseconds, which means a speed-up of about 1453. The program in L. Szalay's article which was written in MAGMA [7] solved the same inequality in about 4 seconds.

5.1. Further Pellian system examples

The following table shows some other simultaneous Pellian equation systems that we have tested and compared with the program created by L. Szalay.

System	Own Program	MAGMA	Speed-up
$-11x^2 + y^2 = 1$ $-56x^2 + z^2 = 1$	104ms	26sec	250
$-7x^2 + y^2 = 2$ $-32x^2 + z^2 = -23$	21sec	40sec	1.9
$-8x^2 + y^2 = 1$ $-2x^2 + 3z^2 = 1$	1.5sec	5sec	3.3
$-5x^2 + y^2 = -20$ $-2x^2 + z^2 = 1$	381ms	3sec	7.8
$x^2 - 2y^2 = -1$ $x^2 - 10z^2 = -9$	113ms	1sec	8.8

The table shows that the program proved to be faster in these cases, however the difference is not always significant.

Consider now again the system belonging to (a, b)-type balancing numbers:

$$x^2 - 5y^2 = 4$$
$$8x^2 - z^2 = 167.$$

From this system the generated Thue inequality is:

$$|27889s^4 + 37408s^3r + 9046s^2r^2 - 972sr^3 + 9r^4| \leq 1.008 \cdot 10^{16}.$$

The Thue inequality has 58 solutions and those provide the sole solution of the original system: (7, 3, 15). The running time in this case was approximately 6 minutes, whereas the MAGMA was unable to solve the system because the right hand side of the inequality causes an overflow in the system.

References

[1] Baker, A., Transcendental Number Theory, *Cambridge*, 1990.

[2] Baker, A., Davenport, H., The equations $3x^2 - 2 = y^2$ and $8x^2 - 7 = z^2$, *Quart. J. Math. Oxford*, 20 (1969), 129–137.

[3] Bilu, Y. and Hanrot, G., Solving Thue equations of high degree, *J. Number Theory*, 60 (1996), 373–390.

[4] Kovács, T., Liptai, K. and Olajos, P. On (a, b) balancing numbers, *Publ. Math. Debrecen*, 77, No. 3-4. (2010) 485–498.

[5] Liptai, K. Fibonacci balancing numbers, *Fibonacci Quart.*, 42 ,No. 4, (2004) 330–340.

[6] Liptai, K. Lucas balancing numbers, *Acta Math. Univ. Ostrav.*, 14, No. 1, (2006) 43–47.

[7] MAGMA Computational Algebra System, Computational Algebra Group School of Mathematics and Statistics, *University of Sydney*, NSW 2006, Australia. `http://magma.maths.usyd.edu.au/magma/`

[8] Pethő, A., Schulenberg, R., Effektives Lösen von Thue Gleichungen, *Publ. Math.* Debrecen (1987)

[9] Szalay, L., On the resolution of simultaneous Pell equations, *Annales Mathematicae et Informaticae*, 34 (2007) 77–87.

[10] Thue, A., Über Annäherungenswerte algrebraischen Zahlen, *J. Reine Angew. Math.*, 135 (1909) 284–305.