Math-Net.Ru

Общероссийский математический портал

К. D. Zhukov, Задача о приближенном общем делителе и цепные дроби, *Матем. вопр. криптогр.*, 2016, том 7, выпуск 2, 61–70

DOI: <https://doi.org/10.4213/mvk184>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 54.172.240.79

26 апреля 2023 г., 13:11:56

Approximate common divisor problem and continued fractions

K. D. Zhukov

TVP Laboratories, Moscow

Получено 03.III.2015

Abstract. We describe two algorithms for computing common divisors of two integers, when one of these integers is known only approximately. We generalize a known method based on the continued fraction technique. In some cases new algorithms overcome the best known algorithm based on Coppersmith's method: not so accurate approximation is required to compute a divisor.

Key words: approximate common divisors, continued fractions, Diophantine approximation

Задача о приближенном общем делителе и цепные дроби

К. Д. Жуков

Лаборатории ТВП, Москва

Аннотация. Описаны два алгоритма вычисления общего делителя двух чисел, одно из которых известно лишь приближенно. Обобщается известный метод, использующий цепные дроби. В некоторых случаях новые алгоритмы превосходят наилучший известный алгоритм, основанный на методе Коперсмита: для вычисления делителя достаточно меньшая точность.

Ключевые слова: приближенные общие делители, цепные дроби, диофантовы аппроксимации

1. Introduction

Let us describe a partially approximate common divisor problem (PACDP). Consider two integers N_1 and N_2 . Let A be a nontrivial common divisor of N_1 and $N_2 - \Delta$ for some small integer Δ . The goal is to find A provided that it is large.

The PACDP was introduced in 2001 by N. Howgrave-Graham [3] who studied continued fraction technique and Coppersmith's method.

An algorithm of S. Sarkar and S. Maitra [4] for solving the PACDP is known to be the best. Based on Coppersmith's method this algorithm finds a common divisor in time which is polynomial of $n = \max\{\lfloor \ln N_1 \rfloor, \lfloor \ln N_2 \rfloor\}$ provided that $l_A > l_{N_1/A} + l_\Delta - l_{N_1/A}^2/l_{N_1}$, where l_A , $l_{N_1/A}$, l_{N_1} and l_Δ are the binary lengths of A , N_1/A , N_1 and Δ respectively. In this paper we describe two algorithms for computing the common divisor and show that their time complexities grow polynomially with n and c provided that $l_A \geq l_{N_1/A} + l_\Delta + 2 - \log_2 c$.

2. Definitions and notation

Let us sketch the main definitions from [1] related to continued fractions. An expression of the form

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \dots}}, \quad (1)$$

where a_0 is integer, and a_i is positive integer for $i = 1, 2, \dots$, is called an infinite continued fraction.

As a shorthand for expression (1), a continued fraction is often expressed as $\langle a_0, a_1, a_2, \dots \rangle$. For $i = 0, 1, \dots$ we define the i -th convergent of continued fraction (1) to be a finite continued fraction

$$\frac{P_i}{Q_i} = \langle a_0, a_1, a_2, \dots, a_i \rangle. \quad (2)$$

For $i = 2, 3, \dots$ there are recurrent formulae for numerators and denominators of convergents:

$$P_i = P_{i-1}a_i + P_{i-2}, \quad Q_i = Q_{i-1}a_i + Q_{i-2} \quad (3)$$

with the initial condition

$$P_0 = a_0, \quad Q_0 = 1, \quad P_1 = a_0a_1 + 1, \quad Q_1 = a_1. \quad (4)$$

It is easy to show that for $i = 0, 1, 2, \dots$ the following equality holds

$$P_{i+1}Q_i - P_iQ_{i+1} = (-1)^i. \quad (5)$$

Any real number α has a continued fraction representation $\alpha = \langle a_0, a_1, a_2, \dots \rangle$. If $\alpha = P/Q$ is a rational then the continued fraction expansion of α has a finite number of elements: $\alpha = \langle a_0, a_1, a_2, \dots, a_r \rangle$, $r \in \mathbb{N} \cup \{0\}$. In that case we can compute elements of the continued fraction with the Euclidean algorithm for P and Q .

Convergents of α are good rational approximations to α . For any $i=0, 1, 2, \dots$ the following estimates hold

$$\frac{1}{Q_i(Q_i + Q_{i+1})} < \left| \alpha - \frac{P_i}{Q_i} \right| < \frac{1}{Q_i Q_{i+1}}. \quad (6)$$

On the other hand the Legendre theorem states that if a rational is a good approximation to some real number, than that rational equals some convergent of this real number.

Theorem 1 ([1], Theorem 245). *Let α be a real number, a and b be nonzero coprime integers ($b > 0$), such that $\left| \alpha - \frac{a}{b} \right| < \frac{1}{2b^2}$. Then $\frac{a}{b}$ is a convergent of α .*

This result has the following generalization.

Theorem 2 ([2], Theorem 1). *Let α be a real number, a and b be nonzero coprime integers ($b > 0$), such that $\left| \alpha - \frac{a}{b} \right| < \frac{c}{b^2}$, where c is a real positive number. Then there exists a pair of adjacent convergents $\frac{P_k}{Q_k}$ and $\frac{P_{k+1}}{Q_{k+1}}$ of α such that the equality*

$$\frac{a}{b} = \frac{uP_{k+1} \pm vP_k}{uQ_{k+1} \pm vQ_k}$$

holds for some nonnegative integers u and v , $uv < 2c$.

3. Continued fraction methods for the PACDP

Suppose that the integers N_1 and $N_2 - \Delta$ have a nontrivial common divisor A for some integer Δ . Then the integers N_1 and N_2 have representations $N_1 = AB_1$ and $N_2 = AB_2 + \Delta$. Howgrave-Graham [3] showed how to find a divisor A using the continued fraction techniques under some bounds on B_1 and Δ . The method is based on a special case of the following lemma.

Lemma 1. *Let A, B_1, B_2 be natural numbers and Δ be an integer. Let $N_1 = AB_1$ and $N = AB_2 + \Delta$. Suppose that the inequality*

$$A > \sqrt{\frac{N_1 |\Delta|}{c}} \quad (7)$$

holds for some positive real number c . Then

$$\left| \frac{N_2}{N_1} - \frac{B_2}{B_1} \right| < \frac{c}{B_1^2}. \quad (8)$$

Algorithm 1 (PACDP).

Require: N_1 and N_2 — natural numbers; c — real parameter of the method.

Ensure: set $S \subset \mathbb{N} \times \mathbb{Z}$ of all pairs (A, Δ) such that $N_1 = AB_1$, $N_2 = AB_2 + \Delta$ and $A > \sqrt{N_1 |\Delta|/c}$.

```

1  $S \leftarrow \emptyset$ 
2 COMPUTE ALL CONVERGENTS  $\frac{P_0}{Q_0}, \dots, \frac{P_r}{Q_r}$  OF  $\frac{N_2}{N_1}$  USING THE EUCLIDEAN ALGORITHM
  AND FORMULAS (3), (4)
3 FOR ALL  $k = 0, 1, \dots, r-1$  DO
4   FOR ALL  $j=0, 1$  AND  $v=1, 2, \dots, [2c]$ ,  $u=0, 1, \dots, [2c/v]$  DO
5     IF  $uQ_{k+1} + (-1)^j vQ_k \mid N_1$  THEN
6        $B_1 \leftarrow uQ_{k+1} + (-1)^j vQ_k$ ,  $B_2 \leftarrow uP_{k+1} + (-1)^j vP_k$ ,  $A \leftarrow \frac{N_1}{B_1}$ ,  $\Delta \leftarrow N_2 - AB_2$ 
7        $S \leftarrow S \cup \{(A, \Delta)\}$ 
8 PRINT  $S$ 
```

The correctness of the algorithm follows from Lemma 1 and Theorem 2.

Proposition 1. *Let S be the set defined in the Algorithm 1. Suppose that $|S| = O(n^{\ln \ln n})$, where $n = \max\{\lfloor \ln N_1 \rfloor, \lfloor \ln N_2 \rfloor\}$. Then the complexity of Algorithm 1 is $O(cn^2 \ln c \ln n \ln \ln n)$ binary operations.*

Consider a special case of Algorithm 1 with fixed $c = \frac{1}{2}$. Then the number of iterations of loop 4–7 equals 1. All the candidates B_1 and B_2 are of the form Q_k and P_k respectively. Hence all the pairs $\{(A, \Delta)\}$ are distinct and we may rewrite step 7 as $S \leftarrow S \sqcup \{(A, \Delta)\}$. In particular, it follows that $|S| = O(n)$ if $c = \frac{1}{2}$.

The method described above with $c = \frac{1}{2}$ was introduced in [3].

Let us show how to modify Algorithm 1 to reduce the complexity by a factor $n = \max\{\lfloor \ln N_1 \rfloor, \lfloor \ln N_2 \rfloor\}$ under additional restriction.

While using Theorem 2 we should guess u and v for all pairs of adjacent convergents. Dujella [2] introduced an idea allowing to guess u and v only for a few pairs of convergents in a particular task. We generalize this idea basing on the following theorem.

Theorem 3. *Let α be a rational number, a and b be nonzero coprime integers ($b > 0$) such that $0 < |\alpha - a/b| < d \leq c/b^2$, where d and c are fixed positive real numbers. Let $\frac{P_0}{Q_0}, \dots, \frac{P_r}{Q_r} = \alpha$ be the convergents of α and $r \geq 3$.*

Suppose that $k \in \{0, 1, \dots, r-3\}$ is the largest integer satisfying inequalities $(\alpha - a/b)(-1)^k > 0$ and $|\alpha - P_k/Q_k| \geq d$. Then

$$\frac{a}{b} = \frac{uP_{k+1} + vP_k}{uQ_{k+1} + vQ_k} \text{ or } \frac{a}{b} = \frac{uP_{k+2} - vP_{k+1}}{uQ_{k+2} - vQ_{k+1}} \text{ or } \frac{a}{b} = \frac{uP_{k+3} + vP_{k+2}}{uQ_{k+3} + vQ_{k+2}},$$

for some nonnegative integers u and v , $uv < 2c$.

It should be mentioned that the interval from 0 to $r - 3$ may not contain a number k from the condition of Theorem 3.

Lemma 2. *Let A , B_1 , B_2 be natural numbers and Δ be an integer. Let $N_1 = AB_1$ and $N_2 = AB_2 + \Delta$. Suppose that inequalities*

$$A \geq D > \sqrt{\frac{N_1 |\Delta|}{c}} \quad (9)$$

hold for some positive real numbers c and D . Then

$$\left| \frac{N_2}{N_1} - \frac{B_2}{B_1} \right| < \frac{D^2 c}{N_1^2} \leq \frac{c}{B_1^2}. \quad (10)$$

Lemma 1 is a special case of Lemma 2.

Suppose that we have to find a divisor A larger than fixed known bound D in the PACDP with the inputs N_1 and N_2 . Then we could use the following

Algorithm 2 (PACDP).

Require: N_1 and N_2 – natural numbers; c , D – real parameters of the method.

Ensure: set $T \subset \mathbb{N} \times \mathbb{Z}$ of pairs (A, Δ) such that $N_1 = AB_1$, $N_2 = AB_2 + \Delta$ and $A \geq D > \sqrt{N_1 |\Delta|/c}$ and $A > 1$.

```

1   $T \leftarrow \emptyset$ 
2  COMPUTE ALL CONVERGENTS  $\frac{P_0}{Q_0}, \dots, \frac{P_r}{Q_r}$  OF  $\frac{N_2}{N_1}$  USING THE EUCLIDEAN ALGORITHM
   AND FORMULAS (3), (4)
3   $k_1 \leftarrow -1$ ,  $k_2 \leftarrow -1$ 
4  IF  $r \geq 3$  THEN
5    FOR ALL  $k=0, \dots, r-3$  DO
6      IF  $\left| \frac{N_2}{N_1} - \frac{P_{k+2}}{Q_{k+2}} \right| < \frac{D^2 c}{N_1^2} \leq \left| \frac{N_2}{N_1} - \frac{P_k}{Q_k} \right|$  THEN
7        IF  $k$  is even THEN
8           $k_1 \leftarrow k$ 
9        IF  $k$  is odd THEN
10          $k_2 \leftarrow k$ 
11  FOR  $k=k_1, k_2$  DO
12    IF  $k > -1$  THEN
13      FOR  $j=0, 1, 2$  AND  $v=1, 2, \dots, [2c]$ ,  $u=0, 1, \dots, [2c/v]$  DO
14        IF  $uQ_{k+j+1} + (-1)^j vQ_{k+j} \mid N_1$  THEN
15           $B_1 \leftarrow uQ_{k+j+1} + (-1)^j vQ_{k+j}$ ,  $B_2 \leftarrow uP_{k+j+1} + (-1)^j vP_{k+j}$ ,
             $A \leftarrow N_1/B_1$ ,  $\Delta \leftarrow N_2 - AB_2$ 
16           $T \leftarrow T \cup \{(A, \Delta)\}$ 
17  PRINT  $T$ 
```

The correctness of the algorithm follows from Lemma 2 and Theorem 3. Note that there is no guarantee that all the divisors $A > 1$ such that condition (9) holds belong to the set T . The interval from 0 to $r - 3$ may not contain a number k such that $\left| \frac{N_2}{N_1} - \frac{P_{k+2}}{Q_{k+2}} \right| < \frac{D^2 c}{N_1^2} \leq \left| \frac{N_2}{N_1} - \frac{P_k}{Q_k} \right|$. Nevertheless, in all our numerical experiments such numbers k were always lying in the specified interval.

Proposition 2. *Let T be the set defined in the Algorithm 2. Suppose that $|T| = O(n^{\ln \ln n})$, where $n = \max\{\lceil \log_2 N_1 \rceil, \lceil \log_2 N_2 \rceil\}$. If $n = O(c \ln c)$, then the complexity of Algorithm 2 is $O(cn \ln c \ln n \ln \ln n)$ binary operations.*

Using m parallel cores (without common RAM) we may gain speed-up for Algorithms 1 and 2 by a factor of m by means of partitioning the set of all pairs (u, v) we are guessing into m subsets of equal sizes.

Let us describe the applicability condition for Algorithms 1 and 2 in terms of the binary representation lengths of the numbers A , B_1 and Δ .

Proposition 3. *Let $N = AB$ be a composite integer. Let l_A , l_B and l_Δ be the lengths of the binary representations of A , B and Δ respectively. Suppose that the condition*

$$l_A \geq l_B + l_\Delta + 2 - \log_2 c \quad (11)$$

holds for some real c . Then

$$A \geq 2^{l_A-1} > \sqrt{\frac{N\Delta}{c}}. \quad (12)$$

It follows from Proposition 3 that given integers N_1 and N_2 we may compute A and Δ such that $N_1 = AB_1$, $N_2 = AB_2 + \Delta$ and condition (11) is fulfilled for some real c , using Algorithm with input parameter c . If additionally l_A (the binary representation length of A) is known, then we may use Algorithm 2 with input parameters c and $D = 2^{l_A-1}$.

4. Comparison of the proposed algorithms with the method of S. Sarkar and S. Maitra

Let us compare our algorithms with the best algorithm for the PACDP. The algorithm of S. Sarkar and S. Maitra based on the Coppersmith's method computes a common divisor provided that $\beta < 1 - 3\alpha + \alpha^2$, where $\alpha \approx \log_{N_1} B_1$, $\beta \approx \log_{N_1} \Delta - \log_{N_1} B_1$. Using inequality $N_1^\beta < N_1^{1-3\alpha+\alpha^2}$, we see that the method of S. Sarkar and S. Maitra may be applied to the PACDP if

$$l_A > l_{B_1} + l_\Delta - \frac{l_{B_1}^2}{l_{N_1}}, \quad (13)$$

where l_A , l_{B_1} , l_{N_1} and l_Δ are the lengths of the binary representations of A , B_1 , N_1 and Δ respectively.

The comparison of inequalities (11) and (13) shows that the method based on the continued fraction techniques is better if the following inequality holds:

$$\log_2 c - 3 > \frac{l_{B_1}^2}{l_{N_1}}. \quad (14)$$

Under the condition (14) our algorithms may be applied to cases with larger values of Δ . For practical values of parameter c see Section 5.

5. Experiments

We have implemented Algorithm 2 using MPIR library for big numbers [5]. We used the Microsoft Visual C++ compiler (64-bit). We ran our program on a single core of an Intel Core i7 processor (3.33 GHz).

In the Table below we denote by l_A the length of the binary representation of A , by l_Δ the length of Δ , by l_B the lengths of B_1 and B_2 , and by c the parameter of the method such that condition (11) holds.

Table. Program implementation of Algorithm 2

l_A	l_B	$\log_2 c$	l_Δ	Time, sec
2795	277	27	2543	1156
		30	2546	10250
3819	277	21	3561	17
		30	3570	12804

The values in the first and the third lines of Table are on the boundary of applicability of the method of S. Sarkar and S. Maitra. The values in the second and the fourth lines do not satisfy condition of applicability of their method. In each case the time of computing a common divisor with our implementation is shown in the last column of the Table.

References

- [1] Buhshtab A. A., *Number Theory*: Moscow: Prosvechenie, 1966 (in Russian).
- [2] Dujella A., “Continued fractions and RSA with small secret exponent”, *Tatra Mt. Math. Publ.*, **29** (2004), 101–112.
- [3] Howgrave-Graham N., “Approximate integer common divisors”. In: “*Cryptography and Lattices*”, CaLC 2001, Lect. Notes Comput. Sci., **2146**, 2001, 51–66.
- [4] Sarkar S., Maitra S., “Approximate integer common divisor problem relates to implicit factorization”, *IEEE Trans. Inf. Theory*, **IT-57**:6 (2011), 4002–4013.
- [5] MPIR: *Multiple Precision Integers and Rationals*, <http://www.mpir.org/>.

Appendices

A. Proof of Lemma 1

Proof. Let us square the both sides of inequality (7), then divide both sides by A . We get $A > \frac{B_1|\Delta|}{c}$. It is clear now that

$$\frac{1}{A} < \frac{c}{B_1|\Delta|}. \quad (15)$$

Let us write down the ratio of N_2 and N_1 :

$$\frac{N_2}{N_1} = \frac{AB_2 + \Delta}{AB_1} = \frac{B_2}{B_1} + \frac{\Delta}{N_1}. \quad (16)$$

Hence the following equality holds:

$$\left| \frac{N_2}{N_1} - \frac{B_2}{B_1} \right| = \frac{|\Delta|}{AB_1}. \quad (17)$$

Notice that combining (15) and (17) we obtain the statement of the lemma. $\square$

B. Proof of Proposition 1

Proof. The complexity of step 2 (the Euclidean algorithm) is not larger than $O(n^2 \ln n \ln \ln n)$ binary operations. The complexities of step 5 and step 6 do not exceed $O(n \ln n \ln \ln n)$ binary operations. We can insert new pair in the set S with $O(n \ln |S|) = O(n \ln n \ln \ln n)$ binary operations. The total complexity of steps 5–7 is $O(n \ln n \ln \ln n)$ binary operations.

Now we estimate the number of iterations of steps 5–7. The number of iterations of loop 3–7 equals the total number of convergents $r = O(n)$. The number of iterations of loop 4–7 is $O(c \ln c)$.

Therefore, the algorithm iterates steps 5–7 $O(cn \ln c)$ times. Hence, the total complexity of the algorithm is $O(cn^2 \ln c \ln n \ln \ln n)$ binary operations. $\square$

C. Proof of Theorem 3

Proof. For any $i=0, \dots, r-1$ and $j=0, 1$ the determinant of the system of linear equations

$$\begin{cases} a = xP_{i+1} + y(-1)^j P_i, \\ b = xQ_{i+1} + y(-1)^j Q_i, \end{cases} \quad (18)$$

equals $(-1)^{i+j}$ according to (5). The pair $(aQ_i - bP_i)(-1)^i, (bP_{i+1} - aQ_{i+1})(-1)^{i+j}$ is a solution of the system (18).

The remaining proof is divided into two parts depending on relation between the fractions $\frac{P_{k+2}}{Q_{k+2}}$ and $\frac{a}{b}$.

Suppose that $\left| \alpha - \frac{P_{k+2}}{Q_{k+2}} \right| < \left| \alpha - \frac{a}{b} \right|$. Consider the triple

$$\begin{aligned} u &= (aQ_k - bP_k)(-1)^k, \\ v &= (bP_{k+1} - aQ_{k+1})(-1)^k = (aQ_{k+1} - bP_{k+1})(-1)^{k+1}, \\ w &= (bP_{(k+1)+1} - aQ_{(k+1)+1})(-1)^{(k+1)+1}. \end{aligned}$$

Obviously the pairs (u, v) and (v, w) are solutions of the system (18) for $i = k, j = 0$ and for $i = k+1, j = 1$ respectively. We claim that the coefficients u and v are positive.

Suppose that the difference $\alpha - \frac{a}{b}$ is negative; then k is odd and the following inequalities hold: $\frac{P_{k+1}}{Q_{k+1}} < \frac{P_{k+2}}{Q_{k+2}} < \frac{a}{b} < \frac{P_k}{Q_k}$. As $b > 0$ and $Q_k > 0$, the last inequality in the chain above is equivalent to $aQ_k - bP_k < 0$. Hence u is positive and so are v and w .

Suppose that the difference $\alpha - \frac{a}{b}$ is positive; then k is even and the following inequalities hold: $\frac{P_k}{Q_k} < \frac{a}{b} < \frac{P_{k+2}}{Q_{k+2}} < \frac{P_{k+1}}{Q_{k+1}}$. In this case coefficients u, v and w are also positive.

Our goal is to show that at least one of inequalities $uv < 2c$ and $vw < 2c$ holds. Let t and s be some positive real numbers determined by the equalities $\left| \alpha - \frac{P_k}{Q_k} \right| = \frac{tc}{b^2}$ and $\left| \alpha - \frac{P_{k+1}}{Q_{k+1}} \right| = \frac{sc}{b^2}$. Notice that the following estimates hold: $u = bQ_k \left| \frac{P_k}{Q_k} - \frac{a}{b} \right| < bQ_k \left| \alpha - \frac{P_k}{Q_k} \right| = \frac{tcQ_k}{b}$, $v = bQ_{k+1} \left| \frac{P_{k+1}}{Q_{k+1}} - \frac{a}{b} \right| = bQ_{k+1} \left(\left| \frac{P_{k+1}}{Q_{k+1}} - \alpha \right| + \left| \alpha - \frac{a}{b} \right| \right) < \frac{(s+1)cQ_{k+1}}{b}$, $w = bQ_{k+2} \left| \frac{P_{k+2}}{Q_{k+2}} - \frac{a}{b} \right| < bQ_{k+2} \left| \alpha - \frac{a}{b} \right| < \frac{cQ_{k+2}}{b}$. Let us obtain upper estimates for $Q_k Q_{k+1}$ and $Q_{k+1} Q_{k+2}$. Firstly note that

$$\frac{1}{Q_k Q_{k+1}} = \left| \frac{P_{k+1}}{Q_{k+1}} - \frac{P_k}{Q_k} \right| > \left| \alpha - \frac{P_k}{Q_k} \right| = \frac{tc}{b^2}.$$

Analogously we have $\frac{1}{Q_{k+1} Q_{k+2}} > \frac{sc}{b^2}$. Now we may upper bound the products uv and vw : $uv < \frac{Q_k Q_{k+1} t(s+1)c^2}{b^2} < (s+1)c$, $vw < \frac{Q_{k+1} Q_{k+2} (s+1)c^2}{b^2} < \frac{(s+1)c}{s} = \left(1 + \frac{1}{s}\right)c$. Hence, if $s \leq 1$, then $uv < 2c$, and if $s > 1$, then $vw < 2c$.

Suppose now that $\left| \alpha - \frac{P_{k+2}}{Q_{k+2}} \right| \geq \left| \alpha - \frac{a}{b} \right|$. Obviously the pair

$$\begin{aligned} u' &= (aQ_{k+2} - bP_{k+2})(-1)^{k+2}, \\ v' &= (bP_{k+3} - aQ_{k+3})(-1)^{k+2}, \end{aligned}$$

is a solution of the system (18) for $i = k+2$ and $j = 0$. By analogy with previous reasoning the coefficient u' is nonnegative and the coefficient v' is positive. Let us obtain upper estimates for coefficients u' and v' :

$$\begin{aligned} u' &= bQ_{k+2} \left| \frac{P_{k+2}}{Q_{k+2}} - \frac{a}{b} \right| < bQ_{k+2} \left| \frac{P_{k+2}}{Q_{k+2}} - \alpha \right| < bQ_{k+2} \frac{1}{Q_{k+2} Q_{k+3}} = \frac{b}{Q_{k+3}}, \\ v' &= bQ_{k+3} \left| \frac{a}{b} - \frac{P_{k+3}}{Q_{k+3}} \right| \leq bQ_{k+3} \left| \frac{P_{k+2}}{Q_{k+2}} - \frac{P_{k+3}}{Q_{k+3}} \right| = \frac{b}{Q_{k+2}}. \end{aligned}$$

Now we obtain upper estimate for the value $(Q_{k+2}Q_{k+3})^{-1}$:

$$\frac{1}{Q_{k+2}Q_{k+3}} = \left| \frac{P_{k+2}}{Q_{k+2}} - \frac{P_{k+3}}{Q_{k+3}} \right| = \left| \frac{P_{k+2}}{Q_{k+2}} - \alpha \right| + \left| \alpha - \frac{P_{k+3}}{Q_{k+3}} \right| < 2 \left| \frac{P_{k+2}}{Q_{k+2}} - \alpha \right| < \frac{2c}{b^2}.$$

So, the product of u' and v' is majorized by

$$u'v' < b^2/Q_{k+2}Q_{k+3} < 2c. \quad \square$$

D. Proof of Lemma 2

Proof. Let us square the both sides of the right inequality from (9), then multiply both sides by c and divide by N_1^2 . We obtain inequality $D^2c/N_1^2 > |\Delta|/N_1$. Using the inequality $A \geq D$ we get

$$\frac{|\Delta|}{N_1} < \frac{D^2c}{N_1^2} \leq \frac{A^2c}{N_1^2} = \frac{c}{B_1^2}. \quad (19)$$

Arguing as in the proof of Lemma 1, we see that

$$\left| \frac{N_2}{N_1} - \frac{B_2}{B_1} \right| = \frac{|\Delta|}{N_1}. \quad (20)$$

Notice that combining (19) and (20) we obtain the statement of the lemma. $\square$

E. Proof of Proposition 2

Proof. The complexity of step 2 (the Euclidean algorithm) is not larger than $O(n^2 \ln n \ln \ln n)$ binary operations. The complexity of loop 4–10 (choosing the right convergent) has the same estimate $O(n^2 \ln n \ln \ln n)$ binary operations.

Arguing as in the proof of Proposition 1 we find that the complexity of loop 11–16 (guessing u and v) is not larger than

$$O(c \ln c (n \ln |T| + n \ln n \ln \ln n)) = O(cn \ln c \ln n \ln \ln n)$$

binary operations. Using $n = O(c \ln c)$ we estimate the total complexity of the algorithm by $O(cn \ln c \ln n \ln \ln n)$ binary operations. $\square$

F. Proof of Proposition 3

Proof. The inequality $A \geq 2^{l_A-1}$ is obvious. We shall prove that $2^{l_A-1} > \sqrt{N\Delta}/c$.

It follows from the condition of proposition that

$$2^{l_A} \geq \frac{42^{l_B} 2^{l_\Delta}}{c} > \frac{4B\Delta}{c}.$$

Hence $2^{l_A-2} > N\Delta/(Ac)$.

Using the inequality $2^{l_A} > A$ we conclude that $2^{2l_A-2} > 2^{l_A-2}A > N\Delta/c$. Extracting the square root we obtain (12). $\square$