Turk. J. Math. Comput. Sci.
17(1)(2025) 33–46

DOI : 10.47000/tjmcs.1569163

Another Approach to Factoring by Continued Fractions

Turgut Hanoymak[1,*], Cihan Kayak[1]

[1]*Department of Mathematics, Faculty of Science, Van Yüzüncü Yıl University, 65080, Van, Turkey.*

Received: 17-10-2024 • Accepted: 24-02-2025

Abstract. The problem of prime factorization is particularly important in fields such as cryptography, where it plays a crucial role, especially in the security of public key cryptosystems like RSA Algorithm. There are numerous factorization algorithms that have been developed over time, each with varying levels of complexity. These algorithms have played a crucial role in fields like mathematics and cryptography, where prime factorization remains a key challenge. In this study, the continued fraction method, one of the factorization methods, is examined. To highlight the importance of the continued fraction factorization method, a brief mention is made of the vulnerability of RSA Algorithm to attacks, such as Weiner's attack, which exploits small private keys. Our approach aims to enhance the efficiency of factorization by integrating this method with relevant theorems by giving concrete examples with detailed tables.

2020 AMS Classification: 94A60, 11A55, 11J70, 11A51

Keywords: Factorization algorithms, continued fractions, RSA algorithm, cryptography.

1. Introduction

Mathematics and security have evolved in a deeply intertwined manner throughout history. Cryptography, situated at the intersection of these two fields, applies advanced mathematical techniques to secure data. One of the most renowned cryptographic methods is RSA Algorithm [13], developed in 1978 by Ronald Rivest, Adi Shamir, and Leonard Adleman. RSA Algorithm is based on a security model that leverages the computational difficulty of factoring large composite numbers into their prime components. This complexity is fundamental to the strength of RSA Algorithm and has positioned it as a cornerstone of modern cryptographic systems.

Various methods have been developed to factorize large numbers, one of which involves the use of continued fractions. Continued fractions represent irrational numbers as an infinite sequence of divisions and are particularly effective in factoring large composite numbers with significant prime factors. This technique is not only essential in number theory but also enhances the security framework of cryptographic algorithms like RSA Algorithm.

The mathematical origins of continued fractions date back to ancient Indian mathematicians such as Aryabhata and Brahmagupta in the 5th century BCE. However, their relevance to modern cryptography became pronounced with the advent of RSA Algorithm. The challenge of factoring large prime numbers underscores the role of continued fractions in strengthening the cryptographic model of RSA Algorithm.

*Corresponding Author
Email addresses: hturgut@yyu.edu.tr (T. Hanoymak), cihankayak1071@gmail.com (C. Kayak)

Among factorization algorithms, Fermat Factorization Algorithm [6] was developed around 1670 by the French mathematician Pierre de Fermat. Fermat proposed this algorithm as a method to find the prime factors of large composite numbers. The algorithm estimates the approximate square root of a given composite number and uses this approximation to identify its factors. It holds particular significance in fields such as cryptography and number theory. The Fermat Factorization Algorithm is based on a fundamental logic of searching for prime divisors to determine the factors of a number.

Euler's Factorization Algorithm [3], introduced by the Swiss mathematician Leonhard Euler in the mid-18th century, holds notable importance in number theory. Similarly, John Pollard's Rho Algorithm [9], developed in 1974, is a prominent method for factoring large composite numbers. Pollard's $p-1$ Algorithm [8], also introduced in 1974, is another key technique that contributes to evaluating cryptographic systems like RSA.

In 1984, Carl Pomerance introduced the Quadratic Sieve Algorithm [11], which proved to be highly effective for factoring large composite numbers. Building on this success, Pomerance later proposed the Number Field Sieve (NFS) Algorithm [12] in 1996. NFS has become the most efficient classical algorithm for factoring numbers exceeding 100 digits, solidifying its role in cryptography.

The $p+1$ Factoring Method [16] emerged in the mid-20th century alongside advancements in number theory and algorithmic cryptography. Interest in this method grew during the 1970s with the advent of computer technology. Researchers such as Richard Brent refined the technique in the 1980s, increasing its effectiveness for factoring large numbers. Today, the $p+1$ method is utilized in both theoretical research and practical applications, contributing to the development of more secure encryption systems.

In this study, an alternative method for factoring a number is proposed, based on certain well-known theorems related to continued fractions. This proposed method is considered to provide faster and more efficient results compared to classical method. The process of factoring selected composite numbers is analyzed step by step, algorithms for both methods are presented, and these methods are compared in detail with examples. Additionally, the results are visualized in tabular form. Furthermore, Wiener attack [15], [1], [7] on RSA Algorithm is briefly discussed in the context of continued fractions. The remainder of our article is structured as follows: In Section 2, we present some definitions and theorems based on continued fractions. In Section 3, we discuss the mathematical foundations of continued fractions. In Section 4, we examine the security vulnerabilities of RSA Algorithm in detail, particularly focusing on Wiener's attack and how continued fractions can serve as an effective method in such attacks. In Section 5, we thoroughly explore the classical continued fraction-based factorization algorithm, supported by step-by-step implementations and examples. In Section 6, we propose a faster and more efficient factorization approach as an alternative to the classical method, presented in detail with examples. In Section 7, we analyze the computational complexity of both methods, evaluating their efficiency and practical applicability. Finally, in Section 8, we summarize the findings of our study and provide suggestions for future research in the fields of cryptanalysis and post-quantum security.

Some definitions and theorems related to continued fractions that we utilize are provided below. The reader can see more information about continued fractions in [14].

2. Preliminaries

This section addresses fundamental concepts and key results related to the applications of continued fractions in number theory, particularly in factorization problems. A critical aspect of factoring large composite numbers lies in constraining their prime factors within specific upper bounds. For instance, the well-known result that any prime factor of a composite number n cannot exceed $\sqrt{n}$ significantly simplifies the search process for factors. Such structural properties, when combined with foundational tools like Euclidean Algorithm, enhance efficiency in both theoretical and practical applications.

In factorization algorithms, numbers with small prime factors play an important role. In this context, numbers whose prime factors are all bounded by a predefined parameter B (termed *B-smooth numbers*) directly influence algorithmic performance. This concept is central to techniques such as the factor base method and, when integrated with continued fractions, enables rapid factorization of large integers. Below, the definitions and theorems forming the foundation of this study are presented in detail.

Definition 2.1. A positive integer n is called *B-smooth* if all its prime factors are less than or equal to a given bound B. Let n be represented as follows:

$$n = p_1^{e_1} \cdot p_2^{e_2} \cdots p_k^{e_k}, \quad \text{where each prime factors } p_i \leq B$$

and B is a predefined upper limit (smoothness bound).

We begin with the following theorem, which plays a fundamental role in the rational approximations of irrational numbers and enhances the accuracy of factorization methods.

Theorem 2.2 (Dirichlet's Approximation Theorem (1842))**.** *For any irrational number α, there exist infinitely many distinct rational numbers $\frac{a}{b}$ with $b \geq 1$ such that*

$$\left|\alpha - \frac{a}{b}\right| < \frac{1}{b^2}.$$

We present the following theorem, which determines the best rational approximations of irrational numbers and serves as a crucial tool in factorization algorithms, particularly contributing to security analysis in cryptography, where small errors can lead to significant consequences.

Theorem 2.3 (Legendre's Theorem on Approximation)**.** *Let a and b be coprime integers ($\gcd(a, b) = 1$), with $b > 0$. For any irrational number α, there exists a rational number $\frac{a}{b}$ such that*

$$\left|\alpha - \frac{a}{b}\right| < \frac{1}{2b^2}.$$

In the study of number theory and integer factorization, the relation between quadratic residues and the greatest common divisor (gcd) is essential for finding non-trivial factors of a composite number n. The following remark emphasizes an important observation:

Remark 2.4. Let x, y be integers and $n \in \mathbb{Z}^+$. If $x^2 \equiv y^2 \pmod{n}$ but $x \not\equiv \pm y \pmod{n}$, then $\gcd(x-y, n)$ is a non-trivial factor of n. The random square methods attempt to find integers x and y at random so that $x^2 \equiv y^2 \pmod{n}$.

3. Contiuned Fractions Background

Continued fractions provide a systematic way to express irrational numbers as sequences of nested fractions, offering highly accurate approximations. Their unique recursive structure makes them particularly useful in number theory and integer factorization. In cryptographic applications, continued fractions play a key role in analyzing modular arithmetic relationships, especially in attacks that exploit small private keys in RSA Algorithm. Before presenting the formal definition, we introduce the fundamental notation and properties that will be used throughout this section.

A continued fraction representation of a number allows for efficient approximations, particularly in cases where traditional fraction expansions fail to provide sufficiently close estimates. This property is crucial in integer factorization methods that rely on rational approximations to deduce hidden numerical structures.

Let d be a non-square positive integer. We use continued fractions to find the positive integer solutions to the equations $x^2 - dy = \pm 1$. Continued fractions represent a mathematical technique utilized to explore the solutions of such equations through the application of the expression $\sqrt{d}$.

Definition 3.1. Let $k \geq 1$ and $a_k > 0$ be a sequence of integers given by $a_0, a_1, a_2, \ldots$.

$$[a_0; a_1, a_2, \ldots] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}}.$$

A sequence of the form $[a_0; a_1, a_2, \ldots]$ is called a simple infinite continued fraction, and the values a_n are referred to as partial quotients. If the limit of the continued fraction exists, it is said to be convergent.

The following theorems form the mathematical foundation of factorization methods that utilize continued fractions. These theorems are used to find rational approximations of irrational numbers and to factorize large numbers into their prime factors using these approximations.

Theorem 3.2 ([5]). *[Convergents of Continued Fractions and Recursive Relations] Given a sequence of positive integers $a_0, a_1, a_2, \ldots$, the sequences p_n and q_n are defined recursively as follows:*

$$p_{-2} = 0, \quad p_{-1} = 1, \quad p_n = a_n p_{n-1} + p_{n-2},$$

$$q_{-2} = 1, \quad q_{-1} = 0, \quad q_n = a_n q_{n-1} + q_{n-2}.$$

Then, the convergents of the continued fraction are given by

$$C_n = [a_1; a_2, \ldots, a_n] = \frac{p_n}{q_n}$$

which satisfies the recurrence relation:

$$\frac{p_n}{q_n} = \frac{a_n p_{n-1} + p_{n-2}}{a_n q_{n-1} + q_{n-2}}.$$

Moreover, for every p_n and q_n, the following inequality holds:

$$\left|\alpha - \frac{p_n}{q_n}\right| < \frac{1}{q_n^2}$$

which demonstrates that the convergents obtained from continued fractions provide the best rational approximations to the irrational number α.

This theorem defines the convergents of continued fractions and the recursive relationships of these convergents. It also demonstrates that continued fractions provide the best rational approximations to irrational numbers. It explains how the sequences p_n and q_n are recursively calculated and how these sequences converge to the irrational number.

Theorem 3.3 ([5]). *Let α be an irrational number. In this case, the continued fraction expansion of α is represented as follows:*

$$\alpha = \alpha_0,$$

$$a_k = \lfloor \alpha_k \rfloor,$$

$$\alpha_{k+1} = \frac{1}{\alpha_k - a_k}, \quad \textit{where} \quad (k = 0, 1, 2, 3, \ldots).$$

If α is defined in this manner, then its continued fraction representation is given by

$$\alpha = [a_0; a_1, a_2, \ldots]$$

which is unique.

This theorem, on the other hand, shows how the continued fraction expansion of an irrational number can be uniquely obtained. It explains how the coefficients (a_k) of continued fractions are calculated and how these coefficients form the fractional approximations of the irrational number. This process is a fundamental step in the computation of the values a_i and b_i used in the factorization algorithm.

These two theorems establish the mathematical basis for factorization methods that employ continued fractions and explain why these methods are effective. In particular, these theorems demonstrate how large numbers can be factored into their prime factors more quickly and efficiently. This method underlies attacks such as Wiener's attack, which threatens the security of cryptographic systems like RSA Algorithm.

4. Factoring RSA Moduli via Continued Fractions

The security of RSA Algorithm relies on the difficulty of factoring large integers. However, certain vulnerabilities can arise when specific conditions are met, particularly when the private key is too small. One such vulnerability is exploited by Wiener's attack, which uses the mathematical concept of continued fractions to break RSA scheme under specific circumstances. This section explores the theoretical foundation of Wiener's attack and how continued fractions can be used to efficiently recover the private key when certain parameters are improperly chosen.

4.1. **Theoretical Background.** [2]

In RSA Algorithm, the public key consists of a modulus $N = p \cdot q$ (where p and q are large prime numbers) and a public exponent e. The private key is represented by the private exponent d, which satisfies the congruence:

$$e \cdot d \equiv 1 \pmod{\phi(N)},$$

where $\phi(N)$ is Euler's totient function. For RSA Algorithm, $\phi(N)$ is computed as

$$\phi(N) = (p-1) \cdot (q-1).$$

Wiener's attack focuses on cases where the private key d is small relative to N. Specifically, if $d < \frac{1}{3}N^{1/4}$, the attack can efficiently recover d using the continued fraction expansion of $\frac{e}{N}$.

4.2. **Continued Fractions and Wiener's Attack.** Continued fractions provide a way to approximate real numbers using sequences of integers. For Wiener's attack, the continued fraction expansion of $\frac{e}{N}$ is used to find convergents $\frac{k}{d}$ that approximate $\frac{e}{N}$. These convergents are potential candidates for the private key d.

A method of attack that threatens the security of RSA aims to determine the value of $\phi(N)$. To achieve this, the following modular arithmetic expression is employed as

$$e \cdot d - k \cdot \phi(N) = 1, \quad \text{where} \quad k \in \mathbb{Z}.$$

When rearranged, this expression yields:

$$\frac{e}{\phi(N)} - \frac{k}{d} = \frac{1}{d \cdot \phi(N)}.$$

From this, it is derived:

$$\frac{e}{N} \approx \frac{k}{d}.$$

These ratios can be approximated using the method of continued fractions. Continued fractions aid in simplifying rational numbers to uncover the values of k and d.

Validation of d Value: The value of d in RSA scheme possesses certain characteristics. First, since $\phi(N)$ is an even number, d must be odd. If d is even, the next convergent should be considered. Furthermore, $\phi(N)$ must be an integer. This can be verified using the following expression:

$$\frac{e \cdot d - 1}{k}.$$

If this expression does not yield an integer, the next convergent should be evaluated.

Finding Roots: To identify the prime factors, a quadratic equation can be constructed

$$(x-p) \cdot (x-q) = 0.$$

Expanding these factors results in the equation:

$$x^2 - (p+q)x + p \cdot q = 0$$

This equation can be reformulated in relation to $\phi(N)$ as follows

$$x^2 - (N - \phi(N) + 1)x + N = 0.$$

If the value of $\phi(N)$ is a precise approximation, the roots of this equation (p, q) will be integers. These roots represent the prime factors of N and provide an opportunity to examine the security of RSA Algorithm.

5. Factorization Algorithm Using Continued Fractions

In this section, the well-known method is explained in detail, along with examples. One of the provided examples is factored using both this method and the proposed method.

Definition 5.1. Let $n > 1$, $n \in \mathbb{Z}$, and $\sqrt{n} \approx a$, where $a \in \mathbb{R}^+$ and $k = 0, 1, 2, \ldots$. From the continued fraction expansion:

$$a = c_0 + \cfrac{1}{c_1 + \cfrac{1}{c_2 + \cfrac{1}{\ddots}}}, \quad a = [c_0; c_1, c_2, \ldots].$$

For $c_0 \in \mathbb{Z}$ and $c_i \in \mathbb{N}$ where $i \geq 1$, the values of c_i are calculated as follows:

$$\begin{aligned}
c_0 &= \lfloor a \rfloor, \quad \varepsilon_0 = a - c_0, \\
c_1 &= \left\lfloor \frac{1}{\varepsilon_0} \right\rfloor, \quad \varepsilon_1 = \frac{1}{\varepsilon_0} - c_1, \\
c_2 &= \left\lfloor \frac{1}{\varepsilon_1} \right\rfloor, \quad \varepsilon_2 = \frac{1}{\varepsilon_1} - c_2, \\
&\vdots \\
c_i &= \left\lfloor \frac{1}{\varepsilon_{i-1}} \right\rfloor, \quad \varepsilon_i = \frac{1}{\varepsilon_{i-1}} - c_i.
\end{aligned}$$

The k-th convergence of $a = [c_0; c_1, c_2, \ldots] \in \mathbb{R}^+$ is calculated as follows

$$a = \frac{a_k}{b_k} = [c_0; c_1, c_2, \ldots, c_k].$$

Convergents are computed iteratively:

$$\begin{aligned}
\frac{a_0}{b_0} &= \frac{c_0}{1}, \\
\frac{a_1}{b_1} &= \frac{c_1 a_0 + 1}{c_1}, \\
&\vdots \\
\frac{a_k}{b_k} &= \frac{c_k a_{k-1} + a_{k-2}}{c_k b_{k-1} + b_{k-2}}, \quad \text{where} \quad k = 0, 1, ...
\end{aligned}$$

$x_k = [a_0, a_1, \ldots, a_k]$ is defined accordingly.

Before factoring n, a factor base $B = \{-1, p_1, \ldots, p_L\}$ is used. Squares that are B-smooth are added for processing. That is,

$$Y_k \equiv x_k^2 \pmod{n}, \quad x \in \mathbb{Z}.$$

The process involves identifying which products of Y_k form a perfect square. If a perfect square is obtained, the product of Y_k equals y^2, and the product of x_k modulo n gives x. Subsequently, x and y are added and subtracted. Finally, the greatest common divisor (gcd) with n is computed to determine the prime factors.

$$\gcd(x - y, n) = p, \quad \gcd(x + y, n) = q.$$

Example 5.2. Let us factorize $n = 11021$ using the well-known method based on the continued fractions.

We compute convergence for $\sqrt{11021} \approx 104,980951...$

$$\frac{a_0}{b_0} = \frac{104}{1}, \frac{a_1}{b_1} = \frac{105}{1}, \frac{a_2}{b_2} = \frac{5459}{52}, \frac{a_3}{b_3} = \frac{11023}{105}, ...$$

Smallest absolute residue Y_i of $a_i^2 \pmod{11021}$;

i	0	1	2	3	...
$x_i = a_i$	104	105	5459	11023	...
$Y_i \equiv a_i^2 \mod n$	205	4	103	4	...

Thus, $y^2 = Y_1 = 2^2$, and $x = x_1 \equiv 105 \pmod{11021}$, with $2^2 \equiv 105^2 \pmod{11021}$. Then, we can factorize n as $\gcd(105 + 2, 11021) = 107$ and $11021 = 107 \times 103$.

Example 5.3. Let us factorize $n = 9073$ using the well-known method based on the continued fractions.

For $\sqrt{9073} \approx 95.2523\ldots$, the approximation is calculated as follows:

$$\begin{aligned}\sqrt{9073} &\approx 95 + \frac{2523}{10000}\\ &= 95 + \frac{1}{\frac{10000}{2523}}\\ &= 95 + \frac{1}{3 + \frac{2431}{2523}}\\ &= 95 + \frac{1}{3 + \frac{1}{1+\frac{1}{26+\frac{1}{2+\frac{1}{\ddots}}}}}.\end{aligned}$$

Since n is a four-digit number, only the first four coefficients are taken into consideration, except the integer part. Thus, the coefficients are $[95; 3, 1, 26, 2]$. After obtaining the coefficients, the values of $\frac{a_i}{b_i}$ are found for $(i = 0, 1, 2, \ldots)$:

$$\begin{aligned}\frac{a_0}{b_0} &= \frac{95}{1},\\ \frac{a_1}{b_1} &= 95 + \frac{1}{3} = \frac{286}{3},\\ \frac{a_2}{b_2} &= 95 + \frac{1}{3 + \frac{1}{1}} = \frac{381}{4},\\ \frac{a_3}{b_3} &= 95 + \frac{1}{3 + \frac{1}{1+\frac{1}{26}}} = \frac{10192}{107},\\ \frac{a_4}{b_4} &= 95 + \frac{1}{3 + \frac{1}{1+\frac{1}{26+\frac{1}{2}}}} = \frac{20765}{218}.\end{aligned}$$

The sequence $x_i = (95, 286, 381, 10192, 20765\ldots)$ is then used to compute the Y_i values:

$$Y_0 \equiv 95^2 \pmod{9073} = 9025.$$

Since the number 9025 is close to 9073, it is subtracted.

$$\begin{aligned}Y_0 &= 9073 - 9025 = 48,\\ Y_1 &\equiv 286^2 \pmod{9073} = 139,\\ Y_2 &\equiv 381^2 \pmod{9073} = 9066,\\ Y_2 &= 9073 - 9066 = 7,\\ Y_3 &\equiv 10192^2 \pmod{9073} = 87,\\ Y_4 &\equiv 20765^2 \pmod{9073} = 9046,\\ Y_4 &= 9073 - 9046 = 27.\end{aligned}$$

The sequence $Y_i = (48, 139, 7, 87, 27, \ldots)$ is obtained. Since none of the Y_i values is a perfect square individually, we look for combinations of Y_i whose product is a perfect square. The combination $Y_0 \cdot Y_4$ is found to be a perfect square.

Thus,

$$y^2 = Y_0 \cdot Y_4 = 48 \cdot 27 = 4^2 \cdot 3 \cdot 3^3 = 36^2$$
$$x = x_0 \cdot x_4 \equiv 95 \cdot 20765 \pmod{9073} = 3834.$$

Finally, x and y are added and subtracted, and the gcd with n is calculated:

$$\gcd(x + y, n) = \gcd(3834 + 36, 9073) = 43$$
$$\gcd(x - y, n) = \gcd(3834 - 36, 9073) = 211$$

Thus, $n = 9073$ is factored as $9073 = 43 \times 211$.

6. An Alternative Approach with Continued Fractions

In this section, we modify the continued fraction method for integer factorization by using Theorems 3.2 and 3.3. These theorems exploit particular numerical properties to identify prime factors both more rapidly and more efficiently, thereby establishing the method as a powerful tool for factoring composite integers. We demonstrate the alternative factorization procedure through detailed examples, illustrating each step to clarify the methodological advancements and highlight the advantages of this approach for factoring large composite numbers.

We can write the followings by using Theorem 3.3. For $n > 1$, $n \in \mathbb{Z}$,

$$\begin{aligned} &\lfloor \sqrt{n} \rfloor = c_0, \\ &\frac{r_0}{b_0} = \frac{1}{\sqrt{n} - c_0} = \frac{\sqrt{n} + c_0}{n - c_0^2}, \quad c_1 = \lfloor \frac{\sqrt{n} + c_0}{n - c_0^2} \rfloor \\ &\frac{r_{i+1}}{b_{i+1}} = \frac{1}{\frac{r_i}{b_i} - c_{i+1}}, \quad c_{i+2} = \lfloor \frac{r_{i+1}}{b_{i+1}} \rfloor \quad \text{where} \quad i = 0, 1, 2, \ldots \end{aligned} \tag{6.1}$$

While solving (6.1), both the coefficients and the b_i values are determined. The critical point is to check whether b_i is a perfect square. B-smooth check is performed, and if b_i is B-smooth, then the product of b_i's is equal to y^2, i.e,

$$\prod b_i = y^2$$

which represents the condition for identifying a perfect square among B-smooth values.

From Theorem 3.2:

$$\begin{aligned} &a_{-2} = 0, \quad a_{-1} = 1 \\ &a_i = a_{i-1}c_i + a_{i-2}, \quad \text{where} \quad i = 0, 1, 2, \ldots \end{aligned} \tag{6.2}$$

This equation (6.2) gives the product of coefficients up to the point where b_i is a perfect square. The result, when taken modulo n, yields the smallest positive integer remainder, x i.e,

$$x = a_i \pmod{n}. \tag{6.3}$$

Using (6.2) and (6.3):

$$x + y \equiv a_i \pmod{n} + \sqrt{\prod b_i}, \tag{6.4}$$

$$x - y \equiv a_i \pmod{n} - \sqrt{\prod b_i}. \tag{6.5}$$

By calculating the greatest common divisor of n with (6.4) and (6.5) separately, the factors of n are found:

$$\gcd(x + y, n),$$
$$\gcd(x - y, n).$$

We give the algorithm for our new method for finding factors of a given number by using continued fractions.

6.1. **The Algorithm for the Proposed Method.** :

Step 1: Enter Input Values

(1) Give a composite integer n to be factored.
(2) Determine the set B of small prime numbers for B-smoothness.

Step 2: Compute Initial Values

(1) Compute the integer part of the square root of n:

$$c_0 = \lfloor \sqrt{n} \rfloor.$$

(2) Compute the initial remainder:

$$b_0 = n - c_0^2.$$

Step 3: Compute Continued Fraction Terms

(1) Initialize values for numerator = c_0 and denominator = b_0.
   - Set lists for c_i and b_i ($i = 1, 2, 3, ...$).
(2) Repeat the following steps until a suitable subset is found:
   - Compute the next term in the continued fraction:

$$c_i = \left\lfloor \frac{c_0 + \text{numerator}}{\text{denominator}} \right\rfloor,$$

$$b_i = \frac{n - (\text{numerator} - \text{new numerator} \times \text{denominator})^2}{\text{denominator}}.$$

   - Compute the new numerator:

$$\text{new numerator} = \text{denominator} \times c_i - \text{numerator}.$$

   - Compute the new denominator:

$$\text{new denominator} = \frac{n - (\text{new numerator})^2}{\text{denominator}}.$$

   - Update the numerator and denominator.
   - Append a_i to the list.
   - If b_i contains only prime factors from B, add it to the list.

Step 4: Find a Perfect Square Product

(1) Check all subsets of b_i values.
(2) Compute the product of each subset.
(3) If a product is a perfect square, select that subset and compute:

$$y = \sqrt{\prod b_i}.$$

Step 5: Compute Modular Values

(1) Compute the sequence $a_i = a_{i-1}c_i + a_{i-2}$ ($i = 1, 2, 3, ...$) by using initial values $a_{-2} = 0, \quad a_{-1} = 1$.
(2) Compute x using the selected indices:

$$x = \prod a_i \mod n.$$

(3) Compute $(x + y)$ and $(x - y)$ modulo n.
(4) Find the factors by calculating the greatest common divisor (gcd):

If $\gcd(x + y, n)$ or $\gcd(x - y, n)$ provides a non-trivial factor of n, the factorization is found.

6.2. **Examples Using the Proposed Method:**

We illustrate this method with some examples.

Example 6.1. Let us factorize $n = 9073$ using our modified method based on continued fractions.

$$c_0 = \lfloor \sqrt{9073} \rfloor = 95,$$
$$\frac{r_0}{b_0} = \frac{1}{\sqrt{9073} - 95} = \frac{\sqrt{9073} + 95}{9073 - 95^2} = \frac{\sqrt{9073} + 95}{48}.$$

$c_0 = 95$ and $b_0 = 48$. Since b_0 is not a perfect square, the process continues:

$$c_1 = \left\lfloor \frac{\sqrt{9073} + 95}{48} \right\rfloor = 3,$$
$$\frac{r_1}{b_1} = \frac{1}{\frac{\sqrt{9073}+95}{48} - 3} = \frac{\sqrt{9073} + 49}{\frac{9073-49^2}{48}} = \frac{\sqrt{9073} + 49}{139}.$$

$c_1 = 3$ and $b_1 = 139$. The process continues by checking whether b_1 is a perfect square:

$$a_2 = \left\lfloor \frac{\sqrt{9073} + 49}{139} \right\rfloor = 1,$$
$$\frac{r_2}{b_2} = \frac{1}{\frac{\sqrt{9073}+49}{139} - 1} = \frac{\sqrt{9073} + 90}{\frac{9073-90^2}{139}} = \frac{\sqrt{9073} + 90}{7}.$$

$c_2 = 1$ and $b_2 = 7$. The process continues for b_2, and: $b_0 \cdot b_2 = 48 \cdot 7$, $b_1 \cdot b_2 = 139 \cdot 7$, $b_0 \cdot b_1 \cdot b_2 = 48 \cdot 139 \cdot 7$ does not yield a perfect square.

$$c_3 = \left\lfloor \frac{\sqrt{9073} + 90}{7} \right\rfloor = 26,$$
$$\frac{r_3}{b_3} = \frac{1}{\frac{\sqrt{9073}+90}{7} - 26} = \frac{\sqrt{9073} + 92}{\frac{9073-92^2}{7}} = \frac{\sqrt{9073} + 92}{87}.$$

$c_3 = 26$ and $b_3 = 87$. b_3 is not a perfect square, so we need to check whether their products result in a perfect square $b_0 \cdot b_3 = 48 \cdot 87$, $b_1 \cdot b_3 = 139 \cdot 87$, $b_2 \cdot b_3 = 7 \cdot 87, b_0 \cdot b_1 \cdot b_3 = 48 \cdot 139 \cdot 87$, $b_0 \cdot b_2 \cdot b_3 = 48 \cdot 7 \cdot 87, b_1 \cdot b_2 \cdot b_3 = 139 \cdot 7 \cdot 87$, $b_0 \cdot b_1 \cdot b_2 \cdot b_3 = 48 \cdot 139 \cdot 7 \cdot 87$ did not result in a perfect square, so the process continues.

$$c_4 = \left\lfloor \frac{\sqrt{9073} + 92}{87} \right\rfloor = 2,$$
$$\frac{r_4}{b_4} = \frac{1}{\frac{\sqrt{9073}+92}{87} - 2} = \frac{\sqrt{9073} + 82}{\frac{9073-82^2}{87}} = \frac{\sqrt{9073} + 82}{27}$$

$$c_4 = 2 \quad \text{and} \quad b_4 = 27.$$

b_4 is not a perfect square, so we need to check whether their products result in a perfect square:

$$b_0 \cdot b_4 = 48 \cdot 27 = 36^2$$

which gave a perfect square, so the process ends.

The process continues similarly for c_3 and c_4. Finally, $b_4 = 27$ forms a perfect square:

$$y = \sqrt{36^2} = 36.$$

It is obtained. Now, x is found:

$$\begin{aligned}
a_{-2} &= 0, \quad a_{-1} = 1 \\
a_0 &= a_{-1}c_0 + a_{-2} = 1 \cdot 95 + 0 = 95 \\
a_1 &= a_0c_1 + a_{-1} = 95 \cdot 3 + 1 = 286 \\
a_2 &= a_1c_2 + a_0 = 286 \cdot 1 + 95 = 381 \\
a_3 &= a_2c_3 + a_1 = 381 \cdot 26 + 286 = 10192, \\
a_4 &= a_3c_4 + a_2 = 10192 \cdot 2 + 381 = 20765 \\
x &= a_4 \cdot a_0 \mod n, \quad x = 20765 \cdot 95 \mod 9073 = 3834
\end{aligned}$$

Then,

$$\begin{aligned}
x + y &= 3834 + 36 = 3870 \\
x - y &= 3834 - 36 = 3798
\end{aligned}$$

Now,

$$\begin{aligned}
\gcd(x + y, n) &= \gcd(3870, 9073) = 43 \\
\gcd(x - y, n) &= \gcd(3798, 9073) = 211
\end{aligned}$$

is obtained.

We provide two more examples with the tables based on this new method.

Example 6.2. Let us factorize $n = 91$ using its continued fraction representation.

Table 1. Factorizing $n = 91$ using continued fractions

Calculated quantity	How it is derived	i=0	i=1	i=2	i=3	...
$[\ \lfloor \sqrt{n} \rfloor\ ]$	$[\ c_i, \quad (i = 0, 1, \ldots)\ ]$	9	1	1	5	...
$\frac{r_i}{b_i}$	$\frac{r_0}{b_0} = \frac{1}{\sqrt{n}-c_0}$					
	$\frac{r_{i+1}}{b_{i+1}} = \frac{1}{\frac{r_i}{b_i}-c_{i+1}}$	$\frac{\sqrt{91}+9}{10}$	$\frac{\sqrt{91}+1}{9}$	$\frac{\sqrt{91}+8}{3}$	$\frac{\sqrt{91}+7}{16}$	...
a_i	$a_{-2} = 0, a_{-1} = 1$					
	$a_{i-1}c_i + a_{i-2}$	9	10	19	105	...
y	$\sqrt{b_i}$	$\sqrt{10}$	3	$\sqrt{3}$	4	...
x	$a_i \pmod n$	9	10	19	14	...
$x + y$	$a_i \pmod n + \sqrt{b_i}$	$9 + \sqrt{10}$	$10 + 3 = 13$	$19 + \sqrt{3}$	$14 + 4 = 18$	...
$x - y$	$a_i \pmod n - \sqrt{b_i}$	$9 - \sqrt{10}$	$10 - 3 = 7$	$19 - \sqrt{3}$	$14 - 4 = 10$	...
$gcd(x + y, n)$		-	13	-	-	...
$gcd(x - y, n)$		-	7	-	-	...

Example 6.3. Let us factorize $n = 253$ using its continued fraction representation.

TABLE 2. Factorizing $n = 253$ using continued fractions

Calculated quantity	How it is derived	i=0	i=1	i=2	i=3	...
$[\ \lfloor \sqrt{n} \rfloor\ ]$	$[\ c_i, \quad (i = 0, 1, \ldots)\ ]$	15	1	9	1	...
$\frac{r_i}{b_i}$	$\frac{r_0}{b_0} = \frac{1}{\sqrt{n}-c_0}$					
	$\frac{r_{i+1}}{b_{i+1}} = \frac{1}{\frac{r_i}{b_i}-c_{i+1}}$	$\frac{\sqrt{253}+15}{28}$	$\frac{\sqrt{253}+13}{3}$	$\frac{\sqrt{253}+14}{19}$	$\frac{\sqrt{253}+5}{12}$	...
a_i	$a_{-2} = 0, a_{-1} = 1$					
	$a_{i-1}c_i + a_{i-2}$	15	16	159	175	...
y	$\sqrt{b_i}$	$2\sqrt{7}$	$\sqrt{3}$	$\sqrt{19}$	$2\sqrt{3}$	...
x	$a_i \pmod n$	15	16	159	175	...
$x + y$	$a_i \pmod n + \sqrt{b_i}$	$15 + 2\sqrt{7}$	$16 + \sqrt{3}$	$159 + \sqrt{19}$	$175 + 2\sqrt{3}$	...
$x - y$	$a_i \pmod n - \sqrt{b_i}$	$15 - 2\sqrt{7}$	$16 - \sqrt{3}$	$159 - \sqrt{19}$	$175 - 2\sqrt{3}$	...
		$(*)$				
$b_1 \cdot b_3$	$3 \cdot 3 \cdot 2^2 = 6^2$					
$a_1 \cdot a_3$	$16 \cdot 175 \pmod{253} = 17$					
x	17					
y	6					
$gcd(x - y, n)$	11					
$gcd(x + y, n)$	23					

$(*)$: The square root of y does not simplify outside of the radical, so we need to check which products of b_i results in a perfect square.

7. Comparison Between the Continued Fraction Factorization Method and the Proposed Method

In this section, the time complexities of the classical and proposed methods are analyzed in detail, and the differences between these two methods are explained based on a mathematical foundation.

7.1. **Time Complexity of the Classical Continued Fraction Method.** The classical continued fraction-based factorization method applies a factorization strategy by performing a continued fraction expansion at each step. The key components determining the time complexity of this method are as follows:

- **Continued Fraction Expansion:**
  - The continued fraction expansion is completed in an average of $O(n^{1/4})$ steps.
  - Each step requires one division operation, two multiplication operations, and one modular exponentiation operation.
- **B-Smooth Testing:**
  - At each step, a modular square is computed and checked for B-smoothness.
  - The B-smooth test involves factorization based on a predefined factor base and requires $O(B)$ division operations per step.
  - The total number of division operations performed is $O(n^{1/4}B)$.

The size of B is generally selected as:

$$B = e^{\sqrt{\log n \cdot \log\log n}}.$$

Thus, the overall time complexity of the classical method is given by:

$$O(n^{1/4} e^{\sqrt{\log n \cdot \log\log n}}).$$

For a comprehensive overview and rigorous complexity analysis of integer factorization techniques, a reader can also refer to [10] and [4].

7.2. **Time Complexity of the Proposed Method.** The proposed method offers a more optimized structure compared to the classical continued fraction method. Specifically, by applying the B-smooth test only once, the total number of division operations is significantly reduced.

- **Continued Fraction Expansion:**
  - The number of steps remains $O(n^{1/4})$.
  - Each step requires one division operation, two multiplication operations, and one modular computation.
- **B-Smooth Testing:**
  - This test is performed only once, requiring $O(B^2)$ division operations in total.
  - Here, B is chosen to be smaller compared to the classical method, specifically $O((\log n)^2)$.

Thus, the total time complexity of the proposed method is expressed as:

$$O(n^{1/4} \log n) + O(B^2).$$

Substituting $B = (\log n)^2$, we obtain:

$$O\left(n^{1/4} \cdot \log n + (\log n)^4\right) \approx O\left(n^{1/4} \cdot \log n\right).$$

In many integer factorization algorithms, the choice of the smoothness bound plays a critical role in determining the overall efficiency. A well-chosen bound can significantly affect the performance of the smoothness testing phase. So, the choice of the bound $B = (\log n)^2$ is intended to minimize the overall computational cost by reducing the number of required divisions during the B-smoothness test, thereby enhancing the practical efficiency of the proposed method. Therefore, the proposed method is more efficient than the classical method, as it avoids the exponential growth caused by repeated B-smooth testing. Instead, it applies the B-smooth test only once, leading to quasi-polynomial time complexity. This makes the proposed approach significantly more scalable and practical for large-scale factorization problems.

8. Conclusion and Future Works

In this study, a novel perspective on the use of continued fractions for factorizing integers is presented. The conventional method exhibits a tendency to decelerate as the parameter B increases due to the repeated B-smoothness checks performed at each step, leading to increased time complexity, particularly for large integers. In contrast, the proposed approach integrates coefficient and modular arithmetic computations concurrently and performs the B-smoothness check only once. This innovative methodology significantly enhances computational efficiency, especially for larger B values. It is suggested that this alternative method may facilitate reaching conclusions and contribute to a better understanding of the topic.The continued fraction factorization method offers a different perspective on factoring large numbers into their prime factors. Advances in factorization methods have significantly impacted cryptography, especially RSA encryption. Wiener attack and the continued fraction algorithm threaten RSA systems with short secret exponents, creating vulnerabilities. Cryptanalytic methods like Wiener attack exploit these weaknesses using continued fractions to break RSA encryption. These concerns highlight the need for research in post-quantum cryptography and cryptanalysis, as quantum computing threatens traditional cryptographic security. Addressing these vulnerabilities is essential to strengthening cryptographic systems against both classical and quantum attacks in the future.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Authors Contribution Statement

All authors jointly worked on the results and they have read and agreed to the published version of the manuscript.

References

[1] Boneh, D., Durfee, G., *Cryptanalysis of RSA with private key* $d < N^{0.292}$, Advances in Cryptology - Proceedings of Eurocrypt '99, Lecture Notes in Computer Science 1952, 1–11, 1999.

[2] Boneh, D., *Twenty years of attacks on the RSA cryptosystem*, Notices Amer. Math. Soc., **46**(1999), 203–213.

[3] Brillhart , J., *A note on Euler's factoring problem*, The American Mathematical Monthly, **116**(10)(2009), 928–931.

[4] Lenstra H.W., Pomerance, C., *A rigorous time bound for factoring integers*, Journal of the American Mathematical Society, **5**(1992), 483–516.

[5] Mollin, R.A., Fundamental Number Theory with Applications, CRC Press, Boca Raton, New York-London-Tokyo, 1998.

[6] Mollin, R.A., An Introduction to Cryptography, Discrete Mathematics and Its Applications, 2007.

[7] Pinch, R.G.E., *Extending the Wiener attack to RSA-type cryptosystems*, *Electronics Letters*, **31**(1995), 1736–1738.

[8] Pollard, J.M., *Theorems on factorization and primality testing*, Proceedings of the Cambridge Philosophical Society, **76**(1974), 521–528.

[9] Pollard, J.M., *A Monte Carlo method for factorization*, BIT Numerical Mathematics, **15**(3)(1975), 331–334.

[10] Pomerance, C, *Analysis and comparison of some integer factoring algorithms*, Computational Methods in Number Theory, **154**(1982), 89–139.

[11] Pomerance, C., The quadratic Sieve Factoring Algorithm in Advances in Cryptology — EUROCRYPT '84, Springer-Verlag, Berlin, LNCS 209, 1985.

[12] Pomerance, C., *A tale of two sieves*, The Notices of the Amer. Math. Soc., **43**(1996), 1473–1485.

[13] Rivest, R., Shamir, A., Adleman, L., *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM, **21**(2), 120–126.

[14] Rosen, Kenneth H., Elementary Number Theory and Its Applications, Addison-Wesley Publishing Company, 1986.

[15] Wiener, M.J., *Cryptanalysis of short RSA secret exponents*, IEEE Transactions on Information Theory, **36**(1990), 553–558.

[16] Williams, H.C., *A* $p + 1$ *method of factoring*, *Mathematics of Computation*, **39**(159)(1982), 225–234.