Adaptive Polynomial Factorization (APF) Method: Enhanced Factorization using Modified Pollard Rho Algorithm

Anil Sharma

February 5, 2024

Abstract

This research paper introduces the Adaptive Polynomial Factorization (APF) Method, an enhanced factorization technique based on the Modified Pollard Rho Algorithm. The method incorporates adaptive polynomial evaluation, providing efficiency in factorization tasks. The paper presents a mathematical representation, performance analysis, and examples showcasing the APF Method's versatility and superiority over the original Pollard Rho algorithm.

1 Introduction

The Adaptive Polynomial Factorization (APF) Method is presented as an enhanced version of the classical Pollard Rho algorithm, renowned for its effectiveness in factorizing composite numbers. The APF Method incorporates dynamic polynomial evaluation, providing adaptability to different input characteristics.

2 Mathematical Representation

2.1 Initialization

- Choose a starting value x_0.
- Set $x = x_0$ and $y = x_0$.

2.2 Iteration

For each iteration, compute:

$$\begin{aligned} x_{i+1} &= FX(x_i) \mod n \\ y_{i+1} &= FX(FX(y_i)) \mod n \end{aligned}$$

2.3 Factor Detection

Compute the GCD of $|x_i - y_i|$ and n:

$$\text{GCD}(|x_i - y_i|, n) = d$$

2.4 Factor Check

If d is not equal to 1 or n, then d is a non-trivial factor of n, and the algorithm terminates.

3 The APF Method

The APF Method stands out as a powerful tool for factorization tasks, offering superior performance compared to the original Pollard Rho algorithm. The adaptive polynomial evaluation, as described in Section 2, provides versatility, making it well-suited for a broad spectrum of input characteristics.

4 Example of Chosen Polynomials

To illustrate the APF Method's functionality, consider the following example of polynomials:

$$FX(x) = 3x^2 + 4x + 5$$
$$F(x) = (x + 1)^3 - x + 1$$

These polynomials showcase the adaptability of the APF Method. The algorithm dynamically evaluates these polynomials during each iteration, efficiently detecting non-trivial factors.

5 Performance Analysis

5.1 Comparative Examples

5.1.1 Example 1

- Number: $n = 917319371937193719317391739137131313$
- Iterations: APF Method (97) vs Original (253)
- Time Complexity: APF Method (0.000544s) vs Original (0.001583s)
- Factors: APF Method (137) vs Original (137)

5.1.2 Example 2

- Number: $n = 91371937132376347856234571229812938129381291$
- Iterations: APF Method (1) vs Original (4)
- Time Complexity: APF Method (0.000021s) vs Original (0.000043s)
- Factors: APF Method (391) vs Original (391)

5.1.3 Example 3

- Number: $n = 9173913718927312893712983713917319371391731937193193719 31$
- Iterations: APF Method (6) vs Original (16)
- Time Complexity: APF Method (0.000082s) vs Original (0.000281s)
- Factors: APF Method (107) vs Original (107)

5.1.4 Example 4

- Number: $n = 1890331997086548101701457236399054118699828392181624283870321942427 9$
- Iterations: APF Method (888) vs Original (3847)
- Time Complexity: APF Method (0.015450s) vs Original (0.098103s)
- Factors: APF Method (17303159) vs Original (17303159)

6 New Example

- Enter the value of n: 973917319371937193719999888887777771111
- **Modified Pollard Rho:**
  - Factors: 8210651
  - Number of Iterations: 1173
  - Time taken: 0.007245 seconds
- **Original Pollard Rho:**
  - Factors: 8210651
  - Number of Iterations: 2983
  - Time taken: 0.021525 seconds

7 Uniqueness of the APF Method

The APF Method's uniqueness lies in its dynamic polynomial evaluation, allowing it to adapt to various input characteristics. This adaptability, showcased in the chosen polynomial example, makes the APF Method a powerful and efficient factorization algorithm.