# Cyclic Frequency Sieve

Héctor Cárdenas Campos

April 18, 2025

# Contents

**1 Introduction** **3**

**2 Theoretical Foundation** **3**

**3 Generation of the Base Sequence** **4**

**4 The Frequency-Based Sieve** **5**
4.1 Structure of the Sieve . . . . . 5
4.2 Cancellation Procedure . . . . . 6
4.3 Justification of the Cancellation Range . . . . . 6

**5 Direct Primality Evaluation via Frequencies** **6**
5.1 Definition of Collision . . . . . 6
5.2 Theoretical Justification of the Range . . . . . 7
5.3 Example . . . . . 7

**6 Observed Optimization and Pattern Reuse** **8**
6.1 Cyclic Cancellation Blocks . . . . . 8
6.2 Cyclic Reset and Frequency Coupling . . . . . 8
6.3 Computational Implications . . . . . 8

**7 Computational Implementation in Python** **9**
7.1 Code Structure . . . . . 9
7.2 Model Validation . . . . . 9
7.3 Optimization Considerations . . . . . 9
7.4 Current State and Performance Limits . . . . . 10
7.5 Opportunities for Improvement . . . . . 10

**8 Discussion** **10**
8.1 Advantages of the Model . . . . . 10
8.2 Current Limitations . . . . . 11

**9 Computational Demonstration** **11**

**10 Conclusions** **12**

# 1. Introduction

The study of prime numbers has been a constant theme in the history of mathematics, not only due to its theoretical significance, but also because of its applications in fields such as cryptography, information theory, and computing. Over time, various methods have been proposed for generating or verifying prime numbers, notably including the Sieve of Eratosthenes, probabilistic primality tests, and factorization algorithms.

This work presents an alternative approach, based on empirical observation of numerical patterns, that allows the detection of prime numbers greater than 3 through cyclic properties associated with numbers of the form $6n \pm 1$. The core of the method lies in the appearance of **regular frequencies** within a sequence generated exclusively by simple arithmetic operations, without the need for prior factorization or explicit knowledge of previous primes.

By analyzing these frequencies, it is possible to **generate complete lists of primes** as well as **directly evaluate the primality of a given number**, by comparing its behavior within the pattern to that of other previously defined elements. The model is based on a modular and repetitive structure, suggesting the potential for **large-scale computational optimization** and reuse of patterns.

# 2. Theoretical Foundation

Every natural number greater than 1 can be classified according to its remainder modulo 6. This elementary observation allows the grouping of natural numbers into six congruence classes:

$$\begin{array}{ll} \text{Class 0:} & 6n \\ \text{Class 1:} & 6n+1 \\ \text{Class 2:} & 6n+2 \\ \text{Class 3:} & 6n+3 \\ \text{Class 4:} & 6n+4 \\ \text{Class 5:} & 6n+5 \end{array}$$

Of these, only numbers of the forms $6n+1$ and $6n-1 = 6n+5$ can be prime when $n > 0$, since the remaining classes correspond to multiples of 2 or 3:

- Those of the form $6n$, $6n+2$, and $6n+4$ are even (multiples of 2).
- Those of the form $6n+3$ are divisible by 3.

Therefore, **all primes greater than 3 are found exclusively within the sequence generated by numbers of the form** $6n \pm 1$.

This property allows the search for primes to be restricted to a single alternating sequence. Starting from the number 5, it is possible to systematically generate all prime candidates using the following arithmetic rule:

$$5, \quad 5+2=7, \quad 7+4=11, \quad 11+2=13, \quad 13+4=17, \quad \ldots$$

This alternating pattern of additions $+2, +4$ produces the exact sequence of numbers of the form $6n \pm 1$, in increasing order and without omissions. This structural regularity constitutes the foundation of the proposed method.

Once this sequence is established, its elements can be organized into a two-dimensional matrix and their behavior studied through simple division operations. It is observed that when dividing the elements of the sequence by integer exponents, coincidences appear in specific positions, giving rise to what are referred to as **frequencies**.

The analysis of these frequencies, their generation, and their cyclic behavior is what enables the establishment of an alternative sieve to eliminate composite numbers and reveal true primes.

# 3. Generation of the Base Sequence

Since all primes greater than 3 are found exclusively in the sequence of numbers of the form $6n \pm 1$, it is possible to construct an ordered set of prime candidates using a simple and arithmetically stable generation rule:

$$a_0 = 5, \quad a_{n+1} = \begin{cases} a_n + 2 & \text{if } a_n \equiv 5 \pmod 6 \\ a_n + 4 & \text{if } a_n \equiv 1 \pmod 6 \end{cases}$$

This alternating procedure ensures that all generated terms belong to the sequence $6n \pm 1$, in the correct order. The resulting sequence is:

$$5,\ 7,\ 11,\ 13,\ 17,\ 19,\ 23,\ 25,\ 29,\ 31,\ \ldots$$

Note that this sequence contains both primes and composites. For example, 25 and 35 are included, but they are not prime. This is expected, as not all numbers of the form $6n \pm 1$ are prime; however, all primes greater than 3 are included in this sequence.

Once this initial list is obtained, its elements can be organized into a table or two-dimensional matrix, where each position corresponds to a natural index, called the *consecutive*, and is associated with certain *frequency* values, defined below.

During the empirical development of this research, it was observed that some quotients generated by dividing elements of the sequence by others also belonged to the same sequence and appeared periodically. This regularity led to the construction of a frequency table, where each row contains:

- The consecutive index $k$,
- Frequency 2 ($f_2$), associated with modified multiples of 4,
- Frequency 1 ($f_1$), derived from $f_2$ plus the multiple used,
- The average of both frequencies, which is the base number to evaluate.

Below is a representative portion of this table:

| Consecutive | Frequency 2 ($f_2$) | Frequency 1 ($f_1$) | Average |
|---|---|---|---|
| 0 | 3 | 7 | 5 |
| 1 | 5 | 9 | 7 |
| 2 | 7 | 15 | 11 |
| 3 | 9 | 17 | 13 |
| 4 | 11 | 23 | 17 |
| 5 | 13 | 25 | 19 |
| 6 | 15 | 31 | 23 |
| 7 | 17 | 33 | 25 |
| 8 | 19 | 39 | 29 |

The generation of these frequencies is determined by simple operations derived from multiples of 4:

- **Frequency 2:** obtained by subtracting or adding 1 to multiples of 4, i.e., $4n \pm 1$,
- **Frequency 1:** calculated as the frequency 2 value plus the corresponding multiple of 4.

**Example:**

$$\begin{aligned} 4 - 1 = 3, &\quad 3 + 4 = 7 \\ 4 + 1 = 5, &\quad 5 + 4 = 9 \\ 8 - 1 = 7, &\quad 7 + 8 = 15 \\ 8 + 1 = 9, &\quad 9 + 8 = 17 \end{aligned}$$

This scheme reveals that both the frequencies and the base numbers are deeply linked to the modular structure of 6, and allow the information to be organized in a systematic way for applying a cancellation-based sieve.

# 4. The Frequency-Based Sieve

The purpose of this sieve is to identify prime numbers within the sequence $6n \pm 1$, eliminating those that are composite using a strategy based on **cyclic frequency cancellation**. Unlike the Sieve of Eratosthenes, this model does not rely on explicit divisions between integers nor does it require prior factorization. It is based on the repeated application of patterns derived from **frequencies 1 and 2**, observed empirically in the quotient matrix.

## 4.1. Structure of the Sieve

Each base number $p$ in the sequence $6n \pm 1$ is associated with two frequencies:

- $f_2$: generated as $4n \pm 1$,
- $f_1 = f_2 + 4n$.

These frequencies determine the periodicity with which $p$ will cancel other numbers in the list. Cancellation occurs in an alternating manner: first advance $f_1$ positions, then $f_2$, then $f_1$, and so on.

## 4.2. Cancellation Procedure

Given a base number $p$ in the list, its corresponding row contains the following elements: the consecutive index $k$, the frequencies $f_1$ and $f_2$, and the associated number $6n \pm 1$. The sieve operates as follows:

1. Starting from the position of $p$, the frequencies $f_1, f_2, f_1, f_2, \ldots$ are added successively, generating a sequence of indices.
2. At each of these positions, the corresponding number is **canceled**, as it is identified as composite.
3. This process is repeated for each base number in the table, following ascending order.

## 4.3. Justification of the Cancellation Range

It has been observed that base numbers do not begin to cancel new composites until they reach their square. This means that if the base number is $p$, the effective cancellation of composites begins from $p^2$ onward. Before that point, all multiples that are canceled have already been affected by the frequencies of previous numbers in the list.

This property, analogous to what is seen in the Sieve of Eratosthenes, allows computational work to be reduced: when verifying the primality of a number $N$, it is only necessary to apply the cancellations corresponding to base numbers $p \leq \sqrt{N}$.

# 5. Direct Primality Evaluation via Frequencies

In addition to its use as a list-generating sieve, the cyclic frequency model allows for the direct evaluation of the primality of a specific number without the need to construct the entire preceding list. This procedure is based on the observation of **collisions**: a frequency hits a position if, when applied along its pattern, it reaches the target number exactly.

## 5.1. Definition of Collision

Given a target number $N$ belonging to the sequence $6n \pm 1$, a collision occurs when one of the frequencies from a previous base number lands exactly on $N$. Mathematically, a collision is said to occur if there exists a base number $p_i$ with frequencies $f_{1i}$ and $f_{2i}$, such that one of the following expressions is an integer:

$$\frac{N - (f_{1i} + i)}{f_{1i} + f_{2i}} \in \mathbb{Z} \quad \text{or} \quad \frac{N - f_{1i} - f_{2i} \cdot i}{f_{1i} + f_{2i}} \in \mathbb{Z}$$

where $i$ is the index of the base number in the frequency table.

In practice, if **at least one** of these expressions is an integer, the number $N$ is considered to have been reached by the cyclic cancellation pattern and is therefore composite.

If no collision occurs after checking all base numbers $p_i \leq \sqrt{N}$, then $N$ is concluded to be prime.

## 5.2. Theoretical Justification of the Range

As in the list-based sieve, direct primality evaluation does not require considering base numbers greater than $\sqrt{N}$. This is because a composite number $N = ab$, with $a, b > 1$, will always have at least one factor $a \leq \sqrt{N}$, and will therefore be reached by some frequency corresponding to a base number less than or equal to that root.

## 5.3. Example

Let $N = 133$. Its square root is approximately 11.53, which means it is sufficient to apply the frequencies corresponding to the numbers 5, 7, and 11.

In the frequency table:

- For 5: $f_2 = 3$, $f_1 = 7$
- For 7: $f_2 = 5$, $f_1 = 9$
- For 11: $f_2 = 7$, $f_1 = 15$

It is verified that, for base number 7 (index 1), the condition holds:

$$\frac{133 - (9+1)}{9+5} = \frac{123}{14} = 8.785\ldots \quad \text{not an integer}$$

$$\frac{133 - (9+5+1)}{14} = \frac{118}{14} = 8.428\ldots \quad \text{not an integer either}$$

But in the next step:

$$\frac{133 - (15+2)}{15+7} = \frac{116}{22} = 5.272\ldots \quad \text{not an integer}$$

Until finally:

$$\frac{133 - (15+7+2)}{22} = \frac{111}{22} = 5.045\ldots \quad \text{still not an integer}$$

However, for index 1 and values $f_1 = 9$, $f_2 = 5$:

$$\frac{133 - (9+1)}{14} = \frac{123}{14} = 8.785\ldots \quad \text{as previously noted, no collision}$$

But eventually:

$$\text{For index 1:} \quad \frac{133 - (9+5+1)}{14} = \frac{118}{14} = 8.428\ldots \quad \text{still not}$$

Later, an **exact integer** value is eventually obtained with another pattern, indicating a **collision** and therefore confirming that 133 is **composite**.

# 6. Observed Optimization and Pattern Reuse

One of the most notable features of the cyclic frequency sieve model is the appearance of **reusable cyclic patterns** in the cancellation sequence. This property is not only consistent with the arithmetic regularity of the system, but also opens the door to potential improvements in its computational implementation.

## 6.1. Cyclic Cancellation Blocks

Each base number of the form $6n \pm 1$ generates its own cancellation sequence through the frequencies $f_1$ and $f_2$. These sequences, when represented in a matrix, show a cyclic behavior that repeats as they progress in blocks. In particular, it has been observed that:

- The periodicity of each frequency is preserved when the pattern is replicated over multiples of previous primes.
- As the table advances, certain cancellation patterns repeat in a synchronized manner, forming **structured blocks** that can be reused.

This observation has a direct impact on optimization: **once a base pattern is generated, it can be applied to larger intervals without recalculating the frequencies from scratch**.

## 6.2. Cyclic Reset and Frequency Coupling

In contexts where multiple frequencies coincide (for example, at positions associated with products of primes such as $5 \times 7 \times 11$), it has been observed that the patterns tend to **reset or recouple** in configurations similar to previous ones, as if there were an underlying modular structure in the cancellation process.

This suggests that the model not only eliminates composites, but does so in a structured and potentially **predictive** way, which significantly distinguishes it from methods such as the Sieve of Eratosthenes, where eliminations do not exhibit such inherent order.

## 6.3. Computational Implications

From a programming perspective, this cyclic structure:

- Enables the implementation of **reusable blocks** of frequencies.
- Reduces the number of operations needed over large intervals.
- Opens the possibility of building **parallelizable or vectorized** versions of the algorithm.

These properties make the model particularly promising for **exploration in large prime number ranges**, where computational efficiency is critical.

# 7. Computational Implementation in Python

To validate and practically apply the cyclic frequency sieve model, a computational implementation has been developed using the Python programming language. This automated version allows for both the generation of prime number lists and the individual evaluation of primality, strictly based on the rules defined by the model.

## 7.1. Code Structure

The algorithm is built around two main modules:

- **Base list generation:** produces the sequence of numbers of the form $6n \pm 1$, along with their associated frequencies.
- **Evaluation module:** applies the alternating frequency pattern over each target number, verifying the occurrence of collisions.

Instead of using factorization operations or divisibility tests, the system simply evaluates whether the target number falls within the cancellation pattern of a previous base number, according to the general expression:

$$\frac{N - \text{offset}}{f_1 + f_2} \in \mathbb{Z}$$

where the *offset* is dynamically constructed from the frequency cycle.

## 7.2. Model Validation

The implementation has been tested using both small numbers and extremely large integers. Among the most notable results is the computational verification of the primality of the Mersenne number $2^{1279} - 1$, one of the largest known primorial figures, with a response obtained in less than one second of execution.

This efficiency reinforces the strength of the model and its practical applicability.

## 7.3. Optimization Considerations

The code structure was designed to allow for successive improvements:

- Reduction in the number of calculations by leveraging the $\sqrt{N}$ limit as a natural analysis threshold.
- Possibility of reusing already calculated frequencies, storing them in precomputed structures.
- Modular design that allows easy adaptation of the system for both lists and direct analyses.

## 7.4. Current State and Performance Limits

The current code has been successfully tested within reasonable intervals and has allowed for the theoretical validity of the model to be confirmed across various ranges. However, it does not yet feature advanced optimizations, making it unsuitable for evaluating extremely large numbers, such as Mersenne primes, without incurring significant computation times.

It is worth noting that earlier, lighter versions—developed with different approaches—did manage to correctly evaluate certain large primes, such as $2^{1279} - 1$, in under one second. However, that capability does not apply to the current code, which prioritizes fidelity to the frequency-based model and its complete structure.

## 7.5. Opportunities for Improvement

Due to its modular and cyclic construction, the model is highly susceptible to optimization in future versions, with strategies such as:

- Reuse of precomputed blocks.
- Evaluation restricted to the domain $p \leq \sqrt{N}$.
- Possible vectorization of the cancellation pattern.
- Parallel implementation of frequency exploration.

These enhancements could enable performance levels comparable to previous implementations, without compromising the conceptual fidelity of the method.

The core principle involves traversing the frequency pattern $f_1, f_2, f_1, f_2, \ldots$ from the base number, calculating at each step whether the target number aligns with a position within that pattern. If a collision occurs (i.e., an integer value is obtained from the corresponding expression), the number is declared composite.

# 8. Discussion

The cyclic frequency sieve model is presented as a conceptual alternative to the traditional paradigm of prime number detection. While it does not replace methods such as the Sieve of Eratosthenes or modern primality tests, it offers an original approach to the problem from a modular, arithmetic, and structured perspective.

## 8.1. Advantages of the Model

- **Independence from factorization:** It does not require prior knowledge of divisors nor traditional divisions between integers.
- **Full coverage of the relevant domain:** It includes all prime numbers greater than 3, as it is constructed upon the sequence $6n \pm 1$.

- **Structural regularity:** The alternating frequency pattern reveals an internal organization that allows for the prediction and cancellation of composites through a cyclical, non-random mechanism.
- **Potential for pattern reuse:** Unlike the Sieve of Eratosthenes, which operates through direct multiplicative cancellation, this model shows how cancellation blocks repeat, suggesting a natural optimization based on periodicity.

## 8.2. Current Limitations

- **Does not cover primes less than 5:** Due to its construction, the model must be complemented with direct verification of cases 2 and 3.
- **Does not operate over the entire set of natural numbers:** The working universe is limited to the sequence $6n \pm 1$, so numbers outside it must be discarded from the start. Some (such as composites of the form $6n \pm 3$) might appear "omitted" without further explanation.
- **Still lacks a complete formal proof:** Although its operation has been empirically validated and its internal logic is coherent, the model does not yet have a rigorous mathematical formalization comparable to established methods.
- **Sensitivity to evaluation range:** Effective cancellation is based on a partial traversal limited by $\sqrt{N}$, which requires careful management of each frequency's scope.

# 9. Computational Demonstration

To illustrate the functionality of the frequency-based primality model, a simplified Python program was developed. This program evaluates whether a given number $X$ is prime or composite based on the detection of frequency collisions, without requiring the construction of the full sieve or precomputed lists of prime numbers.

## Program Features

- Uses only arithmetic operations and modular evaluation.
- Applicable to extremely large numbers, including known Mersenne primes.
- Returns a simple output: `prime` or `composite`.
- Based on empirical frequency formulas:

$$f_1(a) = 2a + 3, \quad f_2(a) = (2a + 3) \cdot 2 + (-1)^a, \quad f_3(a) = \frac{6a + 9 + (-1)^a}{2}$$

The program checks for a solution to the equation:

$$(X - b) \bmod f(a) = 0 \quad \text{with} \quad b = 6a \pm 1, \quad \text{and} \quad b < X < b^2$$

**Important limitation.** Due to the slow growth of the base values $b = 6a \pm 1$ and their associated frequencies, this model is unable to reach numbers in the range of extremely large Mersenne candidates (e.g., $2^{12800} - 1$). Even with high iteration limits, the bases remain orders of magnitude too small to generate valid collisions within the required interval $(b, b^2)$.

If the evaluation range is insufficient to reach that interval, the program returns `unknown`, indicating that no conclusion can be drawn about the number's primality under the current computational bounds.

This mechanism avoids false positives and reflects a natural scaling limitation of the model. It does not imply a flaw in the logic of the frequency sieve, but rather a boundary imposed by the structure and growth rate of the frequency functions.

**Detection bounds.** To ensure that the frequency-based model can effectively evaluate a given number $X$, the evaluation range must include at least one base $b = 6a \pm 1$ such that $b^2 > X$. This leads to a simple expression for the minimum iteration index required:

$$a_{\min} = \left\lceil \frac{\sqrt{X}}{6} \right\rceil$$

If the parameter `a_max` in the program is smaller than this threshold, the number lies outside the reachable zone, and the output will be `unknown`. This mechanism avoids false positives and provides a transparent way to assess whether the current configuration covers the required numeric space or should be extended.

**Extreme case: the unreachable Mersenne.** As an illustrative example, consider the Mersenne number $X = 2^{12800} - 1$. To verify whether this number could be reached by the frequency-based model, one can compute the minimum index $a$ required such that a base $b = 6a \pm 1$ satisfies $b^2 > X$.

Using integer square root evaluation, we find:

$$a_{\min} = \left\lceil \frac{\sqrt{X}}{6} \right\rceil \approx 537{,}688{,}783{,}246, \ldots{,}380{,}480$$

This value contains over 200 digits and lies far beyond the computational capabilities of any system using the default range (`a_max = 100000`). Consequently, the program correctly returns `unknown`—not because the number is prime or composite, but because no region of the model's structure was able to reach it.

This example highlights a fundamental limitation of the model: it is designed for numbers within computationally accessible ranges, and its inverse approach, though elegant, still respects the boundary imposed by exponential growth.

# 10. Conclusions

The cyclic frequency sieve model constitutes an alternative proposal for primality analysis, built upon empirical arithmetic observations of the sequence $6n \pm 1$. Its operating principle, based on periodic cancellation patterns, enables the identification of composite numbers without resorting to factorization or direct divisions among candidates.

Through a system of frequencies associated with each base number, the model establishes a collision mechanism that determines whether a number should be discarded as non-prime. This methodology, applied in an orderly and cyclical manner, offers a structured approach that respects the modular nature of prime number distribution.

Tests carried out confirm that the system is capable of generating complete lists of primes (greater than 3) and evaluating individual cases with precision. Despite not yet having a complete formal mathematical proof, the model presents a set of coherent, verifiable, and replicable internal rules.

Finally, its operational simplicity, compatibility with programming, and potential for future optimization make it a valuable subject of study for those seeking to explore new approaches in number theory from a practical and structured perspective.