

A New Approach for Examining a Given Number to be a Prime Number

Lourens Nicolaas Jacobus Engelbrecht*

Independent Researcher, 33 William Road, Pringle Bay, 7196, South Africa

Email: engelbrechtlourie@gmail.com

Abstract

Any whole number greater than 1 that is divisible only by 1 and itself, is defined as a prime number. Therefore a prime is a number that has only two factors. The chaotic nature of prime numbers in the set of integer numbers makes it important in today's world, particularly in cryptosystems. If it was possible to determine the predictable nature of primes, the cryptosystems of the world could crash. There are a number of mathematical arguments backing the fact that it is not possible to predict the nature of prime numbers. So one can only formulate a tool for testing whether a given number is likely to be a prime number. Following the theme, here in this article, we also formulate a method to check whether a given number is a probable prime number or not. We also derive a connection between the Collatz conjecture and prime numbers.

Keywords: Congruence; Collatz conjecture; Matlab programming; Prime numbers; Primality test.

1. Introduction

Mathematical theorems are important building blocks for mathematics and computing. The theory of prime numbers and the theorems related to prime numbers are widely used in information technology such as public-key cryptography. Although no general formula for prime numbers exist yet, the search for a more efficient or parallel formula for finding probable prime numbers are of the utmost importance as to save time on irrelevant possibilities such as composite numbers. A primality test is an algorithm for determining whether an input number is a probable prime or not. Primality tests only state whether the input number is likely to be prime or not. A few mathematical formulas for identifying probable primes do exist and are translated into computer codes such as Python, C++, Java Script and R which execute the primality testing. One of the examples of the primality tests is Fermat's primality test. In this test, Fermat's theorem and modular exponentiation is used to give a way to test if a number is a probable prime number, (for more see [1]).

* Corresponding author.

Similarly, Rabin–Miller’s primality test is a test using probabilistic methods. In this test, an algorithm using probability is formulated which says whether a given number is likely to be prime, (see [2]). Another example of a primality test using probability is the Solovay–Strassen primality test, which helps to determine whether a number is composite or probable prime, (see [3]). Some other examples of the methods used for finding prime numbers are the sieve of Eratosthenes (250 BCE) and the sieve of Sundaram (1934).

More recent progress was the AKS primality test, a deterministic primality-proving algorithm created and published by computer scientists Agrawal M, Kayal M, and Saxena N, on August 6, 2002, in an article titled "PRIMES is in P". (Also known as the Agrawal–Kayal–Saxena primality test and cyclotomic AKS test), (see [4]). In computational number theory, the APR primality test is an algorithm for determining whether a number is prime. It is named after its discoverers, Leonard Adleman, Carl Pomerance, and Robert Rumely and later improved by Henri Cohen and Hendrik Willem Lenstra, commonly referred to as APR-CL, (see [5]).

This article is specifically based on a novel theorem. A new method is devised, adding one more probability primality test, to check whether a given number is a prime number or not. In the proof, the main notions which we are using are defined as follows:

1.1 Definition

Let m be a fixed positive integer and, a and b be any two integers. If $m \mid (b - a)$, we say that a is congruent to b modulo m and we write it as: (For more see [6])

$$a \equiv b \pmod{m}.$$

1.2 Definition

Let $S = \{a, b, c\}$ be a set of integers (Z). We say that Z is equal to the union of equivalence of the elements of S if $Z = [a] \cup [b] \cup [c]$, where $[.]$ denotes the equivalence class (see [7]).

2. Methods and main result

In this section, we prove our main result.

2.1: Theorem.

Every prime number other than 2 and 3 satisfies one of the following two algebraic expressions:

$$x - n90 - 5 = 6m$$

or

$$x - n90 - 7 = 6m$$

where $n90$ is the nearest multiple of 90 and m is a whole number.

2.2: Proof

Let $x > 1$ be a positive integer, then we have to show that, if x is a prime number, then it satisfies

$$x - n90 - 5 = 6m$$

or

$$x - n90 - 7 = 6m$$

Case studies

We will consider both the expressions one by one.

Case 1

Consider $x - n90 - 5 = 6m$, where m is a whole number.

From the above statement, we can say that 6 is dividing $x - n90 - 5$ perfectly.

In other words, using the notion of congruence modulo, we can write:

$$x - n90 - 5 \equiv 0 \pmod{6}.$$

From above congruence, we get:

$$x - 5 \equiv 0 \pmod{6},$$

or

$$x \equiv 5 \pmod{6}. \tag{2.2.1}$$

Case 2

Consider $x - n90 - 7 = 6m$.

From the above statement, we can say that 6 is dividing $x - n90 - 7$ perfectly.

Using the notion of congruence modulo, we can write:

$$x - n90 - 7 \equiv 0 \pmod{6}.$$

This implies that:

$$x - 7 \equiv 0 \pmod{6}.$$

The above congruence can be written as:

$$x \equiv 7 \pmod{6},$$

or

$$x \equiv 1 \pmod{6}. \tag{2.2.2}$$

Now, we form equivalence classes of integers with the help of elements of the following set:

$$S = \{6n, 6n + 1, 6n + 2, 6n + 3, 6n + 4, 6n + 5\}, \text{ where } n \in \mathbb{Z}.$$

It is straightforward to check that $Z = [6n] \cup [6n + 1] \cup [6n + 2] \cup [6n + 3] \cup [6n + 4] \cup [6n + 5]$.

This means that Z is equal to the disjoint union of the equivalence classes of elements of S . Therefore, S gives us the set of all integers. Out of the S , the elements given by the equivalence classes $[6n]$, $[6n+2]$, $[6n+3]$ and $[6n+4]$ are composite numbers. Then the possible choice for a number (except 2 and 3) to be a prime number is that the number is in $[6n + 1]$ or $[6n + 5]$.

It is then easy to check that an element $x \in [6n + 1]$ if and only if $x \equiv 1 \pmod{6}$.

Similarly, $x \in [6n + 5]$ if and only if $x \equiv 5 \pmod{6}$.

Thus, in view of (2.2.1) and (2.2.2), our assertion is proved.

3. Illustrations

3.1 Example

Let us check whether 280 is prime or not. The nearest multiple of 90 to 280 is 270. Therefore $x - n90 - 5 = 5$, and 5 is not exactly divisible by 6.

Now checking the other possibility, i.e., $x - n90 - 7 = 3$, and 3 is not divisible by 6. Thus, we conclude that 280 is not a prime number.

3.2 Example

Let us check whether 397 is prime or not. The nearest multiple of 90 to 397 is 360. Therefore $x - n90 - 5 = 32$, and 32 is not exactly divisible by 6.

Now checking the other possibility, i.e., $x - n90 - 7 = 30$, and 30 is divisible by 6. Thus, we conclude that 397 is a strong probable prime number, which in this case is a prime number.

3.3 Remark.

It is to be noted that one can also check these two examples with the Matlab code given in Appendix 1.

4. Connection between the Collatz Conjecture and Prime numbers

The Collatz conjecture is named after a mathematician Lothar Collatz who wrote the conjecture in 1937. It is also known as the $3n + 1$ conjecture.

The Collatz conjecture states the following

Consider a number x , if the number x is even, divide it by 2. If the number x is odd, triple it and add one.

Further, it states that regardless of the choice of x , after some iterations of the Collatz conjecture, the number x reaches 1.

Before proceeding further, we set a name for a number whence applied the Collatz conjecture on it.

4.1 Definition [see 8 and 9]

Let x be a number, then we say y as a Collatz number of x defined as:

- If x is even, then $y = x/2$
- If x is odd, then $y = 3x + 1$.

Now using the Collatz conjecture and the previous theorem, we have the following theorem.

4.2 Theorem

Let x be an even number and x is not a multiple of 6, then every prime number except 3 is a Collatz number of some x .

4.3 Proof

To prove this theorem, we construct a three column structure of numbers which will help us in predicting a strong candidate for prime numbers. The three column structure is explained in Figure 1.

Let x be a given number and $n90$ be the multiple of 90 nearest to x .

Now if $x - n90 - 7$ or $x - n90 - 5$ is divisible by 6, then we know from the previous theorem that x is a strong

candidate for prime number.

In these two cases, we will allocate x to column 1 (C1) and column 2 (C2), respectively.

Else if neither of these two things happen, then from the previous theorem, we again know that x is not a prime number and we allocate x to column 3 (C3).

From the three column structure in Figure 1 below, we observe all the primes except 3 lie in C1 or C2 only.

Thus the number arrangement in our three column set-up is as the following:

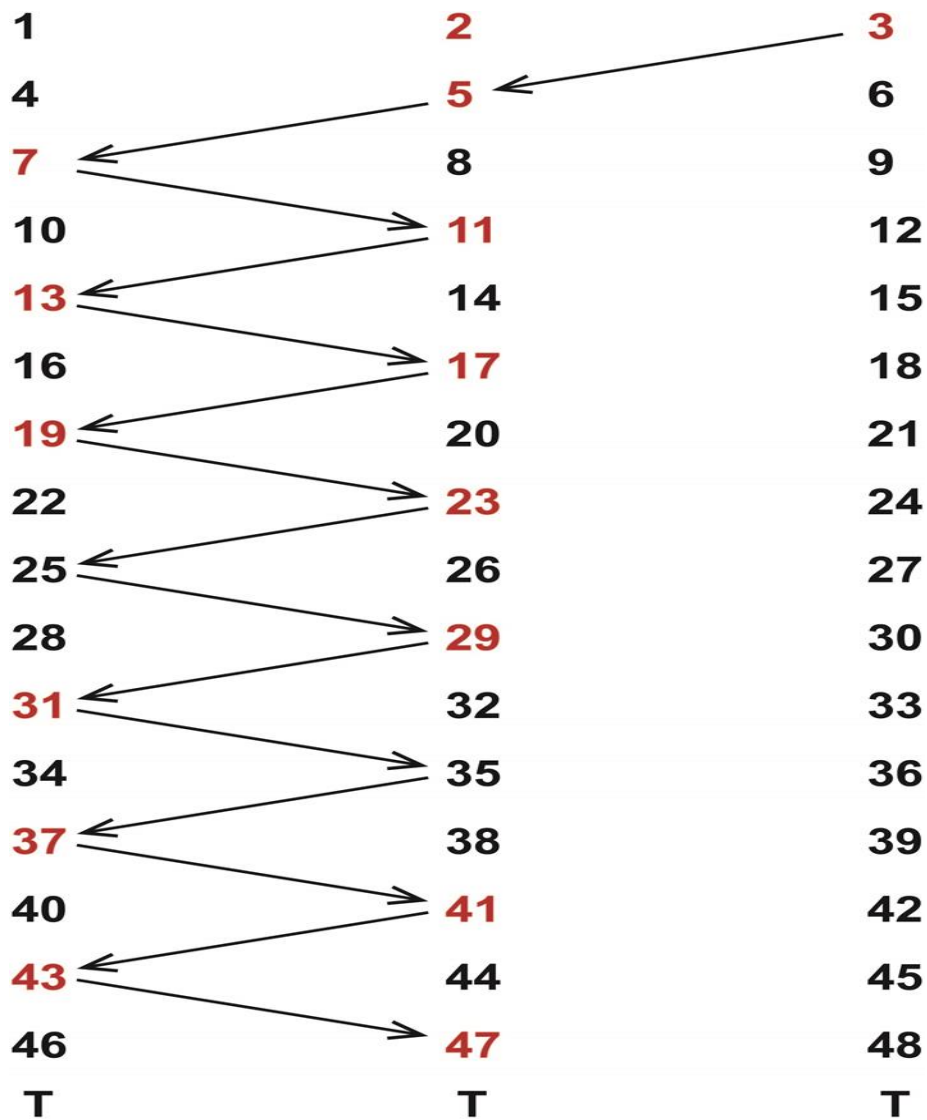


Figure 1: Three Column Structure.

where **T** is the transpose of the array.

Case studies.

Now let's consider the Collatz number of a given number.

$$Y = x/2 \text{ if } x \text{ is even}$$

and

$$Y = 3x + 1 \text{ if } x \text{ is odd.} \tag{4.1}$$

We will discuss the possibility of odd x 's in two cases:

Case 1

If x is odd, then in the Collatz number y of x , we have two operations, magnification by 3 and then translation by one. Thus we observe that every odd x is firstly projected to $C3$ by the multiplication of 3, but when translated by unity, it is projected back to $C1$.

Now, we again look at the operation on odd x .

The number 3 is odd, x is odd, and so $3x$ is odd. When we add up one to $3x$, the number $3x + 1 = y \in C1$ is always even. Thus $3x + 1$ can never be a possible prime number.

Case 2

If x is even, then the Collatz number $y = x/2$ may project to any of the three columns $C1$, $C2$ or $C3$. If y goes to $C1$ or $C2$, then y is a strong candidate for a prime number. If y is in $C3$, then y is never a prime number. So we exclude the choices for y to sit in $C3$.

It is straightforward to see that the numbers of $C3$ are the multiples of 3. In other words, we can say that the even x should not be a multiple of 6. On excluding the even numbers which are multiples of 6, the corresponding Collatz numbers is a strong prime candidate. This proves the result.

From the above Theorem, we can easily state the following.

4.4 .Corollary

Every prime number greater than 30 can be obtained from the sequence of numbers:

$n30 + 1 + 6 + 4 + 2 + 4 + 2 + 4 + 6$, where $n30$ is the nearest multiple of 30 less than the given prime and is a whole number.

4.5. Remark

From the above Corollary, we observe that the sequence of numbers lie in Column 1 and Column 2 only, so they

are always strong prime candidates, and at the same time this sequence is excluding a lot of non-prime numbers.

4.6. Remark

Let us start with few new notations.

If x is an even or an odd number, we say x as a weak Collatz even (or odd) number.

If the first iteration of the Collatz conjecture is odd, for example, let $x = 70$, then the first iteration in the Collatz conjecture is 35. Thus we call 70 a weak Collatz even number.

Otherwise, if x has more than one even iterations, we say x is a strong Collatz even number. For example 48 is a strong Collatz even number having multiple iterations $\{48, 24, 12, 6, 3, \dots\}$.

Similarly, in case of prime numbers, we call a prime number as a weak Collatz prime number if its first Collatz iteration is odd. Otherwise, we call it by strong Collatz prime number. From the above discussions, we observe to reach the final $\{4, 2, 1\}$ loop, the hailstorm numbers are eventually projected from a strong Collatz prime number to a strong Collatz even number and then finally to $\{4, 2, 1\}$ loop.

5. Conclusion

Users of Theorem 2.1 will successfully achieve to bring about a desired result enabling them to identify candidate prime numbers for uses in algorithms, computing and mathematics.

We have written a matlab code for Theorem 2.1 given in appendix 1. This will help the reader to easily check whether a number is a probable prime number or not. This theorem also contributes to determine the predictable nature of prime numbers explained in the 3 column structure (Figure 1), and will hopefully make a contribution on improving prime numbers understanding as building blocks for future research and development in number theory and computing.

We have also obtained a beautiful connection of prime numbers with the very famous Collatz conjecture with Theorem 4.2. (See matlab code in appendix 2) Corollary 4.4 serves as a more condensed look at the 3 column structure and an attempt to a building block for further improvement to formulate a formula for prime numbers. (See matlab code in appendix 3) The observations from this research through the 3 column structure has clearly revealed a pattern for the research on prime numbers and their predicted behavior.

Theorem 2.1

“Every prime number other than 2 and 3 satisfies one of the following two algebraic expressions $x - n90 - 5 = 6m$ or $x - n90 - 7 = 6m$ where $n90$ is the nearest multiple of 90 and m is a whole number.”

4.2. Theorem

“Let x be an even number and x is not a multiple of 6, then every prime number except 3 is a Collatz number of some x .”

4.4. Corollary

“Every prime number greater than 30 can be obtained from the sequence of numbers: $n30 + 1 + 6 + 4 + 2 + 4 + 2 + 4 + 6$, where $n30$ is the nearest multiple of 30 less than the given prime and is a whole number.”

6. Conflict of interest statement

The author declares that there is no conflict of interest in this paper. The author declare that there are no other financial relationships with any organizations that might have an interest in the submitted work.

References

- [1] M. Dorsey and S. Zachary. “Methods of primality testing.” *SMIT Undergraduate Journal of Mathematics*, vol (1), 133-141, 1999.
- [2] S. Ishmukhametov and B. Mubarakov. “On practical aspects of the Miller-Rabin primality test.” *Lobachevskii Journal of Mathematics*, vol. (34)4, pp.304-312, 2013.
- [3] Rosenberg and Burt. “The Solovay-Strassen Primality Test.” University of Miami. Department of Computer Science. Miami, 1993.
- [4] M. Agrawal; N. Kayal; N Saxena. "PRIMES is in P" (PDF). *Annals of Mathematics*. vol. 160 (2), pp. 781–793, 2004.
- [5] Cohen, Henri; Lenstra, W, Hendrik., Jr. "Primality testing and Jacobi sums". *Mathematics of Computation*. vol. 42, (165), pp. 297–330, 1984.
- [6] L.K. Hua. “Introduction to number theory”. Springer Science & Business Media, 2012.
- [7] I.N. Herstein. “Topics in algebra”. John Wiley & Sons, 2006.
- [8] P. Honner. “The Simple Math Problem We Still Can’t Solve”. Internet: www.quantamagazine.org/html, Sep. 20, 2020 [Oct.10, 2020]
- [9] K. Hartnet. “Mathematician Proves Huge Result on ‘Dangerous’ Problem.” Internet: www.quantamagazine.org/html, Dec. 11, 2019 [May.30, 2020]

Appendix 1

Matlab code for Primality test Theorem 2.1.

```
clc

clear all

flag =0;

Input_No=input ( ' Enter the number ' );

if ( Input_No<=1 )

    fprintf ( 'Please enter a valid number ' );

elseif ( Input_No==2||Input_No==3 )

    fprintf( ' The entered number is a prime number ' );

else

    flag=0;

    Ninety_n1=floor ( Input_No/90 ) ;

    N1=Input_No - ( Ninety_n1*90 ) - 5;

    No1=N1 / 6;

    Ninety_n2=floor ( Input_No/90 ) ;

    N2=Input_No - ( Ninety_n2*90 ) -7;

    No2=N2 / 6 ;

    If (mod ( No1,1)==0)

        flag=1;

    elseif ( mod (No2,1)==0 )

        flag=1;
```

```
end

    if ( flag==1 )

fprintf ( '\n The given number is a candidate prime number' );

else

fprintf ( '\n The given number is ---Not--- a prime number ' );

    end

end
```

Appendix 2

Matlab code for Collatz conjecture Theorem 4.2.

```
clc

clear all

flag =0;

Input_No=input ( ' Enter a prime number greater than three ' );

If ( Input_No<=1 || Input_No==3)

    fprintf ( ' Please enter a valid prime number ' );

else

    answer = isprime ( Input_No ) ;

    if ( answer==1 )

        y=2*Input_No ;

        fprintf ( ' the number %d is collatz number of %d',

            Input_No,y ) ;

    else
```

```
fprintf ( ' the number is not prime ' );
```

```
end
```

```
end
```

Appendix 3

Matlab code for Corollary 4.4.

```
clc
```

```
clear all
```

```
flag=0;
```

```
Input No=input ( ' Enter a prime number greater than 30 ' );
```

```
if ( Input No <30 )
```

```
fprintf ( ' Please enter a prime number greater
```

```
than 30 ' );
```

```
else
```

```
answer = isprime ( Input No );
```

```
if ( answer==1)
```

```
R=[1 , 6 , 4 , 2 , 4 , 2 , 4 , 6 ] ;
```

```
a=Input No / 30 ;
```

```
t=floor ( a );
```

```
x1=30* t ;
```

```
s=x1 ;
```

```
f=0;
```

```
j=1;
```

```
for i=1:8

if (Input No==s)

f=1;

break;

else

s=s+R(1,i);

C(1,j)=R(1,i);

j=j+1;

if(i==8)

f=1;

end

end

end

[Row, Col]=size(C);

if(f==1)

fprintf('%d = %d + ( ', Input No, x1)

for k=1:Col

if(k==Col)

fprintf('%d)',C(k))

else

fprintf('%d+',C(k))

end

end
```

```
end
```

```
end
```

```
else
```

```
fprintf('Please enter a valid prime number
```

```
greater than 30');
```

```
end
```

```
end
```