

Hypotheses

# Sequences for Determination of Prime Numbers by Elimination of Composites

<sup>1</sup>Jonathan M. Dugas and <sup>2</sup>Brian M. O'Connor

<sup>1</sup>Department of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, USA

<sup>2</sup>Department of Mathematics, Tennessee Tech University, Cookeville, USA

## Article history

Received: 25-03-2017

Revised: 26-06-2017

Accepted: 14-07-2017

## Corresponding Author:

Jonathan M. Dugas  
Department of Electrical and  
Computer Engineering,  
Tennessee Tech University,  
Cookeville, USA  
Email: jmdugas42@students.tntech.edu

**Abstract:** In this study, sequences are used for direct computation of prime numbers. A single equation for generating all prime numbers with the exception of 2 and 3 and composite numbers that are not divisible by 2 or 3 is presented. A periodicity to the indices of the composites generated by that equation is determined. An equation to determine the indices of the composite numbers is derived. The equation for determining the composite indices is then altered to avoid redundancy by observation of its diagonal, when its values are inserted into a matrix that has the indexes ( $j$ ) for columns and ( $k$ ) for rows. The two equations are presented in several different forms and are used in conjunction to render a technique for computing prime numbers indefinitely. The validity of the technique is proven up to one billion by computation and the prime number theorem.

**Keywords:** Prime, Sequence, Generation, Composite, Elimination

## Introduction

A number of methods for determining prime numbers, such as the sieve of Eratosthenes and sieve of Atkins, have been discovered (Crandall and Pomerance, 2006). The equations shown in this study render a mathematical technique for generating primes, which is very similar to the sieve of Eratosthenes. There are also ways to approximate the density of prime numbers between 1 up to some number, ( $x$ ) (Goldstein, 1973). In 300 B.C., Euclid proved the existence of an infinite number of primes through fundamental arithmetic theorems. In 1737, Euler further proved the existence of an infinite number of primes. Euler showed the sum of the reciprocals of prime numbers is a divergent sum; thus, it contains infinitely many terms (prime reciprocals). In 1791, Gauss discovered that the density of primes is asymptotic. This is known as the prime number theorem (Goldstein, 1973). Further works from Gauss's assertion lead to the following:

$$\lim_{x \rightarrow \infty} \frac{\pi(x)}{\frac{x}{\ln x}} = 1$$

With  $\pi(x)$  representing the density of primes, the above equation implies that the density of prime numbers decreases as ( $x$ ) approaches infinity (Crandall and

Pomerance, 2006). The density of primes between 1 and some number ( $x$ ) can be approximated as follows:

$$\pi(x) \approx \frac{x}{\ln x}$$

The above equation provides a good method of approximating the density of primes for some value ( $x$ ), but an effective mathematical method for generating all prime numbers efficiently and indefinitely seems to have eluded mathematicians for centuries. In fact, many mathematicians and members of the scientific community believe that prime numbers are random and no such formula for generating prime numbers in this way exists. It is difficult and perhaps impossible to disprove such a statement mathematically. However, what if there exists an equation that can indefinitely generate prime numbers as well as some composite numbers and what if there is a periodicity to the composite numbers? An equation could then be written to eliminate the composite numbers. Then, in essence, there would exist a mathematical method for generating primes indefinitely, which avoids the need for finding a sequential order to the prime numbers. The purpose of this work is to present such a method, which is in agreement with the prime number theorem and to prove the validity of that method by computation.

## Materials and Methods

The flowchart in Fig. 1 is a simplified version of the C++ computer algorithm which was used to compute all of the prime numbers between 1 and 1,000,000,000, using a set of sequences. A Dell laptop with an i7 processor was used to run the algorithm. The algorithm used is represented mathematically as follows:

$$a(i) = 1 + \sum_{L=1}^i \left( 4 \left| \sin\left(\frac{\pi L}{2}\right) \right| + 2 \left| \cos\left(\frac{\pi L}{2}\right) \right| \right)$$

$$p = \left( 7 + \sum_{k=2}^n \left( 2 \left| \cos\left(\frac{\pi k}{2}\right) \right| + 6 \left| \sin\left(\frac{\pi k}{2}\right) \right| \right) u(n-2) \right)$$

$$q = (2n + 1)$$

$$i_{(not\ prime)} = n + \sum_{j=1}^m \left( p \left| \sin\left(\frac{\pi j}{2}\right) \right| + q \left| \cos\left(\frac{\pi j}{2}\right) \right| \right)$$

$$1 \leq i \leq N$$

$$1 \leq j \leq m$$

$$2 \leq k \leq n$$

The derivation process for the above equations will be described in detail in the Discussion section. The first step in the development of the computer program entailed writing code that asked the user for a number (N), so that the program knew the range of prime numbers to generate. If the user were to input 100, the program would return every prime number between 1

and 300 (3N). Once (N) was input, the program would calculate all of the values of the (a) function from a(1) to a(N) and store those values into a text file.

The next step involved calculating the indices of the composites generated by the (a) function. Essentially, this was accomplished with a nested for loop (Gaddis and Krupnow, 2007). All of the composite indices were then sorted so they made sense to the rest of the program. The approach used for the logic shown in Fig. 1 involved generating all of the composite indices, sorting them into smaller segments, storing those smaller segments into another text file and breaking the large text file up into smaller text files accordingly. The numbers stored in the smaller text files were used to determine the indices of the composites generated by the (a) function.

The final step was simply reading the results of the (a) function from the file in which they were stored. A counter was used to keep track of each integer as it was read from the file; if the counter equaled any of the composite indices, the corresponding (a) function's result was not output to the prime number file (Gaddis and Krupnow, 2007). Otherwise, the value read from the file was a prime number and was written to the prime number file. The logic described was a very simple program to write and it proved that the described prime generator method could be used with very primitive programming languages to efficiently determine prime numbers.

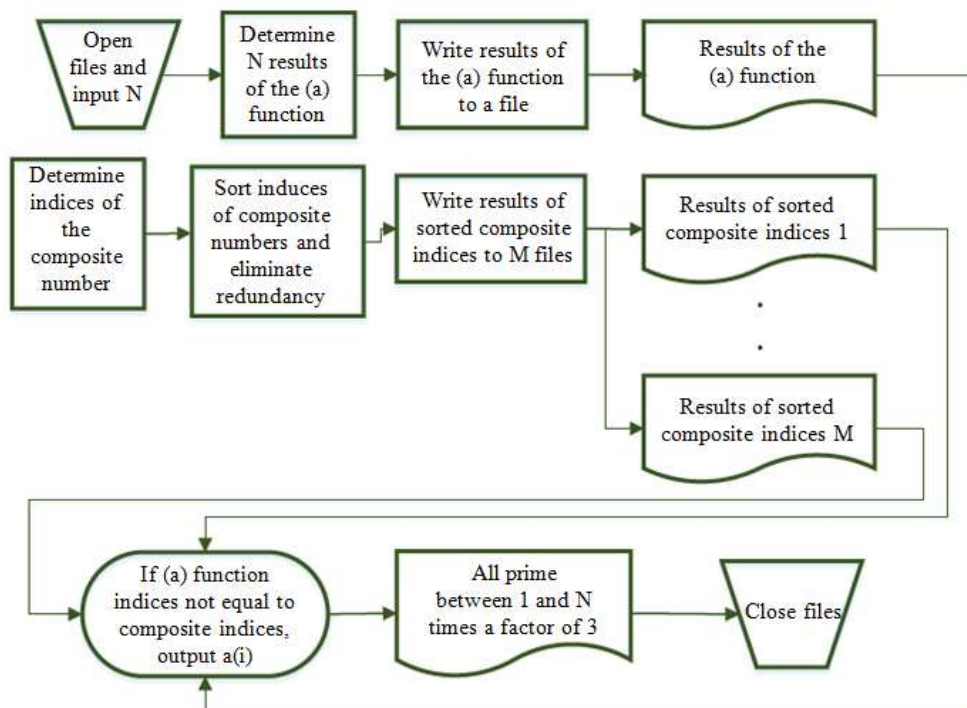


Fig. 1. The flow logic of the simple C++ program used to calculate every prime number between 1 and a billion

## Results

Observe Table 1. All of the primes generated by the described algorithm are not shown in Table 1, as there are over 50 million of them between 1 and a billion. Table 1 shows the last 50 prime numbers generated by the described algorithm as well as the count value associated with that prime number. Recall that an approximation of the density of prime numbers between 1 and a billion can be obtained as follows:

$$\left( \frac{1,000,000,000}{\ln(1,000,000,000)} \right) \approx 48,254,942.43$$

In the described experiment, 50,847,534 prime numbers were generated between 1 and a billion:

$$\pi(x) = 50,847,534$$

Using the prime number theorem, a value close to one should be obtained from the following:

$$\lim_{x \rightarrow 10^9} \frac{50847534}{(48,254,942.43)} = 1.053727$$

The result was a value very close to 1, which indicated a good approximation of the density of primes. The results were in good agreement with the prime number theorem.

In fact, it is well known that there are exactly 50,847,534 prime numbers between one and a billion and all of the integers stored in the prime number file were prime numbers. Therefore, the experiment was successful in proving the validity of the presented prime number generator. Also, it can be observed in Fig. 2 (Fig. 2 will be explained in the following section) that when the  $(a)$  function is iterated to higher values of  $(N)$ , more composites and fewer primes are generated by the  $(a)$  function. In other words, Equation 7 (Equation 7 will also be explained in the next section) eliminates more of the  $(a)$  function values as  $(N)$  increases because  $(n)$  (i.e., the number of steps in Fig. 2) must be increased proportionally with  $(N)$ . Again, the method is in good agreement with the prime number theorem because the density of primes decreases as  $(N)$  increases.

## Discussion

### Derivation of General Prime Number Sequences

The following function (Equation 1), which will simply be termed the  $(a)$  function, generates all prime numbers starting with 5 and it generates composites, which are not divisible by 2 or 3, periodically:

$$\begin{aligned} a(i) &= 6i - a(i-1) \\ a(0) &= 1 \\ 1 \leq i &\leq N \end{aligned} \tag{1}$$

Table 1. Last 50 prime numbers and corresponding count value

Count	Prime number	Count	Prime number
50847485	999998929	50847510	999999337
50847486	999998957	50847511	999999353
50847487	999998959	50847512	999999391
50847488	999998971	50847513	999999433
50847489	999998981	50847514	999999487
50847490	999999001	50847515	999999491
50847491	999999017	50847516	999999503
50847492	999999029	50847517	999999527
50847493	999999043	50847518	999999541
50847494	999999059	50847519	999999587
50847495	999999067	50847520	999999599
50847496	999999103	50847521	999999607
50847497	999999107	50847522	999999613
50847498	999999113	50847523	999999667
50847499	999999131	50847524	999999677
50847500	999999137	50847525	999999733
50847501	999999151	50847526	999999739
50847502	999999163	50847527	999999751
50847503	999999181	50847528	999999757
50847504	999999191	50847529	999999761
50847505	999999193	50847530	999999797
50847506	999999197	50847531	999999883
50847507	999999223	50847532	999999893
50847508	999999229	50847533	999999929
50847509	999999323	50847534	999999937

i	a(i)					
1	5					
2	7					
3	11					
4	13					
5	17					
6	19					
7	23					
8	25	7				
9	29					
10	31					
11	35	3	9			
12	37					
13	41					
14	43					
15	47					
16	49		5			
17	53					
18	55	7		15		
19	59					
20	61					
21	65	3			17	
22	67					
23	71					
24	73					
25	77		9	7		
26	79					
27	83					
28	85	7				23
29	89					
30	91		5		9	
31	95	3				25

Fig. 2. Results of the (a) function for (i) equals 1 to 31 with a staircase showing the periodicity of the composite indices

Equation 1 is a very simple function, which has an initial condition value of 1 (a nonprime number in the strictest sense of the definition of a prime). The index (i) is multiplied by 6 and the previous value of the (a) function is subtracted from that product. Equation 1 generates all of the prime numbers starting at 5 up to as high a number as desired. In fact by plugging all integers from 1 up to some number (N) into the (a) function, all

of the prime numbers from 5 up to (N) times a factor of 3 will be generated.

To understand why Equation 1 determines the primes from 5 up to (N) times a factor of 3, observe the following:

$$\lim_{i \rightarrow N} a(i-1) \approx \frac{1}{2} \cdot (6N)$$

By inspection, it can be seen that a(i-1) is approximately half of (6i), resulting in the following:

$$\lim_{i \rightarrow N} a(i) \approx \lim_{i \rightarrow N} (6i - \frac{1}{2} \cdot (6i)) \approx \frac{1}{2} \cdot (6N)$$

By calculating the results of Equation 1 and letting (i) encompass all integers from 1 to (N), every prime number between 5 and approximately (3N), as well as composites not divisible by 2 or 3, is generated. The (a) function can also be written in terms of the absolute values of the sine and cosine functions, which will be explained later.

As mentioned previously, the (a) function generates composites periodically. To determine the integers generated by the (a) function which are composites, the periodicity of those composites must first be determined. The simplest way to determine the periodicity of the composite numbers is to write out the results of the (a) function up to some number (N). By inspection, the periodicity of the composites generated by the (a) function can then easily be seen. For example, let (N) equal 31 and plug in all integers from 1 to (N) into the (a) function as shown in Fig. 2. Figure 2 displays the indices of the (a) function, (i) and the corresponding results of the (a) function, a(i). The staircase to the right in Fig. 2 is used to show the periodicity of the composite numbers.

For instance, start at 5 (top step, blue) and count down 7 spaces to arrive at a composite number divisible by 5. Then, count down 3 spaces to arrive at another composite divisible by 5. Continue counting down 7 and then 3 to determine the composite numbers divisible by 5, which the (a) function generates.

To eliminate all numbers divisible by 7, simply move down the staircase to 7 (second step, yellow). From 7, count down 9 spaces to arrive at a composite number divisible by 7. Then, count down 5 to arrive at another composite divisible by 7. Continue counting down 9 and then 5 to determine composites divisible by 7. To eliminate all numbers divisible by 11, move down the staircase to 11 (third step, green). From 11, count down 15 spaces to arrive at a composite number divisible by 11. Then, count down 7 to arrive at another number divisible by 11. Continue counting down 15 and then 7 to determine composites divisible by 11. Likewise, from the fourth step, count down 17 and then 9 repeatedly to eliminate all numbers divisible by 13.

Table 2. Explanation of the observed periodicity of the composite indices reflected in Figure 2

Step (n)	For odd iterations of (j)	For even iterations of (j)	Eliminate integers divisible by:
1	add 7	add 3	5
2	add 9	add 5	7
3	add 15	add 7	11
4	add 17	add 9	13
5	add 23	add 11	17
6	add 25	add 13	19

From the fifth step, count down 23 and then 11 repeatedly to eliminate all numbers divisible by 17. From the sixth step, count down 25 and then 13 repeatedly to eliminate all numbers divisible by 19. Table 2 clarifies the process further.

A step counter exists, which will be represented by the variable (n). Begin by counting down 7 and then 3 repeatedly. Then, when moving down to an even step, add 2 to the first number counted down (i.e., 7+2 = 9); when moving down to an odd step, add 6 to the first number counted down (i.e., 9+6 = 15). The second number counted down is always the next highest odd number (i.e., 3, 5, 7,...). Using the pattern observed in Fig. 2 and again in Table 2, an equation to determine the indices of all of the composite numbers generated by the (a) function can be derived.

To write the equation for eliminating all of the composite numbers, an equation for the initial periodic sequence must first be determined. The absolute value of the sine and cosine functions at integral multiples of  $\pi/2$  oscillate between 0 and 1 and they can be used such that 7 is multiplied by the sine function and 3 is multiplied by the cosine function, to yield an equation for summing 7 on odd values of (j) and 3 on even values of (j):

$$i_{(not\ prime)} = 1 + \sum_{j=1}^m \left( 7 \left| \sin \left( \frac{\pi j}{2} \right) \right| + 3 \left| \cos \left( \frac{\pi j}{2} \right) \right| \right) \quad (2)$$

$1 \leq i \leq N$   
 $1 \leq j \leq m$

The not prime subscript of (i) notation in Equation 2 denotes that these indices correspond to composite numbers generated by the (a) function. The (n) in Equation 2 accounts for the step number on the staircase in Fig. 2. With  $n = 1$  and as (j) is iterated, Equation 2 determines the indices of the integers divisible by 5. However, an equation that determines the composites divisible by: 5, 7, 11, 13, 17, 19, 23 and so forth is desired. For odd iterations of (j), let us define an equation for adding 2 to the 7 when starting the count on even steps and 6 when starting the count on odd steps and let us call this Equation (p):

$$p = \left( 7 + \sum_{k=2}^n \left( 2 \left| \cos \left( \frac{\pi k}{2} \right) \right| + 6 \left| \sin \left( \frac{\pi k}{2} \right) \right| \right) u(n-2) \right) \quad (3)$$

$2 \leq n$

Again, the cosine function is used to add 2 when counting down from even steps and the sine function is used to add 6 when counting down from odd steps. Since the only concern is with positive values of sine and cosine, the absolute value of the sine and cosine functions is taken similar to Equation 2. The unit step function,  $u(n-2)$ , is used to ensure that the summation is only initialized after moving down to the second step (i.e.,  $n = 2$ ) to start determining integers divisible by 7 and the summation sums from 2 to (n) (Zill and Cullen, 2008). The upper limits of the summations, (n) and (m), are arbitrary and should be chosen such that (i) does not exceed (N). Now to find the next highest odd number for every even value of (j), the following simple equation is used, which is commonly used in mathematics:

$$q = (2n + 1) \quad (4)$$

Equation 4 is an odd number generator. Equation 2 thru 4 are combined to produce the following:

$$i_{(not\ prime)} = n + \sum_{j=1}^m \left( p \left| \sin \left( \frac{\pi j}{2} \right) \right| + q \left| \cos \left( \frac{\pi j}{2} \right) \right| \right) \quad (5)$$

$1 \leq i \leq N$   
 $1 \leq j \leq m$   
 $2 \leq k \leq n$   
 $1 \leq n$

Notice that the absolute value of the sine function is multiplied by (p) and the absolute value of the cosine function is multiplied by (q). (p) is as it is in Equation 3. For odd values of (j), (p) is added to (n) and for even values of (j), (q) is added to (n).

Equation 5 is a general form for an equation that determines the indices of the composite numbers generated by the (a) function. In essence, using the (a) function in conjunction with Equation 5 is a method for determining prime numbers indefinitely but does

not include the primes 2 and 3. However, a major drawback of Equation 5 is redundancy. There are many occasions when the indices of the composites are determined by Equation 5 more than once, which can slow processing significantly. Observe Fig. 3, which shows the values of  $(i)$  in Equation 5 as  $(j)$  and  $(k)$  are iterated. Whenever any of the vertices seen in Fig. 3 meet, redundancy occurs. There is a way to remove most

of the redundancy from Equation 5, which is observable when the values of Equation 5 are arranged in a matrix that has the indexes  $(j)$  for columns and  $(k)$  for rows. Observe at Table 3.

Table 3 is a matrix containing the values of Equation 5 as  $(j)$  and  $(k)$  are iterated. The values of the diagonal and everything in green above the diagonal are the only results needed from Equation 5.

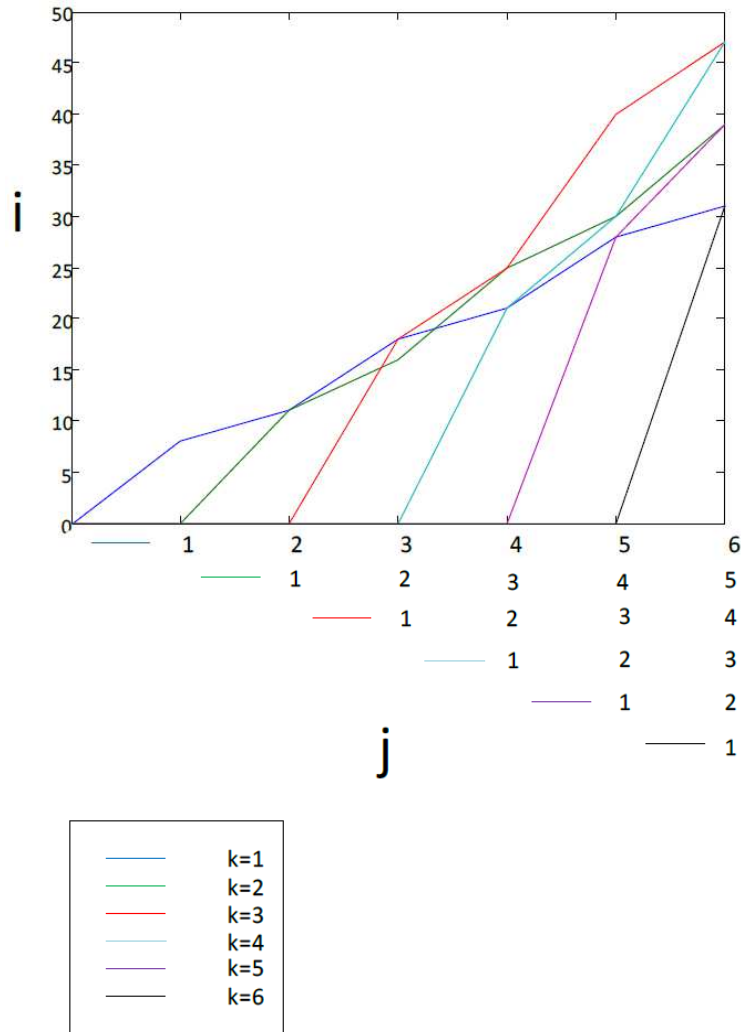


Fig. 3. Graphs of Equation 5 as  $(j)$  and  $(k)$  are iterated

Table 3. Matrix containing  $(j)$  columns and  $(k)$  rows and shows how a large portion of the redundancy in Equation 5 can be removed by observation of the matrix's diagonal

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
$k = 1$	8	11	18	21	28	31
$k = 2$	11	16	25	30	39	44
$k = 3$	18	25	40	47	62	69
$k = 4$	21	30	47	56	73	82
$k = 5$	28	39	62	73	96	107
$k = 6$	31	44	69	82	107	120

■ Composite Indices; ■ Diagonal; ■ Redundancy

All values in red are redundant and can be omitted, as the values in the first column are identical to the values in the first row. The same is true for all elements along the entire diagonal (hence, everything below the diagonal is redundant). There is a periodicity to the diagonal elements similar to previously derived sequences, which can be written as follows:

$$Diagonal = D = \sum_{k=1}^n \left( 4k \left| \cos\left(\frac{\pi k}{2}\right) \right| + 8k \left| \sin\left(\frac{\pi k}{2}\right) \right| \right) \quad (6)$$

Equation 6 can be used to modify Equation 5 to eliminate most redundancy as follows:

$$i_{(not\ prime)} = D + \sum_{j=2}^m \left( p \left| \sin\left(\frac{\pi j}{2}\right) \right| + q \left| \cos\left(\frac{\pi j}{2}\right) \right| \right) u(n+1-j) \quad (7)$$

First, Equation 6 is added to Equation 5 and the step counter ( $n$ ) in Equation 5 is negated. Equation 6 is a function of ( $k$ ) and is used to determine the initial value of Equation 7 for each value of ( $n$ ). The  $u(n+1-j)$  term in Equation 7 is used to properly initialize the sequence associated with Equation 5, which was previously derived. In other words, the unit step function is used to ensure that ( $j$ ) is properly iterated for a given value of ( $n$ ). For example, when ( $n$ ) is equal to 1, ( $j$ ) begins its iterations at 2; When ( $n$ ) is equal to 2, ( $j$ ) begins its iterations at 3. Equation 7 eliminates most of the redundancy in Equation 5 making the algorithm significantly more efficient.

### Derivation of Flexibility Equations

As previously stated, the ( $a$ ) function can be written in terms of the absolute values of the sine and cosine functions (Mandal and Asif, 2007). Observe Fig. 2. Notice that the values of the ( $a$ ) function increase periodically by 4 and then 2. Similar to Equations 5 thru 7, the ( $a$ ) function can be written in terms of the absolute values of the sine and cosine functions as follows:

$$a(i) = 1 + \sum_{L=1}^i \left( 4 \left| \sin\frac{\pi L}{2} \right| + 2 \left| \cos\frac{\pi L}{2} \right| \right) \quad (8)$$

Writing the ( $a$ ) function as shown in Equation 8 eliminates the need to keep track of the ( $a$ ) function's previous values, which simplifies computer programming and helps preserve computer resources. From a signals and systems point of view, computer resources are preserved because the ( $a$ ) function is memoryless when written as shown in Equation 8 (Mandal and Asif, 2007). A final form of the prime number generator, which was programmed in C++ to prove its validity, is:

$$a(i) = 1 + \sum_{L=1}^i \left( 4 \left| \sin\left(\frac{\pi L}{2}\right) \right| + 2 \left| \cos\left(\frac{\pi L}{2}\right) \right| \right)$$

$$i_{(not\ prime)} = \sum_{k=1}^n \left( 4k \left| \cos\left(\frac{\pi k}{2}\right) \right| + 8k \left| \sin\left(\frac{\pi k}{2}\right) \right| \right)$$

$$+ \sum_{j=2}^m \left[ \left( \left( 7 + \sum_{k=2}^n \left( 2 \left| \cos\left(\frac{\pi k}{2}\right) \right| + 6 \left| \sin\left(\frac{\pi k}{2}\right) \right| \right) \right) u(n-2) \left| \sin\left(\frac{\pi j}{2}\right) \right| \right. \\ \left. + (2n+1) \left| \cos\left(\frac{\pi j}{2}\right) \right| \right) u(n+1-j)$$

$1 \leq i \leq N$   
 $2 \leq j \leq n$   
 $1 \leq k \leq n$

Provided the set of equations holds true for all of ( $N$ ), then prime numbers are not random because of their connection to composites not divisible by 2 or 3. The described experiment proves the algorithm works to a billion and it seems very likely to hold true for all ( $N$ ). Supercomputers with sufficient resources could possibly be used to prove the validity of the described method to much higher values of ( $N$ ). It is also worth noting the existence of other methods to write the set of sequences without using the absolute values of sine and cosine. Observe the following two equations:

$$\left| \sin\left(\frac{\pi i}{2}\right) \right| = \left( \frac{1 + (-1)^{i+1}}{2} \right) = \begin{cases} 1, & \text{for odd values of } i \\ 0, & \text{for even values of } i \end{cases}$$

$$\left| \cos\left(\frac{\pi i}{2}\right) \right| = \left( \frac{1 + (-1)^i}{2} \right) = \begin{cases} 0, & \text{for odd values of } i \\ 1, & \text{for even values of } i \end{cases}$$

The above two equations can be used in place of the absolute value of the sine and cosine functions; the ( $a$ ) function and Equation 7 then become:

$$a(i) = 1 + \sum_{L=1}^i \left( 4 \left( \frac{1 + (-1)^{L+1}}{2} \right) + 2 \left( \frac{1 + (-1)^L}{2} \right) \right) \quad (9)$$

$$i_{(not\ prime)} = \sum_{k=1}^n \left( 4k \left( \frac{1 + (-1)^k}{2} \right) + 8k \left( \frac{1 + (-1)^{k+1}}{2} \right) \right)$$

$$+ \sum_{j=2}^m \left[ \left( \left( 7 + \sum_{k=2}^n \left( 2 \left( \frac{1 + (-1)^k}{2} \right) + 6 \left( \frac{1 + (-1)^{k+1}}{2} \right) \right) \right) u(n-2) \left( \frac{1 + (-1)^j}{2} \right) \right. \right. \\ \left. \left. + (2n+1) \left( \frac{1 + (-1)^j}{2} \right) \right) u(n+1-j) \right] \quad (10)$$

$1 \leq i \leq N$   
 $1 \leq j \leq m$   
 $2 \leq k \leq n$   
 $1 \leq n$

Table 4. Alternate method for determining the composite indices by periodically adding and subtracting even numbers to the results of the (a) function

a(i) ±	Even number	=	Addition result	Subtraction result
5 ±	2	=	7	3
7 ±	2	=	9	5
11 ±	4	=	15	7
13 ±	4	=	17	9
17 ±	6	=	23	11
19 ±	6	=	25	13
23 ±	8	=	31	15
25 ±	8	=	33	17
29 ±	10	=	39	19
31 ±	10	=	41	21
35 ±	12	=	47	23
37 ±	12	=	49	25
41 ±	14	=	55	27
43 ±	14	=	57	29

Equation 7 can also be written another way, which is significantly different than its previously derived form. Observe Table 4. By periodically adding and subtracting 2 to a(1), 7 and 3 can be periodically generated to determine the indices of the composites divisible by 5. By periodically adding and subtracting 2 to a(2), 9 and 5 can be periodically generated to determine the indices of composites divisible by 7. Instead of adding and subtracting 2 to a(3) and a(4), add and subtract 4 to those values to first generate 15, then 7 and 17 and then 9, respectively. The technique is obvious by observation of Table 4. Start by adding and subtracting 2 to the (a) function values for (i) equal to 1 and 2. For odd values of (i), increase the number added and subtracted to a(i) to the next even number. Thus, the sequence shown in Table 4 computes the same results as Equation 7 and can be represented as:

$$i_{(not\ prime)} = \sum_{j=2}^m \left[ \sum_{k=1}^n \left( 4k \left| \cos\left(\frac{\pi k}{2}\right) \right| + 8k \left| \sin\left(\frac{\pi k}{2}\right) \right| \right) + \left[ a(n) + (-1)^{j+1} \sum_{k=1}^n (1 + (-1)^{k+1}) \right] u(n+1-j) \right] \quad (11)$$

Alternatively, the sequence in Table 4 can be written as:

$$i_{(not\ prime)} = \sum_{j=2}^m \left[ \sum_{k=1}^n \left( 4k \left| \cos\left(\frac{\pi k}{2}\right) \right| + 8k \left| \sin\left(\frac{\pi k}{2}\right) \right| \right) + \left[ a(n) + (-1)^{j+1} \sum_{k=1}^n 2u \left( 1 - \left| \sin\left(\frac{\pi k}{2}\right) \right| \right) \right] u(n+1-j) \right] \quad (12)$$

The alternate version using the powers of (-1) in place of the absolute value of the sine and cosine functions could also be used for Equations 11 thru 12. Several forms of the (a) function and Equation 7 have now been presented. Having several forms of the same equations could provide some flexibility when

programming the overall algorithm. When writing computer algorithms that are intended to perform billions of calculations, flexibility with the implementation of the algorithms can help preserve computer resources. The use of more advanced programming languages and sophisticated computer systems, such as supercomputers, could significantly exceed the results presented in this study.

Given that the method holds true for all (N), the described prime number generator could possibly be used to break public encryption keys, which are based on the products of higher order prime numbers. In fact, the (a) function and Equation 7 could be used to determine the products of higher order prime numbers. The composites generated by the (a) function are nothing more than the products of prime numbers; therefore, Equation 7 could easily be used to pick which results of the (a) function match the higher order prime number products.

### Conclusion

The results of this work validate the notion that prime numbers are not random. The sequential order to the prime numbers is determinable by a connection to composite numbers which are not divisible by 2 or 3. Although it is difficult to prove by a method other than computation, the sequences presented generate prime numbers indefinitely; however, they do generate composite numbers not divisible by 2 or 3. There is a periodicity to the composite numbers, which allows them to be eliminated using basic mathematical functions and operations. Given the presented sets of sequences for determining primes and eliminating composites hold true for all values of (N), one must first understand the sequence to the composites not divisible by 2 or 3 to understand the sequence to the prime numbers; the sequence to primes is disguised as a sequence to composites.

### Acknowledgement

I acknowledge that Leonhard Euler was not wrong; primes are “a mystery into which the mind will never penetrate.” Thus, computers are needed to perform the very tedious mathematics. Thanks to my father and mother, Roger and Rhonda Dugas, for all of their love and encouragement. Also, thanks to Dr. Satish Mahajan, a professor in the Electrical and Computer Engineering Department at Tennessee Tech University, for providing me with a great opportunity to attend Graduate studies at Tennessee Tech. I also acknowledge that mathematicians before me have noticed very similar patterns to prime numbers reported in this study. However, in doing a literature review, I was unable to find any equations exactly like the ones presented.



## Funding Information

My father, Roger M. Dugas, purchased a Dell laptop, with an Intel i7 processor for me, which I used to write the described algorithm.

## Author's Contributions

The primary author, Jonathan M. Dugas, discovered the  $(a)$  function and the sequences related to Equations 1 thru 10, derived all of the equations and composed this paper. Dr. O'Connor, a professor in the Mathematics Department at Tennessee Tech University and the secondary author, found the fascinating connection between the  $(a)$  function and the sequence shown in Table 4, allowing this work to be much more interesting and useful to cryptography. As a result, Dr. O'Connor helped to validate the authenticity of this work.

## Ethics

The information presented in this study can possibly be used to break public encryption keys. I encourage and hope care is taken with its implementation. The order that exists within the universe is amazing and fascinating; there is even an order to the prime numbers. I only hope it is used so that others, like myself, can learn new universal truths and make the world a better and safer place to live.

## References

- Crandall, R. and C. Pomerance, 2006. Prime Numbers: A Computational Perspective. 2nd Edn., Springer Science and Business Media Inc., New York, ISBN-10: 0387289798, pp: 597.
- Goldstein, L., 1973. A history of the prime number theorem. Am. Math. Monthly, 80: 599-615. DOI: 10.2307/2319162
- Gaddis, T. and B. Krupnow, 2007. Starting out with C++: From Control Structures through Objects. 5th Edn., Pearson/Addison Wesley, Boston, ISBN-10: 032147970X, pp: 904.
- Mandal, M. and A. Asif, 2007. Continuous and Discrete Time Signals and Systems. 1st Edn., Cambridge University Press, New York, ISBN-10: 052172841X, pp: 880.
- Zill, D. and M. Cullen, 2008. Differential Equations with Boundary-Value Problems. 7th Edn., Brooks/Cole, Cengage Learning, Belmont, CA, ISBN-10: 0495108367, pp: 608.