Pseudo Randomness in Prime Numbers

By Ricardo Gil

Ricardo.gil@sbcglobal.net

03/20/2016

Abstract

The purpose of this paper is to show that prime numbers are structured in a Pseudo Random manner. Like the Fibonacci or the Lucas sequence, the prime number sequence is a sequence in which 2 primes when added together (+ or -1) makes the next prime. The sum of the two primes, A+B(+or-1)=C dictates the next prime number in the sequence. Goldbach's conjecture is that every even integer is the sum of two primes, A+B(+or-1)=C is two primes +or-1 make up another prime and dictates the gap between the primes. Progressing along the prime number line is similar to the Fibonacci sequence and the Lucas sequence. In a sense the A+B(+or-1)=C is a sequence but for prime numbers. In the Pseudo Random Prime Number Sequence or A+B(+or -1)=c, 5+3-1=7 and 7+5-1=11. The "A" side progresses or dictates the progression and in the progression or sequence if 5 were used in 5+5+1=11 instead of 7+5-11 it would be out of order in the progression sequence.

I. Observations;

A).If one takes a sequence stating with one that progresses along the prime number line, one will get a pattern of a prime number plus another prime number(+ or – 1) that equal or generates another prime number.

II. Given A+B(+or-1)=C

A). There are no skips in sequential order in the prime numbers used to build A+B(+or-1)=C, on the (A) side.

B). There are skips in sequential order in the primes numbers used to build B(+or-1)=C, on (B) side.

III. In the Pseudo Random prime number sequence ;

A).Given A+B(+or-1)=C

1).(A) side stays or progresses in sequence. (1,2,3,5,7…113)

2).(B) side skips a prime number in sequence.(6,6,6,8……..infinity).

Pseudo Random Composition of Prime Numbers
A (prime#)+ B(prime#) (+ or – 1)= C (prime#)

A +B(+/-1) = C

1+1=2
2+1=3
3+2=5
5+3-1=7
7+5-1=11
7+5+1=13 (B skip gap 11-5= 6)
7+11-1=17
7+13-1=19
11+13-1=23 (Gap between skip gap 23-11=8)
11+19-1=29
13+17+1=31
19+17+1=37
19+23-1=41 (B skip gap 23-17= 6)
19+23+1=43
19+29-1=47
23+31-1=53 (Gap between skip gap 37-23=14)
23+37-1=59
23+37+1=61
23+43+1=67 (B Gap skip gap 43-37=6)
23+47+1=71
29+43+1+73
31+47+1=79
37+47-1=83
37+53-1=89
43+53+1=97
47+53+1=101 (Gap between skip gap 71-43=28)
43+59+1=103
47+59+1=107
47+61+1=109
47+67-1=113
59+67+1=127
61+71-1=131
67+71-1=137
67+71+1=139 (B skip gap 79-71= 8)
71+79-1=149
71+79+1=151
73+83+1=157
79+83+1=163
79+89-1=167
89+83-1=173
97+83-1=179
97+83+1=181
101+89+1=191
103+89+1=193
107+89+1=197
109+89+1=199
113+97+1=211

Pseudo Random Prime Number Sequence

Composition of Prime Numbers
A (prime#)+ B(prime#) (+ or – 1)= C (prime#)

____(X)-axis->

(Y)						
a	A	+	B	(+or -1)	=	C
x	(1)	+	(1)		=	(2)
i	(2)	+	(1)		=	(3)
s	(3)	+	(2)		=	(5)
*	(5)	+	(3)	(+or-1)	=	(7)
*	(7)	+	(5)	(+or-1)	=	(11)
*	(7)	+	(5)	(+or-1)	=	(13)
			(Gap 11-5 = 6)			
*	(7)	+	(11)	(+or-1)	=	(17)
*	(7)	+	(13)	(+or-1)	=	(19)
*	(11)	+	(13)	(+or-1)	=	(23)
*	No skip in progression on (A) side.					
*	(19)	+	(17)	(+or-1)	=	(37)
			(Gap 23-17 = 6)			
*	(19)	+	(23)	(+or-1)	=	(41)
*	(23)	+	(37)	(+or-1)	=	(61)
			(Gap 43-37 = 6)			
*	(23)	+	(43)	(+or-1)	=	(67)
*	No skip in progression on (A) side.					
*	(67)	+	(71)	(+or-1)	=	(139)
			(Gap 79-71 = 8)			
*	(71)	+	(79)	(+or-1)	=	(149)
*	(101)	+	(89)	(+or-1)	=	(191)

The (A) side does not skip a number in the sequence although it repeats. The gap or skip pattern appears in (B) side of sequence (6,6,6,8). The sum of the two primes, A+B(+or-1)=C dictates the next prime number in the sequence.

Pseudo Random Prime Number Sequence Algorithm

(in PyCharm or Python)

```
import math
A=(1+1)                    #(2)
B= (A+1)                   #(3)
C=(B+A)                    #(5)
D=(C+B)-1                  #(7)
E=(D+C)-1                  #(11)
F=(D+C)+1                  #(13)
G=(D+E)-1                  #(17)
H=(D+E)+1                  #(19)
I=(E+F)-1                  # (23)
J=(F+G)-1                  #(29)
K=(F+G)+1                  #(31)
L=(G+H) +1                 #(37)
M=(H+I) -1                 #(41)
N=(H+I)+1                  #(43)
O=(H+J) -1                 #(47)
P=(I+J) +1                 #(53)
Q=(J+K)-1                  #(59)
R=(J+K)+1                  #(61)
S=(J+L)+1                  #(67)
T=(K+M)-1                  #(71)
U=(K+M)+1                  #(73)
V=(M+L)+1                  #(79)............................................infinity.


print(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V)
```

A2 B3 C5 D7 E11 F13 G17 H19 I23 J29 K31 L37 M41 N43 O47 P53 Q59 R61 S67 T71 U73 V79

Zeta Function

$$Zeta(s) := \sum_{n=1}^{\infty} \frac{1}{n^s}$$

$$Zeta(-1) := \sum_{n=1}^{\infty} \frac{1}{n^s}$$

$$Zeta(-1) := \sum_{n=1}^{\infty} \frac{1}{n^s} = \frac{1}{((n)(2664)/(1024))}^{-1}$$

Zeta Function

$$\zeta(-1)\sum_{n=1}^{\infty}\binom{1}{n} = 1/((n)(2664)/(1024))^\wedge - 1 \ \ =$$

$\zeta(-1)\sum_{n=1}^{\infty}\binom{1}{n}1/((n)(2664)/(1024))^\wedge - 1 \ \ +\zeta(-1)\sum_{n=10}^{\infty}\binom{1}{n} = 1/((n)(2664)/(1024)1)^\wedge -$

$1 \ \ +\zeta(-1)\sum_{n=100}^{\infty}\binom{1}{n} = 1/((n)(2664)/(1024)2)^\wedge - 1 \ \ +\zeta(-1)\sum_{n=1000}^{\infty}\binom{1}{n} = 1/((n)(2664)/$

$(1024)3)^\wedge - 1 \ \ +\zeta(-1)\sum_{n=10000}^{\infty}\binom{1}{n} = 1/((n)(2664)/(1024)4)^\wedge - 1 \ \ +\zeta(-1)\sum_{n=100000}^{\infty}\binom{1}{n} =$

$1/((n)(2664)/(1024)5)^\wedge - 1 \ \ +\zeta(-1)\sum_{n=1000000}^{\infty}\binom{1}{n} = 1/((n)(2664)/(1024)6)^\wedge -$

$1 \ \ \zeta(-1)\sum_{n=100000000}^{\infty}\binom{1}{n} = 1/((n)(2664)/(1024)7)^\wedge - 1 \ \ +\zeta(-1)\sum_{n=100000000}^{\infty}\binom{1}{n} =$

$1/((n)(2664)/(1024)8)^\wedge - 1 \ \ +\zeta(-1)\sum_{n=1000000000}^{\infty}\binom{1}{n} = 1/((n)(2664)/(1024)9)^\wedge - 1$

Zeta Function

(in PyCharm or Python)

Zeta function (-1) =(1/1^-1)…+… (1/10^-1)…+ (1/100^-1)…+… (1/1000^-1)…+… (1/10000^-1)…+… (1/100000^-1)…+… (1/1000000^-1)…+ (1/10000000^-1)…+… (1/100000000^-1)…+… (1/1000000000^-1)

Zeta function (-1) = (2)…+…(29)…+…(541)…+…(7919)…+…(104,729)…+…(1,299,709)…+…(15,485,863)…+…(179,424,673)…+… (2,038,074,743)…+…(22,801,763,489)

import math

(1/1^-1)=2

```
n=1 # Nth Prime
C=((n*37*72)/2/2/2/2/2/2/2/2/2/2)*0   # A digits in A-1 ; If A=1 then multiplier = 0
print (C)
# C=2.6015625  Actual Prime Number 2
```

(1/10^-1)=29

```
n=10 # Nth Prime
C=((n*37*72)/2/2/2/2/2/2/2/2/2/2)*1   # A digits in A-1 ; If A=10 then multiplier = 1
print (C)
# C=26.015625 Actual Prime Number29
```

(1/100^-1)=541

```
n=100 # Nth Prime
C=((n*37*72)/2/2/2/2/2/2/2/2/2/2)*2   # A digits in A-1 ; If A=100 then multiplier = 2
print (C)
# C=520.3125 Actual Prime Number 541
```

(1/1000^-1)=7919

```
n=1000 # Nth Prime
C=((n*37*72)/2/2/2/2/2/2/2/2/2/2)*3   # A digits in A-1 ; If A=1000 then multiplier = 3
print (C)
# C=7,804.6875   Actual Prime Number 7,919
```

(1/10000^-1)=104,729

```
n=10000 # Nth Prime
C=((n*37*72)/2/2/2/2/2/2/2/2/2/2)*4   # A digits in A-1 ; If A=10000 then multiplier = 4
print (C)
# C=104,062.5  Actual  Prime Number 104,729
```

(1/100000^-1)=1,299,709

```
n=100000 # Nth Prime
C=((n*37*72)/2/2/2/2/2/2/2/2/2/2)*5   # A digits in A-1 ; If A=100000 then multiplier = 5
print (C)
# C= 1,300,781.25  Actual  Prime Number 1,299,709
```

(1/1000000^-1)=15,485,863

```
n=1000000 # Nth Prime
C=((A*37*72)/2/2/2/2/2/2/2/2/2/2)*6   # A digits in A-1 ; If A=1000000 then multiplier = 6
print (C)
# C=15,609375.0   Actual Prime Number 15,485,863
```

(1/10000000^-1)=179,424,673

```
n=10000000 # Nth Prime
C=((n*37*72)/2/2/2/2/2/2/2/2/2/2)*7   # A digits in A-1 ; If A=10000000 then multiplier = 7
print (C)
# C=182,109,375.0  Actual Prime Number  179,424,673
```

(1/100000000^-1)=2,038,074,743

```
n=100000000 # Nth Prime
C=((n*37*72)/2/2/2/2/2/2/2/2/2/2)*8   # A digits in A-1 ; If A=100000000 then multiplier = 8
print (C)
# C=2,081,250,000.0  Actual Prime Number 2,038,074,743
```

(1/1000000000^-1)=22,801,763,489

```
n=1000000000 # Nth Prime
C=((n*37*72)/2/2/2/2/2/2/2/2/2/2)*9   # A digits in A-1 ; A=If 1000000000 then multiplier = 9
print (C)
# C=23,414,062,500.0  Actual  Prime Number 22,801,763,489
```

Nth Prime Number Finder Algorithm

(in PyCharm or Python)

```
n=1# Nth Prime
B=((n*37*72)/1024)
C=(math.sqrt(B+B)+B)-(n)/(n/1)/2
print(n,B,C)
#1 2.6015625 4.382598448861833  Actual 2

n=3# Nth Prime
B=((n*37*72)/1024)
C=(math.sqrt(B+B)+B)-(n)/(n/1)/2
print(n,B,C)
#3 7.8046875 11.255557657319777   Actual 5

n=5# Nth Prime
B=((((n*37*72)/1024)))
C=(math.sqrt(B+B)+B)-(n)/(n/1)/2
print(n,B,C)
#5 13.0078125 17.608363940775792  Actual 11

n=9# Nth Prime
B=((((n*37*72)/1024)))
C=(math.sqrt(B+B)+B)-(n)/(n/1)/2
print(n,B,C)
#9 23.4140625 29.757170346585497  Actual 23

n=11# Nth Prime
B=(((((n*37*72)/1024)))*1)
C=(math.sqrt(B+B)+B)-(n)/(n/1)/2
print(n,B,C)
#11 28.6171875 35.68252787568701  Actual 31

n=20# Nth Prime
B=(((((n*37*72)/1024)))*1.5)-(n)/(n/1)/2
C=(math.sqrt(B+B)+B)-((n)/(n/1))
print(n,B,C)
#20 77.546875 89.00053911944694   Actual 71

n=30# Nth Prime
B=(((((n*37*72)/1024)))*1.5)-(n)/(n/10)/2
C=(math.sqrt(B+B)+B)
print(n,B,C)

#30 112.0703125 127.04163926151315  Actual 113
```

```
n=40# Nth Prime
B=(((((n*37*72)/1024)))*1.5)-(n)/(n/10)/2
C=(math.sqrt(B+B)+B)
print(n,B,C)
#40 151.09375 168.47729106619246    Actual 173


n=50# Nth Prime
B=(((((n*37*72)/1024)))*2)-(n)/(n/10)
C=(math.sqrt(B+B)+B)
print(n,B,C)
#50 250.15625 272.52391639593856    Actual 229


n=70# Nth Prime
B=(((((n*37*72)/1024)))*2)-(n)/(n/10)
C=(math.sqrt(B+B)+B)
print(n,B,C)
#70 354.21875 380.83523925008706    Actual 349

n=100# Nth Prime
B=(((((n*37*72)/1024)))*2)-(n)/(n/10)
C=(math.sqrt(B+B)+B)
print(n,B,C)
#100 510.3125 542.2597221014598     Actual  541


n=700# Nth Prime
B=(((((n*37*72)/1024)))*3)-(n)/(n/10)
C=(math.sqrt(B+B)+B)
print(n,B,C)
#700 5453.28125 5557.715739034993     Actual 5279

n=999# Nth Prime 1000
B=(((((n*37*72)/1024)))*3)-(n)/(n/10)
C=(math.sqrt(B+B)+B)
print(n,B,C)
#999 7786.8828125 7911.677706726487    Actual  7907

n=1000# Nth Prime
B=(((((n*37*72)/1024)))*3)-(n)/(n/10)
C=(math.sqrt(B+B)+B)
print(n,B,C)
#1000 7794.6875 7919.544918682272       Actual 7919

n=10000# Nth Prime
B=(((((n*37*72)/1024)))*4)-(n)/(n/10)
C=(math.sqrt(B+B)+B)
```

```
print(n,B,C)
#10000 104052.5 104508.68526938076    Actual 104,729

n=100000# Nth Prime
B=(((((n*37*72)/1024)))*5)-(n)/(n/10)
C=(math.sqrt(B+B)+B)
print(n,B,C)
# 100000 1300771.25 1302384.1797876845  Actual 1,299,709


n=1000000# Nth Prime
B=(((((n*37*72)/1024)))*6)-(n)/(n/10)
C=(math.sqrt(B+B)+B)
print(n,B,C)
#1000000 15609365.0 15614952.372369908  Actual 15,485,863


n=10000000# Nth Prime
B=(((((n*37*72)/1024)))*7)-(n)/(n/10)
C=(math.sqrt(B+B)+B)
print(n,B,C)
#10000000 182109365.0 182128449.515451  Actual 179,424,673

n=100000000# Nth Prime
B=(((((n*37*72)/1024)))*8)-(n)/(n/10)
C=(math.sqrt(B+B)+B)
print(n,B,C)
# 100000000 2081249990.0 2081314507.4393477  Actual 2,038,074,743

n=1000000000# Nth Prime
B=(((((n*37*72)/1024)))*9)-(n)/(n/10)
C=(math.sqrt(B+B)+B)
print(n,B,C)
#1000000000 23414062490.0 23414278888.07065   Actual 22,801,763,489
```

Conclusion

Like the Fibonacci or the Lucas sequence, the Pseudo Random Prime Number Sequence is a sequence in which 2 primes when added together + or -1 makes the next prime. Progressing along the prime number line is similar to the Fibonacci sequence and the Lucas sequence. The sum of the two primes, A+B(+or-1)=C dictates the next prime number in the sequence and the gap between the primes. Goldbach's conjecture is that every even integer is the sum of two primes, A+B(+or-1)=C is two primes +or-1 make up another prime and that prime numbers are structured in a Pseudo Random manner.