Fibonacci Random Number Generator using Lehmer’s Algorithm

1. Department of Mathematical Sciences, Nasarawa State University, Keffi, Nasarawa State, Nigeria

* E-mail of the corresponding author: kenresearch@yahoo.com

Abstract

The uniqueness of Fibonacci sequence is been discussed with particular emphasis on its application to random number generation. The Lehmer’s algorithm was employed using fibonacci prime. For multiplier $a=912$, initial seed $x_0=415$, modulus $m=28657$ and multiplier $a=518$, initial seed $x_0=211$, modulus $m=514229$, we generate random numbers with full period $(m-1)$. This suggest that higher values of Fibonacci primes with appropriate choice of a full multiplier $a$, modulus $m$ (fibonacci prime) and a starting seed $x_0$ will produce a full period with finite countable many random numbers. A run test also indicates that the random numbers generated using modulus m as fibonacci prime are truly random.

Keywords: Fibonacci sequence, Fibonacci prime, Random numbers generator, Lehmer’s Algorithm, run test

1. Introduction

The design of nature has been discovered to have underlying mathematical formulation and numerical representations. One such numerical representation found in nature is Fibonacci numbers (Adam, 2006). The Fibonacci numbers are sequence of numbers generated by summing the first two numbers in the sequence to get the next. It is a deceptively simple series of numbers but its ramifications and applications are nearly limitless (Livio, 2002; Conway and Guy, 1996). The Fibonacci sequence is of interest to non-mathematicians primarily because of the possibility of using them to investigate a wide variety of problems. These numbers are researched in the area of number theory, games theory and sequence and it has continued to attract interest among mathematicians to the extent that a quarterly journal is dedicated to Fibonacci series (Hilton & Pedersen, 1994; Matthew & Fink, 2004).

A random number is a number generated by a process which outcome is unpredictable and which cannot be subsequently reliably reproduced. This definition works fine provided that one has some kind of black box, such a black box is usually called random number generator that fulfil the required task. (von Neumann, 1951) Consequently, a random number can also be defined as a number chosen by chance from some specified distribution such that selection of a large set of these numbers produces the underlying statistical distribution. Almost always such numbers are also required to be independent, so that there are no correlations between successive members. The output can be converted to random variate via mathematical transformations.

Historically there are two types of random numbers generators: computer generators (also called True random number generator (TRNG) and algorithmic generators (also called Pseudo-random numbers generator (PRNG)). Pseudo random number generators are algorithm that uses mathematical formulae or simply pre-calculated table to
produce sequence of number that appear random. A good deal of research has gone into pseudo random number theory and modern algorithms for generating pseudo random number and it makes it so good that the number look exactly like they were really random (Knuth, 1997). A good example of pseudo random number generators is the linear congruential method. A linear congruential generator is a method of generating a sequence of numbers that are not actually random but share many properties with complete random numbers. (Neave, 1973; Ferguson, 1960).

Pseudo-random numbers generators are widely accepted because they meet the following criteria: **randomness:** It produces output passes all reasonable statistical tests of randomness; **controllability:** able to reproduce random stream of output, if desired; **portability:** able to produce the same output on a wide variety of computer systems **efficiency:** fast, minimal computer resource requirements and **documentation:** theoretically analysed and extensively tested. When used without qualification the word random usually means random with a uniform distribution, other distribution are of course possible. For example the box-miller transformation allows pairs of uniform random numbers to be transformed to the corresponding random numbers having a two dimensional distribution. It is impossible to produce an arbitrary long string of digits and prove that it is random. When generating random numbers over some specified boundary, it is often necessary to normalize the distribution so that all differential areas are equally computed (Bassiein, 1996).

True random number generators (TRNG) extract randomness from physical phenomenon and introduce it into the computer. The physical phenomenon can be very simple like the little variations in the movement of a mouse or in the amount of time between key strokes. Regardless of which physical phenomenon that is used, the process of generating true random number involves identifying little unpredictable changes in the real life data.

2. An Overview of Lehmer’s Algorithms

Using the note of Leemis and Park, (2006) and Shorey and Stewart, (1981) we present some basic concepts on Lehmer’s algorithm. Lehmer’s algorithm for random number generation is defined in terms of two fixed parameters: **modulus** \( m \), a fixed large prime integer and **multiplier** \( a \), a fixed integer in \( X_m \)

The integer sequence \( x_0, x_1, \ldots \) is defined by the iterative equation \( x_{i+1} = g(x_i) \) with \( g(x_i) = ax_i \mod m \)

\( x_0 \in X_m \) is called the initial seed

We have that \( 0 \leq g(x_i) < m \) because of the mod operator.

However, 0 must not occur since \( g(0) = 0 \)

Since \( m \) is prime, \( g(x) \neq 0 \) if \( x \in X_m \).

If \( x_0 \in X_m \), then \( x_i \in X_m \) for all \( i \geq 0 \).

**Note:** The quality of Pseudo-Random numbers generated depends on a good choice of \( a \) (multiplier) and \( m \) (modulus). The following observations are important:

- \( a \) is a fixed (constant) integer in \( X_m \) also known as multiplier
- \( m \) is a large fixed prime integer also known as the modulus
- \( x_0 \) is the initial starting seed in \( X_m \)
The Mod function ensures a value less than \( m \) is always generated,
m (Modulus) is chosen to be a prime number so that a non-zero remainder always exist, that is \( x_i \) is never 0. If \( x_i \) becomes 0, then all subsequent \( x_i \) will be zero.

### 2.1 The Modulus and Multiplier Selection

Here we discuss how to select a suitable modulus and multiplier that can generate the desired random numbers. When selecting a modulus or multiplier, the following outlined rules must be noted:

(i). The modulus \( m \) should be very large as possible (1 is a good value for modulus \( m \)).

(ii). The modulus must be a prime number in other to avoid the occurrence of zero which subsequently causes \( x_i \) to be zero.

(iii). The multiplier \( a \) should be chosen to guarantee a full period multiplier.

**Theorem 1**

If the sequence \( x_0, x_1, x_2, \ldots \) is a produce by Lehmer’s generator with multiplier \( a \) and modulus \( m \) then \( x_i = a^i x_0 \mod m \)

**Proof**

We know that \( b \mod a = b - \left\lfloor \frac{b}{a} \right\rfloor a \), then there exist a non-negative integer \( c_i = \left\lfloor \frac{a^i x_0}{m} \right\rfloor \) such that

\[
x_{i+1} = g(x_i) = ax_i \mod m = ax_i - mc_i
\]

Therefore (by induction), we have that

\[
x_1 = ax_0 - mc_0
\]

\[
x_2 = ax_1 - mc_1 = a^2x_0 - m(ac_0 + c_1)
\]

\[
x_3 = ax_2 - mc_2 = a^3x_0 - m(a^2c_0 + ac_1 + c_2)
\]

\[
\vdots
\]

\[
x_i = a^i x_0 - m(c_0 + c_1 + \cdots + c_{i-1})
\]

since \( x_i \in \mathbb{Z}_m \), we have that \( x_i = x_i \mod m \)

Therefore letting \( c = a^i c_0 + a^{i-1} c_1 + \cdots + c_{i+1} \), we have that

\[
x_i = a^i x_0 - mc \mod m = a^i x_0 \mod m
\]

Hence \( x_i = a^i x_0 \mod m \)

**Note:** We do not compute \( x_i \) by first computing \( a^i \), this is a wrong approach.

The result of Theorem 1 has a significant theoretical value.

### 2.2 The Period of the Sequence

Consider sequence produced by \( x_{i+1} = a \cdot x_i \mod m \), once a value is repeated, all the sequence is then repeated. That is the sequence: \( x_0, x_1, x_2, \ldots, x_i, \ldots, x_{i+p} \) where \( x_i = x_{i+p} \), \( p \) is the period, that is the number of elements before the first repeat. Clearly we see that \( p \leq m - 1 \)

It can be shown, that if we pick any initial seed \( x_0 \), we are guaranteed this initial seed will reappear.
Theorem 2
If \( x_0 \in \chi_m \) and the sequence \( x_0, x_1, x_2, \cdots \) is produced by the Lehmer’s generator \( x_{i+1} = a \cdot x_i \mod m \) with multiplier \( a \) and (prime) modulus \( m \), then there exist a positive integer \( p \) with \( p \leq m - 1 \) such that:

(i). \( x_0, x_1, \cdots, x_{p-1} \) are all different and

(ii). \( x_{i+p} = x_i, \ \forall \ i = 0, 1, 2, \cdots \)

Proof
We know from modulo arithmetic that
\[
(b_1, b_2 \ldots b_n) \mod a = (b_1 \mod a)(b_2 \mod a) \cdots (b_n \mod a)
\]
Therefore \( x_i = a^i \cdot x_0 \mod m = (a^i \mod m)x_0 \mod m \)

From Fermat’s Little theorem, which states that if \( p \) is a prime which does not divide \( a \), then
\[
a^{p-1} \mod p = 1 \quad \cdots \quad \cdots \quad \cdots \ (\ast.1)
\]

Then \( x_{m-1} = (a^{m-1} \mod m)x_0 \mod m = x_0 \quad \cdots \quad \cdots \quad \cdots \ (\ast.\ast) \)

From (\ast.\ast), we have a more defined generalization, thus
\[
x_{i+p} = (a^{i+p} \mod p)x_i \mod m = x_i
\]
\[\Rightarrow x_{i+p} = x_i \quad \text{Hence the proof} \]

Note:
1. Ideally, the generator cycles through all values in \( \chi_m \) to maximize the number of possible values that are generated, and guarantee any number can be produced.
2. The sequence containing all possible numbers is called a **full-period sequence** (\( p = m - 1 \)).
3. Non-full period sequences effectively partition \( \chi_m \) into disjoint sets, each set has a particular period (not full period).

2.3 Determining if \( a \) is a full period Multiplier
We present the following Algorithm for finding if \( p \) is a full period.

\[
p = 1;
\]
\[
x = a; \quad // \text{assume, initial seed is } x_0 = 1, \text{ thus } x_1 = a
\]
Do
\[
x = (a \times x) \mod m \quad \{/\text{ cycle through numbers until repeat}/*\}
\]
\[
p = p + 1 \quad \{\text{careful: overflow possible}\}
\]
Until \( x = x_0 \)
If \( p = m - 1 \)
\[
\text{Writeln( } a \text{ is a full period multiplier)}
\]
Else
\[
\text{Writeln( } a \text{ is not a full period multiplier)}
\]
End if

3. Numerical Experiments
The following numerical experiments show how random numbers are generated using Lehmer’s algorithm of the formula \( x_{i+1} = a \times x_i \mod m \), considering the multiplier \( a \), the modulus \( m \) and the initial seed \( x_0 \). In each experiment we generate values for \( x_0, x_1, x_2, \cdots, x_i, x_{i+1} \) after making a choice of fibonacci prime as our values for \( m \)
Experiment 1 \((m = 28657, \alpha = 912, x_0 = 415)\) – Five digit fibonacci prime \((m)\)

Experiment 2 \((m = 514229, \alpha = 518, x_0 = 211)\) – Six digit fibonacci prime \((m)\)

The two numerical experiment above produce a full period sequence since \(p = 28657\) (first experiment) and \(p = 514229\) (second experiment) therefore \(\alpha = 912\) and \(\alpha = 518\) are full period multiplier respectively.

3.1 Tests for Randomness

We apply the run test to test the null hypothesis that randomness does not exist in the number generated. Consider a sequence of numbers made up of two set, \(c\) and \(d\), where \(c\) represent the corresponding random numbers generated when it is less than the average and \(d\) represent the corresponding random numbers generated when it is greater than the average.

Suppose we form all possible sequences consisting of \(N_1 c\)’s and \(N_2 d\)’s, for \(N_1 + N_2 = N\) and \(V\) is the total number of runs, then by using the formula

\[
\mu_v = \frac{2N_1N_2}{N_1+N_2} + 1 \quad \text{... \ ... \ ... (1)}
\]

\[
\delta_v^2 = \frac{2N_1N_2(2N_1N_2-N_1-N_2)}{(N_1+N_2)^2(N_1+N_2-1)} \quad \text{... \ ... \ ... (2)}
\]

When \(N\) is relatively large (>20) the distribution of \(V\) is approximately normal and thus

\[
Z = \frac{V-\mu_v}{\delta_v} \sim N(0,1) \quad \text{... \ ... \ ... (3)}
\]

We can test the null hypothesis at the appropriate level of significance using equation (3)

We have that for the first experiment

\(N_1 = 14329, N_2 = 14328, N = 28657, V = 14448.9\)

We have that \(\mu_v = 14329.5\) and \(\delta_v = 84.6404\) and \(Z_{cal} = 1.41076, Z_{Table} = 1.96,\) for \(\alpha = 0.05\) level of significance.

Therefore we reject the null hypothesis and conclude that randomness exist in the random numbers generated since \(Z_{cal}(\text{Test statistics}) < Z_{Table}(\text{Critical value})\)

Using the same approach above, for the second experiment we test for the null hypothesis and conclude that randomness exist in the sets of random number generated.

4 Results and Discussion

From the result of this work, we have shown that a five and six digit fibonacci prime with appropriate choice of full multiplier \(\alpha\), modulus \(m\) (fibonacci prime) and a starting seed \(x_0\) will produce a full period. Table 1 shows the first 600 random numbers generated from the 28, 657 that was generated and Table 2 shows the first 570 random numbers generated from the 514, 229 that was generated. A full period guarantees randomness and a longer length of random numbers sets. The longer the digit of the fibonacci prime, the better and more random the numbers generated will be. Further research should be able to show clearly that not all fibonacci prime will generates a full period no matter the choice of full multiplier \(\alpha\), and a starting seed \(x_0\).
### Table 1: Showing the first 600 random numbers from the 28,657 that was generated \((m = 26,657, x_0 = 912, x_n = 415)\)

<table>
<thead>
<tr>
<th>(n)</th>
<th>(x_n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>615</td>
</tr>
<tr>
<td>2</td>
<td>2070</td>
</tr>
<tr>
<td>3</td>
<td>2497</td>
</tr>
<tr>
<td>4</td>
<td>4669</td>
</tr>
<tr>
<td>5</td>
<td>3713</td>
</tr>
<tr>
<td>6</td>
<td>7371</td>
</tr>
<tr>
<td>7</td>
<td>7851</td>
</tr>
<tr>
<td>8</td>
<td>6982</td>
</tr>
<tr>
<td>9</td>
<td>8128</td>
</tr>
</tbody>
</table>

### Table 2: Showing the first 570 random numbers from the 514,222 that was generated \((m = 514,222, x_0 = 518, x_n = 211)\)

<table>
<thead>
<tr>
<th>(n)</th>
<th>(x_n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>82064</td>
</tr>
<tr>
<td>2</td>
<td>82065</td>
</tr>
<tr>
<td>3</td>
<td>82066</td>
</tr>
<tr>
<td>4</td>
<td>82067</td>
</tr>
<tr>
<td>5</td>
<td>82068</td>
</tr>
<tr>
<td>6</td>
<td>82069</td>
</tr>
<tr>
<td>7</td>
<td>82070</td>
</tr>
<tr>
<td>8</td>
<td>82071</td>
</tr>
<tr>
<td>9</td>
<td>82072</td>
</tr>
</tbody>
</table>

5.
Conclusion

The uniqueness of Fibonacci sequence is been discussed with particular emphasis on its application to random number generation. The Lehmer’s algorithm was employed using Fibonacci prime on a 32 bit machine. Two set of numerical experiments were carried out using a five and six digits fibonacci prime. Both experiments produce large sets of random numbers with full periods. Higher digits fibonacci primes could be studies for randomness and implementation. We suggest that further research be made to devise algorithms that help in finding the appropriate choice of full multiplier $a$, modulus $m$ (fibonacci prime) and a starting seed $x_0$ that will produce a full period.

References

J. London Math. Soc. 23, 17-23
This academic article was published by The International Institute for Science, Technology and Education (IISTE). The IISTE is a pioneer in the Open Access Publishing service based in the U.S. and Europe. The aim of the institute is Accelerating Global Knowledge Sharing.

More information about the publisher can be found in the IISTE’s homepage: http://www.iiste.org

CALL FOR JOURNAL PAPERS

The IISTE is currently hosting more than 30 peer-reviewed academic journals and collaborating with academic institutions around the world. There’s no deadline for submission. **Prospective authors of IISTE journals can find the submission instruction on the following page:** http://www.iiste.org/journals/  

The IISTE editorial team promises to the review and publish all the qualified submissions in a **fast** manner. All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Printed version of the journals is also available upon request of readers and authors.

MORE RESOURCES

Book publication information: http://www.iiste.org/book/

Recent conferences: http://www.iiste.org/conference/

**IISTE Knowledge Sharing Partners**

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digital Library , NewJour, Google Scholar