A STATISTICAL STUDY ON PSEUDO RANDOM NUMBER GENERATORS

BY

NDOBU NGOZICHUKWUKA BLESSING

UNIVERSITY OF BENIN

DEPARTMENT OF STATISTICS

January 2023

UNDERTAKING

This project was duly carried out by me, **NDOBU NGOZICHUKWUKA BLESSING with the matriculation number PSC1707761.** All works were gotten from dutiful research, and reference was made where other author's works were used.

_____ _____

NDOBU N. BLESSING **DATE**

CERTIFICATION

This is to certify that this project work which is titled **PSEUDO RANDOM NUMBER GENERATORS** was carried out by **NDOBU NGOZICHUKWUKA BLESSING,** with the matriculation number PSC1707761 of the Department of Statistics, Faculty of Psychical Science, University of Benin, Benin city.

…….. ……...

Mr. Odijie, C.O. Date

Project Supervisor

……. ……

Prof. C.C Ishiekwene Date

(Head Of Department)

ACKNOWLEDGEMENT

Firstly, I acknowledge and appreciate God almighty for blessing me with strength and wisdom to carry out this research work successfully.

I also acknowledge and appreciate my project supervisor MR. C.O. Odijie, who dedicated his time and effort in teaching and ensuring that I understand my project topic. He was of great support and guidance during the course of this project work.

I also acknowledge my family and friends for their unrelented support and encouragement.

Then to the authors whose works were used to gather the facts and data that were required for this project, I duly acknowledge them.

Finally, I commend myself for the effort and hard work put into this work.

ABSTRACT

This study discusses the pseudo random number generators, one of the categories used in generating random numbers. Random numbers are useful in various simulation processes, such as statistical and numerical analysis, gaming, cryptography, gambling, etc. It is therefore important to the study the process of generating random numbers. The purpose of this study is to observe the concept of the pseudo-random number generating techniques, some examples and required properties of the pseudo random number generators and a specific pseudo random number generating algorithm for the generation of sequence of random numbers.

DEDICATION

I dedicate this work to my project supervisor, MR. C.O. Odijie, to the department of statistics, University of Benin, and to every person that is in the statistical field.

TABLE OF CONTENTS

CHAPTER FIVE: SUMMARY AND CONCLUTION

CHAPTER ONE

INTRODUCTION

1.0 Background of the Study

Random numbers are very essential part of the statistical field and to a very good extent the world at large. This is because a lot of processes seem rather complicated or random, that is, without pattern or unpredictable. For example, the number of calls a person receives per day, the shapes of cloud overtime, the number of cars that drive into a supermarket's garage per hour, the creation of security codes/passwords and so on. With all of these examples, it can be safe to say randomness is part of our everyday life. The study of random numbers becomes very necessary as it is required to simulate these occurrences that are unpredictable. Hence the need to study random numbers and the various methods/techniques in generating them.

Generally, random numbers are classified under two categories which are True Random Numbers (TRN) and the Pseudo Random Numbers (PRN).

The true random numbers are generated using the True Random Numbers Generators (TRNG), which follow a physical method of generation involving a

process whereby the random numbers are generated without a specific pattern. Examples are throwing a fair die, coin, picking random objects from an urn or a container, unpredictable natural processes like thermal noises, etc. It is a basic approach in generating random numbers which is no longer practicable because they are slow and have a limited coverage.

In this study, our concentration is on understanding the concept of the Pseudo Random Numbers generating techniques otherwise known as the PRNG. The PRNG is quite different from the TRNG which follows a completely random process. As the name pseudo (mimic) implies, it does not follow an entirely random process since a predetermined algorithm is required in generating the random numbers. It follows a deterministic procedure in generating the number by starting with a known initial value known as the seed. Once this seed is selected, the other value/number are then generated by applying the algorithm of the specific PRNG. There are several PRNG techniques and the algorithm is dependent on the exact technique of interest. Some of the PRNG include Direct mid square method, linear congruential methods, Blum Blum Shub, etc. Though the PRNG follows a deterministic process of generation, the numbers generated are said to be random (pseudorandom, in fact) because they satisfy the statistical test properties of randomness, uniformity and independence most of the times.

Currently in the world, there are a lot of digital advancement. Creations of Apps, social networks, artificial intelligence, and the likes are fast rising. Majority of these digital works requires cryptography technology which in turn requires the use of random numbers to encode. The creation of the encoding and decoding key requires a process that is not entirely random. This is where PRNG comes to play.

1.1 Aim and Objectives

The aim of this work is to study the concept of pseudo random numbers generators. The objectives are as follows;

- To discuss the theory of the PRNG
- To understand why they are called *pseudo* random numbers
- To study one specific pseudo random number generator and its concept that is the Linear Congruential Generator (LCG)
- To use this known method (LCG) algorithm in generating sequence of random numbers.

1.2 Scope of the Study

This study covers the linear congruential generator. However, since the work on PRNG is quite an elaborate one as there are several techniques/methods under it,

we briefly highlighted some of them which include the middle square method, the linear congruential generator, the Blum Blum Shub, among others.

1.3 Limitation of the Study

As mentioned earlier pseudorandom numbers are not actually random in essence and again computers are needed to generate them. Hence it can be cost effective to implement for students who may not have access to modern computer gadgets.

1.4 Significance of the Study

This study is relevant to formulate various simulation of random numbers. It is also significant to the future undergraduates who will venture into study of generation/simulation of random numbers.

1.5 Definition of Terms

The following terms and their definition are necessary in understanding this work;

- Random numbers: These are numbers gotten from chanced event or who values depends on the outcome of a random experiment.

- Cryptography: This refers to secure information and communication techniques that are derived from mathematical concept and algorithm to transform message sin ways that are hard to decipher.

- Simulation: This is the imitation/presentation of a real-world process or system overtime using mathematical models.

- Encryption: This is the process of converting data from a readable format to scrambled piece of information that cannot be easily understood.

- PRNG: Pseudo random number generators are deterministic techniques used in generating random numbers with a mathematically formulated algorithm.

- Algorithm: These are procedures/steps specifying how to solve a problem.

- Parameter: This can be referred to as any quantity/factor that defines a system and determines its performance

- Model: This is a theoretical description of a complex entity or process.

- Modulus: The modulus is the remainder value gotten after dividing one number by another that is $100\ (mod\ 9)$ equals 1 because $\frac{100}{9} = 11$ with a remainder of 1.

- Function: A function is defined as a relation between a set of inputs having one output each. In simple world, a function is a relationship between inputs where each input is related to exactly one output.

- Integer: An integer is the number zero (0), a positive natural number (1,2,3,4, …) or a negative number with a minus sign (-1, -2, -3, -4, ...)

- Seed: This is the starting/beginning value of every sequence generating an algorithm.

- Sequence: An arrangement of numbers in a particular order.

- Randomness: the quality of lacking any predictable pattern

CHAPTER TWO

LITRATURE REVIEW

2.0 Introduction

Over the years, several researchers have worked on projects on the concepts of PRNG techniques, their specific algorithm and so on. These works have been interesting and valid guides in understanding the PRNG. Hence in this chapter, we review some of the works of these researchers.

2.1 An Overview of the PRNG

The pseudo random numbers generators which are commonly known as PRNG are known algorithm with mathematical model used to generate sequence of random numbers. PRNG models usually have an arbitrary (based on researchers' discretion) starting value which is known as the seed.

The PRNG is an efficient and deterministic model for generating random numbers since the starting point of the output (sequence of random numbers to be generated) can also be reproduced later. It can be said that the name pseudo which implies representative or mimic is where the PRNG derives its name as the process is not entirely random since it follows a predeterminate model. Although, the PRNG follows a deterministic process of generation, the sequence of random

numbers generated are said to be random if they satisfy some statistical test properties such as, test of uniformity and test of independence.

The PRNG are expected to possess some important characteristics, some of the them include;

- Deterministic: The process usually requires a known starting value predetermined by the researcher. This starting value is applied to the specific model of the PRNG which makes it possible for the sequence of random numbers to be reproduced and its quite handy in this computer age.
- Good Distribution: pseudo random numbers are expected to have a good distribution that depicts randomness and uniformity.
- Efficient: there are several applications that requires many random numbers in operation, PRNG becomes efficient for these applications as they can produce many random numbers in a short time.
- Periodic: Periodicity in PRNG means that the sequence of random numbers generated will repeat itself after a while. Most modern PRNG have a period length that is very long that is the terms in the sequence takes a longer time before it starts repeating and hence it can be ignored.

PRNG have several methods/techniques, some of these techniques together with their supposed date of introduction, first proponents and a brief note on them are shown below;

	Generators	**Date**	**First Proponents**	**Notes**
1	Mid-square method	1946	Jon. Von Neuman	In its original form, it is of poor quality and historical interest only
2	Lehmer generator	1951	D. H. Lehmer	One of the very earliest and most influential design
3	Linear congruential generator	1958	W. E. Thomson, A. Rotenberg	A generalization of the Lehmer generator and historically the most studied generator
4	Linear-feedback shift register	1965	R. C. Tausworthe	A very influential design
5	Wichmann-hill	1982	B. A Wichmann and D. I. Hill	A combination of three small LCG suited to 16 bits CPU
6	Blum blum shub	1986	M. Blum, L. Blum and M.	It is considered to be cryptographically secured

			Shub	
7	Mixmax generator	1991	G.K, Savvidy and N.G ter Arutyunyan Savvidy	It is a generalization of the LCG

2.2 Works of some Researchers

According to F. James (1988), pseudo random numbers are generated in the computer by a simple numerical algorithm and are therefore not truly random but any given sequence of pseudo random numbers is supposed to appear random to someone who does not know the algorithm. In the study, the author went further to list and explain some required properties of pseudo random numbers. He stated that amongst all the properties, a good distribution is important for all calculation whereas the other properties are not always needed but a good general-purpose generator should possess them all.

According to a study carried out by Ritu Maheshwari[*], in her review of PRNG, it was stated that a PRNG was introduced by Jon von Neuman. This method was known as the "mid square method". In this method as usual for every PRNG, an

initial seed value (either two, three, four digit) is selected and squared. After squaring it, we select the middle digit of that number as seed value for next pseudo random numbers; and this is clearly where the techniques got its name from. This algorithm is repeated over again to generate a sequence of random numbers. However, the middle square was considered to be insufficient as it only generates a minimal number of random variables. This is because continuous repetition of the algorithm will lead to repeated sequence thereby making the random numbers predictable and no longer sufficient for operation. It is majorly of historical interest only.

Another algorithm for generating PRNG was also stated in the review by Ritu and this method is the linear congruential generator. According to Ritu and her team, this method was proposed in the year 1949 by D. H. Lehmer and it is recognized as one of the oldest and best-known pseudo random number generator algorithm. This algorithm was achieved with the help of a simple linear equation and its seen as fast and easy. The basis of this equation is the modulo arithmetic and it is defined by a recurrence relation;

$$X_{n+1} = (aX_n + c) mod\ M$$

Where;

a is the multiplier

c is the increment

M is the modulus

X_0 is the seed value

X_n is the sequence of pseudo random values.

The LCG becomes predictable as we keep on applying the algorithm this becomes the disadvantage of this technique.

Hernndez *et al*. (2001), in their work, presented a method-based genetic algorithm that can automatically solve the problem of finding good parameter for an LCG. They also showed that the selection of an evaluation function for generated solutions is critical to the problems and how a seemingly good function such as entropy could lead to poor results. In the introductory part of their work, it was stated that the LCG is one of the most widely used pseudo random number generator and they are also one of the best analyzed models used in a great number of applications such as simulation in numerical analysis and optimization. From the study, the LCG equation have the form;

$X_{n+1} = (aX_n + b)\ mod\ m$ with X_0=seed.

The equation is completely characterized by the parameters (Xo, a, b, m), although the LCG is cryptographically limited.

Lukasiewicz (2022) stated that most random numbers used in computer programs are pseudo random which means that they are generated in predictive fashion using a mathematical formula. In the work, it was agreed that pseudo randomness is not a trivial matter and that even the simplest algorithm has a lot of thoughts put in it.

Gentle 2005 gave the LCG equation to be;

$X_i \equiv (aX_{i-1} - 1 + c) mod\ M, 0 \leq X_i < M.$

If $c = 0$ in the equation, then the equation will appear in the form;

$Xi \equiv aX_{i-1}\ mod\ M, 0 < X_i < M.$ In this case, the generator is called a multiplicative congruential generator. A sequence resulting from the first equation is called a **Lehmer sequence.** Each X_i is scaled into the interval (0,1) by dividing by M, that is;

$u_i = \frac{X_i}{M} \sim Uniform$ (0,1). The choice of 'a' and 'M' determines the behavior of the sequence of random numbers generated. If they are both properly chosen, this will look like they are randomly and uniformly distributed between 0 and 1.

CHAPTER THREE

METHODOLOGY

3.0 Introduction

There several techniques used in generating pseudorandom numbers and each one is unique to its specific algorithm. However, we expanded more on the Linear congruential generator.

3.1 The Linear Congruential Generator

The linear congruential generator is a PRNG technique that involves the use of an algorithm to generate a sequence of pseudo-randomized number calculated with a discontinuous piecewise linear equation.

Amongst many other PRNG, the LCG is considered as one of the most common and widely used PRNG. As usual for every PRNG, the LCG has its own unique algorithm used in generating sequence of random numbers. The method follows a linear form whereby each term in the sequence is generated by recursion (repeated solving) of a particular linear expression;

$$Xi \equiv aXi - 1 + c(mod\ M)$$

As we read further, this linear expression will be explained. The LCG is also known for its easy approach in solving, this is majorly one of the reasons why it is widely used.

3.2 The Linear Congruential Generator Recurrence Formula

The simple form of the LCG formula is;

$$X_i \equiv aX_{i-1} + c(mod\ M)$$

$$U_i = \frac{X_i}{M} \sim u(0,1)$$

Where $a, c\ and\ M$ are integers and $X_i, i = 1,2,...$ is the sequence of integer with $0 \leq X_i \leq M$. The normalized sequence $U_i, i = 1,2,...$ is a random sequence in [0,1].

From the formula;

$$X_i = terms\ in\ the\ sequence$$

$$a = multiplier\ (0 \leq a \leq M)$$

$$c = increment\ (0 \leq c < M)$$

$$M = \ modulus \ (0 < M)$$

$$X_{i-1} = the \ term \ preceding \ the \ Xith \ term$$

To generate the first term that is the X_1 term in the sequence, we will need an initial value X_0 which is selected or predetermined by the researcher. X_0 represents the starting value otherwise known as the seed value. The integers a, c, M. the choices of these parameters usually determine the length of the sequence and it also affects the statistical properties of the sequence that is the mean and variance.

There are two different variations of the LCG and they are the mixed congruential method and the multiplicative linear congruential method**.**

In the case where $c \neq 0$, we say it is a mixed linear congruential method. While in the case where $c = 0$, we say it is the multiplicative linear congruential.

3:3 Some Concepts of LCG

To further understand the LCG, the following concepts must be looked into;

1. **The concept of modulus**:

let $X_i \equiv (aX_{i-1} + c) \ mod \ M$ be a linear equation, when the value in the parenthesis is divided by M, the remainder is what we refer to as modulus and also what makes up the Xi in the sequence of random numbers generated.

2. Cycle (Period) Length: The length of an LCG is referred to the number of terms generated before it starts repeating values (the X_i will be regenerated after a finite number of recursion). It is otherwise known as the period. The LCG attains full periodicity when the $length = M$. We say a loop has occurred when the sequence start repeating itself. However, it is quite essential for random sequence to have longer cycle as this makes the sequence appears more random. To achieve a longer period or maximum periodicity, proper choices of parameters a, c, M becomes necessary.

3. Choice of Parameters

Proper choices of parameter are needed to achieve maximum periodicity. The following are sufficient recommendation in obtaining maximum periodicity;

For the mixed congruential method $i.e\ c \neq 0$; an LCG with (a, c, M) has a period of length M if and only if;

1. $gcd(c, m) = 1$ that is c and M are relative primes.
2. $a = 1\ (mod\ p)$ for every prime p dividing M
3. $a = 1\ (mod\ 4)$ if M is a multiple of 4

For the multiplicative linear congruential method $i.e.c = 0$, an LCG with $(a\ , 0, M)$ has a maximal period length $\lambda(M)$ if and only if;

1. $gcd\ (X_0, M)$
2. $a\ is\ a\ primitive\ element\ modulo\ M$

In recent times a common choice of parameter has been to take M as a prime and a as a primitive root modulo M. a popular example is; Lewis et al, (1969)

$$a = 7^5$$

$$c = 0$$

$$M = 2^{31} - 1$$

If M is prime and a is a primitive $modulo\ M$, the period becomes $M - 1$ irrespective of what c is. Also, for a prime M, maximum period is attained only when

$$a = 1\ and\ gcd(c, M) = 1.$$

3:4 Statistical Properties of Pseudo Random Numbers.

To ensure that the sequence of pseudo random numbers generated with the LCG algorithm satisfy desirable statistical properties, some tests must be carried out. They include statistical test of uniformity, randomness and independence.

3.4.1 Test of uniformity: if U_i has a uniform distribution, then $E(U_i) = \frac{1}{2}$ and $var(U_i) = \frac{1}{12}$. The test is therefore to compute the mean and variance of the

sequence of uniform random numbers generated and compare with the theoretical values defined above.

3.4.2 Test of independence: The sample auto correlation function of the generated $U_i's$ can be used as test of independence since auto correlation function at lag k=1, that is identically 0, implies uncorrelated variables and independence. Alternatively, the autocorrelation function (ACF) of MATLAB (which is the major package for our analysis in this work) can be used to plot the autocorrelation function values. It not only plots these values, but also set bounds within which autocorrelation coefficients for different lags should be. The formula for ACF is given by;

$$\rho_k = \frac{\sum_{i=k+1}^{N}(u_i - \bar{u})(u_{i-k} - \bar{u})}{\sum_{i=1}^{N}(u_i - \bar{u})^2}$$

where;

ρ_k is the correlation coefficient at lag k.

$u_i's$ are the generated sequence of uniform random numbers

$\bar{u}$ is the mean of the sequence.

.

3.5 The LCG Algorithm Summary

The LCG algorithm is summarized thus;

STEP 1: Select good choices of seed value X_0 and the parameters $a, c\ , M$

STEP 2: Compute the sequence of random integers

$$X_i \equiv\ aX_{i-1}\ +\ c(mod\ M)$$

for $i = 1, 2, \dots, M$

STEP 3: Compute the sequence of uniform random numbers $U_i's$

$$U_i = \frac{X_i}{M}$$

for $i = 1, 2, \dots, M$

Example: Suppose we want to generate a sequence of uniform random numbers using the LCG algorithm discussed in the foregoing, with parameters $a = 5, c = 1, M = 16, and\ X_0\ = 1.$

Solution

Applying the LCG formula;

$$X_i \equiv (aX_i - 1 + c\ )mod\ M$$

$$X_1 = (5(1) + 1) mod\ 16 = 6\ mod\ 16 = 6$$

$$X_2 = (5(6) + 1) mod\ 16 = 31\ mod\ 16 = 15$$

$$X_3 = (5(15) + 1) mod\ 16 = 76\ mod\ 16 = 12$$

$$X_4 = (5(12) + 1) mod\ 16 = 61\ mod\ 16 = 13$$

As we continue to solve recursively, the sequence generated will be;

1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5, 10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, …

Observe that the loop occurred immediately after the 16^{th} term i.e., the sequence begins recurring after every 16^{th} term. The period in this case is equal to 16 which is the same value with M.

CHAPTER FOUR

SIMULATION STUDY AND RESULT

4.0 Introduction

In this chapter, we present the simulation of the LCG algorithm for generating pseudorandom numbers using the MATLAB software.

4.1 MATLAB Simulation Code to Generate Pseudorandom Number Using the LCG Algorithm

The MATLAB is a computer statistical software package usually used for the computation and analysis of large data. Below is the MATLAB code used for the simulation of the LCG algorithm.

```
clc
x = zeros;  % to store the generated random integers
u = zeros;  % to store the generated uniform random
numbers
x(1) = input('Enter the seed: ');  % equivalent to
seed, Xo, since MATLAB does not accept x(0)
m = input('Enter the modulo/number of random number to
generate: '); % modulo input
u(1) = x(1)/m;  % equivalent to Uo (first uniform r.
n.)
a = input('Enter the multiplier: ');          %
multiplier
c = input('Enter the increment: ');          %
increment

% Table to display result
```

```
disp('=====================================')
disp(' i        random int     uniform rand ')
disp('-------------------------------------')
fprintf('%2.0f %10.0f %20.4f\n', 0, x(1), u(1)) % print
first result

% The actual iteration for the remaining m-1 terms
for i = 2:m
x(i) = rem(a*x(i-1)+c, m);  % the recursive formula for
the r. integers
u(i) = x(i)/m;              % the uniform r. n.
formula/generator
fprintf('%2.0f %10.0f %20.4f\n', i-1, x(i), u(i)) %
print results
end
```

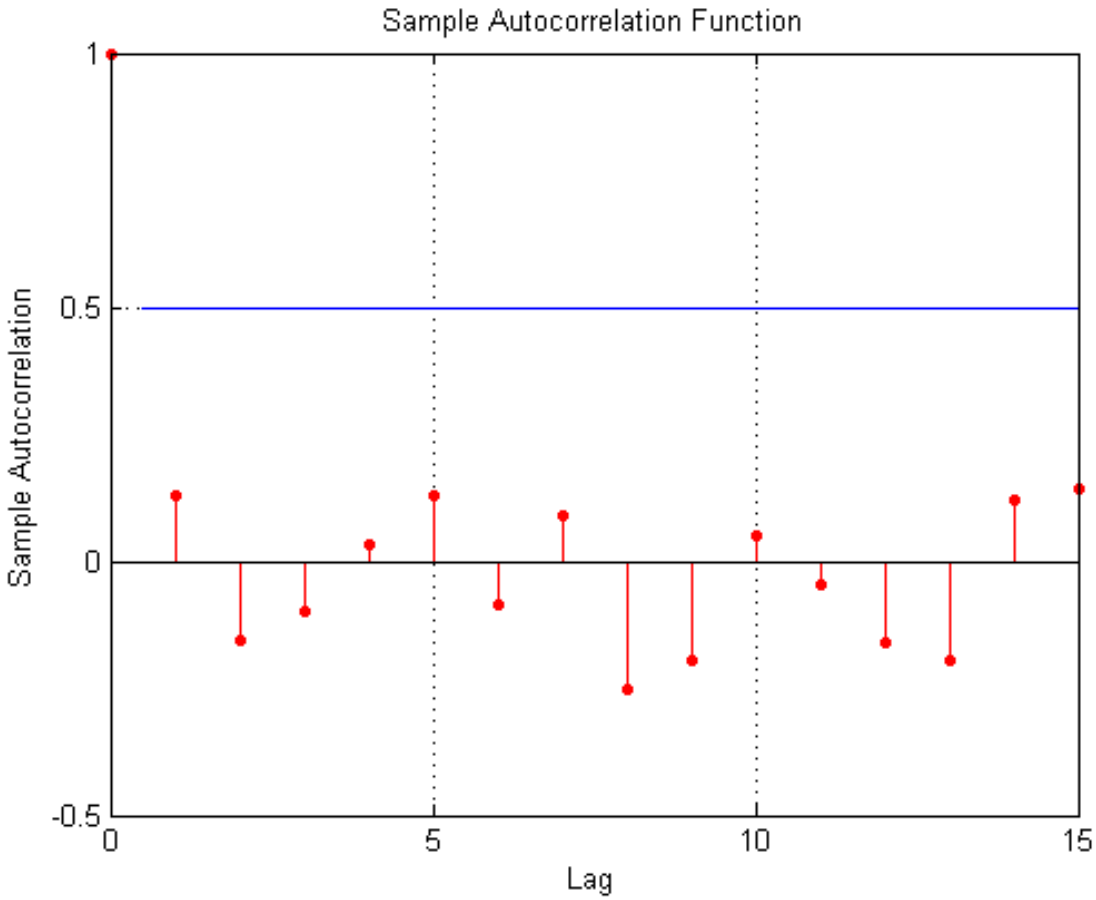
4.2 Simulation Study 1

The MATLAB result for the example in section 3.5 of chapter 3, with $X_0 = 1$, $M = 16,\ a = 5,\ c = 1$ is represented in the Table 2 as follows.

Table 2.

Iteration number	Pseudo random integers (X_i)	Pseudo random integers $U_i = \frac{Xi}{M}$
0	1	0.0625
1	6	0.3750
2	15	0.9375
3	12	0.7500
4	13	0.8125
5	2	0.1250
6	11	0.6875
7	8	0.5000
8	9	0.5625
9	14	0.8750
10	7	0.4375
11	4	0.2500
12	5	0.3125
13	10	0.6250
14	3	0.1875
15	0	0.0000

Fig.1.0 Autocorrelation Plot



From the MATLAB result above, maximum periodicity was achieved from the sequence generated before the loop occurred, but the number of terms in the sequence generated was not large enough due to the choice of the modulo parameter $M = 16$

4.3 Simulation Study 2

In simulation study 1, we observe that the sequence of random numbers had a short period since the modulo M = 16 is short. In this simulation, we chose a larger modulo and relevant parameters for a longer period as follows $X_0 = 5, M = 101, a = 11\,, c = 0$

Using the MATLAB code in section …, we obtained the following results as displayed in Table 2.

Table 2

Iteration number	Pseudo random integers (X_i)	Pseudo random integers $U_i = \frac{Xi}{M}$
0	5	0.0495
1	55	0.5446
2	100	0.9901
3	90	0.8911
4	81	0.8020
5	83	0.8218
6	4	0.0396
7	44	0.4356
8	80	0.7921
9	72	0.7129
10	85	0.8416
.	.	.
.	.	.
.	.	.
91	27	0.2673
92	95	0.9406
93	35	0.3465
94	82	0.8119
95	94	0.9307
96	24	0.2376
97	62	0.6139
98	76	0.7525
99	28	0.2772
100	5	0.0495

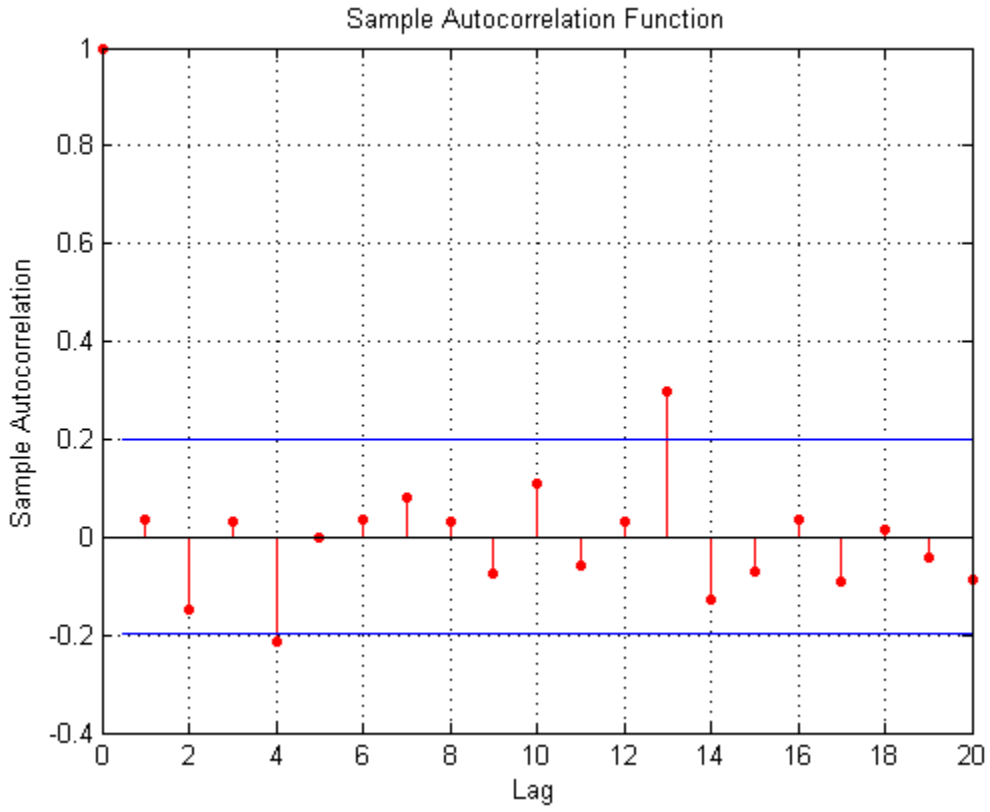
Test of uniformity

Table 3

Statistical properties	theoretical values	Estimated values
Mean	0.5	0.4955
Variance	0.0833	0.0837

Since the estimated values of the mean and variance are close to the theoretical values in Table 3 above, we can say that the sequence generated satisfies the statistical test property of uniformity and randomness, since the theoretical uniform distribution possesses these values

Fig 2 Autocorrelation Function graph



Since each value of the correlation coefficient for each lag k of the generated random variables lies within the automatically set bounds $(-0.2, 0.2)$ except

for $k = 4$ and $k = 13$, which is a little out of the bound (but still within 0.5 outer bounds), we can safely conclude that the sequence of uniform random numbers generated are independent.

CHAPTER FIVE

SUMMARY AND CONCLUSION

5.1 Summary

In this work, we have been able to study the concept of the pseudo random numbers generating techniques. However, emphasis was more on the linear congruential method of generating random numbers. The specific algorithm for the LCG was given and it was also used in generating sequence of random numbers. Also, some statistical tests were carried out on the sequence generated by the LCG algorithm to confirm if they satisfy the statistical properties of random numbers.

5.2 Conclusion

Random number, are a very crucial part of the statistical field and even other fields when it comes to simulating real life phenomena. This study has been able to present one of the methods of generating uniform sequence of random numbers which is the basis for generating other types of random numbers from known distributions.

REFERENCE

F. James (1998), A review of pseudo random number generators. *CERN – Data handling division.*

George Marsaglia (1968), random numbers fall mainly in the planes.

Hui-Chin Tang (2006), An analysis of linear congruential random number generator when multiplier restriction exists. *European journal of operation research 182 (2007) 820 – 828.*

Jakub Luksiewicz (2022), a brief overview of pseudo random number generators and testing of our own simple generators. *Jagnellonian university* – https://orcid.org/0000-0002-4938-504X

Opone. F and Ekhosuehi N. (2017). On some methods of generating random variables. *NSA Conference proceeding (2017).*

Pierre L. Ecuyer (1988). Efficient and portable random number generators comm ACM 31: 742

S. Tezuka, Uniform random numbers. *Kluwer academic publisher*

Online sources;

http://ijsetr.com/uploads/354216review%20paper.pdf

https://en.wikipedia.org/w/index.php?title=List-of-random-number-generators&oldid=1121082575

https://en.wikipedia.org/w/index.php?title=Random-number-generation&oldid=116830870