

Random Number Generator (RNG)

Solution to Probability Problem through Simulation

EE 381 – Project 1, 5 Points

Due date: 2/1/21

Introduction

Random Number Generator

In this project we will explore what is sometimes referred to as a random number generator. The linear congruential generator is a pseudorandom number generator that the reader may already be familiar with. This is the type of pseudorandom number generator that will have a definite pattern yet will appear to have no discernible pattern detectable without knowing the exact formula used. Random means no pattern while pseudorandom means no apparent pattern. The formula can be expressed as

$$S_{i+1} = (M * S_i + A) \bmod N$$

where M is the multiplier, A is the adder, N is the norm. The first *seed* is S_0 and each subsequent seed is determined from each iteration of the formula.

Do the following exercises below on paper by hand.

Exercise 1

Given the values below determine the pseudorandom number sequences for seed values of 0 and 4. Calculate a list of sixteen values.

$M = 6$, $A = 5$, and $N = 11$.

Exercise 2

For the generator $S_{i+1} = (6 * S_i + 3) \bmod 7$ determine the pseudorandom number sequences for the following initial seed values, 0, 1, 4, and 5. Calculate a list of ten values.

There is a theorem that gives the requirements for the optimal values to be used in the generator.

Theorem (Statement without proof.)

The pseudorandom number generator will produce the maximum cycle length N of pseudorandom numbers with any initial value of S under either of these conditions:

For base ten (decimal numbers)

N is a power of 10, A ends in (unit digit) 1, 3, 7, or 9, and $M - 1$ is a multiple of 20.

For base two (binary numbers)

N is a power of 2, A is odd, and $M - 1$ is a multiple of 4.

The interested reader can use the values below and a seed of 0 to observe that the generator has a length of 16.

$N = 16, A = 7, M = 21$

Pseudorandom number generator for use in this project

Write a program in Python that generates 100 pseudorandom numbers in $[0, 1)$. Format it to four decimal positions.

Use $N = 10000, A = 4857, M = 8601$.

Simulations using our pseudorandom number generator

Example 1: Flipping a Coin

Use the pseudorandom number generator to simulate a coin being flipped twenty five times.

Example 2: Rolling a Die

Use the pseudorandom number generator to simulate a die being rolled twenty five times.

To the student: A significant portion of the preceding is from F. Solomon's "Stochastic Processes"

Instructions: During the laboratory meetings of 1/25 and 1/27 our pseudorandom number generator program will be written in Python and it will be used to do the simulations above.

Deliverables: A copy (picture) of the work done by hand, a PDF version of the programs with the output, and the Python file all with name and ID # to dropbox.