Task 2: Pseudorandom Number Generator

Annika Gerigoorian, 990816-4289

IV1013

April 7, 2021

1 Introduction

Choosing my prime number and my constants for my designed pseudo- random number generator was done by following the example on the Linear Congruential Generator in the lecture Symmetric Key Encryption 1. I analysed my prime number and constants in Matlab and compared them to Java's Random class.

2 Discussion

The Java random class use the constants:
b=11
m = 2^{48}-1
a = 0x5DEECE66DL
When deciding my own prime number (m), I chose m to be a large prime number since the period should be large. When deciding on m I looked at different sources. It was stated that the Marsenne prime numbers m = 2^{31}-1 and m = 2^{61}-1 often are used so I decided to choose m = 2 (31)-1 . Furthermore, I picked the b constant to be equal to 0. After this, I chose a= 7^5 to be the primitive root, which leads to a max period of:

$\phi(m) = m - 1 = (2^{31}\text{-}1)\text{-}1 = 2^{31}\text{-}2$

I therefore, have a pseudo random sequence in the interval [1,2^{31}-2].

I created a histogram in MATLAB and as could be seen in Figure 1 and 2. I chose to generate 10 thousand random numbers. I picked the initial seed (x_0) to be x_0 = 10007. I decided on this initial seed value since I read that the initial seed should be a large and odd prime number. In addition, I also tried out different seed values in MATLAB and from the ones I tested, 10007 was the best option.

When comparing Figure 1 and Figure 2, one could see that both my own chosen constants for the linear congruential generator, and Javas Random class generated uniform pseudorandom numbers. The numbers in Figure 1 and 2 are approximately around 1000 which indicates that it is uniformly distributed.

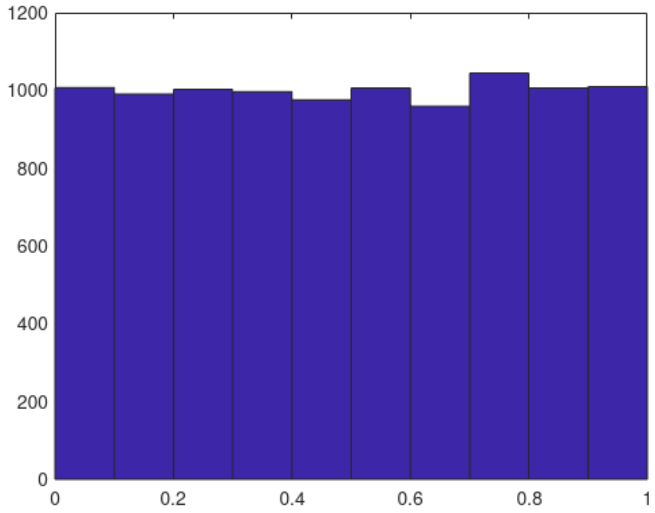

Figure 1: Histogram of the LCG with my chosen constants

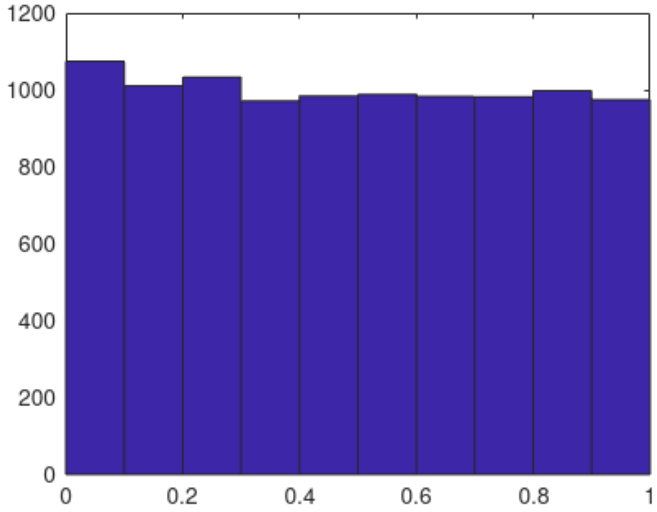

Figure 2: Histogram of the LCG for Javas Random class