%Exercise 4\_1

%Prabavathy Rajasekaran(2130757

% Anjaly Kuriakose(2132537)

clear;

clc;

x= 100000;

seed = 200;

a = 1234557;

c = 765656;

m = 2^24;

z = rng(x, seed, a,c,m);

figure;

histogram(z, 'Normalization','pdf');

title('Normalised histogram');

figure;

[counts,centers] = hist(z);

bar(centers, counts);

title(sprintf('linear congruence Generator z\n Mean = %1.2f\n Variance = %1.2f\n Skewness = %1.3f\n Kurtosis = %1.2f',mean(z),var(z),skewness(z),3-kurtosis(z)));

fprintf('Mean : %d\n',mean(z));

fprintf('Variance : %d\n',var(z));

fprintf('Kurtosis : %d\n',kurtosis(z));

fprintf('skewness : %d\n',skewness(z));

% Linear Congruential Generator − pseudo random number generator

%Prabavathy Rajasekaran(2130757

% Anjaly Kuriakose(2132537)

function [z] = rng(N,seed,a,c,m)

persistent x;

switch nargin

case 1 % One Input argument rng(N)

% default set of magic numbers

a = 140671485;

c = 12820163;

m = 2^24;

x(1)= 200;

case 2 % two Input argument rng(N , seed)

% default set of magic numbers

a = 140671485;

c = 12820163;

m = 2^24;

x(1)= seed;

case 5 % All Input argument rng(N,seed,a,c,m)

x(1)=seed;

magic(a,c,m);%verifies whether the three condition satisfied or not

otherwise

fprintf('Input in the form of rng(N) or rng(N,seed) or rng(N,seed,a,c,m\n');

end

%Recursion formula of Linear congruence random number generator

for i=2:N

x(i) = mod((a\*x(i-1)+c),m);

z= x/m;

end

end

% check whether the generator satisfies the 3 Magic Conditions of Linear Congruence

% Generator

%Prabavathy Rajasekaran(2130757

% Anjaly Kuriakose(2132537)

function magic(a,c,m)

if((intersect(factor(m),factor(c))~=0) || (mod(a-1,m)~=0) || ((mod(a-1,4)~=0 && mod(m,4)~=0)))

fprintf('Warning!! the given values a, c, m doesnot satisfy the 3 magic conditions!');

end

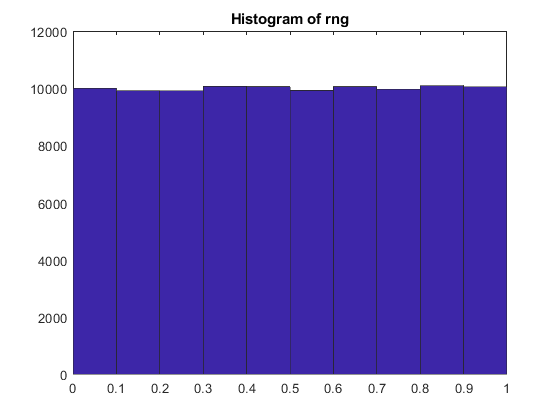
end

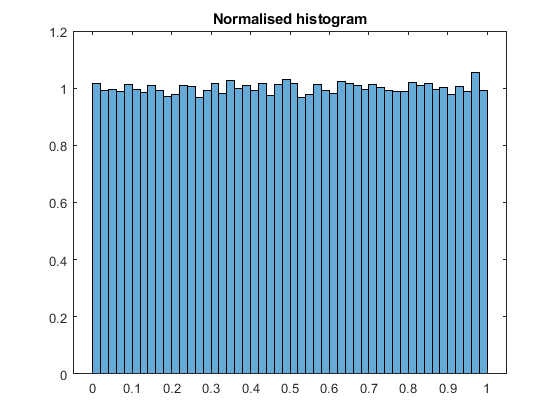
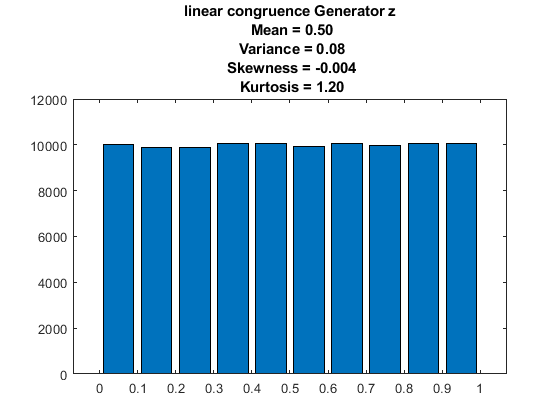
Warning!! the given values a, c, m doesnot satisfy the 3 magic conditions!Mean : 5.008932e-01

Variance : 8.337772e-02

Kurtosis : 1.801492e+00

skewness : -4.005228e-03



[*Published with MATLAB® R2021b*](https://www.mathworks.com/products/matlab/)

%Exercise\_4\_2 Box Muller Transform

%Prabavathy Rajasekaran(2130757

% Anjaly Kuriakose(2132537)

clear;

clc;

z = gaussrng(10000,0.2,4);

figure

hist(z,100);

title(sprintf('Box-Muller Transform z\n Mean = %1.2f\n Variance = %1.2f\n Kurtosis = %1.2f',mean(z),var(z),3-kurtosis(z)));

xlim([-6 6])

fprintf('first order moments %d\n' ,moment(z,1));

fprintf('second order moments %d\n' ,moment(z,2));

fprintf('third order moments %d\n' ,moment(z,3));

%Prabavathy Rajasekaran(2130757

% Anjaly Kuriakose(2132537)

function [z]= gaussrng(N,mu,sig)

u = rng(N);

for k=1:2:N-1

z(k)= sqrt(-2.\*log(u(k))).\*cos(2\*pi\*u(k+1));

z(k+1)= sqrt(-2.\*log(u(k))).\*sin(2\*pi\*u(k+1));

end

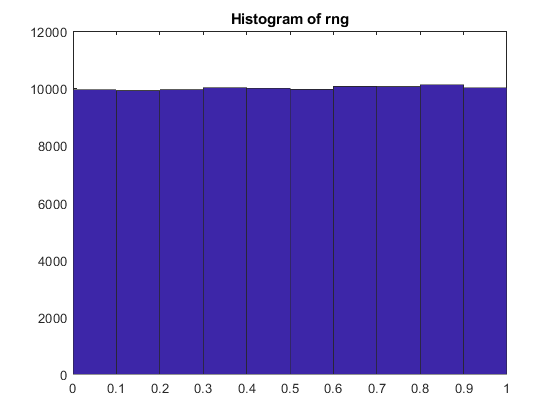
z = sig.\*z+mu;

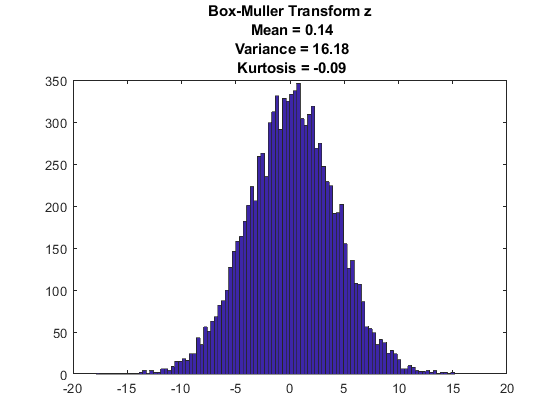
end

first order moments 0

second order moments 1.617872e+01

third order moments -2.010288e+00





[*Published with MATLAB® R2021b*](https://www.mathworks.com/products/matlab/)