Problem Statement Analysis

Firstly, we need to use the rand function to generate the sequence of random numbers in interval of 0 and 1. Secondly, we need to compute mean and variance of the sequence and compare them with the expected mean and expected variance. Then we need to find covariance between the first sequence and other sequences of similar size. And at last we need to convert the uniform continuous random number to uniform discrete random number by grouping the number of sequence in interval of 0.1 from 0 to 1. We will also see the results of goodness of fit test for the rand function.

Theoretical Model

For any given sequence of samples,

Expectation value: $E[X] = \sum_{i=1}^{N} E[X_i]$

Variance: $Var(x) = E[X^2] - (E[X])^2$

Covariance: $Cov(X,Y) = E[XY] - E[X]E[Y]$

For samples from (a, b), expected value of mean and variance are given by

$$\bar{X} = \frac{a+b}{2}$$

$$Var(X) = \frac{(b-a)^2}{12}$$

The weak law of large number states that if the size of samples N increases to infinity then the value of sample mean tends to be expected value as given below:

$$\frac{\sum_{i=1}^{N} X_i}{N} \rightarrow E[X] \; as \; N \rightarrow \infty$$

From the simulation results we will see that as the size of sequence N increases the value of sample mean, sample variance becomes equal to the expected value. We will see that if the covariance between the samples is zero then they are independent in nature. We will also see the results of Chi-square test, KS test for uniformity of the sequence.

Simulation Model

Below is the MATLAB code to understand the functionality of random numbers and compute all the required parameters.

Code:

```
% Matlab program to understand the behaviour of continuous random
% variable its mean, variance, covariance and to generate uniform discrete
% variable
clear

% Calculate Expected Mean and Variance
a=0;          % Lower bound value
b=1;          % Upper bound value
expected_mean=(a+b)/2;
expected_var=(b-a)^2/12;

% Generate the sequence of random numbers between 0 and 1
```

```
size=input('Enter the size of sequence: ');
x=rand(1,size);

% Calculate Mean and Variance of the sequence
mean=mean(x);
variancex=var(x);

% Calulate covariance of (x1 x2),(x,x3),...,(x,x11)
for i=1:10
    X=cov(x(1),x(i+1));            % Calculates covariance between x1 and
xi+1
    covariance(i)=X(1,2);
end

% To prove independence of two sequences
y=rand(1,size);
variancey=var(y);
z=x+y;
variancez=var(z);

% To groupify numbers in sequence in interval of 0.1 from 0 to 1
[N,edges]=histcounts(x,10)

% Plot the frequency distribution of above discreted numbers
figure(1);
bar(0:1:9,N)
xlim([0 9]);
xlabel('Generated values (Bins)')
ylabel('Number of occurences')
title(['Histogram of uniform random numbers in sequence of ' num2str(size)
' samples'])
hold on
A(1:1,1:10)=size/10;
plot(0:1:9,A,'red','LineWidth',3)
xlim([0 9]);
legend('Simulation results','Expected results');
```

Output results:

We will see the results of the simulation for three different sequence sizes viz. 100, 1000 and 10000.

(A) Mean and Variance

	Sequence size	**100**	**1000**	**10000**
Mean	Expected value	0.5000	0.5000	0.5000
	Simulation value	0.5280	0.5069	0.5004
Variance	Expected value	0.0833	0.0833	0.0833
	Simulation value	0.0882	0.0814	0.0844

Table 1 Comparison of expected value and simulation value for mean and variance

From the above data, the expected value of mean is 0.5 and variance is 0.0833. We can see that as the number of samples increases the simulation value comes closer to the expected value. Hence, the law of large numbers holds true for the sequence of random numbers. We can say that as n → ∞, Simulation value → Expected value.

(B) Covariance and Independence tests

Sequence size	100	1000	10000
Covariance (x_1 x_2)	0	0	0
Covariance (x_1 x_3)	0	0	0
Covariance (x_1 x_4)	0	0	0
Covariance (x_1 x_5)	0	0	0
Covariance (x_1 x_6)	0	0	0
Covariance (x_1 x_7)	0	0	0
Covariance (x_1 x_8)	0	0	0
Covariance (x_1 x_9)	0	0	0
Covariance (x_1 x_{10})	0	0	0
Covariance (x_1 x_{11})	0	0	0

Table 2 Sample Covariance for sequences of different size

In the above table, we have covariance (x_1 x_2), (x_1 x_3),…, (x_1 x_{11}) for different sequence sizes. The values of x_1 and x_i does not depend on each other so the value of covariance must be zero. If covariance is positive value, then two parameters have linearly increasing or decreasing behaviour. If covariance is negative value, then two parameters have linearly opposite behaviour. And if two parameters are independent, then their covariance is zero.

Also, if two random distributions X and Y are independent, then

Variance (X) + Variance(Y) = Variance(X+Y)

Variance(X)	Variance(Y)	Variance(X)+Variance(Y)	Variance(X+Y)
0.0838	0.0831	0.1669	0.1661

So, from observations about covariance being zero and property of addition of variance of random number, we can infer that random number generated by rand function are independent in nature.

(C) To generate uniform discrete random variable

Here, the program clubs the values in particular interval and maps them to single positive integer. Function *histcount* is used to do grouping of numbers intervals of 0.1 from 0 to 1. The below table shows the grouped samples for sequences of size 100, 1000 and 10000.

Interval	Mapped Value	Sequence Size		
		100	1000	10000
0-0.1	0	8	105	1016
0.1-0.2	1	12	91	1025
0.2-0.3	2	11	110	1030
0.3-0.4	3	12	102	1020
0.4-0.5	4	9	91	991
0.5-0.6	5	10	100	977
0.6-0.7	6	9	103	1012
0.7-0.8	7	8	108	985
0.8-0.9	8	15	98	957
0.9-1	9	6	92	987

Table 3 Converting continuous distribution to discrete distribution

The frequency distribution of simulation results and the expected frequency distribution for sequences of size 100, 1000 and 10000 are shown below.

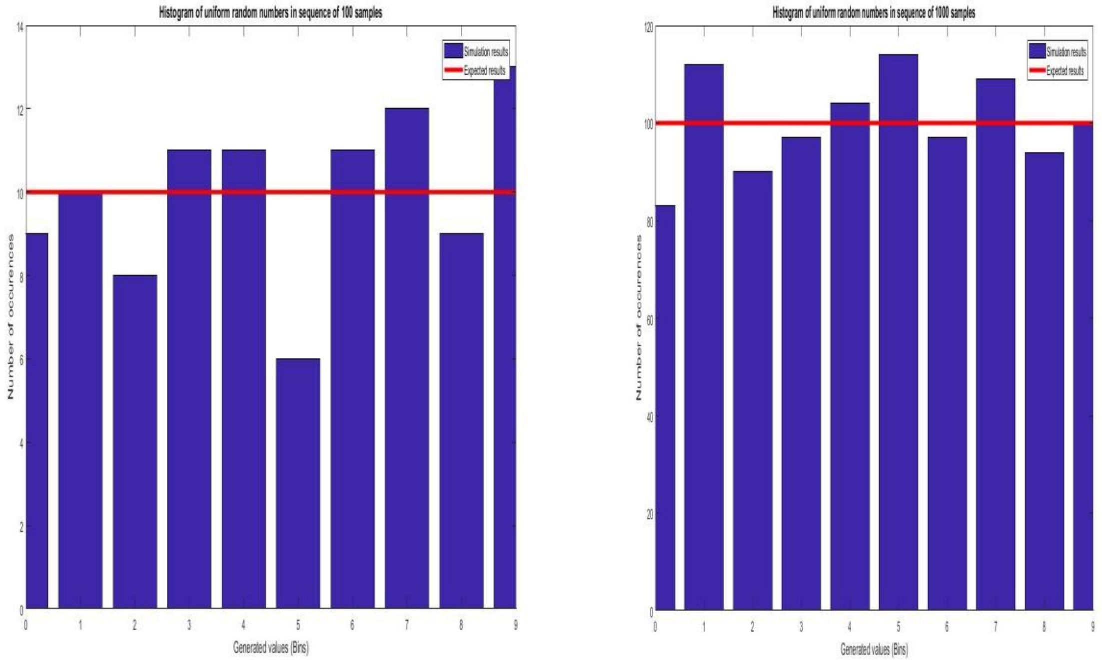

Figure 1 Comparison of simulation results with expected value for 100 and 1000 samples

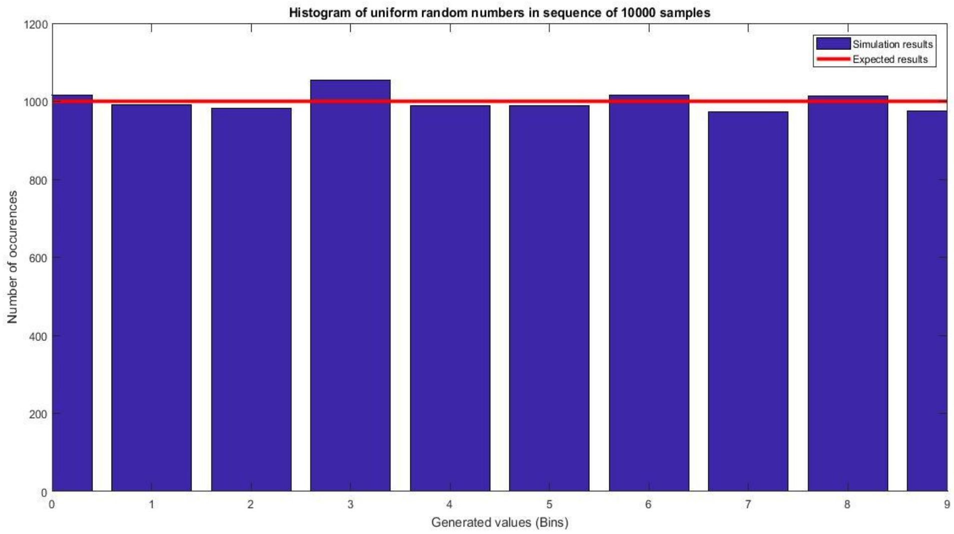

Figure 2 Comparison of simulation results with expected value for 10000 samples

From the plots, we can see that as the sequence size increases from 100 to 10000, the sample becomes more uniformly distributed and number of occurrences from simulation becomes closer to the expected number of occurrences. Hence, law of large number is followed.

Now, we will run some tests for the goodness of fit like Chi-square test, Kolmogorov-Smirnov test to prove that sequence generated by rand function is uniformly distributed between 0 and 1.

Chi-Square Goodness of fit test

We will carry out goodness of fit test to prove that rand function generates sequence of uniform random numbers. For the test, we will consider the sequence of size 10000.

Step 1: Make Hypothesis

H_o (null hypothesis): rand function generates uniform numbers.

H_A (alternative hypothesis): rand function does not generate uniform numbers.

Step 2: Calculate Chi-Square statistical from the sampled data using the following formula

$$x^2 = \Sigma \frac{(Expected(i) - Observed(i))^2}{Expected(i)} \qquad \forall\, i = 0,1,..,9$$

Value	Expected outcomes	Observed outcomes	Chi square contribution
0	1000	1016	0.256
1	1000	1025	0.625
2	1000	1030	0.9
3	1000	1020	0.4
4	1000	991	0.081
5	1000	977	0.529
6	1000	1012	0.144
7	1000	985	0.225
8	1000	957	1.849
9	1000	987	0.169
		Chi-Square Statistical	**5.178**

Table 4 Calculation of Chi-square statistic

Step 3: Calculate degrees of freedom and assume level of significance(α) which denotes the probability of considering null hypothesis false when it is true.

Degree of freedom = 10 – 1 = 9 and α = 0.05

Step 4: From the table of Chi-square distribution, find the p-value which represents degree of freedom as 9 and Chi-square statistical.

p-value = Probability ($x^2 > 5.178$) is between 0.75 to 0.9 by looking at the below table

Percentage Points of the Chi-Square Distribution Source: Google Images

Degrees of Freedom	Probability of a larger value of x [2]								
	0.99	0.95	0.90	0.75	0.50	0.25	0.10	0.05	0.01
9	2.088	3.325	4.168	5.899	8.343	11.39	14.68	16.92	21.67

Conclusion of test: If p-value is less than α, then we reject the null hypothesis. Here p-value > α, so our null hypothesis is correct. Hence, we can conclude that rand function generates uniform numbers.

Kolmogorov-Smirnov Goodness of fit test

The chi-square test is designed for only discrete distributions, so in continuous case the chi-square test statistic is only an approximation. In chi-square test, artificially binning data loses information so should be avoided if possible. The Kolmogorov-Smirnov test (KS test) works for continuous distributions. So, it does not group samples into categories like chi-square test and it is more sensitive on the tails. Hence, KS test makes better use of each sample and is more precise than the chi-square test. KS test basically compares the maximum difference at any point between the CDF of generated sequence to the CDF of expected uniform sequence. This maximum difference can be shown in figure below.

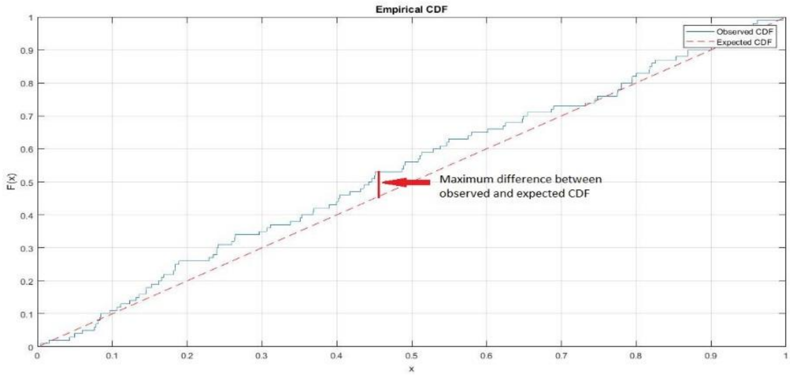

Figure 3 Comparison of Observed CDF with Expected CDF for KS test

To perform KS test, we need to follow below steps considering the sequence of size 10000:

Step 1: Make hypothesis

H_o: Sequence generated is uniformly distributed between 0 and 1.

H_A: Sequence generated is not uniformly distributed between 0 and 1.

Step 2: Sort the sequence in increasing order.

Step 3: Compute KS test statistics D_+ and D_-.

$$D_+ = \max\left(\frac{i}{N} - R_i\right) \qquad 1 \le i \le N$$

$$D_- = \max\left(R_i - \frac{i-1}{N}\right) \qquad 1 \le i \le N$$

Step 4: Compute D=max (D_+, D_-)

Step 5: The critical value Dα for α=0.05 is 0.521. If $D > D_\alpha$ then H_o is rejected.

```
% Kolmogorov-Smirnov test
% Step 2
x=sort(x);
% Step 3
for i=1:size
    temp1(i)=(i/size)-x(i);
    temp2(i)= x(i)-(i-1/size);
end
dplus=max(temp1);
dminus=max(temp2);
% Step 4
d=max(dplus,dminus);
```

D_+	D_-	D
0.0049	-0.9996	0.0049

Conclusion of test: Here, $D < D_\alpha$ by a very big margin and hence there is no possible reason to reject the null hypothesis. So, the sequence generated is uniformly distributed between 0 and 1.

Chi-square test and KS test by MATLAB

We can do chi-square test and KS test in MATLAB using functions *chi2gof* and *kstest* respectively.

Chi-square test:

```
% Using MATLAB function chi2gof to find uniformity
% pd - contains uniform distribution from 0 and 1
% chi2gof function takes arguments as generated sequence and expected
% distribution
% h1 - if 0 then we accept null hypothesis and if 1 we reject null
% hypothesis
% p1 - contains p-value which must be greater than alpha=0.05 to accept
% null hypothesis
pd=makedist('uniform');
[h1,p1]=chi2gof(x,'cdf',pd);
```

h1	p1
0	0.8956

h1 is 0 and p1 > alpha so we accept null hypothesis that rand generates uniform numbers.

KS test:

```
% Using MATLAB function kstest to find uniformity
% kstest function takes arguments as generated sequence and expected
% distribution
% h2 - if 0 then we accept null hypothesis and if 1 we reject null
% hypothesis
% p2 - contains p-value which must be greater than alpha=0.05 to accept
% null hypothesis
[h2,p2]=kstest(x,'cdf',pd)
```

h2	p2
0	**0.5160**

h2 is 0 and p2 > alpha so we accept null hypothesis that sequence generated is uniformly distributed between 0 and 1.

References

[1] Sureiman Onchiri, *“Conceptual model on application of chi-square test in education and social sciences”*, Academic Journals, Vol. 8(15), pp. 1231-1241, 10 August, 2013

[2] Frank J. Massey Jr., *“The Kolmogorov-Smirnov Test for Goodness of Fit”*, Journal of the American Statistical Association, Vol. 46 , Issue. 253, Pages 68-78,1951

[3] Sheldon M. Ross, *“Simulation”*