

Simulation

“Discrete-Event System Simulation”

Dr. Mesut Güneş

Chapter 5

Random-Number Generation

Purpose & Overview

- Discuss the generation of random numbers.
- Introduce the subsequent testing for randomness:
 - Frequency test
 - Autocorrelation test.

Properties of Random Numbers

- Two important statistical properties:
 - Uniformity
 - Independence.
- Random Number, R_i , must be independently drawn from a uniform distribution with pdf:

$$f(x) = \begin{cases} 1, & 0 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

$$E(R) = \int_0^1 x dx = \frac{x^2}{2} \Big|_0^1 = \frac{1}{2}$$

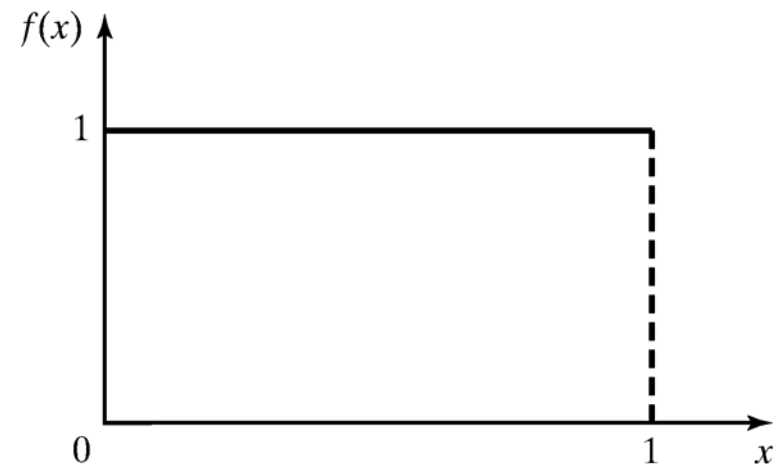


Figure: pdf for random numbers

Generation of Pseudo-Random Numbers

- “Pseudo”, because generating numbers using a known method removes the potential for true randomness.
- Goal: To produce a sequence of numbers in $[0, 1]$ that simulates, or imitates, the ideal properties of random numbers (RN).
- Important considerations in RN routines:
 - Fast
 - Portable to different computers
 - Have sufficiently long cycle
 - Replicable
 - Closely approximate the ideal statistical properties of uniformity and independence.

Techniques for Generating Random Numbers

- Linear Congruential Method (LCM).
- Combined Linear Congruential Generators (CLCG).
- Random-Number Streams.

Linear Congruential Method

- To produce a sequence of integers, X_1, X_2, \dots between 0 and $m-1$ by following a recursive relationship:

$$X_{i+1} = (aX_i + c) \bmod m, \quad i = 0, 1, 2, \dots$$

The multiplier

The increment

The modulus

- The selection of the values for a , c , m , and X_0 drastically affects the statistical properties and the cycle length.
- The random integers are being generated $[0, m-1]$, and to convert the integers to random numbers:

$$R_i = \frac{X_i}{m}, \quad i = 1, 2, \dots$$

Linear Congruential Method – Example

- Use $X_0 = 27$, $a = 17$, $c = 43$, and $m = 100$.
- The X_i and R_i values are:

$$X_1 = (17 \cdot 27 + 43) \bmod 100 = 502 \bmod 100 = 2, \quad R_1 = 0.02;$$

$$X_2 = (17 \cdot 2 + 43) \bmod 100 = 77, \quad R_2 = 0.77;$$

$$X_3 = (17 \cdot 77 + 43) \bmod 100 = 52, \quad R_3 = 0.52;$$

$$X_4 = (17 \cdot 52 + 43) \bmod 100 = 27, \quad R_3 = 0.27;$$

...

Characteristics of a Good Generator

- **Maximum Density**
 - Such that the values assumed by R_i , $i = 1, 2, \dots$, leave no large gaps on $[0, 1]$
 - Problem: Instead of continuous, each R_i is discrete
 - Solution: a very large integer for modulus m
 - Approximation appears to be of little consequence

- **Maximum Period**
 - To achieve maximum density and avoid cycling.
 - Achieve by: proper choice of a , c , m , and X_0 .

- **Most digital computers use a binary representation of numbers**
 - Speed and efficiency are aided by a modulus, m , to be (or close to) a power of 2.

Random-Numbers in Java

- Defined in `java.util.Random`

```
private final static long multiplier = 0x5DEECE66DL;
private final static long addend = 0xBL;
private final static long mask = (1L << 48) - 1;

protected int next(int bits) {
    long oldseed, nextseed;
    ...
    oldseed = seed.get();
    nextseed = (oldseed * multiplier + addend) & mask;
    ...
    return (int)(nextseed >>> (48 - bits));
}
```

Combined Linear Congruential Generators

- Reason: Longer period generator is needed because of the increasing complexity of simulated systems.
- Approach: Combine two or more multiplicative congruential generators.
- Let $X_{i,1}, X_{i,2}, \dots, X_{i,k}$ be the i -th output from k different multiplicative congruential generators.
 - The j -th generator:
 - Has prime modulus m_j and multiplier a_j and period is $m_j - 1$
 - Produces integers $X_{i,j}$ is approx \sim Uniform on integers in $[1, m_j - 1]$
 - $W_{i,j} = X_{i,j} - 1$ is approx \sim Uniform on integers in $[0, m_j - 2]$

Combined Linear Congruential Generators

- Suggested form:

$$X_i = \left(\sum_{j=1}^k (-1)^{j-1} X_{i,j} \right) \bmod m_1 - 1$$

The coefficient:
Performs the
subtraction $X_{i,1-1}$

$$\text{Hence, } R_i = \begin{cases} \frac{X_i}{m_1}, & X_i > 0 \\ \frac{m_1 - 1}{m_1}, & X_i = 0 \end{cases}$$

- The maximum possible period is: $P = \frac{(m_1 - 1)(m_2 - 1) \dots (m_k - 1)}{2^{k-1}}$

Combined Linear Congruential Generators

- Example: For 32-bit computers, combining $k = 2$ generators with $m_1 = 2,147,483,563$, $a_1 = 40,014$, $m_2 = 2,147,483,399$ and $a_2 = 20,692$. The algorithm becomes:

Step 1: Select seeds

- $X_{1,0}$ in the range $[1, 2,147,483,562]$ for the 1st generator
- $X_{2,0}$ in the range $[1, 2,147,483,398]$ for the 2nd generator.

Step 2: For each individual generator,

$$X_{1,j+1} = 40,014 X_{1,j} \bmod 2,147,483,563$$

$$X_{2,j+1} = 40,692 X_{1,j} \bmod 2,147,483,399.$$

Step 3: $X_{j+1} = (X_{1,j+1} - X_{2,j+1}) \bmod 2,147,483,562$.

Step 4: Return

$$R_{j+1} = \begin{cases} \frac{X_{j+1}}{2,147,483,563}, & X_{j+1} > 0 \\ \frac{2,147,483,562}{2,147,483,563}, & X_{j+1} = 0 \end{cases}$$

Step 5: Set $j = j+1$, go back to step 2.

- Combined generator has period: $(m_1 - 1)(m_2 - 1)/2 \sim 2 \times 10^{18}$

Random-Numbers in Excel 2003

- In Excel 2003 new Random Number Generator

$$X, Y, Z \in \{1, \dots, 30000\}$$

$$X = X \cdot 171 \bmod 30269$$

$$Y = Y \cdot 172 \bmod 30307$$

$$Z = Z \cdot 170 \bmod 30323$$

$$R = \left(\frac{X}{30269} + \frac{Y}{30307} + \frac{Z}{30323} \right) \bmod 1.0$$

- It is stated that this method produces more than 10^{13} numbers

Random-Numbers Streams

- **The seed for a linear congruential random-number generator:**
 - Is the integer value X_0 that initializes the random-number sequence.
 - Any value in the sequence can be used to “seed” the generator.
- **A random-number stream:**
 - Refers to a starting seed taken from the sequence X_0, X_1, \dots, X_P .
 - If the streams are b values apart, then stream i could defined by starting seed:
$$S_i = X_{b(i-1)} \quad i = 1, 2, \dots, \left\lfloor \frac{P}{b} \right\rfloor$$
 - Older generators: $b = 10^5$; Newer generators: $b = 10^{37}$.
- **A single random-number generator with k streams can act like k distinct virtual random-number generators**
- **To compare two or more alternative systems.**
 - Advantageous to dedicate portions of the pseudo-random number sequence to the same purpose in each of the simulated systems.

Tests for Random Numbers

- **Two categories:**

- Testing for uniformity:

$$H_0: R_i \sim U[0, 1]$$

$$H_1: R_i \not\sim U[0, 1]$$

- Failure to reject the null hypothesis, H_0 , means that evidence of non-uniformity has not been detected.

- Testing for independence:

$$H_0: R_i \sim \text{independently}$$

$$H_1: R_i \not\sim \text{independently}$$

- Failure to reject the null hypothesis, H_0 , means that evidence of dependence has not been detected.

- **Level of significance α , the probability of rejecting H_0 when it is true:**
$$\alpha = P(\text{reject } H_0 \mid H_0 \text{ is true})$$

Tests for Random Numbers

- **When to use these tests:**
 - If a well-known simulation languages or random-number generators is used, it is probably unnecessary to test
 - If the generator is not explicitly known or documented, e.g., spreadsheet programs, symbolic/numerical calculators, tests should be applied to many sample numbers.

- **Types of tests:**
 - Theoretical tests: evaluate the choices of m , a , and c without actually generating any numbers
 - Empirical tests: applied to actual sequences of numbers produced.
 - Our emphasis.

Frequency Tests

- Test of uniformity
- Two different methods:
 - Kolmogorov-Smirnov test
 - Chi-square test

Kolmogorov-Smirnov Test

- Compares the continuous cdf, $F(x)$, of the uniform distribution with the empirical cdf, $S_N(x)$, of the N sample observations.
 - We know: $F(x) = x, 0 \leq x \leq 1$
 - If the sample from the RN generator is R_1, R_2, \dots, R_N , then the empirical cdf, $S_N(x)$ is:

$$S_N(x) = \frac{\text{Number of } R_i \text{ where } R_i \leq x}{N}$$

- Based on the statistic: $D = \max |F(x) - S_N(x)|$
 - Sampling distribution of D is known
- A more powerful test, recommended.

Kolmogorov-Smirnov Test

- The test consists of the following steps
 - Step 1: Rank the data from smallest to largest
 $R_{(1)} \leq R_{(2)} \leq \dots \leq R_{(N)}$

- Step 2: Compute

$$D^+ = \max_{1 \leq i \leq N} \left\{ \frac{i}{N} - R_{(i)} \right\}$$

$$D^- = \max_{1 \leq i \leq N} \left\{ R_{(i)} - \frac{i-1}{N} \right\}$$

- Step 3: Compute $D = \max(D^+, D^-)$
- Step 4: Get D_α for the significance level α
- Step 5: If $D \leq D_\alpha$ accept, otherwise reject H_0

Degrees of Freedom (N)	$D_{0.10}$	$D_{0.05}$	$D_{0.01}$
1	0.950	0.975	0.995
2	0.776	0.842	0.929
3	0.642	0.708	0.828
4	0.564	0.624	0.733
5	0.510	0.565	0.669
6	0.470	0.521	0.618
7	0.438	0.486	0.577
8	0.411	0.457	0.543
9	0.388	0.432	0.514
10	0.368	0.410	0.490
11	0.352	0.391	0.468
12	0.338	0.375	0.450
13	0.325	0.361	0.433
14	0.314	0.349	0.418
15	0.304	0.338	0.404
16	0.295	0.328	0.392
17	0.286	0.318	0.381
18	0.278	0.309	0.371
19	0.272	0.301	0.363
20	0.264	0.294	0.356
25	0.24	0.27	0.32
30	0.22	0.24	0.29
35	0.21	0.23	0.27
Over 35	$\frac{1.22}{\sqrt{N}}$	$\frac{1.36}{\sqrt{N}}$	$\frac{1.63}{\sqrt{N}}$

Kolmogorov-Smirnov Test

- Example: Suppose $N=5$ numbers: $0.44, 0.81, 0.14, 0.05, 0.93$.

i	1	2	3	4	5	
Step 1: $R_{(i)}$	0.05	0.14	0.44	0.81	0.93	
i/N	0.20	0.40	0.60	0.80	1.00	
Step 2:	$i/N - R_{(i)}$	0.15	0.26	0.16	-	0.07
	$R_{(i)} - (i-1)/N$	0.05	-	0.04	0.21	0.13

Arrange $R_{(i)}$ from smallest to largest

$D^+ = \max \{i/N - R_{(i)}\}$

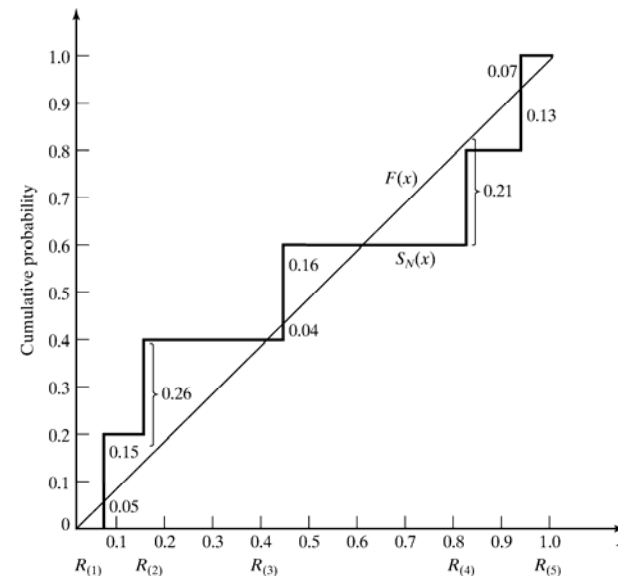
$D^- = \max \{R_{(i)} - (i-1)/N\}$

Step 3: $D = \max(D^+, D^-) = 0.26$

Step 4: For $\alpha = 0.05$,

$$D_\alpha = 0.565 > D$$

Hence, H_0 is not rejected.



Chi-square test

- Chi-square test uses the sample statistic:

The diagram shows the chi-square test formula:
$$X_0^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i}$$
 Three callout boxes provide definitions: 1. A box pointing to the upper limit 'n' of the summation: "n is the # of classes". 2. A box pointing to the denominator 'E_i': "E_i is the expected # in the i-th class". 3. A box pointing to the numerator '(O_i - E_i)^2': "O_i is the observed # in the i-th class".

- Approximately the chi-square distribution with $n-1$ degrees of freedom
- For the uniform distribution, E_i the expected number in each class is:

$$E_i = \frac{N}{n}, \quad \text{where } N \text{ is the total \# of observation}$$

- Valid only for large samples, e.g. $N \geq 50$

Chi-square test

- Example
- 100 numbers from [0,1]
- $\alpha=0.05$
- 10 intervals
- $X^2_{0.05,9}=16.9$
- Accept, since
 - $X^2_0=11.2 < X^2_{0.05,9}$

Interval	Upper Limit	O _i	E _i	O _i -E _i	(O _i -E _i) ²	(O _i -E _i) ² /E _i
1	0.1	10	10	0	0	0
2	0.2	9	10	-1	1	0.1
3	0.3	5	10	-5	25	2.5
4	0.4	6	10	-4	16	1.6
5	0.5	16	10	6	36	3.6
6	0.6	13	10	3	9	0.9
7	0.7	10	10	0	0	0
8	0.8	7	10	-3	9	0.9
9	0.9	10	10	0	0	0
10	1.0	14	10	4	16	1.6
S		100	100	0	0	11.2

$X^2_0=11.2$

Tests for Autocorrelation

- Autocorrelation is concerned with dependence between numbers in a sequence
- Example:

0.12	0.01	0.23	0.28	0.89	0.31	0.64	0.28	0.83	0.93
0.99	0.15	0.33	0.35	0.91	0.41	0.60	0.27	0.75	0.88
0.68	0.49	0.05	0.43	0.95	0.58	0.19	0.36	0.69	0.87

- Numbers at 5-th, 10-th, 15-th, ... are very similar
- Numbers can be
 - Low
 - High
 - Alternating

Tests for Autocorrelation

- Testing the autocorrelation between every m numbers (m is a.k.a. the lag), starting with the i -th number

- The autocorrelation ρ_{im} between numbers: $R_i, R_{i+m}, R_{i+2m}, \dots, R_{i+(M+1)m}$
- M is the largest integer such that $i + (M + 1)m \leq N$

- Hypothesis:

$$H_0 : \rho_{im} = 0, \quad \text{if numbers are independent}$$

$$H_1 : \rho_{im} \neq 0, \quad \text{if numbers are dependent}$$

- If the values are uncorrelated:

- For large values of M , the distribution of the estimator of ρ_{im} , denoted $\hat{\rho}_{im}$ is approximately normal.

Tests for Autocorrelation

- Test statistics is:
$$Z_0 = \frac{\hat{\rho}_{im}}{\hat{\sigma}_{\hat{\rho}_{im}}}$$

- Z_0 is distributed normally with mean = 0 and variance = 1, and:

$$\hat{\rho}_{im} = \frac{1}{M+1} \left[\sum_{k=0}^M R_{i+km} \cdot R_{i+(k+1)m} \right] - 0.25$$

$$\hat{\sigma}_{\hat{\rho}_{im}} = \frac{\sqrt{13M+7}}{12(M+1)}$$

- If $\rho_{im} > 0$, the subsequence has positive autocorrelation
 - High random numbers tend to be followed by high ones, and vice versa.
- If $\rho_{im} < 0$, the subsequence has negative autocorrelation
 - Low random numbers tend to be followed by high ones, and vice versa.

Example

- Test whether the 3^{rd} , 8^{th} , 13^{th} , and so on, for the numbers on Slide 24.
 - Hence, $\alpha = 0.05$, $i = 3$, $m = 5$, $N = 30$, and $M = 4$

$$\hat{\rho}_{35} = \frac{1}{4+1} \left[(0.23)(0.28) + (0.28)(0.33) + (0.33)(0.27) \right. \\ \left. + (0.27)(0.05) + (0.05)(0.36) \right] - 0.25 \\ = -0.1945$$

$$\sigma_{\hat{\rho}_{35}} = \frac{\sqrt{13(4)+7}}{12(4+1)} = 0.128$$

$$Z_0 = -\frac{0.1945}{0.1280} = -1.516$$

- $z_{0.025} = 1.96$ hence, the hypothesis is not rejected.

Shortcomings

- The test is not very sensitive for small values of M , particularly when the numbers being tested are on the low side.
- Problem when “fishing” for autocorrelation by performing numerous tests:
 - If $\alpha = 0.05$, there is a probability of 0.05 of rejecting a true hypothesis.
 - If 10 independence sequences are examined,
 - The probability of finding no significant autocorrelation, by chance alone, is $0.95^{10} = 0.60$.
 - Hence, the probability of detecting significant autocorrelation when it does not exist = 40%

Summary

- **In this chapter, we described:**
 - Generation of random numbers
 - Testing for uniformity and independence

- **Caution:**
 - Even with generators that have been used for years, some of which still in used, are found to be inadequate.
 - This chapter provides only the basic
 - Also, even if generated numbers pass all the tests, some underlying pattern might have gone undetected.