

# Chapter 7

## Random-Number Generation

Banks, Carson, Nelson & Nicol  
*Discrete-Event System Simulation*

# 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:

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

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

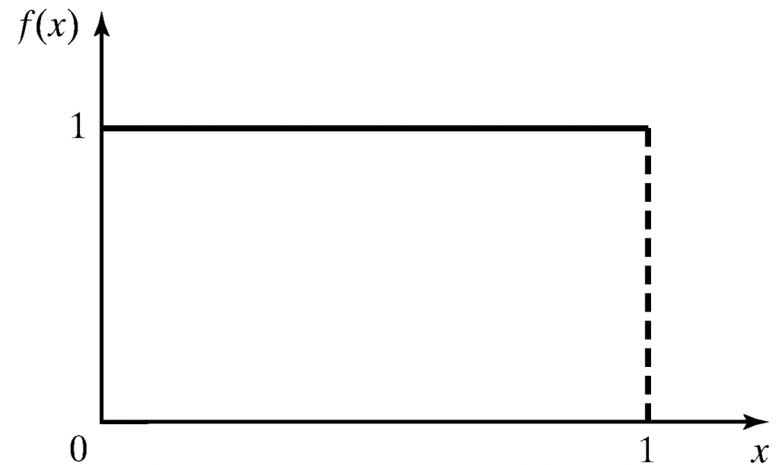


Figure: pdf for random numbers

# 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

[Techniques]

- 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:

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

# Example

[LCM]

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

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

$$X_2 = (17*2+\textcolor{red}{32}) \bmod 100 = 77, \quad R_2 = 0.77;$$

$$X_3 = (17*77+\textcolor{red}{32}) \bmod 100 = 52, \quad R_3 = 0.52;$$

...

# Characteristics of a Good Generator

[LCM]

## ■ 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.



# Combined Linear Congruential Generators

## [Techniques]

- Reason: Longer period generator is needed because of the increasing complexity of stimulated systems.
- Approach: Combine two or more multiplicative congruential generators.
- Let  $X_{i,1}, X_{i,2}, \dots, X_{i,k}$  be the  $i^{\text{th}}$  output from  $k$  different multiplicative congruential generators.
  - The  $j^{\text{th}}$  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  $[1, m-1]$
    - $W_{i,j} = X_{i,j} - 1$  is approx  $\sim$  Uniform on integers in  $[1, m-2]$

# Combined Linear Congruential Generators

## [Techniques]

- Suggested form:

$$X_i = \left( \sum_{j=1}^k (-1)^{j-1} X_{i,j} \right) \bmod m_1 - 1 \quad \text{Hence, } R_i = \begin{cases} \frac{X_i}{m_1}, & X_i \succ 0 \\ \frac{m_1 - 1}{m_1}, & X_i = 0 \end{cases}$$

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

- The maximum possible period is:

$$P = \frac{(m_1 - 1)(m_2 - 1) \dots (m_k - 1)}{2^{k-1}}$$

# Combined Linear Congruential Generators

## [Techniques]

- Example: For 32-bit computers, L'Ecuyer [1988] suggests 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 1<sup>st</sup> generator
- $X_{2,0}$  in the range  $[1, 2,147,483,398]$  for the 2<sup>nd</sup> generator.

Step 2: For each individual generator,

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

$$X_{2,j+1} = 20,692 X_{2,j} \bmod 2,147,483,399.$$

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

Step 4: Return

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

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 Streams

[Techniques]

- 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:  
$$S_i = X_{b(i-1)}$$
  - 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  $\alpha$ , the probability of rejecting  $H_0$  when it is true:  
$$\alpha = P(\text{reject } H_0 | H_0 \text{ is true})$$

# 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

[Tests for RN]

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

# Kolmogorov-Smirnov Test

[Frequency 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:  $F(x) = x, \quad 0 \leq x \leq 1$

- If the sample from the RN generator is  $R_1, R_2, \dots, R_N$ , then the empirical cdf,  $S_N(x)$  is:

$$S_N(x) = \frac{\text{number of } R_1, R_2, \dots, R_n \text{ which are } \leq x}{N}$$

- Based on the statistic:  $D = \max |F(x) - S_N(x)|$ 
  - Sampling distribution of  $D$  is known (a function of  $N$ , tabulated in Table A.8.)
- A more powerful test, recommended.



# Kolmogorov-Smirnov Test

[Frequency Test]

- Example: Suppose 5 generated numbers are 0.44, 0.81, 0.14, 0.05, 0.93.

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
$i/N - R_{(i)}$	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

Step 2:

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

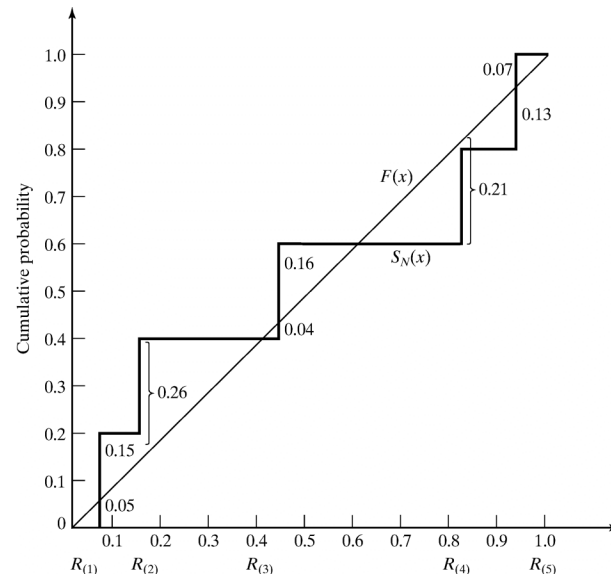
$D^- = \max \{R_{(i)} - (i-1)/N\}$

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

[Frequency Test]

- Chi-square test uses the sample statistic:

The diagram shows the Chi-square test formula: 
$$X_0^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i}$$
 with three callouts: 

- A callout from  $n$  states:  $n$  is the # of classes
- A callout from  $E_i$  states:  $E_i$  is the expected # in the  $i^{th}$  class
- A callout from  $O_i$  states:  $O_i$  is the observed # in the  $i^{th}$  class

- Approximately the chi-square distribution with  $n-1$  degrees of freedom (where the critical values are tabulated in Table A.6)
- For the uniform distribution,  $E_i$ , the expected number in the 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.  $N \geq 50$

# Tests for Autocorrelation

[Tests for RN]

- Testing the autocorrelation between every  $m$  numbers ( $m$  is a.k.a. the lag), starting with the  $i^{th}$  number
  - The autocorrelation  $\rho_{im}$  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  $\rho_{im}$ , denoted  $\hat{\rho}_{im}$  is approximately normal.

# Tests for Autocorrelation

[Tests for RN]

- 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:

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

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

- 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 for Autocorrelation]

- Test whether the 3<sup>rd</sup>, 8<sup>th</sup>, 13<sup>th</sup>, and so on, for the following output on P. 265.

□ Hence,  $\alpha = 0.05$ ,  $i = 3$ ,  $m = 5$ ,  $N = 30$ , and  $M = 4$

$$\begin{aligned}\tilde{n}_{35} &= \frac{1}{4+1} \left[ (0.23)(0.28) + (0.25)(0.33) + (0.33)(0.27) \right. \\ &\quad \left. + (0.28)(0.05) + (0.05)(0.36) \right] - 0.25 \\ &= -0.1945\end{aligned}$$

$$\hat{\sigma}_{\tilde{n}_{35}} = \frac{\sqrt{13(4)+7}}{12(4+1)} = 0.128$$

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

- From Table A.3,  $z_{0.025} = 1.96$ . Hence, the hypothesis is not rejected.

# Shortcomings

[Test for Autocorrelation]

- The test is not very sensitive for small values of  $M$ , particularly when the numbers being tests 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.