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:

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

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

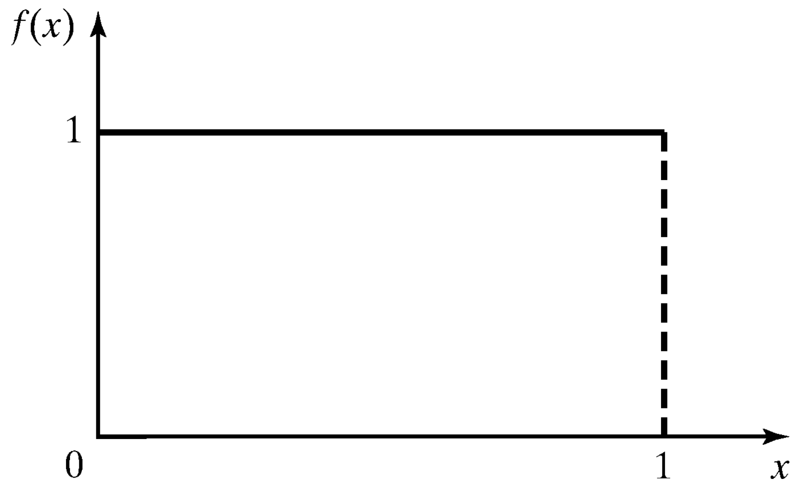


Figure: pdf for random numbers

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

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

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

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:

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

Example [LCM]

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

$X_1 = (17*27+43) \text{ mod } 100 = 502 \text{ mod } 100 = 2,$ $R_1 = 0.02;$

$X_2 = (17*2+43) \text{ mod } 100 = 77,$ $R_2 = 0.77;$

$X_3 = (17*77+43) \text{ mod } 100 = 52,$ $R_3 = 0.52;$

…

Characteristics of a Good Generator

[LCM]

- Maximum Density
  - Such that the values assumed by R_i, $i = 1,2,\ldots$, 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 $X_{i,1}, X_{i,2}, \ldots, X_{i,k}$, 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 ~ Uniform on integers in [$1, m_j$-1]
    - $W_{i,j} = X_{i,j} - 1$ is approx ~ Uniform on integers in [$0, m$-2]

Combined Linear Congruential Generators

[Techniques]

- Suggested form:

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

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

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

- The maximum possible period is:

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

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 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

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

  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 $X_0, X_1, \ldots, X_P$.
  - 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: $R_i \nsim U[0,1]$

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

    H_0: $R_i \sim$ independently

    H_1: $R_i \nsim$ 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: $\alpha = P(\text{reject } H_0 | H_0 \text{ is true})$

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 [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: $$F(x) = x, \quad 0 \le x \le 1$$
  - If the sample from the RN generator is $R_1, R_2, \ldots, R_N$, then the empirical cdf, $S_N(x)$ is:
$$S_N(x) = \frac{\text{number of } R_1, R_2, \ldots, R_n \text{ which are} \le x}{N}$$
- 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

Step 2: (rows $i/N - R_{(i)}$ and $R_{(i)} - (i-1)/N$)

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

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

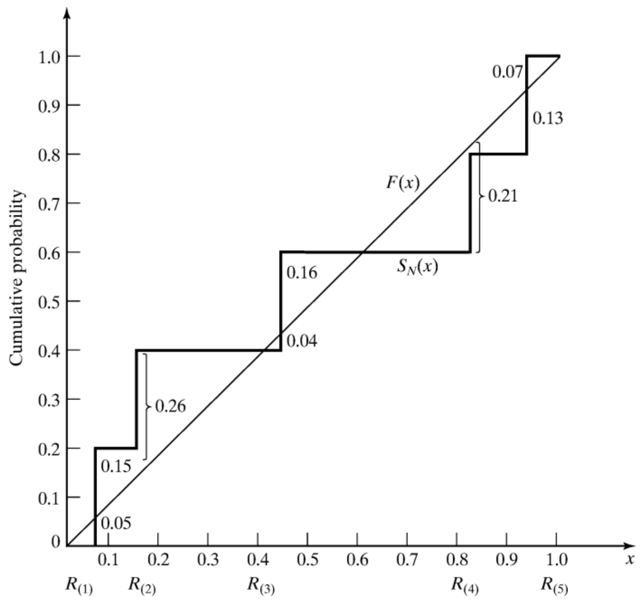
$D^- = max\ \{R_{(i)} - (i-1)/N\}$

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 [Frequency Test]

- Chi-square test uses the sample statistic:

n is the # of classes

$$X_0^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

E_i is the expected # in the i^{th} class

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:

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

- Valid only for large samples, e.g. N >= 50

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}, $R_{i+(M+1)m}$
  - M is the largest integer such that $i+(M+1)m \leq N$
- Hypothesis:

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

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

- 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:

$$Z_0 = \frac{\hat{\rho}_{im}}{\hat{\sigma}_{\hat{\rho}_{im}}}$$

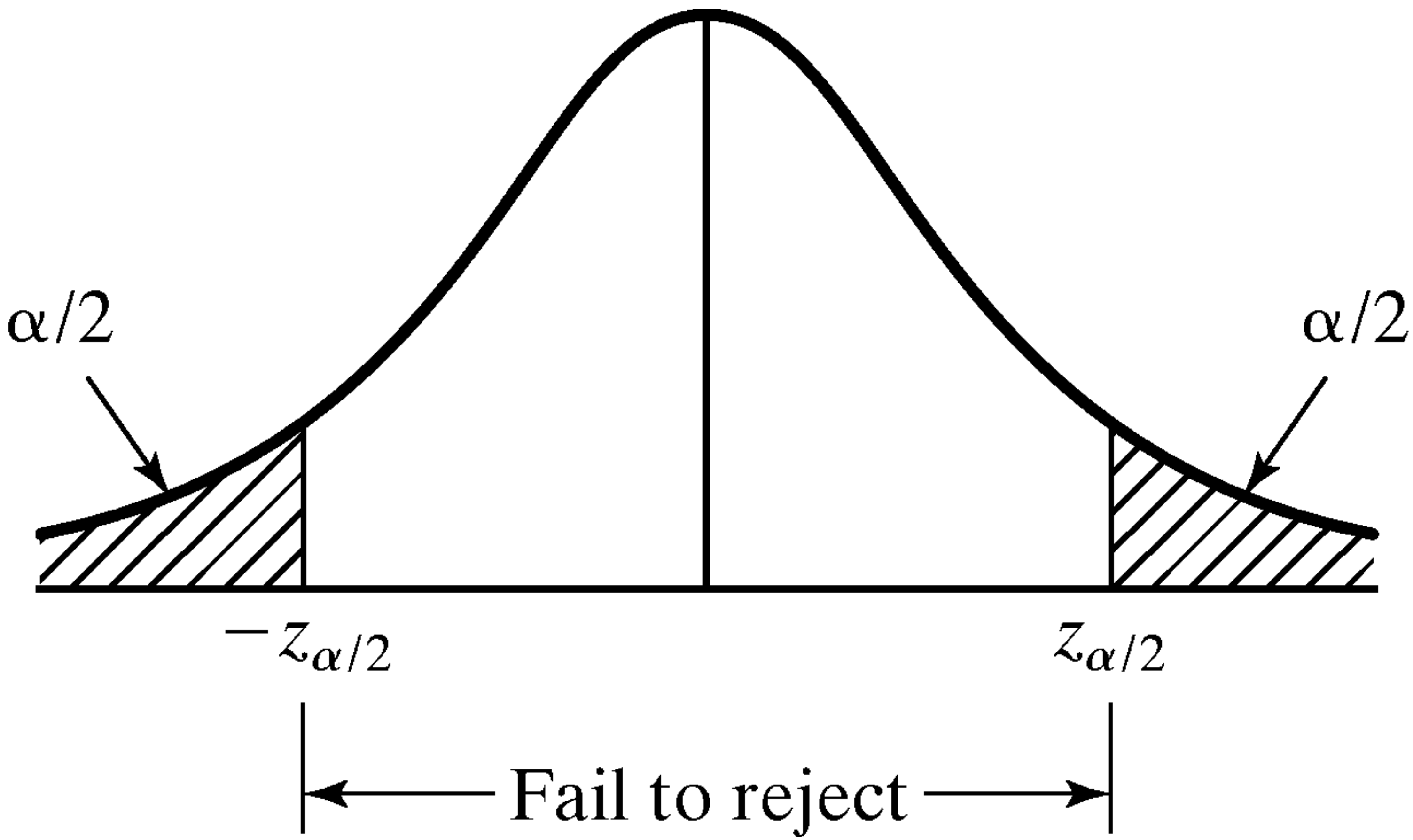
  - Z_0 is distributed normally with mean = *0* and variance = *1*, and:

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

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

- 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.

Normal Hypothesis Test



Example

[Test for Autocorrelation]

- Test whether the *3rd, 8th, 13th*, and so on, for the following output on P. 265.
  - Hence, α = *0.05, i = 3, m = 5, N = 30,* and M = 4

$$\hat{\rho}_{35} = \frac{1}{4+1}\begin{bmatrix}(0.23)(0.28)+(0.25)(0.33)+(0.33)(0.27) \\ +(0.28)(0.05)+(0.05)(0.36)\end{bmatrix} - 0.25$$

$$= -0.1945$$

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

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

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

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.