Lecture Notes 5
Random-Number Generation

- Introduction
- Types of Random Number Generators (RNGs)
- Linear Congruential Generators
- Evaluating RNGs
- Be sure to check course webpage for reading assignment and HW

Introduction

- Heart of any simulation is its random-number generator (RNG), since this is the only source of variability.
- We'll cover simplest and most widely used method of generating random numbers and discuss statistical evaluation of RNG.
- Most commercial software providing random numbers use decent procedures.
- But there are famous cases of bad generators getting widespread use.
- Since a bad generator can nullify a simulation study, important to touch on major issues.

Types of Random Number Generators

Ideal RNG generates sequence $U_1, U_2, \ldots$ with following two properties:

1. Each U_i is equally likely to fall anywhere in interval $[0, 1]$.
2. Values of U_i's are independent of each other.

Remarks:

- Choice of uniform distribution is primarily for computational convenience.
- Statistically, second property is more critical one.

There are three types of sources of random numbers:

1. *Physical devices.*
   - Ticks generated by Geiger counter exposed to radiation are in some sense random.
   - Thermal noise in electronic components is random.
   - last digit in (very precise) computer clock when command is executed is close to being random.

   These types of phenomena can be exploited to generate truly random numbers.

2. *Tables.*
   - Thick volume published by RAND Corporation, titled *A Million Random Digits with 100,000 Normal Deviates*
   - Historically, this was only practical source of random numbers.
   - They were generated using physical device.
   - Such tables are now available on web.

3. *Number-Theoretic Methods.*
   - By far most widely used generators.
   - Technically, not random at all; in fact, every number in sequence is completely determined by first one.
   - But sequence is sufficiently complex to appear random and to pass many statistical tests.
   - Sometimes called **pseudorandom number generators**.
   - All modern simulation software uses RNG of this type.

In addition to statistical considerations, two practical features are important in an RNG:

- possible to reproduce a sequence of random numbers;
- also possible to obtain entirely different sequence.

Linear Congruential Generators (LCGs)

- Common type of RNG.
- Remarkably simple and very effective.
- Works as follows:
  - Sequence starts with *seed* Z_0, often specified by user.
  - At each call to RNG, algorithm computes
    $$Z_i = (aZ_{i-1} + c) \bmod m.$$
    Operation "mod m" takes remainder after division by m.
  - **Example:** if $Z_i = (8Z_{i-1} + 5) \bmod 10$ with $Z_0 = 2$,
    $$\begin{aligned} Z_1 &= (8 * 2 + 5) \bmod 10 = 21 \bmod 10 = 1, \\ Z_2 &= (8 * 1 + 5) \bmod 10 = 13 \bmod 10 = 3, \\ &\vdots \end{aligned}$$

- Recall $Z_i = (aZ_{i-1} + c) \bmod m$
  - a is the *multiplier*
  - c is the *constant* or *increment*
  - m is the *modulus*
  - Z_0 is the *seed*
- Seed Z_0 should be chosen to be integer such that $0 \leq Z_0 < m$.
- a, c, m should all be non-negative integers satisfying
  - $m > 0$
  - $a < m$
  - $c < m$

- Only possible values that can result after mod m operation are $0, 1, \ldots, m-1$.
  - Numbers $Z_1, Z_2, \ldots$ generated through
    $$Z_i = (aZ_{i-1} + c) \bmod m.$$
    can only take these values.
- Obtain number $U_i \in [0, 1)$ by setting
  $$U_i = \frac{Z_i}{m}.$$
- If a, c, and m are chosen carefully
  - sequence $U_1, U_2, \ldots$ appears to be random, though there is no randomness at all! ... Pseudorandom.

Properties of LCGs

- Since each U_{i+1} is completely determined by U_i, ideal property of independence cannot be satisfied exactly
  - Statistical tests help check if numbers "look" independent.
- Before proceeding to tests, let us first examine a few general properties of LCGs.
  1. Only possible values generated by LCG with modulus m are $\{ 0, 1/m, 2/m, \ldots, (m-1)/m \}$.
     - Thus, U_i's cannot be truly uniform on $[0, 1]$.
     - However, if m is chosen large, then the fact that U_i's are discrete should not be a cause of concern.
     - In any case, a computer can only represent a finite number of numbers, so ideal of true uniformity is unattainable.

  2. Sequence of U_i's eventually repeats.
     - Because U_i's can take only finitely many values, sequence $U_1, U_2, \ldots$ must eventually repeat a number.
     - Once a number is repeated, entire rest of sequence is repeated.
     - Thus, all LCGs generate *periodic* sequences.
     - Values of a, c, m should be chosen to obtain a long period.
     - Clearly, longest possible period is m.
       - LCG with period equal to its modulus is said to have *full period.*

Theorem: An LCG has full period if and only if following three conditions hold:

(a) The only positive integer that divides both m and c is 1; i.e., m and c are relatively prime.

(b) If q is a prime number (i.e., divisible by only itself and 1) that divides m, then q divides $a - 1$.

(c) If 4 divides m, then 4 divides $a - 1$.

Remarks:

- Since seed Z_0 determines sequence, we can either
  - reproduce a sequence (using same seed) or
  - obtain new sequence (using different seed).

  Of course, any two full-period sequences will eventually overlap.

Example: Consider LCG

$$Z_i = (aZ_{i-1} + c) \bmod m = (9Z_{i-1}) \bmod 32,$$

so $m = 32$, $c = 0$, $a = 9$. Let $Z_0 = 1$.

i	Z_{i-1}	$9Z_{i-1}$	$Z_i = (9Z_{i-1}) \bmod 32$
1	1	9	9
2	9	81	17
3	17	153	25
4	25	225	1
5	1	9	9
⋮	⋮	⋮	⋮

- This generator has a period of only 4.
- If we choose $Z_0 = 2$ instead, then

$$Z_1 = 18, \quad Z_2 = 2, \quad Z_3 = 18, \ldots,$$

which has a period of only 2.

- Spreadsheets, such as Excel, probably use RNGs of this type, though they do not reveal their methods.
- In the 1980s, one reasonable LCG used

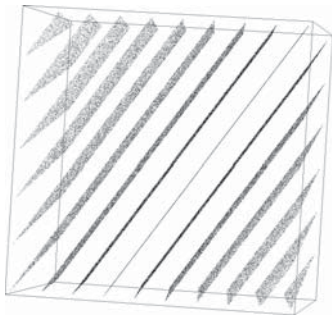
$$a = 7^5 = 16{,}807, \qquad c = 0, \qquad m = 2^{31} - 1.$$

This is a full-period generator ($m \approx 2.1 \times 10^9$).

- Even carefully designed LCGs can turn out to have unexpected bad properties.
  - Most famous case is generator called RANDU with

$$a = 65{,}539, \qquad c = 0, \qquad m = 2^{31},$$

which was widely used before problems discovered.
  - Patterns appear when consecutive triples (U_n, U_{n+1}, U_{n+2}), $n = 1, 4, 7, \ldots$, are plotted as points in 3-dimensional space.



For RANDU, each triple lies on one of only 15 planes.

- In fact, every LCG has a similar problem.
  - Consider any LCG with modulus m.
  - d-tuples $(U_n, U_{n+1}, \ldots, U_{n+d-1})$ lie on at most $(d!m)^{1/d}$ hyperplanes in unit hypercube in $\mathbf{R}^d$.
  - Ref: G. Marsaglia (1968), Random numbers fall mainly in the planes, *Proc. Natl. Acad. Sci.*, 61:25–28.

Other Types of Generators

- More general congruences

$$Z_i = g(Z_{i-1}, Z_{i-2}, \ldots, Z_{i-k}) (\text{mod } m)$$

- Popular example: Tausworthe generator

$$Z_i = (c_1 Z_{i-1} + c_2 Z_{i-2} + \cdots + c_q Z_{i-q}) (\text{mod } 2)$$

generates sequences of bits

- Arena uses a *combined multiple recursive generator* (CMRG)

$$A_n = (1403580 A_{n-2} - 810728 A_{n-3}) (\text{mod } 4294967087),$$
$$B_n = (527612 B_{n-1} - 1370589 B_{n-3}) (\text{mod } 4294944443),$$
$$Z_n = (A_n - B_n) (\text{mod } 4294967087),$$
$$U_n = \begin{cases} Z_n / 4294967088, & \text{if } Z_n > 0, \\ 4294967087 / 4294967088, & \text{if } Z_n = 0, \end{cases}$$

which has a period of 3.1×10^{57}.

Evaluating RNGs

- We now discuss statistical methods of evaluating random number generators.
- All tests check one of the two ideal properties noted before:
  - Uniformity
  - Independence
- In practice,
  - Approximate uniformity is fairly easy to achieve.
  - Approximate independence is more difficult property.
- Let $U_1, U_2, U_3, \ldots$ be sequence of numbers generated by RNG.
- Null hypothesis

$$H_0 : U_1, U_2, U_3, \ldots \text{ is IID sequence of unif}[0, 1].$$

Testing for Uniformity

- χ^2 test is easily implemented to check if **marginal distribution** of $U_1, U_2, \ldots$ is unif$[0, 1]$.
  - For uniform distribution, equally-spaced and equiprobable bins coincide.
- Can also apply Kolmogorov-Smirnov test.
- These tests check only for marginal uniformity, not independence.

Serial Test for Independence

Correlation coefficient $\rho(X, Y)$ of random variables X, Y :

$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\mathcal{V}(X)\,\mathcal{V}(Y)}}$$

Properties of ρ

- Cauchy-Schwartz inequality $\quad \Rightarrow \quad -1 \le \rho \le 1$
- Correlation measures **linear dependence**: for constants c, d,
  - if $Y = cX + d$ with $c > 0$, then $\rho = +1$.
  - if $Y = cX + d$ with $c < 0$, then $\rho = -1$.
- X, Y independent implies $\rho = 0$, *but not vice versa*
- Partial dependence : $\rho \notin \{0, 1, -1\}$
  - Test if there are no correlations in data ...

Idea:

- Group data into d-dimensional vectors $(U_i, U_{i+1}, U_{i+2}, \ldots, U_{i+d-1})$
- If components in vector are indep and vectors indep, then vectors should be uniformly distributed in unit d-dimensional hypercube

Use χ^2 test to check uniformity

$$H_0 : \text{data is IID unif}[0, 1]$$

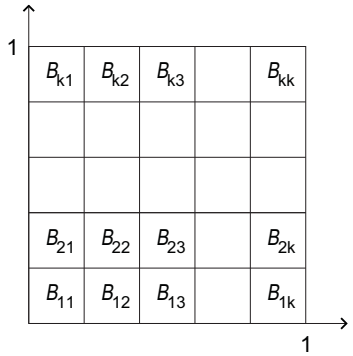
Consider case for $d = 2$ ...

- Group data in pairs :

$$\underbrace{U_1, U_2}_{\mathbf{V}_1}, \underbrace{U_3, U_4}_{\mathbf{V}_2}, \ \cdots, \ \underbrace{U_{2n-1}, U_{2n}}_{\mathbf{V}_n}$$

- Choose k and define bins B_{ij} as follows :

$$(U_\ell, U_{\ell+1}) \in B_{ij} \quad \Leftrightarrow \quad \frac{i-1}{k} \le U_\ell < \frac{i}{k} \text{ and } \frac{j-1}{k} \le U_{\ell+1} < \frac{j}{k}$$



- Dump vectors $\mathbf{V}_\ell = (U_{2\ell-1}, U_{2\ell})$ into bins B_{ij} ...
- Let O_{ij} be number of vectors in bin B_{ij}
- If original sequence $U_1, U_2 \ldots, U_{2n}$ is IID uniform, then

$$E_{ij} = \mathcal{E}[O_{ij}] = \frac{n}{k^2}$$

since all k^2 bins are same size.

- For χ^2 test, define error term $\mathcal{E}_{k^2,n}$ as

$$\mathcal{E}_{k^2,n} = \sum_{i,j=1}^{k} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

- If error term is "big," then reject null hypothesis.
- Define critical value

$$c = F^{-1}_{\chi^2_{k^2-1}}(1-\alpha)$$

where $F_{\chi^2_{k^2-1}}$ is cdf of χ_{k^2-1}.

- **Serial test**: Reject H_0 at level α iff $\mathcal{E}_{k^2,n} \ge c$ (equivalently, iff p-value $< \alpha$)
- Can generalize this to higher dimensions ...

Problems:

1. Starting-point bias since user specifies seed ... easily fixed
   - Arrange sequence U_i on circle
   - Randomly pick starting point
2. To thoroughly check independence, need large dimension d
   - Number of bins is k^d, which goes up *exponentially* in d ... *bad*

Runs-up-and-down test

- *Run* : sequence of events that share a property
  - *run up* : increasing sequence of points
  - *run down* : decreasing sequence of points
- Example :

0.951	0.231	0.607	0.486	0.891	0.762	0.457	0.019	0.821	0.445
	$\ominus$	$\oplus$	$\ominus$	$\oplus$	$\ominus$	$\ominus$	$\ominus$	$\oplus$	$\ominus$

Number of *runs up* (represented by $\oplus$) $= 3$
Number of *runs down* (represented by $\ominus$) $= 4$
Total number of *runs* $= 7$

Length of *runs up* $= 1, 1, 1$
Length of *runs down* $= 1, 1, 3, 1$

- Two types of tests :
  1. based on *number* of runs
  2. based on *length* of runs
- We will only cover method 1.
- H_0 : $U_1, U_2, U_3, \ldots$ are IID unif[0,1]

Runs-up-and-down test

- Idea : If number of runs R_n is "too large" or "too small," then H_0 improbable
- Same old story ... what is "large" and what is "small" ?
- Answer ... choose $l_{n,1-\alpha}$ and $u_{n,1-\alpha}$ such that

$$\mathcal{P}(\, l_{n,1-\alpha} \le R_n \le u_{n,1-\alpha} \mid H_0 \,) = 1 - \alpha$$

We now have *two* bounds because $\mathcal{E}[R_n] \neq 0$.

- **Test**: Do not reject H_0 iff $l_{n,1-\alpha} \le R_n \le u_{n,1-\alpha}$

But how does one get $l_{n,1-\alpha}$ and $u_{n,1-\alpha}$?

- Probabilistic arguments show that under H_0,

$$\mu_n \equiv \mathcal{E}[R_n] = \frac{2n-1}{3}, \qquad \sigma_n^2 \equiv \mathcal{V}[R_n] = \frac{16n-29}{90}$$

μ_n, σ_n are **not** sample averages ... **expectations** for length n

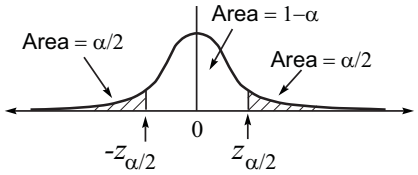
- Under H_0, R_n satisfies a *central limit theorem*:

$$\lim_{n\to\infty} \mathcal{P}\left(\frac{R_n - \mu_n}{\sigma_n} \le x\right) = \Phi(x)$$

  - CLT holds even though R_n is not sum of *independent* events.

- Use approximation for large n,

$$\frac{R_n - \mu_n}{\sigma_n} \approx \mathcal{N}(0,1) \quad \text{[standard normal]}$$



- Let Φ be CDF of $\mathcal{N}(0,1)$, and define $z_{\alpha/2} = \Phi^{-1}(1 - \frac{\alpha}{2})$
- So

$$\begin{aligned} 1-\alpha &= \mathcal{P}\left(-z_{\alpha/2} \le \mathcal{N}(0,1) \le z_{\alpha/2}\right) \\ &\approx \mathcal{P}\left(-z_{\alpha/2} \le \frac{R_n - \mu_n}{\sigma_n} \le z_{\alpha/2}\right) \end{aligned}$$

- **Approximate Test**: Do not reject H_0 (at level α) iff

$$\mu_n - z_{\alpha/2}\,\sigma_n \le R_n \le \mu_n + z_{\alpha/2}\,\sigma_n$$

Example cont.

0.951	0.231	0.607	0.486	0.891	0.762	0.457	0.019	0.821	0.445
⊖	⊕	⊖	⊕	⊖	⊖	⊖	⊕	⊖	

- $n = 10$ and $R_n = 7$
- $\mu_n = \frac{2n-1}{3} = 6.3333$
- $\sigma_n = \sqrt{\frac{16n-29}{90}} = 1.2065$
- Set level $\alpha = 0.05$, and from standard normal z-tables, get

$$z_{0.025} = 1.96$$

- Reject H_0 or not (at level $\alpha = 0.05$) ?

$\mu_n - z_{\alpha/2}\,\sigma_n = 3.9687 \le R_n = 7 \le \mu_n + z_{\alpha/2}\,\sigma_n = 8.698$

- Decision : **do not reject**

Runs above/below mean test

- Same idea as before ... but now compare with mean $\mu = 0.5$
- Example:

0.951	0.231	0.607	0.486	0.891	0.762	0.457	0.019	0.821	0.445
⊕	⊖	⊕	⊖	⊕	⊕	⊖	⊖	⊕	⊖

Number of *runs above mean* (represented by ⊕) = 4
Number of *runs below mean* (represented by ⊖) = 4
Total number of *runs* = 8

Length of *runs above mean* = 1,1,2,1
Length of *runs below mean* = 1,1,2,1

- Again two types of tests :
  1. based on *number* of runs
  2. based on *length* of runs
- We will only cover method 1.

Run-above-below-mean test

- Idea : If number of runs R_n is "too large" or "too small," then H_0 improbable
- Same old story ... what is "large" and what is "small" ?
- Same old answer ... choose $l_{n,1-\alpha}$ and $u_{n,1-\alpha}$ such that

$$\mathcal{P}(\, l_{n,1-\alpha} \;\leq\; R_n \;\leq\; u_{n,1-\alpha} \mid H_0 \,) \geq 1 - \alpha$$

- Same type of approximation: under H_0,
  - $\mu_n = \mathcal{E}[R_n] = \frac{n+1}{2}$ and $\sigma_n^2 = \mathcal{V}[R_n] = \frac{n-1}{4}$
  - *Central limit theorem* holds ... therefore

$$\frac{R_n - \mu_n}{\sigma_n} \;\approx\; \mathcal{N}(0,1) \quad \text{[standard normal]}$$

- Approximate test : Do not reject H_0 if

$$\mu_n - z_{\alpha/2}\,\sigma_n \;\;\leq\;\; R_n \;\;\leq\;\; \mu_n + z_{\alpha/2}\,\sigma_n$$

Notice : test is same as before ... same z's .. only things different are R_n, μ_n and σ_n

Example cont.

0.951 0.231 0.607 0.486 0.891 0.762 0.457 0.019 0.821 0.445

$\oplus$ $\ominus$ $\oplus$ $\ominus$ $\oplus$ $\oplus$ $\ominus$ $\ominus$ $\oplus$ $\ominus$

- $n = 10$ and *number* of runs $R_n = 8$
- $\mu_n = \frac{10+1}{2} = 5.5$ and $\sigma_n = \sqrt{\frac{10-1}{4}} = 1.5$
- Set level $\alpha = 0.05$, and from standard normal tables, get $z_{0.025} = 1.96$
- Reject H_0 or not (at level $\alpha = 0.05$) ?

$$\mu_n - z_{\alpha/2}\,\sigma_n = 2.5601 \;\leq\; R_n = 8 \;\leq\; \mu_n + z_{\alpha/2}\,\sigma_n = 8.4399$$

- Decision : **do not reject**

Primer on stationary stochastic process

- Let $\mathbf{X} = [X_i : -\infty < i < \infty]$ be *discrete-time* stochastic process
- **Definition:** Stochastic process $\mathbf{X}$ is *stationary* if for all $i_1 < i_2 < \cdots < i_d$, $B \subseteq \mathbf{R}^d$, $d \geq 1$, and n,
  $\mathcal{P}\left\{ (X_{n+i_1}, \ldots, X_{n+i_d}) \in B \right\} = \mathcal{P}\left\{ (X_{i_1}, \ldots, X_{i_d}) \in B \right\}$
- Special cases
  - Dimension $d = 1$:
    $\mathcal{P}\{ X_n \in B \} = \mathcal{P}\{ X_0 \in B \} \quad \Rightarrow \quad \mathcal{E}[X_n] = \mathcal{E}[X_0]$
  - Dimension $d = 2$:
    $\mathcal{P}\{ (X_{n+i}, X_n) \in B \} = \mathcal{P}\{ (X_i, X_0) \in B \}$
    $\Rightarrow \quad \mathcal{E}[X_{n+i}X_n] = \mathcal{E}[X_iX_0]$

When $\mathbf{X}$ is stationary,

- *Lag-j autocovariance of* $\mathbf{X}$ (independent of n)
  $$C_j = \mathsf{Cov}(X_{n+j}, X_n)$$
- *Lag-j autocorrelation of* $\mathbf{X}$ (independent of n)
  $$\rho_j = \rho(X_{n+j}, X_n) = \frac{\mathsf{Cov}(X_{n+j}, X_n)}{\sqrt{\mathcal{V}(X_{n+j})\,\mathcal{V}(X_n)}} = \frac{\mathsf{Cov}(X_{n+j}, X_n)}{\mathcal{V}(X_n)}$$
  since $\mathcal{V}(X_{n+j}) = \mathcal{V}(X_n)$ under stationarity.

Autocorrelation test

- $U_1, \ldots, U_n$ is sequence of numbers *supposedly* IID unif[0, 1]
- If true, then $\rho_0 = 1$ and $\rho_j = 0$ for all $j \neq 0$
- Form *hypothesis* testing problem
  - $H_0 : \rho_j = 0$, for all $j > 0$
  - $H_1 : \rho_j \neq 0$ for some j
- *Approximation* .... Fix large J
  - $H_0 : \rho_j = 0$, for all $0 < j < J$
  - $H_1 : \rho_j \neq 0$ for some j with $0 < j < J$

- If $U_i \sim$ unif[0, 1], then $\mathcal{E}[U_i] = 1/2$ and $\mathcal{V}[U_i] = 1/12$, so
  $$\rho_j = \frac{\mathcal{E}[U_jU_0] - (\mathcal{E}[U_0])^2}{\mathcal{V}[U_0]} = 12\,\mathcal{E}[U_jU_0] - 3$$
  $$= 0 \quad \text{if } U_j \text{ and } U_0 \text{ are independent.}$$
- From data, *estimate* value of ρ_j as follows:
  $$\widehat{\rho}_{j,n} = 12\left[\frac{\sum_{i=1}^{n} U_iU_{[(i+j) \bmod n]}}{n}\right] - 3$$

- Idea : Reject H_0 if $\widehat{\rho}_{j,n}$ is "too large" ...
- If H_0 holds, can show that

$$\begin{aligned} \mu_n &= \mathcal{E}[\widehat{\rho}_{j,n}] = 0, \\ \sigma_n^2 &= \mathcal{V}(\widehat{\rho}_{j,n}) = \tfrac{13n+7}{(n+1)^2}. \end{aligned}$$

- Again CLT holds for $\widehat{\rho}_{n,j}$, so for large n,

$$\frac{\widehat{\rho}_{j,n} - \mathcal{E}[\widehat{\rho}_{j,n}]}{\sqrt{\mathcal{V}(\widehat{\rho}_{j,n})}} \approx \mathcal{N}(0,1)$$

- So we have the test ... **Do not reject H_0 (at level α) iff**

$$-z_{\alpha/2}\,\sigma_n \leq \widehat{\rho}_{j,n} \leq z_{\alpha/2}\,\sigma_n$$

- Tests only that sequence has no **serial correlation**, but does not test for **independence**!

Other Tests

- *Poker Test* checks for excessive repetition of digits.
- For example, the sequence

  0.255, 0.577, 0.331, 0.414, 0.828, 0.909, 0.303, ...

  contains suspiciously many repeats.
- Think of each such triple as a poker hand:
  - Probability of "two-of-a-kind" is 0.27
  - Probability of "three-of-a-kind" is 0.01
  - Probability of three distinct digits is 0.72
  - These theoretical frequencies can be compared with observed frequencies using χ^2 test.

Does the RNG Matter?

- One might reasonably ask whether a nearly perfect random number generator is needed to get a meaningful simulation.
- In a sense, answer is No.
  - Every *pseudo*random number generator fails *some* statistical test
  - Yet computer simulations routinely give very accurate results.
- Impossible to make any general statements about what impact deficiencies in RNG have on simulation output
  - But it seems fair to say that best methods currently available are sufficiently good for most applications.
- One should not assume, however, that all built-in RNGs are good. Second-rate software can contain inferior methods.