APMA 3100
Spring 2025

Professor McMillan
Sections 003, 004

Group Project Instructions

Report Due: Wednesday, April 23, 11:59pm, in Gradescope

1. Objective

This project will introduce you to the idea of *Monte-Carlo simulation*, a technique for artificially creating large samples of random variables. This technique is used to simulate the behavior of random processes (defined by a combination of randomness and deterministic procedure), and to solve problems that are analytically intractable (such as those involving nonlinear differential equation systems). In a nutshell, Monte Carlo simulation is a technique for *numerical experimentation* on a computer.

While many software packages offer simulation programs, we do not use those here. To succeed in creative tasks, we must understand the conceptual fundamentals of system simulation. Our goal is to train as engineers – the designers and developers of new or specialized systems.

2. Simulation system

A Monte-Carlo simulation system has three main components (Yakowitz, 1977):

1. ***A random number generator:*** A numerical algorithm that provides a sequence of real numbers from the interval $(0, 1)$, termed *pseudo-random numbers*, which the observer unfamiliar with the algorithm, as well as the standard statistical tests, cannot distinguish from a sample of observations of a uniform random variable.

2. ***A random variable generator:*** An algorithm that maps any realization of the uniform random variable into a realization of a random variable having a given specified CDF. (The specification comes from the third component.)

3. ***A mathematical model*** of the process, system, or problem whose behavior or solution is to be simulated.

Sections 3 and 4 detail algorithms for the first two components. Your task is to implement them on a computer in whatever way you choose. Section 5 presents a problem. Your task is to model it using constructs of probability theory, and then to solve it via Monte-Carlo simulation.

Reference

Yakowitz, S.J. (1977). *Computational Probability and Simulation*, Addison-Wesley, Reading, Massachusetts.

3. Random number generator

A popular algorithm, known as the *linear congruential random number generator*, specifies the recursive rule:

$$x_i = (a\, x_{i-1} + c) \ \ (modulo\ K), \tag{1a}$$

$$u_i = decimal\ representation\ of\ x_i/K, \tag{1b}$$

for $i = 1, 2, 3, \dots$. The meaning of rule (1a) is this: multiply x_{i-1} by a and add c; then divide the result by K and take the remainder; then define x_i as this remainder. Every x_i is a nonnegative integer; a and K are positive integers; c is a non-negative integer. Rule (1b) states that the i^{th} random number u_i is the quotient x_i/K in the decimal representation, which is always a real number between 0 and 1.

Every sequence of pseudo-random numbers eventually cycles. To achieve maximum cycle length, the parameters must satisfy certain rules that we omit here. For this project, we choose:

starting value (seed)	$x_0 = 1000$,
multiplier	$a = 24693$,
increment	$c = 3517$,
modulus	$K = 2^{17}$.

These parameters yield a cycle of length 2^{17}. The first three random numbers are $0.4195, 0.0425$, 0.1274. Please compute the numbers u_{51}, u_{52}, u_{53} and show them in your report.

4. RANDOM VARIABLE GENERATOR

4.1 Discrete Random Variable

Let X be a discrete random variable with the sample space $\{x \mid x = 1, \dots, k\}$ and its PMF defined by $P_X(x) = P[X = x]$ for $x = 1, \dots, k$. The corresponding CDF is F_X, specified by

$$F_X(x) = P[X \le x] = \sum_{y \le x} P_X(y), \quad x = 1, \dots, k. \tag{2}$$

A given random number u_i generates realization x_i of the random variable X via the rule:

$$x_i = \min\{x \mid F_X(x) \ge u_i\} \tag{3}$$

This rule is executed by searching sequentially over $x = 1, \dots, k$ until $F_X(x) \ge u_i$ for the first time.

4.2 Continuous Random Variable

Let X be a continuous random variable having a continuous CDF $F_X(x)$ whose inverse F_X^{-1} exists in closed-form. That is, $F_X(x) = P[X \le x]$, and $x = F^{-1}(u)$ for any $u \in (0, 1)$.

A given random number u_i generates realization x_i of the random variable X through evaluation:

$$x_i = F^{-1}(u_i). \tag{4}$$

5. SIMULATION PROGRAM

A sales rep at a high-speed Internet provider calls customers to assess their satisfaction with the service. She takes 6 seconds to turn on her phone and dial a number; then either 3 seconds to detect a busy signal or 25 seconds to wait for 5 rings and conclude that no one will answer; and then 1 second to end the call. After an unsuccessful call, she re-dials (over the course of several days) until the customer answers or she has dialed 4 times. The *outcome* of each dialing is determined like this: The customer is using the line with probability 0.2; or is away from the phone with probability

0.3; or is available and can answer the call within X seconds, where X is a continuous random variable with exponential distribution and average wait time of 12 seconds. (***Note:*** *it is possible for the customer to be available but take too long to answer and miss the call.)* The *calling process* **ends** when the customer answers the call, or when 4 unsuccessful calls have been placed.

Let W denote the total time spent by the sales rep on calling 1 customer. Your objective is to estimate several 'statistics' of W. For this purpose, do the following:

1. **Formulate a model** of the calling process.
    - 1.1 Define notation for all elements of the model.
    - 1.2 Write the expression for the CDF of X and the inverse of this function.
    - 1.3 Draw a tree diagram for the calling process.
      (*Hint*: Consider the calling process from two points of view: that of the sales rep and that of the customer. The tree diagram may include IF statements or loops, like a flowchart.)

2. **Design a Monte-Carlo simulation algorithm**. The algorithm should be capable of generating n independent realizations (each starting with a different random number) of the calling process and thereby outputting a sample of size n of random variable W. You may decide how to implement the algorithm on the computer. In your report, briefly describe the algorithm design, the computer code, and the computer language used.

3. **Simulate** the calling process $n = 1000$ times.

4. **Estimate** from the generated sample of W: (i) the mean; (ii) the first quartile, the median, the third quartile; (iii) the probabilities of these events:

$$W \leq 15,\ W \leq 20,\ W \leq 30,\ W > 40,\ W > w_5,\ W > w_6,\ W > w_7,$$

where $w_5,\ w_6,\ w_7$ are some values you choose in order to depict well the right tail of the CDF of W.

5. **Analyze** the results and draw conclusions. In particular:
    - 6.1 Compare the mean with the median. What does this comparison suggest about the shape of the PDF of W?
    - 6.2 Determine the sample space of W.
    - 6.3 Graph the CDF of W using the probabilities estimated in Step 5, and interpolating between them whenever appropriate. Could W be an exponential random variable? Justify your answer.

6. **Comment** on the approach. At a minimum, answer these questions:
    - (i) Which of the steps was the most (the least) challenging?
    - (ii) Which of the steps was the most (the least) time consuming?

6. REPORT

Instructions

1. Include an introduction, briefly introducing the problem.
2. Organize the report into 7 sections parallel to the above steps.
3. Explain your work, summarize the results, answer the questions.
4. Do *not submit* computer code or raw data, but *do archive* them.
5. Draw figures professionally: appropriate scale, labeled axes, captions.
6. You may discuss your project with the instructor, TAs, and Groupmates only. All work must be your own. The Honor Pledge must be printed on a cover page and signed by every member of the team.