# Lecture 5: Monte Carlo and Discrete-Event Simulation

Mathematical Modeling Lecture

Mathematical Modeling

Kenneth Sok Kin Cheng

# Session Route

### Launch

**Question**

Motivating context, key vocabulary, and the first modelling decision.

### Build

**Method**

Worked example, diagram or table, calculation, and parameter meaning.

### Test

**Critique**

Checkpoint practice, limitation check, and communication of results.

# Learning Goals

By the end of the lesson, readers should be able to:

1. distinguish a **deterministic model** from a **stochastic model**;
2. map uniform random numbers to outcomes from a discrete or empirical probability distribution;
3. construct and interpret a simple **Monte Carlo estimator**;
4. explain why repeated simulation runs are needed and why simulation error decreases slowly;
5. trace a single-server queue using arrival, service-start, waiting, and departure times;
6. design a simulation experiment that includes outputs, replications, validation, sensitivity, and limitations.

# Why simulate?

Core explanation, worked reasoning, and application

# Why simulate?

Previous lectures represented a system using equations or curves. That approach remains valuable, but some systems contain uncertain events whose exact sequence cannot be predicted: customers arrive at irregular times, service durations vary, and equipment may fail unexpectedly. A **simulation** imitates the operation of such a system and records what happens under specified assumptions.

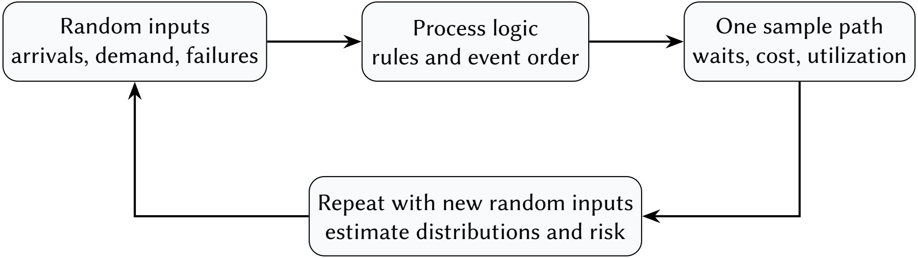
# Activity: Launch: The canteen queue

At lunchtime, students arrive at one canteen counter at irregular times. Some orders take much longer than others. The school is considering opening a second counter.
In pairs, identify:

1. two quantities that vary unpredictably from student to student;
2. two quantities that describe the current *state* of the system;
3. three outputs that the school should examine before making a decision;
4. one factor that might make a simulation unrealistic.

**Modeling lesson:** A simulation requires more than random numbers. It needs explicit input assumptions, update rules, outputs, and a plan for checking credibility.

# Illustration: Why simulate?




A simulation combines random inputs with deterministic update rules. One run gives one possible history; repeated runs reveal the range of possible outcomes.

# Concept: Core simulation language

- A **state variable** describes the system at a given time, such as queue length or inventory level.
- An **event** changes the state, such as an arrival, a service completion, or a delivery.
- A **sample path** is one simulated history of the system.
- A **replication** is an independent run under the same policy and assumptions.
- A **random seed** reproduces the same pseudorandom sequence, which is useful for debugging and fair comparisons.

The computer performs deterministic calculations after the random inputs have been generated. Simulation is therefore structured experimentation, not random guessing.

# From uniform random numbers to model inputs

Core explanation, worked reasoning, and application

# From uniform random numbers to model inputs

A **random variable** assigns a numerical value to each possible outcome of a random experiment. To simulate a discrete random variable, partition the interval $[0, 1)$ according to cumulative probabilities and generate $U \sim \mathrm{Uniform}(0, 1)$.
Suppose a service time $S$ has possible values $s_1, \ldots, s_k$ with probabilities $p_1, \ldots, p_k$. Let

$$F_j = p_1 + \cdots + p_j.$$

Assign $S = s_j$ when $F_{j-1} \leq U < F_j$, where $F_0 = 0$. This is the discrete **inverse transform method**.

# Worked Example: Worked Example 1: Simulating service times

Historical observations at a snack counter suggest

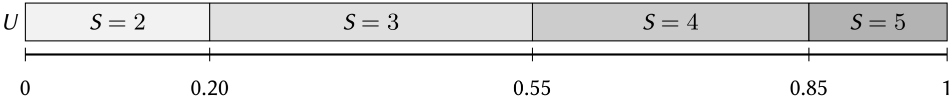
| service time $S$ (min) | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| $P(S = s)$ | 0.20 | 0.35 | 0.30 | 0.15 |

The cumulative probabilities are $0.20$, $0.55$, $0.85$, and $1.00$, so

$$S = \begin{cases} 2, & 0 \leq U < 0.20, \\ 3, & 0.20 \leq U < 0.55, \\ 4, & 0.55 \leq U < 0.85, \\ 5, & 0.85 \leq U < 1. \end{cases}$$

For $U = 0.07, 0.48, 0.61, 0.93$, the simulated service times are $2, 3, 4, 5$ minutes respectively.

# Illustration: From uniform random numbers to model inputs




The length of each interval equals the probability of its outcome. A larger probability occupies more of $[0, 1)$.

# Activity: Guided practice: Interarrival times

Student interarrival times have the distribution

| $A$ (min) | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| $P(A=a)$ | 0.15 | 0.35 | 0.35 | 0.15. |

1. Construct the cumulative probability table and the interval mapping.
2. Convert $U = 0.08, 0.32, 0.71, 0.94, 0.51$ into interarrival times.
3. Explain why a fixed seed is helpful when comparing one counter with two counters.

# Model Critique: The input distribution is part of the model

A simulation can produce precise-looking output from a poor input model. Historical frequencies may be unrepresentative of examination days, special menus, or changing student behavior. The probabilities, dependence between variables, and time period represented by the data must all be justified.

# Monte Carlo estimation

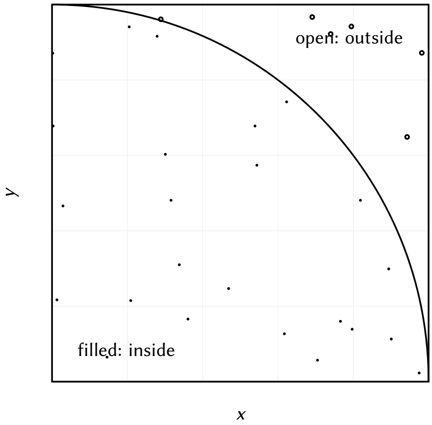
Core explanation, worked reasoning, and application

# Monte Carlo estimation

A **Monte Carlo simulation** estimates a quantity by repeating a random experiment. Consider points drawn uniformly from the unit square. The probability of falling inside the quarter circle $x^2 + y^2 \leq 1$ is its area, $\pi/4$. If $C$ of $n$ points fall inside, then

$$\frac{C}{n} \approx \frac{\pi}{4}, \qquad \boxed{\widehat{\pi} = 4\frac{C}{n}.}$$

# Illustration: Monte Carlo estimation




Thirty reproducible sample points: 24 fall inside the quarter circle and 6 outside. The picture represents one sample path, not a guarantee of accuracy.

# Worked Example: Worked Example 2: One estimate of $\pi$

In Figure 3, $C = 24$ and $n = 30$, so

$$\widehat{\pi} = 4\left(\frac{24}{30}\right) = \boxed{3.20}.$$

The estimate is plausible but rough. A different seed would produce a different count. The value $3.20$ should therefore be reported together with $n$, the random-number procedure, and preferably results from repeated runs.

# Monte Carlo estimation

For independent samples, Monte Carlo uncertainty usually decreases at the rate

$$\text{typical error} \propto \frac{1}{\sqrt{n}}.$$

Thus reducing the typical error by a factor of $2$ requires about $4$ times as many samples; reducing it by a factor of $10$ requires about $100$ times as many.

# Checkpoint: Convergence

A simulation uses $2{,}500$ independent samples. Approximately how many samples are required to reduce its typical Monte Carlo error to one-fifth of the current value? Explain using the square-root rule.

# Concept: Replication and uncertainty

Increasing the number of observations within one run and increasing the number of independent replications answer different questions. A long run may estimate steady performance; several replications reveal run-to-run variability. Report at least a mean and a measure of spread, and retain the individual replication results rather than only the grand average.

# Discrete-event simulation of a queue

Core explanation, worked reasoning, and application

# Discrete-event simulation of a queue

In a **discrete-event simulation**, the state changes only when an event occurs. For a single-server queue, let

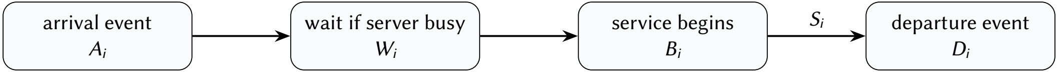
- $A_i$ be the arrival time of customer $i$;
- $S_i$ be the required service time;
- $B_i$ be the service-start time;
- $D_i$ be the departure time;
- $W_i = B_i - A_i$ be the waiting time.

The update equations are

$$\boxed{B_i = \max(A_i, D_{i-1}), \qquad D_i = B_i + S_i, \qquad W_i = B_i - A_i,}$$

with $D_0 = 0$. The maximum expresses the queue logic: service begins only after both the customer has arrived and the server is free.

# Illustration: Discrete-event simulation of a queue




A queue is governed by event order. Randomness enters through arrival and service times; the update logic is deterministic.

# Worked Example: Worked Example 3: Manual trace of one counter

Six customers have arrival times

$$A_i = (0, 3, 5, 10, 12, 16)$$

and service times

$$S_i = (4, 5, 3, 4, 2, 5)$$

in minutes. Applying the update equations gives:

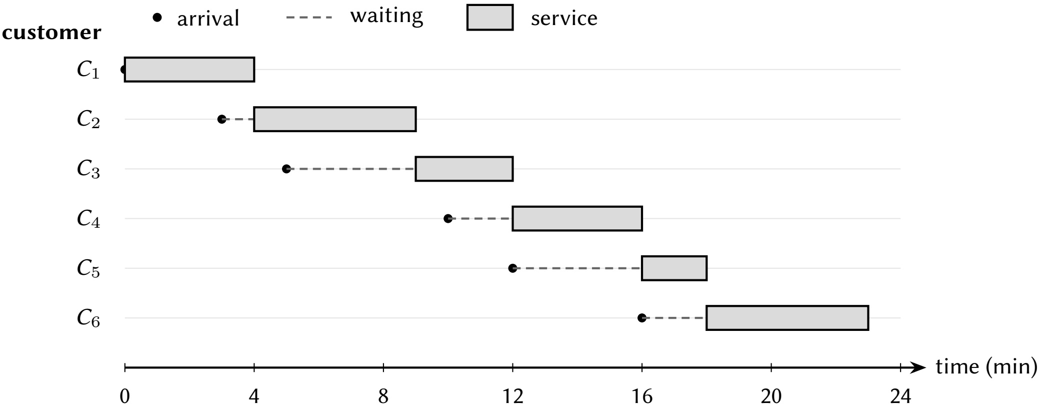
| Customer $i$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Arrival $A_i$ | 0 | 3 | 5 | 10 | 12 | 16 |
| Service $S_i$ | 4 | 5 | 3 | 4 | 2 | 5 |
| Start $B_i$ | 0 | 4 | 9 | 12 | 16 | 18 |
| Wait $W_i$ | 0 | 1 | 4 | 2 | 4 | 2 |
| Departure $D_i$ | 4 | 9 | 12 | 16 | 18 | 23 |

The average waiting time is

$$\bar{W} = \frac{0+1+4+2+4+2}{6} = \boxed{2.17 \text{ min}},$$

and the maximum waiting time is $4$ minutes. These values describe only this six-customer sample path.

# Illustration: Discrete-event simulation of a queue




Customer-centred queue timeline. A filled point marks arrival, a dashed segment shows waiting, and a shaded block shows the service interval.

# Activity: Policy comparison: Add a second counter

Use the same arrivals and service times as Worked Example 3. Assign each arriving customer to the counter that becomes available first.

1. Complete a table containing counter number, start time, waiting time, and departure time.
2. Find the average and maximum waiting times.
3. Explain why this single sample path is insufficient evidence for permanently staffing a second counter.

# From one run to a decision

Core explanation, worked reasoning, and application

# From one run to a decision

A simulation study is an experiment on a model. Before running it, define the policy alternatives, input distributions, performance measures, simulation length, number of replications, and comparison method. A useful preliminary check for a queue is the **traffic intensity**

$$\rho \approx \frac{\text{mean service time}}{c \times \text{mean interarrival time}},$$

where $c$ is the number of servers. If $\rho > 1$, work arrives faster than the system can process it, so the queue tends to grow rather than settle to a stable long-run distribution.

# Worked Example: Worked Example 4: Capacity before detailed simulation

Suppose the mean interarrival time is $3.5$ minutes and the mean service time is $4.7$ minutes.
For one counter,

$$\rho_1 = \frac{4.7}{1(3.5)} \approx 1.34 > 1.$$

The counter lacks sufficient average capacity. A long simulation will show increasing waits, but the key conclusion is structural: one counter cannot keep up under these assumptions.
For two counters,

$$\rho_2 = \frac{4.7}{2(3.5)} \approx 0.67 < 1.$$

A stable system is possible, although waiting-time variability still requires simulation. Capacity checks should accompany, not replace, detailed simulation.

# Model Critique: Do not mistake a simulated average for a universal fact

A result such as "average wait $= 2.17$ minutes" is conditional on the input distributions, queue discipline, staffing policy, operating period, and random sample. A credible report should provide:

1. the assumptions and data source for each random input;
2. the number and length of replications;
3. the mean together with variability or percentiles;
4. sensitivity to uncertain assumptions;
5. verification of the algorithm and validation against real observations.

# Activity: Simulation study design

The school wants to compare one counter, two counters, and one counter with a shorter menu.
Write a short experiment plan specifying:

1. the random inputs and how they would be estimated;
2. at least four outputs;
3. the simulation horizon and number of replications;
4. one fair comparison technique using common random seeds;
5. one validation check and one sensitivity test.

# Concept: Verification and validation

**Verification** asks, “Did we implement the model correctly?” Trace small cases by hand, check conservation rules, and reproduce results with a fixed seed.

**Validation** asks, “Does this model represent the real system adequately for its purpose?” Compare simulated and observed queue lengths, waits, utilization, and arrival patterns. A program can be verified but still model the wrong system.

# Lesson summary

Core explanation, worked reasoning, and application

# Summary: Six ideas to retain

1. Simulation combines random inputs with deterministic process rules.
2. Uniform random numbers can be mapped to empirical outcomes through cumulative probabilities.
3. One Monte Carlo run is one possible outcome, not the answer.
4. Typical Monte Carlo error decreases only as $1/\sqrt{n}$.
5. In a queue, event order is captured by $B_i = \max(A_i, D_{i-1})$.
6. Decisions require replications, uncertainty summaries, verification, validation, and sensitivity analysis.

# Checkpoint: Exit ticket

Complete both statements:
"A fixed random seed is useful because ______________________________."
"A second seed or independent replication is necessary because ______________________________."

# Practice problems

Core explanation, worked reasoning, and application

# Exercises: Core practice

1. A repair time $T$ has distribution

| $T$ (h) | 1 | 2 | 3 | 5 |
|---|---|---|---|---|
| $P(T = t)$ | 0.10 | 0.40 | 0.35 | 0.15. |

   Construct the interval mapping and convert $U = 0.04, 0.36, 0.71, 0.92$ into repair times.

2. A quarter-circle simulation produces $C = 7{,}824$ inside points out of $n = 10{,}000$.
   2.1 Estimate $\pi$.
   2.2 If the sample size is multiplied by 25, by what factor should the typical Monte Carlo error change?
   2.3 Explain why the actual error need not decrease in every individual run.

3. For a single-server queue, arrivals occur at

$$A = (0, 2, 6, 7, 11),$$

   and service times are

$$S = (3, 5, 2, 4, 3).$$

   Construct a complete table of start times, waits, and departures. Find the average wait, maximum wait, and average time in the system.

# Exercises: Core practice

4. The mean interarrival time at a help desk is $6$ minutes and the mean service time is $8$ minutes.
   - 4.1 Calculate $\rho$ for one server and two servers.
   - 4.2 What can be concluded before simulation?
   - 4.3 What important information is still missing?
5. Explain why each statement is inadequate:
   - 5.1 "The simulation was run once and gave an average wait of 4.2 minutes."
   - 5.2 "The program has no coding errors, so the model is valid."
   - 5.3 "The histogram fits last month's data, so it will remain correct all year."
   - 5.4 "The two-counter system has a lower mean wait, so it is automatically the best policy."
6. A team compares two staffing policies using exactly the same sequence of arrivals and service requirements. Explain why this **common random numbers** design can make the comparison clearer. Why should the team still use several seeds?

# Exercises: Modeling challenge

Prepare a one-page simulation brief for one of the following systems:

1. a school canteen queue;
2. bicycle availability at a sharing station;
3. daily inventory of a popular stationery item;
4. patient arrivals at a small clinic.

Include: purpose, state variables, events, random inputs, input-data plan, process logic, outputs, policy alternatives, replication plan, validation, sensitivity, and one limitation. A complete computer program is not required.

# Optional algorithm: Single-server queue

Core explanation, worked reasoning, and application

# Optional algorithm: Single-server queue

This appendix is not part of the two-hour core lesson. It translates the event equations into pseudocode.

# Concept: Queue simulation pseudocode

1. Set previous departure $D \leftarrow 0$ and initialize output lists.
2. For customer $i = 1, \ldots, n$:
   2.1 generate or read interarrival and service times;
   2.2 update the arrival time $A_i$;
   2.3 set $B_i \leftarrow \max(A_i, D)$;
   2.4 set $W_i \leftarrow B_i - A_i$;
   2.5 set $D_i \leftarrow B_i + S_i$ and update $D \leftarrow D_i$;
   2.6 record waiting time, time in system, queue length, and server state.
3. Compute summary statistics for the replication.
4. Repeat with independent seeds and summarize across replications.

# Optional note: Standard error for the quarter-circle estimator

Core explanation, worked reasoning, and application

# Optional note: Standard error for the quarter-circle estimator

Let $I_j$ equal $1$ when sample point $j$ lies inside the quarter circle and $0$ otherwise. Then $I_j$ is Bernoulli with $p = \pi/4$, and

$$\widehat{\pi} = 4\bar{I}.$$

Therefore

$$\mathrm{SE}(\widehat{\pi}) = 4\sqrt{\frac{p(1-p)}{n}} \approx \frac{1.64}{\sqrt{n}}.$$

This is a standard deviation of the estimator, not a guaranteed error bound. For example, increasing $n$ from $100$ to $10{,}000$ reduces the standard error by a factor of $10$.