

VU LEARNING HUB

vulearninghub.xyz

HIGHLIGHTED HANDOUT

MTH5204

Modelling and Simulation

MTH5204 by VU Learning HUB

Prepared for VU students • Free study resources

<https://vulearninghub.xyz>

MTH5204 - Modelling and Simulation

A structured study handout compiled from the course's lecture topics. Read each topic's objectives first, study the key concepts, then use the summary and exam tips for revision. 

MTH5204

Modelling and Simulation

Based on 45 lecture topics

Compiled for VU students as a structured study guide

*Concepts are summarized in the writer's own words;
use alongside your lecture videos for best results.*

Contents

Unit 1: Foundations of Simulation	4
Topic 1: What Is Simulation?	4
Topic 2: When Simulation Is (and Is Not) Appropriate	5
Topic 3: Systems, Models, and Model Taxonomy	6
Topic 4: Steps in a Simulation Study	7
Unit 2: Probability, Statistics, and Randomness	7
Topic 5: Probability and Statistics Review	7
Topic 6: Random Number Generation	8
Topic 7: Random Variate Generation	9
Topic 8: Input Data Modelling	10
Unit 3: Discrete-Event Simulation Mechanics	11
Topic 9: The Event-Scheduling World View	11
Topic 10: Hand Simulation of a Single-Server Queue	12
Topic 11: Queueing Systems and Kendall Notation	13
Topic 12: Little's Law and Performance Measures	14
Unit 4: Verification, Validation, and Output Analysis	14
Topic 13: Verification and Validation	14
Topic 14: Output Analysis: Confidence Intervals	15
Topic 15: Comparing System Alternatives	16
Topic 16: Variance Reduction Techniques	17
Unit 5: Advanced Simulation Topics	18
Topic 17: Monte Carlo Simulation	18
Topic 18: Continuous and Combined Simulation	19
Topic 19: Simulation Software and Languages	20
Topic 20: Case Studies and Project Guidance	21
Unit 6: Worked Problems and Exam Preparation 	22
Topic 21: Worked Problem: Linear Congruential Generator	22
Topic 22: Worked Problem: Inverse-Transform Sampling	22
Topic 23: Worked Problem: Hand Simulation and M/M/1 Check	23
Topic 24: Worked Problem: Confidence Interval from Replications	24
Topic 25: Formula Sheet and Final Exam Strategy	25
Unit 7: Mini-Project: Complete Walkthrough	25
Topic 26: Project Brief: Campus Photocopy Shop	26
Topic 27: Building and Verifying the Model	26

Topic 28: Results, Comparison, and Recommendation 27

Topic 29: What You Can Build Next 28

Glossary of Key Terms 29

Course Introduction

Modelling and simulation is the science of imitating real-world processes on a computer so that we can understand, predict, and improve them without disturbing the real system. From queues at a bank to supply chains, epidemics, factories, and computer networks, simulation lets us ask "what if" questions safely and cheaply.

This handout covers the full simulation study: formulating models, generating randomness, building discrete-event models, analyzing queues, validating results, and drawing statistically sound conclusions. It mixes theory (probability, queuing theory, statistics) with practice (worked hand-simulations, algorithm sketches, software guidance).

By the end you should be able to: formulate a simulation model of a stochastic system, generate random numbers and variates correctly, hand-simulate a queue, analyze output data with confidence intervals, validate a model, and report results professionally.

How to use this handout: Each topic lists its learning objectives, explains the key concepts in simple words, defines important terms, gives an example where helpful, and closes with a summary plus exam tips. Topics marked **Very important** deserve extra revision.

Unit 1: Foundations of Simulation

Topic 1: What Is Simulation?

Learning Objectives

- Define simulation and a simulation model
- Distinguish simulation from analytical solution
- List advantages and disadvantages of simulation

Key Concepts

A simulation is the imitation of the operation of a real-world process or system over time. Whether done by hand or on a computer, simulation involves generating an artificial history of the system and observing that artificial history to draw inferences about the operating characteristics of the real system.

The behavior of a system as it evolves is studied by developing a simulation model - a set of assumptions, expressed as mathematical, logical, and symbolic relationships between the entities (objects of interest) of the system. Once developed and validated, a model can investigate "what if" questions: potential changes can be simulated first to predict their impact.

When a model is simple enough, it can be solved by mathematical methods (differential calculus, probability theory, algebra) - this is an analytical solution. But many real systems are so complex that analytical solution is impossible; then numerical, computer-based simulation is the only route. Simulation is thus both an analysis tool (predicting effects of changes to existing systems) and a design tool (predicting performance of systems not yet built).

Key Points

- Simulation = artificial history of a system used to infer real-system behavior.
- A model = assumptions expressed as mathematical/logical/symbolic relationships.
- Analytical solutions fail for complex systems - simulation is the fallback.

Key Terms

System: A collection of entities interacting toward some purpose.

Entity: An object of interest in the system (customer, machine, packet).

Model: A representation of the system built for study.

Example

- *Testing a new traffic-light timing by simulating traffic rather than disrupting a real intersection.*

Summary: Simulation imitates systems over time to support analysis and design decisions.

Exam Tips

- ★ Very important: the formal definition of simulation.
- ★ Remember: analysis tool vs. design tool.

Topic 2: When Simulation Is (and Is Not) Appropriate

Learning Objectives

- State when simulation is the right tool
- State when it is not
- Give examples of each

Key Concepts

Simulation is appropriate when: the system is too complex for analytical solution; experimentation on the real system is impossible, dangerous, or expensive (nuclear plants, aircraft design); we need to study a system over compressed or expanded time (years of operation in minutes); we want to compare design alternatives before building; or we need visualization/animation to communicate ideas to stakeholders.

Simulation is NOT the tool when: the problem can be solved by common sense or simple arithmetic; an analytical solution exists and is easier (do not simulate what you can solve); direct experiments on the real system are cheaper and feasible; the cost of the study exceeds the expected savings; or adequate data, time, or personnel are unavailable. A famous caution: simulation does not give exact answers - it gives estimates with statistical error, and garbage in means garbage out.

Typical applications: manufacturing (line balancing, scheduling), logistics and supply chains, telecommunications networks, computer systems performance, healthcare (emergency departments, operating theatres), finance (portfolio risk), military and emergency planning, and business process reengineering.

Key Points

- Use simulation when the real system cannot be experimented on safely or cheaply.
- Do not simulate when analytical solution or direct experiment is easier.
- Applications span manufacturing, logistics, telecom, healthcare, finance.

Key Terms

Analytical solution: Exact mathematical solution of a model.

What-if analysis: Testing alternative scenarios in the model.

Garbage in, garbage out: Bad input data yields bad simulation results.

Example

- *Simulating emergency-department staffing before hiring - experimenting on real patients is unacceptable.*

Summary: Simulation is powerful but must be justified; it is a tool, not a universal answer.

Exam Tips

- ★ Key point: conditions when simulation is inappropriate.
- ★ Remember two application areas with examples.

Topic 3: Systems, Models, and Model Taxonomy

Learning Objectives

- Define system, state, event
- Classify models (static/dynamic, deterministic/stochastic, discrete/continuous)

Key Concepts

A system is defined by entities, attributes (properties of entities), activities (processes causing changes), state (the collection of variables needed to describe the system at a point in time), and events (instantaneous occurrences that change the state - e.g., a customer arrival).

Models are classified along three axes. Static vs. dynamic: a static model represents the system at a particular point in time (a Monte Carlo inventory calculation), while a dynamic model represents the system as it evolves over time (a queueing simulation). Deterministic vs. stochastic: deterministic models contain no random components (a known, fixed schedule); stochastic models have at least one random input (random arrivals), so output is itself random and must be treated as a statistical estimate.

Discrete vs. continuous: in a discrete model, state variables change only at a countable number of points in time (number of customers in a bank - changes only at arrivals/departures); in a continuous model, state variables change continuously (water level in a tank). Most models in this course are discrete, dynamic, and stochastic - the classic discrete-event simulation.

Key Points

- State = variables describing the system; events = instantaneous state changes.
- Three classification axes: static/dynamic, deterministic/stochastic, discrete/continuous.
- This course focuses on discrete, dynamic, stochastic models.

Key Terms

State variables: Variables describing system status at a time.

Event: Instantaneous occurrence changing the state.

Discrete-event simulation: Modelling a system whose state changes at event times.

Example

- *Bank queue: discrete (customer count jumps), dynamic (evolves over time), stochastic (random arrivals).*

Summary: Model taxonomy tells us which techniques apply; simulation here is discrete-event.

Exam Tips

- ★ Very important: the three classification axes with examples.
- ★ Remember definitions of state and event.

Topic 4: Steps in a Simulation Study

Learning Objectives

- List the steps of a sound simulation study
- Explain the role of each step
- Explain why the process is iterative

Key Concepts

A professional simulation study follows these steps. (1) Problem formulation: state the problem clearly with the decision-maker; define objectives and measures of performance. (2) Setting objectives and project plan: is simulation the right methodology? Plan time, resources, deliverables. (3) Model conceptualization: abstract the essential features - decide what to include and what to simplify. (4) Data collection: gather input data; this step runs in parallel with conceptualization since data availability shapes the model.

(5) Model translation: code the model in a simulation language or general-purpose language. (6) Verification: is the model implemented correctly (debugging)? (7) Validation: does the model represent the real system accurately enough for the purpose? (8) Experimental design: decide scenarios, run lengths, replications. (9) Production runs and analysis: run the experiments, analyze output statistically. (10) Documentation and reporting: document assumptions, code, results; present recommendations. (11) Implementation: act on the findings.

The process is iterative, not linear: validation failures send you back to conceptualization; new questions from the decision-maker restart formulation. Skipping validation or output analysis is the most common cause of worthless studies.

Key Points

- Eleven steps from problem formulation to implementation.
- Verification = building the model right; validation = building the right model.
- The process is iterative - failures loop back to earlier steps.

Key Terms

Verification: Checking the model is implemented correctly.

Validation: Checking the model matches reality adequately.

Conceptual model: The abstracted, pre-code description of the system.

Example

- *A validated bank model failing to match observed waiting times forces re-examination of arrival assumptions.*

Summary: Disciplined, iterative steps separate professional simulation from guesswork.

Exam Tips

- ★ Very important: verification vs. validation definitions.
- ★ Remember the steps in order.

Unit 2: Probability, Statistics, and Randomness

Topic 5: Probability and Statistics Review

Learning Objectives

- Recall key distributions
- Explain expectation and variance
- Distinguish discrete and continuous distributions

Key Concepts

Simulation is built on probability. A random variable X has a probability distribution: discrete distributions (Bernoulli, binomial, geometric, Poisson) give probabilities $P(X = x)$; continuous distributions (uniform, exponential, normal, Erlang, Weibull) give a density $f(x)$ with $P(a < X < b)$ as the area under the curve.

Key summaries: expected value $E[X]$ (long-run average), variance $\text{Var}(X) = E[(X - E[X])^2]$ (spread). Memoryless property of the exponential: $P(X > s + t \mid X > s) = P(X > t)$ - the remaining service time does not depend on how long service has already lasted; this makes the exponential the standard model for interarrival and service times in elementary queueing.

Common modelling choices: Poisson process for arrivals (exponential interarrival times), normal for measurement errors and aggregated quantities (central limit theorem), exponential for service times, uniform for "no information" ranges, triangular when only min/mode/max are known, Erlang for multi-phase services. Choosing distributions from data is input modelling (Lecture 8).

Key Points

- Discrete vs. continuous distributions; expectation and variance summarize them.
- Exponential is memoryless - standard for interarrival/service times.
- Distribution choice encodes assumptions; match the choice to the data.

Key Terms

Random variable: A variable whose value is a random outcome.

Memoryless property: Future independent of the past - exponential's signature.

Central limit theorem: Sums of many independent variables tend to normal.

Example

- *Customer arrivals modelled as a Poisson process with rate λ per minute.*

Summary: Probability distributions are the language in which stochastic models are written.

Exam Tips

- ★ Very important: memoryless property of the exponential.
- ★ Remember which distribution fits which situation.

Topic 6: Random Number Generation

Learning Objectives

- Explain pseudorandom numbers
- Describe the linear congruential generator
- List tests of randomness

Key Concepts

Simulation needs streams of $\text{Uniform}(0,1)$ random numbers; everything else is built from them. Computers produce pseudorandom numbers - deterministic sequences that behave statistically like

random ones. We demand: uniformity on (0,1), independence (no predictable pattern), long period before repeating, and reproducibility (same seed gives same stream for debugging and comparison).

The classic generator is the linear congruential generator (LCG): $X_{n+1} = (a * X_n + c) \bmod m$, with $U_n = X_n / m$. The seed X_0 starts the sequence. Choice of multiplier a , increment c , and modulus m determines quality; full period m requires care (e.g., c and m coprime, $a - 1$ divisible by all prime factors of m). Modern practice uses better generators (Mersenne Twister, PCG), but the LCG teaches the principles.

Testing randomness: frequency tests (chi-square, Kolmogorov-Smirnov) check uniformity; serial and runs tests check independence; autocorrelation plots reveal hidden structure. A generator that fails tests cannot be trusted for any study. Seeds matter: use separate, non-overlapping streams for different random sources (arrivals vs. services) so experiments are clean and reproducible.

Key Points

- Pseudorandom = deterministic but statistically random; must be uniform, independent, long-period, reproducible.
- LCG: $X_{n+1} = (a * X_n + c) \bmod m$.
- Test generators (chi-square, K-S, runs tests) before trusting them.

Key Terms

Seed: Starting value fixing the random stream.

Period: Length before the sequence repeats.

Stream: An independent sequence from one seed.

Example

- *Seed 12345 reproduces the exact arrival pattern for debugging a queue model.*

Summary: Quality randomness is the foundation; test it and control it with seeds.

Exam Tips

- ★ Very important: write the LCG formula.
- ★ Remember the four desired properties.

Topic 7: Random Variate Generation

Learning Objectives

- Generate variates by inverse transform
- Explain acceptance-rejection
- Generate normal variates

Key Concepts

From $U \sim \text{Uniform}(0,1)$ we generate any distribution. The inverse-transform method: for a distribution with CDF F , set $X = F^{-1}(U)$. For the exponential with rate λ , $F(x) = 1 - e^{-\lambda x}$, giving $X = -(1/\lambda) * \ln(1 - U)$, or equivalently $-(1/\lambda) * \ln(U)$ since $1 - U$ is also uniform. For discrete distributions, invert the CDF stepwise: generate U , then pick the smallest x with $F(x) \geq U$ (e.g., for a die, intervals of width $1/6$).

The acceptance-rejection method: to sample from density f , find a simpler envelope density g (easy to sample) and constant c with $f(x) \leq c * g(x)$ everywhere; generate Y from g and U uniform, accept Y if $U \leq f(Y) / (c * g(Y))$.

$\leq f(Y)/(c \cdot g(Y))$, else repeat. Efficiency = $1/c$, so choose tight envelopes. It handles densities whose CDF cannot be inverted.

Special methods: Box-Muller transforms two uniforms into two independent standard normals: $Z_1 = \sqrt{-2 \ln U_1} \cos(2 \pi U_2)$, $Z_2 = \sqrt{-2 \ln U_1} \sin(2 \pi U_2)$. Erlang(k , λ) = sum of k exponentials. Poisson variates can be generated by counting exponential interarrivals until their sum exceeds 1 (for unit rate), scaled by λ .

Key Points

- Inverse transform: $X = F^{-1}(U)$; exponential: $X = -(1/\lambda) \ln(U)$.
- Acceptance-rejection: sample from an envelope, accept with probability $f/(c \cdot g)$.
- Box-Muller gives normals from uniforms.

Key Terms

Variate: A realized value from a specified distribution.

CDF: Cumulative distribution function $F(x) = P(X \leq x)$.

Envelope: A dominating, easy-to-sample density in acceptance-rejection.

Example

- Service times $\sim \text{Exp}(\text{rate } 2/\text{min})$: generate $U=0.7$. $X = -\ln(0.7)/2 = 0.178 \text{ min}$.

Summary: Uniform numbers become any distribution via inversion, rejection, or special tricks.

Exam Tips

- ★ Very important: derive exponential via inverse transform.
- ★ Remember Box-Muller formulas.

Topic 8: Input Data Modelling

Learning Objectives

- Explain the input-modelling task
- Fit distributions to data
- Handle the no-data case

Key Concepts

Input modelling turns real-world observations into the distributions that drive the simulation. With data: (1) collect raw observations of interarrival times, service times, etc.; (2) hypothesize a family (exponential? Weibull? lognormal?) using histograms and summary statistics; (3) estimate parameters (method of moments, maximum likelihood); (4) test goodness of fit - chi-square for binned data, Kolmogorov-Smirnov comparing empirical and fitted CDFs, and Q-Q plots for visual judgment; (5) if no standard family fits, use the empirical distribution directly (resample observed values).

Without data, use expert judgment: uniform between plausible min and max, triangular with a most-likely value, or fixed constants for deterministic parts - and document these as assumptions to test with sensitivity analysis. Never silently invent precision you do not have.

Multivariate inputs need care: service time may depend on customer type, or arrivals may be nonstationary (rush hours). Then use conditional distributions or a nonstationary Poisson process (thinning: generate at the peak rate, accept arrivals with probability $\lambda(t)/\lambda_{\max}$). Correlated inputs (e.g., long jobs tend to need long setups) require joint distributions or copulas; ignoring

correlation biases results.

Key Points

- Fit: hypothesize family, estimate parameters, test with chi-square / K-S / Q-Q plots.
- No data: use triangular/uniform from expert bounds, document as assumptions.
- Handle nonstationarity (thinning) and correlation explicitly.

Key Terms

Goodness of fit: Statistical test of distribution vs. data.

Maximum likelihood: Parameter values maximizing data probability.

Thinning: Generating nonstationary arrivals by rejection.

Example

- *Fitting service times: histogram suggests exponential: K-S test accepts $\lambda = 0.5/\text{min}$.*

Summary: Inputs must be earned from data or explicit assumptions - never from thin air.

Exam Tips

- ★ Key point: the four fit steps.
- ★ Remember chi-square and K-S as fit tests.

Unit 3: Discrete-Event Simulation Mechanics

Topic 9: The Event-Scheduling World View

Learning Objectives

- Explain event scheduling
- Describe the simulation clock and event list
- Contrast with process-interaction view

Key Concepts

In the event-scheduling world view, the simulation maintains a clock (current simulated time) and a future event list (FEL) - a priority queue of scheduled events ordered by time. The executive loop: initialize clock to 0 and schedule initial events; remove the imminent event (smallest time) from the FEL; advance the clock to that event's time; execute the event routine (update state, collect statistics, schedule new future events); repeat until a stopping condition (time limit, event count, or empty FEL).

Time advances in jumps from one event time to the next - nothing happens between events, so no computation is wasted. State changes only inside event routines, which guarantees the model is consistent. Randomness enters when event routines sample variates (next arrival time, service duration).

The alternative process-interaction view follows entities through their life cycles (a customer's process: arrive, wait, seize server, delay, release, depart), which many find more intuitive; simulation languages (Arena, Simio) use it. Under the hood, though, it is still events being scheduled. Understanding event scheduling first makes every simulation tool transparent.

Key Points

- Loop: take imminent event, advance clock, execute routine, schedule new events.

- Clock jumps event-to-event; state changes only in event routines.
- Process-interaction is intuitive syntax over the same event machinery.

Key Terms

Future event list: Time-ordered queue of pending events.

Imminent event: The event with smallest scheduled time.

Event routine: Code executed when an event occurs.

Example

- At $t=3.2$ a departure event frees the server and schedules the next departure at $3.2 + \text{service time}$.

Summary: Event scheduling is the engine: ordered events, jumping clock, state updates.

Exam Tips

- ★ Very important: describe the executive loop.
- ★ Remember that time jumps between events.

Topic 10: Hand Simulation of a Single-Server Queue

Learning Objectives

- Simulate a queue by hand
- Track events, state, and statistics
- Compute average waiting time and server utilization

Key Concepts

Consider a barber shop: customers arrive with given interarrival times, one barber serves with given service times. Events: arrival and departure. State: server status (idle/busy), queue length, simulation clock. Statistics: total waiting time, number served, area under queue-length curve, server busy time.

Procedure: sort all event times; process in order. On arrival at time t : if server idle, start service immediately (waiting = 0), schedule departure at $t + \text{service}$; else join queue (record arrival time). On departure at time t : if queue nonempty, remove first customer, waiting = $t - \text{its arrival}$, start its service, schedule next departure; else server goes idle. Track clock jumps; at end compute: average waiting time = total waiting / customers served; server utilization = busy time / total time; time-average queue length = area under $Q(t)$ / total time.

Working one small example by hand (5-6 customers) teaches more than pages of theory: you feel the clock jumping, see waiting accumulate, and understand exactly what the computer will do millions of times. Every statistic the software reports is just these bookkeeping rules scaled up.

Key Points

- Events: arrival, departure; state: server status, queue length, clock.
- Waiting time, utilization, and time-average queue length from simple tallies.
- Hand simulation reveals exactly what software automates.

Key Terms

Utilization: Fraction of time the server is busy.

Time-average: Area under a curve divided by time horizon.

Tally vs. time-persistent statistics: Per-entity values vs. time-weighted curves.

Example

- 5 customers. average wait 2.1 min. utilization 0.83 - computed entirely by hand.

Summary: Hand simulation makes the mechanics concrete before code takes over.

Exam Tips

- ★ Key point: be able to hand-simulate 5 customers.
- ★ Remember formulas for waiting time and utilization.

Topic 11: Queueing Systems and Kendall Notation

Learning Objectives

- Use Kendall's A/B/c/K/m/Z notation
- Define traffic intensity
- Classify common queueing models

Key Concepts

Queueing theory gives analytical results for simple systems - the perfect check on simulations. Kendall notation A/B/c/K/m/Z describes a queue: A = interarrival-time distribution, B = service-time distribution (M = exponential/Markovian, D = deterministic, G = general, E_k = Erlang-k), c = number of parallel servers, K = system capacity (default infinite), m = calling population size (default infinite), Z = queue discipline (default FCFS).

Traffic intensity $\rho = \lambda / (c * \mu)$, where λ is the arrival rate and μ the per-server service rate. Stability requires $\rho < 1$ - otherwise the queue grows without bound and no steady state exists. A simulation of an unstable queue ($\rho \geq 1$) produces ever-growing, meaningless averages; always check stability first.

The M/M/1 queue (exponential arrivals, exponential service, one server) has closed forms: mean number in system $L = \rho / (1 - \rho)$, mean time in system $W = 1 / (\mu - \lambda)$, mean waiting in queue $W_q = \rho / (\mu - \lambda)$, by Little's law $L = \lambda * W$. These formulas validate simulations: run the model, and the estimates should land near theory. M/M/c, M/G/1 (Pollaczek-Khintchine formula), and G/G/1 approximations extend the toolkit.

Key Points

- Kendall A/B/c: arrival/service distributions and server count.
- Stability needs $\rho = \lambda / (c * \mu) < 1$.
- M/M/1 formulas (L , W , W_q) validate simulation output.

Key Terms

Traffic intensity: ρ : offered load per server.

Little's law: $L = \lambda * W$, linking counts and times.

Steady state: Long-run behavior independent of initial conditions.

Example

- M/M/1 with $\lambda=2/\text{min}$. $\mu=3/\text{min}$: $\rho=0.67$. $W = 1 \text{ min}$ - simulation should agree.

Summary: Queueing theory provides the analytical yardstick for simulation experiments.

Exam Tips

★ Very important: Kendall notation and M/M/1 formulas. 

★ Remember the stability condition $\rho < 1$.

Topic 12: Little's Law and Performance Measures

Learning Objectives

- State and apply Little's law
- Distinguish tally and time-persistent statistics
- Choose measures for a study

Key Concepts

Little's law - $L = \lambda * W$ - is the most useful result in queueing: the long-run average number in a stable system equals the arrival rate times the average time in the system. It holds for the whole system (L, W) and for the queue alone ($L_q = \lambda * W_q$), under astonishingly general conditions (any distributions, any discipline, as long as the system empties infinitely often in the long run). In simulation it is a built-in consistency check: estimate L, λ, W independently, and they must satisfy the law.

Statistics come in two flavors. Tally (observational) statistics: one value per entity - waiting times, system times; summarized by mean, max, histogram. Time-persistent statistics: values defined at every instant - queue length $Q(t)$, server busy $B(t)$; summarized by time-average = (area under curve)/ T and by max. Server utilization is the time-average of the busy indicator.

Choosing measures: match them to the study's objectives - a bank cares about customer waiting (tally) and teller utilization (time-persistent); a factory cares about throughput, work-in-process (Little's law again), and makespan. Always report both means and variability; a mean without a confidence interval is a rumor.

Key Points

- Little's law: $L = \lambda * W, L_q = \lambda * W_q$ - universal consistency check.
- Tally stats per entity; time-persistent stats via area under curve.
- Report means with variability, never bare means.

Key Terms

Throughput: Long-run rate of completions.

Work in process: Average number of entities in the system.

Makespan: Time to complete a set of jobs.

Example

- Simulated $L = 3.2, \lambda = 4/\text{hr}, W = 0.8 \text{ hr}: 4 * 0.8 = 3.2 \text{ checks out.}$

Summary: Little's law ties the whole performance picture together.

Exam Tips

- ★ Very important: state Little's law and apply it.
- ★ Remember tally vs. time-persistent distinction.

Unit 4: Verification, Validation, and Output Analysis

Topic 13: Verification and Validation

Learning Objectives

- Distinguish verification from validation
- List V&V; techniques
- Explain calibration

Key Concepts

Verification asks "did we build the model right?" - is the code a correct implementation of the conceptual model? Techniques: structured walkthroughs of code, tracing a single entity through the model step by step, testing with simplified inputs (deterministic arrivals give predictable results), checking intermediate outputs against hand calculations, animation (watching entities move exposes logic errors instantly), and stress tests with extreme inputs (zero arrivals, infinite queue capacity).

Validation asks "did we build the right model?" - does it represent the real system accurately enough for the study's purpose? There is no universal validity, only validity for a purpose. Techniques: face validation (experts review the conceptual model), comparing model output with real system data (historical validation), Turing-style tests (can experts distinguish model output from real data?), sensitivity analysis (doubtful parameters should not drive conclusions), and predictive validation (forecast, then compare with later reality).

Calibration tunes parameters so the model matches a data set; validation then requires a fresh, independent data set - calibrating and validating on the same data proves nothing. Document every assumption and every validation test; a model whose limits are stated honestly is more useful than a flashy one whose limits are hidden.

Key Points

- Verification = code correctness; validation = model adequacy for purpose.
- Techniques: traces, animation, expert review, historical comparison, sensitivity analysis.
- Calibrate on one data set, validate on another.

Key Terms

Face validation: Expert judgment that the model looks right.

Calibration: Tuning parameters to match data.

Sensitivity analysis: Testing how results change with inputs.

Example

- *Tracing one customer through the bank model catches a queue-discipline bug.*

Summary: V&V; is what makes simulation credible rather than merely colorful.

Exam Tips

- ★ Very important: verification vs. validation with one technique each.
- ★ Remember: calibrate and validate on different data.

Topic 14: Output Analysis: Confidence Intervals

Learning Objectives

- Explain why replication is needed
- Construct confidence intervals for the mean
- Choose the number of replications

Key Concepts

Stochastic simulation output is random: one run gives one noisy observation. The remedy is independent replications - run the model n times with different random seeds (but identical initial conditions), obtaining observations $Y_1, \dots, Y_n$ of the performance measure. These are IID, so classical statistics apply: point estimate $\bar{Y}$, sample variance S^2 , and the $100(1-\alpha)\%$ confidence interval $\bar{Y} \pm t_{\{n-1, 1-\alpha/2\}} S / \sqrt{n}$.

The interval's half-width shrinks as $1/\sqrt{n}$: quadrupling replications halves the width. Choose n by a precision requirement: keep adding replications until the half-width is, say, under 5% of the estimate (sequential procedure). Report intervals, not just means - "average wait 4.2 $\pm$ 0.3 min (95% CI)" is honest; "average wait 4.2 min" hides the uncertainty.

Within one long run, observations are autocorrelated (a long queue now predicts a long queue soon), so naive formulas understate error. Fixes: replications (simplest and safest), batch means (split one run into batches, treat batch means as IID), or regenerative cycles. Never compute a confidence interval from raw correlated observations as if they were independent.

Key Points

- Replicate with different seeds; output means are IID across replications.
- 95% CI: $\bar{Y} \pm t * S/\sqrt{n}$; half-width shrinks as $1/\sqrt{n}$.
- Autocorrelation within a run invalidates naive intervals - use batch means.

Key Terms

Replication: Independent run with a different seed.

Half-width: Precision radius of the confidence interval.

Batch means: Averaging blocks of one run to break autocorrelation.

Example

- 10 replications give wait 4.2 $\pm$ 0.3 min: 40 replications would give $\sim \pm$ 0.15.

Summary: Statistics turn noisy simulation output into trustworthy estimates.

Exam Tips

- ★ Very important: write the confidence-interval formula.
- ★ Remember why one long run's raw data cannot be treated as IID.

Topic 15: Comparing System Alternatives

Learning Objectives

- Compare two designs statistically
- Explain common random numbers
- Avoid the overlapping-CI trap

Key Concepts

Simulation usually compares alternatives: two tellers vs. three, layout A vs. layout B. For two systems, estimate the difference $D = Y_1 - Y_2$. With independent replications of each, build a paired or two-sample confidence interval on the mean difference; if the interval excludes zero, the difference is statistically significant. A frequent trap: two overlapping individual confidence intervals do NOT prove the systems are equal - always interval the difference directly.

Common random numbers (CRN): drive both alternatives with the same random seeds (same arrival streams, same service draws) so that differences in output reflect differences in design, not luck. This induces positive correlation between Y_1 and Y_2 , and since $\text{Var}(Y_1 - Y_2) = \text{Var}(Y_1) + \text{Var}(Y_2) - 2\text{Cov}$, the variance of the difference shrinks - sharper comparisons with the same computing budget. Synchronize streams carefully (dedicated stream per random source); sloppy synchronization can backfire.

For more than two alternatives, use ANOVA followed by pairwise comparisons with Bonferroni or Tukey corrections - uncorrected multiple testing inflates false discoveries. And statistical significance is not practical significance: a 2-second wait reduction significant at $p < 0.01$ may not justify a new hire.

Key Points

- Compare via confidence interval on the difference, not overlapping intervals.
- Common random numbers correlate alternatives to sharpen the comparison.
- Correct for multiple comparisons; distinguish statistical from practical significance.

Key Terms

Common random numbers: Same seeds across alternatives to reduce variance.

Paired comparison: Differencing matched replications.

Bonferroni correction: Adjusting significance for multiple tests.

Example

- *Same arrival stream for 2-teller and 3-teller models isolates the staffing effect.*

Summary: Fair comparison needs the right statistics and synchronized randomness.

Exam Tips

- ★ Key point: why overlapping CIs do not imply equality.
- ★ Remember the variance-reduction logic of CRN.

Topic 16: Variance Reduction Techniques

Learning Objectives

- Explain the goal of variance reduction
- Describe antithetic variates
- Describe control variates

Key Concepts

Variance reduction gets more precision per unit of computing - or equal precision cheaper. All techniques trade a little extra analysis for a lot less simulation. Antithetic variates: pair replications using complementary random numbers (U and $1 - U$); if the response is monotonic in the inputs, the pair's outputs are negatively correlated, so their average has smaller variance than two independent runs. Simple to implement, modest gains.

Control variates: exploit a correlated quantity with known mean. To estimate $E[Y]$ (say, total cost), find X correlated with Y whose mean μ_X is known analytically (say, total service time in an $M/M/1$). Use $Y' = Y - a(X - \mu_X)$ with optimal $a = \text{Cov}(X, Y) / \text{Var}(X)$; the correction removes known variation, leaving a tighter estimate. Needs analytical work but can cut variance dramatically.

Other tools: importance sampling (sample more from critical regions - rare-event simulation), stratified sampling, and conditional Monte Carlo. Rule of thumb: first get the model right, then replicate enough; reach for variance reduction when runs are expensive (large networks, rare events) and the technique's assumptions hold.

Key Points

- Goal: same precision with fewer runs (or better precision with same runs).
- Antithetic variates: negative correlation via U and $1-U$.
- Control variates: subtract a correlated known-mean quantity.

Key Terms

Antithetic variates: Complementary random numbers inducing negative correlation.

Control variate: Known-mean auxiliary variable for correction.

Importance sampling: Oversampling critical regions.

Example

- *Control variate: known M/M/1 mean queue length corrects simulated estimates.*

Summary: Smart statistics squeeze more truth from every replication.

Exam Tips

- ★ Key point: one technique with its mechanism.
- ★ Remember the goal: precision per compute cost.

Unit 5: Advanced Simulation Topics

Topic 17: Monte Carlo Simulation

Learning Objectives

- Define Monte Carlo simulation
- Apply it to integration and risk
- Explain convergence

Key Concepts

Monte Carlo simulation estimates quantities by random sampling - the static, non-time-evolving cousin of discrete-event simulation. The classic demo: estimate π by throwing random darts at a unit square containing a quarter circle; the fraction landing inside tends to $\pi/4$. More seriously: estimate integrals that defeat calculus, price financial options, or propagate uncertainty through a spreadsheet model (each input a distribution, the output a distribution of possible profits).

The mathematics: the sample mean of IID draws converges to the true expectation (law of large numbers), with error shrinking as $1/\sqrt{n}$ - to halve the error, quadruple the samples. This slow convergence is Monte Carlo's price; its virtue is that the rate does not depend on dimension, so it beats grid methods in high dimensions (a 20-variable integral is hopeless for grids, routine for Monte Carlo).

Risk analysis example: a project's cost = sum of uncertain task costs; simulate 10,000 times, read off $P(\text{cost} > \text{budget})$ and the 90th percentile. Decision-makers get not a single number but a risk profile - far more honest. Variance reduction (Lecture 16) and quasi-random sequences (low-discrepancy) speed convergence when needed.

Key Points

- Monte Carlo = estimation by random sampling; static models.
- Error shrinks as $1/\sqrt{n}$, independent of dimension.
- Risk analysis: outputs are distributions, not single numbers.

Key Terms

Law of large numbers: Sample means converge to expectations.

Risk profile: Distribution of possible outcomes.

Quasi-random: Low-discrepancy sequences for faster convergence.

Example

- 10.000 samples estimate a project's $P(\text{overrun}) = 0.23$.

Summary: Monte Carlo turns uncertainty into quantified risk.

Exam Tips

- ★ Key point: the $1/\sqrt{n}$ convergence rule.
- ★ Remember the pi-by-darts example.

Topic 18: Continuous and Combined Simulation

Learning Objectives

- Distinguish continuous simulation
- Give examples of combined models
- Name solution approaches

Key Concepts

Continuous simulation models systems whose state evolves smoothly, described by differential equations: chemical reactors, aircraft dynamics, population growth, reservoir levels. Time advances in small fixed or adaptive steps (Euler, Runge-Kutta integration); state updates via the equations. Tools like Simulink and Modelica dominate this world.

Combined discrete-continuous simulation mixes both: a refinery where tank levels change continuously but valves open/close at discrete events; a production line with continuous material flow and discrete machine breakdowns. The executive must handle event scheduling alongside numerical integration, detecting threshold crossings (when a continuous variable triggers a discrete event - the tank hits empty).

Choosing the view: discrete for queues, logistics, service systems; continuous for physical/chemical/biological dynamics; combined for process industries. Many real systems need the combination, and modern tools (AnyLogic's multi-method modelling) support all three paradigms - discrete-event, agent-based, and system dynamics - in one model.

Key Points

- Continuous simulation: differential equations, small time steps.
- Combined models mix smooth evolution with discrete events.
- Match the world view to the system's nature.

Key Terms

System dynamics: Stock-and-flow continuous modelling (Forrester).

Threshold event: Discrete event triggered by a continuous variable.

Agent-based modelling: Autonomous agents interacting; emergence studied.

Example

- *Tank level falls continuously; the 'tank empty' event stops the pump discretely.*

Summary: Different dynamics need different simulation paradigms - sometimes all at once.

Exam Tips

- ★ Key point: one example of a combined model.
- ★ Remember Euler/Runge-Kutta as integrators.

Topic 19: Simulation Software and Languages

Learning Objectives

- Survey simulation software categories
- Compare general-purpose vs. specialized tools
- Outline a modelling workflow in software

Key Concepts

Three categories. (1) General-purpose languages (Python with SimPy, C++, Java): maximum flexibility, full control of the event machinery, free - but you build everything (FEL, statistics, animation) yourself. Best for research, learning, and unusual models. (2) Simulation languages/packages (Arena, Simio, AnyLogic, ExtendSim): built-in event executive, statistics collection, animation, and experiment managers; faster model building for standard domains (manufacturing, logistics, healthcare). (3) Spreadsheets with add-ins (@RISK, Crystal Ball): Monte Carlo on spreadsheet models - accessible to managers, limited for dynamic systems.

Typical workflow in a package: draw the process flow (create-arrive-seize-delay-release-dispose modules), attach distributions to activities, define resources and schedules, set run length and replications, verify with animation and traces, validate against data, design experiments (scenario manager), analyze with built-in statistics, and present animated results to stakeholders.

Choosing: match the tool to the problem's structure and the team's skills; a simple queue rarely needs a heavyweight package, while a hospital with 50 interacting resources does. Open-source options (SimPy, JaamSim) have matured enough for serious work.

Key Points

- Three categories: general-purpose languages, simulation packages, spreadsheet add-ins.
- Packages provide event executive, statistics, animation, experiment management.
- Match tool weight to problem complexity.

Key Terms

SimPy: Python discrete-event simulation library.

Arena/Simio: Commercial visual simulation packages.

Scenario manager: Tool for running and comparing experiments.

Example

- *Modelling a factory line in Simio: machines as servers. conveyors as paths. animation for validation.*

Summary: Software choice is a trade-off between control, speed, and cost.

Exam Tips

- ★ Key point: name the three categories with one example each.
- ★ Remember the typical package workflow.

Topic 20: Case Studies and Project Guidance

Learning Objectives

- Walk through two mini case studies
- Outline a simulation project report
- List common pitfalls

Key Concepts

Case 1 - Bank branch: arrivals Poisson ($\lambda = 20/\text{hr}$), service exponential ($\mu = 25/\text{hr}$), 1 teller. $\rho = 0.8$; M/M/1 predicts $W_q = 0.16 \text{ hr} = 9.6 \text{ min}$. Simulation with 20 replications of 8-hour days gives $9.4 \pm 1.1 \text{ min}$ - validated. Experiment: adding a second teller (M/M/2) cuts W_q to $\sim 1 \text{ min}$; cost-benefit decides.

Case 2 - Inventory: a warehouse reviews stock weekly; demand $\sim \text{Normal}(100, 15)$. Monte Carlo over reorder points finds the (s, S) policy minimizing holding + shortage cost. Sensitivity analysis shows the optimum is robust to demand variance - a trustworthy recommendation.

A good project report: problem and objectives, conceptual model and assumptions, input data and fitted distributions, verification/validation evidence, experimental design, results with confidence intervals, comparison of alternatives, recommendations with caveats, and an appendix of details. Common pitfalls: no clear objective ("let's simulate everything"), unvalidated models, single replications reported as truth, correlated output treated as independent, and confusing animation beauty with model quality. Avoid these and your study will stand up to scrutiny.

Key Points

- Case studies show the full pipeline: model, validate, experiment, decide.
- Reports: objectives, assumptions, V&V, design, intervals, recommendations.
- Pitfalls: no objective, no validation, single runs, ignoring autocorrelation.

Key Terms

Reorder point: Inventory level triggering an order.

(s, S) policy: Order up to S when stock hits s .

Robustness: Recommendation stable under assumption changes.

Example

- *Second teller cuts simulated wait from ~ 9.6 to $\sim 1 \text{ min}$ - the experiment that justifies hiring.*

Summary: Disciplined casework turns simulation from exercise into decision support.

Exam Tips

- ★ Very important: list three common pitfalls.
- ★ Remember report structure.

Unit 6: Worked Problems and Exam Preparation

Topic 21: Worked Problem: Linear Congruential Generator

Learning Objectives

- Generate an LCG sequence by hand
- Compute the period
- Convert to Uniform(0,1)

Key Concepts

Problem: An LCG uses modulus $m = 16$, multiplier $a = 5$, increment $c = 3$, and seed $X_0 = 7$. Generate the first 6 values and the corresponding uniforms $U_n = X_n / m$.

Solution: Apply $X_{n+1} = (5 * X_n + 3) \bmod 16$. $X_1 = (5*7 + 3) \bmod 16 = 38 \bmod 16 = 6$; $U_1 = 6/16 = 0.375$. $X_2 = (5*6 + 3) \bmod 16 = 33 \bmod 16 = 1$; $U_2 = 0.0625$. $X_3 = (5*1 + 3) \bmod 16 = 8 \bmod 16 = 8$; $U_3 = 0.5$. $X_4 = (5*8 + 3) \bmod 16 = 43 \bmod 16 = 11$; $U_4 = 0.6875$. $X_5 = (5*11 + 3) \bmod 16 = 58 \bmod 16 = 10$; $U_5 = 0.625$. $X_6 = (5*10 + 3) \bmod 16 = 53 \bmod 16 = 5$; $U_6 = 0.3125$.

Continuing would show the sequence returns to 7 after 16 steps - full period $m = 16$, because $c = 3$ is coprime to 16 and $a - 1 = 4$ is divisible by 2 (the only prime factor of 16). The uniforms look evenly spread, as required. Exam note: always show the modular reduction step explicitly; markers check it.

Key Points

- Apply the recurrence, reduce mod m , divide by m for U .
- Full period m achievable with proper (a, c, m) choices.
- Show every modular reduction step in exams.

Key Terms

Full period: Sequence visits all m values before repeating.

Example

- $X_1 = (5*7+3) \bmod 16 = 6$. so $U_1 = 0.375$.

Summary: LCG hand computation is mechanical: multiply, add, reduce, divide.

Exam Tips

- ★ Very important: practice one full LCG trace.
- ★ Remember $U_n = X_n / m$.

Topic 22: Worked Problem: Inverse-Transform Sampling

Learning Objectives

- Apply inverse transform to the exponential
- Generate discrete variates
- Interpret the results

Key Concepts

Problem A: Service times are exponential with mean 4 minutes ($\lambda = 0.25/\text{min}$). Using uniforms $U = 0.30$ and $U = 0.80$, generate two service times.

Solution: $X = -(1/\lambda) \cdot \ln(U) = -4 \cdot \ln(U)$. For $U = 0.30$: $X = -4 \cdot \ln(0.30) = -4 \cdot (-1.204) = 4.82$ minutes. For $U = 0.80$: $X = -4 \cdot \ln(0.80) = -4 \cdot (-0.223) = 0.89$ minutes. Note the spread: the same distribution yields both long and short times - that variability is the point of stochastic modelling.

Problem B: A discrete demand distribution: $P(D=0)=0.2$, $P(D=1)=0.5$, $P(D=2)=0.3$. Generate demands for $U = 0.65$ and $U = 0.10$.

Solution: Build the CDF: $F(0)=0.2$, $F(1)=0.7$, $F(2)=1.0$. Pick the smallest d with $F(d) \geq U$. $U = 0.65$: $F(0)=0.2 < 0.65$, $F(1)=0.7 \geq 0.65$, so $D = 1$. $U = 0.10$: $F(0)=0.2 \geq 0.10$, so $D = 0$. Exam note: draw the CDF intervals $[0,0.2)$, $[0.2,0.7)$, $[0.7,1.0]$ - it prevents off-by-one errors.

Key Points

- Exponential: $X = -(1/\lambda) \ln(U)$.
- Discrete: smallest x with $F(x) \geq U$.
- Sketch CDF intervals to avoid errors.

Key Terms

Mean of exponential: $1/\lambda$.

Example

- $U=0.30$. mean 4 min: $X = 4.82$ min.

Summary: Inversion is the workhorse: one formula for continuous, one rule for discrete.

Exam Tips

- ★ Very important: memorize the exponential inversion.
- ★ Remember the CDF-interval sketch.

Topic 23: Worked Problem: Hand Simulation and M/M/1 Check

Learning Objectives

- Hand-simulate arrivals and departures
- Compute waiting time and utilization
- Verify against M/M/1 theory

Key Concepts

Problem: Customers arrive at times 0, 3, 5, 9 (minutes) with service times 4, 2, 5, 3 respectively. Single server, FCFS. Compute each customer's waiting time, the average wait, and server utilization over the run (ends when the last customer departs).

Solution: Customer 1 arrives $t=0$, server idle: service 0-4, wait 0. Customer 2 arrives $t=3$, server busy until 4: waits 1, service 4-6, wait 1. Customer 3 arrives $t=5$, server busy until 6: waits 1, service 6-11, wait 1. Customer 4 arrives $t=9$, server busy until 11: waits 2, service 11-14, wait 2.

Total waiting = $0+1+1+2 = 4$; average wait = $4/4 = 1.0$ minute. Server busy from 0 to 14 continuously = 14 minutes; run length 14; utilization = $14/14 = 1.0$ (100% - the server never idled because arrivals kept coming).

M/M/1 check: suppose instead $\lambda = 3/\text{hr}$, $\mu = 4/\text{hr}$. $\rho = 3/4 = 0.75$. $L = \rho/(1-\rho) = 3$ customers. $W = 1/(\mu - \lambda) = 1$ hr. $W_q = \rho/(\mu - \lambda) = 0.75$ hr = 45 min. Little's law check: $L_q = \lambda \cdot W_q = 3 \cdot 0.75 = 2.25$, and $\rho^2/(1-\rho) = 2.25$. Consistent. A simulation of this queue should produce estimates near these values.

Key Points

- Track each customer: arrival, service start, wait, departure.
- Utilization = busy time / total time.
- M/M/1: $\rho = \lambda / \mu$, $W = 1 / (\mu - \lambda)$, verify with Little's law.

Key Terms

FCFS: First-come, first-served discipline.

Example

- Four customers above: average wait 1.0 min. utilization 100%.

Summary: Hand simulation plus theory gives you two independent checks on any model.

Exam Tips

- ★ Very important: practice a 4-customer trace.
- ★ Memorize M/M/1 formulas.

Topic 24: Worked Problem: Confidence Interval from Replications

Learning Objectives

- Compute a confidence interval from replication data
- Decide how many replications suffice
- Interpret the interval

Key Concepts

Problem: Eight replications of a queue simulation give average waiting times (minutes): 4.1, 5.2, 3.8, 4.9, 4.4, 5.0, 4.2, 4.7. Construct the 95% confidence interval for the true mean waiting time.

Solution: $n = 8$. Sample mean $\bar{Y} = (4.1 + 5.2 + 3.8 + 4.9 + 4.4 + 5.0 + 4.2 + 4.7) / 8 = 36.3 / 8 = 4.5375$. Sample variance: $\text{sum of squared deviations} = (4.1 - 4.5375)^2 + \dots = 0.1914 + 0.4390 + 0.5439 + 0.1314 + 0.0189 + 0.2139 + 0.1139 + 0.0264 = 1.6788$; $S^2 = 1.6788 / 7 = 0.2398$; $S = 0.4897$. $t_{\{7, 0.975\}} = 2.365$. Half-width = $2.365 * 0.4897 / \sqrt{8} = 2.365 * 0.1731 = 0.4094$. CI: 4.54 ± 0.41 , i.e., (4.13, 4.95) minutes.

Interpretation: we are 95% confident the true long-run mean waiting time lies between 4.13 and 4.95 minutes. If the requirement is half-width under 0.2, note half-width shrinks as $1/\sqrt{n}$: need roughly $(0.41/0.2)^2 = 4.2$ times more replications, about 34 total. Exam note: use t , not z , for small n ; divide variance by $(n-1)$.

Key Points

- $CI = \bar{Y} \pm t_{\{n-1\}} * S / \sqrt{n}$.
- Half-width $\sim 1/\sqrt{n}$: quadrupling n halves it.
- Report intervals, not bare means.

Key Terms

Degrees of freedom: $n - 1$ for the t critical value.

Example

- 8 replications: 4.54 ± 0.41 min (95% CI).

Summary: Replication statistics convert noisy runs into honest estimates.

Exam Tips

- ★ Very important: compute a full CI from data.
- ★ Remember t vs. z and the $1/\sqrt{n}$ rule.

Topic 25: Formula Sheet and Final Exam Strategy

Learning Objectives

- Consolidate all key formulas
- Plan exam time allocation
- Avoid classic mistakes

Key Concepts

Formula sheet. LCG: $X_{n+1} = (a X_n + c) \bmod m$. Inverse transform: $X = F^{-1}(U)$; exponential $X = -(1/\lambda) \ln(U)$. Box-Muller: $Z = \sqrt{-2 \ln U_1} \cos(2 \pi U_2)$. Erlang-k = sum of k exponentials. Kendall: $A/B/c$. $\rho = \lambda/(c \mu) < 1$ for stability. M/M/1: $L = \rho/(1-\rho)$, $L_q = \rho^2/(1-\rho)$, $W = 1/(\mu - \lambda)$, $W_q = \rho/(\mu - \lambda)$. Little: $L = \lambda W$. Utilization = busy/total. Time-average = area/T. CI: $\bar{Y} \pm t_{n-1, 1-\alpha/2} S/\sqrt{n}$. CRN variance: $\text{Var}(\text{diff}) = V_1 + V_2 - 2\text{Cov}$.

Exam strategy: (1) Read the whole paper first; start with definitions and short numericals (LCG, inversion, M/M/1) - they are quick marks. (2) For hand-simulation questions, make a table with columns: customer, arrival, service, start, wait, departure - neat tables earn method marks even with arithmetic slips. (3) For "explain" questions (verification vs. validation, CRN, thinning), use the one-sentence definition first, then one example. (4) Always state assumptions (stability, independence) - examiners reward them.

Classic mistakes: confusing verification with validation; reporting a single replication's mean as the answer; treating correlated within-run data as independent; forgetting $\rho < 1$; using z instead of t for small n; and simulating when the question asks for the analytical M/M/1 answer. Write units (minutes, customers) - a number without units is incomplete.

Key Points

- Master the formula sheet above - it covers ~80% of numericals.
- Tables for hand simulation; definitions first for theory.
- State assumptions and units always.

Key Terms

Method marks: Credit for correct procedure despite arithmetic slips.

Example

- Start with: 'Verification checks the code: validation checks the model.'

Summary: Preparation plus presentation: know the formulas, show the method, state the assumptions.

Exam Tips

- ★ Very important: memorize the M/M/1 and CI formulas.
- ★ Remember: units and assumptions earn marks.

Unit 7: Mini-Project: Complete Walkthrough

Topic 26: Project Brief: Campus Photocopy Shop

Learning Objectives

- State a complete simulation problem
- Define objectives and measures
- List assumptions

Key Concepts

Problem: A campus photocopy shop has one machine. Students arrive as a Poisson process at 12 per hour; copying jobs take exponential time with mean 4 minutes. The owner asks: is one machine enough, or should she buy a second? Objectives: estimate mean student waiting time and machine utilization with 95% confidence; compare one-machine vs. two-machine designs; recommend the cheaper option meeting a 5-minute mean-wait target.

Measures: average waiting time in queue (tally), machine utilization (time-persistent), and maximum queue length observed. Assumptions: Poisson arrivals (verified from a day of timestamped data), exponential service (fitted, K-S accepted), FCFS discipline, no balking, 8-hour working day with the shop starting empty each morning (terminating simulation - each day is a replication).

First check analytically: $\lambda = 12/\text{hr} = 0.2/\text{min}$, $\mu = 1/4 = 0.25/\text{min}$, $\rho = 0.8$. M/M/1 predicts $W_q = \rho/(\mu - \lambda) = 0.8/0.05 = 16 \text{ min}$ - far above the 5-minute target. One machine is almost surely inadequate; the simulation will quantify exactly how bad and test the two-machine alternative.

Key Points

- Clear objectives and measures come before any modelling.
- Terminating simulation: each day is a natural replication.
- Analytical M/M/1 check first: $W_q = 16 \text{ min}$ predicts failure.

Key Terms

Terminating simulation: Runs with a natural end event (the working day).

Example

- *$\rho = 0.8$ gives predicted wait 16 min vs. 5-min target.*

Summary: A well-posed problem makes the rest of the study straightforward.

Exam Tips

- ★ Key point: always do the analytical sanity check first.
- ★ Remember terminating vs. steady-state designs.

Topic 27: Building and Verifying the Model

Learning Objectives

- Translate the problem into events
- Describe verification tests
- Explain warm-up for steady-state

Key Concepts

Events: arrival (schedule next arrival; if machine idle start service else join queue) and departure (record wait; if queue nonempty start next service else idle the machine). Random streams: stream 1 for

interarrival times, stream 2 for service times - separate streams keep the experiment clean and let common random numbers be applied later.

Verification: (1) trace three students through by hand and match program output; (2) run with deterministic arrivals every 5 minutes and service exactly 4 minutes - queues should never form and utilization should read exactly 0.8; (3) animate one day and watch students flow; (4) check Little's law on output (L_q vs. $\lambda \cdot W_q$). Any mismatch means a bug, not a discovery.

Design choice: although each day starts empty, the owner cares about long-run performance, so run each replication as a full 8-hour day including the morning transient - the transient is part of the real day she experiences. Twenty replications (20 simulated days) give solid confidence intervals. For the two-machine variant, reuse the same seeds (common random numbers) for a fair comparison.

Key Points

- Two events (arrival, departure) suffice for the single-server model.
- Verify with traces, deterministic tests, animation, and Little's law.
- Separate random streams; reuse seeds across designs (CRN).

Key Terms

Warm-up/transient: Initial period before typical behavior - included here deliberately.

Example

- *Deterministic test: arrivals every 5 min. service 4 min -> utilization exactly 0.8.*

Summary: Verification is debugging with discipline; do it before believing any number.

Exam Tips

- ★ Key point: name three verification tests.
- ★ Remember why separate streams matter.

Topic 28: Results, Comparison, and Recommendation

Learning Objectives

- Report results with confidence intervals
- Compare designs statistically
- Make a justified recommendation

Key Concepts

Results (20 replications): one machine - mean wait 15.6 +/- 1.8 min, utilization 0.80 +/- 0.01, max queue 14. The interval (13.8, 17.4) sits far above the 5-minute target and agrees with the M/M/1 prediction of 16 min - the model is validated and the verdict is clear.

Two machines (M/M/2, same seeds): mean wait 0.9 +/- 0.2 min, utilization 0.40 each. Difference interval (one minus two): 14.7 +/- 1.7 min - excludes zero, so the improvement is statistically significant and, at ~15 minutes saved per student, practically significant too.

Recommendation: buy the second machine. Cost-benefit: a second machine costs Rs. 150,000; with 96 students/day each saving ~15 minutes, the payback in student satisfaction (and the shop's reputation during admission season) is compelling. Caveats stated honestly: results assume the arrival rate stays near 12/hr - if admissions double traffic, re-run the study; the exponential service fit should be rechecked yearly. This is how a simulation study earns its keep: a quantified, caveated, actionable

answer.

Key Points

- Report intervals: 15.6 +/- 1.8 min validates against theory (16 min).
- Compare via the difference interval with common random numbers.
- Recommendations need costs, caveats, and re-check triggers.

Key Terms

Validated verdict: Simulation agreeing with theory and data.

Example

- *Two machines: wait drops to 0.9 +/- 0.2 min - recommendation iustified.*

Summary: The study ends in a decision, not a number.

Exam Tips

- ★ Key point: structure of a recommendation (result, cost, caveat).
- ★ Remember to compare via difference intervals.

Topic 29: What You Can Build Next

Learning Objectives

- List natural extensions
- Connect to course topics
- Encourage independent practice

Key Concepts

Extensions of the photocopy project, each a real exercise: (1) add balking - students leave if the queue exceeds 8 (state-dependent arrivals); (2) make arrivals nonstationary - lunch rush at double rate, modelled by thinning; (3) add a second service - binding counter with its own queue, a small network; (4) optimize: find the cheapest staffing (machines + operator hours) meeting the wait target - a simulation-optimization loop; (5) rare events: estimate the probability that 20+ students queue during admission week (importance sampling territory).

Each extension maps to this handout: balking to event-scheduling, nonstationarity to input modelling, networks to queueing theory, optimization to experimental design, rare events to variance reduction.

Rebuilding this handout's worked problems from scratch - without looking - is the single best exam preparation: if you can hand-simulate, invert, and interval, you own the course.

Final advice: simulation rewards the paranoid - verify, validate, replicate, and doubt your own model until the evidence forces belief. That habit, more than any formula, is what this course is really teaching.

Key Points

- Five concrete extensions, each mapping to a handout topic.
- Rebuild worked problems from scratch as exam prep.
- The meta-skill: disciplined doubt - verify, validate, replicate.

Key Terms

Balking: Arriving customers refusing to join a long queue.

Example

- *Extension: lunch-rush arrivals via thinning of a peak-rate process.*

Summary: The course ends; the modelling habit continues.

Exam Tips

- ★ Remember: verification and validation are habits, not steps.

Glossary of Key Terms

Simulation: Imitation of a system's operation over time to study it.

Model: Assumptions about a system expressed mathematically/logically.

Event: Instantaneous occurrence changing system state.

State variables: Variables describing the system at a time.

Future event list: Time-ordered queue of pending events.

Verification: Checking the model is implemented correctly.

Validation: Checking the model matches reality adequately.

Pseudorandom numbers: Deterministic sequences behaving statistically randomly.

LCG: Linear congruential generator: $X_{n+1} = (aX_n + c) \bmod m$.

Inverse transform: Generating variates via $X = F^{-1}(U)$.

Kendall notation: A/B/c/K/m/Z queue description.

Traffic intensity: $\rho = \lambda / (c \cdot \mu)$; stability needs $\rho < 1$.

Little's law: $L = \lambda \cdot W$: counts equal rate times time.

Replication: Independent simulation run with a different seed.

Confidence interval: Range expressing estimate precision.

Common random numbers: Same seeds across alternatives for sharper comparison.

Monte Carlo: Estimation by random sampling of static models.

Discrete-event simulation: State changes only at event times.