

CSE-6730-A/CX-4230-A: Modeling and Simulation — Foundations and Implementation

Fall 2026 · Georgia Institute of Technology · School of Computational Science and Engineering

Instructor: Peng Chen **Credits:** 3 · **Section:** A · **CRN:** 89226 **Meeting time:** Monday and Wednesday, 5:00–6:15 PM **Location:** Instructional Center 211 **Instructor office hours:** Tuesdays, 2:00–3:00 PM, **virtual** — [Zoom](#). **Course communication:** Canvas for announcements, grades, and submissions of record; **Ed Discussion** for questions. Please use Ed rather than email for any question whose answer could help another student.

1. Course Description

The Georgia Tech catalog describes CSE 6730 as covering the foundations and algorithms needed to develop conceptual models of systems and realize them as computer simulations, including both discrete and continuous models.

This offering treats modeling and simulation as the science of constructing **trustworthy computational representations of natural, engineered, and computational systems**. The organizing workflow is:

real system → **scientific question** → **conceptual model** → **mathematical/computational model** → **numerical algorithm** → **implementation** → **verification** → **computational experiment** → **validation and uncertainty** → **prediction or decision**

Students will learn how to decide what to model, choose an appropriate simulation paradigm, construct and implement the model, determine whether the computation is correct, assess whether the model is credible for its intended purpose, and communicate what the resulting simulation can and cannot support. Classical foundations in continuous and discrete simulation are combined with selected modern topics including model reduction, uncertainty quantification, inverse problems and data assimilation, scalable simulation, scientific machine learning, and digital twins.

2. Course Philosophy

Three questions recur throughout the semester.

What should we model? Identifying system boundaries, states, parameters, inputs, outputs, scales, interactions, assumptions, and neglected effects.

How should we simulate it? Selecting among continuous-time, discrete-event, agent-based, stochastic, spatial/PDE, reduced-order, and hybrid approaches; choosing numerical methods based on accuracy, stability, robustness, cost, and problem structure.

Why should anyone believe the result? Distinguishing and investigating: model-form error; parameter and input uncertainty; initial- and boundary-condition uncertainty; discretization error; iterative/solver error; sampling error; surrogate and model-reduction error; measurement error; software implementation error.

A central theme of the course is therefore **simulation credibility**.

3. Learning Outcomes

By the end of the course, students should be able to:

1. **Formulate a question-driven model** — identify the states, parameters, inputs, outputs, assumptions, scales, and system boundary required by a measurable quantity of interest.
2. **Select and justify a simulation paradigm** — compare continuous, discrete-event, stochastic, agent/network, spatial/PDE, reduced, and hybrid representations against a stated question and credible alternative.
3. **Implement reproducible simulation experiments** — translate a model into organized code, configurations, tests, figures, and instructions that another computational scientist can run.
4. **Analyze simulation behavior** — reason about numerical error, conditioning, stability, convergence, stochastic uncertainty, output analysis, sparsity, and computational cost in continuous, spatial, and event-driven settings.
5. **Design a credibility case** — distinguish and execute appropriate verification, validation, calibration, sensitivity, and uncertainty evidence; state the domain and limits of the resulting claim.
6. **Evaluate advanced modeling choices** — compare the role, assumptions, cost, and failure regimes of reduction, inference, data assimilation, scalable computing, scientific machine learning, and digital twins.
7. **Communicate and defend modeling decisions** — make evidence-supported claims and limitations clear in writing, code, figures, and oral discussion.

Outcomes 1–5 are the course’s **mastery outcomes**. Outcome 6 is a principled comparison and selection outcome rather than a claim of implementation mastery in every advanced method.

4. Prerequisites

Students should be comfortable with: undergraduate calculus (including basic multivariable), matrix/vector operations and elementary linear algebra, basic differential equations, basic probability, writing and debugging programs, manipulating arrays and numerical data, and making and interpreting plots.

An **ungraded diagnostic** covering prerequisite mathematics, probability, and programming opens on Canvas during the first week. It identifies material individual students should refresh (refresher notebooks are provided); it does not affect enrollment or grades.

5. Relationship to Other CSE Courses

CSE 6730 is an **integrative modeling-and-simulation course**, not a replacement for specialized courses. Georgia Tech offers deeper coursework in numerical linear algebra, numerical PDEs, high-performance computing, distributed simulation, uncertainty quantification, model reduction, and scientific machine learning. This course emphasizes **when and why a method is needed**,

what assumptions it introduces, how it affects a simulation, how to assess its correctness, and when more specialized methodology is required.

6. Materials, Computing Environment, and Course Tools

Required materials

There is **no required textbook purchase**. Required readings, notes, papers, code, and datasets are provided through Canvas or legally accessible online resources. Short **course-written reading notes** (20–30 minutes each) are assigned before selected lectures; each names its source trail for students who want a fuller treatment. Students must have access to a computer capable of running standard scientific-computing software.

Recommended reference: Michael T. Heath, *Scientific Computing: An Introductory Survey*. Supplemented by module-specific readings on conceptual modeling, discrete-event simulation, V&V, uncertainty, model reduction, and modern scientific computing. An **annotated book list**, organized by course module and noting which titles are freely available online and which the GT Library holds, is posted on Canvas; nothing on it is required.

Computing environment

Course examples use **Python**. A pinned course environment (`environment.yml`) is provided and installed in Lab 0; all labs are designed against it. A **course case-study code repository** (epidemic models, a nonlinear oscillator, a queueing network, and heat/advection–diffusion solvers) is used throughout lectures and labs — students work with the same four systems all semester rather than a new toy problem each week.

Students may use Julia, MATLAB, C/C++, or another appropriate language for **project** work with instructor approval, provided the final artifact meets the same reproducibility contract (tests + scripted figures).

Course tools

Tool	Used for
Canvas	announcements, grades, submissions of record, diagnostic
Ed Discussion	Q&A
Gradescope	lab autograding, regrade requests
Canvas quizzes	timed autograded quizzes (60 minutes, within a 3-day window)
GT GitHub	lab starter code, project repositories, CI
(github.gatech.edu/ModSim-CSE6730-Fall2026)	
PACE-ICE cluster	Week 14 compact hands-on segment; computationally expensive project runs
Zoom	virtual office hours (instructor Tue 2:00–3:00 PM)

All tool setup is completed in **Lab 0** during the first two weeks.

7. Course Structure and a Typical Week

The course combines conceptual/mathematical development, computational laboratories, recurring case studies, and a semester project (Sections 10–12).

A typical week

- **Before each lecture that has one:** a short course-written reading note (20–30 min). Four are assigned in Weeks 1–3; see the reading schedule on Canvas.
- **Monday lecture:** new material with live demos and prediction exercises.
- **Wednesday lecture:** new material continues; on quiz weeks the quiz window opens after class (Section 9).
- **Outside class:** active lab (about 3–4 hrs/week averaged across a two-week release window) and/or project work; office hours.

Expected total workload is approximately **9–11 hours/week** including class time.

8. Assessment and Grading

Component	Weight
Quizzes (6; lowest dropped)	25%
Computational laboratories (6; lowest dropped)	25%
Semester team project (M1–M5)	50%
Total	100%

These weights apply to CSE 6730 students; the paired undergraduate section's weights appear in its own syllabus (see Section 14 for the full requirements differentiation).

Grade scale: A 90–100 · B 80–89 · C 70–79 · D 60–69 · F below 60. Grades are based on demonstrated achievement of the learning outcomes, not a predetermined distribution. There is no routine extra credit. (Note for graduate students: most CSE degree programs do not accept a D toward degree requirements.)

Feedback: quizzes are autograded with results released on Canvas. At least 30% of the course grade will be returned before the withdrawal deadline (**Saturday, October 31, 2026**). The Canvas gradebook is configured with exactly these weights.

Regrades: submitted in writing via Gradescope within 7 days of grade release; the full item is regraded.

9. Quizzes — 25%

Six quizzes are given **on Canvas, outside class**. Each quiz opens on the date below and stays open for **three days**; once started, you have **60 minutes** to complete it, at any time within the window. Quizzes are **autograded** (multiple-choice, multiple-select, and numeric-entry questions drawn from randomized pools). They are open notes — including the course [quiz reference sheet](#) — but individual: no communication with others and no AI tools (Section 22). The lowest score is dropped, so each counted quiz is worth 5%.

Quiz	Window (3 days)	Coverage
1	Wed Sep 2 – Fri Sep 4	modeling and computational foundations
2	Wed Sep 23 – Fri Sep 25	ODE simulation and dynamical systems
3	Wed Oct 7 – Fri Oct 9	stochastic and discrete simulation
4	Wed Oct 28 – Fri Oct 30	output analysis, paradigms, PDE simulation
5	Wed Nov 11 – Fri Nov 13	model reduction, forward UQ, and sensitivity
6	Wed Dec 2 – Fri Dec 4	synthesis and credibility, incl. inverse problems and calibration

Quizzes emphasize reasoning over computation. Question formats are published in advance and recur: **predict-before-run** (given code/configuration, predict the outcome), **diagnose** (given a wrong result, identify the most likely error source), **choose-the-best-justification** (a scenario, a choice, and competing rationales — only one of which is right), and **judge-the-evidence** (select all statements a given verification/validation result actually supports). At least one question per quiz uses a case-study system from class. Each quiz includes one designated **[G] question** for CSE 6730 students (Section 14), typically a numeric-entry result derived from first principles. Open-form reasoning is assessed in the laboratory write-ups and milestone reports.

Quizzes are individual work. Communication with others and use of generative AI are not permitted (Section 22).

Makeups: the dropped quiz absorbs one ordinary missed window. Students with an **Institute-approved absence**, a documented emergency, a religious observance, or an election-day conflict spanning a quiz window are offered a **makeup quiz** — the drop is never required to cover an excused absence.

10. Computational Laboratories — 25%

Six laboratories connect theory to implementation. **Each lab is worth 5% and the lowest score is dropped** (the best five count, 25% total). Lab 0 is completion-graded.

Lab	Released	Due (11:59 PM)	Topic
0	Mon Aug 24	Fri Sep 4	Reproducible setup: environment, Git, tests, scripted figures
1	Mon Aug 31	Fri Sep 11	Numerical credibility: floating point, conditioning, residual vs. error
2	Mon Sep 14	Fri Sep 25	Dynamical simulation: convergence, stability, stiffness, verification
3	Wed Oct 7	Fri Oct 16	Stochastic/DES: replications, warm-up bias, output uncertainty
4	Mon Oct 19	Fri Oct 30	PDE simulation: manufactured solutions, refinement study, conservation
5	Wed Nov 4	Fri Nov 20	Track A: forward UQ + sensitivity, or Track B: parameter estimation

Every lab has two graded layers:

1. **Correctness (autograded):** implementation checks via provided tests, with instant feedback and unlimited resubmission before the deadline.
2. **Interpretation (rubric-graded):** a short `RESULTS.md` whose mandatory final section answers: **why should this result be trusted or distrusted?** This layer carries at least 40% of each lab's credit.

Each of Labs 1–5 designates one **[G] graduate investigation** required of CSE 6730 students (Section 14).

Labs 0–4 are individual; Lab 5 may be completed individually or in declared pairs.

11. Semester Project — 50%

The project is the intellectual center of the course. Teams of **3–5 students** address a modeling-and-simulation problem of their choosing, subject to instructor approval. A **curated topic menu** (with vetted questions and data sources) is published in Week 1; teams may choose from it or propose their own topic at the same approval bar.

A strong project begins with a question rather than a software package or algorithm.

Weak: “We will simulate traffic.” Stronger: “How does signal timing affect the probability of queue spillback under uncertain arrival rates?”

Weak: “We will implement an epidemic model.” Stronger: “Which uncertain behavioral parameters most strongly determine whether a proposed intervention prevents hospital-capacity exceedance?”

Scope scales with team size: a 5-person team must include either two graduate extensions (Section 13) or one extension plus a substantially deeper validation/UQ study; the expectation is set at topic approval.

Team composition across sections: teams may include students from the paired undergraduate section. Any team containing a CSE 6730 student is responsible for the graduate extension (Section 13); expectations for each member follow their own section's requirements. Because multiple teams may choose related topics, each team must register a distinct research question.

Required project elements

1. **Scientific or engineering question** — what the simulation is intended to answer.
2. **Conceptual model** — states; parameters; inputs; outputs/quantities of interest; spatial and temporal scales; interactions; assumptions; neglected phenomena.
3. **Computational formulation** — why the chosen model type and simulation approach are appropriate.
4. **Baseline simulator** — the simplest useful working implementation, before added complexity.
5. **Verification** — convincing evidence the implementation solves the claimed model, normally by **at least two** of: analytic/limiting-case comparison; grid or timestep convergence; conservation/invariant checks; independent implementation; manufactured solutions; regression/unit tests.
6. **Validation or credibility argument** — comparison with observations, experiments, published results, or a trusted reference; if direct validation is impossible, a precise account of what evidence would be required and what conclusions remain supportable.
7. **Computational experiment** — the simulation must answer a substantive question; trajectories and animations alone are insufficient.
8. **Sensitivity or uncertainty study** — how the principal conclusion depends on uncertain parameters, inputs, modeling choices, or numerical choices.
9. **Reproducibility** — a repository meeting the course contract: README, pinned environment, source, configurations and seeds, scripted figures, verification tests running in CI, and instructions that let another computational scientist reproduce the principal results.
10. **Limitations** — an explicit discussion of what the model does **not** justify.

12. Project Milestones and Timeline

Milestone	Date	Weight
Team formation (survey opens Aug 24)	Fri Sep 4	formative
Topic registration / approval	Fri Sep 11	formative
M1: Model specification and research question	Fri Sep 18	5%
M2: Literature and credibility plan (+ grad extension declared)	Fri Oct 2	5%
M3: Executable baseline simulator + verification evidence (+ peer evaluation 1)	Fri Oct 23	8%
M4: Credibility checkpoint (verification, validation, sensitivity, UQ, cost)	Fri Nov 13	10%

Milestone	Date	Weight
M5: Final report + reproducible repository + team video presentation (+ peer evaluation 2)	Mon Dec 7, 11:59 PM (final instructional day)	22%

The **team video presentation** is a 10–15 minute recording in which the team walks through the project — question, model, methods, and credibility case — uploaded to the Canvas **Media Gallery**. The report, repository, and video constitute the course’s **alternative final assessment** (Georgia Tech requires a final examination or an alternative final assessment in letter-grade courses; this section states that policy). There is no written final examination and the scheduled final-exam period is not used. Nothing is due, and no required meetings are held, during the Reading Periods.

Milestone specifications and rubrics are posted on Canvas. Highlights:

- **M1** — research question (falsifiable, with a measurable quantity of interest), conceptual diagram, state/parameter/input/output inventory, primary assumptions, proposed paradigm, planned Qols.
- **M2** — comparative engagement with ≥ 3 prior modeling works; validation data/reference identified **and confirmed accessible**; concrete verification and validation strategies; ranked anticipated uncertainties; graduate extension declared and justified.
- **M3** — simulator runs reproducibly from README instructions on a clean environment (graded by a TA attempting exactly that); CI running at least one meaningful verification test; preliminary results; identified risks. Teams are told in advance that reproduction is attempted cold.
- **M4** — a claim–evidence–gap table covering verification (two techniques), validation or a rigorous impossibility argument, first sensitivity and uncertainty passes, measured computational cost, and an honest answer to whether the original question is still answerable. **Teams may re-scope here; a well-argued pivot is rewarded, not penalized.**
- **M5** — final report (all ten elements; the limitations section is graded as seriously as the results), repository passing the reproducibility contract, the team video presentation, individual contribution statement, and AI Use Statement.

Peer evaluation: structured, confidential peer evaluations are collected at M3 (formative) and M5 (summative). Where sustained, corroborated evidence shows a material contribution imbalance, an individual’s **M3–M5 team-milestone score** may be multiplied by a factor from **0.50 to 1.0**. The adjustment is based on peer evidence, observable project artifacts, and a conversation with the affected student.

13. Graduate-Level Project Requirement

Each project must include **at least one substantial advanced extension** beyond the baseline simulation, for example: reduced-order modeling; multifidelity modeling; uncertainty quantification; Bayesian parameter inference; data assimilation; optimal experimental design; parallelization or GPU acceleration; advanced stochastic simulation; hybrid mechanistic/data-driven modeling; scientific machine learning or neural/operator surrogates.

The extension must be **scientifically justified** — not added because it is fashionable. It is declared at M2, feasibility-checked at M3, and evidenced at M4. Students must engage relevant primary literature and be prepared to explain or derive the essential mathematical/computational ideas underlying their extension.

14. Undergraduate/Graduate Requirements — Co-Taught Section Differentiation

This offering is co-taught with a paired undergraduate computer-simulation section (same room, same meeting times). Per College of Computing policy, requirements are **explicitly differentiated by level**. The table below is definitive for **CSE 6730 (graduate) students**; the paired section's own syllabus is definitive for its students. The same table is posted on Canvas.

Component	CSE 6730 (graduate) requirement	Paired undergraduate section
Quizzes	All questions, including one designated [G] question per quiz requiring derivation or justification from first principles	Same quizzes without the [G] question (section-specific Canvas quiz)
Labs 1–5	Core investigations plus one designated [G] graduate investigation per lab (e.g., <i>derive</i> the stability bound that undergraduates locate experimentally; <i>construct</i> the manufactured solution that undergraduates are given)	Core investigations only
Project	All ten required elements plus at least one graduate extension (Section 13) , with comparative engagement of primary research literature at M2	Ten elements at baseline scope; no extension requirement; curated readings in place of primary-literature review
Grading weights	Section 8 table	Defined in the paired section's syllabus

Mixed teams follow the Section 11 composition rule; within a mixed team, each member is assessed against their own section's requirements, and the graduate members own the extension.

In substance: graduate students must demonstrate greater mathematical depth (derivations, not only experiments), research-literature engagement, and a substantive advanced extension.

15. Project Evaluation Philosophy

The project is not graded primarily on model complexity. A sophisticated simulation that has not been verified, or whose results cannot be interpreted, may receive a substantially lower score than a simpler simulation that is carefully formulated, verified, validated, and analyzed.

Dimension	Emphasis
Problem formulation and modeling assumptions	20%
Numerical/computational correctness and verification	20%
Validation, uncertainty, and sensitivity	20%
Computational experiments and scientific interpretation	20%
Software quality, reproducibility, and communication	20%

16. Course Schedule

Organized around **modeling questions** rather than isolated algorithms. Quizzes, labs, and milestones appear in Sections 9–12; single-lecture weeks are scoped deliberately.

WkDates	Driving question / topics
1 Aug 24, 26	What is a computational model? Purposeful abstraction; states/parameters/inputs/outputs; model hierarchy; conceptual vs. mathematical vs. computational; verification vs. validation; credibility. In-class: competing models of one system.
2 Aug 31, Sep 2	Can the computer reliably evaluate our model? Floating point; conditioning; residual vs. error; cost and complexity; linear systems as recurring substructures (concepts only — depth returns in Week 9).
3 Sep 9 (Labor Day Mon)	How do mechanisms become dynamical models? State space; conservation and balance laws; constructing ODEs; mechanistic vs. empirical; parameter interpretation.
4 Sep 14, 16	When does a numerical trajectory represent the model? Euler and Runge–Kutta; local/global error; stability; stiffness; implicit vs. explicit; adaptive stepping; solver selection in practice.
5 Sep 21, 23	What qualitative behavior does the model predict? Equilibria; linearization; eigenvalues; phase portraits; bifurcations; chaos; separating physical from numerical instability.
6 Sep 28, 30	How do we simulate randomness and events? RNG and variates; Monte Carlo estimation and its error; discrete-event simulation: event lists, clocks, queues; worldviews.
7 Oct 7 (no class Mon Oct 5 — Fall Recess)	What can finite stochastic simulations tell us? Transient vs. steady state; warm-up bias; independent replications; output analysis; confidence intervals; variance vs. cost.
8 Oct 12, 14	When do individuals and connections matter — and how do local laws make fields? Agent-based and network models (paradigm selection, credibility pitfalls); conservation-based PDE construction: fluxes, constitutive laws, boundary and initial conditions.

WkDates	Driving question / topics
9 Oct 19, 21	When is a computed field trustworthy — and how do we compute things that flow? Finite differences for diffusion and the wave equation : consistency, stability, convergence and refinement studies, dispersion, conservation and energy checks; transport and conservation laws: CFL, numerical diffusion, the finite-volume viewpoint, shocks; shallow-water flood modeling (blood flow as the same hyperbolic structure).
10 Oct 26, 28	Expanding the PDE toolkit. Finite elements: weak forms, Galerkin, assembly, boundary conditions in practice; two-dimensional problems and meshes; applications: structures and tumor growth; sparse direct and iterative solvers for discretized systems.
11 Nov 2, 4	Can we replace an expensive model with a smaller one? SVD and low-rank structure; POD; projection-based reduced-order models; offline/online cost; when reduction fails.
12 Nov 9, 11	How uncertain is the prediction, and what can observations identify? Aleatory vs. epistemic uncertainty; Monte Carlo propagation; local and global sensitivity; inverse problems; parameter estimation; identifiability; regularization.
13 Nov 16, 18	How should observations update and justify a model? Bayesian inference; data assimilation; model-data discrepancy; calibration vs. validation; credibility synthesis; experimental design and multifidelity outlook.
14 Nov 23 (Student Recess Wed)	What changes at scale? Cost, memory, and data movement; parallelism patterns; strong/weak scaling; GPUs; compact hands-on cluster segment (PACE-ICE).
15 Nov 30, Dec 2	Where do learned models belong — and does the credibility case hold? Surrogates; physics-informed and operator-learning concepts; digital twins as the course pipeline run continuously; structured credibility audit of students' own projects; course synthesis.
16 Dec 7	Project studio and peer review. Reproducibility smoke tests, final clinic. Final report + repository + team video due 11:59 PM.
— Dec 9–10	<i>Reading Periods — nothing due, no required meetings.</i>
— Fri Dec 11	<i>Scheduled final-exam period — not used (the alternative final assessment is the M5 report, repository, and video, due Dec 7).</i>

The instructor may adjust topic sequencing with notice (Section 25); assessment dates change only per Section 25.

17. Georgia Tech CSE Research Connections

The methods of this course are the working tools of [research in the School of CSE](#), especially its [Scientific Computing, Simulation, and Computational Math](#) and [High Performance Computing](#) areas: scientific machine learning, uncertainty quantification, Bayesian inference, data assimilation, optimal experimental design, and digital twins; model reduction and multifidelity methods; computational plasma physics and scalable solvers; cardiac electrophysiology; seismic imaging

and inversion; iterative solvers and preconditioning; AI-enabled PDE-constrained design; and performance engineering for simulation.

About five short **CSE Research Capsules** (10–15 minutes each) during the semester connect lecture ideas to this research, and many project topics and graduate extensions are drawn from it. Students interested in research careers or theses should treat the capsules as invitations: the modeling → simulation → credibility pipeline taught here is the entry point to every area above.

18. Reproducibility Standard

Every major computational result must have an identifiable path from code and parameters to reported output; every important project figure or table must be associated with a workflow that regenerates it. Students are expected to: track code with version control; specify dependencies; document parameters; manage random seeds; preserve raw outputs where appropriate; distinguish exploratory code from final experiments; test important components; and document how final figures and claims were produced. **Lab 0 establishes this workflow; the M3 and M5 “cold reproduction” checks grade it.**

19. Attendance and Participation

Regular attendance is strongly encouraged: class sessions include modeling exercises, live code experiments, prediction questions, failure demonstrations, and project feedback not fully reproduced in posted materials. Routine attendance is not separately graded. Attendance **is required** for announced project presentations and team review sessions where a student’s participation is evaluated, except as covered by Institute policy or approved accommodations. Communicate foreseeable conflicts early.

20. Late Work and Missed Assessments

- **Quizzes:** the dropped quiz absorbs one ordinary missed window; excused absences receive a makeup quiz (Section 9).
 - **Lab 0:** completion is due Sep 4 because it gates later work. Students who add after the first meeting or report a documented setup blocker by Sep 4 receive a no-penalty completion plan of up to seven additional days; Lab 0 must nevertheless be complete before Lab 1 is submitted.
 - **Labs 1–5:** each student has **4 free late days** across Labs 1–5, at most **2 per lab**, applied automatically in whole-day increments. Beyond free late days: 10% deduction per day, up to 3 days, then zero.
 - **Project milestones:** due on time — they affect teammates and feedback schedules. Teams facing significant circumstances should contact the instructional team **before** the deadline.
 - **Final artifacts (M5 report, repository, video):** due Mon Dec 7, 11:59 PM; extensions only through the Institute processes governing final assessments.
-

21. Collaboration Policy

- **Quizzes:** individual.
 - **Labs:** students may discuss concepts and debugging strategies; submitted code, figures, analysis, and explanations must be the student's own (or the declared pair's, for Lab 5). Substantive discussions must be acknowledged in the submission.
 - **Project:** collaboration within the team is expected; every student must make a clearly identifiable intellectual and technical contribution and be able to explain and defend their own work. Git history alone is not sufficient evidence of contribution.
-

22. Generative AI Policy

Generative AI can be useful in computational research, but it can also replace precisely the reasoning this course develops. Acceptable use depends on the assessment. (Georgia Tech/USG requires instructors to establish and communicate a course AI policy.)

- **Quizzes:** generative AI may **not** be used. Quizzes are open notes but individual: no other people, no AI tools. This is an honor-code obligation.
- **Labs:** AI tools may be used for limited productivity assistance (syntax help, bug identification, refactoring suggestions, interpreting error messages, documentation). Students remain responsible for the mathematical reasoning, algorithm selection, experiment design, interpretation, and all submitted work. Material AI contributions must be disclosed on the provided form.
- **Project:** AI tools may be used for brainstorming, debugging, refactoring, documentation, writing assistance, and API exploration. The following must remain the team's intellectual work: the central question, conceptual model, major assumptions, numerical-method choices, verification and validation strategies, computational experiments, interpretation, and conclusions.

Two standing rules. First, every project team submits an **AI Use Statement** (provided form) identifying tools used, what for, which files/sections were materially affected, and how outputs were checked. Second, **AI-generated code is subject to the same verification standard as any other code:** it must be covered by your tests, and every team member must be able to explain any line of it. (Week 15 includes an in-class exercise auditing an AI-generated simulation for planted credibility flaws.)

Students are responsible for all submitted mathematics, code, citations, figures, and claims, regardless of origin. An honest, detailed AI Use Statement never lowers a grade; an inaccurate one is an academic-integrity matter.

23. Academic Integrity

Students are expected to uphold the Georgia Tech Academic Honor Code. Unauthorized collaboration, plagiarism, fabrication or falsification of computational results, misrepresentation of individual contributions, prohibited assistance on assessments, or presenting another person's or system's work as one's own may constitute misconduct. In computational work, integrity also

includes honest reporting of failed experiments, parameter choices, numerical uncertainty, simulation conditions, data preprocessing, AI assistance, and limitations. Suspected violations may be referred through Georgia Tech's academic-integrity process.

24. Student Conduct and Student–Faculty Expectations

Students and instructional staff share responsibility for a professional, respectful, intellectually engaged environment. Students are expected to arrive prepared, engage constructively, communicate professionally, contribute equitably to team activities, meet commitments to teammates, and raise concerns early. Students can expect clearly communicated policies, accessible office hours, timely grading information, and reasonable notice of changes, consistent with Georgia Tech's Student–Faculty Expectations Agreement.

25. Accommodations, Support, and Syllabus Changes

Accommodations: Georgia Tech is committed to equitable access. Students who experience disability-related barriers should contact the Office of Disability Services; students with approved accommodations should communicate with the instructor as early as practical.

Student support: students experiencing personal, academic, or wellbeing difficulties are encouraged to use Institute resources, including the Office of the Dean of Students (Students' Temporary Assistance and Resources), the Center for Mental Health Care and Resources, and campus academic support services. Contacting the instructor early when circumstances affect coursework is always appropriate. Concerns about academic policies or decisions may also be raised through Georgia Tech's **Academic Grievance Policy** (see the Catalog); students are always welcome to raise concerns with the instructor first.

Incomplete grades: an Incomplete (I) is considered only for a significant documented non-academic circumstance arising after a student has completed a substantial portion of the course with satisfactory performance, per Registrar policy; otherwise, students should consult the instructor and their advisor about support options or withdrawal while the W deadline permits.

Syllabus changes: the instructor may make reasonable changes to readings, topic sequencing, laboratory details, or project logistics. Changes affecting grading, major deadlines, or assessment formats will be communicated clearly and with reasonable notice; any such change after the first week of classes will be made **before the withdrawal deadline (October 31, 2026)**, consistent with Institute policy and the Student–Faculty Expectations framework. The current syllabus is also posted to the public GT syllabus system (syllabus.gatech.edu) and updated there when it changes.

26. Criteria for Successful Completion — and the Central Principle

Successful students do substantially more than execute working code. They demonstrate that they can **formulate** → **implement** → **verify** → **experiment** → **validate** → **quantify uncertainty** → **interpret** → **communicate**, and their project answers four questions convincingly:

What did you model? Why did you model it that way? Why should we believe the computation? What does the evidence actually allow us to conclude?

A simulation is not credible because it produces a plausible-looking result. A computational prediction becomes useful when its assumptions are understood, its implementation is verified, its relationship to trusted evidence is examined, its uncertainty is characterized, and its conclusions are stated at a level justified by that evidence.

The goal of CSE 6730 is not merely to teach students how to run simulations. It is to teach them how to construct, interrogate, and defend computational models.