MMAE 450: Computational Mechanics II

Spring 2026

Department of Mechanical, Materials, and Aerospace Engineering
Illinois Institute of Technology

Instructor Information

- **Instructor:** Dr. Michael Gosz
- **Office:** 207A Rettaliata Engineering Center
- **Phone:** (312) 567-3198
- **Email:** `gosz@illinoistech.edu`
- **Office Hours:** TBA
- **Class Meetings:** Tue, Thu (1:50pm–3:05pm), Stuart Building 238

Course Information

Course:	MMAE 450
Title:	Computational Mechanics II
Credits:	3 credit hours
Prerequisites:	MMAE 350

Course Description

This course builds on MMAE 350 and develops advanced computational methods for modeling engineering systems governed by partial differential equations. Emphasis is placed on translating physical assumptions into governing equations, discretizing those equations using finite difference, finite volume, and finite element methods, and interpreting numerical results in the context of what the model predicts and what physics permits.

This course emphasizes engineering modeling and numerical reasoning rather than programming for its own sake. Computational tools are used to support physical insight, model verification, and engineering design decisions.

A central theme of the course is the integration of physics-based modeling with data-driven approaches. Students are introduced to regression, classification, and physics-informed neural networks (PINNs) as tools for prediction and surrogate modeling that complement and extend traditional numerical techniques. Students gain experience implementing algorithms in Python using Jupyter notebooks, validating numerical results, and communicating computational findings clearly and professionally in an engineering context.

Learning Objectives

By the end of this course, students will be able to:

1. Formulate governing equations for heat transfer, wave propagation, and transport phenomena in engineering systems.
2. Discretize partial differential equations using finite difference, finite volume, and finite element methods.
3. Implement and analyze numerical solvers for linear and nonlinear systems arising from discretized models.
4. Evaluate numerical accuracy, stability, conditioning, and convergence of computational methods.
5. Apply weak formulations and finite element approximations to one- and two-dimensional problems.
6. Develop regression and classification models for engineering datasets and interpret model parameters.
7. Construct and train physics-informed neural networks (PINNs) for boundary value problems.
8. Compare physics-based, data-driven, and hybrid modeling approaches, identifying advantages and limitations.
9. Implement reproducible computational workflows using Python, Jupyter notebooks, and version-controlled resources.
10. Deploy computational and machine learning models using modern cloud-based tools.

Weekly Topics

The course is organized around a progression from governing equations, to numerical discretization, to modern data-driven and hybrid modeling approaches. The following outline indicates the primary topics covered during the semester; exact pacing may be adjusted as needed.

1. Review of nonlinear equations, matrix systems, and classification of partial differential equations (elliptic, parabolic, hyperbolic).
2. Time-dependent partial differential equations: formulation and physical interpretation.
3. Finite difference methods for transient heat conduction in one and two dimensions.
4. Wave propagation problems in solids: numerical discretization and stability considerations.
5. Stability, conditioning, and convergence analysis of numerical schemes (including von Neumann analysis).
6. Conservation laws and flux formulations: introduction to the finite volume method.
7. Finite volume methods for advection–diffusion problems, including numerical diffusion and upwinding.

8. Weak formulations and variational principles for boundary value problems.
9. Finite element methods for heat conduction: shape functions, assembly, and solution of global systems.
10. Finite element modeling of transport and coupled problems in one and two dimensions.
11. Introduction to data-driven modeling in engineering: regression and classification methods.
12. Physics-informed machine learning: physics-informed neural networks (PINNs) for boundary value problems.
13. Surrogate modeling and hybrid approaches: combining numerical simulation with machine learning.
14. Reproducible and cloud-based computational workflows for engineering analysis and model deployment.
15. Integrated computational project development and presentations.

Assignments & Assessment

Student performance in this course will be evaluated using the following components:

- **Homework Assignments (35%)**
  Regular problem sets emphasizing advanced numerical methods, implementation of algorithms in Jupyter notebooks, and interpretation of computational results for engineering models governed by partial differential equations.
- **Midterm Exams (30% total)**
  Two in-class midterm exams, spaced throughout the semester, assessing conceptual understanding, mathematical formulation, and analysis of numerical methods and computational models covered up to each exam.
- **Final Exam (20%)**
  A comprehensive exam evaluating students' ability to integrate numerical methods, physical modeling, and computational reasoning across the full scope of the course.
- **Final Project (15%)**
  A focused computational project in which students develop and analyze a computational mechanics model. Projects may emphasize physics-based simulation, data-driven modeling, or hybrid approaches that integrate numerical methods with machine learning. Results will be documented in a written report, with selected projects presented orally.

Homework assignments consist primarily of weekly coding exercises in Jupyter notebooks. Midterm exams include closed-book theoretical components and open-book computational components. The final project is a team-based effort and may focus on either a mechanics-based numerical model or a machine learning–based surrogate model.

Tools & Resources

- **Primary Computational Environment:**
  Python is the primary programming language used in this course. Students will work extensively with Jupyter notebooks and standard scientific libraries, including NumPy, SciPy, SymPy, Matplotlib, and scikit-learn. Selected examples will also use PyTorch for physics-informed machine learning applications.

- **Software Setup:**
  All students are expected to work in a local Python virtual environment, following the same setup procedure introduced in MMAE 350. This ensures reproducibility and consistency across computational assignments and projects.

- **Textbook (Primary Reference):**
  Computational Mechanics: A Modern Introduction with Machine Learning and AWS Workflows
  by M. Gosz. This text serves as the primary reference for the course and provides the theoretical foundations, numerical methods, and computational examples used throughout the semester.

- **Companion Code Repository:**
  A public GitHub repository accompanies the textbook and contains Jupyter notebooks, example codes, datasets, and a `requirements.txt` file. All computational assignments in the course build directly on these materials. The repository is available at:

  `https://github.com/gosz450/computational-mechanics-companion`

- **Cloud-Based Workflows:**
  Selected modules and the final project may involve cloud-based execution and deployment of computational or machine learning models. These activities will build on local workflows and will be introduced incrementally during the semester.

- **Supplementary References (Optional):**
  - Chapra & Canale, *Numerical Methods for Engineers*
  - Ferziger & Perić, *Computational Fluid Dynamics*
  - Géron, *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*

Course Calendar — Spring 2026

Note: The course calendar is subject to minor adjustments based on class progress.

Date	Day	Topic
Jan 13	Tue	Course introduction and computatonal workflow
Jan 15	Thu	Local polynomial approximation and Taylor series
Jan 20	Tue	Newton's method for nonlinear systems

Date	Day	Topic
Jan 22	Thu	Finite difference methods for 1D transient heat conductions
Jan 27	Tue	Finite difference methods for 1D transient heat conduction (Boundary conditions and Crank–Nicolson)
Jan 29	Thu	Finite difference methods for 2D heat conduction and wave propagation
Feb 3	Tue	von Neuman Stability for 1D Heat Conduction (FTCS Method)
Feb 5	Thu	Numerical wave propagation problems; stability considerations
Feb 10	Tue	Divergence Theorem and Material Time Derivative (Lagrangian and Eulerian Viewpoints)
Feb 12	Thu	Conservation Laws (Continuity and Momentum Equations)
Feb 17	Tue	Review and problem session
Feb 19	Thu	**Midterm Exam 1**
Feb 24	Tue	Conservation laws (energy equation/advection-diffusion)
Feb 26	Thu	Finite volume methods for advection-diffusion problems
Mar 3	Tue	Numerical diffusion, upwinding, and stability in finite volume methods
Mar 5	Thu	Weak formulations and variational principles for boundary value problems
Mar 10	Tue	Finite element method for heat conduction: shape functions and assembly
Mar 12	Thu	Finite element method for 2D heat conduction: shape functions and assembly
Mar 17	Tue	**Spring Break**
Mar 19	Thu	**Spring Break**
Mar 24	Tue	**Review for midterm exam 2**
Mar 26	Thu	**Midterm Exam 2**
Mar 31	Tue	Regression models for mechanics data; linear and multiple regression
Apr 2	Thu	Regression models for mechanics data; logistic regression
Apr 7	Tue	Physics-informed machine learning: PINNs for boundary value problems
Apr 9	Thu	Physics-informed machine learning: PINNs for boundary value problems
Apr 14	Tue	Workshop: project/problem selection and scope definition
Apr 16	Thu	Cloud-based workflows for computational mechanics and ML models
Apr 21	Tue	Cloud-based workflows for computational mechanics and ML models
Apr 23	Thu	Review and integration: numerical methods, FEM, and ML
Apr 28	Tue	Project/Problem Presentations

Date	Day	Topic
Apr 30	Thu	Project/Problem Presentations
May 5	Tue	Project/Problem presentations
May 7	Thu	Project/Problem presentations (if needed)

Course Policies

Attendance

Regular attendance is expected. Students are responsible for all material covered in class and all announcements made, regardless of attendance. Excessive unexcused absences may negatively impact your performance and grade.

Late Work

Assignments are due on the date and time specified. Late work will generally not be accepted unless prior arrangements are made with the instructor, or unless extraordinary circumstances can be documented. In such cases, partial credit may be given at the discretion of the instructor.

Academic Integrity

Illinois Tech expects all students to uphold the highest standards of academic honesty. Cheating, plagiarism, or any other form of academic dishonesty will not be tolerated. Violations will be reported and may result in failure of the assignment, failure of the course, and/or additional disciplinary action as outlined in the Illinois Tech Code of Conduct. For more information, see: `https://www.iit.edu/student-affairs/student-handbook`

Collaboration

Collaboration on homework assignments is permitted at the level of discussing concepts and approaches. However, all work turned in must be your own. Copying code, solutions, or written responses from another student or from online sources constitutes academic dishonesty.

Use of Technology

Students are encouraged to use Python, and other computational tools as required by the course. Any use of technology during quizzes or exams must be explicitly permitted by the instructor. Unauthorized use of technology during exams will be considered a violation of academic integrity.

Communication

Course announcements will be made in class and via email or the course LMS (Canvas). It is your responsibility to check email and the course site regularly. Email is the preferred method of communication outside of class hours.

Professional Conduct

Respectful behavior is expected in class, in labs, and in all course-related activities. Disruptive conduct will not be tolerated. Students should contribute to a learning environment that supports diversity of thought and experience.

Accessibility Statement

Illinois Tech is committed to providing an inclusive educational environment and making every effort to ensure equal access for all students. If you are a student with a documented disability and require reasonable academic accommodations, please contact the Center for Disability Resources (CDR) as soon as possible. Accommodations are determined on a case-by-case basis, considering the student's documented needs and the technical requirements of the course.

To request accommodations or learn more, contact:

- **Center for Disability Resources (CDR)**
  Phone: 312-567-5744
  Email: `disabilities@illinoistech.edu`

Students must submit documentation and meet with CDR to establish eligibility and receive an accommodation letter. Provide your letter to the instructor early in the semester to discuss appropriate implementation of accommodations.

For more information, visit: `https://www.iit.edu/cdr`

University Academic Calendar — Spring 2026

Date	Event
January 12	Spring Courses Begin
January 19	Martin Luther King, Jr. Day — No Classes
January 20	Last Day to Add/Drop for Full Semester Courses with No Tuition Charges
January 27	Last Day to Request Late Registration
March 13	Midterm Grades Due
March 16–21	Spring Break Week — No Classes
March 30	Last Day to Withdraw for Full Semester Courses
April 6	Fall Registration Begins
May 2	Last Day of Spring Courses
May 3	Last Day to Request an Incomplete Grade
May 4–9	Final Exam Week (Final Grading Begins May 4)
May 13	Final Grades Due at Noon (CST)