

MMAE 450 — Computational Mechanics II

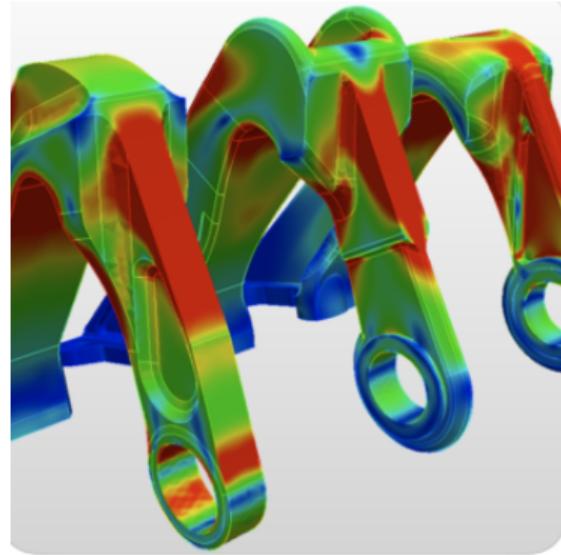
Day 1: Course Introduction & Computational Workflow

Mike Gosz

Spring 2026

Why Computational Mechanics?

- ▶ Many engineering problems have **no closed-form solution**
- ▶ Computation bridges **theory** and **real systems**
- ▶ Central in:
 - ▶ heat transfer
 - ▶ solid mechanics and dynamics
 - ▶ transport phenomena

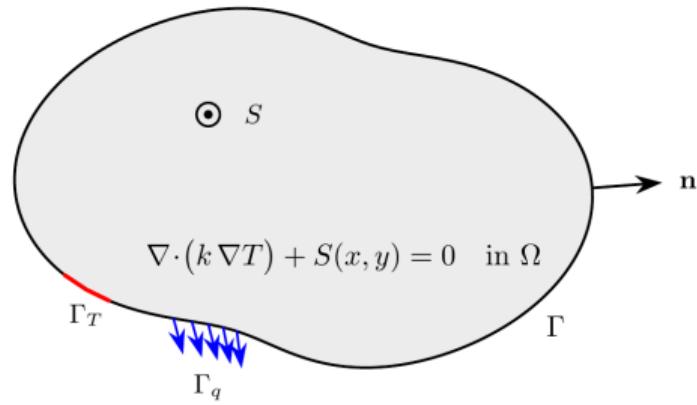


Stress distribution in mechanical part with complex geometry.

What Is Computational Mechanics?

1. Physical system (assumptions + model)
2. Governing equations
3. Numerical approximation (discretization)
4. Computational solution
5. Interpretation and validation

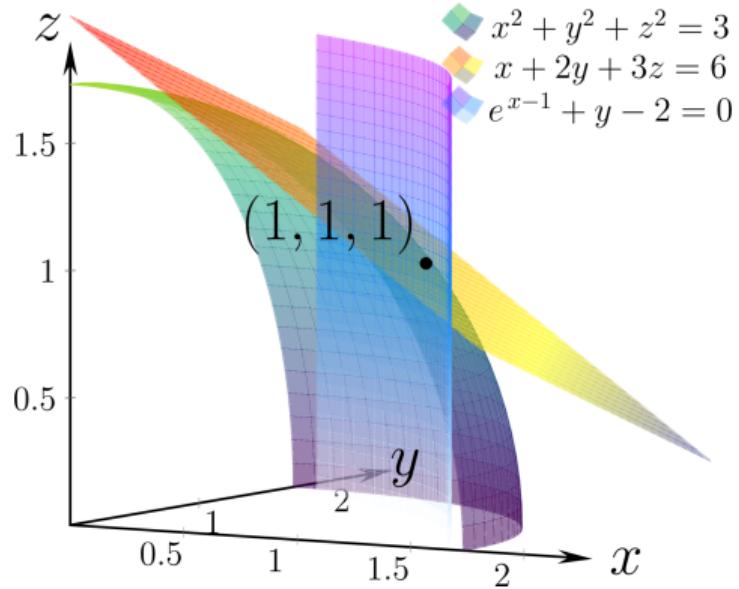
Key message: Computational mechanics is about **models**, not just numbers.



Complex geometry \Rightarrow discretization \Rightarrow computation

Where This Course Fits

- ▶ Builds on:
 - ▶ calculus
 - ▶ differential equations
 - ▶ linear algebra
- ▶ Develops:
 - ▶ numerical thinking
 - ▶ computational implementation
 - ▶ interpretation of results
- ▶ Prepares you for:
 - ▶ research and modeling
 - ▶ engineering simulation tools
 - ▶ machine learning models and cloud workflows
 - ▶ data analytics



Intersection of nonlinear surfaces representing a system of equations.

What This Course Emphasizes

- ▶ Modeling, not memorization
- ▶ Algorithms, not button-pushing
- ▶ Understanding:
 - ▶ accuracy
 - ▶ stability
 - ▶ limitations and failure modes

We will care why methods work—and when they fail.

Tools We Will Use

- ▶ **Python** as the computational language
- ▶ **Jupyter notebooks** for code + math + narrative
- ▶ Core libraries:
 - ▶ NumPy (arrays, linear algebra)
 - ▶ SymPy (symbolic math, derivations)
 - ▶ Matplotlib (plots and visualization)
 - ▶ SciPy (numerical solvers, eigenproblems)
 - ▶ Pandas (data handling and post-processing)
 - ▶ scikit-learn (regression and surrogate models)
 - ▶ PyTorch (physics-informed neural networks)
 - ▶ Cloud workflows for scalable computation

We will briefly review Python and notebooks to ensure everyone is aligned.

The screenshot shows a Jupyter Notebook interface with several code cells and their outputs:

- Example 1.1--Python Warm UP**
 - Code cell 1: A series of imports including numpy, pandas, matplotlib.pyplot, and sympy.
 - Code cell 2: A plot of the sine function from 0 to 6. The plot is titled "Plot of the sine function".
 - Code cell 3: Matrix operations involving matrices A and B, and a solution vector x.
 - Code cell 4: A for loop that prints square values from 1 to 5.
 - Code cell 5: A function definition for area_rectangle followed by its call with arguments 3 and 4.
 - Code cell 6: Data reading and writing operations using pandas.
- Create a simple NumPy array**
 - Code cell 1: A single-line command to create a 1D NumPy array of 1000 elements ranging from 0 to 1.
- Create and manipulate a matrix**
 - Code cell 1: Matrix multiplication of A and B.
 - Code cell 2: Solution vector x.
- The for loop**
 - Code cell 1: A for loop that prints square values from 1 to 5.
- Define and call a function**
 - Code cell 1: Function definition for area_rectangle.
 - Code cell 2: Call to the function with arguments 3 and 4.
- Read and write data files**
 - Code cell 1: Data reading and writing operations using pandas.

Why Jupyter Notebooks?

- ▶ Combine: formatted text, equations, executable code, and plots
- ▶ Ideal for:
 - ▶ exploration
 - ▶ verification of derivations
 - ▶ clear communication of results

How the Course Is Structured

- ▶ Weekly modules on Canvas
- ▶ Mix of:
 - ▶ lectures
 - ▶ guided notebooks
 - ▶ homework assignments
- ▶ Two midterms + final exam or final project

The screenshot shows the Illinois Tech Canvas course page for Spring 2026 Computational Mechanics (MMAE-350-01). The course ID is 21667.202602. The left sidebar includes links for Home, Syllabus, Modules, Discussions, Assignments, Canvas, Courses, Zoom, Precept Videos, People, Grades, Grade Library, and Course Evaluation. The main content area displays course details and several course modules.

A consistent weekly rhythm: learn → practice → assess

What You Should Do This Week

- ▶ Review the syllabus (PDF)
- ▶ Complete Module 0 setup (Python + environment)
- ▶ Run Notebook 01 and make small edits
- ▶ Start Homework 1 (reflection + small computations)