UNIVERSITY AT ALBANY
STATE UNIVERSITY OF NEW YORK

CSI 401 (Fall 2025)

Numerical Methods

Lecture 1: Course Introduction & Numerical Errors

Chong Liu

Department of Computer Science

Aug 25, 2025

About myself

- Education:
  - PhD in Computer Science, UC Santa Barbara, 2018-2023
  - Data Science Institute Postdoc, University of Chicago, 2023-2024
- Research areas:
  - Machine Learning: Bayesian optimization, bandits, generative models
  - AI for Drug Discovery: experimental design, drug screening, binding affinity prediction
- Past courses:
  - CSI 436/536 Machine Learning (Fall 2024, Spring 2025)
- Contact:
  - Homepage: https://chong-l.github.io/
  - Email: cliu24@albany.edu

Meet your TA!

- Charles DeGennaro
  - CS PhD student at UAlbany
  - cdegennaro@albany.edu

Agenda

- Course Information

- Self-evaluation
  - Don’t worry – it doesn’t count towards your final grades!
  - For me to get to know your math backgrounds

- Why numerical methods?

- First technical part: Numerical errors

Course information

- Class webpages:
  - Syllabus: https://chong-l.github.io/CSI_401_25F.html
    - Your primary information source of this course, updated regularly
    - Lecture slides, deadlines
  - Brightspace
    - Posting grades, discussion
  - Gradescope
    - Homework assignments/submissions, course projects/report submissions
    - https://www.gradescope.com/courses/1093542
    - Use **7XJ4N5** to enroll ASAP using your albany.edu email
- Office hours (starting next week):
  - Instructor: Monday 3-4pm at UAB 426
  - TA: Wednesday 1:30-2:30pm at UAB 412E

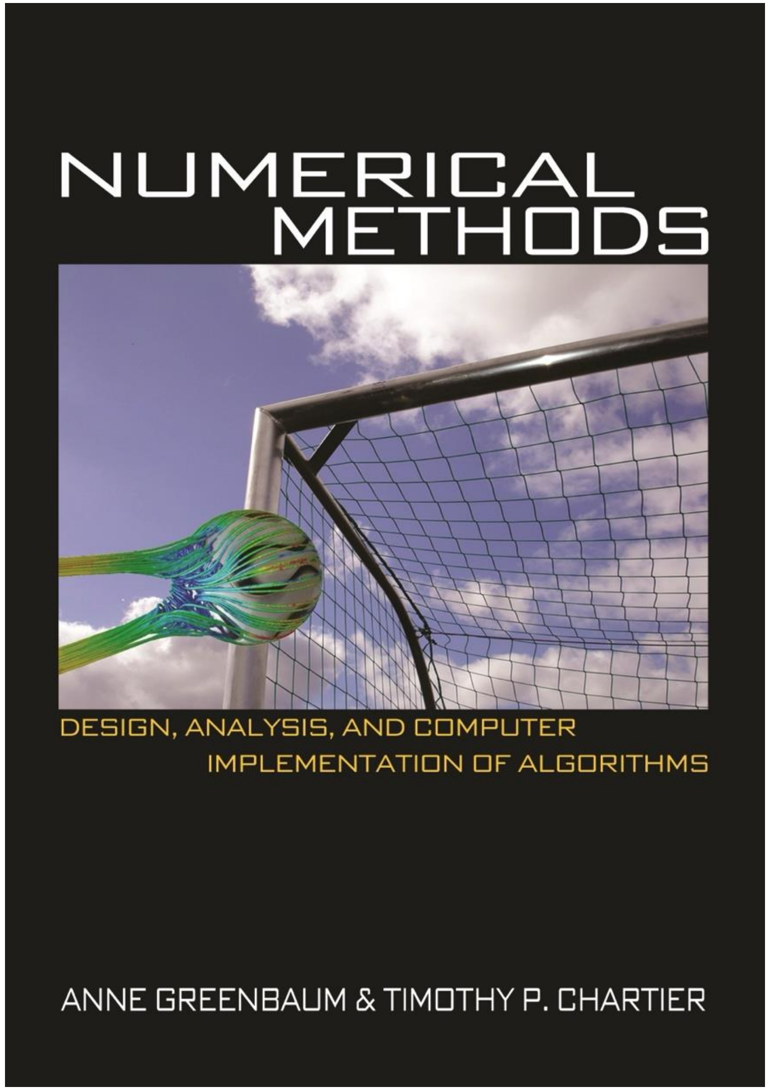
Course information - Requirements

- Data structure, linear algebra
- Programming:
  - Python
    - A very nice tutorial: https://colab.research.google.com/github/cs231n/cs231n.github.io/blob/master/python-colab.ipynb
  - Matlab
    - Official tutorials: https://www.mathworks.com/support/learn-with-matlab-tutorials.html
- Document editing:
  - LaTeX
    - A Tutorial by Overleaf: https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes

- We will review Python/Matlab/LaTeX on Sep 8 (Week 3)!

Course information

- Reference book
  - Numerical Methods: Design, Analysis, and Computer Implementation of Algorithms
    - Anne Greenbaum and Timothy P. Chartier.
    - Princeton University Press, 2012.
  - Just for your reference. Lecture slides are your best source!



Course information

- Scale
  - A: 95-100 points
  - A-: 90-94 points
  - B+: 85-89 points
  - B: 80-84 points
  - B-: 75-79 points
  - C+: 70-74 points
  - C: 65-69 points
  - C-: 60-64 points
  - D+: 55-59 points
  - D: 50-54 points
  - E: 0-50 points
- Grading:
  - Homework: 24%
  - Course project: 21%
  - Midterm exam: 20%
  - Final exam: 30%
  - Participation: 5%

  - I reserve the right to curve up the points.

Course information

- Study group
  - All homework assignments and course project are completed **in groups.**
  - A group consists of **3-5** students.
  - All students in the same group receive the same credits.

Course information

- Group homework (24%)
  - 4 homework assignments, each 6%
  - No handwritten homework: LaTeX -> PDF submissions
  - Due at 11:59 pm (midnight) in Eastern Time on due dates

  - Late homework **within** 24 hr period receives **half** credits
  - Late homework **beyond** 24 hr period receives **0** credits

Course information

- Group course project (21%)
  - Each group chooses to work on one project from project list
    - Group may work on a project beyond the list, subject to my approval.
  - Outcomes:
    - Midterm presentation practice (0%)
    - Midterm project one-pager (2%)
    - Final presentation (15%)
    - Final project report (4%)
    - Submit project code (0%)
      - **Lose all 21 credits** if your code is copied from somewhere or doesn't work!
  - Due at 11:59 pm Eastern Time on the due date, no late submission

Group course project list

- Has been released on Gradescope!

- Data Compression via Singular Value Decomposition (SVD)
- Data Fitting Kaggle Competition
- Nonlinear System Solvers
- Function Optimization
- PDEs in Engineering Applications

Course information

- Exams (50%)
  - Midterm exam (20%) – all topics before midterm exam
  - Final exam (30%) – all topics throughout this semester
  - Given **individually**
  - Tip: Try to understand all solutions to your homework!

Course information

- Participation (5%)
  - How to earn?
    - Starting Week 4, ask questions in class or voluntarily show/explain your solutions to in-class exercise problems.
    - Register your name to me after each class meeting.
    - Up to 3 points can be given to each student.
  - 2 points are reserved for all students if the percent of submitted course evaluation goes above 60%.

Course information

- A few remarks:
  - Some topics might be **very technical**, but the lectures will be self-contained.
  - Attending the lectures is required as we have many helpful in-class exercise questions that we will work together!
  - Do homework on time. Never hesitate to answer questions!

Why learn Numerical Methods?

- Motivated by
  - most real-world problems in science, engineering, and data science **cannot** be solved exactly using **closed-form solutions**.
  - For example,
    - Many equations, such as nonlinear systems, high-dimensional integrals, and differential equations, either lack analytical solutions or are too complex to solve by hand.
- Advantages:
  - Providing systematic algorithms to approximate these solutions with controllable accuracy and efficiency
  - Bridging the gap between mathematical theory and computer implementation
  - Enabling the simulation and prediction of complex problems—such as climate modeling, structural design, or machine learning
  - Ensuring stability, convergence, and error control!

Topics of Numerical Methods covered

- Source of numerical errors
- Asymptotic notations and floating point arithmetic
- Review:
  - Linear algebra, Python, Matlab, LaTeX
- Linear systems:
  - Direct linear equations solvers
  - Eigenvalues and eigenvectors
  - Iterative linear equations solvers
  - Conditioning of linear equations
- Numerical interpolation
  - Data fitting and regression problems
- Nonlinear equations solvers
- Optimization methods
- Numerical integration and differentiation

Expected outcomes

- Understanding the foundation, major techniques, applications, and challenges of numerical methods
- The ability to apply numerical methods for solving real-world problems
- Familiarize the tools for more in-depth computer science / data science / engineering studies

- You will **not** be:
  - An expert in numerical methods **- yet**
  - Knowing all the subareas of numerical methods **- yet**

Self-evaluation (0% towards grades)

- Q1. Given $A = \begin{bmatrix} 2 & 7 & 3 \\ 1 & 0 & 9 \\ -1 & 2 & 10 \end{bmatrix}, B = \begin{bmatrix} -2 & 0 & 3 \\ 2 & -1 & 7 \\ 6 & 4 & -3 \end{bmatrix}$. Is $AB = BA$?

- Q2. Given the function $f(x, y) = e^{x+y} + e^{3xy} + e^{y^4}$, find the partial derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$.

Solutions to self-evaluation

- A1.
- $AB \neq BA$ since $(AB)_{11} = 2 * (-2) + 7 * 2 + 3 * 6 = 28, (BA)_{11} = (-2) * 2 + 0 + 3 * (-1) = -7$.

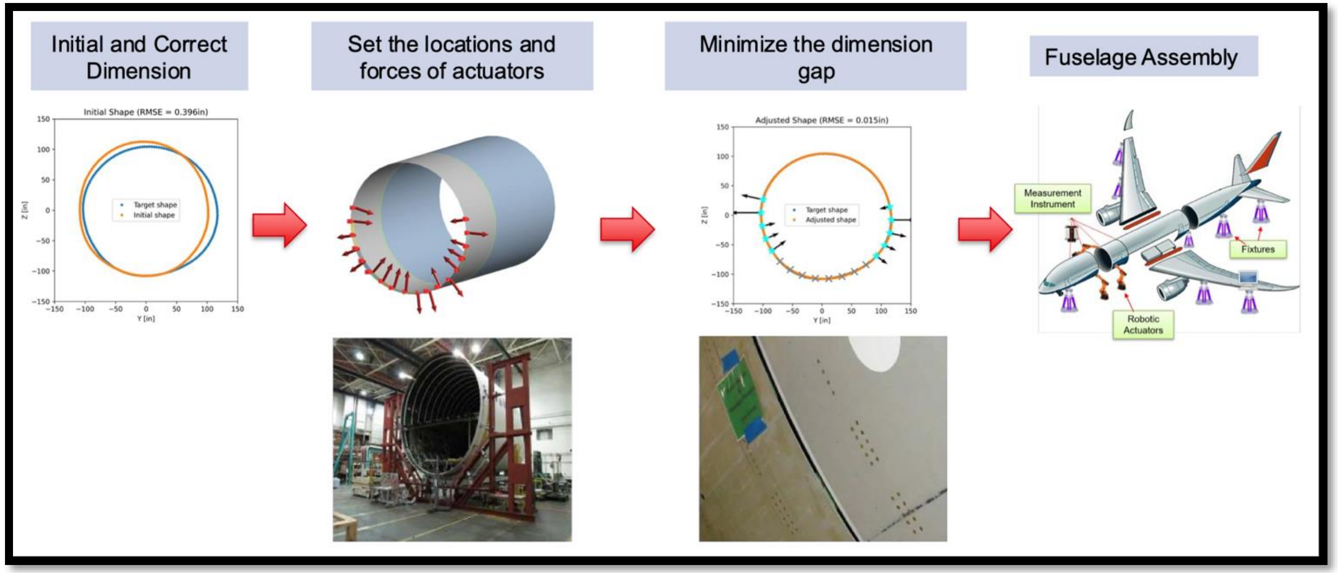
- A2.
- $\frac{\partial f}{\partial x} = e^{x+y} + 3ye^{3xy}, \frac{\partial f}{\partial y} = e^{x+y} + 3xe^{3xy} + 4y^3e^{y^4}$.

This course heavily uses mathematics

- Points:
  - 2 points: you are ready for this course!
  - 0-1 points: we will have review sessions in Week 3, but you need to catch all technical details.
- Why math is so important in numerical methods?
  - This is the only language that we talk with data and computers
    - define a root finding problem == define a math problem
    - Solving high-dimensional problems == applying linear algebra
    - ...
  - This course aims at helping you **understand** numerical methods, rather than teach you to use tools

Sources of numerical errors

- Data are always noisy
  - Discussion: Any example in your mind?
  - Aircraft fuselage design
- Computers can only handle discrete data
  - Think about data structure: string, array, tree, …
- No measuring device is perfect
  - Discussion: Any example in your mind?
  - Exact rainfall of Albany, NY in July 2025? No way!



Types of errors

- Discretization error: we can only deal with values of a function at finitely many points. For example, a very simple way to do numerical differentiation for a function f is to use a finite difference formula:

$$\hat{f}(x) = \frac{f(x+h) - f(x)}{h}. \tag{2.1}$$

  Here, the parameter h is some small number. It cannot be 0, so this introduces discretization error. Recall that the definition of the derivative of a function at a point x is

$$\lim_{h \to 0} \frac{f(x+h) - f(x)}{h}. \tag{2.2}$$

  Later in the course, we'll considered better methods than this.

- Convergence error: in which we, say, truncate a power series expansion, stop an iterative algorithm after finitely many iterations, etc.

- Rounding error: This arises because computers have only finite precision. We can only store a finite amount of data in any given machine. Interestingly, in numerical differentiation, there is a tradeoff between discretization error and rounding error (since we cannot make h infinitely small), and this leads to some optimal choice of h! So multiple types of error can play an important role simultaneously in some problems.

Example of error tolerance

Example 2.1 (Figuring out error tolerance from a problem specification)**.** *Suppose that we are observing an asteroid and would like to know whether or not it will collide with Earth. We can boil this down to numerically solving a differential equation for $X(t)$, the position of the asteroid at time t with respect to the center of the Earth (it is a three-dimensional vector of real numbers, measured in kilometers). The radius of Earth is approximately* 6378.1*km, so if we use a numerical method to solve the differential equation and find that the asteroid will at time t be in position*

$$X(t) = \begin{pmatrix} 6380 \\ 0 \\ 0 \end{pmatrix}, \tag{2.3}$$

then we only know for certain that it will not collide with Earth at time t if we can guarantee that the amount of numerical error is small enough so that $X(t) + error$ is not within the ball of radius 6378.1 *km.*

How do we measure error?

- Unknown true number is u
- We are approximating u using our number v
- Discussion: how can we measure error?

One way is to take the absolute difference

- Definition of **absolute error**:
- $|u - v|$

- It's so simple, but it has some problems:
  - Suppose we try to figure out how much air passes through a Boeing 737 engine per minute during the flight.
  - The true answer is 120,000 pounds.
  - Your solution shows 119,000 pounds, so absolute error is 1,000 pounds.

  - Discussion: You missed 1,000 pounds? Are you doing a good job?

Another way to measure error

- Definition of relative error:
- $\left|\frac{u-v}{u}\right|$
  - note u is the true number

- Discussion: What's the relative error in previous example?
  - Suppose we try to figure out how much air passes through a Boeing 737 engine per minute during the flight.
  - The true answer is 120,000 pounds.
  - Your solution shows 119,000 pounds.