# CSCI 3656 Numerical Computation Spring 2016

| | |
|---|---|
| **Instructor:** | Prof. Liz Bradley<br>`lizb@cs.colorado.edu`; 303-492-5355; ECOT 747. |
| **Office Hours:** | Mondays 1:15pm-2pm, thursdays 11am-noon, fridays 10-11am, and by appointment at other times. Please email me to set up a meeting. |
| **Class Meetings:** | Tuesdays and thursdays 2:00–3:15pm in ECCR 155. |
| **Webpage:** | `www.cs.colorado.edu/~lizb/na-course.html` |
| **Email:** | I will use the email addresses provided by D2L for class and individual messages. |
| **Textbook:** | Sauer, *Numerical Analysis* |
| **Prerequisites:** | Calculus 1 and 2 and Linear Algebra, all with grades of C- or better, and programming competence. |

**Course Description:**

This course covers the design, implementation, and analysis of methods for solving applied mathematical problems on computers. Numerical methods are at the core of much computer-based research and design in science, engineering, animation, game design, and many, many other fields. Applications range from airplane design and climate modeling to the evaluation of techniques for cleaning up radioactive waste. Numerical methods also play a role in such computer science applications as graphics, information retrieval, and artificial intelligence. We will study a selection of numerical tools that are used in the aforementioned applications. A rough outline follows; a more-detailed schedule is posted on the course website. That schedule is subject to change–check it often.

- Administrivia and course overview
- Computer representation of numbers
  *Binary floating-point arithmetic and the IEEE standard.*
- Error in floating-point computation
  *What can cause you to get the wrong answer.*
- Problem solving strategies
  *Techniques and strategies for tackling mathematical (and other) problems.*
- Solving systems of linear equations
  *Finding the vector $x$ that solves $Ax = b$: Gaussian elimination, LU factorization with pivoting.*
- Least squares
  *How to find the solution to an overdetermined set of equations.*
- Interpolation and curve fitting
  *How to take a sparse data set and find a curve that "connects the dots": splines and company.*
- Solving nonlinear equations
  *Finding the zeros of a function: Newton's method and friends.*
- Numerical integration and differentiation
  *What to do when integrals and derivatives are needed on the fly.*

- Numerical solution of ordinary differential equations
  *Cranking out a point-by-point solution of a differential equation.*
- Numerical solution of *systems* of ordinary differential equations
  *Moving on to more complicated and interesting problems: a baseball in flight, molecular dynamics, etc.*
- Advanced topics: solving PDEs, computer graphics, animation, ... depending on time and interest.

**About Language**

MATLAB is the accepted language for prototyping of code and testing of concepts in many areas of numerical computation. You are not required to use MATLAB in this course (any language is acceptable), but you'll find it convenient for much of the programming you'll do. Solutions to programming assignments will be published in MATLAB.

MATLAB is available in ITS labs. The CU site license also includes student computers now, which is a major win. There are instructions and links on the course webpage for downloading and installing your copy.

**Using the Course Website**

Lecture topics, reading assignments, solutions, exam review sheets, and all other course materials will be posted on the course website. You are responsible for keeping up with reading and assignments even if they are not announced in class.

Note that the schedule may change at any time.

**Requirements and Grading**

Grades will be based on homework assignments (40%) and hour exams (60%). Class participation will have a positive impact on your final grade.

There will be 12 homework assignments. These must be turned in at the **beginning** of class on the day that they are due (thursdays for the first month or so, then tuesdays through most of April). It is your responsibility to check the website each week for the homework assignment. Solutions will be posted after the assignment's due date. *Late homeworks will not be accepted* (but early submissions are fine). Please note: the problem sets vary in difficulty, which is reflected in the point weights shown on the corresponding handouts: some will be worth 10 or 20 points, but others will be worth 60 or 70 points. Please plan your time accordingly.

I encourage you to work together to figure out the homework problems, but writeups must reflect your own individual work and understanding. You may write any program together with one or two partners, but you **must** include the names of your collaborators on any joint work you hand in. There will be code problems on all of the exams, so turning in code that you do not understand will probably backfire on you.

There will be three one-hour exams. You will be given 75 minutes to work on these exams. The first two will take place in class on February 16th and March 17th. (*Note*: that is the thursday before spring break.) The third hour exam will be held during the first half of the final exam period. An optional *comprehensive* fourth make up exam covering the material of the first three exams will be offered during the second half the final exam period. You may elect to replace one of your first three exam grades with your grade on the make up exam. Please study together for the exams; it's a great way to learn.

**Feedback**

If you have thoughts about how the course is going (good or bad) or suggestions for improvement, please pass them along. Your opinion is important!