

CE 201 – CIVIL ENGINEERING COMPUTING

Part I - Numerical Methods with MATLAB

Having a working knowledge of numerical methods and basic programming concepts is important because many practical problems in engineering cannot be solved with analytical formulas. Examples include flow in pipe networks, analysis of beams of varying cross-section and loading, seepage through an earthen dam, and routing of storm water runoff through a detention basin. Although all of these problems can be solved using commercially-available software packages (SAP2000, WaterCAD, etc) that were developed using numerical methods, being able to run a software package is not equivalent to understanding what it is doing!

In your calculus and differential equations classes you have already learned about some numerical solution methods (e.g. Trapezoidal rule for integration, Newtons for root-finding, Runge-Kutta for ordinary differential equations). In this class you will learn that such methods are also very useful and widely used by professional engineers. In addition, acquiring basic programming skills helps you think logically and clearly in solving engineering problems. The first four weeks of the course will focus on learning how to program with MATLAB. For the remaining three weeks, students will use their programming skills to implement numerical solutions for problems of interest to civil and environmental engineers.

Prerequisite: CE 271 – Design I and MATH 264 – Differential Equations

Professor: Anne Raich, Ph.D. 322 ACE raicha@lafayette.edu
330-5590 <http://ww2.lafayette.edu/~raicha>

Lecture: MW 9:00 – 9:50 am, AEC 315
Computer Lab: Th 8:00 – 9:50 am or Th 1:10 – 3:00 pm, AEC 223B
Opt. Recitation: F 9:00 – 9:50 am, AEC 315 (Note: Students are not required to attend recitation)
Office Hours: MW 10:30 – 12:00 & 1:00 – 2:30 and T 2:30 – 4:00 pm
Other times, open door policy

Textbook: None required. The Intro to MATLAB section provides several useful references and websites for additional MATLAB information. For the numerical methods, you can use a basic text on numerical methods as a reference. Information, however, concerning each method we cover will be provided in class notes and handouts.

Grading: Six projects (assigned once a week) 75%
Final programming project 25%

Course Objectives:

The seven-week course provides students with an introduction to numerical methods in civil engineering, using MATLAB as programming tools. Each student will:

- Gain an understanding of the role of numerical methods in civil and environmental engineering
- Develop an understanding of how MATLAB can assist in solving problems using numerical methods
- Apply MATLAB programming skills to applications involving numerical solution methods from two areas within civil and environmental engineering (e.g., structures, geotech, water resources, environmental)
- Develop a final program, along with a detailed users manual, that solves an application in one of the two areas of civil engineering covered in class, or another application of your choosing

Attendance and Participation:

Attendance and effort will also play a role in your final grade. You are responsible for all material covered in class, even if absent for authorized activities. Attendance is mandatory during labs on Thursdays. If you will miss a scheduled lab, you must arrange to make it up before your absence.

Week	Tentative Course Topics	Programming Assignment	
1	Introduction to Course Intro to MATLAB Numerical Methods for Root Finding	Root-Finding Bisection Method Newton's Method	
2	Advanced Programming - Variables - Control Statements - Plotting in 2D	Numerical Integration Trapezoidal Rule Simpson's Rule	
3	Input/Output using the screen, files, and dialogs Advanced Graphics MATLAB functions	Manning's Equation	$Q(cfs) = \frac{1.49}{n} R^{2/3} S^{1/2} A$
Structures Application – Beam Deflection			
4	Systems of linear equations Arrays and Matrices Advanced Control	Solving Systems of Linear Equations Gaussian Elimination Pivoting	$\begin{matrix} 3x_1 + 2x_2 = 3 & \Rightarrow & \text{find } \{x\} \\ 4x_1 - 2x_2 = 7 & \Rightarrow & \{x\} \end{matrix}$ $[A] \{x\} = \{b\}$ OR
5	Nonlinear ODEs Boundary Value Problems Basic GUI Development	Numerical Differentiation Forward/Backward/Central Difference Formulas Solving BVP ODEs using the Finite Difference Method	
Water Resources Application – Detention Basin Routing			
6	Nonlinear ODEs Initial Value Problems	Solving IVP ODEs Euler Method Runge- Kutta Method Routing Problems	
7	Detention Basin Routing Advanced GUI Callbacks	Beam Deflection Problems Routing Problems	

Programming Assignments:

Programs will be assigned each week in Lab. The labs will involve programming in MATLAB. Students can discuss the assignments and help each other with MATLAB syntax issues, but the work submitted for grading must be each individual student's work. Programming assignment grades will be based on:

- Program works correctly (instructor will use test cases) = 5 pts
- Logic, User friendliness, and Documentation (all code must be commented) = 3 pts
- Professionalism and Aesthetics = 2 pts

Assignments will be submitted by copying your program and documentation files into a single folder, which is named with your last name and the lab number, and placing it in the 'ghost' folder located at P:/CEEdrive/CE201/dropbox. Labs must be submitted by 5:00 pm on the date due. Grades are marked down by 10% for each day late.

Final Project:

The final project will be a more advanced version of a program developed previously in the semester, or a program that uses numerical methods to solve a problem of particular interest to you, such as in your area of research. The final project also must include a detailed user manual to accompany your program. Students are encouraged, but not required, to work in teams of two on the project. Additional details will be provided during the third week of class in a separate handout.

Academic Integrity Statement: “Students are expected to be honorable, ethical, and mature in every regard”

No form of scholastic misconduct will be tolerated. Academic dishonesty includes cheating, fabrication, falsification, multiple submissions, plagiarism, complicity, copying homework, etc. It is the student’s responsibility to comply with the *Lafayette College Student Handbook* (http://www.lafayette.edu/student_life/download_handbook.html) and to be familiar with the *Principles of Intellectual Honesty* (<http://www.lafayette.edu/academics/honesty.pdf>). Violations will be handled in accordance with the Procedural Standards in Disciplinary Proceedings outlined in the *Student Handbook*.

More on Academic Dishonesty:

You are encouraged to use a USB memory stick or your home directory to save your work, not a publicly accessible drive. If programs show evidence of direct copying between students, the assignments will be forwarded to the Dean of Studies for disciplinary action according to college policy (as faculty members, we are required to report any suspected cases of academic dishonesty).

ABET Outcomes:

This course focuses on ABET program outcome (a.)) an ability to apply knowledge of mathematics, science, and engineering; (e) an ability to identify, formulate, and solve engineering problems; and (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. Other outcomes addressed include (i) a recognition of the need for, and an ability to engage in life-long learning

Students with Disabilities: The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. If you believe you have a disability requiring an accommodation or require assistance with academic concerns/accommodations, please contact the Office of the Dean of Studies (610-330-5080).

Information about MATLAB:

The current version of MATLAB is V7, which is (surprisingly!) an update of Version 6. MATLAB is found as a stand-alone program on computers in the CE student lounge, the CE computing labs, and several locations across campus. Starting MATLAB requires double-clicking on the desktop icon or locating MATLAB in the program listing under the Start menu.

MATLAB is powerful, yet relatively easy to use (you may have already used it for plotting functions or other calculations). There is no required text; notes and handouts on the topics covered will be provided. However, because we don't have the time to cover all the details of MATLAB, and lots of questions can come up during programming, having a basic MATLAB reference such as the following can be helpful in working on the labs and project for the course. MATLAB Demystified, McMahon, 2007, 9780071485517 or Mastering MATLAB 7, Hanselman/Littlefield, 2005, 9780131430181. There are many other MATLAB books out there that you can use.

You can also find A LOT of information concerning MATLAB on-line, starting with the Mathworks site <http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html>

This website includes tutorial, demonstrations, videos, sample code, and almost anything else you would want to know or learn about MATLAB. Of specific interest are the Tutorial PDFS - which cover a broad range of topics. These are accessible in web form by using the links on the left-hand side of the webpage given above (Getting Started, Examples, Programming, etc). The same tutorials are also available as PDFs, which if you scroll down on the webpage you can view or download. CAUTION: do not print these out as many of the PDFs are very, very long!

Coding and Debugging:

IMPORTANT! Writing and debugging code is time-consuming, and often frustrating, process. Very rarely do engineers write programs that run perfectly on the first try! If the program runs, but produces errors or crazy results, there is something wrong with your programming logic. If the program doesn't run, you have a syntax problem. Sometimes MATLAB will try to give you clues as to where and what the error is and sometimes it won't be able to. Since MATLAB is new to you, I am happy to help with syntax issues - logic issues you may have to work through on your own. To also help you in this process, MATLAB provides some sweet help and debugging aids and color indicators also. You can type the following at the command prompt learn more about any function or command: > help *subject* or > doc *subject*

In MATLAB you can also often run portions of the code you are writing in the command window to test out what happens. Other ways to help debug a program include tracking the progress and calculations made by outputting information about current variable values or current location in the program to a file or to dialog and message boxes. We will cover how to open files, write data, and close files in class. Any extra code you write to help debug a program can be easily commented out with a percent sign (%) at the beginning of any line of code.