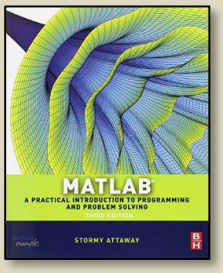
Quantitative Data Analysis for Earth and Environmental Scientists

ENV3455/GLY3455: Section 101

Fall Semester 2015

Instructor ..... Dr. Scott T. Marshall
Office ..... Rankin Science South, Room 111
Office Hours ..... Mon 10-11, 12-1; Tues 12-1; Wed 10-11; Fri 10-12
Phone 828-265-8680
Email ..... marshallst <at> appstate <dot> edu
Course Website ..... http://www.appstate.edu/~marshallst/GLY3455



Required Textbook

"Matlab: A Practical Introduction to Programming and Problem Solving" By: Stormy Attaway

We will supplement the textbook with several readings from peer-reviewed publications in the Earth sciences.

Additional Reading: MATLAB has an extensive built-in help database and all commands are well-documented. Also, there are millions of webpages devoted to various aspects of MATLAB coding. So, if you get stuck, consider performing a Google search.

Class Meeting Time / Location

Lecture: Mon/Wed: 11:00 - 11:50am Location: Rankin Science West 023
Laboratory: Wed: 12:00 - 2:50pm Location: Rankin Science South 118 (Computer Lab)

Course Summary

Modern Earth and environmental scientists deal with complex and often very large quantitative data sets that are typically not useful or understandable in raw form. Thus, quantitative data analysis skills are highly desired and useful in quantitative Earth science subdisciplines. This course provides an introduction to processing, visualizing, and interpreting quantitative Earth and environmental science data using scientific computing techniques widely used in the Earth sciences. Computational methods and visualization will be performed using the scientific computing language, MATLAB. Previous programming experience is helpful, but not required. Biweekly lectures introduce the necessary computational background and the weekly laboratory assignments demonstrate applications to Earth science data. The weekly laboratories will involve writing algorithms that utilize various quantitative methods to process and visualize data relevant to the Earth sciences. Topics covered include Earth science applications of: conditional statements, loops, vector operations, automated data analysis & visualization, differentiation, interpolation, curve fitting, image processing, and three-dimensional visualization.

Course Goals

The main goal of this course is to provide a computational and quantitative skill set relevant for processing, filtering, analyzing, and visualizing quantitative Earth science data efficiently and accurately. By completing this course, students will gain experience in the basics of computer programming, data visualization, and mathematical principles relevant to Earth sciences. Students will learn to make their own custom tools that automate computation and visualization tasks so that a problem need only be solved once. While much of the lecture content will focus on the programming basics necessary to utilize MATLAB, the lab will focus on Earth science applications of programming. The overarching goal is that the course will demonstrate the wide applicability of computation in the Earth sciences and provide students with the confidence to pursue quantitative research projects during their academic and professional careers.

LABORATORIES

Weekly laboratory sections will meet in the geology computer lab in Rankin Science South room 118. Laboratories will be mostly computer-based; however traditional pencil and paper planning and calculations will be performed to validate algorithms and to test knowledge of computational tasks. Thus, laboratory assignments will be a mix of basic MATLAB methodology questions and applications of computational methods to Earth science data. Laboratory exercises have been designed to reinforce programming methods taught in lecture and are key to understanding course material. Students should come to every lab with paper, pencils, and a USB flash memory stick (often called thumb drives, or flash drives). Memory sticks can be purchased online or at virtually any store in town that sells electronic devices. I recommend at least 4 GB; however, smaller sizes may suffice if you are exceptionally organized with your data. The purpose of the USB devices will be to save your lab assignments because the GLY computer lab is set up so that each machine erases itself upon each and every reboot.

GRADING

Final grades will be determined using the following rubric:

Exams I, II, & III	**20% each (60% total)**
Lab	**40%**

At the end of the semester, your final grade will be tallied and reported as a letter grade. Each letter grade corresponds to the numeric value below.

A	**93 – 100**	**B+**	**87 - 89**	**C+**	**77 - 79**	**D+**	**67 - 69**	**F**	**≤ 59**
A-	**90 - 92**	**B**	**83 - 86**	**C**	**73 – 76**	**D**	**63 – 66**		
		B-	**80 - 82**	**C-**	**70 - 72**	**D-**	**60 - 62**		

LATE POLICY & MAKE-UP EXAMS

All lab assignments are due by the beginning of the next lab period unless otherwise stated. Any late assignments will be penalized 10% per day (including weekend days) and will not be accepted after 5 days. Many assignments in this class will be handed in via email attachment. In these cases, **I will only accept a single submission**. Additional emails will be ignored. In the case that something is missing, a resubmission will be allowed, but you will lose 10% per resubmission on top of any other late penalties. So, if you know of a scheduling problem that may prevent you from turning an assignment in on time, talk to me in advance so that we can work out an alternate plan. I am very flexible when you come to me ahead of time. Make-up exams are generally not offered, unless we work something out at least 7 days prior to a scheduled exam. To be fair to students that show up for exams, I reserve the right to make any make-up exam to follow an all essay format, or any other format of my choosing. In the rare case that a make-up exam is granted, the exam must be made up within one week of the original exam date.

PLAGIARISM/CHEATING AND OTHER UNIVERSITY POLICIES

Plagiarism/cheating is here defined as a student claiming someone else's work as his/her own. For example, but not limited to, failing to reference an original work in a research paper, or copying answers during an exam or lab exercise, or **copying and pasting code from a web source**. Students should be familiar with the University's Academic Integrity Code. Those students suspected of infringement on the Integrity Code will be confronted. Those students who are caught cheating or plagiarizing will be asked to meet with the professor, will have their rights explained to them, and will be dealt with according to the procedure dictated by the Office of Student Conduct. The University has posted important policies that apply to all syllabi. You can find these policies online @ http://academicaffairs.appstate.edu/syllabi

WORKLOAD AND TIME MANAGEMENT

The typical rule of thumb for University courses is that each credit requires 3 hours per week of outside work. Thus, this course should be expected to take at least 9 hours of work per week outside of class. The lecture portion of class will have no significant assignments other than keeping up on reading the assigned textbook chapters. Thus, most of the out of class time will be spent working on lab assignments. Lab assignments will typically require only a few hundred lines of code (including comments), which does not take that long to write if the user uses his/her time wisely. With that said, I do understand that most students in this course have not programmed before, so some lab exercises may require longer than one might expect. Unfortunately, there is no way for me to know how long each exercise will take any/all students since each student is different in ability and experience. **Below is some hopefully helpful advice:**

1) The absolute best way to learn a programming language is to take simple examples, and then play around with various commands to see how things work. When you are reading through the text and my lecture notes, have MATLAB open so you can try the examples. My lab assignments assume that this is how you are reading the text every week.
2) Don't forget to read the built-in MATLAB documentation. It is very complete and comes with simple examples that you can experiment with. This is how I learned MATLAB. I did not have a formal MATLAB course in college or grad school.
3) Stay in lab for the entire lab period and ask questions to make sure you have made significant progress before lab ends. Lab periods are your chance to get guided help in getting your code started.
4) Try to finish your labs several days before they are due. This way, if you get stuck, you will have time to ask questions to me or your classmates. If you try to do the assignments on the night before they are due, you will likely become very stressed and frustrated.
5) If you are stuck on the same problem for more than an hour, take a break. I cannot count how many times I have realized a mistake, or figured out the solution to a problem while hiking or biking. If you give it some time, and still cannot figure out a problem, come see me.
6) Always remember that there are usually many ways to code up the same result. So, just getting the correct result is only part of the game. A good algorithm is mathematically correct, easy to understand, well-documented, and computationally efficient. This is how you will be graded in this course.

MATLAB AVAILABILITY

All computers in the GLY department have MATLAB installed. This includes our lab classroom as well as the URL. If you would like to get MATLAB for your own personal computer, you can buy it from http://www.mathworks.com/academia/student_version/ . All you need is the "MATLAB Student" version, which currently costs $49. We do not use any of the toolboxes, so the "MATLAB and Simulink Student Suite" is not necessary. Alternatively, if you are a Linux/Unix geek like me, you can download Octave, which is the open source version of MATLAB. The downside to Octave is that it has no GUI (Graphical User Interface) and it is not always 100% compatible with MATLAB. You must use it in a terminal window, so it requires that you have some sort of Linux/Unix terminal available to you. Incompatibilities between Octave and MATLAB will not be excused when grading.

NEED HELP?

This course is somewhat unique because it is cross-listed in both environmental science (ENV3455) and geology (GLY3455). I expect that students in this class will have a wide range of academic backgrounds. If you feel that the course is covering material that you do not have a strong background in, feel free to stop by my office or email me to set up an appointment. We can determine what your needs are and how you can be most successful in this course. Even if you have taken lots of geology, physics, computer science, and mathematics courses, you are still welcome to ask for supplementary materials on topics relevant to computational methods.

TENTATIVE COURSE SCHEDULE

*DID I MENTION THIS IS TENTATIVE?

Date	Topic	Reading
August 17	Course Introduction	N/A
August 19	Introduction to MATLAB	Chapter 1
August 19	Lab 1: Computing Basics	
August 24	Introduction to MATLAB	
August 26	Introduction to MATLAB	Chapter 1
August 26	Lab 2: MATLAB Basics	
August 31	Introduction to MATLAB Programming	Chapter 2
September 2	Introduction to MATLAB Programming	Chapter 2
September 2	Lab 3: Scripts	
September 7	*No Class: Labor Day*	
September 9	Introduction to MATLAB Programming	Chapter 2
September 9	Lab 4: Functions	
September 14	*No Class: Dr. Marshall at SCEC Meeting*	
September 16	*No Class: Dr. Marshall at SCEC Meeting*	
September 16	*No Class: Dr. Marshall at SCEC Meeting*	
September 21	Selection Statements	Chapter 3
September 23	Selection Statements	Chapter 3
September 23	Lab 5: Selection Statements	
September 28	Loop Statements	Chapter 4
September 30	Loop Statements	Chapter 4
September 30	**Exam I**	**Chapters 1-2**
October 5	Loop Statements	Chapter 4
October 7	Vectorized Code	Chapter 5
October 7	Lab 6: Loops	
October 12	Vectorized Code	Chapter 5
October 14	Vectorized Code	Chapter 5
October 14	Lab 7: Vectorized Code	
October 19	String Manipulation	Chapter 7
October 21	Data Structures: Cell Arrays and Structures	Chapter 8
October 21	Lab 8: Derivatives and Stream Knickpoints	
October 26	Advanced File I/O	Chapter 9
October 28	Advanced File I/O	Chapter 9
October 28	Lab 9: Cell Arrays and File I/O	
November 2	Image Processing	Chapter 14
November 4	Image Processing	Chapter 14
November 4	**Exam II**	**Chapters 3-5**
November 9	Image Processing	Chapter 14
November 11	Curve Fitting and Interpolation	Chapter 15
November 11	Lab 10: Image Analysis	Chapter 8
November 16	Curve Fitting and Interpolation	Chapter 15
November 18	Curve Fitting and Interpolation	Chapter 15
November 18	Lab 11: Curve Fitting and Interpolation	
November 23	Advanced Plotting Techniques	Chapter 11
November 25	*No Class: Thanksgiving Break*	
November 25	*Lab 12 (Take-Home): Fancy 3D Plotting*	
November 30	Advanced Plotting Techniques	Chapter 11
December 2	Advanced Plotting Techniques	Chapter 11
December 2	Work Period. Lab 12 due Dec 3 by noon	
Dec 9 (Monday)	**Exam III (9:00-11:30am in RSW 118)**	**Cumulative: Covers all Course Content**

** There are absolutely no make-ups or early exams for the final exam. It is the responsibility of each student to schedule his/her departure after all exams are complete.