Math290

Wednesday 6:00-8:00pm Philips Hall 332

Instructor: Austin Baird
Office: Philips 368
Email: abaird@live.unc.edu

Course Description and Goals

This course is designed to supplement math 528 *or* be a standalone lab course which will begin with basic Matlab programing skills. The math topics covered will mirror math 528 but consideration on pacing will be taken to make sure that everyone gets up to speed with Matlab (I will not assume any prior experience). The goal of this course is for the student to learn to use technology to help aid, or sometimes supplant, their analysis in applied math topics. How do you solve a deterministic system when no analytical solution exists? How would one visualize this solution? How accurate is it and to what lengths can we trust this solution? These questions will be addressed and explored through the students work computer work and analysis. This will not be a course where I give you a program and you hit play. You will be expected to take mathematical topics and transfer them into a program with proper algorithm design. You will also be expected to analyze the result properly. Additional Python implementation to the Matlab material will be a voluntary addition if the student wishes to use free software which is quickly becoming the scientific computing standard at many institutions and companies.

Objectives

- Learn to install and use the most basic features of Matlab and mathematica.
- Learn to take a mathematical subject and transform it into a computational algorithm.

- When is a computation result needed? When can Mathematica aid in our analytical understanding?
- Analyzing data and create an understandable answer using visualizations (i.e. graphs).

Optional

- Installing Python and using it's basic functions.
- Why Python?
- How to convert a Matlab program to Python.

Expecations for Students:

This is a course for credit and as such there will be a homework, midterm and Final component. You are expected to attend every course (since there is only one class per week) and a grade will be attributed to in course participation. You *are* expected to google things! Problems In scientific computing often arise from improper use of the software or confusion as to how to install/use certain programs. I will be available for all inquiries regarding these questions, but I recommend googling your problem first. There is also a chat room on the sakai site which can also be used to ask about problems on homework. I encourage all people to work together because often times this is how computational problems are worked through. The midterm and final will be individual efforts so please don't take this as an excuse to completely copy your peers work!

Prerequisits

I don't believe there are any.

Reading

- *Required Text:* `http://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf`
- Various papers and code will be posted on the course website.

Software

Students will need to use Matlab and Mathematica. These programs are free from UNC and are easily obtainable from UNC's software acquisition desk on the basement of the undergraduate library. Any prior knowledge of these programs will not be needed and everything will be taught from the ground up.

Course Requirements

Homework	40%
Participation	10%
Midterm	20%
Final	30%

Policies and Late Assignments

Late homework will not be accepted, but if your schedule doesn't permit you to be in class the day homework is due you are more than welcome to turn it in early, just email me for details. For *extreme* circumstances late homework may be accepted and course participation not necessary.

*Students with medical or learning disabilities: I strongly encourage you to contact me during the first week of the semester to let me know of your situation. I am happy to work with you to accommodate any special needs or requirements.

Course Schedule

Week 1: Introduction to matlab, installing, using the interface, and manipulating array objects.

Week 2: Control sequences (for, if), using built in functions, creating our own functions, and plotting.

Week 3: Numerical differentiation, solving linear systems and linear/nonlinear ode systems.

Week 4: Numerical integration: rectangle and trapezoid rule, errors in numerical outputs.

Week 5: Runge-Kutta methods, error analysis.

Week 6: Local truncation error, determining the order of a numerical method, global error, and norms.

Week 7: Stability analysis, equilibrium solutions, and eigenvalues and eigenvectors.

Week 8: Modeling physical systems, pendulum example, analyzing output using Matlab, and phase plane diagrams.

Week 9: Modeling population growth and epidemiology: stability, equilibrium solutions, and eigenvalues/vectors.

Week 10: Approximating functions using series representations, errors and truncation.

Week 11: Periodic functions, Fourier series representation of periodic functions, Fourier integral representation of non-periodic functions, discrete Fourier transform and the FFT.

Week 12: FFT of real data, how to use fft function in matlab.