Applied Mathematics 111
Introduction to Scientific Computing

Instructor:	Mauricio Santillana	
Office:	Pierce Hall 312	
Phone:	617-495-2891	
Email:	msantill@fas.harvard.edu	
TFs:	Armin Pourshafeie	Clifton Dassuncao
Email:	apoursh@college.harvard.edu	cld292@mail.harvard.edu

Meeting time and location: Tue and Thu, 10am-11:30am in Cruft Hall 309.

Office hours:
Mauricio: Tue and Thu, 11:30am-12:30pm in Pierce 312
(Please e-mail me to request an appointment outside of the official office hours)

Armin: TBD

1. Prerequisites:

Applied Mathematics 21a and 21b, or Mathematics 21a and 21b, or permission of instructor. Programming experience in Matlab is helpful but not required.

2. Course description:

Relevant issues in our society such as climate change and financial strategies are being addressed using computational techniques. Moreover, the use of computers as a tool for scientific research has become pervasive. Careful understanding of how frequently-used computing methods work provides us with a powerful way to address current open problems and helps us identify the validity of the results obtained with such techniques. In this course, we will introduce the background needed to understand well-established numerical and computational approaches and will explore their use through practical examples from different disciplines such as physics, chemistry, biology and finance. *We will use matlab in the course.*

2.1 Course objectives: At the completion of the course, students will be able to:

- identify appropriate mathematical algorithms for specific goals/applications,
- understand the requirements, limitations and validity of such algorithms in a given mathematical problem,
- implement well-known mathematical algorithms from primitive programming structures (from scratch) using matlab,

- utilize professionally implemented routines/functions/algorithms in order to formulate appropriate strategies to solve challenging and complex real-life questions.

3. Required work and grading criteria

You are expected to attend and participate in class.

3.1 Problem sets: They will be assigned approximately weekly. Most of them will have a theoretical and a programming component. You are expected to turn them in a week after they are assigned. For each problem set, you will upload a zip file containing (I) a working Matlab code and (II) a write up that summarizes your findings to the course dropbox. Please name your zip file using the following convention: *familyname_firstname_ps#.zip*.

Late homework policy: We will not accept late homeworks. However, for your final grade you can drop one problem set in the semester.

Collaboration policy: Working with your peers is a great way to learn. While we expect this to happen, we require you to submit your own work separately and explain your findings with your own words.

3.2 Final project: One of the main goals of the class is to help you understand how scientific computing techniques can be powerful tools to solve real-life problems. With this in mind, you will work on a final project (depending on the size of the class you may work with a partner). We will help you choose a problem that requires the use of computational tools of interest to you during the semester. You will work on your project at least during the last three weeks of classes. I will be available to discuss your ideas regularly during those three weeks and throughout the semester. You are required to turn in a one-page **project proposal by April 1st, 2014** (this submission will not be graded; it is intended to get you started with your project). A brief written report and a presentation of your **final project is due on April 29th**. You will be required to present your work in front of your peers at the AM 111 fair on April 29th.

Final project report (due on April 29th, 2014): This brief written report will contain (I) a motivation stating why this topic/problem is relevant, (II) a well posed mathematical problem to be solved, (III) background of the numerical/computational approach utilized to solve it, (IV) results, and (V) a discussion of the results and their relevance to the problem at hand. A working Matlab program file containing the implemented numerical approach used is expected. A clear and concise presentation of all these aspects will contribute equally to your grade.

3.3 Grading: Problem sets 60%, Final project 40% (20% written report and 20% final presentation), and an additional potential 10% for in-class participation (in-class problem solving competitions, etc). There will be no exam.

You are expected to take the online Matlab tutorials (2 hours approx.) found at:
http://www.mathworks.com/academia/student_center/tutorials/register.html

4. Topics covered in this class:

- Computer representation of numbers
- Numerical solution of systems of linear equations
- Eigenvalue problems
- Interpolation
- Least squares problems: data fitting and supervised machine learning
- Fast Fourier Transform
- Numerical Root finding
- Numerical Optimization
- Numerical integration
- Numerical differentiation
- Solving initial value ordinary differential equations

4.1 Optional

- Solving boundary value problems
- Numerical solution of partial differential equations

5. Textbook: While there is no required textbook for the class, a good introductory-level book for the theoretical background for the class is: *Scientific Computing. An introductory survey* by M. Heath, and a very practical textbook containing many of the algorithms that we will study is: *Numerical Recipes: The Art of Scientific Computing* by W. Press et al, 2007. The latter is available online free of charge for personal use. During the semester, I will present material from diverse sources that I will make available as new topics are introduced.

Reference books that you may find interesting for the theoretical and mathematical foundations for the class include:

- J. Stoer and R. Bulisch, Introduction to Numerical Analysis. Springer-Verlag, 1993
- A. Quarteroni, R. Sacco, and F. Saleri. Numerical Mathematics. Springer-Verlag, 2000.
- D. Watkins, Fundamentals of Matrix Computations, 2002.

- T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning, 2008.

As for the practical aspects and programming resources you may find the following interesting:

- C. Moler, Numerical Computing with Matlab.
- D. J. Hingham and P. J. Hingham, MATLAB guide. Society of Industrial and Applied Mathematics, 2005.
- MATLAB documentation. Online reference. http://www.mathworks.com/.