Applied Mathematics 205
Practical Scientific Computing

Instructor: Mauricio Santillana
Office: Pierce Hall 312
Phone: 617-495-2891
Email: msantill@fas.harvard.edu

Meeting time and location: Mon, Wed, Fri, 10-11 at Maxwell-Dworkin, G 135

Office hours: Mon, Wed, Fri, 11-12 in Pierce 312
(Please e-mail me to request an appointment outside of the official office hours)

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 nowadays. Careful understanding on 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.

The course is designed to introduce you to the basic ideas behind commonly used numerical strategies and will require you to gain practical experience implementing them in order to increase your understanding of them. This approach will eventually prepare you to be a responsible user of professionally written mathematical software for your research.

We will use matlab as the scientific computing language for the course, as it provides us with a simple interface to implement ideas and graphically visualize them. Also, many well known numerical strategies have been incorporated in this software allowing us to easily expand these in order to customize them for our research purposes. Students familiar with other programming languages (and visualizing tools) are welcome to use them as long as they are able to demonstrate a full understanding of the material covered in the class with these tools. Programming support will only be offered for matlab.

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 research questions.

Required work and grading criteria

You are expected to participate in class and suggest topics that are relevant to you that may be covered in the last two weeks of classes. Also, you are expected to take the online Matlab tutorials (2 hours approx.) found at:
http://www.mathworks.com/academia/student_center/tutorials/register.html

Homeworks: They will contribute to 60% of your grade, and will be assigned approximately biweekly. You are expected to turn them in two weeks after they are assigned. These homeworks will help you become familiar with both the concepts introduced in the classs and the matlab programming environment. Grading of homeworks will emphasize your understanding and ability to solve problems using the techniques introduced in the class, and as many homeworks will involve the use of matlab, I will consider three aspects when grading the computational sections: (I) whether the program works, i.e. whether you get the expected results using your code, (II) whether the program is easy to understand: well-structured and well-documented, and (III) whether it could easily be used or adapted to solve similar problems. In other words I will encourage you to write clean, efficient and easy to understand codes.

Late homework policy: I will accept homeworks up to one week late with a 20 % penalty (provided the solution of the problems have not been discussed in the class or posted online).

Final project: As one of the main goals of the class is to help you use scientific computing techniques in your research, 40% of your grade will come from a final project. In this final project, I will help you choose a topic (desirably related to your current research area) and design a (fun and not trivial) problem that you can solve using a collection of numerical techniques (either covered in the class or not) approximately in a month. I will be available to discuss your ideas during the whole semester, and especially one week before the short description of your project is due (Nov 8th, 2010). In order for you to focus on your final project, there will not be any homework assigned for the last two weeks of classes. The final projects need to be approved by me by Nov 12th, and you will have at least a month to work on them.

Short description (due on Nov 8th, 2010): This brief (2 pages approximately) report should contain a description of the problem that you intend to solve as your final project along with a proposal containing the numerical strategies that you intend to use in order to solve it. This description will not be graded but you are required to turn it in so that I can approve your project.

Final project report (due on Dec 12th, 2010): This 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 in 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. Further details on the format and legth of this final project will be available on Nov 12th, but basically the amount of work equivalent to a section in a paper is expected.

Topics covered in the class:

- Computer representation of numbers. (*Floating point numbers and arithmetics, finite precision, error propagation and catastrophic cancellation*)
- Numerical linear algebra (*Gaussian elimination, LU factorization, banded systems, iterative methods, eigen value problems*)
- Numerical integration (*Quadrature rules: Newton-Cotes, Romberg and Gauss methods, Monte Carlo methods*)
- Root finding (*Bi-section, Regula-Falsi, Newton-Raphson*)
- Optimization (*Section search, Newton's methods: steepest descent, conjugate gradient, simulated annealing, other ones*)
- Numerical differentiation (*finite differences, spectral differentiation*)
- Initial value ordinary differential equations (*Forward and backward Euler, predictor-corrector, Runge Kutta schemes, Stiff ODE's*)
- Boundary value problems (*Finite difference method*)
- Partial differential equations (*Linear Elliptic, Parabolic and Hyperbolic equations*)

Textbook:

There is no required textbook for the class, however, 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, . 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.

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

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