Computational Methods
CMSC/AMSC/MAPL 460

Ramani Duraiswami,
Dept. of Computer Science

Course Goals

- Introduction to the use of scientific computing techniques to solve problems in various domains
- Understand principles behind algorithms
- Intelligent choice and use of available software
- Understand how to
  - Convert a model into a discrete system on the computer
  - How to deal with data
  - perform simulations for applications
  - Display and evaluate simulation results
  - Appreciate which computations are feasible

"New Paradigm"

- Scientific Discovery through Computing
- Paradigm?
  - A set of assumptions, concepts, values, and practices that constitutes a way of viewing reality for the community that shares them, especially in an intellectual discipline.
- Engineering (aeronautics, fluid dynamics, circuit design, radar, antennas, signal processing, …)
- Physics (stellar dynamics, materials, …)
- Economics/Sociology (modeling and analyzing data, computational statistics, stock picking, …)
- Biology (biostatistics, computational biology, genomics and proteomics, …)
- Computer Science (modeling systems/network performance, information retrieval, …)
- Your field …

Another “paradigm”: Data driven science

- Grab data and process it
- Audio, video, text, MRI, X-Ray, weather, strain-gage, flow, gene-chip, seismograph, …
- Moore’s law drives both processing power, memory, sensor cost and capability
  - Moore’s law: Processor speed doubles every 18 months
  - More generally: Technology X capability will double in Y months
- Need algorithms to process larger and larger data sets, and extract information from them
  - Fit data, Extract model parameters, Learn relationships
  - In general compute with the data

The Course

- Two lectures a week
- Homework every week or other week
- 40% homework, 25% exam 1, 35 % final
  - Attendance/participation will be a factor
- Class web site: http://www.umiacs.umd.edu/~ramani/cmsc460/index.html
- Required Book

Numerical Computing with MATLAB by **Cleve Moler**

- The good news
- The complete book is online!
- Book is also not as expensive as some others (~$40)

Course

- Course comes with Matlab software that is downloadable from the book web site
- Another excellent book/resource:

Numerical Recipes by William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery

Also available online!

Go to www.nr.com

Homework

- Homework will involve programming in MATLAB
- mainly problems from the text
- Style/Clarity/Cleanliness of output will count
- Work/Results must be easily understood to be interpreted
  - Visualization (graphs)
  - Commented code

Syllabus

- **Introduction, Computer Arithmetic and Errors** (Chapter 1) (approx. 3 lectures)
  - course survey
  - introduction to Matlab
  - machine arithmetic and error analysis
  - stability and conditioning
- **Solving Linear Systems of Equations** (Chapter 2) (approx. 4 lectures)
  - Gaussian elimination
  - well-conditioning vs. ill-conditioning, matrix and vector norms
  - Notions of algorithm complexity
  - sparse systems: direct and iterative methods

Syllabus

- **Interpolation** (Chapters 3) (approx. 4 lectures)
  - polynomial interpolation
  - Other basis functions and polynomials
  - piecewise polynomial interpolation
  - spline interpolation
- **Zeros and Roots** (Chapter 4) (approx. 3 lectures)
  - Linear and Nonlinear systems of equations
  - Bisection, Secant and Newton method
  - Introduction to optimization
- **Solving Linear Least Squares Problems** (Chapter 5) (approx. 3 lectures)
  - data-fitting and least squares
  - QR factorization

Syllabus

- **Integration/Quadrature** (Chapter 6)
  - elementary integration formulas (midpoint, trapezoid, etc.)
  - compound and adaptive integration formulas
  - Gaussian quadrature
- Fourier Analysis (Chapter 8)
- **Ordinary Differential Equations** (Chapter 9) (approx. 4 lectures)
  - ordinary differential equations and Euler's method
  - adaptive methods for ordinary differential equations
  - methods for stiff systems

Introduction to MATLAB

- Vectors, Matrices, Syntax
- Vector operations, including the \dot commands
  - length, size, linspace, logspace, size, rand, randn, randperm
- Special vectors and matrices: zeros, ones, eye, magic
- Scripts and functions
  - Diary
- Graphing:
  - plot, special fonts, plot3, semilogx, semilogy, title, xlabel, ylabel, axis, grid, legend, subplot,
- Formatted output:
  - Sprintf, ;, disp, input
- Programming:
  - for, if, while, &, |, ~
- General/misc commands
  - ginput set, size, max, sum, close, figure, hist, any, all , floor, fix, round,
- Graphical programming and callbacks