Scientific Computing with Matlab

Professor Donna Calhoun

Office : MG241A

Spring 2014
Math 365

http://math.boisestate.edu/~calhoun/teaching/Math365_Spring2014

Computational science and scientific computing

"Computational science now constitutes what many call the third pillar of the scientific enterprise, a peer alongside theory and physical experimentation." (Report to the President).

Report to the President : "Computational Science : Ensuring America's Competitiveness", June 2005.

What is scientific computing?

Sidebar 1
Definition of Computational Science

As a basis for responding to the charge from the Office of Science and Technology Policy, the PITAC developed a definition of computational science. This definition recognizes the diverse components, ranging from algorithms, software, architecture, applications, and infrastructure that collectively represent computational science.

Computational science is a rapidly growing multidisciplinary field that uses advanced computing capabilities to understand and solve complex problems. Computational science fuses three distinct elements:

- Algorithms (numerical and non-numerical) and modeling and simulation software developed to solve science (e.g., biological, physical, and social), engineering, and humanities problems
- Computer and information science that develops and optimizes the advanced system hardware, software, networking, and data management components needed to solve computationally demanding problems
- The computing infrastructure that supports both the science and engineering problem solving and the developmental computer and information science

How does Matlab fit into this?

- Matlab, along side more traditional programming languages such as C, C++, Fortran has become one of the major platforms on which computational science is carried out.

- Originally designed as a computational "laboratory" for easily solving matrix equations, it has now branched into most areas of mathematics, with numerical tools for solving a wide range of complex problems in science and engineering,

- Integrated graphing capabilities make it an ideal tool for quickly prototyping codes and visualizing output,

- Easy manipulation of vectors and matrices.

Topics we will cover in this course

Some general topics we will cover

- Using Matlab as a graphing calculator,
- Writing Matlab scripts and functions,
- Program flow control (if-else, do-while loops, for loops)
- Solving large linear systems of equations (LU decomposition)
- Finding curves through a given set of data points using polynomial interpolation and approximation,
- Numerical root finding, (fixed point iteration, Newton's Method)
- Numerical quadrature (approximating integrals)
- Introduction to solving initial value problems
- Creating plots and graphs in Matlab

Textbook

- *Numerical Computing with Matlab*, Cleve Moler (Mathworks, Inc.). Available online at

  http://www.mathworks.com/moler/chapters.html

The textbook will be used in combination with course notes (often made available online) and in-class labs.

There are dozens of other online resources for learning Matlab. Let me know if you find one that is particularly useful, and I will post it on the course website.

Course schedule

- Class time will be roughly 45 minutes **lecture**, followed by hands on demonstrations and labs.
- You will each turn in a **homework** assignment roughly every two weeks, on Wednesday **at 5PM and will be considered late after 5PM on Friday.** You will each be given Dropbox accounts for turning in your homework.
- We will have occasional in-class **quizzes**
- Teams of two students will present **final projects** during finals week.

Grading policy

Grading

Homework projects will count for 70% of your final grade, final project will count towards 25% of your grade, and the remaining 5% will be in class quizzes.

Academic Honesty

Honesty Policy for Homework

- You must type in and edit your own code and exercise answers.
- Copying, either electronically or visually, someone else's code is not allowed.
- Allowing someone else to copy from you is not allowed.
- Copying code you may find on the Web is not allowed.

What you may discuss with other students

- "High level" discussions are fine, e.g. discussions about the problem statement.
- "Low level" discussions are fine, .e.g. discussions about Matlab syntax or understanding error messages
- "Mid level" discussions require discretion.
- A student who needs help with debugging a program **may** show another student his or her code.
- A student who is helping someone else **may not** show his or her code.
- Use your best judgment. If I suspect unwarranted collaboration, all participating parties are penalized (both helpers and helpees).

and finally...

Under no circumstances should you send an electronic copy of your code to another classmate. You will both be penalized for turning in homeworks that appear identical or nearly identical.