AMSC/CMSC 662
Computer Organization and Programming for Scientific Computing

Fall 2011

Introduction

Dianne P. O'Leary

Computer Science for Scientific Computing

Topic of the course: What every computational scientist needs to know in order to produce efficient and well-designed computer programs.

The plan

Goal: Give you a view of the state-of-the-art in computer science from the programmer's viewpoint.

Big difficulty: This is a moving target!

We seek codes that are:

- reliable. (In particular, we must use stable algorithms.)
- well documented.
- modular, so that they share pieces.
- convenient for the user, so we reduce the temptation to tinker.
- efficient.
- easy to modify.
- portable.

The course organization

Handouts:

- Course information: text, grading, etc.
- Course syllabus.
- Information form: please fill out at end of class.

Getting off to a good start

This week:

- Buy the textbook.
- Study the website to see what information is available to you. Find the old quizzes, the lecture notes, the schedule, and the Survival Guide for Scientific Computing.
- Begin Homework 1.
- Make a list of questions you have and ask them in office hours or at the beginning of class.
- Determine how you will access MATLAB. Try it out. Make sure that you can view and print figures and save a `diary` of your work.

About the notes:

The lecture notes are a work-in-progress. There are lots of typos in them. I'll appreciate your help during and after class to try to find all of the errors.

- I will borrow notes from many sources in order to better coordinate with the text and avoid redrawing complicated pictures.
- I gratefully acknowledge all of these sources.
- The first person to find each substantive error in material on the website will earn extra credit.

A quick overview and motivation: nontechnical

Problem: Drive your car 20 miles (on a closed racetrack) using the smallest possible amount of fuel.

- At what speed should you drive?
- What factors matter?

A quick overview and motivation: more technical

- Useful data structures.
- Basic non-numeric algorithms.
- Anatomy of a computer.
- Under the hood: from program input to output.
- Parallel computing.
- Issues: programming tools and reliability, privacy, reproducibility, language choice, the internet, green computing, etc.

Textbook overview

Slides labeled "Carnegie Mellon" are taken from the instruction site for

> Computer Systems: A Programmer's Perspective, 2/E (CS:APP2e)
> Randal E. Bryant and David R. O'Hallaron, Carnegie Mellon University `http://www.cs.cmu.edu/afs/cs/academic/class/15213-f10/www/schedule.html`

`01-overview.pdf` pp 3-21.