CS382 - Topics - Scientific Computing

Syllabus

Fall, 2007

Charlie Peck
Email: charliep at cs dot earlham dot edu
Office: Dennis 202, x1667
Course URL: `http://cs.earlham.edu/~charliep/courses/senior-seminar`
Course listserv: `cs488 at cs dot earlham dot edu`

Description

Scientific computing will introduce students to the computer science tools and techniques which support computational science and high performance computing. Computational methods are an integral part of modern science, including multidisciplinary research into *e.g.* climate change, the origins of the universe, and the underlying cause of diseases such as Alzheimer's. Major topics include scientific libraries and kernels, parallel, distributed, and grid resources, and the principle software patterns found in this domain. Applications from biology, chemistry, geology, physics, and environmental science will be explored in the labs.

As part of the course students may have an optional opportunity to go to the annual SuperComputing conference (SC07), the largest conference in the world dedicated to scientific and high-performance computing. While there we will participate in the SC Educational Program and enter a team in a scientific programming contest.

Prerequisites: CS-256 or consent of the instructor. Interested students who have completed CS-128 or posses software experience from another source are encouraged to contact Charlie about possibly taking the course.

Textbook and Other Readings

The textbooks we'll be using in this course are:

- Shiflet, Angela. *Introduction to Computational Science*, 1/E, Princeton University Press, 2006.
- Quinn, Michael. *Parallel Programming in C with MPI and OpenMP*, 1/E, McGraw-Hill, 2004.

Additionally, we'll be reading a number of papers and monographs on topics not covered in the text, these will include:

- Asanovic, Krste, *et. al. The Landscape of Parallel Computing Research: A View from Berkeley*, Electrical Engineering and Computer Sciences, University of California at Berkeley Technical Report UCB/EECS-2006-183. December, 2006.
- Goldberg, D. *What Every Computer Scientist Should Know About Floating-Point Arithmetic*, ACM Computing Surveys 23 (1991), pp. 5-48.

- Gray, Jim. *Distributed Computing Economics*, Microsoft Research Technical Report MSR-TR-2003-24, March, 2003.

There is also at least one videocast of a presentation we'll watch.

Organization

The basic components of the course are:

1. Class sessions
2. Exercises
3. Labs
4. Library research
5. Field trip - SuperComputing Conference (SC07), November 9-14 (optional)

See below, *Grading*, for a breakdown of the weights assigned to each of these components.

From a high level this course is about 1/4 high performance computing, 1/4 computational science, and 1/2 tools, libraries, and techniques. You will read prose; read, design, and write code; use other people's code; and write prose for this course.

One prominent thread you will find in this course is how we as computer scientists work with people in other disciplines to understand and model a wide range of physical systems. Much of the science done today is computational and multi-disciplinary in nature, this course will introduce you to a variety of current research problems from across the natural sciences.

Class Sessions

Class meets on Monday's and Thursday's from 2:30-3:50p, probably in D209. In early October I'll be giving a presentation at the Oklahoma Supercomputing Symposium and will miss one day of class. In mid-November I hope that many of us will be going to SC07, the SuperComputing Conference, together for six days (see *Field Trip* below).

Class participation is showing-up to each class on-time and prepared, doing all the work, and actively engaging your fellow students and myself in the enterprise of learning.

Exercises

There will be one or two exercises each week, about half of them will come from the text and half of them from other sources. Periodically we'll have short in-class quizzes, typically on terms and definitions, which will also count as exercises. Exercises are to be done individually.

Often exercises will be a way of learning a new software tool or library, some of the packages we'll use in this course include:

- Octave

- Message Passing Interface (MPI)
- GNUplot
- GSL
- GNUmp
- Goto's BLAS
- Fastest Fourier Transformation in the West (FFTW)

Labs

The labs for this class will center around learning the principle techniques of scientific computing, included here are topics such as:

- Pseudo-random numbers
- Linear algebra systems
- Genetic algorithms
- Monte Carlo simulations
- Agent simulations
- Cellular automata
- Fourier transformations
- Simulated annealing
- Validation and verification

Typically labs will involve software, possibly some real-world scientific hardware, data collection and analysis, and a write-up. Labs will be done with a lab partner, early in the semester we'll pair-off and barring catastrophe we'll stick with those through the semester.

Library Research

Towards the end of the semester we'll do a library research project. This will be an opportunity for you to familiarize yourself with the electronic and print (yes, paper!) resources available in scientific computing, computational science, and computer science. For those of you that don't already know LaTeXthis assignment will serve as your introduction to it.

Each of you will choose a topic from a list developed by Sara Penhale, Mary Bogue, and me. The list will include specific types of scientific computing done in geoscience, physics, biology, chemistry, environmental science, and mathematics. You will be responsible for preparing a summary of the underlying science, a description of the principle model(s), a description of commonly employed the simulation technique(s), available software, etc.

Field Trip

As part of this class some number of us will hopefully be taking a field trip to the SuperComputing Conference's (`http://sc07.supercomputing.org`) SC Education Program (`http://sc-education.org`). We'll leave campus on November 9th and return on the 14th. This is a 4 day program designed for groups of undergraduate faculty and students to learn about high

performance computing and computational science, there will be about 125 people in the program. I'm on the steering committee for the SC Education Program for 2007-2009 and one of the (many) instructors for it. In addition to participating in the SC Education Program you will have the opportunity to help with some of the technical logistics (on-site networks, servers, and clusters) and enter a team in the scientific programming contest.

The SuperComputing conference isthe largest scientific computing conference in the world, drawing over 10,000 attendees from academia, national labs, and industry each year. It'ssponsored by the ACM and the IEEE Computer Society, the two academic professional organizations in computer science. Since they are non-profit entities they use the "profit" from the conference each year to support the next year's SC Education Program.

I'm working with the SC Education Program to make sure that they can cover travel, lodging, food, and the conference registration for all of us that are interested in going. We're a large group compared to what they usually have enrolled, I hope to know for certain by the beginning of September what the exact financial arrangements will be.

Bonita Washington-Lacey sent me this note when I told her about the trip:

> This sounds like a wonderful opportunity for our students. Please revisit the statement on "approved absences for field trips" and remind your students to inform their faculty as early as possible in the fall of their intent to attend this conference. The students should also be reminded that they remain responsible for all academic work during that time frame.

Please tell your other professors, advisor, and coaches if you have any, about this trip *now* if you are thinking of going. You should remind them again at the beginning of October and at the beginning of November. It's very important that they know the dates and educational purpose of the trip. You will have time to keep-up, to some extent, with your other classes while we're gone.

Grading

Each of the components of the course are weighted as follows:

1. Exercises	40%
2. Labs	30%
3. Library Research	20%
4. Class participation	10%

Academic Honesty

Often you will find it useful to discuss specific problems, techniques, etc. with tutors and fellow students. The sharing of ideas is a helpful and normal part of learning, and is encouraged. In particular one of the best ways to really learn something is to teach it to other people.

However, it's also possible to abuse those resources and turn-in work that isn't your own, particularly in computer science classes. See the Academic Integrity Policy in the Curriculum Guide for the official Earlham College policy,
`http://www.earlham.edu/curriculumguide/academics/integrity.html`

Disabilities

Please let me know as early in the semester as possible if there are any adaptations or accommodations you require, if there is any emergency medical information I should know about, or if you might need special arrangements in the case the building needs to be evacuated. The Earlham policy is:

> *Any student with a documented disability (e.g., physical, learning, psychiatric, vision, hearing, etc.) who needs to arrange reasonable accommodations must contact Academic Support Services and the instructor at the beginning of each semester. Accommodation arrangements must be made during the first-two weeks of the semester.*

It is important to follow this procedure.

Mantra

"Work hard, learn lots, and have fun." Ray Ontko, *circa* 1999.