PHYS 610: Computational Physics (Spring 2011)

Instructor: Daniel A. Steck
Office: 277 Willamette  Phone: 346-5313  email: dsteck@uoregon.edu
Office hours: walk-in anytime or by appointment (best to email first)
Course home page: http://atomoptics.uoregon.edu/~dsteck/teaching/11spring/phys610

Schedule: MWF 10:00-10:50, 318 Willamette
Course reference number: 36932
Credits: 4
Prerequisites: none

Links:
- news
- course notes
- homework
- software resources

Course overview

This course will provide a broad overview of computational methods in physics, including: numerical analysis, coding in several programming languages, code performance optimization, and parallel computation. No prior programming experience is necessary, though with very little experience you may need to work hard in the early part of the term.

Programming is something that can only be learned by doing. Furthermore, computational physics is fairly close to experimental physics, and this course will operate as something of a lab course. A significant part of the classroom time will be devoted to programming exercises. (At the beginning of the term we will also discuss rescheduling classroom time in an evening, 2+ hour block, to avoid time constraints on in-class programming.) The in-class programming component will involve programming in groups of 2 or 3. You will turn in your own writeups and codes each week. To facilitate this, you should be prepared to bring a laptop computer to class (see the Computer Access section for details).

To avoid getting bogged down in the details of programming languages, most of the programming will involve understanding and modifying instructor-supplied example codes. One of the most important aspects of becoming a competent computational physicist is building up a “bag of tricks”—a library of examples that you can refer to whenever you attack a new problem.
Topics in numerical methods to be covered in class will include numerical accuracy and stability, linear algebraic methods, and methods for solving ODEs and parabolic PDEs.

The primary language for this course will be Fortran 95, with some programming time also devoted to Perl, shell programming, Octave, and possibly also C.

---

**Computer Access**

You will be given access to the qc cluster, where you will have ample resources to perform your calculations, and even to do high-performance, parallel calculations for your term project. All interactions with the cluster are via remote login; see the cluster page for details. You will have individual accounts, and I recommend you keep your default umask set to 022 so you can share code and data with others in the class (in particular, your teammates). This is the “primary” computer for the course in the sense that you should store the final versions of your codes there. You should also post your term-project writeup with any animations etc. on a web page on the cluster.

You should also use the cluster for in-class programming, by logging in via your laptop. Your laptop should therefore have wireless network access (set up for the UO Secure network).

The unix environment is the de facto standard for high-performance numerical computation, and the programming examples will be for unix. Your codes must also compile and function under unix, although you can of course run the codes wherever you like. For remote access to the cluster (say, from your personal computer), you will need a terminal application that supports secure shell (ssh), as well as an X Window server. These are all included on unix/linux and Mac OS X systems, but need to be installed on Windows systems. Contact the instructor for assistance with remote access.

---

**Grades**

Grades for the course will be based on weekly assignments (much or most of which you will do in class) and an individual term project. The relative weights will be as follows:

- Weekly assignments: 50%
- Term project (40% writeup, 10% in-class presentation): 50%

**Weekly assignments:** there will be programming assignments each week, to be started in groups of 2 or 3 students in class. You should turn in an *individual* writeup of your assignment at the beginning of the next week. This includes printouts of your codes and relevant output, and commentary on your results. In-class groups will be assigned by the instructor based on the skill levels of the class members, to keep a mix of experienced and novice students. Groups will rotate each week.
**Term project:** The term projects can be on any topic in computational physics (topics related to your research are okay), but the topic must be approved by the instructor. The instructor can suggest problems for term projects, if necessary. By Friday, 19 April, you must turn in a one-paragraph summary of your intended term project, and *by this point you should have cleared the topic with the instructor.* You will present your project results in a short (~10 min) class presentation. The writeups are due the following week, by noon on Friday, 10 June (during the nominal final-exam slot). As for the assigned projects, writeups for the term projects must be typed (preferably in LaTeX), submitted to the instructor on paper, and posted online.

**Pass/fail grading option:** does not make sense due to the cooperative nature of the course work. The level of effort required to pass is essentially the same as the effort required for a good grade.

---

**Syllabus and Calendar**

*The following is a tentative list of topics to cover. This schedule will change depending on the progress of the course.*

<table>
<thead>
<tr>
<th>Monday</th>
<th>Wednesday</th>
<th>Friday</th>
</tr>
</thead>
</table>
| 28 March  
Planning class;  
Floating-Point Numerics and Data Processing | 30 March | 1 April |
| 4 April  
Linear Algebra and Root Finding | 6 April | 8 April |
| 11 April  
ODEs: Initial and Boundary-Value Problems | 13 April | 15 April |
| 18 April  
Performance Optimization | 20 April | 22 April |
| 25 April  
Elliptical PDEs | 27 April | 29 April  
Term project summaries due |
| 2 May  
FFTs  
Class to be rescheduled: CLEO | 4 May  
Class to be rescheduled: CLEO | 6 May  
Class to be rescheduled: CLEO |
| 9 May  
Parabolic PDEs | 11 May | 13 May |
| 16 May  
Parallel Programming:  
OpenMP, HPF, MPI | 18 May | 20 May |
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>23 May</td>
<td>Minimization and Optimization</td>
</tr>
<tr>
<td>25 May</td>
<td></td>
</tr>
<tr>
<td>27 May</td>
<td></td>
</tr>
<tr>
<td>30 May</td>
<td><strong>No Class:</strong> Memorial Day</td>
</tr>
<tr>
<td>1 June</td>
<td><strong>Term project presentations</strong></td>
</tr>
<tr>
<td>3 June</td>
<td><strong>Term project presentations</strong></td>
</tr>
</tbody>
</table>