Physics 411: Computational Physics

Winter 2014

Room: 1250 USB
Time: TTh 10–11:30am

Instructor: Mark Newman
Email: mejn@umich.edu

Office: 322 West Hall
Office hours: Mondays 1:30–3:30pm

Web site:

Everything on this information sheet, and more, including a complete course schedule, can be found on the course web site:

`http://www.umich.edu/~mejn/courses/2014/phys411/`

Description: This course gives an introduction to the solution of physics problems using computers. Assuming no previous computer programming experience, the course will introduce the basic ideas and programming skills of computational physics and students will develop their own computer software to solve problems in quantum mechanics, atomic physics, astrophysics, condensed matter, nonlinear dynamics and chaos, biophysics, and other areas.

The course will use the Python programming language, a powerful modern language invented in the 1990s, which finds wide use in many different fields, including physics.

Textbook: *Computational Physics*, M. Newman, Createspace Independent Publishing, Seattle, WA. ISBN 1480145513. We'll cover pretty much the entire content of this book during the semester, as well as many of the exercises.

I know of no other textbooks on computational physics using Python, but there are several good books that make use of other languages. If you want a second opinion on a particular topic you might like to consult one of the following:

- *Computational Physics: Problem Solving with Computers,* Rubin H. Landau, Manuel J. Páez, and Cristian C. Bordeianu, Wiley-VCH, Weinheim (2007). This book covers most of the material in the course, plus some additional material as well, and it's clear and well-written. Its disadvantage is that it focuses largely on the math and computer science and doesn't discuss actual physics very much. It uses the Fortran and C programming languages.

- *Numerical Methods for Physics,* 2nd Edition, Alejandro L. Garcia, Prentice Hall, Upper Saddle River, NJ (2000). This book is at a slightly higher level than the book by Landau *et al.*, and has a good selection of physics applications, but it has an odd balance of material. A large fraction of the book is taken up with discussion of differential equations, and it's rather light on everything else. Programs are presented in C++ and Matlab.

- *A First Course in Computational Physics*, Paul L. DeVries and Javier E. Hasbun, Jones & Bartlett, Burlington, MA (2010). This book has a good balance of material and some nice examples and exercises, but it's written in a quirky and sometimes difficult style, and skips over a lot of the details, leaving you to work out many things for yourself. Programs are written in Matlab.

- *Numerical Recipes: The Art of Scientific Computing,* William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, 3rd Edition, Cambridge University Press, Cambridge (2007). This is not a textbook; it's a reference work. But it's extremely well written and it is the classic book on numerical methods. It's not a physics book *per se*—it's a general book about computer methods for the sciences—but it explains the math in clear terms and it's very practical. Every professional physicist has a copy on their bookshelf. It is available in versions that use several different computer languages, including C and Fortran, but not Python.

Coursework: There will be weekly graded **problem sets** consisting mainly of programming challenges. Classroom time will be spent primarily on learning the principles and techniques behind computational physics methods; the homeworks are your chance to put those principles into action. A typical exercise will describe a physics problem that can be solved using a method studied in class and ask you to write a program to solve it and present your results.

Collaboration is allowed when doing exercises, but you must write your own programs and turn in your own work. Direct copying from others is not allowed. You may use example programs from the textbook as a starting point for your work, but copying from any other source, including books or the Internet, is not allowed.

There will be four **take-home midterms** during the semester, which will consist of programming challenges similar to those on the homeworks. The exams differ from the homeworks in that **collaboration is not allowed**. All exam problems must be solved without consulting others or copying from any source, except again that you may use programs from the textbook as a starting point. The dates of the midterms are listed on the web site. There will be no other homework in the weeks when there are midterms. There will be **no final exam** for this course.

There are **reading assignments** for each lecture. The assignments are listed on the schedule on the web site. Students are expected to do the reading for each lecture in a timely manner.

Grading: The course grade will be 40% on the homeworks and 15% on each of the midterms.

Computer facilities: The course will require you to do a significant amount of programming work on your own time. One of the advantages of the Python programming language is that it is available for free, which means that if you have a computer of your own you can download a copy of Python and work on your own computer. Instructions for installing Python can be found in Appendix A of the textbook. Most people should install version 3 of Python, which is the most recent version, but you can also use version 2 if you wish. (Some Mac users, particular those with older Macs, have reported that version 2 works better with their computers.) If you use version 2, then for compatibility you need to include the following line at the start of each program you write:

```
from __future__ import division,print_function
```

Note that there are two underscore symbols before the word `future` and another two after it. For further details about Python versions, see Appendix B in the textbook.

If you have difficulty installing Python on your computer, technical support is available from the physics computing support office on the second floor of Randall Lab.

If you do not own your own computer or you do but you prefer not to use it, computers are available in the Physics Department's Physics Authorized Users Lab (PAUL) on the second floor of Randall Lab (vertically above the physics help room, which is on the first floor, and next to the computing support office). Both Windows and Mac computers are available in PAUL, but only the PCs have the Python language installed, so you should choose a PC if you're going to work there. PAUL also provides printers that you can use to print out material to be handed in. Even if you use your own computer for doing homework you can still use the PAUL printers for printing out your work.