Physics 305: Computational Physics

Fall 2009

Section 1: Room 272 Physics (PAS) building

	Prof. Doug Toussaint
Telephone:	621-2554
Email:	doug@physics.arizona.edu
Office Hours:	PAS room 386A Thursday 2:00-3:00 (PAS 386A) or by appointment

	Walter Freeman
Telephone:	xxx-xxxx
Email:	wfreeman@physics.arizona.edu
Office:	PAS xxx
Office Hours:	XXX and by appointment

Introduction

This course is an introduction to the use of computers as a tool for answering scientific questions, especially in the physical sciences. By the end of this course, we hope you view the computer as a fundamental tool to which you will often turn in your course work and research.

Prerequisites

No previous experience with Unix or programming is required, but it is expected that everyone to learn C and a bit of Unix early in the course. If you are new to this, you should plan to invest extra time early in the semester as you get up to speed. Ask questions either to the instructors or to your classmates; computer skills are often best learned in an informal manner.

You should be fluent with first-year calculus and physics. Experience with ordinary differential equations and linear algebra will be useful. MATH 254 is required, at least as concurrent registration.

Texts

Each week a packet of notes will be handed out in class. These notes will contain most of information discussed in lecture, contain homework assignments and contain useful information related to UNIX, C and the packages we will be using in this course.

The only recommended book for this course is a reference for the C programming language:

- "The C Programming Language", 2nd edition by Kernighan & Ritchie, (1988, Prentice Hall)

  Be sure your copy is the 2nd edition; the 1st edition is different in important ways.

Other References

- "C in a Nutshell" by Prinz & Crawford (2005, O'Rielly Media)
- "A First Course in Computational Physics" by P. DeVries, (1994, John Wiley & Sons)
- "An Introduction to Computer Simulation Methods: Applications to Physical Systems," 3rd edition, by H. Gould & J. Tobochnik (2006, Addison Wesley)
- "Numerical Recipes in C, Second Edition" by W. Press, S. Flannery, W. Teukolsky and B. Vetterling, (1992, Cambridge University Press)
- "Numerical Recipes, Third Edition" by W. Press, W. Teukolsky B. Vetterling, and S. Flannery, (2007, Cambridge University Press), with or without the CD-ROM of the programs

There are many books available on C and Unix.

Grading

Your grade will be based on the following ingredients: Homework 60%, In-class Quizes 15%, Midterm Project 10%, Final Project 15%. The lowest grade for a homework set will be dropped.

The midterm project will be due Monday, October 19. The final project will be due Wednesday, December 9. These projects will be normal homework sets. The difference here is that you must complete them strictly independently from other students, as you would a take-home exam.

This is not to say that the normal homework is to be treated as a group project. We expect you to work independently. However, it is acceptable to discuss the methods with your classmates and to try to help one another if you are stuck (or come talk to the instructors). What one must avoid is the situation in which you and your classmates write your codes together, comparing as you go along. In past years, we have seen cases in which multiple codes are so close that the students were clearly looking over each other's shoulders while writing. This will be penalized and is considered a violation of the code of academic integrity. Please talk to you professor if you have questions about this.

It is of course the case that copying computer codes from books, the web, or solution sets from previous years is not allowed and will be considered a violation of the code of academic integrity. We will explicitly distribute code that you are allowed to use.

There will be 6-8 in-class quizzes given during the term. These will not be announced ahead of time and will be limited to 15 minutes. The quizzes will cover recent lecture material and course concepts.

There is no final exam for the course and no work will be due after Wednesday, December 9.

You may always consult your classmates about how to use Unix or the text editors; these are tools that we are using, but they are not intended to be graded. Similarly, you are free to make graphs by any method. However, the animations we will do later in the course will need to be done with the supplied graphics package and this is a part of your grade.

There will be a heavy level of coordination between the two sections of PHYS 305, taught by Profs. Toussaint and Eisenstein respectively. You should interact with students in the other section by the same rules as you do students in this section.

Students requiring accommodation in testing or note taking must notify your section's instructor and deliver to him the Disability Resource Center faculty letter within the first few days of the Course.

Homework

Homework problems will be usually be assigned every Tuesday and will be due by 10:00 pm on the Monday following.

Note: We will accept late homework up to the two days after their due date and they will only be worth 50% of full credit. Homework is normally due on Monday night at 10 pm so late homework will only be accepted till 10 pm the following Wednesday. A week is plenty of time to complete all assignments. Computer or network failures, especially, are no excuse — get it done early or risk not being able to get it done at all! If there are extenuating circumstances, arrangements should be made with the instructor **before** the homework is due.

The correct answer matters! If you hand in code which clearly doesn't do what it is supposed to do, *let us know that you know this is so,* and at least speculate on what kept you from getting the correct answer. Don't try to foist garbage on us in hopes that we won't notice — we will!

The correct answer is required but not sufficient! A big part of this course is making *sure* that the answer you get is correct and providing evidence of its correctness.

Since the purpose of this course is to learn about the entire process of solving a problem, your homework submission is not just a computer program. Instead it is a report, somewhat like a lab report in your earlier classes. Computer programs, graphs, and possibly tables of output are parts of this report. Exactly what the report contains will depend on the problem, but here is a list of likely ingredients:

- What is the problem, and why is it important?
- What algorithms were used?
- What codes did you write? You should send complete computer codes as attachments to your email, and you will probably need to embed fragments of code in your discussion.
- How did you test your code? This is essential - why should we believe your answers? There are many ways to do this, but testing cases where you know the answer is the most common one. For example, in the planetary orbits assignment, do you get a circular orbit for the expected initial conditions?
- What results did you get? This includes discussion of how well your algorithms performed (did you see the expected dependence of error on step size, for example) and the physics answers.
- What did you learn about computational methods? What did you learn about physics?

Style matters. Your grade depends not only on whether your computer code compiles and runs, but on our judgment of how well you understand the material. So a clear, coherent and complete explanation is important.

It often happens that we will need to compile and run your programs, sometimes to make sure they work and sometimes to fix some small error and see if that helps. To help use do this, please use a header comment in each source code file telling us how to do this. In any event, this is always good practice. A suitable header might look something like this:

```
// midpoint.c   -- integrate a function by the midpoint rule
// Joe Student,   9/10/2009
//
// Use the midpoint rule (class notes section 99.2) to
// integrate a function.
// the function is named "myintegrand()" and is in a separate file.
// to compile:   cc -o midpoint midpoint.c myfunction.c -lm
// to run: midpoint   (prompts for input: integration limits and
// number of bins )

\#include <stdio.h>
...
```

All work *must* be submitted by email to the account `p305@physics.arizona.edu`. Further description of this procedure will be handed out.

Code of Academic Integrity and other Policies

This class will follow the University of Arizona policies listed at the web-site:
http://policy.web.arizona.edu/~policy/

The UofA Code of Academic Integrity can be found by clicking "Index" and "Academic Integrity" or the web-site:
http://dos.web.arizona.edu/uapolicies/cai1.html

Changes to Syllabus

Information contained in the course syllabus may be subject to change with reasonable advance notice, as deemed appropriate by the instructor.

Web Site

http://www.physics.arizona.edu/~doug/phys305.html

Tentative Topics

- Introduction to basic tools and programming: Unix shell, Email, using a programming editor (EMACS), basic C programming.
- Numerical Integration: left-hand, midpoint and trapezoid rules, Taylor series expansions, truncation and round-off errors, arrays, functions.
- Root finding methods: bisection and Newton's Method, one dimensional minimization via Newton's method.
- Ordinary differential equations: Euler, Runga-Kutta, and other methods; errors in solutions; solutions of coupled, second-order, and higher-dimensional problems; adaptive time steps; applications to N-body problems.
- Using computers to explore chaos.
- Linear Algebra, Maximum Likelihood Statistics, Eigenvalue problems.
- Monte Carlo Methods: sampling from distributions, direct simulation.
- Molecular Dynamics Simulations.