Comptuational physics: syllabus and goals for the course

The goals for this course are to develop in you the ability to write simple computer programs to solve physics problems that require computers (or which are much more easily solved by computers, in the case of things which you can do with paper and pencil, but which must be done in a repetitive manner). In order to do this, we will learn some basics of the Linux operating system and the C programming language.

We will aim to get you to the level where you don't need any more formal programming classes unless you want to use advanced techniques like parallel programming. This is not the same as saying that we will get you to the level where you are fully proficient in all types of programming in C. That takes more practice than would be reasonable to expect in a one semester class. One you reach a certain level, however, you should be comfortable with looking things up in a book or on google, and trying them out when you are confronted with situations where you don't know exactly what to do next.

The other main aim of the course is to teach you a bit of numerical analysis. Like with programming, in the course of a semester, we cannot do everything. Additionally, some types of numerical analysis involve having a higher level of mathematical background than what is in just the prerequisites for this course. Thus in many cases, I will mention some advanced topic by name in the notes, but I will not teach it in detail. My goal with this is just that if, several years from now, you are faced with a problem you don't know how to solve, but that sounds like a more challenging version of what you did in this course, there might be a chance that you can look up a buzzword in my lecture notes, and then investigate the topic yourself in detail.

We will cover the following topics and solve the following computational problems:

1	August 26-30	Why we do computational physics
2	September 2-6	Some basics of Linux and gnuplot
3	September 9-13	Basic ideas of programming, and initial programs in C
4	September 16-20	Pointers, functions, and arrays in C
5	September 23-27	Fourier analysis
6	September 30-October 4	Root finding
7	October 7-11	Integration I: Simple numerical integration schemes
8	October 14-18	Integration II: Simple differential equations
9	October 21-25	Integration III: Monte Carlo integration
10	October 28-November 1	Curve fitting
11	November 4-8	Make-up time
12	November 11-15	Final project lab sessions
13	November 18-22	Final project lab sessions
14	November 25-26	Final project lab session
15	December 2-5	Final project lab session

The use of week 11 as "make-up time" is to allow for the possibility that some topic may take a little bit longer to teach than I have anticipated. If all goes well, we'll cover another topic, with an unassessed assignment. For the first 10 weeks' lessons there will be assignments due one week after the last lecture on the topic. In each week, there will be one lecture and one lab session. I will be present at the lab sessions to give you help with the assignments. I strongly recommend that you make a try at the assignments before the lab session, so that then you can approach me quickly with well-targeted questions, after you have seen where you have gotten stuck.

Homework assignments

Week 1: None
Week 2: Making simple plots
Week 3: Simple algorithm development – long multiplication
Week 4: Making a calculator
Week 5: Find the pulsar; essay due
Week 6: Solving polynomials and other functions
Week 7: Solving integrals: I
Week 8: How far does a baseball fly?
Week 9: Hyperspheres
Week 10: Modeling of data
Week 11: No assignment
Week 12-15: Projects from the list of projects, to be given out soon

The final project will be due on December 9, which is during finals' week. There will be a range of choices for the final projects.