Physics 261: Computational Physics I — Syllabus

Paul A. Nakroshis

Sept 2011

Physics 261: Computational Physics I is an introductory course on scientific programming using the Python programming language. We will also learn how to use LaTeX for typesetting your reports, as well as several numerical and scientific graphics toolkits. By the end of the semester, you should feel comfortable using Python to solve everyday problems, from data analysis to numerical simulation.

Introduction and Background

Suppose you take an introductory calculus-based physics course; you will have learned all about Newton's laws of motion and solved many problems with an analytical approach. However, one problem you will not have solved is a simple problem of throwing a ball while including air resistance. One of the reasons you didn't solve this problem is that it's impossible to solve analytically, since air resistance is a nonlinear force[1] However, adding air resistance which is proportional to the square of the particle's velocity, while impossible to solve analytically, is not so complicated to solve computationally. So, one of the great uses of computation is to be able to solve problems via computer that are difficult or impossible to solve with pen and paper. Computers are also incredibly useful at processing and visualizing data, and, accordingly we will introduce you to the numerical python library (numpy) and Matplotlib (useful for visualizing data and functions).

[1] Actually, it's even worse—air resistance is not really a simple function of velocity at all; for low velocities, one can approximate the air resistance as linear, and but as the speed increases, it's not even correct to treat it as a simple function of the velocity to a fixed power.

Textbook

A Primer on Scientific Programming with Python, by Hans Petter Langtangen
This is the only required textbook for the course, and we will make extensive use of it, from readings to homework problems. It is available at the Portland USM bookstore, or online through Amazon.com or many other booksellers.

Atendance Policy

I expect that everyone will be at every class except in extenuating circumstances. If I find that you are missing class too often (i.e. more than three times), you can expect that I will talk with you and that you will likely receive a lower grade for the course, or asked to leave if this repeated absence is coupled with poor quality work written work.

Outside Help/Office Hours

In general, if my office door is open, I am happy to help you, so feel free to stop in and ask questions. I have set aside several hours where I will make a point to be in my office. Please take advantage of my willingness to help you! I can be much more effective one-on-one than I can in front of the whole class.

Grading

I am unfortunately contractually obligated to submit a grade for every student. Grading is subjective, and I do my best to submit a grade which represents my sense of your level of understanding and your level of improvement in understanding for the course. You will get regular feedback on each report, as well as on your final written report. In attempt to make this quantitative, Table 1 shows the grading scheme (note that late reports lose 25% per day late) for the course:

Item	Points
Attendance +Perceived Effort	50
Assignments & Reports*	700
Final Project	250
TOTAL	1000

Table 1: Grading Scheme for the course. Late reports lose 25% per day late.

General structure of the course

We will start with several weeks devoted to becoming familiar with some of the tools we'll use on a daily basis. We'll get you up and running with a Python installation on either Linux or the Mac OS X (you're one you're own if you use windows, though I can give some pointers). We'll also install a working LaTeX distribution.

After your computer is configured, we'll introduce the typesetting package LaTeX , which practically all physicists use on a regular basis (if not, they should!). This will be a typesetting tool that you will be able to use for all your typesetting needs for the rest of your career (probably!).

Then we will embark on an introduction to Python, and see how to do a few basic everyday tasks, such as plotting simple figures and reading and writing data files. These first few weeks will involve you completing several short assignments to get you familiar with LaTeX and Python.

After this, we'll go a little bit more in depth into formal pieces of the Python language, and the rest of the semester will be devoted to the *Physics* in Computational Physics. We will learn how to solve ordinary differential equations by numerical integration, how to solve some physics problems that we avoided in freshmen physics, and move on to other interesting systems involving random processes. The bulk of your work here will be in composing programs, running simulations and writing formal reports on your investigations.

The capstone for this course will be your final project. Your final project will be 25% of your grade and will have multiple components as seen in Table 2. I've had to learn from previous incarnations of this course to start the final project early. I've also broken down the final project into 4 milestones; the Proposal, two Project Updates, and the Final Report. For this final project, these dates are immutable, and I won't take late submissions. So, put these dates on your physical and mental calendars and plan accordingly.

Item	Due Date	Points
Project Proposal	27 Oct 2011	50
Project Update 1	10 Nov 2011	25
Project Update 2	1 Dec 2011	25
Final Project	14 Dec 2011	150
TOTAL		**250**

Table 2: Grading Scheme for the final project

Items to include in Formal Reports

Introduction - The introduction should give an overview of the problem and an indication of where it fits into the subject of physics.
Physics & Numerical Method - describe the background physics of the problem, and detail the algorithm that is used to solve the problem. This section should also list relevant snippets of your code to show how it is implemented. A full listing of your code should always be attached as the last section of your report.
Verification - tell what you did to verify that the program gives correct results; this typically involves showing that your code gives reasonable results for simple cases where an analytic solution is known or obvious. Generally speaking there should be more than one test used to verify program integrity.
Results - present the results of running the program to demonstrate the behavior of the system under different circumstances. Results might be presented in graphical form or as tables, as appropriate. Be sure that results that are presented are labeled properly, so that the

reader can figure out what has been calculated and what is being displayed. Make sure that all figures and tables should have descriptive captions.
Conclusion - present a discussion of the physical behavior of the system based on your simulations and answer any special questions posed in the assignment.
Code - Always provide a full listing of your code at the end of the paper. In LaTeX, there is an excellent package called listings that does an excellent job of formatting code.

This document

Should you lose this syllabus, an electronic .pdf version of this file (with clickable hyperlinks) is available online at the course homepage which can be found at
http://people.usm.maine.edu/pauln/physics261.html.
Information about other physics courses can be found at the Physics Department Homepage.