Comptuational physics: grading policies

For this course you will hand in 9 short, homework-length computer-work assignments, one short essay, and one project-length assignment.

In each week, there will be a main assignment and a challenge problem. To pass the course, you must complete all the main assignments successfully. This means getting the right answers to the problems using a computer program you have written yourself. I will not give any D's in the course, except possibly in the case where a student completes all the regular assignments and turns in a very poor final project; if you successfully complete all the assignments, you will get at least a C-, while if you do not successfully complete all the assignments, you will get an F or an incomplete, depending on the circumstances of why you have not completed the assignments.

You can then boost your grade from a C- for the bare minimum job all the way up to an A+ for a truly exceptional job by doing the following:

1. Turning a well-written pseudo-code (if you don't know what pseudo-code is yet, relax – you'll learn soon enough)
2. Writing a comprehensive and yet concise report on your work that you turn in with your code
3. Commenting your code properly
4. Writing good, structured code
5. Completing the challenge problem
6. Coming up with an idea of your own of how to extend the problem I have set you in a way that is not trivial

As one example of a way that you can always show some extra effort, you may try different approaches to writing the program that both work, and determine which one runs faster, or in some cases, which gives more accurate answers. The purposes of giving higher grades for doing the challenge problem or for coming up with an idea of your own are to encourage you to exercise some creativity by indulging your own sense of curiosity, and to get you in a position to learn more. At the same time, I don't want to have a course that students who make a good effort struggle to pass.

The purpose of these extensions beyond the basics are *not* just to give better grades to students who spend *more time* working on the project; they should be about going to deeper or broader levels of knowledge, and not just about doing some extra busy work. The challenge problems will be things that require you to give some more thought to the problem assigned. To earn an A+ on an assignment, you will have to come up with something on your own that is at least as difficult as the challenge problem and you will have to execute it properly (with a few exceptions, for cases where I assign a challenge problem that has some very difficult parts to it). If you receive an A+ in the course, it will demonstrate some curiosity and creativity, in a way that will make it

very easy for me to write you good letters of reference in the future, since I will have something specific to point to. Please bear that in mind if you find the course a bit easy (if, for example you already have some significant programming experience).

Applied Science and Technology requirement

This course counts for the core requirement for Applied Science and Technology, which will apply to those of you who started at Tech on or before Fall 2012. As a result, I need to give you an assignment related to ethics and technology. It will be to write a short essay about the free and open source software movements.

Collaboration in the course

I strongly encourage students to discuss programming and physics with one another as part of this course *but* you may not copy sections of one another's computer codes, and you should not be looking at another student's screen while you are working on your own code. You may look at another student's screen in order to get an idea of what they have done to make the program work, but then you should go back to your own program and make sure that you understand what to do before using the information you have just obtained. As a rule of thumb, make sure that you could re-do the assignment locked in a room by yourself.

Use of programs from the web and from books and journal articles

For the regular assignments for the course, you must write your entire programs yourself. You may use the programs I give you as a starting point, but you may not download programs from the web and modify them, and you may not copy code from your fellow students. These assignments are set up to drag you through the basics of scientific programming in C. If you use programs I give you, you should write your own pseudocode to explain how they work.

The final project is different – its goal is to ensure that you have some feel for how to do science with a computer program. Thus, once you get to the final project stage of the course, you should feel free to make use of functions that you take from any source you wish, but not to make use of full programs to do the job for you. If you use code from the web or a book, you must do the following: (1) add a comment to the function that states exactly where you found the code and (2) add comments throughout it that give me evidence that you understand how it works OR run a series of tests that show that it does what you think it is doing. The first is an issue of academic honesty, and while the second is merely good scientific practice. While it's important not to spend too much time re-inventing the wheel, it's also important to make sure that you're not using black boxes for vital things without being sure that they work.

I will even suggest that you may find the book *Numerical Recipes in C* helpful in particular. That book gives reasonably good explanations of a broad range of topics in numerical analysis, and it includes computer codes that work for many of these topics.[1] The book is available for free online. It represents

[1]The licensing rules for using the codes are fairly restrictive, so bear that in mind if you decide to use them. It turns out that the GNU Scientific Library has free version of almost all

its authors' best compromises between writing a book which is accessible and which advanced the state of numerical analysis in physics. There are better methods for a lot of things, but they are often hard to learn and implement and not always worth the effort – but you should bear in mind later in your career if you run into a tough problem that there may be better approaches in the applied and computational mathematics literature.

Late work and failed assignments

Work will be penalized by one full letter grade if it is up to one week late. It will be penalized by two letter grades if it is more than one week late. Any work that is turned in which successfully completes the assignment will be given a grade of at least a C-. If your first version of the assignment that you turn in is totally unsatisfactory, then you may re-submit after receiving some comments and get a C- grade on the assignment. All assignments must be received by three days after the end of lectures for the semester, or you will receive either an incomplete or an F for the course. You may choose to turn in all the assignments on that deadline if you are happy with getting a C- for a grade, and you are confident that nothing will have to be re-done, but I strongly recommend against that approach.

Since this is a relatively small class, I can be flexible about deadlines in the case of illness or serious issues in your personal life, but it will definitely be to your advantage to get caught up as soon as possible if you have such problems, since this is a course where we will continue to build on what we do throughout the course.

Distribution of points from assignments

The first 10 weeks of the course will each have assignments, except for the first week. One of these does not lend itself to being graded, and will just be a pass/fail assignment. I will then drop your two lowest marks on the eight assignments that are graded, as long as you pass all the assignments. In this way, you won't have to be overly stressed if you have midterms or other tests for some other course, and don't have time to complete the challenge problems every week. After that, these will all carry equal weight, and will, together make up 66% of your final grade. Then, there will be a final project which will be worth the remaining 34% of your grade. For the final project, because there will be no opportunity to re-submit, you will receive a passing grade for a good attempt at the problem that shows that you made some progress in solving it.

The essay will be graded pass-fail. That is, you must turn in an acceptable essay to pass the course, but beyond that, it will not affect your grade. It will be due toward the end of week 5, so that you will have some time to make revisions if I find your essay not to be acceptable.

(perhaps all) of the programs in Numerical Recipes, with the only licensing restriction being that you cannot then incorporate the routines into software that is sold without the right for people to copy and pass it on.