WHICH COMPUTATIONAL-PHYSICS COURSE SHOULD I TAKE?

The physics department currently offers two computational-physics courses, PHY 4151 and PHZ 5156C. Both are open to undergraduates, but they differ in focus.

**PHY 4151** In addition to the laboratory and theoretical competence expected of a physicist, computational physics must become part of the physicist’s toolkit. Many software packages are incapable of handling the types of problems confronted by physicists, and so a physicist must be able to express his problem in a way that enables computer solution, i.e., he must learn to think algorithmically. He must know enough programming to translate his problem into a suitable computer language, and must be aware of the ways that computers can introduce errors and distortions in order to be confident that the computer output is indeed the solution to his problem.

This is primarily a laboratory course although, as in all physics courses, there is necessarily an essential theoretical component. This course will teach you how to formulate your problem in a fashion suited to computer solution, write, debug and document a computer program. **PREREQUISITE:** Modern Physics (PHY 3101).

**PHZ 5156C** Many universities’ advanced “computational-physics” courses show how to solve cute textbook problems in 20 lines of computer code. Real research is not about 20-line codes. The present course has three goals: (1) to equip students with the computational tools they will need to work in scientific research; (2) to introduce some interesting areas of current research in computational science and engineering; and (3) to give students a high level of programming skill that will make them immediately employable.

After briefly reviewing some programming ideas and writing some code for digital signal processing and data analysis, we complete projects taken from neurobiology, celestial mechanics, mathematical finance, and magnetism. In the final project, students work as a team on a large-scale molecular-dynamics simulation of fracture run on a massively parallel computer. **PREREQUISITE:** CGS 5420 (Introduction to C and Unix) or comparable skill in a compiled language (e.g., C, C++, Fortran 90), including some level of comfort with pointers. In addition, students should have enough experience with upper-level science, engineering, or mathematics to enable them to understand the written notes (on the Web at [http://ewald.cas.usf.edu](http://ewald.cas.usf.edu)). We make use of the integral and differential calculus, of ordinary and partial differential equations, and of linear algebra.