Teaching Computer Simulation

2:30 pm, Wednesday, August 4, 2004

AAPT Meeting, Sacramento, session FG05

Harvey Gould, Clark University
<stp.clarku.edu>

Collaborators:

Jan Tobochnik, Kalamazoo College

Wolfgang Christian, Davidson College

Kipton Barros, Clark University

Natali Gulbahce, LANL

Joshua Gould, MIT

support: National Science Foundation DUE-0127363

Punchlines

- Teaching computational physics in a laboratory context creates an active learning environment.

- 3rd edition of *Introduction to Computer Simulation Methods*, Harvey Gould, Jan Tobochnik, and Wolfgang Christian. Drafts at <sip.clarku.edu/3e>.

Nature of Course

- Computer simulations provide an opportunity for students to do open-ended projects in a way closer to the way research is done. The course is an excellent predictor for how graduate students will do in research.
- The course is project oriented, with a minimum background in physics and no programming experience required. Students write laboratory reports, an excellent vehicle for improving their writing. In this context they learn LaTeX without difficulty.
- Converting an abstract model into a program and obtaining visual feedback makes the model and algorithm more meaningful.
- Numerical methods are more interesting in the context of a simulation.

- Simulations encourage open-ended questions and creative thinking in contrast to memorization and routine problem solving.
- Simulations allow a broader vision of physics than is found in most courses. Students can study models of interest to geologists, biologists, materials scientists, and social scientists. Because of the open-ended nature of the course, the course is only for motivated students.
- Open-ended laboratory courses are time consuming. The help of a TA is essential.
- Once students know how to write and test their own programs and know some important algorithms, they will use the computer in meaningful ways in other courses even if their instructors do not encourage it.

TextBook

- The text and course use Java. Good choices of programming languages include Python (objected oriented) and F (Fortran 90). The syntax of Java is introduced as needed.
- Open Source Physics library makes writing programs much easier.

Twenty-two chapters:

1. Overview
2. Introduction to Java
3. Inheritance and Interfaces
4. Input and Output: The Open Source Physics Library
5. Motion in One Dimension
6. Oscillations and Threads
7. Two-Dimensional Trajectories and the ODE Interface

8. Few-Body Problems
9. Three-Dimensional Visualization and Rigid Bodies
10. Chaotic Motion of Dynamical Systems
11. Random Processes
12. Dynamics of Many Particle Systems
13. Normal Modes and Waves
14. Electrodynamics
15. Seeing in Relativity
16. Numerical Integration and Monte Carlo Methods
17. Percolation and Clusters
18. Kinetic Growth Processes
19. Complex Systems
20. Monte Carlo Simulations of Thermal Systems
21. Quantum Systems
22. Epilogue