Integrating Computation into the Physics Curriculum*

Harvey Gould[1] and Jan Tobochnik[2]

[1] Department of Physics, Clark University, Worcester, MA 01610-1477, USA
http://physics.clarku.edu/~hgould
[2] Department of Physics, Kalamazoo College, Kalamazoo, MI 49006-3295, USA
http://cc.kzoo.edu/~jant

Abstract. Some of the challenges of incorporating computational methods into the physics curriculum are discussed. These challenges include teaching the methods of computational physics in the same spirit as we presently teach mathematics, changing the curriculum to reflect the new ways of thinking that have arisen due to the use of computer technology, and making use of the Web to make the process of teaching more like that of research.

1 Introduction

Every discipline has its own language that helps its practitioners communicate ideas efficiently with one another. The goal of much of education is to empower students to learn and use that language. Every discipline also uses and contributes ideas and techniques from other disciplines. For example, physicists use mathematics and computational methods developed by practitioners in other disciplines. Ideally, students take courses in many disciplines and can utilize ideas and skills they learn in one discipline and apply them to another. For example, biologists expect their students to learn chemistry from the chemists, and engineers expect their students to learn physics from physicists.

Because computation has become an important way of doing physics, it is important that physics students learn how to program and use computers effectively. Should we expect that students will learn how to program from computer scientists? How can we integrate computational methods into the physics curriculum?

2 An Analogy to Mathematics

The existing physics curricula already provides models for incorporating the methods of another discipline, mathematics, into physics education. It is useful to review the various approaches that have been adopted and analyze how well these approaches have worked.

* Supported in part by National Science Foundation grant PHY-9801878.

V.N. Alexandrov et al. (Eds.): ICCS 2001, LNCS 2073, pp. 1031–1040, 2001.

The usual approach is to require physics students to take a certain number of courses in the mathematics department. For example, introductory physics courses are frequently divided into those that use algebra and those that use calculus, and students in the latter courses usually take calculus as a co-requisite. Upper level physics courses usually have prerequisites such as linear algebra, vector calculus, and differential equations. How well does this model work? In general, the answer is not very well. A common complaint is that students learn very little from their math courses that is useful for physics. Sometimes it is argued that math courses spend too much time proving abstract theorems rather than teaching the mathematical tools that physicists need. However, our experience has been that even when this argument does not apply, there still is little carry over. The reason seems to be that even if students learn to use a skill in one context, it is difficult for them to use the same skill in a different context. Skills taught out of context are rarely learned very well. We note that the new accrediting requirements for undergraduate engineering education are based on the same conclusion [1]. Courses in physics are no longer required. Instead engineering departments can incorporate physics into their own courses. Most engineering departments have not taken this approach, at least not in the short term, because of the need for more staff, the large effort in time and resources necessary to change their courses, and the political obstacles.

Should physics departments drop the mathematics course requirements for a physics major and teach the mathematics ourselves? We suggest that if this approach were adopted, our students would be better able to use mathematics in physics and other application areas as well. Of course, we believe that students can learn important concepts from mathematicians. Students should take math courses for their intrinsic interest that are taught by mathematicians as if all the students in the course were math majors. Physics and other science students would then become more familiar with the language of mathematics, would understand the foundations underlying the methods they use in their physics courses, and would learn new ways of thinking that might give them a better understanding of physical phenomena. It would also allow physicists to communicate with mathematicians.

On a smaller scale the debate on how to introduce mathematics is repeated in many physics courses, particularly standard theoretical courses such as classical mechanics, electrodynamics, and quantum mechanics. For example, some textbooks and courses begin with a discussion of mathematical techniques. Other texts and courses integrate the mathematics throughout as needed. Even the same author might take both approaches. Griffiths has written two excellent and popular undergraduate texts. In his electrodynamics text he begins with a chapter on vector calculus [2]. However, in his quantum mechanics text he starts with the physics and introduces the solution of partial differential equations, ideas of probability, delta functions, and linear algebra in later chapters as they are needed [3]. We conclude that the best approach either depends on the material being covered or is a matter of taste.

Physics departments are aware of the difficulties that students have using the skills they learned in mathematics courses in their physics courses. These difficulties have led many physics departments to offer courses in mathematical methods for physicists. In this approach only those mathematical techniques that are useful for physicists are discussed, and the number and rigor of the mathematical proofs are reduced. The techniques are usually discussed with specific applications in mind. There is the obvious advantage of efficiency in this approach, but one limitation is that the physics context is not usually fully discussed because of lack of time. For various reasons, the majority of physics departments have not taken this approach.

A related issue is that many upper division physics courses have become very theoretical. For example, Griffiths' Quantum Mechanics text contains only one graph of experimental data. Admittedly, the text does discuss experimental data in the text and problems, but the presence of only one graph is telling. The lack of integration of theoretical and experimental physics is also a problem in the physics curriculum. The relative lack of experimental physics in the curriculum is driven in part by the fact that modern experimental equipment is very expensive and that creating good laboratory experiments is very time consuming.

The issues related to the integration of mathematics and experimental physics into the physics curriculum are relevant to a discussion of how to integrate computation. However, we have the advantage that we are doing something relatively new, and for the most part we do not need to undo an already existing set of courses. On the other hand, creating a whole new approach can be daunting.

3 Computer Simulations

We look to how the computer is used in physics research to determine what is important to include in physics education [4]. Many theoretical problems require numerical calculations such as the evaluation of an integral, obtaining the roots of an equation, and matrix manipulations. In some cases, such as diagrammatic calculations, symbolic manipulation is essential for keeping track of the different classes of diagrams. Computers are essential for the control of experiments and the collection and analysis of data, and are used in this context to implement statistical procedures for extracting a signal from noise, do fourier transforms to obtain power spectra, fit data to functional forms, and plot data in a way that helps to interpret its meaning.

These uses of the computer in scientific research can be viewed as applications of specific tools, much like an oscilloscope is used to make voltage measurements or a mathematical transformation is used to simplify a mathematical expression. In these cases we believe it makes sense to integrate the use of the tool with the physical application. Although there are numerical methods courses taught in mathematics departments and some computational physics courses focus on numerical analysis, we believe that such courses are not as effective for physics students as would integrating the specific tools directly into physics courses.

However, there is another way of using computers, known as *computer simulation*, that allows us to do physics in a fundamentally different way [5]. One distinction between computer simulation and numerical analysis is that in the latter way, the user does as much as possible analytically before giving the problem to a computer. In some ways, doing numerical analysis on a computer is an extension of techniques used before computers were invented. For example, in the past a human calculator would generate tables of sines and cosines; now a computer can generate these values very quickly. However, there are ways of doing numerical analysis that were inconceivable before the advent of computers. For example, the generation of high-order Feynman diagrams and the evaluation of the corresponding integrals using computational graph theory and Monte Carlo methods. The distinction between numerical analysis and simulation is one of emphasis.

Computer simulation is a way of doing physics that is distinct from the way physics was done before the advent of computers. In a computer simulation we might use various numerical methods and even some symbolic manipulations. However, the style and motivation is very different and is analogous to a laboratory experiment. For this reason computer simulations are sometimes referred to as computer experiments. We first develop a model that can be represented by an algorithm. The model plays a role analogous to the physical system of interest in an experimental system. Sometimes these models are formulated in ways that are related to traditional ways of doing theoretical physics. For example, a molecular dynamics simulation involves solving Newton's equations of motion for a large collection of particles. In other cases, the models are designed with the nature of digital computers in mind. Examples include cellular automata models of fluids and traffic flow.

Just as we need to calibrate a measurement apparatus, we have to test our program and compare its outcome with known results in limiting cases. Then data is collected and finally, the data is displayed and analyzed. In the real world, our initial data frequently leads to further improvements in the program and more data collection.

Computer simulations also have some features that are distinct from laboratory experiments. For example, in experimental physics each new kind of measurement requires a new piece of equipment and may preclude other measurements. In a computer simulation a new measurement requires only some additional code within the existing program. Simulations also allow us to determine quantities that cannot be measured in a laboratory experiment. Of course, simulations are limited by computer resources that limit the size of the system studied and the time that it can be simulated. In many cases these limitations are severe.

One advantage of analytical solutions is that a solution frequently can be written in terms of a parameter so that more than one case is readily available. In contrast, separate simulations frequently must be done for each value of the parameters of interest. However, an advantage of simulations is that many modifications of the model require simple changes in the program (for example, changing the force law in a molecular dynamics simulation), whereas even minor changes in a theoretical model can make an analytical calculation impossible. Analytical calculations usually require approximations whose consequences are not known. Simulations frequently use numerical procedures that are exact in principle, although the results are approximate due to statistical errors and limitations due to the effects of finite time and size. However, in many cases the nature of these limitations is not known.

Computer simulations have become an important part of research in physics and are increasing in importance in other fields of science. This importance is reason enough to incorporate them in the undergraduate curriculum. More importantly, computer simulations provide the easiest way of involving physics students in the process of scientific research. Writing and running simulations includes many of the aspects of scientific research such as model generation, testing, analyzing data, interpreting data, and drawing general conclusions. And the flexibility of simulations means that they can be done at many different levels for just about any field of physics as well as many fields outside of physics. Hence, we have much freedom in determining where in the curriculum to add computer simulations.

4 Integrating Computer Simulations into the Curriculum

We now discuss ways of integrating computer simulations into the physics curriculum. As we have mentioned, we believe that most other uses of the computer should be integrated directly into existing courses and discussed as needed. We will find that the same issues arise as we discussed earlier in the context of integrating mathematics. We will divide our discussion into the introductory curricula and upper division curricula.

Until recently, most attempts to integrate programming and simulations into introductory level physics courses have been unsuccessful. There are too many demands on students at this level. Most students still have difficulty with basic calculus concepts and many are not even adept in algebraic manipulations. In addition, we know from research in physics education that there are many fundamental concepts that students have much difficulty understanding. They come to physics with a view of the world that frequently is inconsistent with the physicist's world view. At the introductory level we must use Occam's razor and include only those uses of the computer that can enhance conceptual understanding. We cannot expect to teach computational science skills such as programming, algorithm design, and other ingredients that go into designing a computer simulation.

However, there is a noteworthy exception that is being developed by Ruth Chabay and Bruce Sherwood [6]. They are using the programming language Python in their introductory physics courses and are having students write sophisticated computer simulations with powerful three-dimensional graphics with only a couple of hours of instruction [7]. One of the keys to their success is that the graphics statements are largely hidden from the users who can add graphical objects to their programs with very little effort. Nevertheless, many of the ways of thinking that we would want to introduce when using computer simulations are present. So far, this work has been done with a relatively sophisticated group of students and experienced instructors at Carnegie-Mellon University, and it remains to be seen how large a population of introductory students can be handled in this way.

Another successful approach is the use of *physlets* as pioneered by Wolfgang Christian and his collaborators [8]. Physlets are Java applets built into Web pages using Javascript. The advantages of this approach include that there is a common user interface, the ability of instructors to tailor the physlets to their own needs, and a good set of questions that have been developed to go along with existing physlets. Students can use physlets to make plots and animations and collect data. The disadvantage of this approach is that in most cases students do not learn what is behind the simulation. However, we believe that physlets can help student learning as well as introduce students to the possibilities of simulations.

A commercial product, Interactive Physics, it useful for learning about mechanics [9]. Students can create various kinds of objects and forces between objects, and can collect data and draw graphs. Although many problems have been written for introductory texts using Interactive Physics, most of them are animated versions of traditional textbook problems and do not help the student learn much about more realistic situations nor understand why systems behave the way they do.

There are many other simulation programs available (for example through Physics Academic Software) [10]. However, none of these programs have as their goal the teaching of computational physics.

To summarize the situation for introductory physics courses, we believe that at present the only realistic goal for most instructors is to use already written simulations to help give students a visual and dynamic representation of the physical systems that they are studying. In this context it is very difficult to teach the kinds of thinking that goes into designing a computer simulation.

There are more possibilities in the upper level courses. We believe all physics students should learn how to do computer simulations. This involvement is probably the only realistic way for many undergraduate students to engage in an activity analogous to actual physics research within the context of a course. Laboratory experiments are frequently too costly and time consuming to have students plan an experiment, get the apparatus to work, collect data, and analyze the results. Such work is possible in a separate junior/senior lab, but is too difficult to do in most courses. The analogous process of doing theoretical research is in general too difficult for undergraduates. However, computers are inexpensive, readily available, and fast, so that every student can have access to a physics research tool.

One of the obstacles is learning a programming language well. For over fifteen years we have advocated that students' first programming language should have a simple and clean syntax that incorporates easy to use graphics statements and is platform independent. We began with True Basic, which has worked very well and serves as an excellent introduction to Fortran, C, and other procedural languages. However, because of the advantages of object-oriented programming and the popularity of Java, we are now using and recommending Java. In collaboration with Wolfgang Christian of Davidson College, we currently are writing the third edition of our computer simulation text in Java [11].

How can students learn a programming language? This issue is similar to the questions we raised for mathematics. Although, physics students frequently take courses from the computer science department, our experience is that the introductory programming courses are not very effective. Also many issues that are important in simulations are of much less interest to computer scientists. More importantly, it takes too many computer science courses before students become sufficiently proficient to write computer simulations on their own.

How can we expect students to learn programming while they are learning physics? The answer is that we can focus on those parts of the language that are useful for doing computer simulations, and we can provide templates and other utilities that the student can use. Also, because there is a context that provides a meaningful reason to write a program, physics students have a higher motivation to write a program that works and does something useful. In fact, most of us learned how to program without taking any courses, just as every child learns how to speak without taking a course in public speaking. Just as we urge students to take mathematics courses to learn mathematics on its own terms, we should urge students to do the same for computer science. There are ways of thinking in computer science that provide a foundation to the tools we are using and that will become more important in computational science in the future. However, we should not rely on computer science courses as a prerequisite for learning to do computer simulations.

We are still confronted with the question of whether we should teach a separate computer simulation course or integrate computer simulations into other physics courses. In most cases we believe that physics departments should do both. The motivation for a separate course is much the same as for a junior/senior course on mathematical methods or a junior/senior laboratory. However, because of the nature of computer simulations, it is possible to teach a meaningful course on computer simulation methods and applications to even first-year students. Once the students have the basic skills, they can utilize these skills in other courses. We stress that for such a course to have maximum impact on the students and the physics curriculum, the emphasis of the course should be on computer simulations rather than numerical analysis.

Unfortunately, even when a separate course is offered, it has not yet led to much use of computer simulations in other courses in most physics departments. The reasons for this lack of impact are that the course is rarely required of all majors, faculty teaching other courses are unfamiliar with simulations or do not have the time to change their courses, and the lack of readily available resources for incorporating simulations. We are trying to rectify this last obstacle for thermal and statistical physics courses by developing applets and various Java utilities and templates and other curricular materials [12].

Some textbooks in classical mechanics have included a few computer exercises for which the programming is minimized. The focus of these exercises is usually similar to the standard problems and does not allow students the opportunity to engage in more open-ended possibilities that are analogous to research problems. However, it might be possible to add a computer simulation lab to a course or several courses so that students can learn enough programming to write a simulation. The advantage is that their work can be done in a specific physical context such as modeling particle motion in mechanics or wave packet propagation in quantum mechanics.

So far much of our discussion has focused on using the computer in standard physics contexts. Such contexts can be very useful. For example, much of the interesting behavior in particle dynamics such as chaotic phenomena is very difficult to analyze without a computer. The motion of wave packets in a potential cannot be easily visualized without a computer. These are obvious applications that should be done in the relevant physics courses, and we can design open-ended problems that will allow students the opportunity to explore interesting physics as well as obtain a better understanding of traditional concepts.

However, real change in the curriculum will come when the use of computer simulations pushes us to broaden the focus of the physics curriculum. For example, we can extend the simulation of particle systems to the simulation of dynamical systems in general. Why not expand simulations of statistical mechanics models using Monte Carlo and molecular dynamics methods to include more general studies of complex systems such as traffic flow, epidemiology, and neural networks? In both cases we see that the extension of computer simulations beyond traditional topics in physics leads naturally to what we call computational science. Many computational approaches such as genetic algorithms and cellular automata have their origins outside of physics. Including these approaches would provide a powerful pathway to understanding complex systems through computational science. Physics courses may be the most natural setting for introducing computational science because physics historically has been at the forefront in developing new ways of solving problems experimentally, theoretically, and now computationally.

5 Making Teaching Count

The use of the Internet as a vehicle for delivering curricular materials has many implications for teaching. In particular, it allows us to share our course materials and our approach with instructors at other institutions. One reason that teaching is not taken as seriously as research is that our teaching reputations are local and the quality is not easily evaluated. The advent of the Web has already started to change this situation.

We can also use the Web to develop curricular materials in a way that takes advantage of the collective work of many people. We are inspired by the freely available Linux operating system, which was inspired by the vision of a single individual, but which has been developed by substantial contributions from many people. Anyone is free to work on any additional features or improvements and all new code is easily available for others to use and critique. At the same time, there is always a stable release available that incorporates new features only after they have been thoroughly tested. We are trying to follow this example and develop a core of curricular materials for statistical and thermal physics. It is too soon to say, but we almost have a sufficient core of material that would be useful to other instructors, and without any advertising on our part, other scientists are beginning to make suggestions for improvements. It remains to be seen if others will contribute substantially to this curriculum development project, but the possibility exists for "open source" curriculum development projects in physics and other areas.

Acknowledgements

The authors would like to thank Wolfgang Christian and Joshua Gould for their patience while teaching us much about Java and for their development of the Java templates and utilities that we believe will make Java programming accessible to a much wider group of physics instructors and their students. We also acknowledge the National Science Foundation for its support.

References

1. More information about the Accreditation Board for Engineering and Technology can be found at `http://www.abet.org/`.
2. D. J. Griffiths, *Introduction to Electrodynamics*, 3rd Edition, Prentice Hall (1999).
3. D. J. Griffiths, *Introduction to Quantum Mechanics*, Prentice Hall (1995).
4. This section is based on J. Tobochnik and H. Gould, "Teaching Computational Physics to Undergraduates," D. Stauffer, ed., *Annual Reviews of Computational Physics IX*, World-Scientific Press (to be published).
5. See for example, K. Binder in *Thermodynamics and Statistical Physics: Teaching Modern Physics*, pp. 45–66, Proc. 4th IUPAP Teaching Modern Physics Conf., M. G. Velarde and F. Cuadros, eds., World Scientific Press (1995).
6. R. Chabay and B. Sherwood, *Matter & Interactions*, John Wiley & Sons, to be published. More information about the text and their course at Carnegie-Mellon University is available at `http://cil.andrew.cmu.edu/mi.html`.
7. More information about the three-dimensional graphics module for Python is available in D. Scherer, P. Dubois, and B. Sherwood, "VPython: 3D interactive scientific graphics for students," Comput. Sci. Engin., Sept./Oct. 2000, 82–88 and at `http://cil.andrew.cmu.edu/projects/visual/`.
8. See for example, W. Christian and M. Belloni, *Teaching Physics with Interactive Curricular Material*, Prentice Hall (2001), W. Christian, M. Belloni, and M. Dancy, "Physlets: Java Tools for a Web-Based Physics Curriculum," this conference proceedings, and `http://webphysics.davidson.edu/Applets/Applets.html`.
9. More information about Interactive Physics can be found at `http://www.workingmodel.com/products/ip.html`.

10. Information about Physics Academic Software can be found at `http://webassign.net/pasnew/`.
11. H. Gould, J. Tobochnik, and W. Christian, *Introduction to Computer Simulation Methods*, third edition (unpublished). A preliminary version of the third edition will be available this summer from the Simulations in Physics Web site, `http://sip.clarku.edu`. The second edition by H. Gould and J. Tobochnik, Addison-Wesley (1996) is still available.
12. For information about the Statistical and Thermal Physics (STP) curriculum development project, see `http://stp.clarku.edu`.