Developing Components and Curricula for a Research-Rich Undergraduate Degree in Computational Physics

Invited paper, 2001 International Conference on Computational Science,

San Francisco, May 2001

Rubin H Landau[**]

Physics Department, Oregon State University, Corvallis, OR 97331, USA.
`rubin@physics.orst.edu`, `http://www.physics.orst.edu/~rubin`

Abstract. A four-year undergraduate curriculum leading to a Bachelor's degree in Computational Physics is described. The courses, texts, and seminars are research- and Web-rich, and culminate in an Advanced Computational Science Laboratory derived from graduate theses and research from NPACI centers and national laboratories. There are important places for Maple, Java, MathML, MatLab, C and Fortran in the curriculum.

1 Overview

We are presently experiencing historically rapid advances in science, technology, and education driven by a dramatic increase in the power and use of computers. Whereas a decade ago computational science educators were content to have undergraduates view scientific computation as "black boxes" and to wait for graduate school for them to learn what is inside the boxes [1], our increasing reliance on computers makes this less true today, and much less true in the future. To adjust to changes in scientific computing, we are developing a four-year, research-rich curriculum leading to Bachelor of Science and Bachleor of Arts degrees in Computational Physics (CP).

Our department has already developed an award-winning undergraduate course in *Computational Physics* [2], a text book [3] that is joining others [4] as proposed models for undergraduate CP courses [5], web-based tutorials and demonstrations that enhance the course and text [6], and a newer-still one-quarter course in *Introductory Scientific Computing* for which we are developing curriculum materials. With the addition of one new course, an *Advanced Computational Science Laboratory*, the modification of one other, and the use of courses offered in other departments, we believe we have assembled a coherent and innovative undergraduate degree program in CP.

[**] Supported in part by National Science Foundation Grant 9980940 and the National Partnership for Advanced Computational Infrastructure, EOT.

By teaching some of the computing classes within the Physics Department we will be able to adjust their content and depth to provide a balanced program within the allowed credit limit. This will also permit us to work around the budget difficulties that restrain other departments from teaching shortened versions of the courses taught to their own majors. Our program may well act as a stepping stone for further interdisciplinary programs at the undergraduate and graduate levels.

2 Need for Program

A bachelor's degree in any computation science is rare. To the best of our knowledge there is only one bachelor's degree in CP in the United States [7], and just several physics degrees with minors or specialties in CP [8, 9]. Impressively, Trinity College, Dublin has obtained national support to start a B.S. in CP program [10]. We hope that the rarity of degrees in computational science and our promotion will draw new students to our program. In addition, with cooperation across campus, incoming undergraduates will now be presented with a variety of ways to combine computers with science, some in the College of Engineering and some in the College of Science.

Our program is challenging and requires dedicated students. Considering the serious commitment required for interdisciplinary studies, the awarding of a specific degree in CP will, in part, provide recognition for professional schools and employers of the student's extraordinary achievement. This program will also help meet the broadly recognized need to provide undergraduates with research experience, an experience usually associated with highly-ranked universities. Because the research laboratory for computational physics is a virtual world created by the computer, it is easier and quicker to have undergraduates work in this lab than in a "wet" one.

Our goal is to present the job market and graduate schools with undergraduate students possessing competent education and training in physics, applied mathematics, computing, and complex problem-solving. Since the operation of our program also nurtures the communication and team-work skills valued by present employers [11], our students should be valued new hires. We suspect that other physics departments might also follow our model as a good way to revitalize their physics offerings, and that the nation will be served well with these types of graduates.

One need for a program such as ours arises from the documented observations that many of the computer science students now finding jobs do not have the background in mathematics and science needed for technical fields, and that most of the science and engineering students now finding jobs in computer-related fields do not have the requisite background in computation. Another need has been noted by the President's Information Technology Advisory Committee [12] who emphasized the severe shortage of information technology workers. All Computer Science Department throughout the country graduating students at full capacity could not meet this need. The same observation has been made by the US Department of Commerce and by InfoWeek. PITAC recommends that disciplines in addition to CS supply the workers.

2.1 Educational Objective and Approach

Our objective is to have students understand how to perform scientific computations and experience the interweaving of high-performance computing and communications into the fabric of modern science and engineering. When successful, the mathematical equations and connections among physical idea become alive before students' eyes, and the students understand physical systems at a level usually attained only in a research environment. To do good science and engineering with computers, the student should understand a) how the computer works, b) the relevant science and mathematics, and c) how algorithms and computer simulations are used to connect a) and b).

Much of the computational materials students will encounter in our program come from basic research projects, and will be set in the scientific problem-solving paradigm [2, 13]:

$$\boxed{\text{problem} \rightarrow \text{theory} \rightarrow \text{model} \rightarrow \text{implementation}} \leftrightarrow \boxed{\text{assessment}}$$

where the assessment links back to all steps in the big box. This paradigm distinguishes the different steps in scientific problem solving, encourages the use of a variety of tools, and emphasizes the value of continual assessment. Building our material according to this paradigm not only emphasizes the need to know more than what is in the traditional physics curriculum in order to be creative in computational physics, but also makes it easier for students from non-physics disciplines to follow the material and take the courses. While a benefit of our project will be increased understanding of physics content, use of the problem-solving paradigm will deepen scientific process skills [14].

Key components of our program are having students get actively engaged with projects, as if each were an original scientific investigation, and having projects in a large number of areas. In this way students experience the excitement of individual research, get familiar with a large number of approaches, acquire confidence in making a complex system work for them, and continually build upon their accomplishments. We have found this project approach to be flexible and to encourage students to take pride in their work and their creativity. It also works well for independent study or distant learning.

A valuable part of our students' education will be summer involvement with computational research programs within the university and at laboratory and industrial sites. In particular, our program is part of the Education, Outreach, and Training thrust area [15] of the National Partnership for Computational Infrastructure [16], and they are providing us with both support and guidance. They have world-leading supercomputing facilities, training classes, summer workshops, internship programs and collaborations with other NPACI projects, in which we plan to involve our students.

2.2 Computer-Mediated Learning and Accessibility

Our curriculum will be rich with web materials that are used and, at times, developed by the students. This reflects developments in our department and research groups over the last eight years in web-enhanced education [17, 18], and our view that the web and computer-mediated instruction will play an increasing role in future scientific computing and education.

While we do not view the web as a good single choice for the teaching medium for general physics and for most students, one cannot beat having a motivated student sit at a computer in a trial-and-error mode in order to learn scientific computing [19]. Further, the web is an ideal environment for computational science: projects are always in a centralized place for students and faculty to observe, codes are there to run or modify, and visualizations can be striking in 3-D, color, sound, and animation. In fact, our planned mix of the web, computer-mediated learning, projects, and lectures for the computational courses is similar to the new pedagogical strategy known as *Just-in-Time Teaching* [20] and the use of *physlets* [21]. Taken together, our approach combines the use of technology and hypermedia to assist learning abstract concepts [22], and the pedagogical strategy known "active learning" or "interactive engagement" [23, 24].

The Physics Department has a research group, the Science Accessibility Project [25], that develops ways to make scientific materials accessible to print-disabled (blind and dyslexic) students. In addition, our research group has benefited from a number of academically-gifted students who are seriously dyslexic or physically disabled. Modern computing equipment has helped these students produce high quality research projects and excel in their careers. We will continue to look for ways to use the intellectual and physical leverage provided by computer and communication technology to permit people with disabilities to become productive scientists, and will ensure than our program is open and welcoming to them. Specifically, our program, with NPACI [16] assistance, will incorporate SAP developments as well as assist in SAP research by incorporating the techniques used to produce accessible documents with MathML and XML [26, 27] into the course materials.

2.3 Course of Study

In Table 1 we give a sample schedule of the B.S. in CP curriculum. The two-credit courses are parts of the separate *Paradigms* project [28] that has reorganized our department's mid-level undergraduate courses into smaller blocks, with a block covering related ideas normally found in a number of traditional classes. Table 1 is just one possible arrangement of the required courses; others exist, as well as ones in which substitutions are made dependent upon the student's interests and the advisor's consent. The program has 21 credit hours of electives compared to the Physics B.S. degree that has 25 hours. Essentially, we are picking some of the "electives" a student might choose to specialize in computational science.

Table 1. Sample schedule showing proposed curriculum for **B.S. in Computational Physics** with 180 total credits (1 credit = 10 class hours). Computer-intensive courses shown in **bold.** Courses suggested for electives or approved substitution: PH 415, Computer Interfacing; PH 435, Classical Mechanics; MTH 452, Numerical Solution of Ordinary Diffrntl Equations; MTH 453, Numerical Solution of Partial Diffrntl Equations; CS 311, Operating Systems; CS 361, Fundamentals of Software Engineering; PH 428, Rigid Bodies; Ph 441, Physical Optics; PH 481, Thermal and Statistical Phys; PH 621, Classical Dynamics.

	Fall	**Winter**	**Spring**
Fresh (46)	Diffrntl Calculus (MTH 251, 4) Gen Chemistry (CH 201, 3) Fitness/Writing I, 3 Perspective, 3 **CP/CS Seminar** (PH 407, 1)	**Scientific Comptng I** (PH/MTH/CS 265, 3) [or Fall term] Integral Calculus (MTH 252, 4) Gen Chemistry (CH 202, 3) Perspective, 6 [or Fitness/Writing I, 3]	**Intro Computer Sci I** (CS 161, 4) Vector Calculus I (MTH 254, 4) Gen Phys, Rec (PH 211,221; 4,1) Fitness/Writing I, 3 [or Perspective, 3]
Soph (45)	**Intro Computer Sci II** (CS 162, 4) Writing II, 3 Vector Calculus II (MTH 255, 4) Gen Phys, Rec (PH 212,222; 4,1)	**Discrete Math** (MTH 231/235, 3) Infinite Series and Seqncs (MTH 253, 4) Gen Phys, Rec (PH 213,223; 4,1) Perspective, 3	**Scientific Comptng II** (PH 365, 3) **Linear Algebra** (MTH 341, 3) App Diffrntl Eqs (MTH 256, 4) Intro Modern Phys (PH 314, 4)
Jr (44)	**CP Simulations I** (PH 465, 3) **CP Seminar** (PH 407, 1) **Intro Probability** (MTH 361, 3) Oscillations (PH 421, 2) Static Vector Fields (PH 422, 2) Writing III/Speech, 3	**CP Simulations II** (PH 466, 3) **Data Structures** (CS 261, 4) Waves in 1D (PH 424, 2) Quantum Measurement (PH 425, 2) Central Forces (PH 426, 2) Elective/Perspsective, 3	Periodic Systems (PH 427, 2) Class/Quant Mechan (PH 435/451, 3) Energy and Entropy (PH 423, 2) Biology, 4 Perspective/Elective, 3
Sr (45)	**Num. Lin Alg.** (MTH 451, 3) Electromagnetism (PH 431, 3) Mathematical Methods (PH 461, 3) Elective, 6	**Adv CP Lab** (PH 417,517; 3) **Social & Ethical CS** (Synthesis, CS 391, 3) Elective, 6 Synthesis, 3	**Thesis** [**CP Lab+WIC**] (PH 401, 4) **Interact Multi Media** (CS 395, 4) **CP Seminar** (PH 407, 1) Electives, 6

Physics/Mathematics/Computer Science 265, Scientific Computing I

1 Unix, Windows, Maple, Numbers	6 Logical Flow Control
2 Basic Maple, Functions	7 Loops, Numerical Integration
3 Floating Points, Symbolic Computing	8 Complex Arithmetic, Objects
4 Visualization, Calculus, Root Finding	9 OOP, Matrix Computing
5 Classes and Methods	10 General I/O, Applets

An introductory course designed to provide the basic computational tools and techniques needed by lower division students for study in science and engineering. The course is based on a project approach using the problem solving environment *Maple* and the compiled language *Java*. For most students this course will be their first experience with visualization tools, the use of a cluster of workstations sharing a common file system, and the Unix operating system. (Learning Unix is assisted by the web-based *Interactive Unix Tutorial* we have developed and distributed nationally [17].)

While the scientific programming of applications in C and Java is similar, our recent switch to Java in place of C provides an *object-oriented* view towards programming (inclusion of methods with variables), demonstrates the developing potential of platform- and operating-system independent programming, and emphasizes that the web is an integral part of future scientific computing. We have found Java's handling of precision, errors, variable types, and pointers to be superior to C for scientific computing. We also find Java's platform and system independence attractive since this may modify the too rapid (2-3 year) obsolescence of educational software, and encourages distributed computing over the web.

Physics 365, Scientific Computing II

1 Software Basics	6 Diffrntl Equations
2 Errors and Uncertainties	7 Hardware: Memory and CPU
3 Integration & Differentiation	8 Matrix Computing
4 Data Fitting	9 Profiling and Tuning
5 Random Numbers	10 Parallel Computing

An intermediate level course that provides the basic mathematical, numerical, and conceptual elements needed for utilizing computers as virtual scientific laboratories using Java, C, and Fortran. The basics of computer hardware, such as memory and CPU architecture, and shell programming with the Unix operating system are presented. Also studied are the basics of scientific computing: algorithms, precision, efficiency, verification, numerical analysis and associated approximation and round-off errors, algorithm scaling, code profiling, and tuning. Examples are taken from elementary physical systems that make the concepts clear, as well as being easy to compute. The limits of model and algorithm validity is demonstrated by investigating the physics simulation examples in regions for which there is manifest numerical failure.

Physics 407 Computational Physics Seminar Reports of modern happenings, campus research results, and journal articles are presented and discussed. Undergraduates will hear about and learn to think about research topics while advanced students will present results of their projects and research.

Physics 417/517 Advanced Computational Laboratory

Dilos (Giebultowicz)	Monte-Carlo ordering in dilute semiconductors
DFT (Jansen)	Density functional theory of super lattices
DFT2 (Jansen)	Molecular dynamics
Gamow (Landau)	Bound states & resonances of exotic atoms
HF (Jansen)	Hartree-Foch calculations of atoms & molecules
LPOTT (Landau)	K and π elastic scattering from spin 1/2 nuclei
LPOT$_{II}$ (Landau)	MPI code for polarized proton scattering
LPOTp (Landau)	Nucleon-nucleus scattering in momentum space
MD (Rudd, CS)	Molecular dynamics simulations of SiO_2
MEG (Landau)	Principal components of magnetic brain waves
Monte (Giebultowicz)	Monte-Carlo simulations of magnetic thin films
nScatt (Giebultowicz)	Monte-Carlo simulations of neutron diffraction
PiN (Landau)	Quark model of $\pi - N$ interaction
Qflux (Landau)	QCD of 3D quark flux tubes
Shake (Jansen)	Earthquake analysis
Transport (Palmer, NE)	Transport simulations of nuclear storage

We are developing a completely new, advanced computational laboratory in which senior CP students and graduate students will experiment with computer simulations taken from previous M.S. and Ph.D. research projects, as well as from research projects at national laboratories. We are writing and will publish the laboratory manual for this course. The research descriptions and computer simulations will be modified in order to provide a research experience accessible to undergraduate students in a short time (in contrast to the people-years required to develop the research codes originally).

To learn that codes are pieces of scientific literature designed to be read and understood by more than just their authors, the students will run, profile, modify, parallelize, and extend these working codes. The students will run some simulations without knowing what results to expect as a way of learning that codes also function as virtual laboratories built to explore nature. Since the projects will be based on existing research codes, many of the programs will have been written in some version of Fortran. This will be a valuable experience for students since often Fortran is no longer taugh even though the majority of high performance computing applications are presently written in some version of it. In fact, we have heard from some of our industrial colleagues that lack of knowledge of Fortran and lack of experience with running large codes written by others are some of the weakness they find in present graduates.

Physics 465, 466 Computational Physics Simulations

1 Quantum Eigenvalues, Root Finding	9 Quantum Path Integration
2 Anharmonic Oscillations	10 Fractals
3 Fourier Analysis of Oscillations	11 Electrostatic Potentials
4 Unusual Nonlinear Dynamics	12 Heat Flow
5 Differential Chaos in Phase Space	13 Waves on a String
6 Bound States in Momentum Space	14 KdeV Solitons
7 Quantum Scattering, Integral Eqns	15 Sine-Gordon Solitons
8 Thermodynamics: The Ising Model	16 Electronic Wave Packets

The techniques covered in Scientific Computing, PH 265 and 365, are applied and extended to physical problems best attacked with a powerful computer and a compiled language. The problems are taken from realistic systems, with emphasis on subjects not covered in a standard physics curriculum. The students work individually or in teams on projects requiring active learning and analysis.

The course is designed for the student to discuss each project with an instructor and then write it up as an "executive summary" containing: *Problem, Equations, Algorithm, Code listing, Visualization, Discussion, and Critique.* The emphasis is professional, to make a report of the type presented to a boss or manager in a workplace. The goal for the students is to explain just enough to get across that they know what they are talking about, and be certain to convey what they did and their evaluation of the project. As part of the training, the reports are written on the web.

3 Program Evaluation, Student Learning Assessment

The initial and periodic evaluation of our materials will be made by the students in the classes we teach. This will be done regularly through discussions conducted in the Computational Physics Seminar, with web surveys, as well as with the mandatory class evaluation forms. An evaluation of the technical content of our program will be requested from our Advisory Board. Secondary evaluation will be made by the Physics Department of Western Oregon University, who will also enroll interested WOU students in our program. Third-party formative and summative evaluation will be provided by the University of Wisconsin-Madison's *Learning through Evaluation, Adaptation and Dissemination Center* [29], a team that has developed a national reputation for its evaluations of educational reforms that utilize high performance computer technologies, and that aim to recruit and retain women and under represented minorities into the fields of science, mathematics, engineering, and technology.

References

1. *Workshop Report, Undergraduate and Graduate Education in Computational Science*, D. Greenwell, R. Kalia, P. Vashista, and H. Myron, Eds., Louisiana State Univ., Argonne National Lab., April 1991.
2. The Undergraduate Computational Engineering and Sciences (UCES) Project, http://www.krellinst.org/UCES/index.html.
3. R.H. Landau and M. J. Paez (Coauthors), H. Jansen and H. Kowallik (Contributors), *Computational Physics, Problem Solving with Computers*, John Wiley, New York, 1997; http://www.physics.orst.edu/˜rubin/CPbook.
4. Introduction to Computer Simulation Methods, Applications to Physical Systems, Second Edition, Harvey Gould and Jan Tobochnik, Addison-Wesley, Reading, http://sip.clarku.edu/.
5. Harvey Gould and Jan Tobochnik, Amer. J. Phys, **67** (1), January 1999; William H. Press, Phys. Today, p 71, July 1998.

6. Northwest Alliance for Computational Science and Engineering, an NSF Metacenter Regional Alliance centered at Oregon State University, http://www.nacse.org.
7. Degree Sequence in Computational Physics, Illinois State University, http://www.phy.ilstu.edu/CompPhys/CP.html;
8. Syracuse University, Bachelor of Arts degree with an Option Physics and Computation, http://suhep.syr.edu/undergraduate.
9. Rensselaer Bachelor of Science in Applied Physics Curriculum, http://www.rpi.edu/dept/phys/Curricula/currAppPhysComp.html.
10. Computational Physics and Chemistry Degree Courses, Trinity College, Dublin (Ireland), http://www.tcd.ie/Physics/Courses/CCCP/CCCPflyer.html
11. *Skills Used Frequently by Physics Bachelors in Selected Employment Sectors*, American Institute of Physics Education and Employment Statistics Division, (1995).
12. President's Information Technology Advisory Committee, http://www.ccic.gov/ac/.
13. The Shodor Education Foundation, Inc., http://www.shodor.org/.
14. R. Root-Bernstein, Discovering, Random House, New York (1989).
15. Education, Outreach, and Training thrust area of the National Partnership for Computational Infrastructure, http://www.npaci.edu/Outreach.
16. National Partnership for Advanced Computational Infrastructure, http://www.npaci.edu/.
17. The Landau Research Group, NACSE in Physics, Oregon State University, http://nacphy.physics.orst.edu.
18. R.H. Landau, H. Kowallik, and M. J. Paez, *Web-Enhanced Undergraduate Course and Book for Computational Physics*, Computers in Physics, 12, (1998); http://www.aip.org/cip/pdf/landau.pdf.
19. P. Davis, *How Undergraduates Learn Computer Skills: Results of a Survey and Focus Group*, T.H.E Journal, **26**, 69, April, 1999.
20. *Just-in-Time Teaching: Blending Active Learning with web Technology*, G.M. Novak, E.T. Patterson, A.D. Gavrin, and W. Christian, Prentice Hall, Upper Saddle River, 1999.
21. *Physlets: Teaching Physics with Interactive Curriculum Material*, W. Christian and M. Belloni, Prentice Hall, Upper Saddle River, 2001.
22. C. Dede, M. Salzman, R.B. Loftin, and D. Sprague, *Multisensory Immersion as a Modeling Environment for Learning Complex Scientific Concepts*, Computer Modeling and Simulation in Science Education, eds. N. Roberts, W. Feurzeig, and B. Hunter, Springer-Verlag, New York, 1999.
23. R.R. Hake, *Interactive-engagement vs. traditional methods*, Am. J. Phys., **66**, 64–74 (1998);
    T.E. Sutherland and C.C. Bonwell, eds., *Using active learning in college classes; a range of options for faculty*, Jossey-Bass, San Francisco (1996).
24. D. R. Sokoloff, *Using Interactive Lecture Demonstrations to Create an Active Learning Environment*, The Physics Teacher, 35, 340 (1997).
25. Science Access Project, http://dots.physics.orst.edu/.
26. HTML Math Overview, World Wide Wed Consortium, http://www.w3c.org/math.
27. XML Bridges the Gap, Info World Electric, June 1, 1998 20, Issue 22, p.88-90, http://www.infoworld.com/cgi-bin/displayArchive.pl?/98/22/i09-22.88.htm; S. Mace, U. Flohr, R. Dobson, and T. Graham, Weaving a Better Web, Byte, p58, March 1998.
28. Paradigms in Physics Project, http://www.physics.orst.edu/paradigms/.
29. Learning through Evaluation, Adaptation and Dissemination (LEAD) Center, University of Wisconsin, http://www.cae.wisc.edu/ lead/.