COMPUTATIONAL SCIENCE AND ENGINEERING: How Physicists Can Be Useful

by **Martin Plumer**, Centre de Recherche en Physique du Solide et Département de Physique, Université de Sherbrooke, Sherbrooke, Québec, Canada J1K 2R1

ABSTRACT

As traditional employment opportunities in academic as well as in industry and government-funded basic research environments continue to diminish, an increasing number of physics (and other science) graduates have found jobs developing and solving "numerical models" of a wide variety of physical (and other) phenomena. An alternative route to learning some of these marketable skills is available through the twenty or so *Computational Science and Engineering* university programs which have recently been developed in the U.S. and Canada. A review is given here of the goals of such programs, some examples of their implementation, and a summary of the information available on this rapidly developing educational focus.

INTRODUCTION

"Computational Science and Engineering" (CSE) is one of the focal points of the *Grand Challenges* outlined in the High Performance Computing and Communications Act, passed by the U.S. Congress in 1991. Partly as a result of this initiative, as well as a perceived need by industry, there is an increasing number (about 22 at present) of universities in the U.S. which offer, or plan to offer, degrees (or certificates or minors) in CSE. Such programs are an effort to fill a wide gap in the education which scientists and engineers traditionally receive at university. Computational science is an attempt to develop a coherent methodology for the numerical study of complex physical systems that are either too difficult or that would be too costly to investigate by other means, such as experimentation. This description could well apply to the activities of a large number of physics graduates hired by industry and government laboratories in recent years. Few of them are using the pure science learned beyond the undergraduate level. It was their experience in the scientific method and computational techniques that proved attractive to employers. In view of the continually diminishing role of university and other fundamental research opportunities, it may be time to examine alternative science education programs that attempt to address the current needs of high-tech industry. A review is made here of the CSE movement in the U.S., and in particular of the CSE university degree programs, as well as information on how to obtain further information on this expanding educational focus. Similar degree programs at Canadian universities are also summarized.

A good initiation to the CSE community may be found in the article by D.E. Stevenson, "Science, Computational Science, and Computer Science: At a Crossroads."[1] In his words, "Computational science focuses upon the development of computationally feasible models for physical systems, developing algorithms for solving issues arising in the modeling process and matching algorithms to computer architectures." In the ideal case, it is through a profound understanding of the physical system that a suitable model arises. Even this aspect may require a return to very basic physical ideas since so much of what we have learned about a given physical process has been guided by the approximations that must be made in order to obtain any kind of solution. It is easy for the undergraduate to forget that the simple pendulum is a highly non-linear device that requires a numerical solution. Much of physical modelling involves such linearization or other approximations of a more fundamental nature. The hidden assumptions made in the development of physical models are a possible source of danger to the realization of practical solutions for complex systems. It is for this reason that Stevenson suggests that the types of numerical solutions possible should play an integral part in the modelling process itself. One of the more obvious targets of the CSE movement is to teach scientists and engineers the effective use of computational techniques beyond the usual introductory computer course and *ad hoc* learning that subsequently occurs. Stevenson points out that the result of most scientists' programming efforts are "inappropriate, ineffective, and inefficient" algorithms. Another essential focus of CSE is its multidisciplinary aspect: many difficult problems would greatly benefit from the collaboration of physical scientists, engineers, mathematical scientists, and computer scientists. Cross-mixing of expertise is commonplace in the private sector and CSE programs should reflect this reality by educating scientists to work on interdisciplinary problems in interdisciplinary groups.

An indication of the popularity of CSE programs is the existence of the new U.S. journal *IEEE Computational Science and Engineering*, published four times a year since 1994. John Rice's article "Computational Science and the Future of Computing Research"[2] reminds us of the increasing importance of *strategic research*. Due to recent budget cuts, as well as the possibility that there are simply too many scientists investigating fundamental (and thus not immediately useful) phenomena, the pressure to do research that will have practical value is increasing. He points out that computational scientists are well positioned to meet this challenge of providing a reasonable expectation of useful results. For example, the time involved in the design, testing, and improvement of complex machinery (e.g., an automobile engine) can be greatly reduced by simulation (in place of actually building a physical structure).

In "What Industry Needs from Academia: CSE Education in the 21st Century"[3], industry representatives voiced their concerns regarding science education. Most agreed that they would benefit from a higher educational system that involved open lines of communication with industry, including the exchange of research ideas, team teaching, and industry-based student projects. Attempts at a multidisciplinary education were also considered very important. In the industrial setting, researchers from diverse fields are often required to discuss problems common to a number of different research divisions. Current students experience difficulty in communicating with people outside their own field. The call for people with a flexible "renaissance" education, who can work well in teams, was made by several of the representatives. The emerging importance of CSE-trained personnel is illustrated by the revelation from Boeing's K. Neves that "The 777 was completely designed on a computer..."

Further indications of the commitment by the U.S. federal government to CSE may be found in the education award and grant programs sponsored by the U.S. Department of Energy (DOE). These include Adventures in Computing, intended to initiate interest in CSE-related activities in the K-12 student population, the Undergraduate Computational Engineering and Science (UCES) Project, as well as the Computational Science Graduate Fellowship Program.[4] The three principal objectives of the UCES project are to identify and strengthen the CSE community, to form cohesive CSE university curricula, and to develop readily accessible CSE *instructional* material (courseware). Motivated by the observation that there is a growing sector within science and engineering in which advanced computational methods are used extensively, yet few university programs go beyond the basic courses in computational mathematics, the UCES has attempted to solidify its community through the sponsorship of workshops and awards. The first education workshop held in 1994 had 80 participants and met to examine different models for CSE academic programs.[5] It was generally agreed that there should not be an undergraduate CSE major but that CSE courses should be offered at the undergraduate level. The "Industrial Check Panel" noted that industry wants prospective engineers to have greater computational sophistication oriented towards practical problems. The award programs, also begun in 1994, are in recognition of excellence in CSE teaching, of outstanding CSE education projects, and individual student class projects. 1995 education project winners include "The Pit and the Pendulum," where students gain an understanding of what factors influence the period of a pendulum. A focus of CSE education is teaching through problem-based modules, by which students learn the five basic CSE components: problem, model, method, implementation, and assessment. Such on-line, interactive courseware is intended to be self-contained teaching units. An example is "RealSim," which calculates the ground state of a mutually repulsive N-body problem in two dimensions, within neutral or replusive boundaries, through Monte Carlo and simulated annealing techniques.[6] The user can investigate such problems as quenching versus annealing for a given N and set of boundary conditions. Such projects are the result of the DOE's 1991 initiative which brought together 35 authors to write and test a collection of "hypermedia" teaching aids, resulting in the 1993 version of the "e-book".[7] These CSE teaching units available on the internet are currently being used by educators to support their coursework. The most substantial support from the DOE is in the form of one-year fellowships (renewable) available to graduate students in applied science and engineering whose research involves applications in high-performance computing. There are currently about 30 such fellows.

The NSF also sponsors several programs within its Education Directorate.[8] These include funds for the purchase of instructional computers within the Instrumentation and Laboratory Improvement Program, as well as for workshops, short courses, and seminars within the Undergraduate Faculty Enhancement Program. In addition, there is support for the improvement of education within the Undergraduate Course and Curriculum Development Program, for young graduates within the Postdoctoral Research Associateships in CSE program, for educational programs within the five NSF Supercomputing Centers, as well as for universities which do not offer graduate studies within the Research in Undergraduate Institutions Program.

WHAT SHOULD COMPUTER SCIENTISTS TEACH TO PHYSICAL SCIENTISTS AND ENGINEERS?

One of the principal objectives of new CSE academic programs is to teach scientists and engineers the effective use of computational techniques beyond the usual introductory computer course and *ad hoc* learning that occurs subsequently. In a series of articles entitled "What Should Computer Scientists Teach to Physical Scientists and Engineers?"[9] suggestions are made for the content of an intensive one-week course that might be given to every new graduate student in science and engineering (or every new employee in a company's R&D division).

G.V. Wilson (IBM Canada) suggests that the one-week course should provide a formal introduction to programming aids and the widely available tools and platforms. His day by day schedule for the week includes

(1) Introduction to Unix, Editing, The World Wide Web, and HTML.[10]

(2) Latex[11] and Make.[12]

(3) Perl.[13]

(4) Programming tools: Revision management, Debugging, Profile and Tuning.

(5) Software testing and other tools.

He notes that a longer course might also include introductions to Mathematica, modular visualization environments, spreadsheets and data bases, X Windows setup, modern program languages, algorithm and data structures, computer architecture, software engineering, numerical methods, databases, operating systems, compilers, parallel processing, as well as a 3-month project.

A somewhat contrasting view is presented by R.H. Landau (Oregon State) where the role of applications is emphasized. His university's computational physics course uses learning-through-example and stresses (among other things) "the development and organization of thinking about physical systems in a manner compatible with advanced computational analysis." He believes that programming should deepen our understanding of physical systems and provide visualizations of numerical solutions that serve to enhance this understanding. His course also involves error and accuracy questions. Course topics include

I. Computational Concerns: kinematics, randomness, vector and parallel computing.

II. Numerical Analysis: Errors, DE's and oscillations, Fourier analysis of nonlinear oscillations, libraries.

III. Applications: Data fitting, Monte Carlo, dynamics of nonlinear systems, chaos, fractals.

IV. PDE's: Electrostatic potentials, heat flow, waves, solitons.

Landau remarks that the course's Monte Carlo project, for example, is successful if it allows a student to "see" a process that they could previously only imagine or read about." The course projects attempt to give students a deeper insight into scientific modeling and a "new view of reality." His book *A Project Approach to Computational Physical Science* is to be published by John Wiley in 1997.[14]

S. McConnell (Construx Software Builders) believes that the techniques for writing small computer programs learned by typical physical science undergraduates are completely inappropriate to the development of the larger professional codes required to solve complex research problems. Such a one-week course should not teach tricks on how to marginally increase efficiency but rather it should provide the knowledge of how to "avoid catastrophe." Instead of focusing on the detailed use of specific tools and computing practices, his approach would be to awaken students to a "world of possibilites" of how to build a computer program, with a course covering the following topics:

1. Programming practices: Coding for humans, control issues, integration strategies and sources of information.

2. Software design: The importance of design, information hiding and modularity.

3. Quality assurance: Testing, debugging, and peer reviews.

4. Software project management: Revision management, examples, maintenance, coordination of group projects.

5. Tools and wrap-up: Operating systems, editors, Mathematica, etc.

Several other suggestions are also put forward by physicists which combine the above approaches of Wilson's tool-oriented, Landau's application-oriented, and McConnell's practices-oriented one-week courses. The exception is T. Issaevitch (Wolfram Research), whose article is entitled "Forget Multiple Tools; Use Mathematica."

It is clear from this series of articles that many believe traditional science and engineering university programs which have changed little over the past twenty years are not preparing students for today's computation-oriented job market.

U.S. UNIVERSITY PROGRAMS

The 1994 summary of "Academic Programs in Computational Science and Engineering"[2] serves as the best starting point for information on U.S. universities offering or planning graduate degrees in CSE or related subjects. Implementations of these programs are diverse. Some have "stand-alone" CSE degrees; in other cases, students receive a CSE certificate in conjunction with their home-department degrees; and other universities offer a CSE minor. The degrees awarded can be at the Professional Masters, Masters, or Doctoral level, depending on the university. Administration of the degrees is also varied. Most have separate CSE departments, institutes, or centers, a few with their own distinct faculty. The majority involve the participation of regular faculty members from up to 15 separate departments and are run by committee and report directly to the university provost. In many cases (certificates or minors), the separate home departments control an individual student's curriculum. The initiation of most of these programs has required the development of only a few new courses. The remaining credits are taken from the existing math, computer science, and regular science and engineering courses which emphasize desired aspects of computational science.

Updated academic information, as well as a summary of all aspects related to CSE education, may be found by visiting the Computational Science Education web site created by C.D. Swanson at Cray Research.[8] It contains further details on each undergraduate and graduate degree program as well bibliographical information as to where program and course descriptions may be found. It is instructive to examine in more detail an individual program.

At Syracuse University, with a total graduate student population of about 4400, three Computational Science degree options are available: A Master's degree, a Master's Level Certificate, and a Doctoral Level Certificate. The Master's degree program is open to any student with a B.Sc. degree who has taken calculus, discrete mathematics, and programming courses. 33 credits are required, 9 of which come from the three mandatory core courses:

CPS 615 *Introduction to Computational Science*

CIS 675 *Design and Analysis of Algorithms*

MAT 683 *Methods of Numerical Analysis I.*

The student must also take one or more of the following courses involving applications:

CPS 713 *Case Studies in Computational Science*

AEE/MEE 771 *Computational Fluid Mechanics*

NEU 651 *Computational Neuroscience.*

The remaining credits are taken from existing computer science, computer engineering, and mathematics. It is typical of many CSE programs that only two or three courses specifically targeted at CSE students are offered (designated

TABLE 1

U.S. Graduate School Enrollments in Computational Science and Engineering (CSE) and similar programs†

University	Total Grad Students	CSE Debut Year	1995 CSE Enrollment	1996 CSE Enrollment
* Austin	12000	1993	15	21
* George Mason	9700	1992	133	160
Illinois	9000	1994	59	65
* Minnesota	9000	1994	8	6
* Mississippi	2200	1991	18	18
North Carolina	5300	1990	61	?
* Princeton	1800	1984	20	20
Purdue	6500	1995	9	43
Rensselaer	2100	1996		50
* Stanford	7500	1988	40	48
* Syracuse	4400	1995	?	10
Utah	5000	1995	5	5

* Indicates stand alone degrees

† The information contained in this Table was obtained by informal survey only and is not intended to be complete.

CPS above in the case of Syracuse). Additional program objectives are usually satisfied by existing courses. In the case of Syracuse, a thesis is written and defended for the Master's degree. Graduate certificates are available to any graduate student who has an advisor in the Computational Science faculty and completes 15 credits of courses taken from a prepared list (such as CPS 615 and CPS 713). The programs are administered by the School of Computer and Information Science in the College of Engineering and Computer Science and involve faculty from computer science, computer engineering, mathematics, physics, chemistry, mechanical engineering, and neuroscience.

Some measure of the success of these programs is contained in Table I. Indicated are the total number of graduate students, the year each program started, and the total number of CSE students enrolled in each of the past two years. Several aspects are noteworthy. Nearly all of the programs were started within the past few years. In general, there is an encouraging and increasing interest by students. There appears to be little correlation between the popularity and the type of program offered (stand-alone degree or not). It is also clear that some programs can be extremely successful (e.g., George Mason and Rensselaer) or not (Minnesota). Representatives from several of the less popular programs suggested that poor advertising was the reason.

CANADIAN PROGRAMS

In contrast with the situation in the U.S., there has been relatively little effort by Canadian universities to initiate CSE programs. There are, however, several institutions which plan to offer such degrees and several already have somewhat related programs. (Most offer joint Science/Computer Science undergraduate degrees, but these appear to be very demanding and largely unpopular). As a result of an informal survey of Canadian physics departments, the following (probably incomplete) information has been revealed:

University of Prince Edward Island. Had an undergraduate speciality degree *Physics with Computing* between 1988 and 1992, but it was dropped due to a lack of interest.
Contact: D. Dahn (dahn@upei.ca).

Trent University. Has a new (since 1992) Masters degree in *Applications of Modelling in the Natural and Social Sciences* which is motivated by the observation that research of the future will have a strong multidisciplinary character. It has three primary teaching objectives: analytic modelling techniques, cross-fertilization of ideas, and adequate home-department training in the fields of psychology, chemistry, mathematics, biology, sociology, physics, economics, geography, computer science, and environmental science. The inclusion of non-traditional physical science projects (and students) in the fields of geography, psychology, sociology, and economics makes this program unique. Admission requires a solid foundation in calculus, some knowledge of linear algebra, as well as a computer language. Required courses include The Foundations of Modelling, Mathematical Aspects of Modelling, and Computational Aspects of Modelling. There are currently 12 students in the program.
Contact: A. Slavin (aslavin@trentu.ca)

Concordia. Graduate (Masters as well as Professional Masters) and undergraduate programs in *Applied Physics* were recently created as a result of Co-op student feedback. Industry needs were not being satisfied by traditional physics curricula. Both experimental methods and computational techniques are emphasized. The Professional Masters in Physics program is 16 months long and involves 8 courses in physics (including Computational Methods in Physics I and II), computer science, engineering, and commerce and administration. In addition, an 8-month work term is to be completed involving both written and oral presentations.
Contact: N. Eddy (eddy@alcor.concordia.ca).

Also at Concordia is the Computational Fluid Dynamics (CFD) Laboratory, associated with the Department of Mechanical Engineering. The CFD Lab has 7 full-time researchers with doctorates and carries out synergetic university-industry research, contractually or through consortia, with a number of prominent companies and organizations such as Atmospheric Environment Service, Applied Aeronautical Systems, CAE Electronics, Bombardier-Canadair, Bombardier-de Havilland, Chrysler, Fluid Dynamics International, Ford Motor, ICEM Engineering, Motorola, National Research Council, Pratt & Whitney Canada and others). Research activities include the application of finite element methods to the solution of large-scale aerodynamic problems and multi-disciplinary aspects such as aero-elasticity, aero-acoustics and aero-icing. Its applications range from domains such as external aerodynamics over aircraft and naval platforms to the internal aerodynamics of jet engines, as well as in-flight icing. There are currently 8 graduate students associated with the laboratory.
Contact: W.G. Habashi (habashiw@cfdlab.concordia.ca)

CERCA. The Centre de recherche en calcul appliqué was created through the initiatives of l'Ecole polytechnique, Concordia, University of Montreal, and McGill in 1992 and is under the responsibility of the ministère de l'Industrie, du Commerce, de la Science et de la Technologie. Its main objective is to promote the transfer of technology from universities to industry by gathering together researchers in the field of applied scientific computation. Currently, there is a staff of 9 full-time research professionals in addition to university professors, postdocs, and students who are associated with the center. The administrative council is composed of industry and provincial government representatives and university professors and deans. Eleven students have obtained graduate degrees in association with the centre. Industrial members include Astra, Bio-Méga/Boehringer Ingelheim Recherche, Brais, Martrès et associés, Merck-Frosst, Alcan International, Bombardier-Canadair, Ressources naturelles Canada, GEC Alsthom l'Electromécanique, GE Canada, Hydro-Québec (Groupe équipmement), Pratt & Whitney Canada, and Environment Canada. Current research areas include the study of turbulence, fluid flow through turbines, thermophysical processes, ocean and atmospheric circulation, pharmaceutical chemistry, and combustion.
Contact: A. Biron (biron@cerca.umontreal.ca)

Universities of Ottawa and Carleton. An undergraduate degree with a *Computational Physics* option will be offered in the autumn of 1997 at the University of Ottawa. It will focus on two new Computational Physics courses. One involves deterministic numerical solutions of Newton's, Maxwell's, and Schrodinger's equations, Molecular Dynamics and non-linear dynamics. The other concentrates on stochastic methods, random numbers, Monte Carlo simulations, percolation and data analysis. The joint Ottawa Carleton Institute for Physics is presently seeking approval for a masters Co-op program in *Physics in Modern Technology* whose objective is to make the graduate more "marketable" to industry by providing coursework and practical experience in computation and instrumentation. Students will take two core courses (Computational Physics and Computer Simulations in Physics) and four additional electives from a prescribed list of existing physics courses. In addition, a work term will be completed at an area industrial or

government laboratory site, involving both a written and oral report.
Contact: R. Hodgson (hodgson@physics.uottawa.ca).

Memorial University of Newfoundland is seeking approval for a *Multidisciplinary M.Sc. Degree Programme in Computational Science* much like the CSE programs offered in the U.S. Its objectives are to train students in state-of-the-art numerical methods, high performance computing, use of software development tools for parallel and vector computers, use of graphics, visualization and multi-media tools, the acquisition, processing and analysis of large experimental data sets, and the application of these techniques to at least one scientific area within the participating departments of chemistry, computer science, earth science, mathematics and statistics, and physics and physical oceanography. The program will be flexible in that a student may choose a thesis option (with 5 courses), a project option (with 8 courses), and also a Co-op option, in thesis or non-thesis form. A core of three courses will be required: Concurrent Programming, Introduction to Computer Systems, and Numerical Algorithms. In the proposed multidisciplinary environmant, students will gain an exposure to a broader range of science than that offered by traditional M.Sc. programs.
Contact: J. Lagowski (jolantal@smaug.physics.mun.ca)

Sherbrooke. A proposal for a "Diplôme" (professionnal masters) degree in *Calcul Scientifique Appliqué* with multidisciplinary aspects similar to the proposal at Memorial was recently rejected by the Council of the Faculty of Science due to budgetary considerations.
Contact: A.-M. Tremblay (tremblay@physique.usherb.ca)

CONCLUSION

Although the multidisciplinary CSE programs in the U.S. and Canada described above involve a variety of physical science departments, it is not too surprising that most of the initiative is coming from physics. This can be explained by the growing awarness of the vulnerablity of financing for fundamental, non-strategic, research activity as well as the fact that many physicists are well-trained in the general techniques of modeling and numerical calculation that industry seeks in an effort to improve competitiveness. The usefulness of physicists to society perhaps needs a little re-focussing, and a lot of marketing.

ACKNOWLEDGEMENTS

I wish to thank A. Caillé and especially J. Robin for their help in finding the sources for this article, as well as all those who took the time to provide information on the individual academic programs.

REFERENCES

1. D.E. Stevenson, Communications of the ACM, Dec. 1994, p.85.
2. J.R. Rice, IEEE Computational Science and Engineering, Winter 1995.
3. F. Santosa, IEEE Computational Science and Engineering, Summer 1996.
4. T.L. Marchioro, D.M. Martin, and W.D. Payne, IEEE Computational Science and Engineering, Fall 1995; W.D. Payne and T.L. Marchioro, *ibid* Winter 1995.
5. R. Bramley, IEEE Computational Science and Engineering, Summer 1994.
6. R.M. Panoff, IEEE Computational Science and Engineering, Winter 1995.
7. V. Meiser, M.R. Strayer, and C. Edward, IEEE Computational Science and Engineering, Winter 1994; also see http://csep1.phy.ornl.gov/csep.html and http://nacphy.physics.orst.edu/.
8. http://www.cray.com/PUBLIC/HPC/ind/univ/Comp-Sci-Paper.html.
9. G.V. Wilson, IEEE Computational Science and Engineering, Summer 1996; E.R. Jessup and R.C. Giles, *ibid,* Fall 1996.
10. *Hypertext markup language* for creating your own www homepage.
11. A popular scientific word processing language.
12. An intelligent (i.e., fast) method of compliation, using modular (e.g., lots of subroutines) programming.
13. *Practical extraction and report language*, is a computer language (like Fortran) for manupilating and presenting large data files.
14. A list of other useful texts may found at http://www.physics.orst.edu/~rubin/rubin/booklist/html.
15. J.R. Rice, IEEE Computational Science and Engineering, Spring 1994. Also see: R. Bramley and E. Gallopoulos, *ibid*, Summer 1994.