Parallel Computing and Education

Geoffrey Fox

CRPC-TR91123
April 1991

Center for Research on Parallel Computation
Rice University
P.O. Box 1892
Houston, TX 77251-1892

Parallel Computing and Education

Geoffrey C. Fox

Syracuse University
Northeast Parallel Architectures Center
111 College Place
Syracuse, New York 13244

gcf@nova.npac.syr.edu

I. Introduction

In this issue we have learned that parallel architectures will lead to computers of dramatically greater performance and capability. How and why will this affect education? Will the impact be incremental, as happens when one discovers a new elementary particle, and this appears as an extra chapter in a multivolume book on introductory physics? Or will the impact be dramatic, as in biology where areas such as genetic and molecular biology have revolutionized the core knowledge taught in the field? I believe the latter is a better analogy, and that as educators adopt the use of parallel computers, these machines will lead to major changes, both in the way we teach, and in what we teach. Many agree with this assertion, and a few have started to implement its consequences.

Educators can address many educational challenges with parallel computers, and through realistic simulations, help scientists explain their predictions and discoveries. For example, I was impressed by a set of talks about global climate change at a recent conference. Each participant agreed that major, and probably undesirable changes in the environment were inevitable, unless strong action is taken soon. They may or may not have agreed that parallel computers will accurately predict the temperature rises coming from global warming. However, no one could see how to explain to the public, and in particular, to politicians, the urgency of the situation. The key, I believe, lies in widespread implementation of parallel computers in science and education.

II. Parallel Computers in Education

Many children now have substantial exposure to computers in schools (K–12), and even more "benefit" from the VLSI revolution with specialized video games, such as those from Nintendo. Parallel computing will allow us to bring the sophistication of a military flight to the individual's video game. In this way, tomorrow's video games could teach our children more than hand-eye coordination, and be truly educational. Parallel computing will bring realism to simulations at the low end in hand-held units, and at the high end in tomorrow's theme parks, where exhibits could be controlled by a parallel teraflop supercomputer. This performance will allow not only realistic simulations, but wonderful graphics experienced perhaps

with "virtual reality." Hopefully, this realism will be used to teach about the earth as it is and not just about idealized worlds in galaxies far away in space and time.

THIS PROBABLY NEEDS TO BE EXPANDED

It is clear that computation must join theory and experiment as a fundamental research methodology. This cannot happen easily within our current educational system, and fundamental changes, which will challenge our universities, community colleges, and schools in the next decade are needed. Computers are now used in schools, and as discussed earlier, educators will need to adopt them further as an innovative aid to teaching.

However, if we look at the teaching of computation, we see a different picture. Non-Computer Science faculty view the computer as a useful but rather tiresome tool—best regarded as a black box programmed by graduate students and junior researchers. Computer scientists often see the use of computers as grubby numerical work outside of the core of the field, and centered on the elegant mathematics of idealized machines. Defining computation as the use of computers, we see that this is usually taught as a technical skill, but not studied as part of the academic mainstream. Computational science, i.e., the science of computation, falls into an academic void between computer science and such fields as chemistry and physics that make use of computers. We can understand how this situation developed; the basic design and programming methodology for sequential computers has remained unchanged for thirty years. There have been improvements, such as time-sharing, interactive computing, desktop workstations, UNIX, and so on. However, these did not change the fundamentals and could well be considered as technical skills. The supercomputers pioneered by Cray introduced some important new concepts, such as vector processing, but these were not pervasive. Indeed, if computational science had been a vital academic discipline fifteen years ago when the first Cray-1 was introduced, I believe that vector (super)computing would have advanced much faster. As it was, computational science was smothered by industry standards—the IBM 370 architecture, DEC's VAX, and operating systems such as VM, VMS, and UNIX. These promoted the view of computation as a technical skill and upheld the gap between computers and users. The increasing performance of computers with new architectures, especially parallel machines, makes it clear that computational science is an important, fundamental, and useful field; it should be taught as such.

The discussion of parallel computers in this issue has shown that these computers are effective because many important computational problems have an intrinsic parallelism. For example, simulations of the physical world can naturally exploit parallel machines. Clearly, an understanding of both parallel computers and parallel problems is helpful, and sometimes essential in developing the software and numerical algorithms for the so-called "grand challenges." These are the computer simulation applications that have been identified as the major goals of the Federal High Performance Computing and Communications Initiative.

III. Computational Science and Engineering

The dramatic performance promised by parallel computers will change the nature of science and engineering in research and in practice. Clearly, an interdisciplinary education in computational science will allow scientists and engineers to perform better. Those who understand the basic principles of computer architecture and modern software techniques will be the leaders in using the first teraflop machines; indeed, they will help the computer industry design machines that will perform well on the grand challenges in physics, chemistry, aeronautics, etc.

A training in computational science could include the basics of applied computer science, numerical analysis, and simulation. Although the details of parallel computing are changing rapidly, the graduate of such an education now will be able to track future changes. Computational science naturally links scientific fields to computer science. Here again, a specialization in computational science is an attractive option for computer scientists. An understanding of applications will allow computer scientists to develop better hardware and software. Computational scientists, whether in computer science or in an application field, will benefit directly from technology that improves the performance of computers by a factor of two each year. Their theoretical colleagues will not be assisted as well by technologic improvements, and so computational science is expected to be a field of growing rewards and opportunities, as compared to traditional areas.

I believe that students educated in computational science will find it a rewarding and exciting experience, which should give them excellent job opportunities. Only a few universities offer such a degree, however, and often, only at the Ph.D. level. The NSF Supercomputer Centers at Cornell, Illinois, Pittsburgh, and San Diego have played a major role in enhancing the visibility and progress of computational science. However, these centers are set up outside of the academic framework of universities, and do not contribute directly to developing computational science as an academic area. In particular, the best computational scientists do not easily get faculty positions in universities. This is, of course, the major reason that there are so few educational programs in this area. While at Caltech, I supervised several top-quality Ph.D. students who were really computational scientists, although their formal degrees were usually in physics. These students received many good job offers from industry and the national laboratories, but were not given opportunities for faculty positions comparable to those of Caltech students in more traditional areas. This is clearly a significant deterrent to good students entering the field. It will not be an easy issue to address, and I expect that only slow progress will be made as computational science gradually gains recognition as a fundamentally exciting field. The inevitable dominance of parallel computing will help, as will the use of parallel computers in the NSF centers that have provided such a critical stimulus for computational science. Most important will be initiatives from within universities to hire, encourage and promote new faculty, and educate students in computational science.

Consider the issues controlling the development of computational science in universities. As this field borrows and extends ideas from existing fields—computer science, biology, chemistry, physics, etc.,—it will naturally face campus political hurdles as it challenges traditional and firmly held beliefs. These inevitable difficulties are exacerbated by administrative problems; many universities are facing a scenario of no growth, or even of declining funding and faculty size. This will mean that creation of new areas implies reductions in other areas. Computational science shares difficulties with other interdisciplinary areas, such as those associated with the growing interest in Planet Earth. The peer referee system used in the hiring and promoting of new faculty is perfect for ensuring high standards within the referees' domain of expertise. This tends to lead to very high-quality, but isolated departments that find it hard to move into new areas. The same effect is seen in the peer review system used for the refereeing of scholarly papers and federal grants. Thus, universities find it hard to change, and so computational science will not grow easily in academia. A key hurdle will be the development of some consensus in the community that computational science is, as I have asserted, fundamental and existing. This needs to be quantified academically with the development of a core curriculum—a body of knowledge on which one can build computational science as an academic discipline.

IV. Developing and Supporting Computational Science Education

There are two obvious approaches to filling the academic void identified as computational science. The boldest and simplest approach is to create an entirely new academic degree, "Computational Science," administered by a new university department. This would give the field great visibility, and once created, the independent department would be able to develop its educational program, research, and faculty hiring without direct interference from existing academic fields. Such a department would need strong support from the university administration to flourish, and even more so for its creation. This approach would not be easy to implement. There would be natural opposition from existing academic units for both good and not so good reasons. A telling critic could argue that a free-standing Computational Science program is premature; there is as yet no agreement on a core body of knowledge that could define this field. Students graduating from this program might find it hard to progress up the academic ladder at the vast majority of universities that do not have such a department.

These difficulties are avoided by the second strategy for computational science, which rather than filling the void with a new department, would broaden the existing fields to "meet in the middle." Students could graduate with traditional degrees and have a natural academic future. This is the approach taken by the existing university Computational Science and Engineering programs (surveyed later in this article). For example, consider the two fields of chemistry and computer science. A computational scientist would graduate with either a Chemistry or Computer Science degree. Later academic progress would be judged by the scientist's contributions to the corresponding base field. I have already argued that such an interdisciplinary education would allow the student to be a better chemist or a better computer scientist, respectively. Naturally, the chemistry graduate from the Computational Science program would not have received as complete an education in chemistry. Some of the chemistry elective courses would have been replaced by computational science requirements. This change would need to be approved and evaluated by the Chemistry faculty, who would also need to identify key chemistry requirements to be satisfied by computational scientists. New courses might include computational chemistry and those covering the basics of computer science, numerical analysis, and simulation. The latter set would be taught either by computer scientists or interdisciplinary Computational Science faculty. The education of a computational scientist within a Computer Science department could be handled similarly. This would have an emphasis on applied computer science, and a training in at least one application area.

In this scenario, a degree in Computational Chemistry is equivalent to one in "Chemistry within the Computational Science program." On the computer science side, one could see degrees in "Computer Science with a minor in Chemistry," or a "Ph.D. in Computer Science with a master's degree in Chemistry." At the academic level, we see an interdisciplinary program in computational science, but no separate department; faculty are appointed and students admitted to existing academic units. This approach to computational science allows us to develop and understand the core knowledge curricula in an evolutionary fashion. Implementing this more modest plan is certainly not easy, as one must modify the well-established degree requirements for the existing fields, such as chemistry and computer science. These modifications are easiest at the master's and especially at the Ph.D. level, and this is where most of the new programs have been established.

These seem to be very good reasons to establish undergraduate level Computational Science programs. We also need to create an awareness in the (K–12) educational system of the importance of computation, and the possibility of Computational Science degrees. Thereby, more high school students will hopefully choose Computational Science educational programs and careers.

To conclude, some of the existing Computational Science programs are surveyed below. They show a rich diversity indicative of an emerging field.

SYRACUSE UNIVERSITY

Contact: G. FOX
gcf@nova.npac.syr.edu
(315) 443-1723

Syracuse intends a university-wide Computational Science program involving many departments, initially implemented by broadening the existing degree programs. A unique feature is the plan to offer undergraduate, master's, and Ph.D. degrees. Initial plans are to offer admission into a Computer Science-based program for the semester starting in January, 1992. This Computational Science degree involves, at the undergraduate and master's level, requirements that include 50% of the traditional computer and information science degree. The application and new applied computer science courses that will complete the program are still under discussion.

For further information, contact: Professor Geoffrey Fox, Director, NPAC, 111 College Place, Syracuse University, Syracuse, New York 13244-4100.

CALIFORNIA INSTITUTE OF TECHNOLOGY

Contact: T. PRINCE
tap@caltech.edu
(818) 356-6605

The Physical Computation and Complex Systems (PCCS) Program at Caltech focuses on aspects of computational science to be built on the Physics Ph.D. program at Caltech. It is designed for students with a strong undergraduate training in physics who are interested in a career in computational science.

The objective of this new Ph.D. program is a unified approach to abstraction, modeling, and computation applied to the natural world. This approach is based on a systematic use of physical analogies and methods. The program involves fundamental education in mathematical physics, simple classical and quantum physical systems, fundamental properties of complex systems, physical optimization methods, and the appropriate computational techniques needed for large-scale problem solving on advanced architecture computers.

Specific subjects of attention in the PCCS program include: the relation of information, complexity, and computation; the role of classical and quantum information in the history of the Universe; performance modeling with computers considered as complex systems; nonadaptive but nonlinear dynamical systems; fundamental general properties of adaptive complex systems, with examples of such systems being chemical, biological and cultural evolution, learning and thinking; computational techniques for use of high-performance, advanced-architecture computers in modeling, simulation, and data analysis; mathematical, algorithmic, and programming issues underlying such applications; the impact on problem solving of new approaches, such as genetic algorithms, symbolic methods, cellular automata, neural networks, and multiscale methods; visualization and interpretation of massive data sets generated by high-performance computers. This broad range of topics is unified by the use of analogies, methods, and approaches suggested by physical systems.

For further information, contact: Professor Tom Prince, Physics Department, California Institute of Technology, Pasadena, California 91125.

The following descriptions come from an article by Professor John Rice of Purdue in the January 1991 issue of *Computing Research News.*

THE UNIVERSITY OF MICHIGAN

Contact: W. MARTIN
lasc_info@um.cc.umich.edu
(313) 936-3130

The University recently approved a Doctoral Program in Scientific Computing. The joint-degree program allows students to pursue their doctoral studies in a home department—typically one of the traditional engineering, science or mathematics disciplines—and take additional courses in areas such as numerical analysis, scientific computation, applications or the study of algorithms for advanced computer architectures. This interdisciplinary program is intended for students who will make extensive use of these subjects in their doctoral studies.

This program recognizes a firm knowledge of the science is an essential ingredient for scientific computation research. Students are expected to complete the normal doctoral requirements for their home department, as well as additional course requirements in scientific computation, numerical analysis and algorithms for advanced computer architectures. The title of the degree will have "... and Scientific Computing" appended to the title, such as a Ph.D. degree in Aerospace Engineering and Scientific Computing.

The Laboratory for Scientific Computation, in cooperation with the student's home department, administers the doctoral degree program in scientific computing. The following list of research topics is representative of the various activities in scientific computation available for prospective doctoral students: computational fluid dynamics; algorithms for advanced computer architectures; computational particle transport; computational solid mechanics; simulation of semiconductor devices; simulation of AIDS transmission; simulation of VLSI circuits; scientific visualization; high-performance materials; molecular dynamics; computational chemistry and computer-aided molecular design.

For further information, contact: Professor William R. Martin, Director, Laboratory for Scientific Computation, The University of Michigan, Ann Arbor, Michigan 48109-2104.

NORTH CAROLINA STATE UNIVERSITY

Contact: R. FUNDERLIC
ref@adm.csc.ncsu.edu

Well-known national influences and initiatives, strong local institutional support, and excellent faculty from several departments have propelled high-performance computing research and teaching at North Carolina State University. Several shared-memory and message-passing parallel computers are available on campus for researchers and graduate students, and a Cray Research, Inc. YMP/4 is at the North Carolina Supercomputer Center in Research Triangle Park. This center is managed by the Microelectronics Center of North Carolina, and Cray, and is available via a high-speed educational network connecting more than a dozen campuses. The Center for Research in Scientific Computation (a joint effort between Computer Science and Mathematics) acts as a focal point for academic programs in scientific-computing degree programs. The Computer Systems Lab (a joint effort between Computer Science and Computer Engineering) provides strong computational infrastructure support.

Academic programs at North Carolina State include Computational Mathematics (CMA) within the mathematics department, Scientific Computing (SC) within computer science and Computational Engineering and Science (CES). The latter provides a well-structured, expanded, split minor in math and computer science, and is available in all engineering and physical science graduate programs. Computer science is a vital component of scientific-computing research and teaching at NC State. Of the 23 courses supporting the CES program, 18 are computer science, and eight of those courses are also listed with mathematics.

The SC and CMA programs are very similar, and lead to master's and Ph.D. degrees in Computer Science and Applied Mathematics. The CES program replaces the minor requirement and is available in all engineering and science departments at the master's and Ph.D. levels. With proper advising, a *de facto* scientific computation track is available within the Computer Science undergraduate program.

Many universities will wrangle over the role of computer science in scientific computing. Important questions include: Are we talking about something that is merely a subset of computer science and engineering? Is computer science willing to become more service-oriented? What faculty administers the programs? Who administers the qualifying exams? Can prerequisites for computer science graduate courses be accommodated? Will some colleges or departments draw lines to exclude others or mitigate cooperation?

We expect North Carolina State's success will be influenced strongly by the cooperative efforts of the Computer Science and Applied Mathematics faculties, despite the fact they are in different colleges. North Carolina State has a superior reputation in interdisciplinary research.

Complete descriptions of the CES, CMA and SC programs are available from: Professor Robert E. Funderlic, Department of Computer Science, North Carolina State University, Raleigh, N.C. 27695-8206.

RICE UNIVERSITY

Contact: D. SORENSEN or R. TAPIA
sorensen@rice.edu, or rat@rice.edu

Because of the rapid increase in computing power over the past decade, modern science and engineering have become increasingly reliant on computation as an aid to research, development, and design. Indeed, one can hardly imagine a large-scale engineering project that will not call upon some aspect of the mathematical and computational sciences. However, using the newest and most powerful computers requires a knowledge of parallel and vector capabilities, and a variety of other things, such as visualization, networking and programming environments. In addition, newly developed algorithms and analytic techniques enhance the power of these computational tools. Despite the obvious relevance of these techniques to science and engineering, Rice has no focused program in this area, providing specialized training in the use of high-performance computing technology.

To address this deficiency, the Mathematical Sciences Department, in conjunction with the Computer Science, Chemical Engineering and Electrical Engineering Departments, intends to initiate a new degree program leading to advanced degrees in Computational Science and Engineering (CSE). The program focuses on modern computational techniques and provides a resource of training and expertise. The program is designed to provide this training throughout the university at the master's and Ph.D. levels.

The program is governed by a faculty committee selected by the Dean of Engineering, with the Provost given ultimate oversight. This Computational Science Committee (CSC) is responsible for assisting the student in designing an appropriate course of study, setting examination requirements and ensuring the integrity of the degree program. The CSC will not be a new department, but rather, a mechanism for initiating the interdisciplinary research required to advance computational science.

The professional master's degree is intended to produce a scientific computing expert who can be a member of an interdisciplinary research team. A recipient of this degree will be well-trained in state-of-the-art numerical methods, high-performance computer architectures, use of software development tools for parallel and vector computers, and application of these techniques in at least one scientific or engineering area. The curriculum for this degree consists of a variety of topics, including mathematical sciences, computer science and a selected application area.

Requirements include successful completion of 30 semester hours or more of advanced courses. There is no thesis requirement. The program of study will be designed by the student with advice and approval of the CSC.

It has been possible to construct this program from existing courses with one exception. A new course, Introduction to Computational Science, is the central, core course of the program. This one semester course is an introductory survey of the topics making up the scientific computing program. The course is designed to help students appreciate the scope of the program, so they will be better prepared to design an effective, individual selection of other courses. The class also introduces engineering and physical science students to state-of-the-art scientific computing technology.

The Ph.D. program starts with the advancement to doctoral candidacy after the student successfully completes approved program course work and performs satisfactorily on preliminary and qualifying examinations. The foreign language requirement of the student's department will not change. The student must complete an original thesis under the direction of a member of the participating faculty of the CSE program. The thesis must be accepted by the Computational Science Committee.

For further information, contact: Professor Danny Sorensen or Richard Tapia, Department of Mathematical Sciences, Rice University, Houston, Texas 77251.

STANFORD UNIVERSITY

Contact: G. GOLUB
golub@patience.stanford.edu

For the last three years, Stanford has had a program for granting degrees in Scientific Computing and Computational Mathematics. The interdisciplinary program admits students and grants degrees. The purpose of this program is to train students in the use of modern advanced computer architectures and software tools in various fields of science and engineering. The main thrust is the fusion of ideas from computer science and applied mathematics with a number of application areas. The apparent waning interest and support for numerical analysis and scientific computing within the Computer Science Department helped spur creation of the program.

The Scientific Computing and Computational Mathematics program currently resides in the School of Engineering. Students are admitted directly into the program independent of other departments. The faculty is made up of teachers from other departments and features three levels of participation. The core faculty includes Joseph Keller, Joseph Oliger and Gene Golub, who are responsible for administration. The associate faculty consists of people heavily involved in computing within the program and affiliated faculty whose disciplines rely on computing to some extent.

Although the program was formally approved in 1987, the 1989–90 academic year marked the first year of operation. Ten of the 40 applicants entered the program. Most applicants were from engineering fields. However, applicants were from a variety of other fields, including medicine. At the start of the 1990–91 academic year, 18 students were enrolled in the program.

The curriculum emphasizes applied mathematics, numerical analysis and computer science and requires demonstrated expertise in some application areas, such as fluid mechanics. In addition to this, working relationships have been formed with local research institutes, such as RIACS, LLNL, and IBM.

More information is available from: Professor Gene Golub, Department of Computer Sciences, Stanford University, Stanford, California 94305.

UNIVERSITY OF CALIFORNIA AT DAVIS

Contact: G. RODRIGUE
rodrigue@lll-crg.llnl.gov

A Computational Science program was initiated within the Departments of Applied Science and Chemistry because computing has emerged as a third way of doing science, complementing the time-honored theoretical and experimental approaches. The computational approach to science has made significant contributions in several disciplines, including aerodynamics, meteorology and nuclear engineering, where previously intractable problems have been solved. Great promise and future growth lie in other disciplines, such as molecular biology, materials science, chemistry and physics. The possibilities opened up by the availability of high-speed computing have many similarities across scientific disciplines, so a science of computation does exist.

Questions of science, computational techniques, computer science and mathematics are inseparable in addressing the large issues in computational science. A practitioner of computational science must have some skills in each of these areas and be able to interact with researchers from all of these areas. U.C.-Davis established the Computational Science program with this philosophy in mind.

The university's program is designed for the graduate student interested in the application of computers to the physical, chemical, mathematical and engineering sciences. The program involves course work from the traditional areas of physics, chemistry, computational mathematics and computer science, as well as in the area of the student's specialization. Ph.D. candidates in participating departments declare a designated emphasis in computational science, then take a special set of core courses in the student's enrolled department and a set of core courses in computational science. For example, in the Department of Applied Science, the core courses are Mathematical Physics, Computational Mathematics and Computational Science, a course designed especially for physical scientists that covers such topics as computer architecture with emphasis on parallel computers, algorithms, and numerical methods. After passing a written/oral examination, the student begins graduate research by taking electives from a variety of courses available in the enrolled department. The degree awarded to the student is: "Doctor of Philosophy in (the department) with emphasis in Computational Science."

For additional information, contact: Professor Gary Rodrigue, Department of Applied Science, University of California at Davis, Davis, CA 95616.

THE UNIVERSITY OF ILLINOIS

Contact: A. SAMEH
sameh@csrd.uiuc.edu

The availability of powerful computers has made it possible to use computational methods in larger and broader areas of science and engineering. Yet, making effective use of such computers is difficult, partly due to the complexity of their architecture and system software. Ordinary users of these machines seldom realize performance near what experts can obtain. Even then, experts work hard to get reasonable performance results. This situation highlights the need for formalizing the education and training of students in the field of computational science and engineering. Unlike what is regarded now as traditional computer science, a CES program would attempt to focus on the whole computational process. A graduate program in CES has been established at the University of Illinois within the departments of Computer Science and Electrical and Computer Engineering. It covers the topics listed below.

- Computers: architecture for parallel and pipeline processing; simulation from the chip to the system level; hardware to the level of device simulation and packaging; and reliability and fault tolerance.
- System Software: compilers, especially restructuring source code and code generation; programming and problem solving environments; operating systems, including interface with compilers; and scheduling and dynamic control of systems.
- Applications: design of robust parallel numerical and non-numerical algorithms; and specialization in one application area, such as digital circuit simulation, computational fluid dynamics or computational chemistry, and development of application software achieving "performance portability" across a wide class of architectures.
- Performance Evaluation: measuring performance of existing and proposed architectures, compilers, algorithms and whole application codes; and analysis and performance improvement suggestions, and validation via measurements.

For more information, contact: Professor Ahmed Sameh, Department of Computer Science, University of Illinois, Champaign, Illinois 61801.