Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Future Generation Computer Systems 19 (2003) 1285–1292

FGCS
FUTURE GENERATION COMPUTER SYSTEMS

www.elsevier.com/locate/future

The computational science major at SUNY Brockport

Leigh J. Little

Department of Computational Science, SUNY Brockport, 350 New Campus Drive, Brockport, NY 14420, USA

Abstract

The field of computational science is a recent addition to academic study. While the content of such an education is generally agreed upon, effective methods for imparting this knowledge are still being investigated. This paper describes the current state of the computational science degree programs at SUNY Brockport and the successes that have been obtained. Issues relating to the implementation of such programs in the context of a small, liberal arts college are also discussed.

Keywords: Computational science; Education; Undergraduate; Graduate

1. Introduction

1.1. Motivation

Starting with the early development of computers, countless researchers have utilized computational methods for the investigation of scientific problems and in hindsight, the emergence of academic programs in computational science seems obvious. The notion of formally trained computational scientists is appealing to both industry and research laboratories. The problems that need to be solved nowadays often require individuals with multidisciplinary experience and that is precisely what computational science education strives to achieve.

Both the government and industry have expressed a need for highly trained computational scientists and have indicated a strong commitment to the development of academic programs necessary to produce such individuals. Some examples are the National Science Foundation Metacenter for Computational Science and Engineering (http://www.sdsc.edu/MetaScience) as well as the Department of Energy's Graduate Fellowship Program (http://www.krellinst.org/csgf) and Computational Science Initiative (http://www.seesar.lbl.gov/ccse/news/summary). In addition, recently formed non-profit organizations such as the Krell Institute (http://www.krellinst.org) and the Shodor Education Foundation (http://www.shodor.org) actively participate in the introduction of computational science in the classroom at all levels of education.

Many institutions offer areas of concentration or certification in specific disciplines involving computation at the undergraduate and graduate levels (see the compiled lists in Refs. [2,3,5]). Examples include the mathematics and computational science degree at Stanford University, the computational chemistry program at the University of Minnesota and the computational finance program at Carnegie Mellon University. Separate degree programs in computational science are comparatively new. In the fall of 1998, the State University of New York, college at Brockport (SUNY Brockport) began offering both undergraduate and graduate degrees in computational science. In

E-mail address: llittle@brockport.edu (L.J. Little).
URL: http://www.cps.brockport.edu/little.

0167-739X/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0167-739X(03)00086-4

particular, SUNY Brockport became the first institution in the United States to offer an undergraduate degree in computational science.

The desired outcomes of an education in computational science have been discussed in Refs. [1,4,6,7]; however, the most effective manner of achieving such goals, particularly at the undergraduate level, is still being examined. The purpose of this paper is to provide specific details regarding the implementation and success of the undergraduate and graduate curricula of the Computational Science Department at SUNY Brockport.

1.2. The Computational Science Department

SUNY Brockport is a selective, liberal arts college with approximately 6760 undergraduate and 1870 graduate students. It is located 20 miles west of Rochester and 40 miles east of Buffalo. The college has a historical role of supplying highly trained mathematics and science teachers to Rochester area high schools.

Recent trends in technology, both in education and industry, led to an increase in computer science enrollments. Because of the proximity to a large number of technologically based industries (Kodak, Xerox, Lockheed-Martin, Corning, Bausch & Lomb) and the need for computational scientists, a computational science program was created in the fall of 1998. It became a full-fledged department in the fall of 2000.

The department was established to achieve the following objectives: (a) to be a national leader in curriculum development of computational science education; (b) to successfully integrate the three main components of computational science (computer science, applied mathematics and application sciences) into a cohesive, interdisciplinary program of study; (c) to boost enrollment in the college by creating an interest in the study of science and technology [8].

The department currently has three full-time faculty members that have obtained undergraduate degrees in chemistry, mechanical engineering, chemical engineering and physics and advanced degrees in physical chemistry, numerical analysis, computer science, physics and nuclear physics. The current research interests of the faculty are in parallel and cluster computing, molecular simulation, nanotechnology and combustion physics.

An essential component in the educational process is access to adequate computational resources. Students and faculty in the department have access to a wide array of hardware and software. The main campus computing site houses four computer-aided instruction (CAI) labs with 25 Pentium-4 based PCs each, a UNIX laboratory containing 25 SUN terminals (SunRay front-ends into a Sparc 4500 server) and a general use area containing 200 PCs and 30 Macintosh computers. In addition, there are 11 satellite laboratories (PC and Macintosh) spread throughout the campus that are networked with the main site.

The Computational Science Department also has its own computing laboratory that is mainly used to house more powerful computing hardware. This consists of a 4-processor SGI Onyx2 workstation, three SGI O2 workstations, and a 16-processor Beowulf cluster. The department also has two Intel Paragon parallel computers that were donated by Intel.

There are many software packages available for student and faculty use. PC-based applications include MATLAB, Compaq Visual FORTRAN, SIMPROCESS, Advanced Visualization System (AVS), Origin and HyperChem. Commercial UNIX tools include the SGI compiler suite, Mathematica, MATLAB, CHARMM and AMBER. Other available UNIX software tools are GPL or free-for-educational-use applications such as ScaLAPACK, PETSc, Octave, SPARSKIT2 and PSPARSLIB.

The department also has access to the use of the computational facilities at other institutions and computing laboratories including the Center for Computing Research (CCR) at SUNY Buffalo, SUNY Stony Brook and the National Center for Supercomputing Applications (NCSA) at the University of Illinois, Champaign-Urbana.

1.3. Implementation of the degree program

The implementation of a degree program in computational science aimed at achieving the goals stated in Section 1.2 presents numerous challenges. In the case of SUNY Brockport, many of these stem from the nature of a small liberal arts college and the limited resources that are present in such an environment.

Because SUNY Brockport is a traditional liberal arts college, the main focus in all undergraduate programs is in providing a well-rounded education. In particular, the number of so-called general education courses comprises a higher proportion of the degree than what would be experienced at a larger school. Of the 120 h required to complete a BS degree, approximately 75 of these hours are general education courses. This leaves only 45 h for the core courses in the major field of study. This is in direct contrast with what is encountered at other institutions and places unusual limits on what a degree program can contain. At first glance, the BS degree requirements given in Section 2.1 indicate potential deficiencies. For example, the computer science component does not address advanced topics such as data structures and object oriented programming while the applied mathematics portion does not include differential equations. This is not the case, however, because the emphasis in computational science education is in the practical knowledge of the component disciplines [8]. Thus it is not necessary to have an exhaustive background in these areas.

Another difficulty that arises is the high faculty course loads resulting from having so few members. The need to frequently develop new courses specific to the discipline adds to the faculty workload. Most of the departments at SUNY Brockport have fewer than seven full-time faculty members, most of whom carry a load of 9 semester hours. One aspect that alleviates a portion of the high course loads is that the Computational Science Department is fully independent and not a program in another department or a merging of resources from several departments. As a result, most of the prerequisite material is taught elsewhere in the college. This allows time for departmental faculty to concentrate fully on the development of new course material.

Given the interdisciplinary nature of computational science and the small size of the department, the diversity of faculty backgrounds is essential to the educational mission of the department. Furthermore, the degree of experience present is equivalent to what might be found in larger departments. For example, only two faculty members possess a formal education in computer science and mathematics; however, all members have extensive experience in applying computers to investigate problems in the physical sciences and engineering.

History reveals that small faculty size has not been an obstruction to the success of undergraduate programs at SUNY Brockport. For example, the Physics Department has only three faculty members, yet it has long held a reputation for providing highly qualified secondary education teachers in the entire upstate New York area. Similar statements apply equally to our Chemistry Department (five members) and Earth Science Department (six members).

Another difficulty that frequently arises in small colleges is the potential for limited computational facilities. As outlined in the previous section, the current department hardware and software resources in this area are appropriate given the size of the student enrollment (see Section 3.1) and are comparable to what is available in other programs. Establishing the department been a difficult and time-consuming process, but it is felt that the limitations posed by a small college environment has not presented obstacles that could not be overcome. In addition, the recent national emphasis in producing computational scientists will result in the formation of new programs at colleges with resources similar to those at SUNY Brockport. It is important to explore what can be achieved in such an environment.

2. Computational science degree programs

2.1. Undergraduate (BS) degree

One aspect of computational science that is certain is the need to frequently revise and update course content. New software tools and languages are being developed at a rapid rate and it is a challenge to determine which tools will prove most beneficial to student success. The entire faculty is active in the development of the curriculum at all levels and also participate (both as attendees and instructors) in computational science education workshops offered by the National Computational Science Institute (http://www.computationalscience.net). The availability of such resources helps to expand knowledge of available tools and pedagogical techniques, and are a tremendous benefit to development and improvement of the program.

In addition to courses drawn from mathematics and computer science, the undergraduate degree in computational science consists of departmental courses specifically designed to address the seven core areas shown in Table 1. The degree requirements are shown in Table 2.

Table 1
Core areas in computational science with desired student learning outcomes

Number	Area	Student learning outcomes
I	Computational tools	Programming languages, libraries, operating systems, software packages
II	High performance computing	Parallel architectures, parallel programming, benchmarks, code optimization
III	Applied and computational methods	Numerical analysis, grid generation
IV	Simulation and modeling	Modeling of problems from a variety of disciplines
V	Visualization	Techniques and tools
VI	Application sciences	Computational physics, chemistry, biology, mechanics, finance
VII	Communication	Effective written and oral communication

Table 2
Undergraduate computational science major degree requirements

Areas and courses	Credits
Mathematics courses	9
Calculus III (MTH 203), three credits	
Elementary statistics (MTH 243), three credits	
Linear algebra (MTH 424), three credits	
Computer science courses	4
Fundamentals of computer science (CSC 203), four credits	
Computational science courses	15
Computational tools I (CPS 201), three credits	
Computational tools II (CPS 202), three credits	
High performance computing (CPS 303), three credits	
Simulation and modeling (CPS 304), three credits	
Applied and computational mathematics (CPS 404), three credits	
Application science and elective courses	18
200-level and higher courses in areas of concentration such as	
Embedded computing (CPS 411), three credits	
Computational chemistry (CPS 417), three credits	
Computational physics (CPS 421), three credits	
Scientific visualization (CPS 433), three credits	
Computational finance (CPS 441), three credits	
Computational fluid dynamics (CPS 455), three credits	
Computational biology (CPS 461), three credits	
Independent study (CPS 499), three credits	
Prerequisites	15
Calculus I and II (MTH 201 and 202), six credits	
Discrete mathematics I (MTH 281), three credits	
Introduction to computer science (CSC 120), three credits	
Introduction to computational science (CPS 101), three credits	
Total credits	61

Students are first introduced to computational science in introduction to computational science (CPS 101). This class also satisfies a college general education requirement and as a result, students from other scientific and technical disciplines also enroll. Several CPS 101 students have discovered that they enjoy the field of study and have decided to add computational science as a dual major. Approximately one-third of our undergraduate students are dual majors, usually in mathematics or computer science, and two students have declared triple majors in all three.

The first core courses that students encounter are computational tools I and II (CPS 201 and 202). These are typically taken during the second year. These courses are taught almost entirely in a laboratory setting.

In keeping with the evolving nature of our program in computational science, the content of these courses has evolved since their inception. Original course design called for the following tools to be studied: FORTRAN 77, FORTRAN 90, LATEX, Mathematica, effective communication, UNIX fundamentals, modeling and basic parallel computing. It was decided that students needed greater focus on a smaller subset of skills and the content was modified by replacing Mathematica with MATLAB while LATEX, FORTRAN 77, modeling and parallel programming were dropped entirely from the content. Since parallel programming and modeling are covered more extensively in later courses, elimination of these topics at this stage is not detrimental.

In their third year, students usually take simulation and modeling (CPS 304) and high performance computing (CPS 303). CPS 303 emphasizes basic parallel computing hardware and programming paradigms. Architecture models from the basic von Neumann SISD computer up to distributed and shared memory MIMD computers are discussed as are problem decomposition and thinking in parallel. Programming is performed in a distributed memory environment using MPI with applications from areas such as numerical differentiation and integration, linear algebra and graph theory. In addition, students receive experience in operating in a batch processing environment and in obtaining reliable benchmark results for evaluating parallel performance.

CPS 304 is the first concentrated experience students obtain in simulation and modeling. Discrete event simulation is covered because no knowledge of physical principles is necessary; only a basic knowledge of statistics is required. Students are later exposed to deterministic and stochastic methods.

The final required class is CPS 404. This is a numerical methods course that covers numerical integration, differentiation, ordinary differential equations, interpolation, fixed point methods, the fast Fourier transform and random number generation. Programming techniques and implementation for these methods is also covered. While substantially more theoretical than the other courses, the main emphasis is on practical methods.

The BS degree also requires six elective courses. Students with interests not covered by courses within the department are allowed to substitute up to three of these courses with material from other disciplines. The department elective courses include upper-level applications courses which were created in order to give students more concentrated study in an application area of interest. This is the area where the diversity of the faculty backgrounds becomes critical. The knowledge and experience of the faculty admits a concentration on the most effective methods and provides a very enriching experience for the students.

All 400 level courses are taught in a swing-level environment; both undergraduate and graduate students attend the same section. Graduate students are required to do additional homework and projects and are graded on a more strict standard than undergraduate students. This environment is advantageous because the department offers a total of 18 courses during the academic year. It is the only way that the upper-level courses can be offered frequently enough to allow students to graduate within a 4-year period. Also, the different grading standards has resulted in an unanticipated side effect.

Table 3
Undergraduate computational science minor requirements

Areas and courses	Credits
Computational science courses	12
Computational tools I (CPS 201), three credits	
Computational tools II (CPS 202), three credits	
High performance computing (CPS 303), three credits	
Simulation and modeling (CPS 304), three credits	
Application science and elective courses	6
200-level and higher courses in areas of concentration	
Total credits	18

Undergraduate students, particularly those who intend to obtain an advanced degree, will generally perform their work at the level of the graduate students.

The department has contacts with local technical industries (Kodak, Xerox, Bausch & Lomb). Many students opt to do an internship at one of these companies, for which they receive credit as an independent study course. Students having a specific interest not addressed by an existing course may also take an independent study course with a faculty member serving as advisor.

Finally, the department offers a minor in computational science. The requirements for the minor are given in Table 3.

2.2. Graduate (MS) degree

The graduate degree program (see Table 4) is designed to stress practical computing and applications. The core courses in the graduate program are scientific visualization (CPS 533), advanced computational tools (CPS 602) and supercomputing and applications (CPS 644).

Scientific visualization stresses the presentation and analysis of application data using visual means. Topics include fundamental graph types, animation, GUI design and detection of errors in computed data. The tools employed here are MATLAB (for high end graphics and animation) and the X-Windows environment (for low end graphics).

The emphasis in advanced computational tools is on the use of tools for large-scale, practical problems from engineering and the sciences. The main

Table 4
Graduate computational science major requirements

Areas and courses	Credits
Mathematics courses	3
Discrete mathematics II (MTH 581), three credits	
Computer science courses	4
Algorithms and data structures (CSC 506), four credits	
Computational science courses	16
Scientific visualization (CPS 533), three credits	
Advanced computational tools (CPS 602), three credits	
Supercomputing and applications (CPS 644), three credits	
Graduate seminar (CPS 698), one credit	
Independent study (CPS 699), three credits	
Project paper (CPS 700), three credits	
Application science and elective courses	12
500-level and higher courses in areas of concentration such as	
Applied and computational methods (CPS 504), three credits	
Embedded computing (CPS 511), three credits	
Computational chemistry (CPS 517), three credits	
Computational physics (CPS 521), three credits	
Computational finance (CPS 541), three credits	
Computational fluid dynamics (CPS 555), three credits	
Computational biology (CPS 561), three credits	
Computational methods in physical sciences (CPS 604), three credits	
Deterministic dynamical systems (CPS 632), three credits	
Stochastic dynamical systems (CPS 633), three credits	
Statistical methods I and II (MTH 541/542), three credits	
Differential equations (MTH 555), three credits	
Math modeling I and II (MTH 561/562), three credits	
Numerical analysis (MTH 571), three credits	
Total credits	35

topics are parallel computing libraries and network problem-solving environments. Issues arising in code optimization, both by hand and through compiler options, are also covered.

Supercomputing and application stresses parallel programming development and covers advanced topics such as communicators, topologies, non-blocking communication, parallel data structures and parallel numerical methods.

Table 5
Student enrollment and graduation rates

AY year	Undergraduate	Graduate	Graduated (undergraduate/graduate)
1998–1999	13	7	0/0
1999–2000	18	9	0/1
2000–2001	18	10	3/2
2001–2002	19	13	3/3
2002–2003 (projected)	28	15	2/3

Students are also required to take four elective courses. These are selected to match the individual interests of the students. The list of electives shown in Table 4 is used primarily as a guideline and these courses can be substituted with material from other disciplines provided the content of the courses is in accordance with the plan of study. For example, two recent graduate students have taken classes from earth science and business to generate their own programs in computational meteorology and economics.

As a final task, students must perform a research project in conjunction with a faculty member and present their findings in a publicly announced seminar.

3. Results

3.1. Student enrollment and graduation data

The history of student enrollment and graduation rates from the inception of the computational science program are shown in Table 5. The enrollment has steadily increased to double the amount in 1998. Following an extremely successful recruiting effort in AY 2001–2002, the department is expecting the undergraduate enrollment to increase by nearly one-third in AY 2002–2003.

The small size of the department lends itself to close interaction with students. The faculty know all the students and are aware of their own personal interests in computation. This allows for the use of model problems and applications that are of particular interest to the students. In many courses, the last few class meetings are reserved for topics that the students select.

Due to the evolutionary nature of the discipline, the computational science program at SUNY Brockport is still in development. We have found that it is important to stress firm understanding of fundamental concepts in introductory courses and that these courses should not cover too much material.

3.2. Future curriculum plans

A combined BS/MS degree that will allow students to obtain two degrees in 5 years is currently under development. The combined program is a natural extension for students who wish to obtain an advanced degree because of the swing-level structure of our upper-division courses. Students in the BS/MS program take swing-level courses at the graduate level in addition to two additional elective courses. The remainder of the requirements for the BS and MS degrees are unchanged.

Another change will be the introduction of a course in applied linear algebra which will replace the current linear algebra course. This is being pursued because the current implementation of linear algebra is highly traditional, relying primarily on formal proofs. As such, students acquire very little experience in the practical application of this subject. The planned course would be split evenly between theoretical and applied linear algebra. An additional elective course on the direct and iterative solution of linear and non-linear systems is also being developed.

4. Conclusion

The infancy of the department makes it difficult to quantitatively assess the quality of education that students receive. The evaluation of Student Learning Outcomes (SLO) is still in progress; however, it is felt that the program to date has been very successful. At the current time, all graduates have found employment either in academics or industry (Xerox, Lockheed-Martin, PayChex), or have been accepted to graduate programs in computational science or other disciplines (Worcester Polytechnic Institute, Michigan Technological Institute).

It is illuminating to reveal the comments that three students have made regarding their educational experience. The first of these is an undergraduate student who studied as an intern at the General Electric Company in Schenectady, NY, in the Summer of 2001. Of his experiences, he writes

> The tasks I performed while being an intern at GE did not directly relate to my course studies, but the indirect relation was evident ... Companies may not want me to do computational fluid dynamics, computational physics or C++ programming but they know that I can quickly understand and grasp technology problems and that is what they are looking for—Mike Riedy Jr., BS, 2002.

The second student is an international student who was offered a job by PayChex (a corporation local to the Rochester area) well before her graduation. She comments

> I am so happy with my job ... and I owe it all to my favorite teachers in the Computational Science Department. I plan to return in the future to get my masters' degree in computational finance—Nathalie Semon, BS, 2001.

The final student was one of the first students admitted to the graduate program, having obtained his undergraduate degree in mechanical engineering from the Pennsylvania State University. He writes

> Lockheed-Martin had advertised for three different positions: a computer scientist, a mathematician and an engineer ... they told me they found all three in one person—Edward Ovando, MS, 2000.

Comments such as these indicate that students who complete either degree program are satisfied with the education and experience they receive and that they are able to apply what they have learned in their post-collegiate careers.

Acknowledgements

The author wishes to acknowledge the helpful assistance of the other department faculty members, Osman Yasar (Chair) and Robert Tuzun (Graduate Director), for their helpful information on the history and development of the department as well as providing feedback on early revisions of this work. He would also like the acknowledge suggestions provided by the referees and the editor for making the presentation more clear.

References

[1] M.L. Ennis, Update on the status of computational science and engineering in US graduate programs, Technical Report AHPCC99-023, Albuquerque High Performance Computing Center, The University of New Mexico, 1999.
[2] The National HPCC Software Exchange, 2002. http://www.nhse.org/cse_edu.html.
[3] SIAM Links, Graduate programs in computational science, 2002. http://www.siam.org/world/compsci/cplsci.htm.
[4] SIAM Working Group on CSE education, Graduate education in computational science and engineering, SIAM Rev. 43 (2001) 163–177.
[5] C.D. Swanson, Computational science education, The Krell Institute, 2002. http://www.krellinst.org/learningcenter/CSE_survey/index.html.
[6] O. Yasar, Computational science education: standards, learning outcomes and assessment, in: V.N. Alexandrov, et al. (Eds.), Proceedings of the International Conference on Computational Science (ICCS 2001), Springer, Berlin, 2001, pp. 1159–1169.
[7] O. Yasar, R.H. Landau, Elements in computational science education, SIAM Rev., 2002, submitted for publication.
[8] O. Yasar, K. Rajasethupathy, R.E. Tuzun, R.A. McCoy, J.A. Harkin, A new perspective on computational science education, CiSE 2 (2000) 74–79.

Leigh J. Little received his BS (1989) and MS (1991) degrees in Mechanical Engineering and MNS (1994) and PhD (1998) degrees in Numerical Analysis from Arizona State University. He was a Postdoctoral Associate in the Department of Computer Science and Engineering at the University of Minnesota from 1998 until 2000. He has since been serving as an Assistant Professor in the Department of Computational Science at SUNY Brockport. His research interests are scientific computing, numerical linear algebra, preconditioning of iterative methods for linear systems and sparse matrix algebra.