COMPUTATIONAL SCIENCE EDUCATION

Charles D. Swanson
Krell Institute

November 2003

COMPUTATIONAL SCIENCE EDUCATION

DISCLAIMER

Sources for this document include web pages and direct mailings from the universities. Before using any information about any computational science education program mentioned in this document, please contact the relevant institution.

COMMENTS

Comments or questions relating to this document are welcome and can be addressed to:
Charles D. Swanson
Department of Computer Science and Engineering
University of Minnesota
4-192 EE/CS Building
Minneapolis, MN 55455
cswanson@cs.umn.edu

COMPUTATIONAL SCIENCE EDUCATION

Charles D. Swanson
Krell Institute

November 2003

This paper surveys undergraduate computational science and engineering (CSE) programs at universities in the United States. These interdisciplinary programs were developed to educate science and engineering students in the methods of scientific computing, including computer modeling and simulation.

In addition, the paper includes information on graduate CSE programs, national supercomputer center programs, and resources for implementing new computational science education programs.

CONTENTS

I. Introduction

II. Current Computational Science Education Programs
   A. Undergraduate Programs
   B. Graduate Programs

III. Resources for Computational Science Education

IV. Bibliography

Appendix 1 - Undergraduate Programs in Computational Science

Appendix 2 - Graduate Programs in Computational Science

I. INTRODUCTION

Computational science and engineering (CSE) is an interdisciplinary field that applies the techniques of computer science and mathematics to solving physical, biological, and engineering problems. CSE emerged as a discipline with the advent of supercomputers in the 1970s. Today, the methods of CSE can also be used with cost-effective PCs and workstations making them available to a large number of scientists and engineers studying complex events such as car crashes, protein folding, and the formation of galaxies.

The increased use of computational science in government laboratories and industry has generated a corresponding demand to educate science and engineering students in the techniques of CSE. Colleges and universities are meeting this demand by developing courses and programs. Two recent initiatives (by the University of New Mexico, Albuquerque High Performance Computing Center[1] and the Working Group on CSE Education at SIAM[2]) have surveyed graduate level programs. This report focuses on undergraduate programs at thirty four US institutions, including liberal arts colleges, small universities, and large research universities.

After all these current courses and programs have been considered, however, two questions still remain without complete answers: (1) what is the intellectual core of computational science and engineering? (2) what constitutes an interdisciplinary undergraduate CSE program? The Krell Institute is helping to organize resources and opportunities for the computational science and engineering community to more fully answer these questions. In addition to this survey paper, web pages devoted to undergraduate computational science and education will be available on the Krell Institute site (www.krellinst.org), various groups will be convened to evaluate relevant issues, and workshops and/or conferences will be organized to involve the broader science and engineering community.

---

1. Martha Lee Ennis, *Update on the Status of Computational Science and Engineering in U.S. Graduate Programs*, Albuquerque High Performance Computing Center, The University of New Mexico, AHPCC99-023, September 28, 1999.

2. Working group on CSE Education, *Graduate Education in Computational Science and Engineering*, SIAM Review, Vol. 43, No. 1, pp. 163-177, March 2001.

II. CURRENT COMPUTATIONAL SCIENCE EDUCATION PROGRAMS

1. Undergraduate Programs in the United States

The undergraduate computational science programs included in this survey are listed in Table 1.

Table 1. Undergraduate Computational Science Programs in the United States

Degrees/Formal Programs (16)

Capital University	Minor in Computational Science
Clark University	Concentration in Computational Science
Illinois State University	B.S. in Computational Physics
Oregon State University	B.S. in Computational Physics
Princeton University	Undergraduate Certificate in Applied and Computational Mathematics
Rice University	B.A. in Computational and Applied Mathematics
Salve Regina University	Minor in Computational Science
SUNY Brockport	B.S. in Computational Science
Stanford University	B.S. in Mathematical and Computational Science
Syracuse University	Minor in Computational Science
University at Buffalo (SUNY)	B.S. in Computational Physics
University of Chicago	B.A. and B.S in Computational and Applied Mathematics
University of Wisconsin - Eau Claire	Minor in Computational Science
University of Wisconsin - LaCrosse	Minor in Computational Science
Wittenberg University	Minor in Computational Science
Wofford College	Emphasis in Computational Science

Courses only (18)

Boston University
California Institute of Technology
Clarkson University
Clemson University
University of Colorado
Duke University
Elizabeth City State University
University of Houston-Downtown
Indiana University of Pennsylvania
Indiana University - Purdue University at Indianapolis

Macalester College
University of Minnesota
New Mexico Tech
North Carolina State University
University of Rochester
San Diego State University
SUNY Institute of Technology at Utica
United States Naval Academy

These programs include sixteen that offer a formal degree, minor, or certificate and eighteen that offer undergraduate courses in computational science without any formal program. Although several universities offer degrees in computational physics or computational mathematics, the B.S. in Computational Science now offered by SUNY Brockport is the only major degree program in the United States specifically for computational science.

Another dimension of undergraduate education in computational science is the range of institutions. Early CSE programs were primarily at large research universities. Now courses and programs are being offered at smaller universities (e.g., the University of Wisconsin at Eau Claire) and liberal arts colleges (e.g., Wofford College, Macalester College).

The colleges and universities included in this survey have all supplemented existing traditional courses in mathematics, computer science, and the scientific/engineering disciplines with new interdisciplinary courses to create a CSE curriculum. There is some commonality among the courses, so an answer to the question of what constitutes an interdisciplinary undergraduate CSE program is beginning to appear.

Existing courses that typically appear in the CSE curriculum include:

Programming (Fortran, C/C++, Java)
Data Structures
Graphics
Linear Algebra
Numerical Analysis
Ordinary and Partial Differential Equations

New, interdisciplinary courses that often appear are:

Introduction to Computational Science
Scientific Programming
Scientific Visualization
High performance Computing
Parallel Programming/Architecture
Computational Physics/Chemistry/Biology/etc.

2. International Undergraduate Programs

Undergraduate computational science programs have also been developed outside of the United States. Table 2 lists several representative programs at international universities.

Table 2. International Undergraduate Computational Science Programs

Australian National University room.anu.edu.au/bcomptlsci/index.html	Australia	Bachelor of Computational Science
Carleton University www.carleton.ca/chemistry	Canada	Honours and Major in Computational Chemistry
National University of Singapore www.cz3.nus.edu.sg/student-course.html	Singapore	B.Sc. in Computational Science
Trinity College www.tcd.ie/Physics/Courses/page39.html	Ireland	Moderatorship in Computational Physics
Trinity College www.tcd.ie/Chemistry/Computational/	Ireland	Moderatorship in Computational Chemistry

B. Graduate Programs

In the 1998 survey[1] of computational science education programs that preceded this survey, 31 graduate programs were described. In 1999 that survey was updated by Martha Lee Ennis at the Albuquerque High Performance Computing Center at the University of New Mexico[2]. The Ennis report described 47 programs in the United States including, or many schools, information about the number of years the program has existed, the number of graduates, and plans for the future. The schools included in the Ennis report are listed in Appendix 2.

In 2001, the SIAM Working Group on CSE Education published a comprehensive report on Graduate Education in Computational Science and Engineering[3]. This report discussed the definition of CSE and its role in science and industry. It then summarized graduate level objectives and curricula. Several example programs in the U.S. (Stanford University, University of Texas at Austin, University of Illinois, and Purdue University) and Europe (ETH Zurich, Switzerland, and Royal Institute of Technology (KTH), Sweden) were covered in detail. The report concluded with a consideration of how SIAM may contribute to graduate CSE education in the future.

---

1. Charles D. Swanson, *Computational Science Education*, SGI, March 1998.

2. Martha Lee Ennis, *Update on the Status of Computational Science and Engineering in U.S. Graduate Programs*, Albuquerque High Performance Computing Center, The University of New Mexico, AHPCC99-023, September 28, 1999.

3. Working group on CSE Education, *Graduate Education in Computational Science and Engineering*, SIAM Review, Vol. 43, No. 1, pp. 163-177, March 2001.

III. RESOURCES FOR COMPUTATIONAL SCIENCE EDUCATION

Many resources are available to help educators introduce computational science courses and programs at all levels. These include faculty workshops, conferences, textbooks, magazines, and materials available on the web.

A. National Computational Science Institute (http://www.computationalscience.org)

The National Science Foundation awarded a three-year grant to the Shodor Education Foundation (http://www.shodor.org) for the creation of the *National Computational Science Institute* (NCSI). NCSI expands the already popular regional workshops known as the Shodor Computational Science Institute (SCSI). At over 18 partner sites across the country, NCSI will introduce the hands-on use of computational science, numerical models, and data visualization tools across the curriculum.

B. Supercomputer Center Programs

Many national and state supercomputer centers and government laboratories have programs in computational science education. These programs include workshops, student research opportunities, and seminars.

The National Computational Science Alliance (Alliance) and the National Partnership for Advanced Computational Infrastructure (NPACI) are the two NSF-funded partnerships that help to create a national computing infrastructure to support computational science and engineering. They jointly support an education, outreach, and training effort called EOT-PACI. (http://www.eot.org)

Information on other supercomputer center programs in education can be found on the web:

- National Center for Atmospheric Research (NCAR) (http://www.scd.ucar.edu/)
- National Center for Supercomputing Applications (NCSA) (http://www.ncsa.uiuc.edu/)
- North Carolina Supercomputing Center (http://www.ncsc.org/)
- Ohio Supercomputer Center (OSC) (http://www.osc.edu/education/)
- Pittsburgh Supercomputing Center (PSC) (http://www.psc.edu/education.html)
- San Diego Supercomputer Center (SDSC) (http://www.sdsc.edu/Education/)
- SDSC Education Center on Computational Science & Engineering (http://www.edcenter.sdsu.edu/index.html)

C. Computational Science Education Project

The *Computational Science Education Project* (CSEP) produced teaching material for high performance computing. The material is targeted for advanced undergraduates and beginning graduate students in science and engineering. Although this project is no longer active, the material is still available on the CSEP web site. (http://csep1.phy.ornl.gov/csep.html)

D. Undergraduate Computational Engineering and Sciences Initiative

The Department of Energy sponsored a project titled *Undergraduate Computational Engineering and Sciences: An Educational Initiative in Computational Science* (UCES). The emphasis of the project was on collecting, developing, and distributing a set of educational materials in computational science at the undergraduate level. The developed material was modular, interactive, and logically grouped into courses, beginning with introductory material targeted towards freshmen and sophomores (http://www.krellinst.org/UCES/)

E. Textbooks and Reference Books

The Bibliography lists many articles and textbooks devoted to computational science.

F. Magazine

In addition to the several journals dedicated to computational science and engineering, the following magazine is particularly relevant to computational science education:

Computing in Science and Engineering - a joint publication of the IEEE Computer Society and the American Institute of Physics. (http://www.computer.org/cise/)

G. Conferences

For several years the Supercomputing 'xy (now called SCxy) conferences on high performance computing and communications have had significant sessions on computational science education. SC2003, to be held in Phoenix on November 15-21, 2003, will have an education programming focusing on undergraduate computational science education. (http://www.sc-conference.org/sc2003/edu_general.html)

The second SIAM *Conference on Computational Science and Engineering* was held in San Diego, February 10-13, 2003 (http://www.siam.org/meetings/cse03/index.htm). The next conference will be held in 2005.

IV. BIBLIOGRAPHY

Articles and Reports

Geoffrey C. Fox, Parallel Computing and Education, Daedalus, Proceedings of the American Academy of Arts and Sciences, 121, 1, 111-118 (Winter 1992).

John R. Rice, Academic Programs in Computational Science and Engineering, Computational Science and Engineering, 1, 1, 13-21 (Spring 1994).

High Performance Computing and Communication: Toward a National Information Infrastructure, A Report by the Committee on Physical, Mathematical, and Engineering Sciences, Federal Coordinating Council for Science, Engineering, and Technology, Office of Science and Technology Policy, (1994).

D. E. Stevenson, Science, Computational Science and Computer Science: At a Crossroads, Communications of the ACM, 137,12, 85-96 (December 1994).

NSF Blue Ribbon Panel on High Performance Computing, Lewis Branscomb (Chairman), From Desktop to Teraflop: Exploiting the US Lead in High Performance Computing, NSF Report NSB 93-205, (August 1993).

Randall Bramley and Efstratios Gallopoulos, CSE Conference Gets High Marks, Computational Science and Engineering, 1, 2, 76-79 (Summer 1994).

Martha Lee Ennis, Update on the Status of Computational Science and Engineering in U.S. Graduate Programs, Albuquerque High Performance Computing Center, The University of New Mexico, AHPCC99-023 (September 28, 1999).

Working group on CSE Education, Graduate Education in Computational Science and Engineering, SIAM Review, Vol. 43, No. 1, pp. 163-177 (March 2001).

Textbooks and Reference Books

Steven E. Koonin and Dawn C. Meredith, Computational Physics, (Addison-Wesley, 1990).

David M. Hirst, A Computational Approach to Chemistry, (Blackwell Sci., 1990).

J. L. Henessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, (Morgan Kaufmann, 1990).

W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, Numerical Recipes in Fortran: The Art of Scientific Computing, Second Edition, (Cambridge University Press, 1992).

Gene Golub and James M. Ortega, *Scientific Computing: An Introduction with Parallel Computing*, (Academic Press, 1993).

Robert S. Wolff and Larry S. Yaeger, *Visualization of Natural Phenomena*, (TELOS/Springer-Verlag, 1993).

V. Kumar, A. Grama, A. Gupta and G. Karypis, *Introduction to Parallel Computing: Design and Analysis of Algorithms*, (Benjamin/Cummings, 1994).

Kai Hwang, *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, (McGraw-Hill, 1993).

Richard E. Crandall, *Projects in Scientific Computation*, (TELOS/Springer-Verlag, 1994).

Lloyd D. Fosdick, Elizabeth R. Jessup, Carolyn J. C. Schauble, and Gitta Domik, *Introduction to High-Performance Scientific Computing*, (MIT Press, 1996).

Joseph L. Zachary, *Introduction to Scientific Programming: Computational Problem Solving Using Maple and C*, (Springer-Verlag, 1996).

Tao Pang, *An Introduction to Computational Physics*, (Cambridge University Press, 1997).

D.M. Etter, *Engineering Problem Solving with MATLAB, 2/e*, (Prentice-Hall, 1997).

Rubin H. Landau and M. J. Paez, *Computational Physics, Problem Solving With Computers*, (John Wiley, 1997).

Dan Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, (Cambridge University Press, 1997).

Joseph L. Zachary, *Introduction to Scientific Programming: Computational Problem Solving Using Mathematica and C,* (Springer-Verlag, 1998).

Kevin Dowd and Charles Severance, *High Performance Computing, 2nd Ed*., (O'Reilly & Associates, 1998).

Charles F. Van Loan, *Introduction to Scientific Computing: A Matrix-Vector Approach Using MATLAB, 2/e*, (Prentice-Hall, 2000).

Michael T. Heath, *Scientific Computing, Second Ed*., (McGraw-Hill, 2001).

John R. Berryhill, *C++ Scientific Programming - Computational Recipes at a Higher Level*, (John Wiley, 2001).

William H. Press, Saul A. Teukolsky, Brian P. Flannery, and William T. Vetterling, *Numerical Recipes in C++: The Art of Scientific Computing*, (Cambridge University Press, 2002).

APPENDIX 1
UNDERGRADUATE PROGRAMS IN COMPUTATIONAL SCIENCE

Boston University
California Institute of Technology
Capital University
Clark University
Clarkson University
Clemson University
Duke University
Elizabeth City State University
Illinois State University
Indiana University of Pennsylvania
Indiana University - Purdue University at Indianapolis
Macalester College
New Mexico Tech
North Carolina State University
Oregon State University
Princeton University
Rice University
Salve Regina University
San Diego State University
Stanford University
SUNY Brockport
SUNY Institute of Technology at Utica
Syracuse University
United States Naval Academy
University at Buffalo (SUNY)
University of Chicago
University of Colorado
University of Houston-Downtown
University of Minnesota
University of Rochester
University of Wisconsin - Eau Claire
University of Wisconsin - LaCrosse
Wittenberg University
Wofford College

Boston University

Degree: Interdisciplinary undergraduate courses

Departments: The Center for Computational Science in coordination with the Departments of Chemistry, Computer Science, Physics, and Electrical, Computer & Systems Engineering

Description: The Boston University Center for Computational Science brings together researchers, educators, and students who are pioneering advanced application of computing to problems in the sciences, engineering, arts and humanities. The Center works with the Scientific Computing and Visualization Group of the Office for Information Technology to support high performance computing and visualization systems at Boston University. Several courses in computational science and engineering are available to undergraduate students.

Courses:

CAS CS 512 - Parallel Algorithms and Programs
Design, analysis, and programming of parallel and distributed algorithms. Fundamental limits of parallelism. Measures of complexity of parallel programs. Combinatorial and numerical parallel algorithms for different inter-processor communication topologies. Analytical and experimental performance evaluation.

ENG EK 420 - Introduction to Parallel Computing
Introduces fundamental methods for scientific computing in the context of massively parallel computation. Discussions are organized around important algorithmic concepts and specific applications chosen to illustrate the methods. Different parallel computation models are evaluated within the framework of specific algorithms. Students are required to observe, modify, and/or design programs suitable for running on highly parallel architectures such as the Connection Machine and on current multiprocessor systems. In addition, students are required to develop competence with a variety of tools useful in the parallel computing environment including graphical methods to analyze large data sets, the high-level parallel language C++, and X-windows.

ENG EK 521- Parallel Computation for Engineering
Methods of parallel computing for science and engineering applications. Presented through lectures and programming exercises drawn from continuum mechanics, diffusive transport, magnetic materials, and molecular modeling. Given the appropriate equations of motion, each student is guided to develop parallel algorithms, design the simulations software, and analyze the resulting data using proper statistical and graphical analysis methods. In addition to the weekly laboratories, each student will complete a term project.

Boston University (cont.)

CAS PY 421 - Advanced Scientific Computing in Physics
Introduces advanced computational techniques for research problems in physics, with emphasis on computationally intensive applications in a massively parallel supercomputing environment. Corequisite laboratory focuses on algorithms and computational tools with cross-disciplinary application.

CAS PY 502 - Computational Physics
Fundamental methods of computational physics and applications; numerical algorithms; linear algebra, differential equations; computer simulation; vectorization, parallelism, and optimization. Examples and projects on scientific applications.

Contact:

Ilona Lappo
Boston University
Center for Computational Science
3 Cummington Street
Boston, MA 02215
617-353-6078
lappo@bu.edu

http://ccs.bu.edu

California Institute of Technology

Degree: Courses in computational science

Departments: Computer Science, Applied and Computational Mathematics

Description: See course descriptions.

Courses:

ACM 113 - Introduction to Optimization
Unconstrained optimization: first and second order conditions, Newton-like methods, conjugate direction methods, trust region methods. Constrained optimization: linear programming, general theory for nonlinear constrained optimization, quadratic programming and general linearly constrained optimization, nonlinear programming, optimizing functionals using the calculus of variations. Combinatorial optimization: integer programming, dynamic programming.

ACM/CS 114 ab - Parallel Algorithms for Scientific Applications
Introduction to parallel program design for numerically intensive scientific applications. First term: parallel programming methods; distributed-memory model with message passing using the message passing interface; shared-memory model with threads using open MP; object-based models using a problem-solving environment with parallel objects. Parallel numerical algorithms: numerical methods for linear algebraic systems, such as LU decomposition, QR method, Lanczos and Arnoldi methods, pseudospectra, CG solvers. Second term: parallel implementations of numerical methods for PDEs, including finite-difference, finite-element, and shock- capturing schemes; particle-based simulations of complex systems. Implementation of adaptive mesh refinement. Grid-based computing, load balancing strategies.

ACM 214 - Methods of Tomography and Computational Electromagnetism
Discrete Fourier transform (DFT) and its properties. Applications of DFT to integration, interpolation, and digital filtering. Radon transform, properties, inversion formulas, relation to the Fourier transform. Tomography algorithms. Fundamentals of electromagnetism. Problems of electrostatics and Laplace equations. The variational principle. Integral equation techniques for the interior and exterior boundary problems. Acoustic waves and Helmholtz equations. Addition theorem, boundary integral equations, Maxwell's equations. Exact and approximate methods in wave propagation: Born-Rytov approximations, geometrical optics, fast methods. Inverse problems. The course will contain a variety of computational exercises and projects.

Ae/ACM 232 abc - Computational Fluid Dynamics
Introduction to the use of numerical methods in the solution of fluid mechanics problems. First term: Review of basic numerical techniques: interpolation, integration, application for systems of ordinary differential equations, stability and accuracy. Treatment of partial differential equations in one space variable. Nonlinear convective-diffusive and

California Institute of Technology (cont.)

convective-dispersive phenomena. Treatment of discontinuous solutions. Second term: Survey of finite difference, finite element, and spectral approximations for the solution of the incompressible Navier-Stokes equations in two and three dimensions. Numerical study of problems of hydrodynamic stability, transition, and turbulence. Third term: Methods for the numerical solution of the compressible Euler and Navier-Stokes equations in one, two, and three dimensions. Finite-difference and finite-volume methods. Methods based on solution of the Riemann problem. Flux-splitting. Shock-capturing methods and related stability problems. Implicit artificial viscosity for the Euler equations. Total variation diminishing approximations.

Contact:

Dan Meiron
Applied and Computational Mathematics and Computer Science
MC 08-31
California Institute of Technology
Pasadena, CA 91125
626-396-8157
dim@its.caltech.edu

http://www.acm.caltech.edu

Capital University

Degree: Minor in Computational Science

Department: Computational Science Across the Curriculum (CSAC)

Description: Computational Science is a field at the intersection of mathematics, computer science, and science (hereafter, broadly defined to include biology, chemistry, engineering, environmental science, finance, geology, medical science, neuroscience, physics, and psychology). Computational Science offers an interdisciplinary approach to scientific research and provides an important tool, alongside theory and experimentation, in the development of scientific knowledge. This emerging and rapidly growing interdisciplinary field integrates computing, mathematical modeling, and visualization to solve problems in the physical, natural and behavioral sciences, finance and engineering.

Students who participate in the Computational Science curriculum will:

1. Experience an interdisciplinary, team-based approach to science problem solving;
2. Explore the creative nature of Computational Science;
3. Improve written and oral communication related to scientific and technical projects;
4. Use current and emerging computing technologies; and
5. Prepare to pursue graduate degrees in science and mathematics.

Computational Science complements major courses of study in biology, chemistry, computer science, education, environmental science, mathematics, preengineering, pre-professional health and pre-medicine, and psychology.

Requirements for a minor - 22 hours: CS 160 (Introduction to Computer Science), CSAC 245, 335, 435, 445 and two elective CSAC courses.

Courses:

CSAC 200 - Introduction to Computational Science Seminar
Students explore the possibilities and realities of careers in computational science and enhance their knowledge base in mathematics, science, and technology. This course is designed for non-science and non-mathematics majors.

CSAC 245 - Computational Science I
An introduction to the problems and solution methodologies in computational science. Computational tools such as a computer algebra system, a high performance computing engine, visualization software and Internet resources will used to explore and solve mathematical problems drawn from various fields of science.

Capital University (cont.)

CSAC 335 - Differential Equations and Dynamic Systems
A modeling approach is used to cover techniques for solving a variety of Ordinary Differential Equations (ODE): linear (first and higher order), non-linear, and systems of ODEs. The course also examines methodologies for solving linear and non-linear continuous and discrete dynamic systems. Furthermore, an introduction to the solution of Partial Differential Equations (PDE) is presented.

CSAC 376 - Introduction to Parallel Computing
A first course in the design, analysis and development of parallel and distributed algorithms on different architectures.

CSAC 391 - Computational Chemistry
An introduction to the theory and practice of computational chemistry. Examples and applications are drawn from different chemical specializations.

CSAC 392 - Computational Biology
This course is designed to stimulate critical thinking in computational biology and bioinformatics. Topics include DNA sequence and alignment, philogeny, and modeling biological phenomena using conceptual modeling tools.

CSAC 393 - Computational Environmental Science
This course is designed to stimulate critical thinking about environmental science principles using computational modeling methodologies. Some topics to be covered include groundwater and contaminant transport, phosphorus cycling in surface waters, and global climate change.

CSAC 394 - Computational Neuroscience and Psychology
The course is divided into modules in which different, but related problems in neuroscience and psychology are explored. A key aspect of this course is to have students use computer visualization techniques to assess the various mathematical models that they create.

CSAC 397 - Scientific Visualization
This course provides a general introduction to the terminology, methodology, and applications of scientific visualization.

CSAC 435 - Computational and Numerical Methods
This course is divided into modules providing an in-depth exploration into the solution methodology of Computational Science problems. Such methodology primarily employs numerical techniques while presenting their theoretical development. A cadre of software tools (Computer Algebra System, Numeric libraries, Visualization software and Parallel processing tools) are used in the solution process.

Capital University (cont.)

CSAC 445 - Research Experience in Computational Science
A capstone research experience usually comprising a comprehensive literature review, design, and implementation of computational science techniques to solve a problem in the behavioral, computer, financial, mathematical, natural, physical, or social sciences.

CSAC 476 - Introduction to High-Performance Scientific Computing
Introduction to the use of high-performance computing systems in science and engineering. The physical aspects of a variety of problems are surveyed and techniques for solving the problems on a variety of high-performance computers are analyzed.

Contact:

Ignatios Vakalis
Associate Professor
Department of Mathematics and Computer Science
Capital University
East Main Street
Columbus OH 43209
614-231-7848
ivakalis@capital.edu

http://www.capital.edu/acad/as/csac/

Clark University

Degree: undergraduate concentration in Computational Science

Departments: Biology, Chemistry, Computer Science, Mathematics, and Physics

Description: The enormous progress in computational technology has generated a new methodology for learning and advancing the traditional sciences such as physics, chemistry, and biology. Computational science combines the application of numerical methods, models, and algorithms in the context of solving problems that are intractable by traditional methods. It is distinct from computer science, which is the study of computers and computation, and it is different from theory and experiment, the traditional forms of science, in that it seeks to gain understanding principally through the analysis of mathematical models.

The goal of the interdisciplinary computational science concentration is to provide an opportunity for students to learn about the interplay between science and computation. The concentration is especially suitable for undergraduate students majoring in the sciences, mathematics, or computer science. Students completing this concentration may enter graduate programs in either their majors or in newly created interdisciplinary graduate programs in computational science, and be well prepared to go into industry.

Courses: Students choose courses from the participating departments, including the following computational course:

Physics 127 - Computer Simulation Laboratory
Introduces methods of computer simulation and its diverse applications. Course is project-oriented, with students proceeding at their own pace depending on their background and interests. Projects include planetary motion, chaotic systems, fractal phenomena, random systems, and thermal systems. Methods include the numerical solution of differential equations and Monte Carlo techniques. Course emphasizes structured programming, and is recommended for prospective science majors as an introduction to programming rather than Computer Science 101.

Contact:

Harvey Gould, Professor of Physics
350 Biophysics
Clark University
Worcester, MA 01610-1477
508-793-7485
hgould@clarku.edu

http://science.clarku.edu/compu_sci.html

Clarkson University

Degree: Course in computational science.

Department: Mathematics and Computer Science

Description: See course description.

Course:

MA570 - Fundamentals of Scientific Computation
This course introduces the basic concepts, methods, and software of scientific computation. Topics include programming in a high-level language (e.g., Matlab), floating-point arithmetic, sources and propagation of errors, accuracy and efficiency of algorithms in the context of nonlinear equations, linear algebraic systems and eigenvalues, interpolation and numerical integration.

Contact:

Peter Turner
Chair, Mathematics & Computer Science Department
Clarkson University
PO Box 5815
Potsdam, NY 13699-5815
315-268-2334
pturner@clarkson.edu

http://www.clarkson.edu/mcs/

Clemson University

Degree: Courses in computational science

Department: Computer Science

Description: See course description.

Courses:

CP SC 455, 655 - Computational Science
Introduction to the methods and problems of computational science. Course uses problems from engineering and science to develop mathematical and computational solutions. Case studies use techniques from Grand Challenge problems. Emphasizes the use of networking, group development, and modern programming environments.

Contact:

D. E. Stevenson
442 Edwards Hall
Department of Computer Science
Clemson University
Clemson, SC 29634-0974
864-656-5880
steve@cs.clemson.edu

http://www.cs.clemson.edu/

Duke University

Degree: Undergraduate/graduate course

Department: Biomedical Engineering

Description: See course description.

Course:

BME 246 - Computational Methods in Biomedical Engineering
An Introduction to practical computational methods for data analysis and simulation with a major emphasis on implementation. Methods include numerical integration and differentiation, extrapolation, interpolation, splining FFTs, convolution, ODEs, and simple one and two-dimensional PDEs using finite differencing. Examples from biomechanics, electrophysiology, and imaging. Project work included and students must have good working knowledge of Unix, Fortran, or C

Contact:

Craig Henriquez
Department of Biomedical Engineering
P.O. Box 90281
Duke University
Durham, NC 27708-0281
919-660-5168
ch@epi.mc.duke.edu

http://bme-www.egr.duke.edu/fandr_indivprofiles.php?id=5

Elizabeth City State University

Degree: training programs and workshops in computational science

Department: Computational Science - Scientific Visualization (CSSV) Center in The School of Mathematics, Science and Technology

Description: The CSSV Center is a Center specializing in "an interdisciplinary approach to research, problem solving and visually displaying of data in the mathematical sciences, natural sciences and technological applications." The Center provides user friendly support services for students and faculty who are pursing research or educational endeavors which make significant uses of computational mathematics-numerical methods, mathematical modeling, high performance computer programming, using specialized computer application packages, and/or computer visualization tools and techniques.

Contact:

Johnny L. Houston
Department of Mathematics and Computer Science
Elizabeth City State University
Elizabeth City, NC 27909
252-335-3361
jlhouston@mail.ecsu.edu

http://cssvc.ecsu.edu/

Illinois State University

Degree: B.S. in Computational Physics

Department: Physics

Description: Computational physicists use high-performance computers to explore physical phenomena, from those involving the most fundamental objects such as quarks and black holes to those important in applications like weather prediction, medical technology, and environmental cleanup. This new approach to physics opens avenues to problems whose solution is otherwise impossible. The Computational Physics degree at Illinois State University is designed for students seeking either industrial employment or graduate study in computationally-oriented science, engineering, and related fields. Majors complete a well-balanced curriculum in theoretical, experimental, and computational physics paralleling the regular physics major through the intermediate level courses and then specializing in computational physics.

Courses:

PHY 288 - Methods of Computational Science
Introduction of a wide variety of computational techniques and their application to physics and chemistry problems. The applications will be restricted to computer simulations of simple physical systems from mechanics, electromagnetism and chemistry.

PHY 388 - Advanced Computational Physics
Application of computational methods to contemporary topics in physics, including nonlinear classical and quantum dynamics or physical problems that involve many degrees of freedom.

PHY 390 - Computational Research in Physics
A capstone experience during the Spring semester of their senior year. It is an independent research project course, where each student applies a particular computational technique to a problem in physics. A project may be a current research project of a faculty member or any other substantial computational project.

Contact:

Richard F. Martin, Professor and Acting Chair
Department of Physics
Illinois State University
Campus Box 4560
Normal, IL 61790-4560
309-438-8756
info@entropy.phy.ilstu.edu
http://www.phy.ilstu.edu/CompPhys/CP.html

Indiana University of Pennsylvania

Degree: Undergraduate course

Department: Mathematics

Description: See course description

Course:

MA 451/551/CO 451 - Numerical Methods for Supercomputers (http://www.ma.iup.edu/projects/Supercomputing/Summary.html) Supercomputers make use of special computer architectures - vector and parallel processors -in order to achieve the fastest processing speed currently available. Students will be introduced to these features and will learn how numerical algorithms can be constructed to exploit supercomputers capabilities. Students will gain practical experience in programming for CRAY supercomputers, in incorporating existing scientific software packages into user-written programs, in submitting remote jobs to the Pittsburgh Supercomputing Center, and in producing animated graphical output to summarize the typically large volume of output data generated by large scientific programs.

Contact:

H. Edward Donley
Mathematics Department
Indiana University of Pennsylvania
215 Stright Hall
Indiana, Pennsylvania 15705
724-357-3792
hedonley@grove.iup.edu

http://www.ma.iup.edu/people/hedonley.html

Indiana University - Purdue University at Indianapolis

Degree: Scientific computing track

Department: Computer and Information Science

Description: An interdisciplinary program with a series of courses linking mathematics, computing science, and applications.

Courses:

CSCI 475 - Scientific Computing I
Solving scientific problems on computers. Languages for scientific computing. Software development on workstations: using tools the environment provides; organization of programs. Computer architecture: impact on software and algorithms. Problem formulation modes selection/simplification; relationship to numerical methods. Solution of linear equations: methods and packages. Nonlinear equations and optimization problems.

CSCI 476 - Scientific Computing II
Elementary statistical computing: time series analysis, model fitting ,robust methods, generation of pseudo-random numbers and Monte-Carlo methods. Interpolation and curve fitting; numerical integration. Solving ordinary differential equations. Use of packaged environments and symbolic computation for scientific purposes.

CSCI 477 - High Performance Computing
Architecture of supercomputers: pipelined, vector, SIMD, MIMD; implications for algorithm and program design; vectorization, parallelization, loop restructuring, non-standard language features. Split ting computation between supercomputers and workstations; interactive analyses of remote machines' output. Numerical methods for large scale problems: examples from continuum mechanics, graphical visualization, statistical computing. A project is required.

Contact:

Ray Chin
Computer and Information Science Department
Indiana University - Purdue University at Indianapolis
Indianapolis, IN 46202
317-274-9738
rchin@indyvax.iupui.edu

http://www.cs.iupui.edu

Macalester College

Degree: Courses in computational science

Department: Mathematics and Computer Science

Description: See course descriptions.

Courses:

Computer Science 121 - Introduction to Scientific Programming
This course is intended to give students from diverse areas of science --- e.g., economics, biology, physics, chemistry, geology, geography, mathematics, engineering, statistics, etc, --- an effective ability to write software for solving problems in those disciplines and carrying out research. The course provides an introduction to programming and computation as well as to a number of important techniques and algorithms widely used throughout science: scientific graphics, equation solving, function fitting and optimization, storing and searching data, and simulation. There is an emphasis on ways to represent and transform information generally on the computer: in addition to numbers and text, images, sounds, databases, and so on.

Computer Science 365 - Scientific Computation Techniques and algorithms for computational solutions to scientific problems with applications to diverse disciplines. Topics include: numerical integration; root finding; interpolation, splines, and Bezier curves; statistical function estimation; modeling via simulation and Monte Carlo techniques; optimization; transforms; symbolic computing; controlling numerical error.

Contact:

Daniel T. Kaplan
Macalester College
Olin/Rice 224
1600 Grand Avenue
St. Paul, MN 55105
651-696-6599
kaplan@macalester.edu

http://www.macalester.edu/mathcs/index.html

New Mexico Institute of Mining and Technology

Degree: Undergraduate courses

Department: Computer Science and Mathematics

Description: Two courses dealing with high performance computing.

Courses:

CS451 - Introduction to Parallel Processing
Introduction to supercomputers and massively-parallel machine architecture, models of parallel computation, parallel algorithms, synchronization, parallel languages, data and functional parallelism, parallel performance analysis, popular interfaces, and parallel debugging. Students will gain experience in parallelization of sequential algorithms and implementation of parallel algorithms.

Mathematics 414 - Introduction to High Performance Computing
Solving scientific problems in high performance computing systems. Topics include: numerical methods, using software libraries and packages such as MATLAB, Mathematica, NAG, LAPACK, etc., matching algorithms to machines, measuring performance, and scientific visualization. A number of computing architectures will be used to solve a small set of prototype problems.

Contact:

Andrew Sung
Computer Science Department
New Mexico Institute of Mining and Technology
Socorro, NM 87801
505-835-5949
sung@cs.nmt.edu

http://www.cs.nmt.edu

North Carolina State University

Degree: Courses in computational science

Department: Mathematics

Description: Computational mathematics courses for undergraduates.

Courses:

MA 132 - Computational Mathematics for Life and Management
Computational aspects of calculus for the life and management sciences; use of spreadsheets and a computer algebra system; applications to data models, differential equation models, and optimization.

MA 402 - Computational Mathematics: Models, Methods and Analysis
Introduction to high performance computing and numerical modeling. Matrix models and boundary value problems with an emphasis on heat and mass transfer. Assessments of all approximations in the computational engineering and science process.

MA/CSC 583 - Introduction to Parallel Computing
Introduction to basic parallel architectures, algorithms, and programming paradigms; message passing collectives and communicators, parallel matrix products; domain decomposition with direct and iterative methods for linear systems; analysis of efficiency, complexity and errors; applications such as 2D heat and mass transfer.

Contact:

Robert E. White
Department of Mathematics
308 Harrelson Hall
Box 8205
North Carolina State University
Raleigh, NC
white@math.ncsu.edu

http://www4.ncsu.edu/eos/users/w/white/www/white/bob.html

Oregon State University

Degree: B.S. in Computational Physics

Department: Physics

Description: The OSU Physics Department has developed a two-quarter course in *Computational Physics*, a textbook that serves as an international model for an undergraduate computational physics course, web-based tutorials and demonstrations that enhance the course and the text, and a course in *Introductory Scientific Computing*. With the addition of an advanced computational laboratory and the use of courses offered in other departments, a coherent and rigorous B.S./B.A. degree program in computational physics has been developed.

Courses:

PH 265 - Introductory Scientific Computing
A course designed to provide freshman and sophomores with the basic computational tools and techniques needed for their study in science and engineering. Students learn by doing projects that solve problems in physical sciences and mathematics using symbolic and compiled languages with visualization. By use of the Maple problem-solving environment and the Java programming language, the students learn programming and numerical analysis in parallel with scientific problem solving.

Ph 465/565-466/566 - Computational Physics
Course aims:

- To teach through direct experience the use of scientific workstations and supercomputers in thinking creatively and solving problems in the physical sciences.
- To advance the development and organization of thinking about physical systems in a manner compatible with advanced computational analysis.
- To use the graphical capabilities of advanced workstations to visualize numerical solutions into highly interpretable forms.
- To instill attitudes of independence, personal communication, and organization, all of which are essential for mastery of complex systems.
- To understand physical systems at a level often encountered only in a research environment, and to use programming to deepen that understanding.
- To understand why hard work and even properly functioning and powerful software and hardware do not guarantee meaningful results. In an experimental science there are limits to accuracy and applicability.
- To instill in students an objected-oriented view of problem solving.

Oregon State University (cont.)

Contact:

Rubin H. Landau
CPUG Program Director
Department of Physics
Oregon State University
Corvallis, OR 97331
541-737-1693
rubin@physics.orst.edu

http://www.physics.orst.edu/~rubin/CPUG/

Princeton University

Degree: Undergraduate Certificate in Applied and Computational Mathematics (ACM)

Department: Program in Applied and Computational Mathematics

Description: In engineering and science today, there is a trend toward the use of modern sophisticated mathematical techniques of both computational and theoretical types. One reason for this is the revolutionary development of computational resources. Modern engineers and scientists have at their disposal user-friendly, powerful computers with sophisticated graphics to display results. Such modeling and simulation techniques are also increasingly used in the biological and social sciences, especially economics. With such tools, interdisciplinary problems which are complex and nonlinear have become the norm in contemporary science and engineering. The engineer/scientist must learn to use such tools wisely, accurately, and to their full power. A natural time to be introduced to such skills is as an undergraduate. During these formative years the modern engineer or scientist should learn the use of mathematical and computational techniques in an interactive interdisciplinary environment. At Princeton, The Program in Applied and Computational Mathematics offers a unique opportunity for a small group of students to learn to perform accurate and controlled numerical studies and, perhaps most important, to expose them to the philosophy and tools of interdisciplinary applied mathematics in a very personal and individualized manner. Refer to web page below for more information.

Contact:

Valerie Marino
Program Secretary
The Program in Applied and Computational Mathematics
Princeton University
Fine Hall, Washington Road
Princeton, NJ 08544-1000 USA
609-258-3703
vmarino@princeton.edu

http://www.math.princeton.edu/PACM

Rice University

Degree: B.A.

Department: Computational and Applied Mathematics (CAAM)

Description: Courses within this major can provide foundations applicable to the many fields of engineering, physical sciences, life sciences, behavioral and social sciences, and computer science. Undergraduate majors have considerable freedom to plan a course of study consistent with their particular interests. The professional degree (M.C.A.M.), for persons interested in practicing within this field, emphasizes general applied mathematics, operations research and optimization, and numerical analysis, while the M.A. and Ph.D. programs concentrate on research. Faculty research interests fall in the four general areas of numerical analysis and computation, physical mathematics, operations research and optimization, and mathematical modeling in physical, biological, or behavioral sciences.

Courses:

Students choose from a wide selection of courses in associated departments, including the following offered by Computational and Applied Mathematics:

CAAM 210 - Introduction to Engineering Computation
Introduction to engineering and scientific computation: Engineering workstations, programming concepts via Matlab, software systems, and numerical methods. Laboratory to illustrate the application of computational and visualization methods to problem analysis. Matlab serves as the primary computational and display tool. Optional supplemental instruction is presented in both C and f95.

CAAM 420 - Computational Science I
Scientific programming using high level languages, including C, Fortran, and C++. Emphasis on use of numerical libraries. Basic techniques of project planning, source management, documentation, program construction, i/o, visualization. Object-oriented design for numerical computation.

CAAM 520 - Computational Science II
Vector shared-memory, and message-passing parallel computer architectures. Numerical linear algebra for these architectures. Memory hierarchy issues, analysis and enhancement of performance, and use of programming tools and environments. Portable parallel scientific programming concepts using OpenMP and MPI. Introduction to component software architectures. Parallel numerical algorithms and scientific visualization.

Rice University (cont.)

Contact:

Dept. of Computational & Applied Mathematics
Rice University
6100 Main St. - MS 134
Houston, TX 77005-1892
713-348-4805
caam@rice.edu

http://www.caam.rice.edu/

Salve Regina University

Degree: Minor in Computational Science

Departments: Chemistry and Mathematical Sciences

Description: Computational science is a relatively young interdisciplinary field that involves the development and application of computational methodologies and techniques on high-performance computers to help solve problems in science and engineering. The emergence of computational science as a discipline has resulted primarily from advances in both computer architectures and computational algorithms which exploit these architectures.

The minor in Computational Science provides students with the opportunity to:

- Acquire an understanding of the development, analysis, and implementation of computational algorithms on high-performance computers.
- Develop skill in applying computational methodologies as tools for scientific investigation.
- Obtain knowledge of at least one application area chosen from the physical and life sciences.

Courses:

CMP201 - Scientific Programming
In this course students are introduced to the basic tools and methods of computational science. The primary tools are programming concepts using a high level programming language and basic end-user skills in the UNIX operating system. The methods stressed are techniques of problem analysis and the development of algorithms. Most examples and applications involve mathematics and science.

CMP301 - Scientific Computation
An introduction to numerical analysis and scientific computation. Topics include nonlinear equations, linear systems, interpolation and curve-fitting, numerical differentiation and integration, numerical solution of differential equations, approximation of functions, and models of computation on serial and parallel computers.

Contact:

Ernest E. Rothman
Mathematical Sciences Department
Salve Regina University
Newport, RI 02840-4192
401-847-3237
rothman@salve.edu

http://www.salve.edu/dept_compsci/index.html

San Diego State University

Degree: Undergraduate courses

Department: The Education Center on Computational Science and Engineering

Description: The mission of the Education Center on Computational Science and Engineering (ECCSE) is to foster the incorporation of high performance research tools for scientific investigation into the undergraduate curriculum. Our main goal is to better prepare learners for post-Baccalaureate activities where collaborative interdisciplinary teams, sophisticated computer tools, and effective communication are part of the research and problem solving environment.

Courses: Computational science courses are offered in several departments.

Chemistry

Chemistry 515 - Computational Chemistry
Overview of modern computational chemistry. Use of computational chemistry tools and their application to problems of chemical interest.

Computer Science

CS 205 - Introduction to Computational Programming and Visualization
Problem solving skills for needs of science. Use of computing and software tools of computational science introduced to gain competence in computer communications, programming and visualization. Supervised computer laboratory.

CS 575 - Supercomputing for the Sciences
An interdisciplinary course to introduce students in the sciences, engineering and other disciplines to the world of high performance computing using campus computers and the supercomputers at the San Diego Supercomputer Center (SDSC). Additional architectures may also be explored. The computing projects have a goal to effectively use, by the end of the course, MPI on Rohan as well as MPI on the IBM SP, Blue Horizon. Prerequisites: Programming background in C or Fortran 90.

Physics

Physics 580 - Computational Physics
Computer programming for numerical solution of problems in classical mechanics, electromagnetism, optics, and quantum mechanics. Use of Fortran and C programming languages and the Unix operating system. Incorporation of standard subroutines for linear algebra and differential equations into student written programs.

San Diego State University (cont.)

Physics 585 - Computer Simulations in the Natural Sciences
Computer simulations methods for investigating physical properties of classical many-particle systems. Molecular Dynamics. Monte Carlo methods. Metropolis algorithm. Applications to liquids, magnetic, biological and planetary systems. Introduction to the numerical study of phase transitions. Elements of Quantum simulations: Path Integral MD and MC. Complexity and Cellular Automata. Global Optimization methods. Neural Networks.

Contact:

Kris Stewart
Computer Science Department
San Diego State University
San Diego, CA 92182-0314
619-942-1012
stewart@cs.sdsu.edu

http://www.stewart.cs.sdsu.edu/

http://www.edcenter.sdsu.edu/

Stanford University

Degree: B.S., an honors option, and a minor

Department: Mathematical and Computational Science

Description: Mathematical and Computational Science is an interdepartmental undergraduate program designed as a major for students interested in the mathematical sciences, or in the use of mathematical ideas and analysis of problems in the social or management sciences. It provides a core of mathematics basic to all mathematical sciences and an introduction to the concepts and techniques of automatic computation, optimal decision-making, probabilistic modeling, and statistical inference. It also provides an opportunity for elective work in any of the mathematical science disciplines at Stanford.

The program utilizes the faculty and courses of the departments of Computer Science, Mathematics, Management Science and Engineering, and Statistics. It prepares students for graduate study or employment in the mathematical and computational sciences or in those areas of applied mathematics which center around the use of computers and are concerned with the problems of the social and management sciences.

Courses:

Students choose from a wide selection of courses in the associated departments. The Computer Science Department offer the following course in scientific computing.

CS 137 - Introduction to Scientific Computing
Fundamental issues of numerical computation for mathematical, computational, and physical sciences, and engineering. Emphasis is from the perspective of the computer scientist. Use of numerical algorithms in engineering practice. Problems of accurately computing solutions in the presence of rounding errors and of computing discrete approximations of solutions which are defined on the continuum. The taxonomy of problem classes with methods for their solution and principles useful for analysis of performance and algorithmic development. Topics: error analysis, the solution of linear and nonlinear equations, interpolation and numerical differentiation, the approximation of integrals, and the solution of differential equations.

Contact:

Helen Tombropoulos
Mathematical and Computational Science Department
Stanford University
Sequoia Hall
390 Serra Mall
Stanford, CA 94305-4065

Stanford University (cont.)

650-723-2620
helen@stat.stanford.edu

http://www.stanford.edu/group/mathcompsci/

State University of New York - Brockport

Degree: Bachelor of Science with a major in computational science

Department: Computational Science

Description: The SUNY College at Brockport began the Computational Science Program (CPS) in Fall, 1998 under the School of Letters and Sciences. The program was promoted to a department status in Spring 2000. The CPS department offers BS and MS degrees to its majors. Students in this program get a diploma in computational science. This is different from offering a certificate or specialization in conjunction with another science program. Undergraduate students at Brockport, however, can declare more than one major during their study which makes it an option for students who like to tightly combine computational science with another major.

CPS students take courses in computational science (listed below), mathematics, and the natural sciences. CPS students learn advanced computer skills such as the use of graphics workstations and high-performance computers. Furthermore, CPS students acquire interdisciplinary knowledge and apply scientific principles in solving real-world problems. CPS graduates are well prepared for future employment in industry, research, and academia. The department has already graduated, perhaps the first ever, undergraduates in computational science in 2001. The employment record for these students in industry (Xerox, Lockheed Martin, U.S. Navy) is quite good. One area of employment has been embedded computing where a tight combination of programming, applied math, and application sciences is needed.

Courses:

CPS 101 - Introduction to Computational Science
Prerequisites: MTH 121, or Instructor's approval. Provides an introduction to computation in science and engineering. The emphasis is on practical applications of formulas to real-life problems and on tools for their solution. The course content includes 3 distinct areas: 1) techniques (linear regression for data-fitting, determination of areas and volumes, rate changes (differentiation), use of graphical calculator), 2) programming in FORTRAN and C, 3) UNIX operating system (basic commands, editors, input/output).

CPS 201 - Computational Science Tools I
Provides an introduction to the use of computers in science, engineering and business applications for prospective computational science, computer science, mathematics, and other majors. It includes an introduction of the impact computers have on our lives; examples that help us understand how computation is recognized as a third way of doing science besides theory and experiment; examples of common applications and related industry and job market; brief introduction to high performance computing; common computation techniques in a variety of science, engineering, and business fields;

State University of New York - Brockport (cont.)

examples and brief introduction to visualization as it relates to applications and the job market. It also includes topics 1) computer performance (speed), architecture and supercomputers, 2) data representation, algorithms, programming, and compiler directives (all in FORTRAN 77), 3) visualization basics.

CPS 202 - Computational Science Tools II
Techniques and software tools commonly used in scientific computing applications. Topics include high-level programming languages such as Fortran 90 and C/C++; the UNIX operating system; general strategies for scientific computing; graphics, symbolic manipulation, and multi-purpose software packages such as MAPLE, MATLAB, and MACSYMA; numerical libraries such as BLAS, ScaLAPACK, problem solving environments such as NetSolve, industrial benchmarks, profilers, code analyzers and generators (i.e., FORGE), grid-generation techniques, and communication libraries such as PVM, MPI. Applications in chemistry, physics, and other fields are discussed. Extensive programming in F90 and C is required.

CPS 303 - High Performance Computing
Computational methods commonly used in scientific applications. Parallel programming strategies and general principles of scientific computing are illustrated in the context of numerical methods. Use of parallel supercomputers on the campus is covered. Computing topics to include are: modern computer architectures, understanding parallelism, evaluating benchmarks, parallel computing, and language support for performance. Mathematical topics to include are: differentiation, integration, and interpolation; solution methods for linear systems; calculation of eigenvalues and eigenvectors; error analysis; data fitting, regression, and smoothing.

CPS 304 - Simulation and Modeling
An introduction to continuous and discrete simulation methods used in scientific applications. Includes steps required to model and simulate a system, including discussion of generic partial differential equations and governing equations, discretization of these equations (finite difference, finite-element, spectral methods), generation of computational grid to solve these governing equations, basic numerical schemes to solve the discretized equations, specification of initial conditions, and the formulation and development of simulation problems, programming strategies, and data analysis. Representative applications include scheduling problems, molecular dynamics, weather prediction, engine combustion modeling and others.

CPS 404 - Applied and Computational Mathematics
Provides mathematical skills for the development of efficient computational methods for several topics including: elementary numerical methods and their computer implementations; linear and nonlinear equations; ordinary differential equations; initial and boundary value problems; modeling of data; statistical distributions; generation of

State University of New York – Brockport (cont.)

random numbers, discrete-event simulations; introduction to stochastic processes; Markov decision chains and applications from transportation, inventory control and health care; discrete Fourier transforms and its application to digital signal processing.

CPS 433/533 - Scientific Visualization
Provides concepts and techniques for visualization and its implementation. Specifically, use of visualization tools in mathematical simulation modeling such as data entry and data integrity, code debugging, and code performance analysis, interpretation and display of final results will be emphasized. Hands-on experience with visualization software packages in X-Windows environment will be provided. Students may be required to develop new visualization software designed to aid in the analysis of a chosen problem.

CPS 455/555 - Introduction to Computational Fluid Dynamics
This course will introduce concepts in fluid mechanics, particularly the derivation of governing equations via a control volume approach. It will focus on computational solution and numerical methods in compressible flow. Students will get hands-on experience by the use of both existing software libraries as well as the class projects involving software development in this field.

CPS 421/521 - Introduction to Computational Physics
This course will introduce basic physics laws and principles that require use of high powered computers. Examples include particle systems, structural mechanics, particle transport, plasma hydrodynamics, and chemical rate equations.

CPS 441/551 - Computational Methods in Finance (pending approval)
An introductory survey course on high performance computer modeling and simulation techniques for finance applications. This course also provides hands-on experience to computational methods and tools, such as EXPO, and MATLAB Financial Toolbox. Methods include linear algebra, finite difference solution of partial differential equations, artificial intelligence methods, genetic algorithms, expert systems, and parallel computing methods. Applications will include financial markets forecasting, trading systems, asset and option pricing.

CPS 461/561 - Introduction to Computational Biology (pending approval)
An introductory survey course on the applications of computer modeling and simulation to biological problems. Topics include molecular simulation for structure determination and dynamical properties of biological molecules, and bioinformatics. Computational tools such as Biology Workbench, MATLAB, and AMBER will be used for biological simulations.

State University of New York - Brockport (cont.)

CPS 411/511 - Introduction to Embedded Computation
This is an introductory survey course to embedded systems programming. Almost every electronic device today (cellular phones, space probes, medical devices, copy machines, etc) has a computational algorithm embedded in the hardware. This course will use high-level languages to review concepts of embedded computation. It will cover both hardware and software issues necessary to develop embedded software for an industrial product by Xerox Corporation.

Contact:

Dr. Osman Yasar, Chair
Department of Computational Science
SUNY Brockport
350 New Campus Drive, Room 294 FOB
Brockport, NY 14420
716-395-2529
oyasar@brockport.edu

http://www.cps.brockport.edu/

SUNY Institute of Technology at Utica

Degree: Undergraduate course

Department: School of Arts and Sciences, Applied Mathematics Program

Description: The Institute's program in Applied Mathematics educates students to solve real world problems in areas where knowledge of applied mathematical methods is required. This program provides students with a selection of courses focusing upon applied mathematical techniques. These courses provide a solid background in the more traditional areas of applied mathematics such as Calculus, Differential Equations, and Analysis. The elective requirements provide a broad selection of modern and cutting edge courses designed to provide our graduates with the necessary mathematical and computational skills needed to succeed in today's competitive job market. These courses allow students to focus on numerical methods and more advanced applied mathematical methods, including mathematical modeling and scientific computing.

Course:

MAT 465 - Scientific Computing

Contact:

William Thistleton
School of Arts and Sciences
SUNY Institute of Technology at Utica
Utica, New York 13504
315-792-7508
thistlet@sunyit.edu

David A. Coker
School of Arts and Sciences
SUNY Institute of Technology at Utica
Utica, New York 13504
315-792-7397
coker@sunyit.edu

http://www.math.sunyit.edu/

Syracuse University

Degree: Undergraduate minor in computational science

Departments: Computer and Information Science

Description: The Computational Science (CPS) minor provides an opportunity for students to explore the interplay between science and computation, an area which has become of great importance with the advent of supercomputers, especially parallel supercomputers. The minor is especially suitable for students majoring in the sciences, mathematics, computer science or engineering.

Courses:

CPS 211 - Computational Science I
Modeling of scientific and engineering problems, matrix methods for scientific applications, representations of functions, interpolation and extrapolation, finite difference methods, computer representation of floating point numbers and their accuracy, linear programming, introduction to modeling. Coreq: CPS 313.

CPS 213 - Scientific Programming I
Introduction to data parallel programming languages suitable for scientific and engineering applications, language constructs, array operations, language mechanics. Coreq: CPS 311.

CPS 212 - Computational Science II
Continuation of CPS 311,representative numerical methods for ordinary and partial differential equations, random numbers, Monte Carlo methods, Metropolis calculations, discrete event simulation. Coreq: CPS 314.

CPS 214 - Scientific Programming II
Design and implementation of scientific and engineering applications using data parallel techniques. Coreq: CPS 312.

Contact:

Ernest Sibert
Department of Electrical Engineering and Computer Science
Syracuse University
Syracuse, NY 13244
315-443-3110
eesibert@ecs.syr.edu

http://www.cis.syr.edu/undergraduate_program/CPS_minor.html

United States Naval Academy

Degree: Courses in computational science

Department: Center for Computational Science and Engineering (CSE) (involving Computer Science, Economics, Mathematics, Mechanical Engineering and Physics departments)

Description: The Center for Computational Science and Engineering (CSE) is a multidisciplinary organization promoting the use of computation in science and engineering with strongly positive implications for enhanced faculty research, midshipmen research and highly relevant undergraduate educational programs. In broad terms Computational Science and Engineering involves using computers to study scientific and engineering problems, complementing the areas of theory and experimentation in traditional scientific investigation. CSE combines domain expertise with expertise in modeling and computational areas such as numerical analysis, algorithm development, visualization, and software implementation.

Courses:

SM486E - Introduction to Computational Science and Engineering
A multidisciplinary team-taught course, using faculty from several departments. There will be four projects drawn from a variety of subject areas in science, engineering and economics. Each will begin with some background science and culminate in the computer-assisted solution of the resulting mathematical or computational problems. The basic idea behind the course is to introduce students to the overall scientific process of solving real problems using teams with varied expertise and experiences to proceed from practical problem to solution. The computational techniques presented will be typical of processes that are widely used in solving scientific and engineering problems so that their applicability will be much wider than the particular problems discussed.

SM481T - Numerical and Statistical Analysis of Experimental Data
This course covers data collection, errors and their propagation, curve and data fitting, parameter estimation, statistical analysis, and dangers of extrapolation.

Contact:

Reza Malek-Madani
Mathematics Department &
Center for Computational Science and Engineering
United States Naval Academy
Annapolis, MD 21402
410-293 - 6700
rmm@usna.edu

http://mathweb.mathsci.usna.edu/faculty/cse/CSE_HOME.HTM

University at Buffalo - The State University of New York

Degree: B.S. in Computational Physics

Departments: Physics and Computer Science and Engineering

Description: The recently established undergraduate degree program in Computational Physics is an interdisciplinary program between the Department of Computer Science and Engineering and the Department of Physics. The Computational Physics program is designed for students who wish to seek employment after the BS degree, as well as students who wish to pursue graduate studies. The physics content of the program is such that students can pursue a traditional graduate program in physics; on the other hand, there are now a number of graduate programs in Computational Science, for which students with the BS in Computational Physics will be well qualified. The recently established UB Center for Computational Research provides an outstanding supercomputing facility for students in this program to complete undergraduate research projects.

Courses:

Students take a variety of courses from the Physics, Mathematics, and Computer Science and Engineering departments, including the following computational physics courses:

PHY 410 - Computational Physics I
Numerical solution of problems in dynamics, electrodynamics, quantum and statistical physics. Root-finding, numerical differentiation, quadrature, matrix inversion, ordinary differential equations. Structured programming in FORTRAN 90, C++, or Java. Computer graphics.

PHY 411 - Computational Physics II
More advanced physics problems involving partial differential equations. Numerical simulation and Monte Carlo methods. Data analysis and fast Fourier transforms. Use of mathematical library routines and computer algebra programs.

Contact:

Michael G. Fuda
Department of Physics
North Campus
Buffalo, NY 14260-1500
716-645-2017 x192
fuda@acsu.buffalo.edu

http://www.physics.buffalo.edu/undergrad/cp-bs-ms.html

University of Chicago

Degrees: B.A. and B.S in Computational and Applied Mathematics from the Mathematics Department

Department: Computational and Applied Mathematics Program (CAMP). Participating departments include Mathematics, Computer Science, Physics, Astronomy and Astrophysics, and Geosciences.

Description: the Computational and Applied Mathematics program involves education and research that uses mathematical methods to study the physical world. Areas of study include reactive flows, radiative transport, quantum dynamics and spectral theory, bacterial colony growth patterns, thin films and moving contact lines, crystal growth, turbulence, and dynamical Systems and Chaos

Courses:

Students choose from a wide variety of courses from the participating departments.

Contact:

Lenya Ryzhik
Department of Mathematics
University of Chicago
5734 University Avenue
Chicago, IL 60637
773-702-0552
ryzhik@math.uchicago.edu

http://www.math.uchicago.edu/camp/

University of Colorado

Degree: Undergraduate course

Department: Computer Science

Description: A one-year course in scientific computation.

Courses:

CS 4576/4586 - High-Performance Scientific Computing 1 & 2
Introduction to computing systems, software, and methods used to solve large-scale problems in science and engineering. Students use high-performance workstations and a supercomputer.

Contact:

Elizabeth Jessup
Department of Computer Science
430 UCB
University of Colorado
Boulder, CO 80309-0430
303-492-0211
jessup@cs.colorado.edu

http://www.cs.colorado.edu/~jessup

http://www-ugrad.cs.colorado.edu/csci4576/

University of Houston-Downtown

Degree: Undergraduate course

Department: Computer and Mathematical Sciences

Description:

A parallel computing course that includes:

- A comprehensive introduction to different architecture models of parallel computers. Materials on processor-memory organizations, interconnection network configurations, and basic communication operations.
- Development of efficient parallel algorithms for some application problems on parallel computers to best suit given hardware architecture.
- Implementation of parallel algorithms on realistic parallel machines.
- Performance analysis for parallel algorithms and their implementation on parallel machines.

Course:

CS 4328 - Parallel Computing
Consideration of the issues involved in various forms of parallel computing including pipelines, vector processors, multiprocessor systems, related algorithms and parallel distributed processing (neural networks).

Contact:

Department of Computer and Mathematical Sciences
University of Houston-Downtown
Houston, TX 77002
713-221-8012

http://cms.dt.uh.edu/

University of Minnesota

Degree: Undergraduate courses

Departments: Computer Science and Engineering, Physics

Description: See course descriptions

Courses:

CSci 1107 - Introduction to FORTRAN Programming for Engineers and Scientists
Algorithm development and principles of computer programming using FORTRAN. Topics covered include introduction to computers and computing, program development, FORTRAN programming language syntax, and elementary numerical methods for scientists and engineers.

CSci 1113 - Introduction to C/C++ Programming for Engineers and Scientists
This course covers algorithm development and the principles of computer programming using C and C++. Topics include introduction to computers and computing, program development, C/C++ programming language syntax, and elementary numerical methods for scientists and engineers.

CSci 5109 - Visualization
Fundamental theory/practice in data visualization. Emphasizes programming applications. Volume visualization, vector field visualization, information visualization, multivariate visualization, visualization of large datasets, visualization in immersive virtual environments, and perceptual issues in effective data representation. Projects are implemented in C++ using VTK or similar visualization API.

CSci 5451 - Introduction to Parallel Computing: Architectures, Algorithms and Programming
Parallel architectures design, embeddings, routing, examples of parallel computers, fundamental communication operations, performance metrics, parallel algorithms for sorting, matrix problems, graph problems, dynamic load balancing, types of parallelisms, parallel programming paradigms, message passing programming in MPI, data parallel programming in HPF, shared-address space programming in threads.

CSci 5481 - Computational Techniques for Genomics
Techniques to analyze biological data generated by genome sequencing, proteomics, cell-wide measurements of gene expression changes. Algorithms for single/multiple sequence alignments/assembly. Search algorithms for sequence databases, phylogenetic tree construction algorithms. Algorithms for gene/promoter and protein structure prediction. Data mining for micro array expression analysis. Reverse engineering of regulatory networks.

University of Minnesota (cont.)

Phys 5041/5042 - Analytical and Numerical Methods of Physics I/II
The course aims to introduce the students to analytical and numerical mathematical methods and their application in physics. The course will cover basic material like complex analysis, Fourier transforms, differential equations, probabilities, asymptotic analysis. Computational methods will be discussed, such as Mathematica - symbolic calculations with a computer. An additional goal of the course will be to expose the students to applications of these methods in contemporary research.

Contact:

Charles D. Swanson
Computer Science and Engineering
University of Minnesota
200 Union Street SE
Minneapolis, MN 55455
612-625-0330
cswanson@cs.umn.edu

http://www.cs.umn.edu

University of Rochester

Degree: Undergraduate courses

Department: Physics and Astronomy

Description: See course descriptions.

Courses:

PHY 322A - Computational Physics I
Introduction to modeling of simple physical systems by computer. Introduction to basic numerical methods to solve integrals, differential equations, boundary and eigenvalue problems of physics, and application of these techniques to contemporary topics in physics. Error propagation and numerical considerations; realistic projectile motion; oscillatory motion and chaos; potentials and fields; waves; random numbers and evaluation; random walk and diffusion; cluster growth; fractual and percolation; Ising model and Monte Carlo method; time-independent; Schrodinger equation; variation approach. This is a two-credit course held the first six weeks of the semester. May be taken independently of PHY 322B.

PHY 322B - Computational Physics II
Introduction to use of computer control, interfacing, and data acquisition in the laboratory. Topics include introduction to digital electronics, interface devices, data conversion devices, A/D converters, I/O ports, interface standards, microprocessor basics, introduction to P-Basic, and application of microprocessor with PC. This is a two-credit course held the last six weeks of the semester. May be taken independently of "322A". The order of the materials may change between 322A and 322B.

Contact:

University of Rochester
Department of Physics & Astronomy
Bausch & Lomb Hall
P.O. Box 270171
600 Wilson Boulevard
Rochester, NY 14627-0171
585-275-4351
shirl@pas.rochester.edu

http://www.pas.rochester.edu

University of Wisconsin - Eau Claire

Degree: Minor in Computational Science

Departments: Science, Mathematics, Computer Science

Description: There is a need for individuals who have an advanced knowledge of a particular discipline and a significant understanding of the role that computation plays in solving the complex problems of that discipline. Students who study the computational methods needed to solve problems such as those inherent in the human genome project, or in creating systems that model global warming, or who study Geographic Information Systems, or who process images taken by satellites or planetary probes will be increasingly valuable. The Computational Science minor is an interdisciplinary minor that is available to all students at UWEC.

Courses:

Core computational science courses

CS170 - Computing for the Sciences and Mathematics
Aspects of computation that are particularly relevant to scientific computation. Discussion of numeric processing, symbolic processing, data management, data representation, and scientific visualization. Students design and implement computer programs using a language specifically tailored to the sciences.

CPSC250 - Computational Science I
An introduction to the architecture, communication, design, and implementation of computational science systems. Topics include fundamental design principles of computer science including an introduction to programming in a higher level language with practical course projects taken from physics, chemistry, biology, geology, and geography.

CPSC300 - Computational Science II
Case studies of problems in computational science. Examples include molecular modeling, computational chemistry, stellar and planetary systems, thermodynamics, seismic modeling, geographic information systems, and atmospheric/oceanic circulation models. Solutions involve various computational tools and techniques.

CPSC494 - Computational Science Practicum

MATH 245 - Introduction to Statistics
Basic statistical analysis, including descriptive statistics, probability, confidence intervals, hypothesis testing, simple linear regression, and correlation.

University of Wisconsin - Eau Claire (cont.)

MATH 354 - Introduction to Mathematical Modeling
Graphs as models, the modeling process, proportionality, model fitting, optimization, experimental modeling, dimensional analysis, probability and Monte Carlo simulation, modeling using the derivative, numerical methods, interactive dynamic systems.

Contact:

Andrew T. Phillips
Department of Computer Science
University of Wisconsin - Eau Claire
131 Phillips Science Hall
Eau Claire, WI 54702-4004
715-836-2526
phillips@cs.uwec.edu

http://www.cpsc.uwec.edu

University of Wisconsin - LaCrosse

Degree: Minor in Computational Science

Department: The College of Science and Allied Health sponsors an academic minor in Computational Science.

Description: See course descriptions.

Courses:

CMP 390 - Survey of Computational Science
This course will survey the computational and mathematical tools and techniques currently being applied to problems in the sciences. Specific problems drawn from biology, chemistry, meteorology and physics will be explored in detail. Computational tools such as Mathematica, Explorer and PVM will be introduced and used to solve problems. In addition to small projects, students will be required to complete a larger project selected from their major discipline.

CMP 490 - Senior Computational Science Project
This course is a capstone course for students pursuing a computational science minor. The student is expected to pursue a project that integrates a problem(s) from their major scientific discipline together with computation. Such work should demonstrate the student's ability to apply the tools and techniques acquired from prerequisite study in science and computation. The work must be performed under the direction of a faculty member from the student's major department. The student is also expected to submit a paper and an oral presentation on the project results to the computational science committee.

Contact:

David Riley
Computer Science Department
275 Morris Hall
University of Wisconsin - La Crosse
La Crosse, WI 54601
608-785-6806
riley@cs.uwlax.edu

http://www.uwlax.edu/SAH/CMP/

Wittenberg University

Degree: Computational Science Minor

Department: Mathematics and Computer Science Department

Description: The Computational Science Minor curriculum includes a course in *Computational Models and Methods* and a capstone computational science project along with courses from the Mathematics and Computer Science department and science departments.

Course:

MATH/COMP 260 - Computational Models and Methods
Introduction to the principles and approaches of computational science through the use of standard scientific problem solving methods. This includes the understanding, development, and use of mathematical models as well as their effective computer implementation using languages such as Mathematica, C/C++, and FORTRAN. Approximately fifteen models in eight categories (continuous vs. discrete, static vs. dynamic, deterministic vs. stochastic) will be investigated. These models are adapted from a variety of scientific disciplines. Simulation and optimization techniques will also be discussed and used, together with the study of such topics as the SI System of physical units and floating-point arithmetic. Appropriate visualization strategies will be discussed. Each student will undertake a realistic modeling project in one of the sciences. Laboratory required.

Contact:

James L. Noyes
Mathematics and Computer Science Department
Wittenberg University
Post Office Box 720
Springfield, OH 45501
937-327-7858
jnoyes@wittenberg.edu

http://www4.wittenberg.edu/academics/mathcomp/computational_science/index.html

Wofford College

Degree: Emphasis in Computational Science

Department: Computer Science

Description: The Emphasis in Computational Science was designed with scientists and for science majors. Applications discussed in the computational science courses are scientific in nature. The Emphasis is truly interdisciplinary among science, computer science, and mathematics. Students major in one of five mathematics or science disciplines and complete a summer internship involving computing in the sciences. The five required courses include two interdisciplinary courses developed for the program, described below.

Courses:

<u>CS 370 - Data and Visualization</u>
Data and Visualization is part of the interdisciplinary field of computational science. The course contains a brief introduction to the network environment and the UNIX operating system. Because large Web-accessed databases are becoming prevalent for storing scientific information, the course will cover the concepts and development of distributed relational databases. Effective visualization of data helps scientists extract information and to communicate results. Thus, students will learn fundamental concepts, tools, and algorithms of computer graphics and scientific visualization in two and three dimensions. Throughout, applications in the sciences will be emphasized. Prerequisites: Computer Science 350 or permission of the instructor.

<u>CS 375 - Scientific Programming</u>
Scientific Programming is part of the interdisciplinary field of computational science. Large, open-ended scientific problems often require the algorithms and techniques of discrete and continuous computational modeling and Monte Carlo simulation. Students will learn fundamental concepts and implementation of algorithms in various scientific programming environments: a computer algebra system, distributed computation, and parallel processing. Throughout, applications in the sciences will be emphasized.

Contact:

Angela B. Shiflet
Computer Science Department
Wofford College
429 North Church Street
Spartanburg, SC 29303
864-597-4528
shifletab@wofford.edu

http://wofford.org/computerscience/emphasis_in_cs.htm

APPENDIX 2
GRADUATE PROGRAMS IN COMPUTATIONAL SCIENCE*

Baylor College of Medicine	PhD in Structural & Computational Biology & Molecular Biophysics
Boston University	CSE courses
California Institute of Technology	PhD in Applied Mathematics with several concentrations
Carnegie-Mellon University	Minors in Science Departments
Clemson University	CSE courses
Colorado State University	CSE courses
Cornell University	CSE courses; PhD in Applied Mathematics include Computational Science
Duke University	CSE certificate
Florida State University	PhD from home department; CSE department degrees
George Mason University	PhD with research programs in many scientific fields
George Washington University	Graduate Certificate and Master's Degree Program
Louisiana Technical University	Interdisciplinary PhD in Applied Computational Analysis and Modeling
Louisiana State University	Dual Physics PhD/Computer Science MS
Massachusetts Institute of Technology	CSE courses
Mississippi State University	MS and PhD in Computational Engineering
New York University, Courant Institute	MS in Scientific Computing
North Carolina State University	Specialty degree as an extended minor
The Ohio State University	CSE courses
Princeton University	PhD, Program in Applied and Computational Mathematics
Purdue University	MS, PhD in traditional department with a specialization in CSE
Rensselaer Polytechnic Institute	MS and PhD specialty
Rice University	Specialty degree
San Diego State University	MS in Computational Science
Stanford University	MS and PhD, Scientific Computing and Computational Mathematics
Syracuse University	Internetics program
University of Arizona	PhD minor
University of California at Davis	CSE program started in 2000
University of California at San Diego	PhD, Program in Scientific Computation (planned)
University of Colorado	CSE courses
University of Illinois, Chicago	PhD and MS degrees
University of Illinois, Urbana-Champaign	MS, PhD in traditional department with a specialization in CSE
Indiana University at Bloomington	PhD minor in Scientific Computing
University of Iowa	Interdisciplinary PhD, department of Applied Mathematics and Computer Science
University of Louisiana, Lafayette	MS and PhD in Computational Science and in Computational Engineering
University of Michigan	PhD in discipline with "..and Scientific Computing"; MS in Scientific Computing
University of Minnesota	MS, PhD in Scientific Computing
University of Nevada	CSE courses
University of New Mexico	MS, PhD in home department with a CSE certificate
University of North Carolina, Chapel Hill	Interdisciplinary programs
University of Oregon	Interdisciplinary CSE Institute
University of Pittsburgh	PhD in Computational Biology
University of San Francisco	Computational science thesis for MS

University of Texas at Austin	MS and PhD in Computational and Applied Mathematics
University of Utah	Graduate certificate in computational engineering and science
University of Wisconsin	MS in CSE
Vanderbilt University	CSE courses

* Martha Lee Ennis, *Update on the Status of Computational Science and Engineering in U.S. Graduate Programs*, Albuquerque High Performance Computing Center, The University of New Mexico, AHPCC99-023, September 28, 1999.

Links to the programs mentioned in the report are available in the web version of this paper at http://www.krellinst.org/.