SIAM REVIEW
Vol. 53, No. 3, pp. 000–000

Undergraduate Computational Science and Engineering Education*

SIAM WORKING GROUP ON CSE UNDERGRADUATE EDUCATION
Peter Turner[†] and Linda Petzold,[‡] Co-Chairs
Angela Shiflet[§]
Ignatios Vakalis[¶]
Kirk Jordan[‖]
Samuel St. John[**]

Abstract. It is widely acknowledged that computational science and engineering (CSE) will play a critical role in the future of the scientific discovery process and engineering design. However, in recent years computational skills have been *deemphasized* in the curricula of many undergraduate programs in science and engineering. There is a clear need to provide training in CSE fundamentals at the undergraduate level. An undergraduate CSE program can train students for careers in industry, education, and for graduate CSE study. The courses developed for such a program will have an impact throughout the science, technology, engineering, and mathematics (STEM) undergraduate curriculum. This paper outlines the content of a CSE curriculum, the skills needed by successful graduates, the structure and experiences of some recently developed CSE undergraduate programs, and the potential career paths following a CSE undergraduate education.

Key words. computational science and engineering, education, undergraduate programs

AMS subject classifications. 97, 65, 68

DOI. 10.1137/07070406X

1. Introduction.

In many areas of science and engineering, computation has become an equal and indispensable partner, along with theory and experiment, in the quest for knowledge and the advancement of technology. Numerical simulation enables the study of complex systems and natural phenomena that would be too expensive or dangerous, or even impossible, to study by direct experimentation. An increase during the past 30 years of over six orders of magnitude in computer speed, and another six orders of magnitude in algorithm speed, along with advances in mathematics in understanding and modeling complex systems, and in computer science of manipulating and visualizing large amounts of data, has enabled computational scientists

*Received by the editors October 1, 2007; accepted for publication (in revised form) April 13, 2011; published electronically August 5, 2011.
http://www.siam.org/journals/sirev/53-3/70406.html
†Clarkson University, P.O. Box 5800, 305 Snell Hall, Potsdam, NY 13699 (pturner@clarkson.edu).
‡Department of Computer Science, University of California–Santa Barbara, Santa Barbara, CA 93106 (petzold@engineering.ucsb.edu).
§Department of Computer Science, Wofford College, 429 North Church St., Spartanburg, SC 29303 (shifletab@wofford.edu).
¶Department of Computer Science, California Polytechnic State University, San Luis Obispo, CA 93407 (ivakalis@csc.calpoly.edu).
‖IBM Corporation, 1 Rogers Street, Cambridge, MA 02142 (kjordan@us.ibm.com).
**University of Cincinnati, Rhodes Hall, Cincinnati, OH 45221 (s4mwise@mac.com).

and engineers to solve large-scale problems that were once thought intractable. It is widely acknowledged that computational science and engineering (CSE) will play a critical role in the future of the scientific discovery process and engineering design [2, 3, 4, 6]. Computation informs policy makers in areas as diverse as climate change, public health, and environment.

A recent international study [2] found that a worldwide shortage of scientists and engineers trained in the fundamentals of CSE is a bottleneck for progress in science and technology. The shortage exists at all levels and in all sectors: industry, academia, and education. There is a clear need to provide training in CSE at both the undergraduate and graduate levels, but what form should that training take, and what should be its objectives? A previous SIAM report [7] outlined the issues and set an agenda for CSE graduate education. In this paper we focus on undergraduate CSE education and describe some of the nascent efforts in this area.

Why should you be interested?

1. CSE graduate programs of one form or another are widespread in the U.S. and Europe [7], although the number of students they are attracting is modest. Why? This may be explained in part by the fact that the vast majority of incoming science, technology, engineering, and mathematics (STEM) graduate students have never even heard of CSE, because in most institutions it does not exist as a well-defined subject area in the undergraduate curriculum.
2. Undergraduate courses developed for CSE programs can provide an important foundation of analytical and computational skills for traditional engineering and science majors. These courses can also be an important resource for beginning graduate students in engineering and science. We note that programming is no longer a part of the engineering curriculum in many U.S. undergraduate engineering programs.
3. CSE education is an opportunity to attract a more diverse student body into computing. The number and proportion of female undergraduates in computing fields has been declining in recent years. CSE, and especially CSE applied to the biological sciences, typically attracts a much higher proportion of female students.
4. Graduates trained in CSE who choose a career in K-12 teaching will be a unique resource in the educational system because they will understand the connection between mathematical and computing tools with real-life scientific and engineering applications.

The remainder of this paper is organized as follows. In section 2 we outline core competencies for undergraduate CSE education and examine some of the different models for CSE undergraduate programs. Section 3 highlights the valuable role that internship programs can play. Section 4 outlines the needs that undergraduate CSE education should address to prepare students for careers in industry, K-12 education, or for further training in graduate school. In section 5 we present a case study of an industrial career path that illustrates the opportunities and needs for undergraduate CSE education.

2. CSE and Undergraduate Education.

2.1. Introduction. What is CSE? In [7], computational science and engineering is defined as

> a broad multidisciplinary area that encompasses applications (science/engineering), applied mathematics, numerical analysis, and computer science and engineering. Computer models and computer simulations have be-

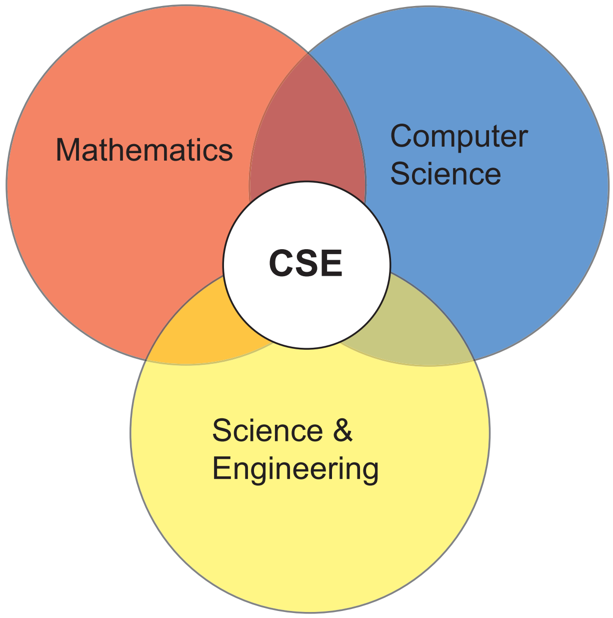

Fig. 1 *CSE includes,* but is greater than*, the intersection of mathematics, computer science and science & engineering.*

> come an important part of the research repertoire, supplementing (and in some cases replacing) experimentation. Going from application area to computational results requires domain expertise, mathematical modeling, numerical analysis, algorithm development, software implementation, program execution, analysis, validation, and visualization of results. CSE involves all of this.
>
> Although it includes elements from computer science, applied mathematics, engineering, and science, CSE focuses on the *integration* of knowledge and methodologies from all of these disciplines, and as such is a subject which is distinct from any of them.

The graphical representation of CSE in Figure 1 illustrates our view that CSE is larger than the pure intersection of the three component pieces, but is nonetheless included in their union.

We believe that the undergraduate arena is the most important segment of the educational pipeline, since it prepares the science/math teachers for the high school environment, invigorates students to pursue graduate studies in cutting edge technical fields, and produces a vast number of future employees for industry and the "knowledge-based" economy. Decision makers in industry and elsewhere will be relying on CSE results; we should ensure that they have an understanding of where they come from. Therefore, it is critical that computational science courses and curricula are a viable option for every undergraduate STEM major.

Figure 2 shows the central position of undergraduate CSE education in the pipeline. It is the one place that feeds three different markets. The primary objectives of preparing students for graduate studies in CSE and for careers in industry are joined by a potentially critical contribution: preparation of teachers for the K-12 system who have a thorough appreciation of the integrated nature of the STEM disciplines and

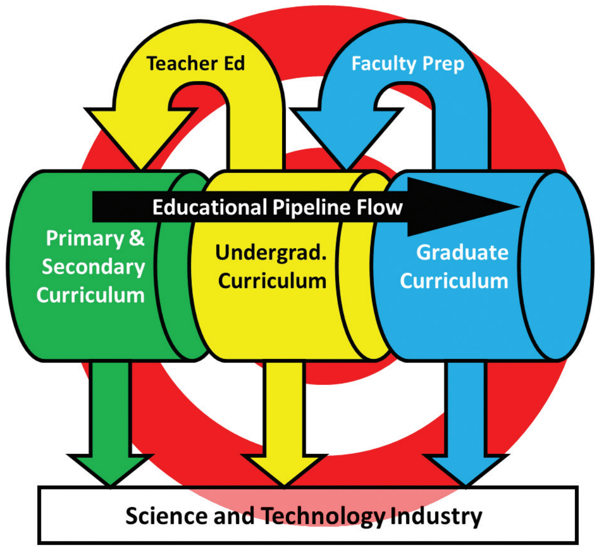

Fig. 2 *The CSE educational pipeline.*

the use of relevant applications and technology in problem solving for mathematics and science education.

2.2. Core Competencies and Models for CSE Programs. Current undergraduate CSE programs take a number of different forms, including B.S. degree in CSE; Minor program in CSE; Emphasis or Concentration in CSE; B.S. degree in Computational X (where X = STEM discipline or Finance).

Common features of most CSE programs include a core collection of courses, including Calculus (two-course sequence); Programming (at least one course); Computational Modeling; Numerical Analysis (or Scientific Computing); and either a course in Visualization or a more advanced course in Computational Modeling. Most CSE programs require an independent learning experience in the form of a capstone project, an industrial internship, or an undergraduate research experience. Projects may be single or team-based and include some form of written or oral presentation at the internship site and on campus, or at a professional conference.

An approach based on a set of core competencies has been implemented for the development of CSE statewide programs based on a set of common competencies by the Ralph Regula School of Computational Science (http://www.rrscs.org) in Ohio, a statewide virtual school focused on the emerging and diverse area of CSE. The school is directed by the Ohio Supercomputer Center (http://www.osc.edu) under the auspices of the Ohio Board of Regents. Its long-term education mission is to infuse CSE in all segments of the educational pipeline (K-20), including the development of associate degrees as well as certificate programs for adult learners. The virtual school has developed a set of competencies and standards for a statewide CSE curriculum at the undergraduate level. The competencies include the following areas: simulation and modeling (conceptual models, accuracy, use of modeling tools, assessment of computational models, team-based projects, effective technical analysis and presentation); programming and algorithms (a high-level language, elementary data structures and analysis); applied mathematics (concepts in a calculus sequence as well as differential equations and discrete dynamical systems); numerical methods (errors, nonlinear equations, solving systems of linear equations, interpolation, curve fitting, optimization, Monte Carlo, ODEs, and PDEs); parallel programming (knowledge of MPI and OpenMP); scientific visualization; research experience (independent research, presentation of solution methodologies).

It should be noted that the successful development of a specific style of a CSE program depends on the structure and mission of a particular university, the collection of faculty expertise, and most importantly on pragmatic considerations (i.e., which and how many courses can be approved by the institution? what are the local politics?). The authors of this report have observed that although the number of B.S. degree curricula in CSE (or Computational X) has been increasing, the establishment of a minor program in CSE is more pervasive and minors are often easier to implement. Some reasons to support this view include the following: (i) CSE is a multidisciplinary area and a minor program in CSE complements any traditional STEM major (the latter provides the necessary disciplinary depth); (ii) a minor program is not viewed as a threat to well-established traditional majors; (iii) a CSE minor that contains an array of Computational X courses can serve as a common arena for true multidisciplinary collaborations of faculty and students that belong to different STEM-based departments; it can also serve as a catalyst for reducing (or even eliminating) existing compartmentalization among departments.

Yet another model has begun to emerge as an alternative to the "Discipline Major–CSE Minor" model. As a result of the observed need for developers of CSE solutions to have a deeper understanding of the underlying mathematics and computer science, there is support for a "Computational Applied Mathematics major–Applications Field minor." The major part of this program would not be the traditional mathematics major, though it would certainly include significant pieces of it. It would have a strong emphasis on applied mathematics with a larger than usual computational component. These fields offer good opportunities for project-based learning and teamwork as well as exposure to the relevance of mathematics to real-world problems. This major entails a greater exposure to computer science including high-level language programming, data structures and algorithms, and scientific visualization as outlined for the general CSE content above. The applications field can be in any STEM discipline or a more general engineering science. Again, whether it is called a minor, a second discipline, a concentration, or an emphasis will vary according to local terminology.

2.3. Representative CSE Programs. The following give a short overview of some representative CSE programs at the undergraduate level. The intention is to provide the reader with samples of existing styles of successful CSE programs.

B.S. with a Major in Computational Science, SUNY College at Brockport
http://cps.brockport.edu

Students take courses in computational science (computational tools, computational modeling and simulation), applied computational mathematics, and support courses in a variety of STEM disciplines. The program also includes courses in scientific visualization and high-performance computing, along with an array of electives in Computational X fields (X = Physics, Fluid Dynamics, Chemistry, Biology, Finance). Undergraduate research experience is required.

Computational Science and Engineering Bachelors Program, ETH Zurich
http://www.cse.ethz.ch

In 2008 the first freshman students began a major in CSE at ETHZ. Nearly 30 faculty members participate in the program, which is hosted by the Department of Mathematics and Physics. In addition to the CSE degree program, students specialize in CSE (Masters) while enrolled in a traditional degree program. Many graduate students and postdocs from multiple departments take the senior-level undergraduate CSE courses, particularly parallel programming. The program has spawned a new course in Computer Science on high-performance computing.

Minor in Computational Science, Capital University
http://www.capital.edu

The minor program was established in 2004 and has been supported by a number of grants (i.e., National Science Foundation, W.M. Keck Foundation). The curriculum was developed by the collaboration of all math- and science-based departments within the college of A&S along with the Finance and Economics departments within the School of Business. The curriculum contains a set of core courses (Computational Science, Programming, Differential Equations/Dynamical Systems, Numerical Methods) and a set of electives (Computational X, Parallel and High-Performance Computing, and Scientific Visualization). Specialized courses include Computational Biology, Chemistry, Environmental Science, Physics, Psychology, Finance, and Economics. Undergraduate research experience is required.

Minor in Computational Science, University of Wisconsin–Eau Claire
http://www.cs.uwec.edu

The interdisciplinary curriculum at this liberal arts college was developed with the collaboration of the Biology, Chemistry, Computer Science, Mathematics, Geography, and Physics/Astronomy departments. It consists of a calculus sequence, a two-course sequence in computational modeling, a course in mathematical modeling, and a course in numerical methods. A computational science practicum is required.

Minor in Computational Science, Clarkson University
http://www.clarkson.edu

This minor serves as an example of bringing together existing courses in applied mathematics, computing, and applications fields to create a minor in a largely science- and engineering-based institution. The requirement for an internship is part of the university's graduation requirements and so is not included in the minor.

Emphasis in Computational Science, Wofford College
www.wofford.edu

At Wofford College the Emphasis in Computational Science is a truly interdisciplinary program among science, computer science, and mathematics. Students major in one of the math or science disciplines and complete a required summer internship in a CSE subfield. Required courses for this program include Programming, Data Structures, Calculus I, Modeling and Simulation, and a course in Data and Visualization.

B.S. in Computational Physics, Oregon State University
http://www.physics.orst.edu

The Oregon State Board of Higher Education approved the Computational Physics degree in October 2001. The specialized B.S. degree includes a two-course sequence in

Scientific Computing, a course in Computational Physics Simulation, a course labeled Advanced Computational Physics Laboratory, and a Computational Physics seminar. The degree program offers a well-balanced blend of physics, computational modeling, computer programming, and applied mathematics courses.

B.S. in Computational Biology, Florida State University
http://www.cs.fsu.edu

Under the auspices of the Department of Computer Science, Florida State University offers a Bachelor's degree in Computational Biology. The overall goal of this program is to give students the broad-based education that is needed to create a set of models directed towards solving practical biological and biomedical problems. The comprehensive undergraduate degree includes courses in Programming, Data Structures, Algorithms, Data Bases, Theory of Computation, Bioinformatics, as well as a cadre of courses from various fields of biology.

B.Sc. in Computational Engineering, Universitaet Erlangen-Nuernberg
http://www.ce.uni-erlangen.de/lang-pref/en

The Erlangen Computational Engineering program includes a three-year Bachelor degree and Master and Ph.D. degrees. The objective of the CE program is to give students a genuine interdisciplinary education from the ground up. It consists of roughly equal number of credits in Mathematics, Computer Science, and an application field that is currently limited to one of the engineering disciplines (Micro Electronics, Automatic Control, Thermo- and Fluid Mechanics, Sensor Technology, Material Sciences, Applied Chemical Engineering). Students are required to take a selection of core courses that include the traditional four-semester engineering math sequence and the most fundamental courses from the computer science curriculum.

3. The Value of Internships. Internships can be extremely valuable for a student in any area, but particularly in CSE, where interdisciplinary teams work on large, "real-world" problems. Internship experiences expose students to a wealth of new ideas, techniques, and applications that enhance their knowledge of CSE and make classroom education more meaningful. As an advantage to the host institution, undergraduate interns can make significant contributions to its research effort. Moreover, a student can leverage an internship to obtain subsequent professional experience, including admission to a better graduate school program, a graduate assistantship or fellowship, or a better professional position than would have been possible otherwise. As an added benefit, most internship positions allow students to visit different areas of the country and to meet students from other parts of the country or world. For these reasons, most undergraduate CSE programs encourage students to obtain summer internships or research experiences, and a few require an internship.

Considering the importance of this component of a CSE program and the inexperience of students in pursuing positions, faculty members often must actively help in finding opportunities, developing research proposals, and making professional presentations following internships. Advisors must encourage the CSE student to start the process early in the fall, because many deadlines are early and the application process can be time consuming. Because positions are very competitive, the student usually should apply for a number of internships.

Some specific internship experiences are detailed in [8]. Faculty members' recommendations often play a major role in selecting interns. It is important that letters mention specific accomplishments of a student, especially if they are relevant to a position. Unfortunately, some created or even established positions, particularly if far from the home institution, do not provide adequate funding for the student. Thus, it is very helpful if the home institution can provide supplementary money for worthwhile projects.

After the internship, the student's advisor should contact the intern's mentor to obtain an evaluation of the student's performance. Such an evaluation can provide valuable insights to note in recommendations for the student's future studies and career. Moreover, the conversation can help to build rapport and increase the likelihood of other internships at the same organization.

Faculty members should also have the interns give presentations on their work. Student discipline-related organizations, such as local student chapters of SIAM, the Association for Women in Mathematics, the Association for Computing Machinery, or Tri-Beta, are usually delighted to have presentations. Students and faculty can learn from the presentations, which might also inspire other students to pursue CSE or seek internships. Many conferences have sessions for undergraduate papers or posters. For example, SIAM's Computational Science and Engineering Conference has a series of minisymposia on undergraduate CSE research. A few conferences and organizations, including SIAM, provide assistance to some students making presentations, and universities and colleges also might pay all or part of students' expenses.

Internship opportunities can also develop when faculty collaborate with industry colleagues. A particular project may be spun off for a student who already has a significant background or can be prepared by the faculty member so that the student can make substantial progress in a short time. Internships often lead to careers.

4. Career Preparation. As with many undergraduate programs, there is a conflict between providing the necessary academic depth of background for potential graduate studies and preparing students for the workplace. We address preparation for the primary career paths below.

Industry. CSE has been a factor in the aerospace, automotive, chemical, computer, electronics, petroleum, and pharmaceutical industries for some time. Industries that might not come immediately to mind that are now using CSE include banking and finance, digital media (especially in content creation), consumer products, manufacturing and processing, and even transportation. With the impact that computation has had in the sequencing of the human genome, CSE is playing an increasing role in the life sciences and healthcare. The need for trained computational scientists at all levels in the healthcare and life sciences sectors continues now and for the foreseeable future to outpace the supply.

When industry seeks to fill a position, they look for a person who has the expertise to do the immediate task and the versatility for future—as yet to be determined—assignments. The breadth and multidisciplinarity of a CSE education are excellent preparation for the needed adaptability and versatility. But it is also important for team members to have a niche where they are particularly well placed to contribute. For this reason, it is important that training in CSE emphasize, in addition to the core CSE skills and multidisciplinary training, a strong foundation in a traditional discipline.

Training in CSE provides the type of background that industry often seeks in an individual to hire. CSE undergraduate training emphasizes problem-solving skills. In addition, a CSE undergraduate is often exposed to working as part of a team. In industry, individuals with a variety of backgrounds work together. The team experience for the CSE undergraduate provides the opportunity to learn how to communicate ideas and concepts quickly and persuasively. It also helps a student to see how one's work fits into a bigger effort and how to communicate the impact of one's individual effort to the larger project.

Research and Development (Graduate School). It is important to recognize that CSE graduate programs, like CSE undergraduate programs, take a number of different forms [7]. They range from degree-granting CSE programs or departments emphasizing research in computational and applied mathematics and high-performance computing, to degree emphases (the graduate degree is from a traditional department, with a formal emphasis in CSE) that are spread across the entire spectrum of science and engineering departments.

The breadth and multidisciplinarity of a CSE education are excellent preparation for the increasingly multidisciplinary research environment found in many graduate schools. Just as in industry, it is also important for incoming graduate students to have acquired some depth in the scientific or engineering discipline (or mathematics or computer science) to which they intend to apply or develop CSE methodology.

K-12 Teaching. Precollege teaching is an important area into which interested students should be encouraged to move. While this may not have significant impact on the course content, it does provide additional opportunities for professional experience through educational outreach programs. For example, through a variety of state and federal grant programs, students at Clarkson University have opportunities to experience classroom teaching in several ways. Clarkson undergraduate students work as coaches for local school MATHCOUNTS teams as either paid or course-credit-bearing internship opportunities under the New York state–funded Mathematics Science Partnership, the *St. Lawrence County STEM Partnership* (http://stlawcostempartnership.org/). Others are funded by this same grant to work with robotics or high school COMAP teams. Another New York state–funded STEP (Science & Technology Entry Program) program based on an integrated mathematics and physics project on roller coaster design offers further opportunities for students to gain teaching experience through after-school activities and a summer camp. This project encompasses the full range of CSE experience incorporating mathematics, science, and computational experiments and discovery. Similar programs exist in most states and at the U.S. national level. Professional experiences of this type can be valuable in improving students' understanding and their communication skills in particular. Such internships can be used as an alternative to more conventional industrial or academic research experiences for students who are considering a career in teaching. Increasing the flow of K-12 system teachers with a good understanding of applications and computation is an important aspect of the CSE pipeline in Figure 2.

Students who wish to pursue a career in K-12 teaching will typically need to investigate standard teacher certification opportunities after graduation, unless they are in schools with undergraduate teacher education programs.

5. Potential Career Paths: A Case Study. In many senses the opportunities and needs for students embarking on CSE careers can be summarized by the experiences described below by one recent student who graduated as a Chemical Engineering major *before the CSE minor at Clarkson was available.* The following story describes how he has adjusted and how his very first significant assignment was one which fits squarely in the CSE area. There is little doubt that Sam would have added the CSE minor. For the rest of this section, Sam tells his story in his own words.

"I graduated from Clarkson University in Potsdam, NY, and worked for The

Fig. 3 *Sam on the Great Wall.*

Procter & Gamble Company in Cincinnati, OH, for five years. I worked as an engineer in R&D designing new Oral Care products after graduating with a bachelor's degree with great distinction in chemical engineering and a minor in mathematics from Clarkson University in 2003.

"I had the fortune of living with a math major, a mechanical engineer, a computer scientist, and an electrical engineer for two years. This sort of cross discipline exposure helped me appreciate the similarity of the solutions. That may seem like an odd statement, but so many of the problems that my contemporaries and I faced could be modeled using similar fundamental equations that success was driven by those who had the greatest familiarity with the tools to process the data. Mathematics is the underlying tool used to understand relationships spanning topics from Nicomachean Ethics by Aristotle to multivariate data analysis with multiple bases; computers are the workhorses that allow us to analyze the data. Understanding how to build the computational tools separates a good engineer from a great engineer.

"Let's frame some of my contemporary challenges in the trials of an engineering education. My background is chemical engineering; I currently use my education to develop upstream whitening technologies for the Crest brand. One of the benefits of working at P&G is that I am afforded the opportunity to return to Clarkson to recruit fellow alumni to work at P&G. On nearly every return visit I am asked by students or professors what classes have served me best, what was lacking, or what would have been useful? I have one consistent message that I truly believe—modern engineers must have three definite strengths related to computational mathematics: (1) engineers must understand underlying equations governing the first principles of mechanical and chemical systems; (2) engineers must understand how to interpret experimental data and relate them to first principles; and (3) engineers must be able to craft the tools using mathematical modeling/computational software to design data collection systems, interpret experimental data, or make mathematical extrapolations to first principles. There can no longer be two separate schools of thought keeping algebraic analysis and numerical analysis separate; they must be taught conjoined. Now, the above discourse may seem generic enough to encompass the entirety of an undergraduate engineering education. That conclusion is simply not true—too few students take the key courses to be able to function across the three criteria I have listed. These courses include (in addition to the standard engineering curricula) applied linear algebra, applied statistics and linear regression analysis, MATLAB/C++ introduction/intermediate computer programming, boundary value problems, and computational logic. The value of a rigorous mathematics and numerical computation minor cannot be understated. Let me use two examples requiring knowledge spanning the three categories that I encountered within 18 months of working at P&G.

"The most recent example is one where I had to compare data collected using the same experimental design but collected at two different times. I was collecting information on the absolute rate of surface roughness change of an enamel substrate. The rate data are nonlinear in time and naturally contain some experimental variation. The rates of the control groups between the two runs were different enough to prevent direct comparison of the experimental treatments without any data manipulation. Admittedly, this is a relatively simple problem to solve, but difficult to recognize without the proper tools. First, because the rates are nonlinear, the data had to be transformed—in this case, a power transformation on time allowed the linearization of the data. Using the rate of roughness change of the two control groups from the two different experimental runs, the information existed to transform the basis of the first experiment into that of the second (or vice versa) while maintaining the scale of the experimental groups. I was able to recognize the solution to the problem by blending my linear algebra experience with engineering statistics. Linear algebra is not a class typically taken by engineering students but is absolutely vital when transforming data gathered using different experimental methods or comparing data from different experimental runs. This simple problem can become significantly more complicated if a high-order design of experiments is used to investigate multiple phenomena. In these cases, it is vital to understand data transformation to make clear comparisons across the entire DOX (design of experiments). Alternatively, in cases where the data set is larger than the 4 samples I had, it simply becomes more efficient for the engineer to develop an algorithm to handle the data processing.

"In a second example, I had to design a system to compare the amount of surface wear generated on a surface by an abrasive in terms of material lost by mechanical removal. The output data from the measurement tool was simply a three-dimensional map of a surface in an array of x, y, and z coordinates. This is illustrated in Figure 4. I eventually solved the problem by maintaining two control surfaces alongside an abraded region. From these control surfaces, I calculated the volume of the material removed, and the average material removed is reported as a step change down from the control surface. Over the course of 4 months, I had to process nearly 500 samples, each sample taking approximately 10 minutes to process after I developed a computer program to handle the data analysis. Using MATLAB, I wrote my own program to map the bounds of the region of the abraded portion, determine the volume of the material removed in the abraded region, and report the numerical error associated with the numerical integration technique used. An example of the output from this MATLAB program is shown in Figure 5. This programming knowledge alone was not sufficient; the problem also required a rigorous understanding of the experimental setup, the detection resolution of the measuring system, and the ability to program, set up, and write an effective and efficient computer algorithm, as each sample contained nearly 100,000 sets of surface coordinates.

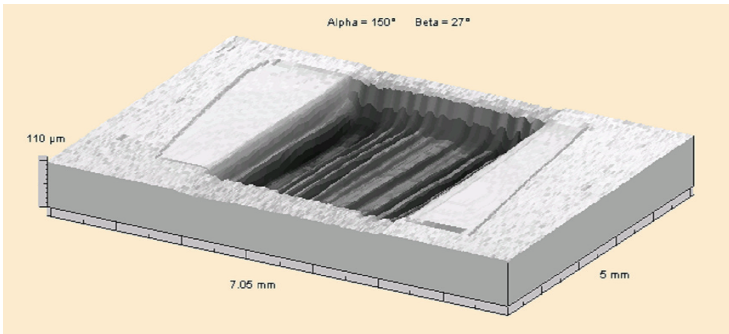

Fig. 4 *Abraded substrate with two control portions to the right and left of the abraded channel.*

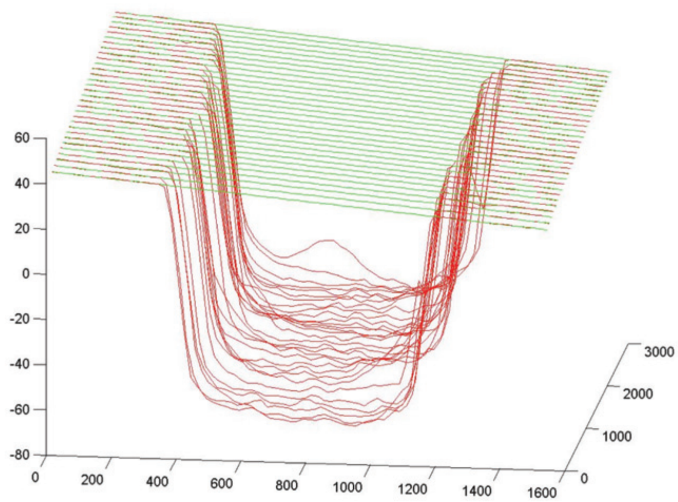

Fig. 5 *MATLAB surface wire frame. The region of integration was determined automatically by the computer algorithm, and the volume was determined by numerical integration between the green surface and the red substrate surface map.*

"The brilliant thing about computing today is that the solutions to complex technical problems are within reach of the undergraduate engineer if that engineer has the proper education to frame the experiment and the understanding of the tools to analyze the data effectively. It is simply inexcusable for an engineer today not to be able to program in MATLAB, Maple, Fortran, or similar languages or tools to solve real problems. The power of computers today is that they afford us the ability to examine complex multivariate experimental designs. These problems often do not have "black box" or "off the shelf" software solutions. The available mathematical software packages must be manipulated. Many companies turn to new engineers who have a fresh working knowledge of the latest computational packages to solve these problems. In a world of shrinking timelines where speed to market is a driving force behind innovation, the engineer must be able to quickly manipulate the tools necessary to solve the problems of the moment while staying fluent in the language of the problems of the future.

"I now attend the University of Cincinnati and am a Ph.D. candidate in the Department of Chemical Engineering. The work I did at P&G blending computation with experimentation really catalyzed my decision to return to graduate school. My thesis focuses on exploring the effects of catalyst size and alloying properties on the kinetics of the oxygen reduction reaction for novel Pt-based alloy shell on noble metal cores. One of the keys to our work is precisely controlling the size of the core that forms the seed for shell reduction and ultimately controls the size of the nanoparticles. Several models for nanoparticle growth exist, each with an emphasis on different fundamental properties of the synthesis. Using UV-Vis spectroscopy, I am able to isolate the behavior of many of the reactant/intermediate species and am able to track them during the course of synthesis as a function of easily accessed parameters like temperature and time. The principles of this work are not that different than those studied in undergraduate chemistry lab; however, in many nonideal problems like the one here, we are faced with data convolution not present in the undergraduate setting. Using computation, I was able to deconvolute the contribution of different species, describe their behavior using pseudo-first-order rate law kinetics expressions, and ultimately describe the evolution of particle size over time. With a robustly designed set of experiments, I have explored the contributions of reactant concentration, time, and temperature on particle size. My best research has always joined computation and experimentation to describe, model, and predict future behavior."

6. Conclusions. In this paper we have described several different models for Computational Science and Engineering education at the undergraduate level. These cover a wide range of possibilities from a small number of courses, through a minor to a fully developed B.S.-level degree. Often local conditions may dictate the level of commitment that individual departments or institutions will be able or willing to adopt.

The reasons for developing such a program are many. The impact is discussed in terms of the pipeline of suitable graduates to fill educational, research, and industrial needs. There are several common components of CSE undergraduate programs, and some guidelines on appropriate content are also provided. One feature that has been strongly encouraged is the inclusion of some internship or similar professional experience.

The varying perspectives of educators, researchers, potential industrial employers, and students are all explored. All in their different ways speak strongly to the need for continued expansion of the interdisciplinary experience that CSE necessarily entails.

7. Appendix. Resources for CSE Undergraduate Education. An extensive and diverse collection of a variety of CSE resources that goes beyond the STEM disciplines is being assembled at the Computational Science Reference Desk—CSERD (http://www.shodor.org/refdesk/). The CSERD, a Pathways project of the National Science Digital Library (http://nsdl.org/) funded by the National Science Foundation, aims to help students learn about CSE and to help teachers incorporate it into the classroom. CSERD attempts to (i) collect a catalog of quality resources from across the internet; (ii) provide a forum for the verification, validation, and accreditation of catalog items both by users and by expert reviewers; and (iii) create original CSE resources for use in education.

Foundations such as Krell [1] and Shodor [5] have been valuable contributors to the development of materials and their dissemination via their own websites and those of other projects. The Krell website is valuable as a source of links to a number of articles and presentations covering curricular materials on computational aspects of atmospheric science and chemistry.

A collection of freely available CSE materials developed through grants from the National Science Foundation (CCLI program) and the W.M. Keck Foundation (Keck Undergraduate Computational Science Education Consortium) is available at http://www.capital.edu/keck-consortium.

One of the noticeable trends is the growing influence of biological science on applied mathematics, and computational science in particular. The report *Math-Bio 2010* [9] is devoted to the growth of mathematical (and especially computational) biology within educational programs.

CSE programs often place significant importance on the use of projects. General publications such as *Computing in Science and Engineering* and *Scientific Computing World* provide a wealth of current research applications of CSE at work in the research environment. *SIAM News* also carries topical stories of new computationally enhanced advances in many areas of science and engineering. SIAM's undergraduate research publication *SIAM Undergraduate Research Online* (SIURO, http://www.siam.org/students/siuro/) was launched in 2008. SIURO provides an excellent outlet for publication of undergraduate research results.

REFERENCES

[1] Computational Science Learning Center, Krell Institute, http://ascr-discovery.science.doe.gov/.

[2] S. C. Glotzer (Chair), S. Kim, P. T. Cummings, A. Deshmukh, M. Head-Gordon, G. Karniadakis, L. Petzold, C. Sagui, and M. Shinozuka, *International Assessment of Research and Development in Simulation-Based Engineering and Science*, WTEC Panel Report, 2009, http://www.wtec.org/sbes/.

[3] D. Keyes (Chair), P. Colella, T. H. Dunning, Jr., and W. D. Gropp, eds., *A Science-Based Case for Large-Scale Simulation*, Vols. 1 and Vol. 2, 2003, http://www.pnl/gov/scales/.

[4] V. Kiernan, *Presidential Panel Recommends Steps to Promote Computational Science*, Chronicle of Higher Education, April 15, 2005.

[5] National Computational Science Institute, http://www.computationalscience.org/.

[6] J. T. Oden (Chair), T. Belytschko, T. J. R. Hughes, C. Johnson, D. Keyes, A. Laub, L. Petzold, D. Srolovitz, and S. Yip, *Revolutionizing Engineering Science Through Simulation: A Report of the National Science Foundation Blue Ribbon Panel on Simulation-Based Engineering Science*, 2006, http://www.nsf.gov/pubs/reports/sbes_final_report.pdf.

[7] SIAM Working Group on CSE Education (Linda Petzold, Chair), *Graduate Education in Computational Science and Engineering*, SIAM Rev., 43 (2001), pp. 163–177.

[8] SIAM Working Group on Undergraduate CSE Education (Peter Turner, Chair), *Undergraduate Computational Science and Engineering Education*, http://www.siam.org/about/pdf/CSE_Report.pdf.

[9] L. A. Steen, ed., *Math & Bio* 2010: *Linking Undergraduate Disciplines*, Mathematical Association of America, 2005.