

The State of Undergraduate Computational Science Programs

Steven I. Gordon
The Ohio State University
Columbus, OH
gordon.1@osu.edu

Katharine Cahill
Ohio Supercomputer Center
Columbus, OH
kcahill@osc.edu

ABSTRACT

A number of efforts have been made to introduce computational science in the undergraduate curriculum. We describe a survey of the undergraduate computational science programs in the U.S. The programs face several challenges including student recruitment and limited faculty participation in the programs. We describe the current state of the programs, discuss the problems they face, and discuss potential short- and long-range strategies that might address those challenges.

Keywords

Computational science and engineering education; modeling and simulation; model curricula; undergraduate computational science education

1. INTRODUCTION

Modeling and simulation has become an integral part in the advancement of knowledge in science and engineering along with theory and experimentation. Computer modeling allows for the exploration of systems that are too complex, too large or sensitive for experiments, or too small to instrument. A majority of large companies in the U.S. use modeling and simulation to produce goods faster and cheaper. The ability to use this technology is essential to commercial competitiveness. Recognition of the importance of computational modeling has led to a widespread call to educate students on the principals of modeling and simulation and the use of computational tools and algorithms to address those modeling needs.

SIAM (The Society for Industrial and Applied Mathematics) formed a working group on computational science in 1998. Subsequently, a seminal article by Yasar and Landau provided one of the first, comprehensive descriptions of the nature of the field and curricular elements necessary to provide students with the appropriate expertise [1]. A further SIAM task force completed a comprehensive report in 2006 [2].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright ©JOCSE, a supported publication of the Shodor Education Foundation Inc.

© 2020 Journal of Computational Science Education
DOI: <https://doi.org/10.22369/issn.2153-4136/11/2/2>

There have also been a number of national science and technology groups that have cited computational science as a key to future discoveries in science and engineering as well as crucial to the competitiveness of US industry.

In 2005, the President's Information Technology Advisory Committee highlighted the importance of computational science to the national economy and cited the lack of qualified personnel to fill the needs of both research and commercial enterprises [3].

The National Science Foundation Blue Ribbon Panel on Simulation-Based Engineering Science indicated that this discipline is "central to advances in biomedicine, nanomanufacturing, homeland security, microelectronics, energy and environmental sciences, advanced materials, and product development." [4] They went on to say that the education of engineers and scientists in the use of simulation techniques is a major challenge.

There have been many efforts to insert computational science into the undergraduate curriculum in an attempt to meet these needs. The National Computational Science Institute (NCSI), developed by the Shodor Education Foundation aimed to develop a national community of faculty interested in incorporating computational science into their undergraduate curriculum [5]. Thomley and Searcy provide a brief overview of this effort along with a comprehensive review of the history of computational science education [6].

Searcy and Thomley [7] completed an evaluation of the Shodor program which points to a number of barriers to the implementation of new academic programs. They surveyed 768 individuals that attended the NCSI workshops. The respondents reported a number of issues with implementing computational science into their courses.

"Between one quarter and one half highlighted the following issues: staying current with changes in technology (49 %), deciding where to make a big shift in their department's curriculum (43 %), lack of available computational science educational materials (39 %), making choices of which technologies/software to use (35 %), having no one else to discuss computational science ideas within their department (32 %), implementing computational science in the face of indifference from other faculty in their department (31 %), trying to incorporate other disciplines' content into a course (31 %), lack of understanding of how software package(s) work (28 %), trying to coordinate content across multiple professors and/or multiple sections of a course (26 %), and trying to coordinate programmatic changes across departments (26 %)." [7 page 3].

Other efforts at integrating computational science in the curriculum have focused on the introduction of formal emphasis or minor programs that include four to six courses focused on the tools and techniques used in the field. Gordon, Carey, and Vakalis [8] review some of these efforts. They then summarize their efforts to start computational science programs at multiple institutions in Ohio. Those efforts included the creation of a set of competencies for undergraduate students in computational science. Those competencies have been updated and are part of the efforts of the XSEDE education program to help other institutions start computational science programs [9].

A number of grants by the National Science Foundation and other institutions have supported efforts such as those cited above. Yet, the number of formal undergraduate computational science programs has grown very slowly. A recent web based search for such programs yielded a list of only 29 programs in the U.S. [10]. The program links on that site were checked and a further search conducted by the authors to ensure that the list is up-to-date.

There are many questions related to the state of these programs. How have the programs fared in producing graduates with computational science knowledge and skills? Are there continuing barriers to the integration of these important skills in the undergraduate curriculum? What are the institutional, personnel, and resource issues that have contributed to the success and/or limitations on the programs? What sorts of institutional or environmental changes that might help to scale up the programs? In order to address these questions, we conducted a survey of the existing programs focusing on the current state of their efforts as well as continuing barriers to program implementation. That is the subject of this paper.

2. THE SURVEY

For each of the undergraduate programs in the updated web list, contact information for the program advisors was assembled either as indicated on a program webpage or via a phone call to the appropriate person. For the 29 programs at 26 institutions, a survey of 11 questions addressing some of the motivations and barriers to program implementation was assembled and sent via an email link to the program lead. Of those, ten programs responded.

The survey questions were based on the previous work by Thomley and Searcy as well as informal discussions with program coordinators as part of the work on the XSEDE education program. The survey was distributed via email using the Qualtrics survey tools. A copy of the basic survey can be found in Appendix 1. Email reminders were sent to non-respondents weekly over a one month period.

Follow-up in-depth interviews were also made with three of the respondents to gain additional insight into the state of their programs. Two of the programs selected for interviews were of long-standing and were selected to provide insights into the barriers to program implementation as well as possible changes that would enhance program success. The third is a program that was recently started, hoping to gain insights into continuing motivations and barriers to program initialization. Those interviews were conducted by telephone and guided by a series of the follow-up questions. Those questions are also shown in the appendix. Those interviews were more open-ended, asking for the broader opinions of the program directors.

3. SURVEY RESULTS

3.1 Nature of the Programs

Of the 29 programs, 10 (34%) responded to the survey. The responding programs surveyed varied in their composition (4 department level programs, 3 college-wide, and 2 university-level programs) and all have been active for more than 5 years.

Table 1 shows the distribution of respondents and non-respondents. Those responding appear a reasonable representation of the population across types of institutions, public or private, and Carnegie classification.

Table 1. Comparison of Respondents and Non-Respondents

	Carnegie Classification				
	Public	Private	R1/R2	M1/M2	Other
Responded	6	4	5	3	2
No Response	8	9	8	5	4

All of the responding programs are modest in size. The average number of students completing the program annually ranged from 1 (two programs) to 3-5 (four programs) to 15 (two programs). Administratively, two of the programs are university-wide programs. Three of the programs are college-wide while three others are departmentally based.

Most of the programs are marketed through announcements in basic, required courses. One depends entirely on a website for recruitment.

Respondents were asked what percentage of their students went on to graduate school or professional jobs. Several did not answer this question or answered inconsistently. Of those who responded, there was a range of 33 to 70 percent that go on to graduate school or an average of 41 percent. Similarly, an average of 44% go to professional jobs with a similar range. However, the inconsistency in the responses leads one to believe that there the institutions are not fully able to track what happens to their graduates.

3.2 Program Challenges

A series of questions focused on some of the program challenges that were cited in the study of the Shodor program. These are shown in Table 2. The first two questions focused on student recruitment. Here, the majority of respondents indicated that student recruitment is a major problem. Seventy-one percent of respondents strongly agree or somewhat agree that it is difficult to recruit students into their programs. Likewise, the same percentage somewhat disagree or strongly disagree that that they have little or no difficulty getting students into their program.

A second problem facing the programs is the burden of instruction. Here again, 71% of the respondents indicated that the burden for teaching courses falls to too few faculty.

Most programs found their students were prepared in math (78% strongly or somewhat agree) but less in programming (44% strongly or somewhat agree).

People were then asked to indicate the top three problems impacting their programs. Almost 30% indicated that recruiting students was a major problem. Next was engaging faculty in other departments to participate (22%), distribution of teaching loads (17%), and getting advisors to recommend the program to students early in their careers (17%).

Table 2. Responses to Program Challenges Survey Questions

	Strongly Agree	Somewhat Agree	Neither Agree Nor Disagree	Somewhat Disagree	Strongly Disagree
It is difficult to recruit students to enroll in our program.	4	2	2	0	1
We have little or no difficulty getting a sufficient number of students in our program courses.	1	1	0	4	3
The burden for teaching the courses at our institution falls to only a few faculty.	3	2	1	2	1
Students enrolling in our program have the required pre-requisite skills and knowledge in mathematics.	2	5	0	2	0
Students enrolling in our program have the required pre-requisite computing skills.	1	3	0	4	1

Follow-up telephone interviews were conducted with three programs. Two were programs of long standing while the third has just started their program. All three indicated that student recruitment was a major problem.

The telephone interviews focused on possible policies that might alleviate some of the barriers to program success. One suggestion that was discussed was a possible national effort to publicize the need for computational scientists like that given to computer science. The response was that this might be helpful, but it still may not address the overall problem faced while trying to recruit students: what is the job or career path that this program will prepare me for?

Another possible boon to student recruitment would be a university requirement that all science and engineering majors take an introductory modeling and simulation course. This, along with the need for an introductory programming course for non-computer science majors, would potentially increase student interest in computational science.

Addressing the issue of “what is the job,” the respondents were asked whether stronger ties with businesses that use computational science would assist in improving program numbers. Such ties would be welcomed but the one respondent to this question indicated that they did not have enough contacts or time to make those connections.

All of the telephone interviews echoed the problems of recruiting students. Indications were that some students still take one or two of the core computational science courses but do not complete the entire program.

4. DISCUSSION

The response rate to the survey of 34% was disappointing, yet it is a decent response rate to an online survey. Nevertheless, we believe that those long-standing programs that did respond are emblematic of the problems facing computational science education. We can only surmise whether the non-respondents represent programs that are inactive, are run by faculty with too little time to make responding a priority, or some combination of these and other factors.

Based on the responses, it is clear that current, undergraduate computational science education efforts are making only modest progress in helping to build a workforce competent in this area. They continue to face ongoing problems in student recruitment, in the engagement of the full range of disciplines for which computational science is important, and the active engagement of

businesses that likewise are seeking graduates ready to contribute to their computational science endeavors.

Student recruitment efforts are foremost among the problems facing the existing programs. There appear to be a number of reasons for this. First, the interdisciplinary nature of the field makes it difficult to point to particular career paths associated with computational science. That confusion may also carry forward to academic advisors that may not fully understand what computational science is and therefore do not advise students to look into those programs early in their academic careers. Recruiting of students is also hampered by the fact that most of the programs are minors that require an additional 15-24 credit hours of additional courses. Students may take part of the sequence but do not complete the program. Should they start the program later in their academic career, they may not have enough time to complete all of the courses. At a time when the costs of higher education are high, delays in graduation pose a significant barrier for students to take on supplementary course work for a career path that is fuzzy at best.

Computational science continues to suffer from the limitations on the number of faculty that are prepared to participate in the programs. A combination of lack of expertise along with the required teaching loads for more traditional courses is probably to blame.

There are no simple solutions to these program impediments. One possible approach might be to introduce a university-wide course that introduces modeling and simulation to all students. This could be done without requiring pre-requisite programming expertise. For example, the University of California at Berkeley has developed a “data science for all” course for freshman students (11). A modeling and simulation course could be one of several alternatives in this vein, introducing students to the area early in their careers and promoting their continued interests.

If modeling and simulation expertise is truly required in the current workforce, then both business and the research community must also play a larger role in supporting program development. Businesses will need to more actively engage with academic institutions providing internships and help with student recruitment by publicizing the need for modeling and simulation skills. Grant programs that encourage the integration of computational science into the curriculum should be put in place in parallel with the efforts for computer science and data science.

Getting widespread participation for computational science across the faculty is probably the most difficult challenge. It may take a generational change in the faculty to fully address the problem. A much larger proportion of recent Ph.D. graduates in science and engineering are using modeling and simulation as part of their research and thus more likely to embrace the integration of those skills in the curriculum. Currently the science and engineering labor force is aging with 33% of the workforce in the ages between 51 and 75 years while only 16% was in the under the 30 age group (12). This implies a high rate of retirement and replacement in the coming years. Perhaps that will help the community to fully embrace the need for computational science expertise.

5. REFERENCES

- [1] Yasar, O. and Landau, R. H. (2003). Elements of computational science and engineering education. *Review*, 45(4), 787-805.
- [2] SIAM Working Group on CSE Undergraduate Education. (2006). Undergraduate Computational Science and Engineering Education. https://www.siam.org/Portals/0/Publications/Reports/CSE_Report.pdf?ver=2018-03-16-161618-620. Accessed November 19, 2018.
- [3] Reed, D., Bajcsy, R., Fernandez, M., Griffiths, J., Mott, R., Dongarra, J., Johnson, C., Inouye, A., Miner, W., Matzke, M., and Ponick, T. (2005). Computational Science: Ensuring America's Competitiveness.: The President's Information Technology Advisory Committee, National Coordination Office for Information Technology
- [4] SBES 2006. Simulation Based-Engineering Science: Report of the NSF Blue Ribbon Panel on Simulation-Based Engineering Science, NSF, February 2006. Available online at https://www.nsf.gov/pubs/reports/sbes_final_report.pdf
- [5] National Computational Science Institute. (n.d.). About NCSI. Shodor/NCSI web site: <http://www.computationalscience.org/about>. Accessed November 19, 2018.
- [6] Jill Tomley and Mary Searcy, (2009). Computational Science: Not Just for Researchers Anymore. *The International Journal of Science in Society*, 1(3), 27-42.
- [7] Searcy, ME and Thomley, JE. (2007). Barriers to interdisciplinary computational innovations in education. In Milková, E, and Prázák, P (Eds.), *Proceedings of the 8th International Conference on Technology in Mathematics Teaching*. Prague, Czech Republic: Faculty of Informatics and Management, University of Hradec Králové.
- [8] Gordon, S., Carey, K., and Vakalis, (2008). A shared, interinstitutional undergraduate minor program in computational science. *Computing in Science and Engineering*, 10(5), 12-16.
- [9] Minor Program in Computational Science ompetency/Topic Overview. <http://hpcuniversity.org/educators/undergradCompetencies/> Accessed May 28, 2019.
- [10] Undergraduate Programs in Computational Science. <http://hpcuniversity.org/students/undergraduatePrograms/>. Accessed on May 28, 2019.
- [11] Foundations of Data Science: A Data Science Course for Everyone <https://data.berkeley.edu/education/courses/data-8>. Accessed on May 28, 2019.
- [12] National Science Board. Science and Engineering Indicators 2018 (NSB-2018-1). January 2018. <https://www.nsf.gov/statistics/2018/nsb20181/figures/fig03-22>. Accessed on May 28, 2019.

A. SURVEY INSTRUMENTS

A.1 Undergraduate Computational Science Programs

Q1 Is your undergraduate computational science program still active?

- Yes (1)
- No (2)

Q2 How many credit hours are required for students to complete the program?

Q3 What is the average number of students that complete the program each year?

Q4 For each of the following questions, please indicate the degree to which you agree or disagree with the statement.

(Strongly agree, Somewhat agree, Neither agree nor disagree, Somewhat disagree, Strongly disagree)

- It is difficult to recruit students to enroll in our program. (1)
- We have little or no difficulty getting a sufficient number of students in our program courses. (2)
- The burden for teaching the courses at our institution falls to only a few faculty. (3)
- Students enrolling in our program have the required prerequisite skills and knowledge in mathematics. (4)
- Students enrolling in our program have the required prerequisite computing skills. (5)

Q5 What proportion of students graduating from your program go to:

- Graduate school (1)
- Professional Jobs (2)
- Other (3)
- Not sure (4)

Q6 What do you think are the major problems associated with maintaining your program?

Q7 Administratively, where is your program located?

- University-wide program (1)
- College-wide program (2)
- Departmental program (3)

Q8 Please list all of the departments that play a role in teaching courses in your program.

Q9 How many years has your program been operating?

Q10 How do students find out about your program?

- Announced in various basic courses in related disciplines (1)
- Website (2)
- Program brochures (3)
- Listed in university catalog (4)
- Other (5)

Q11 Please choose what you see as the top three problems with your undergraduate computational science program.

- Difficulty recruiting students (1)
- Limited dedicated resources for program implementation (2)
- Distribution of teaching loads (3)
- Availability of relevant hardware and software (4)
- Getting advisors to recommend the program to their students early in their careers (5)
- Engaging faculty in other departments to participate (6)
- Other (7)

A.2 Follow Up Questions for Respondents

We would like to thank you for responding to our survey about your computational science program. We would like to take a few minutes to follow-up with you on some questions that arose from the survey. Do you have a few minutes to speak now or can we set a time that is more convenient for you?

A number of problems associated with maintaining a program were cited by those completing the survey. We would like to get your thoughts on these problems and actions that might help to reduce them.

The first problem is cited is the difficulty of recruiting students. Is this a major problem for your program?

Which of these actions might help to alleviate that problem:

- National attention given to the need for scientists and engineers to understand modeling and simulation similar to that given recently to computer science?
- University requirement for introductory modeling and simulation class for all science majors
- An introductory computer coding class oriented for non-computer science majors
- Working with high schools to bring computational science into HS courses
- Other?

A second problem noted is the lack of resources from the university to offer courses and related limitations on the number of a faculty who can offer courses. What do you see as possible solutions to these problems?

Which of these might help:

- On-going funds the college or university to “buy” courses from other faculty to release them to teach a computational science course
- Funding for adjunct (is this the right word) faculty from industry to teach or co-teach some of the courses
- Sharing course instruction with other institutions using distance learning infrastructure

Several programs require an internship or research experience as part of their programs. Does your program have such a requirement? What do you see as problems managing this program? How might these problems be overcome?

- Deeper connections with businesses that use computational science to employ your students as interns
- A central database of internship opportunities at a national scale University program for undergraduate research opportunities