Computational Science Program Self-Assessment Report
July, 2011

0. Background
The Director of the Computational Science (COSC) program, Eric Stahlberg, left Wittenberg in late 2010. Since then, Interim Assistant Provost Elizabeth George and a reconstituted Computational Science Advisory Committee (made up of representatives from the participating departments) have been overseeing the program. The following departments currently participate in the Computational Science minor: Biology, Chemistry, Economics, Geology, Mathematics and Computer Science, Physics, and Psychology.

The requirements for the COSC minor are given in the Appendix at the end of this report.

The last Computational Science self-assessment was part of the Math/Computer Science/Statistics/Computational Science departmental self-assessment completed in 2009. This 2011 Computational Science self-assessment was prepared by Elizabeth George.

1. Learning Goals for the Computational Science Minor

Computational science minors should:

a. Treat computational science as a new paradigm for understanding the natural world.

b. Use observation, modeling, computational software (numeric and symbolic), visualization tools, and analysis to improve this understanding.

c. Learn and apply the classes of mathematical computer models (continuous vs. discrete, static vs. dynamic, and deterministic vs. stochastic) toward the solution of computationally intensive science problems.

d. Become familiar with the use of approximation techniques, simulation, and optimization methodologies, as well as the limitations of these approaches.

e. Gain experience in laboratory work and apply the scientific method in their studies and research.

f. Be able to communicate effectively through writing and speaking and to articulate their ideas both in technical and in non-technical form.

They should also be able to:

g. Understand the fundamentals of algorithm design and the implementation of algorithms through the use of high-level programming languages and tools.

h. Understand the fundamentals of discrete and continuous mathematics, including differential and some integral calculus, and be able to apply these fundamentals in problem solving situations.

2. Assessment Methods

Learning goals for computational science minors are assessed in the specific courses or other requirements that pertain to the goals. Listed below for each learning goal are the required courses in which that goal is addressed and assessed (via assignments and exams).
a. Treat computational science as a new paradigm for understanding the natural world.
*MATH/COMP 260 and the required Elective courses in the major department; required COSC capstone project*

b. Use observation, modeling, computational software (numeric and symbolic), visualization tools, and analysis to improve this understanding.
*MATH/COMP 260 and the required Elective courses in the major department; required COSC capstone project*

c. Learn and apply the classes of mathematical computer models (continuous vs. discrete, static vs. dynamic, and deterministic vs. stochastic) toward the solution of computationally intensive science problems.
*MATH/COMP 260*

d. Become familiar with the use of approximation techniques, simulation, and optimization methodologies, as well as the limitations of these approaches.
*MATH/COMP 260*

e. Gain experience in laboratory work and apply the scientific method in their studies and research.
*COSC minors are required to take two laboratory courses that meet the Natural World goal.*

f. Be able to communicate effectively through writing and speaking and to articulate their ideas both in technical and in non-technical form.
*COSC capstone project*

They should also be able to:

g. Understand the fundamentals of algorithm design and the implementation of algorithms through the use of high-level programming languages and tools.
*COMP 150Q (Introduction to Computer Science) and MATH/COMP 260*

h. Understand the fundamentals of discrete and continuous mathematics, including differential and some integral calculus, and be able to apply these fundamentals in problem solving situations.
*MATH 201Q or MATH 131Q (Calculus) and MATH/COMP 260*

3. Summary and interpretation of findings
Because most of the learning goals are linked to specific requirements for the minor, students who successfully complete the requirements for the Computational Science minor have met the learning goals at least to some extent. Goal f, however, currently is only partially addressed in the COSC minor requirements. The capstone requirement includes a formal presentation, “either written, oral, or both,” and although many COSC minors have done both written and oral presentations of their capstone work, there is no explicit requirement for them to do so, and there is currently no formal means of evaluating their ability to communicate ideas in both a technical and non-technical form.

4. Changes to be implemented
The last self-assessment report of the COSC program stated, “Although no changes were made to departmental minors in computational science given that the program is new and has had a chance to mature, consider revisiting the learning goals to see if they are still appropriate.” This will be done now that the advisory committee has been reconstituted. As part of that process, we plan to work with the faculty who have been and will be teaching MATH/COMP 260 (Brian Shelburne and the new Applied
Math hire) to make sure the learning goals that pertain to MATH/COMP 260, and the assessment methods used in that course, are appropriate. The committee will also work with the participating departments to make sure the elective courses in the major departments are still appropriate for the minor and that the assessment methods used in those courses are also appropriate.

The other suggestion from the last self-assessment report is “Since the computational science minor has a capstone requirement, consider including a learning-goals-based assessment element to the computational science capstone.” Currently capstone presentations are to be evaluated by the Director, Chairperson of the major department, and supervising faculty member, though there are no specific criteria for evaluation. The advisory committee will also look at the question of what should be required for the capstone, how capstone projects should be evaluated, and whether additional assessments of the capstone are needed. Finally, we would like to discuss the possible implementation of additional methods of assessment of the minor (other than those associated with the required courses and the capstone).

5. Resource needs
As is evident from the listing of learning goals, the MATH/COMP 260 course is an essential component of the minor. Thus, a major resource need is the continuing need for a qualified faculty member to teach MATH/COMP 260. The new visiting position in Applied Math will fill this need for the time being.

Supervision and evaluation of the required capstone projects, as with all independent studies and student research, does not count as part of the teaching load. If the number of minors continues to grow, the need to supervise capstones will require a large time commitment by the Director of the program and the participating faculty members in the various departments.

Finally, as is also true for many other programs and departments, Computational Science would benefit from an improved web (and other media) presence and assistance with setting up internships. Those tasks were a major focus of Eric Stahlberg before his departure.

6. Plan for continued assessment
Because the capstone is intended to integrate many of the learning goals, systematizing and keeping records of the capstone evaluations will be a major step in ensuring continuing assessment of the program. We will investigate additional ways to collect assessment data on the learning goals, perhaps through a survey administered to graduating seniors or through common exam questions.

Appendix. Computational Science minor requirements

- Computer Science 150Q. Introduction to Programming. 5 semester hours.
- One of the following courses:
  - Mathematics 201Q. Calculus I. 4 semester hours.
  - Mathematics 131Q. Essentials of Calculus I. 4 semester hours.
- Mathematics/Computer Science 260. Computational Models and Methods. 5 semester hours.
- Capstone Experience (0-4 semester hours)
- Elective Courses (8 Semester hours from any two or more of the courses listed below)
  - Biology
    - 316. Molecular Genetics and Bioinformatics. 5 semester hours.
    - 341. Limnology. 5 semester hours.
    - 342. Stream Ecology. 5 semester hours.
    - 346. Ecology. 5 semester hours.
    - 347. Evolution. 4 semester hours.
Chemistry
311. Physical Chemistry I. 5 semester hours.
321. Inorganic Chemistry. 5 semester hours.
352. Physical Chemistry II. 5 semester hours.
372. Biochemistry II. 5 semester hours.

Computer Science
Computer Science/Mathematics 320. Numerical Analysis. 4 semester hours.
Computer Science 350. Artificial Intelligence 4 semester hours.
Computer Science/Mathematics 380. Optimization

Economics
300. Econometrics. 4 semester hours.
370. Mathematics for Economists. 4 semester hours.

Geology
240. Process Geomorphology. 5 semester hours.
260. Sedimentology. 5 semester hours.
291. Spatial Analysis in the Natural Sciences. 2 semester hours.

Mathematics
Mathematics 205. Applied Matrix Algebra. 4 semester hours.
Mathematics 215. Differential Equations. 4 semester hours.
Mathematics 227. Data Analysis. 4 semester hours.

Physics
311. Classical Mechanics. 4 semester hours.
320. Computational Physics. 2 semester hours.
321. Signal Processing. 2 semester hours.
332. Electromagnetism. 4 semester hours.
410. Mathematical Physics. 4 semester hours.
411. Quantum Mechanics. 4 semester hours.

Psychology
207. Experimental Design. 5 semester hours.
211. Sensation and Perception. 4 semester hours.
311. Behavioral Neuroscience. 5 semester hours.
321. Learning, Memory and Cognition. 5 semester hours.
341. Psychological Testing. 5 semester hours.
361. Experimental Social Psychology. 5 semester hours.