Overview of Computational Science and Engineering Education Programs
Steve Gordon (sgordon@osc.edu)
The Need For Computational Scientists

- A number of national studies document the need for computational scientists
  - "...computer modeling and simulation are the key elements for achieving progress in engineering and science." NSF Blue Ribbon Panel on Simulation-Based Engineering Science
  - "A persistent pattern of subcritical funding overall for SBE&S threatens U.S. leadership and continued needed advances..." International Assessment Of Research And Development In Simulation-Based Engineering And Science
  - Nearly 100% of the respondents indicated that HPC tools are indispensable, stating that they would not exist as a viable business without them or that they simply could not compete effectively. IDC Study for Council on Competitiveness of Chief Technology Officers of 33 Major Industrial Firms
Examples of Modeling Problems

• Tracing the spread and evolution of disease (http://supramap.osu.edu/)
• Collaborations to explore historical and contemporary events and social interaction (http://www.ichass.illinois.edu/Projects/Projects.html)
• Predicting the impacts of earthquakes (http://nees.org/)
• Designing and testing new nanomaterials and devices (http://nanohub.org/)
• Discovering oil reserves (http://access.ncsa.illinois.edu/Stories/oil/)
• Designing new packaging (http://phx.corporate-ir.net/phoenix.zhtml?c=104574&p=irol-newsArticle&ID=651774&highlight)
• Discovering how the brain works (http://www.compete.org/publications/detail/503/breakthroughs-in-brain-research-with-high-performance-computing/)
XSEDE Education Program Goals

• Prepare the current and next generation of researchers, educators and practitioners.
• Create a significantly larger and more diverse workforce in STEM.
• Inculcate the use of digital services as part of their routine practice for advancing scientific discovery.
Creating Computational Science Programs

• Inherently interdisciplinary
  – Science, engineering, or social science domain
  – Mathematics
  – Computer science

• Expertise often dispersed across multiple departments, colleges, institutions

• Difficulty of negotiating requirements, responsibilities, and institutional arrangements
Providing a Curriculum Model

• Based on our experience in Ohio creating an interdisciplinary, inter-institutional minor program in computational science
• Effort supported by an NSF grant
• Allowed us to work through many of the issues associated with creating an interdisciplinary program
• Demonstrated the feasibility of an interdisciplinary, inter-institutional program
• See
  http://hpcuniversity.org/educators/competencies/
Program Requirements

• Created a competency-based curriculum
  – Provides detailed outlines of the background and skills desired for students completing the program
  – Bridged the differences across disciplines
  – Allows for flexibility in implementation to fit the program into multiple institutional situations and majors with different backgrounds and focus areas

• Serves wide range of needs
  – Provides essential skills for all students regardless of whether they complete the minor
Undergraduate minor program overview

- Undergraduate minor program
  - 4-6 courses
  - For majors in variety of fields
- Faculty defined competencies for all students
- Reviewed by business advisory committee
- Currently being updated to reflect changes in hardware and software technologies
- Requirements adjusted to reflect the needs of different majors

<table>
<thead>
<tr>
<th>Competencies for Undergraduate Minor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation and Modeling</td>
</tr>
<tr>
<td>Programming and Algorithms</td>
</tr>
<tr>
<td>Differential Equations and Discrete Dynamical Systems</td>
</tr>
<tr>
<td>Numerical Methods</td>
</tr>
<tr>
<td>Optimization</td>
</tr>
<tr>
<td>Parallel Programming</td>
</tr>
<tr>
<td>Scientific Visualization</td>
</tr>
<tr>
<td>One discipline specific course</td>
</tr>
<tr>
<td>Capstone Research/Internship Experience</td>
</tr>
</tbody>
</table>
Flexibility in Implementation

• Adapt existing courses by adding computationally oriented modules
• Discipline oriented courses dependent on existing faculty expertise and interests
• Different subsets of required and optional competencies tied to major, required math, and example projects
Community College Program

• Created competencies for science majors in two-year schools along with a concentration in computational science
  – http://www.rrscs.org/associate
  – Program at Stark State College in Ohio
    • http://www.starkstate.edu/academic-programs/computational-science
Benefits to Students

• Inquiry-based learning is more effective than traditional lecture oriented instruction
  – Students are actively engaged in the learning process
  – Students gain deeper insights and have higher retention rates for the information
  – Facilitates the integration of information across academic disciplines – math, science, engineering, computer science
Two Major Approaches

• Update existing courses with computational modeling examples
  – Demonstrations during lectures
  – Exercises for students to complete

• Create formal minors or concentrations that provide deeper expertise in selected areas

• Approaches are not mutually exclusive
Getting Started – Existing Courses

• Review current course objectives and identify concepts and applications with possible modeling examples
• Incrementally substitute modeling examples for more traditional evaluations
• Coordinate changes across classes and disciplines to provide reinforcement and avoid duplication of effort
• Revise as needed over time
Identifying Additional Needs

• What kinds of resources and assistance do you need to meet these goals?
  – Access to example materials
  – Workshops on special topics for faculty
  – Organizational assistance to review the curriculum
  – Other?
Next Steps

• Create working committees to review courses and materials
• Identify candidates and needs for workshops
• Discuss formal program creation options
Discussion Program Plan for Doane