SECANT- One year later

Susanne Hambrusch, Tim Korb, Chris Hoffmann, Mark Haugan
A first computing course for science majors

- Purdue science majors have a computing requirement
- Currently they can choose among four CS courses
- A first course should
  - Teach computational thinking and problem solving skills
  - Increase the interest in computing
  - Promote computational scientific inquiry
Needs from science departments (1)

Physics
- Currently using VPython in an introductory physics course
- Faculty are interested in students gaining a more complete understanding of computation

Chemistry
- Learn computational methods relevant in chemical research, in particular MC, SA
- Be able to use and run existing programs (e.g., written in Fortran)
Needs from science departments (2)

Bioinformatics and Statistics

- Teach R for statistical computing and visualization
- Use a language for which bioinformatics software packages exist or can be integrated
CS 190C: Introduction to Computational Thinking

- Lay the groundwork for computational thinking
- Programming language should allow a focus on computational principles
- Present examples in a language familiar to the students
- Teach in a problem-driven way
- Make effective use of visualization
Lecture Material

- 6 weeks on programming basics in Python
- 6 weeks on computational tools and methods
  - Numerical issues
  - Introduction to simulation and Monte Carlo methods
  - Guest lectures in Physics and Bioinformatics
  - Scientific visualization
  - Grand challenges
- 3 weeks on important CS material
  - Object-oriented design
  - History of computing
  - Limits of computation, future models

Format: 2 lectures per week, one 2-hour lab
Software used

- Python
- VPython
- NumPy
- PyLab from MatPlotLib
- NetworkX
- Cytoscape
Course Projects

- Four projects
  - Manipulating Digital Audio
  - Percolation in Grids
  - Simulating Physical Systems
  - Analyzing Protein-Protein Interactions

- Each project had a
  - programming component
  - experimental component (with a write-up)
  - [http://secant.cs.purdue.edu/cs190c:projects](http://secant.cs.purdue.edu/cs190c:projects)
Digital Audio

- Explore the generation and manipulation of digital sound
- Students implemented several basic functions explored new operations
- Array operations, loop structures, numerical issues, and listen to sound and visualize waves
- **Feedback**
  - Challenging for some to get started
  - Enjoyed the flexibility and creativity the project allowed
  - Most recognized later that it was an easy project and a great learning experience
Percolation on Grids

- Modeled after the chapter on percolation in the Sedgewick-Wayne text
- Two-dimensional structures, recursion, random number generation
- VPython visualization and plotting graphs during the computation

Feedback
- For many, this was the favorite project
- Nice combination of visualizing progress during the computation and generating plots visualizing the results
Simulating Physical Systems

- Use the Monte Carlo “demon algorithm” to estimate parameters in physical system
  - an ideal gas with moving molecules
  - Ising Spin model
- Adapt a generic Demon Algorithm specification to each simulation, run fairly large the simulations, generate various plots and histograms from generated data
- **Feedback**
  - Seemed natural material for physics and chemistry majors
  - Physics lectures were crucial for understanding and motivation
Analyzing Protein-Protein Interaction

- Use data sets from large-scale experiments characterizing protein-protein interactions in *S. cerevisiae* (baker’s yeast) to generate graphs
- Apply graph operations using NetworkX and a provided clustering routine (MCL) to “clean” graph and determine clusters
- Graph visualization using Cytoscape
- Evaluate the quality of the clustering by using biological information obtained from the Gene Ontology (GO)
Protein-Protein Interaction: Feedback

- Students found abstraction using graphs challenging
- There were no biology students in the class to “spread” excitement
- Graph visualization provided NetworkX does not scale; students switched to Cytoscape too late
- Project would be easier with more class preparation
- Most research-oriented project
Teaching CS: Language Issues

- Python strengths…
  - Low-syntactic overhead
  - Rich primitives and built-ins
  - Straightforward 3D visualization (with VPython)

- Python weaknesses…
  - Successful transition to C/C++ and Java unclear
  - Larger programs can be more difficult
    - No compile-time type checking
    - Testing crucial for, e.g., library use
Teaching CS: Concept Issues

- Delayed “programming in the large” concepts
- Focused on techniques needed to solve problems
  - Many declarations not needed (or not available)
  - Used built-in and library functions where possible
  - Introduced abstractions (declarations, functions, classes) when motivated
- Concepts needed to understand what’s going on…
  - Audio sampling -> binary representation
  - Performance -> cost of some primitives (e.g., max(array))
  - Parameter passing and side effects -> the heap
  - Recursion -> the runtime stack
Visualization

- Visualization is important
  - For learning how to solve problems
  - For understanding what is computed
  - For finding bugs

- 2D graphing is fine, but 3D is better –
  - Especially when you can interact immediately

- Bad random number example:
  \[ r_{k+1} = (65539 \times r_k + 12345) \mod 32767 \]
Some Difficulties

- Information visualization tools are in infancy:
  - Graph layouts, data structures, etc require considerable programming effort and are cumbersome to use

- For 3D, viewing interactions are challenging:
  - Managing camera, frustum and illumination can be tricky
  - Direct manipulation interfaces an issue

- Scaling issues:
  - Viewing large data, viewing at high resolution
Impact on Science Disciplines: Physics

Impact on students

- Physics students gain
  - a healthy new perspective on the nature of and interplay between physics and (applied) mathematics by solving realistic problems computationally
  - computational knowledge and skills that create new opportunities for learning and research in our department

- Challenges
  - overcoming a resistance to computing evinced by some of our students so that more of them realize the gains
Impact on Science Disciplines: Physics

Impact on faculty and our science

- Physics faculty gain
  - students with new resources for learning in the classroom and the lab
  - students prepared for undergraduate research in the expanding field of computational physics

- Challenges
  - increasing faculty awareness of the resources above, especially as it relates to revising classroom and lab instruction to take advantage of them [NB. Phys 344]
Feedback from the students

- 15 students, 2/3 were freshmen
- Physics and Chemistry majors, some Math double majors
- For the freshmen, the course satisfied the College of Science computing requirement
- About ¾ had some prior programming experience
- High-ability students
  - Clearly among the 30% of the freshmen completing a degree in science
Entry and exit survey

Two questions from the survey

How would you rate your current interest in:

**Taking another computer science course?**

**Pursuing a career that requires programming skills?**

- 0: not interested, 1: somewhat uninterested
- 2: undecided, 3: interested, 4: very interested
Interest in taking another CS course

Interest in a career that requires programming
Increasing interest in computing

- In our usual introductory programming courses, interest in computing decreases; echoed in other studies
- In 190C, interest increased
  - Half the students plan to minor in CS
  - Students realize that computational skills make them more competitive for internships
  - 2 students are currently REUs in CS
  - One student is thinking about becoming a CS major
The problem-driven format and quickly writing meaningful programs increases student interest.

Python is an excellent language vehicle.

Visualization is an important component and brings many quantitative scientific facts to life.

Interaction with science faculty is critical to designing an effective course.

Getting institutional buy-in should not be underestimated.
Challenges and Future Plans (1)

- **What comes after the first course?**
  - Planning a second course in which CS and science students collaborate in teams
  - Provide REU opportunities
  - Get Physics students involved in the use of Python in the Physics courses

- **Should different versions of the course be taught to different majors?**
  - We believe that the same first course can serve all science disciplines well
  - Motivation of domain specific projects needs consideration
Challenges and Future Plans (2)

- Make the first course a regular course
- Work with science departments
  - understand how the material relates to other courses
  - Give advice on effectively integrating computation into their later courses
Course Projects under consideration

- Lennard-Jones simulation using Simulated Annealing
- Using Scribbles Robots (obstacle avoidance, driving on roads)
- Spread of disease simulation
- Better visualization for large graphs