Computing Courses for the Sciences: A Small School Perspective

John M. Zelle, Ph.D.

Wartburg College
Background

• Wartburg College: Small (~1800) Liberal Arts College in Waverly, Iowa

• Strong Science Programs
  – Biology
  – Chemistry and Engineering Science
  – Mathematics, Computer Science, and Physics
Motivation

• Increasing Importance of Interdisciplinary Connections
• Expand on Existing Strength of Combined Department
• Effort to Attract Quality Students to CS Classes
Obstacles

• Small College Environment
  – Can't support specialized computing classes for various sciences
  – Little computing expertise outside of CS
  – Must have relatively flat prerequisite structure

• Must “Grow” a program organically
  – No new resources likely
Incremental Steps

- Science Friendly CS 1
- CS Senior Project Collaboration
  - Stereographic Display, Biological Visualizations, Rat Tracking
- Introduction to Computational Science
- Computing in Physics Curriculum
- Revision of Calculus Sequence
Friendly CS 1

• Using Python as a First Language
  – Very-High level (executable pseudocode)
  – Cover more material with better practical understanding.
  – Popular as a scientific scripting language

• Include Scientific Applications:
  – chaos, graphics/visualizations, simulations, cellular automata.
Computational Science

• CS Elective Covering
  – Modeling (differential equations)
  – Numerical Methods
  – Visualization
  – Parallel Programming

• Issues:
  – Diverse Majors (CS, Math, Physics, Bio, Chem)
  – Few Prerequisites
Solution: Project Orientation

• Projects:
  – Non parallel: Lotka-Volterra, Lorenz Attractor, Visualization of Cygnus A,
  – Parallel: Newton's Basins of Attraction, LaPlace heat flow, Galaxy Collisions

• Progression
  – Mathematical Model, Computational Methods, Visualization, //ism
Project Orientation (2)

• Each Project Includes Multiple Levels
  – Phase I: doable (required) for all students
  – Advanced Phases: optional for those students with interest/background
  – Grading structure encouraged all students to attempt some advanced phases

• Some Projects Done in Teams
Computational Tools

- Python
- MPI/LAM (PyPar package) On a 19 Node Pentium-4 Linux Cluster.
- Vpython, Graphics, and PPM Files for Visualization
- Stereographic Presentation with SVEN
Outcome

• Projects Keep Students Active
• Concrete Focus Allows Learning of Necessary Math and Computing
• Project Options Allow Students to Leverage Their Background/Strengths
• Very High Student Evaluations
Future Developments

• Calculus Sequence Revision
  – Reslicing Calc I and II
  – Applied Calculus: Derivatives and integrals conceptually/computationally through applications.
  – Foundational Calculus: ditto, but symbolically

• Computational Science Minor
Conclusions

• Our CS Curriculum is Evolving to Better Serve Science (and other) Majors
• Process Also Strengthens CS Major
  – Attracting Quality Students
  – Interdisciplinary Experience
  – Practical Problem-Solving Experience
• Open Question: Buy-In From Sciences