CIS 455/555
Computational Science

Introduction
Course Outline
Projects
Course Information

- Instructor:
  John Conery
  conery@cs.uoregon.edu
  Office hours TBA

- Class website:
  http://www.cs.uoregon.edu/cis455

- Textbook:
  Parallel Processing
  Wilkinson and Allen
  Prentice Hall (2005)
What is Computational Science?

- **Short answer:**
  - the use of modeling and simulation in scientific research

- **Characterized by**
  - high performance computing (very big problems)
  - visualization
  - application to “real world” problems
What is Computational Science?

- Old answer:
  - “scientific computing” or “scientific programming”

- Typically referred to creation of software for scientific applications
  - development of new algorithms
    - e.g. methods for solving systems of linear equations
  - languages and libraries for scientific applications
    - e.g. FORTRAN90, Lapack, …
  - data analysis
    - post hoc number crunching
What is Computational Science?

Better answer:
- using computers to solve scientific problems
- “computational” is an adjective that describes the type of science

Common phrase: “three-legged stool”
- theoretical science: construction of abstract descriptions, mathematical models
- experimental science: characterized by physical measurements, laboratory models
- computational science: simulation, visualization, analysis based on computational models
Example: Mutational Meltdown

- Lynch, Conery, and Bürger (1994, 1995)
- Question: can mutations alone cause extinction in sexually reproducing populations?

- Background:
  - majority of mutations are harmful
  - in finite asexual populations (bacteria, yeast, other very small organisms) deleterious mutations build up over time
  - since offspring are clones of their mothers fitness declines steadily (“Muller’s ratchet”), extinction is inevitable
Mutational Meltdown (cont’d)

- For sexually reproducing organisms, recombination can “flush out” deleterious mutations
  - mutation occurs in germ cell of one parent
  - offspring will have one mutant, one wild gene
  - mutation may not be passed on to children
- “Random genetic drift” in small populations can lead to fixation
  - all offspring have two copies of mutated gene
- Simulation results:
  - mildly deleterious mutations will accumulate, eventually cause extinction
  - mean time to extinction increases exponentially with population size
Mutational Meltdown (cont’d)

- Experimental approach: grow populations in a lab
  - need fast-reproducing organisms, method for tracking mutations, controls for other factors
- Theoretical approach: analytical model to predict mean time to extinction as function of population size, mutation rate, etc
  - tractable for only very simple (e.g. asexual) populations
- Computational approach: individual-based modeling
  - simulate new generations from previous generations
  - add random mutations
  - update model for new situations, e.g. “age structure”, demographics, …
Numeric Solution vs *ab initio* Simulation

- The meltdown project is an example of *ab initio* simulation
  - application based on a simulation of real-world objects
  - results determined by final state of the system
  - other examples: N-body problem in astrophysics, molecular dynamics in computational biology

- Other projects involve numeric solutions of mathematical models
  - exact solution, e.g. by symbolic integration, is preferred
  - realistic systems might be too complex, so numeric methods are used
  - examples: systems of differential equations for fluid dynamics (hydrology, weather prediction, lava flows, …)
A common acronym for this field is CSE, for Computational Science and Engineering

- “CS” has been taken already…

Many of the methods of computational science carry over directly to computational engineering

- example: car crash simulations
- wire-frame model of car is similar to mesh used to compute fluid flow
- time-stepped evaluation of energy flow through mesh
Grand Challenges

- Computers have been widely used in science since the 1940’s
- CSE as a new discipline emerged in the late 1980’s
- Turning point: High Performance Computation and Communication Program (HPCC)
  - funded by US Congress in 1991
  - identified several “Grand Challenge” science and engineering problems that might be solved with the help of high performance computers
  - see “Overview of Scientific Computing” (Fosdick et al, 1994) and “Grand Challenges” (ITRD, 2003), both on line
Grand Challenges (cont’d)

- Examples of Grand Challenge research areas:
  - atmospheric modeling
    - weather prediction, storm-scale models
  - molecular dynamics
    - protein-DNA interactions; “rational drug design”
  - material science
    - development of superconducting materials
  - automotive engineering
    - car crash simulations (varying speeds, angles, road conditions, …)
    - more efficient engines
Informatics

- A growing area of CSE is related to Information Technology (IT)
- “Informatics” refers to management and use of large amounts of scientific information
  - scientific databases
    - PubMed
    - model organism databases in biology
  - advanced user interfaces and database query languages
  - ontology (knowledge representation)
  - federated databases
- Examples: bioinformatics, medical informatics, GIS, NASA’s earth-orbiting satellite (EOS), digital libraries, …
CSE @ UO

- Every science department at UO now has one or more computational science research projects
- Computational Science Institute
  - members from CS, Math, Physics, others
  - established 1995 to promote CSE research and campus resources
- Some CSE projects:
  - electronic structure of solids (Haydock)
  - models of lava flow (Cashman), geotomography (Toomey, Cuny)
  - neuroinformatics, brain imaging (Malony, Tucker, Nunnally)
  - neural networks and chemotaxis (Lockery, Conery)
  - ZFIN (Westerfield, Douglas)
Course Goals

- The main goal for this course is an understanding of CSE from a computer science perspective
  - high performance computer architecture
  - languages and libraries for parallel programming
  - algorithms and data structures commonly used in CSE applications
  - computer graphics for scientific visualization
  - databases and web interfaces
Deliverables

- Your responsibilities this term:
  - parallel programming projects
    - “hello world” (MPI warmup)
    - ab initio simulation (N-body project)
    - solution of differential equation (heat flow)
    - numeric optimization (TBD)
    - global grid service (Mandelbrot)
  - term paper
    - 455: report on current CSE project
    - 555: research paper
Course Outline

- Parallel Processing I
  - distributed computing; MPI

- N-Body Problems
  - molecular dynamics, force fields, astrophysics
  - Project 1: galaxy formation (C++ and MPI)

- Visualization
  - basic techniques; Matlab and/or R (use in writeups for all projects)

- Parallel Processing II
  - shared-memory multiprocessing; OpenMP
Course Outline (cont’d)

- Partial Differential Equations
  - grid construction, finite difference, finite element solvers
  - Project 2: Laplace’s equation (C++ and OpenMP)

- Optimization
  - hill-climbing, simulated annealing, genetic algorithms, gradient descent
  - Project 3: ??

- Databases and Servers
  - The Global Grid; GLOBUS toolkit; MPI-G2
  - Project 4: Mandelbrot server
Computing Resources

- New this term!
- IBM p690
  - 16 1.3GHz CPUs
  - 64GB RAM
  - 3rd floor Deschutes Hall
  - not available yet -- too hot :(
- Four IBM p655s
  - 8 1.5GHz CPUs
  - 32GB RAM
  - Neuroinformatics Center (RRP)
  - systems we’ll use for our course projects
Computing Resources (cont’d)

- Coming soon:
  - 8-node IBM JS20 cluster (Psych)
    - 2 CPUs per node
    - 4GB RAM per node
  - SGI “MARS” visualization server (CIS)
    - 16 CPUs
    - 8 graphics pipes
    - 16GB RAM
  - 5TB Storage Area Network (SAN) virtual file system (CIS)

- All systems use Linux O/S
Getting Started

- Read the first two chapters of the text
- Browse the course web site -- make sure you know where to find on-line documentation for MPI, OpenMP, etc
- Sign up for an account on the p655 system
  - if you don’t have one yet, get a CIS login
  - send me e-mail from your CIS account (or telling me your CIS login name)
  - a (semi-)automated process will create accounts on the p655
  - use ssh to connect, sftp to transfer files to/from CIS
- Log in and initialize your environment (.tcsh, .emacs, X, etc)
Project 1

- Due Fri Apr 9 (2 weeks from now)
- “Hello, World” in MPI on the p655
  - `main()` launches n processes
  - process 0 sends “hello i” to process i
  - each process prints its message when it receives it

- Download project tar file from the class web site
- Add a small amount of code, compile and test the program
- Submit new tar file by e-mail to conery@cs.uoregon.edu
Project 0

- Important!
- Send e-mail to conery@cs.uoregon.edu
  - your CIS login name
  - your nickname