Computer Organization and Programming for Scientific Computing
Ramani Duraiswami
Particulars

• Course on aspects of Computer Science necessary to achieve good performance in Scientific Computing
• Directed mostly at non CS majors who use computers extensively, but do not necessarily know how they are organized
• CS majors cannot use this as a “core” course
• Part of the Scientific Computing Certificate
Instructor

• Faculty member in computer science
• Research interests in scientific computing, machine learning, computational acoustics, fast algorithms, computer vision, etc.
  – Fluid mechanics research in a previous career
• Office: 3365 A.V. Williams
• Email: ramani AT umiacs.umd.edu
• Office hours: by appointment via email
Prerequisites

• Undergraduate numerical methods or numerical analysis course (460 or 466)

• Background Needed
  – Linear Algebra
  – Numerical Analysis
  – Programming
    • Matlab, C/C++ and/or FORTRAN 9x
  – Experience and/or interest in scientific computing

• Participation essential!
Homework

• Will try to have it at least every other week
• Will not be excessive
• Essential for learning --- must do as opposed to just read.
• Homework handed out last class of a week.
• Due last class of next week
• Thanksgiving week no homework
Projects & Exams

• There will be a final project that will require you to implement an algorithm in a field of your choice,
  – account for 20% of the grade.
  – Demonstrate how you use knowledge of the computer to improve performance

• Project to be chosen latest by October 13.
  – If you already have a project in mind you can discuss it with us

• Exams
  – intermediate exam worth 20%, week of October 1
  – final exam worth 20%. Finals Week
Scientific Computing
Big Picture

• Object of all science
• Efficiency and better understanding
• Scientific Method: Experiment/Hypothesis
• Now Simulation/Hypothesis
**Algorithm**: graham scan

Find rightmost lowest point; label it $P_0$.
Sort all other points angularly about $P_0$. (or all but one copy for multiple points).
Stack $S = (p_1, p_0) = (p_i, p_{i-1})$; indexes top.
$i = 2$
while $i < n$ do
    if $p_i$ is strictly left of $p_{i-1} P_0$
        then Push($p_i$, $S$) and set $i \leftarrow i + 1$
        else Pop($S$).
Problem Sizes Continue to Grow in all Fields

• Sensors are getting varied and cheaper; and storage is getting cheaper
• Cameras, microphones
• Text (all the newspapers, books, technical papers)
• Genome data
• Medical/biological data (X-Ray, PET, MRI, Ultrasound, Electron microscopy …)
• Climate (Temperature, Salinity, Pressure, Wind, Oxygen content, …)
• Finer detail in meshes
• From a 2006 talk by B. Chazelle
Good workmen know their tools

- Primitive model
Other models