Week 14
Review + What I Do

Paul Chew
CS 212 – Spring 2004

Announcements

- Part 4
  - There is no type-casting in Bali4, but you *can* assign to a supertype
    - If A is a supertype of B and aa and bb are declared as types A and B, respectively then
      - aa = bb; // is legal
      - bb = aa; // is not
  - Reminder: Download the latest SaM Simulator
- Part 4 regrades
  - Grades available by Wednesday
  - Regrade requests must be in by Friday
- CMS cleaning
  - If you are still in the course, but have been removed from CMS, let me know right away
- Sections
  - There *is* a Section meeting today (7:30 in 205 Upson)
  - There is *one* Section meeting on Monday (12:20 in ???)
- Please participate in the online Course Evaluation
  - There is also a paper survey to fill out today

Quick Overview

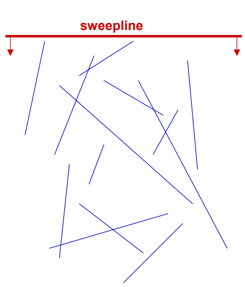
- Introduction
  - Computer architecture
  - Machine language & assembly language
  - Intro to SaM
- Compilers
  - Lexical analysis & parsing
  - Abstract Syntax Trees
  - Recursive descent parsing & code generation
- Software Engineering
  - Use of *abstraction*
  - Specification & validation
  - Testing & debugging
  - Models for software development
  - Top-down vs. bottom-up design
  - Evaluating a design

More Overview

- Implementing recursive functions
  - Stack frames
- Implementing arrays
- Implementing objects
  - Use of the *Heap*
  - Dispatch vectors
- Software Engineering Tools
  - Unix
  - Programming languages
  - Scripting languages
  - Regular expressions
  - Makefiles
  - Version control
  - UML
  - Profiling

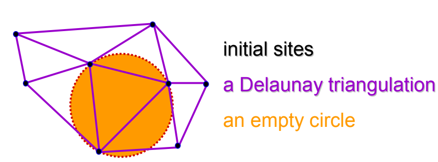
What I Do: Computational Geometry

- Using a computer to solve geometric problems
  - Get to use lots of data structure ideas
  - Example
    - Given n line segments in the plane, report all intersections
    - Uses both a PQ and a Balanced Tree
- Areas I work in
  - Motion Planning
  - Meshing
  - Shape Matching
    - computer vision
    - protein matching
  - More theoretical questions



The Delaunay Triangulation

- Has the "Empty Circle Property" (each Delaunay triangle's circumcircle is empty)
- Is commonly used for *meshing*



Adaptive Software Project (ASP)

- Adaptivity at three levels
  - Application Level
    - Choosing among physical models
  - Algorithm Level
    - Choosing among algorithms
  - System Level
    - Using system resources effectively
- Problem domains
  - fracture mechanics
  - reactive fluid flows

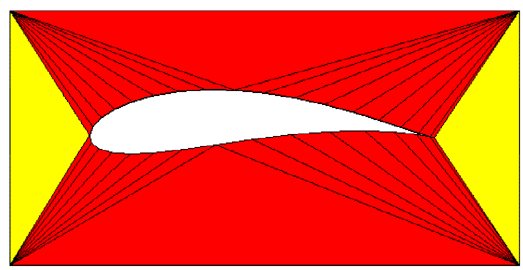
Necessary Tools

- Geometric modelers
  - Existing modelers are mostly inadequate
- Mesh generators
  - Need quality guarantees
- Visualization tools
- Components for different physical models
  - Different domains
    - fluid flow, structural mechanics, heat flow
  - Different scales
    - atomic scale, grain scale, structural scale
- Components for different solution techniques (algorithms)
  - finite elements, boundary elements, finite differences, …
- Framework for combining components
- System tools
  - Runtime environment
  - Dynamic load balancing
  - Fault tolerance (processor failure)
  - Compiler support

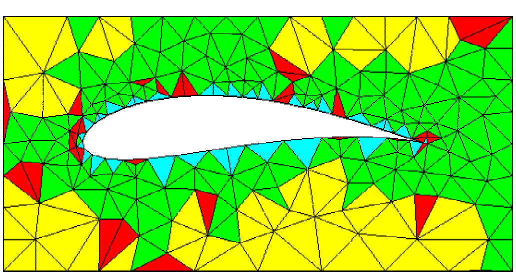
ASP Meshing Requirements

- Control of element density
  - Small elements (in “interesting” regions) for accuracy
  - Large elements (elsewhere) for efficiency
- Allow internal boundaries
  - Needed to represent, e.g., a crack
- Ideally: guarantee of element quality
  - Nice, but unnecessary for a single mesh
  - But we have many meshes as geometry changes over time

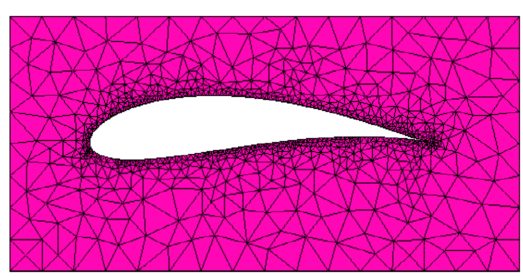
Initial Crude Mesh



During Improvement

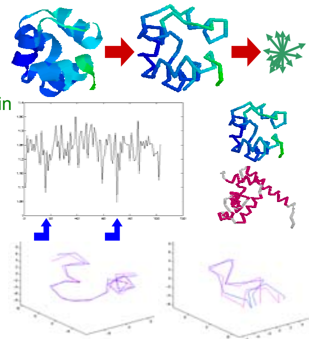


Final Mesh



Protein Shape and URMS

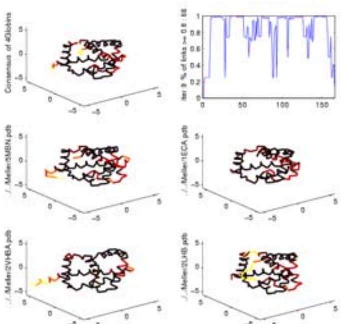
- Protein function is largely based on the protein's geometric shape
- How do we analyze protein shape?
- Our technique: URMS (Unit-vector Root Mean Square distance)
- Advantages
  - Insensitive to outliers
  - Efficient to compute
  - Equal weight for all portions



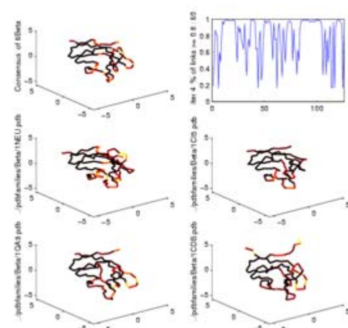
Protein Families & Consensus Shape

- Evolution theory ⇒ a protein ancient ancestor evolved into a family of proteins
- Membership in a protein family is expressed by sequence similarity, but is more strongly expressed by structure similarity
  - 25-30% sequence resemblance (almost always) ensures shape resemblance
- Goal: Create a *Consensus Shape Algorithm* that produces
  - a multiple alignment of structures, and
  - a single (core) structure that summarizes the structural information for a protein family

An Alpha Protein Family (Globins)



A Beta Protein Family



Unrelated Proteins