Computational Methods in Biology

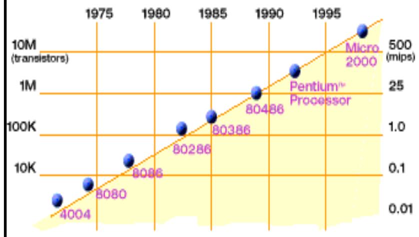
Guest Lecture
CS267
Spring 2005
UC Berkeley

Reading assignment (not mandatory):

Y. Duan and PA Kollman, "Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution," Science 282, 740 (1998).

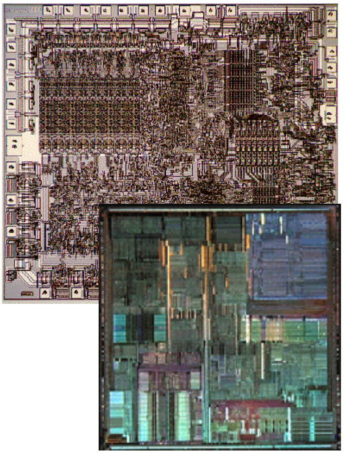
The Golden Age of Computing

Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.

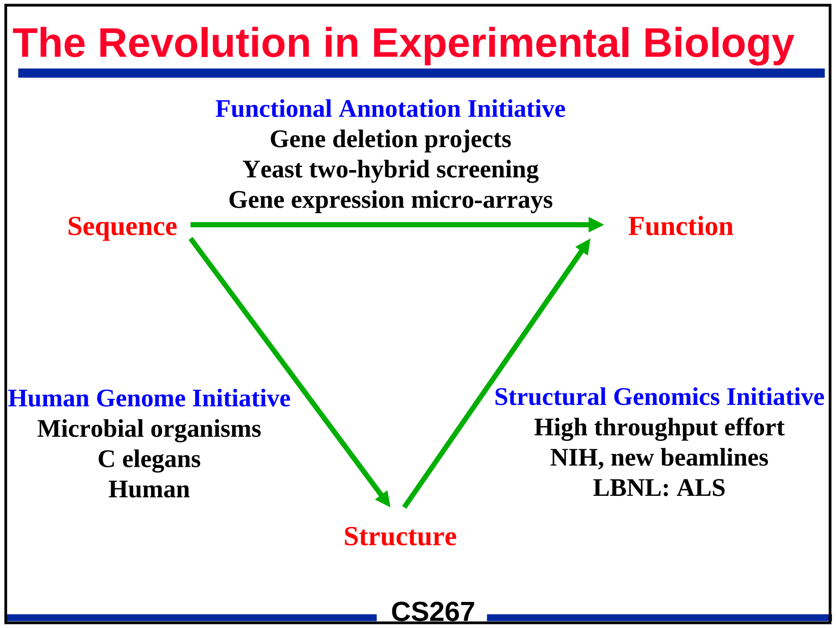


http://www.nersc.gov/~simon/cs267/Lec1.html

Intel 8080, 1975, 29K transistors



Intel Pentium Pro, 1995, 5.5M transistors



The Revolution in Experimental Biology

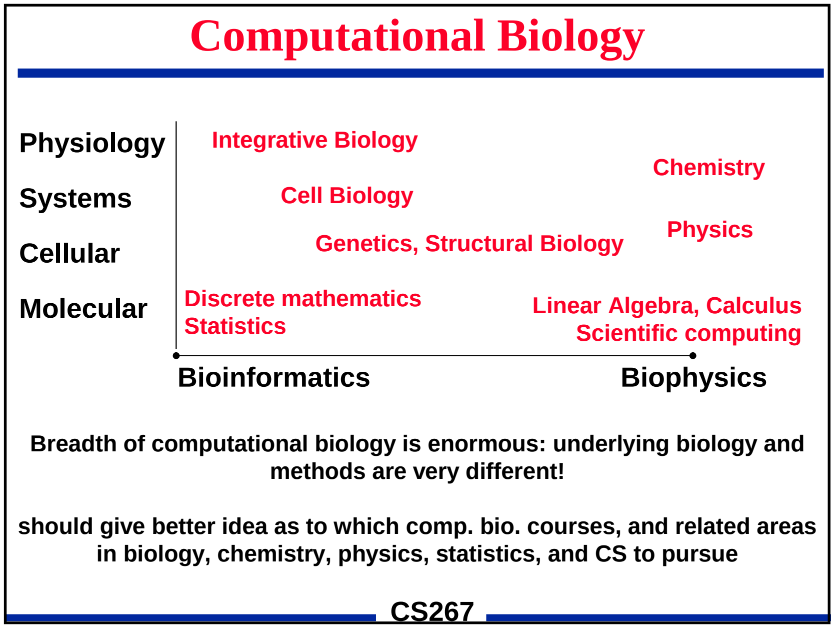
Functional Annotation Initiative
Gene deletion projects
Yeast two-hybrid screening
Gene expression micro-arrays

Sequence → Function

Human Genome Initiative
Microbial organisms
C elegans
Human

Structural Genomics Initiative
High throughput effort
NIH, new beamlines
LBNL: ALS

Structure



Computational Biology

Physiology — Integrative Biology — Chemistry

Systems — Cell Biology — Physics

Cellular — Genetics, Structural Biology

Molecular — Discrete mathematics, Statistics — Linear Algebra, Calculus, Scientific computing

Bioinformatics — Biophysics

Breadth of computational biology is enormous: underlying biology and methods are very different!

should give better idea as to which comp. bio. courses, and related areas in biology, chemistry, physics, statistics, and CS to pursue

BE143/243: Class Information

Course Time and Place:	MWF 3-4P 310 Hearst Mining
PreReqs:	Lower division physics/chem/bio Math 53 & 54
Lab:	Tu, 5-6pm, 1171 Etcheverry
Instructor:	Teresa Head-Gordon Department of Bioengineering Donner 272 TLHead-Gordon@lbl.gov
TA:	TA in charge of computer lab, all homework assignments

Text/Assessment

Text: Understanding Molecular Simulation: From algorithms to applications, D. Frenkel and B. Smit (Academic Press, 1996).

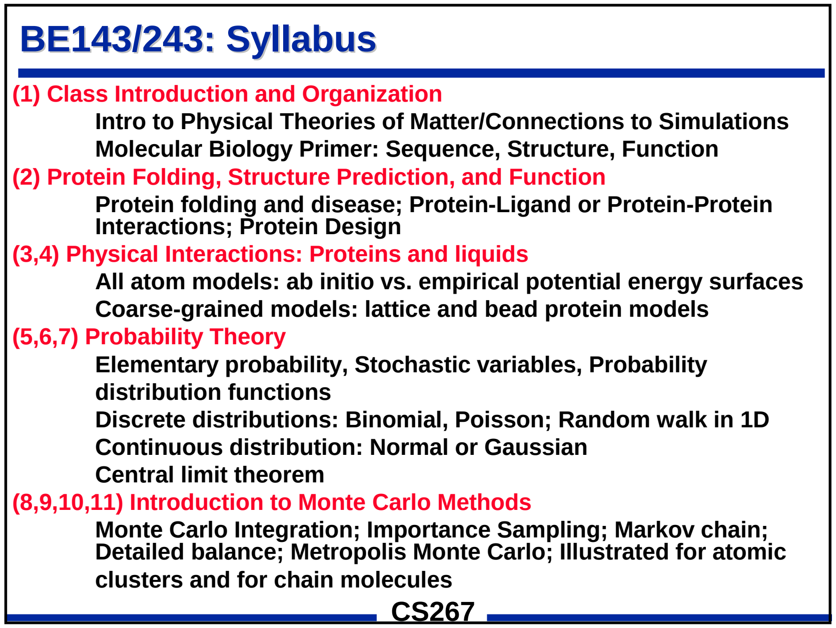
Text Resources:

- Computer Simulation of Liquids, M. P. Allen and D.J. Tildesley (Oxford Univ. Press) 1997.
- Numerical Recipes, the Art of Scientific Computing, W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling (Cambridge) 1989.
- Molecular Modelling: Principles and Applications, Andrew R. Leach, Prentice Hall.

Web-based notes and hand-outs

Assessment:		
	Homework	(40%)
	Mid-term	(20%)
	Final Project	(40%)

Homework is critical for final project that involves a class competition



BE143/243: Syllabus

(1) Class Introduction and Organization

Intro to Physical Theories of Matter/Connections to Simulations

Molecular Biology Primer: Sequence, Structure, Function

(2) Protein Folding, Structure Prediction, and Function

Protein folding and disease; Protein-Ligand or Protein-Protein Interactions; Protein Design

(3,4) Physical Interactions: Proteins and liquids

All atom models: ab initio vs. empirical potential energy surfaces

Coarse-grained models: lattice and bead protein models

(5,6,7) Probability Theory

Elementary probability, Stochastic variables, Probability distribution functions

Discrete distributions: Binomial, Poisson; Random walk in 1D

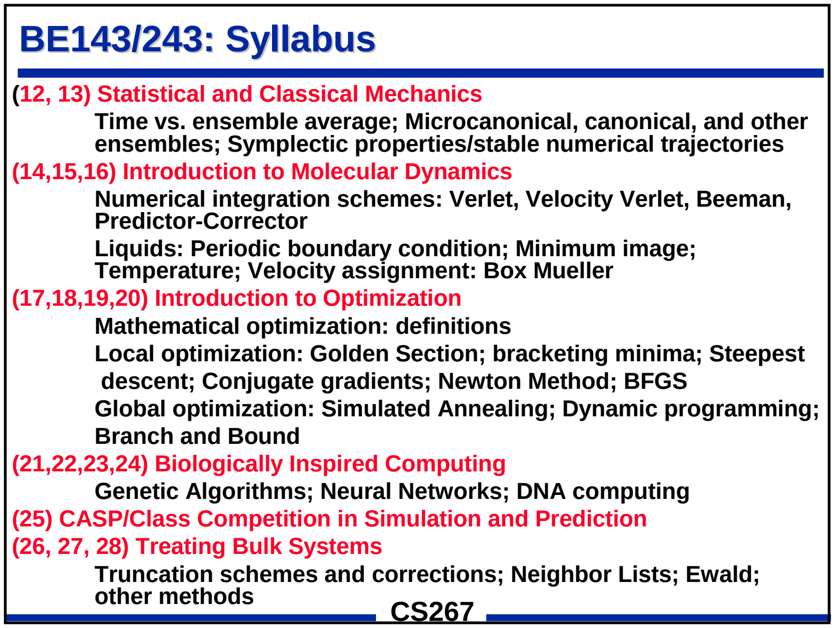
Continuous distribution: Normal or Gaussian

Central limit theorem

(8,9,10,11) Introduction to Monte Carlo Methods

Monte Carlo Integration; Importance Sampling; Markov chain; Detailed balance; Metropolis Monte Carlo; Illustrated for atomic clusters and for chain molecules

CS267



BE143/243: Syllabus

(12, 13) Statistical and Classical Mechanics

Time vs. ensemble average; Microcanonical, canonical, and other ensembles; Symplectic properties/stable numerical trajectories

(14,15,16) Introduction to Molecular Dynamics

Numerical integration schemes: Verlet, Velocity Verlet, Beeman, Predictor-Corrector

Liquids: Periodic boundary condition; Minimum image; Temperature; Velocity assignment: Box Mueller

(17,18,19,20) Introduction to Optimization

Mathematical optimization: definitions

Local optimization: Golden Section; bracketing minima; Steepest descent; Conjugate gradients; Newton Method; BFGS

Global optimization: Simulated Annealing; Dynamic programming; Branch and Bound

(21,22,23,24) Biologically Inspired Computing

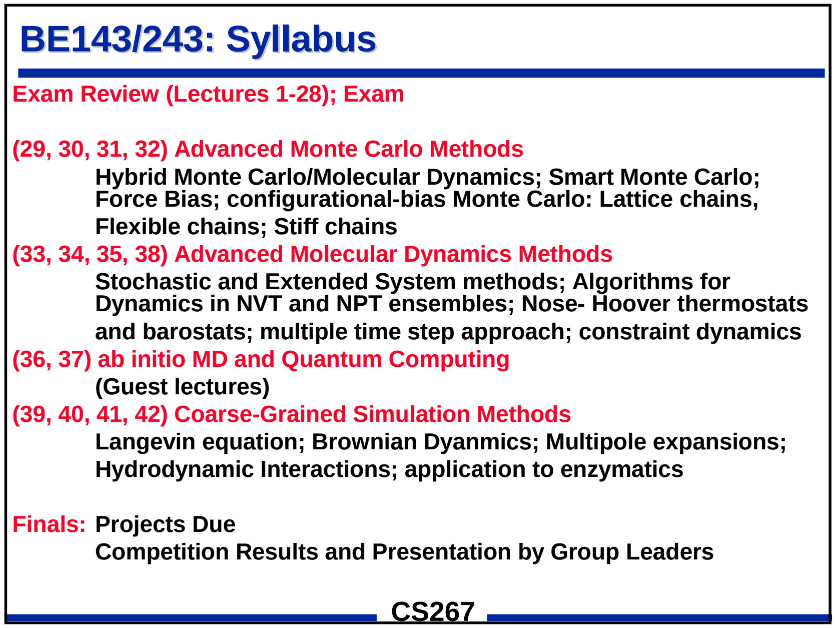
Genetic Algorithms; Neural Networks; DNA computing

(25) CASP/Class Competition in Simulation and Prediction

(26, 27, 28) Treating Bulk Systems

Truncation schemes and corrections; Neighbor Lists; Ewald; other methods

CS267



BE143/243: Syllabus

Exam Review (Lectures 1-28); Exam

(29, 30, 31, 32) Advanced Monte Carlo Methods

Hybrid Monte Carlo/Molecular Dynamics; Smart Monte Carlo; Force Bias; configurational-bias Monte Carlo: Lattice chains, Flexible chains; Stiff chains

(33, 34, 35, 38) Advanced Molecular Dynamics Methods

Stochastic and Extended System methods; Algorithms for Dynamics in NVT and NPT ensembles; Nose- Hoover thermostats and barostats; multiple time step approach; constraint dynamics

(36, 37) ab initio MD and Quantum Computing

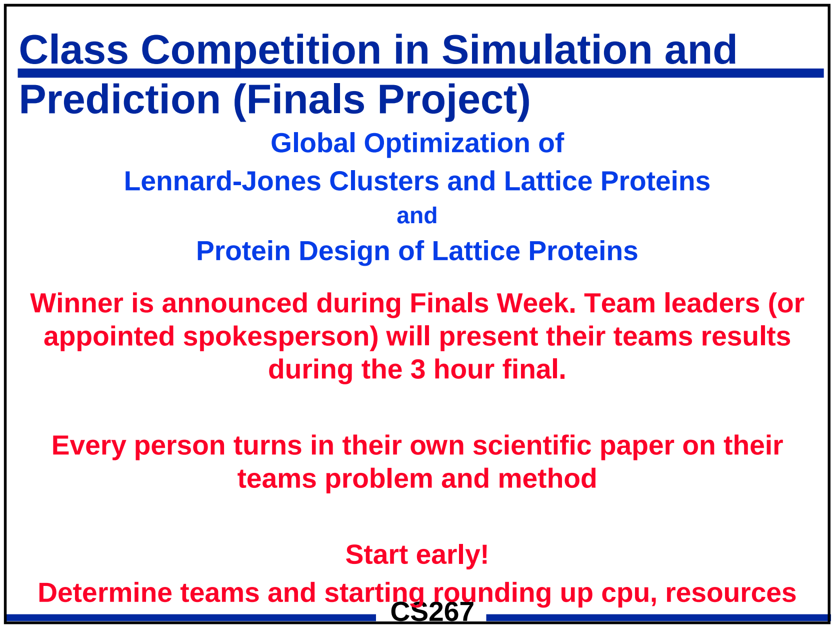
(Guest lectures)

(39, 40, 41, 42) Coarse-Grained Simulation Methods

Langevin equation; Brownian Dyanmics; Multipole expansions; Hydrodynamic Interactions; application to enzymatics

Finals: Projects Due

Competition Results and Presentation by Group Leaders



Class Competition in Simulation and Prediction (Finals Project)

Global Optimization of

Lennard-Jones Clusters and Lattice Proteins

and

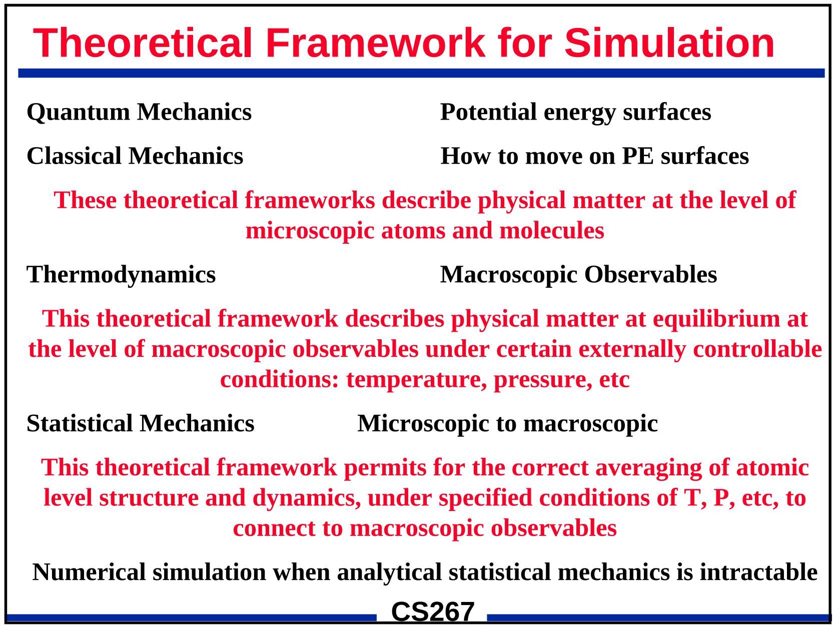
Protein Design of Lattice Proteins

Winner is announced during Finals Week. Team leaders (or appointed spokesperson) will present their teams results during the 3 hour final.

Every person turns in their own scientific paper on their teams problem and method

Start early!

Determine teams and starting rounding up cpu, resources



Theoretical Framework for Simulation

Quantum Mechanics — **Potential energy surfaces**

Classical Mechanics — **How to move on PE surfaces**

These theoretical frameworks describe physical matter at the level of microscopic atoms and molecules

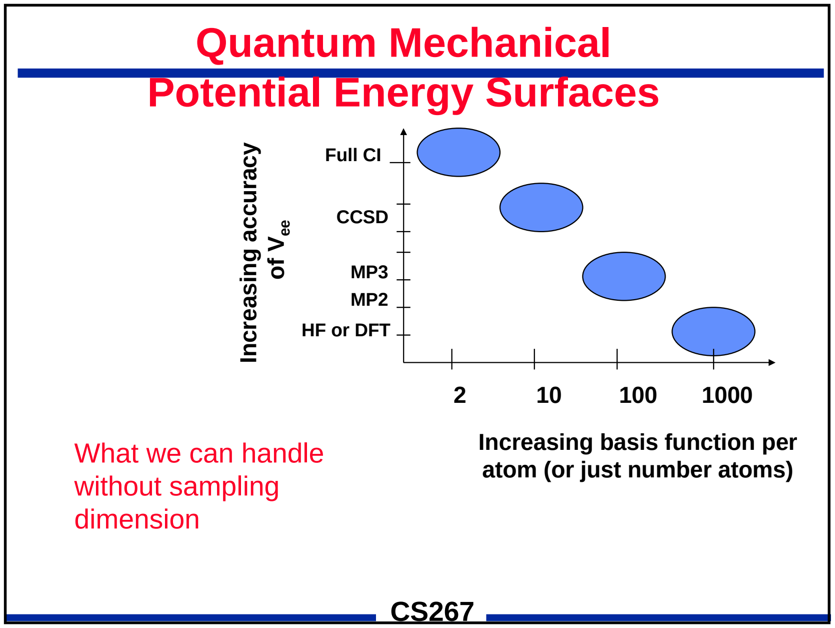
Thermodynamics — **Macroscopic Observables**

This theoretical framework describes physical matter at equilibrium at the level of macroscopic observables under certain externally controllable conditions: temperature, pressure, etc

Statistical Mechanics — **Microscopic to macroscopic**

This theoretical framework permits for the correct averaging of atomic level structure and dynamics, under specified conditions of T, P, etc, to connect to macroscopic observables

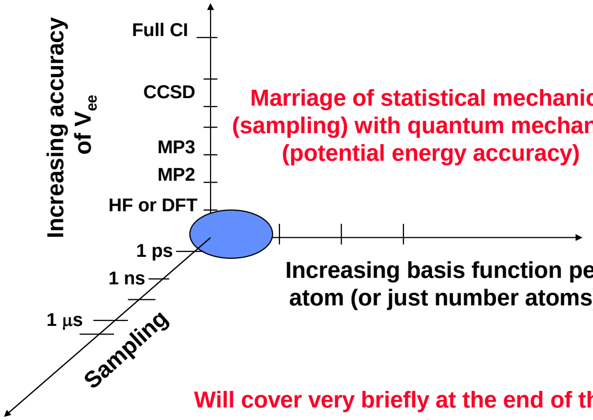
Numerical simulation when analytical statistical mechanics is intractable



Quantum Mechanical Potential Energy Surfaces

What we can handle without sampling dimension

Quantum Mechanical Potential Energy Surfaces



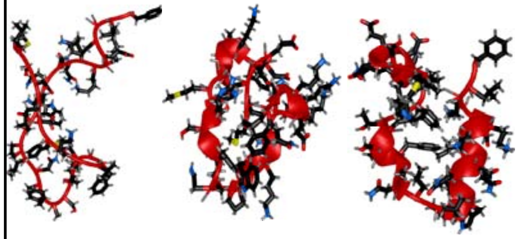
Marriage of statistical mechanics (sampling) with quantum mechanics (potential energy accuracy)

Increasing basis function per atom (or just number atoms)

Will cover very briefly at the end of the semester

Protein Folding

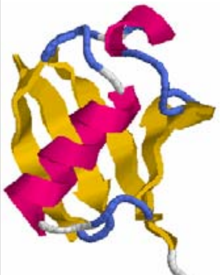
The theoretical framework of quantum mechanics is what allows us formulate Potential Energy Surfaces (PES) from “the beginning”



Y. Duan and PA Kollman, "Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution," Science 282, 740 (1998).

What quality of the potential energy surface to be sampled with statistical precision can we afford?

Empirical potential energy surfaces



It will not be tractable to do quantum mechanical potential energy functions for proteins because (1) too many atoms for typical (or even small) proteins, and (2) for the amount of sampling we will need to do

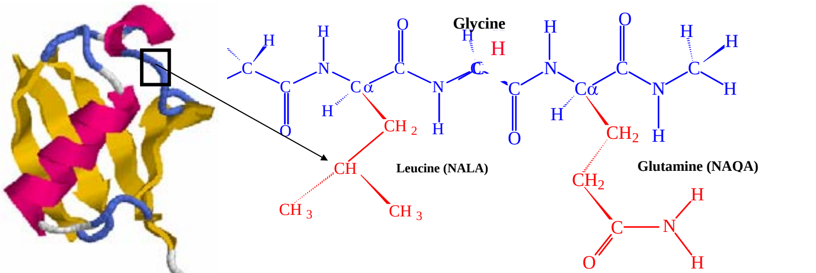
Instead we will consider empirical potential energy functions

$$V_{MM} = \sum_{i}^{\#Bonds} k_b\left(b_i - b_o\right)^2 + \sum_{i}^{\#Angles} k_\theta\left(\theta_i - \theta_o\right)^2 + \sum_{i}^{\#dihedrals} k_\phi\left[1 + \cos\left(n\phi + \delta\right)\right] +$$

$$\sum_{i}^{\#atoms}\sum_{i<j}^{\#atoms}\left\{\frac{q_i q_j}{r_{ij}} + 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right]\right\}$$

The first three sums are covalent or “chain connectivity” terms
The last double sum over i,j describes “non-bonded” terms

Empirical potential energy surfaces



Empirical PES are based on following approximations:

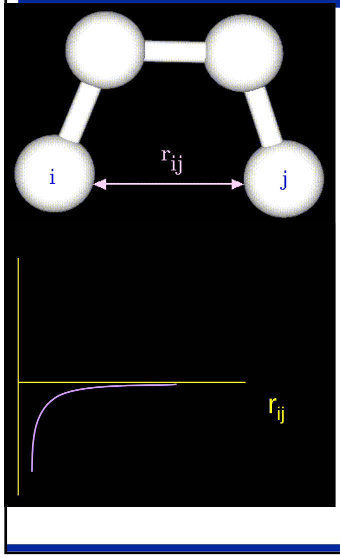
(1) Nuclei and electrons are lumped into atom-like particles.

(2) Atom-like particles are spherical and have a net charge

(3) Interactions are based on classical models that mimic or approximate QM functional forms

(4) Interaction parameters assigned to particular atoms:
C: aliphatic carbon, carbonyl carbon, etc

Coulomb’s Law for Electrostatics



$$V_{Electrostatics} = \sum_{i}^{\#atoms} \sum_{j}^{\#atoms} \frac{q_i q_j}{r_{ij}}$$

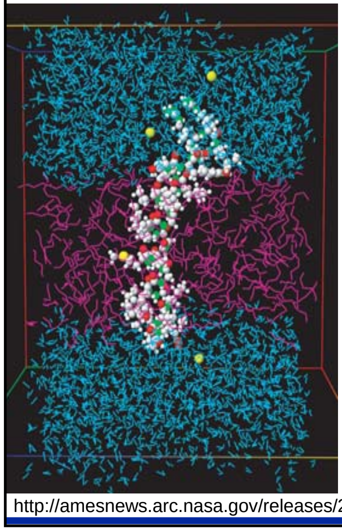
q_i : atomic charge

Short vs. Long-ranged interactions:

$$\begin{array}{lll} r^{-n} & n<3 & long \\ & n>3 & short \end{array}$$

We will talk about Ewald descriptions of long-ranged electrostatics later in the semester

Water and Protein Interactions



$$V_{LJ} = \sum_{i}^{\substack{\#atoms \\ protein}} \sum_{j}^{\substack{\#atoms \\ water}} 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right]$$

$$\varepsilon_{ij} = \left(\varepsilon_i \varepsilon_j \right)^{1/2}$$

$$\sigma_{ij} = \frac{\sigma_i + \sigma_j}{2}$$

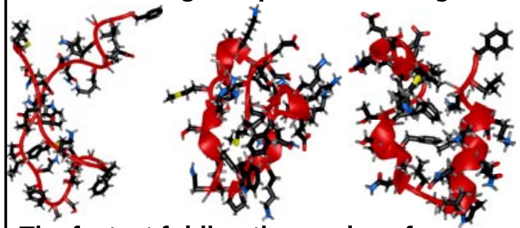
$$V_{Electrostatics} = \sum_{i}^{\substack{\#atoms \\ protein}} \sum_{j}^{\substack{\#atoms \\ water}} \frac{q_i q_j}{r_{ij}}$$

The empirical description of water (parameters) as focused on pure water liquid as opposed to its interaction with protein.

http://amesnews.arc.nasa.gov/releases/2001/01images/512/512.html

The Computational Cost of Protein Folding

Are all atom empirical force fields computationally tractable for something like protein folding?



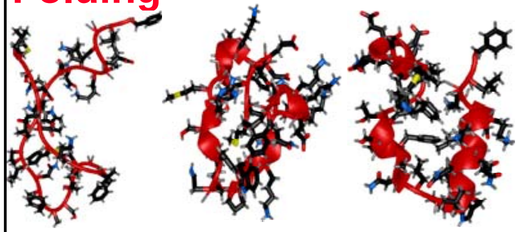
Y. Duan and PA Kollman, "Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution," Science 282, 740 (1998).

The fastest folding timescales of measurable protein folding is on the order of tens of microseconds: ~10^{-6} seconds=1μs.

Some of the earliest folding events (formation of secondary structure, hydrophobic collapse) occur faster than 1 microsecond

What does it take (computationally) to simulate a microsecond?

The Computational Cost of Protein Folding



Y. Duan and PA Kollman, "Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution," Science 282, 740 (1998).

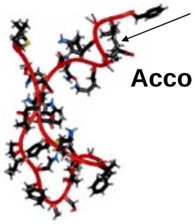
Let’s consider the heroic calculation by Duan and Kollman of 1μs simulation of the small 36 amino acid protein villin in a molecular description of water:

~500 protein atoms ~11,500 water atoms

N=12,000 atoms

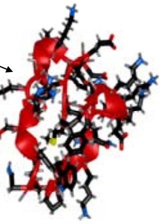
The Computational Cost of Protein Folding

In BE143/243 we will learn about basic molecular dynamics simulation. It is sufficient right now to say that we are evolving configurations in time of villin and water molecules



$\vec{R}^N(t=0)$

$\vec{R}^N(t=\tau)$



According to Newton's equations of motion

$$\vec{F} = m\vec{a}$$

$\vec{F}_i(t) \leftarrow$ and we need to evaluate the force on all atoms at every time step

The Computational Cost of Protein Folding

The computational cost of the force, which is the position derivative of the potential energy at each time step

$$\vec{F}_i = -\frac{\partial V}{\partial \vec{r}_i}$$

is dominated by the evaluation of the double sum over non-bonded interactions

$$V_{Non-bonded} = \sum_i^N \sum_j^N 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \frac{q_i q_j}{4\pi\varepsilon_o r_{ij}}$$

which scales as N^2 where N=number of atoms.

Later lets improve on this

The Computational Cost of Protein Folding

Lets say that each force evaluation costs 100 operations (computer evaluations such as adds, divides, multiples, memory fetches, etc). Therefore for villin in water:

100 ops x $(12{,}000)^2$= $1.44 \text{ x}10^{10}$ ops per time step

How many time steps do we have to do? To execute stable trajectories we need a time step of

t=1.0 femtosecond (fs)   where   1fs=10^{-15} seconds

and 1.0 microsecond (10^{-6} seconds) of simulation requires

10^{-6} seconds/(10^{-15}seconds/timestep)=10^{9} time steps

The Computational Cost of Protein Folding

Therefore one 1us simulation of villin protein in water requires

($1.44 \text{ x}10^{10}$ ops/time step)x(10^{9} time steps)=$1.44\text{x}10^{19}$ ops

However, one folding trajectory is only anecdotal. We require thousands of trajectories to get the correct folding measure of a population or ensemble of folding events (more typical of real experiments).

$10^{3}\text{x}1.44\text{x}10^{19}$ ops=$1.44\text{x}10^{22}$ ops

This outlines how many computer operations we need to simulate the fastest protein folding experiment for a very small protein in water

The Computational Cost of Protein Folding

Current best supercomputers are 10-100 teraflops (teraops) or 10^{13} ops/second wall time

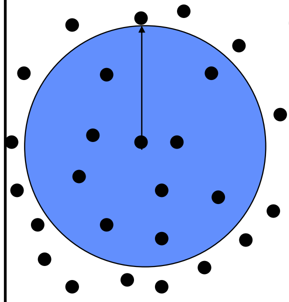
Lets imagine that we have exclusive and dedicated access to this supercomputer for as long as we need to finish this protein folding calculation.

(1.44×10^{22} ops)/(10^{13} ops/second wall time)=1.44×10^{9} seconds

(1.44×10^{9} seconds)/(8.64×10^{4} seconds/day)

1.67×10^{4} days~46 years

How did they do it?



(1) They did one trajectory

(2) This published calculation truncated electrostatic interactions at 8Å when ranges more like 15-20Å are a better estimate. So effectively $N^2 \sim M^2$

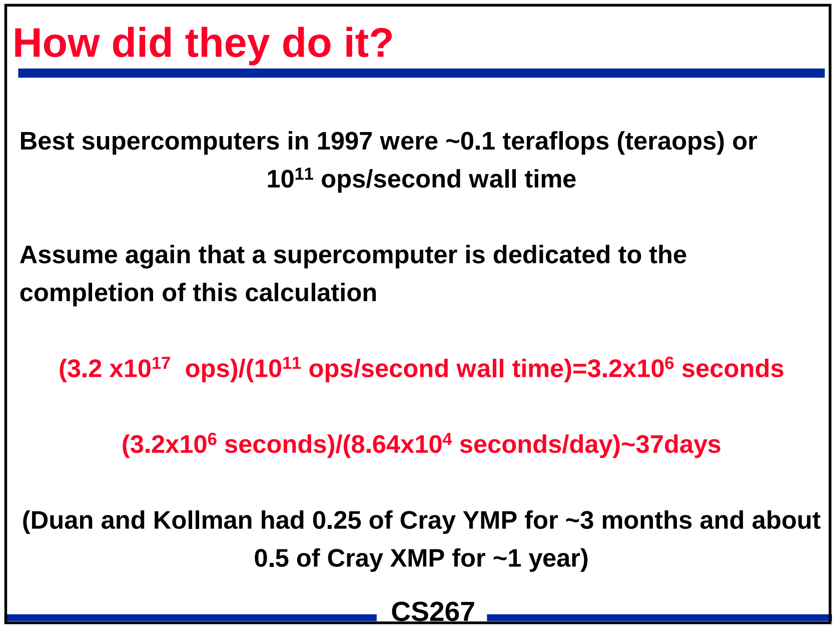
What is M? Assume a constant density of atoms, so that atom number increases with larger volume elements

$\sim 8^3/15^3 \sim 15\%$ of 12,000

or ~1800 atoms

100 ops x $(1800)^2$= 3.2×10^{8} ops per time step

(3.2×10^{8} ops/time step)x(10^{9} time steps)=3.2×10^{17} ops



How did they do it?

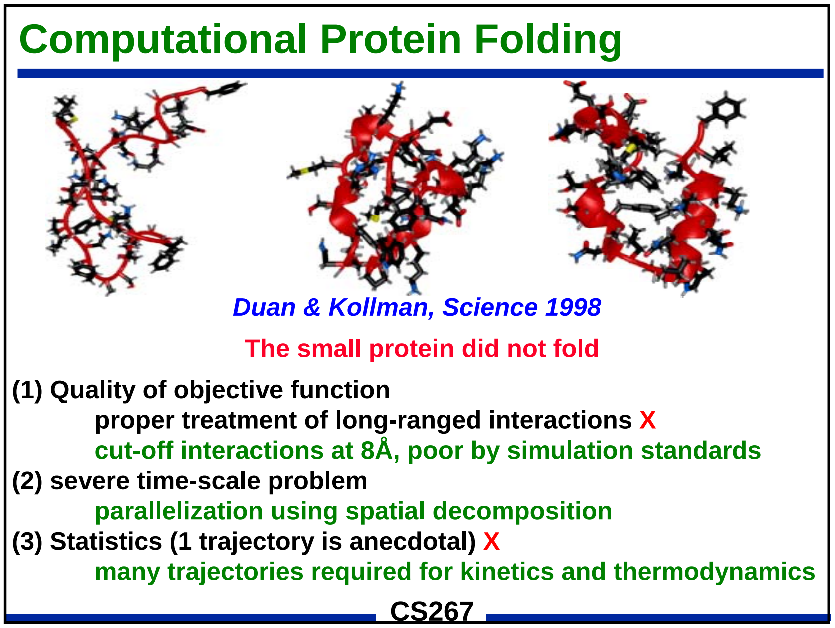
Best supercomputers in 1997 were ~0.1 teraflops (teraops) or 10^{11} ops/second wall time

Assume again that a supercomputer is dedicated to the completion of this calculation

$(3.2 \times 10^{17}$ ops)/(10^{11} ops/second wall time)=3.2×10^{6} seconds

(3.2×10^{6} seconds)/(8.64×10^{4} seconds/day)~37days

(Duan and Kollman had 0.25 of Cray YMP for ~3 months and about 0.5 of Cray XMP for ~1 year)



Computational Protein Folding

Duan & Kollman, Science 1998

The small protein did not fold

(1) Quality of objective function

proper treatment of long-ranged interactions X

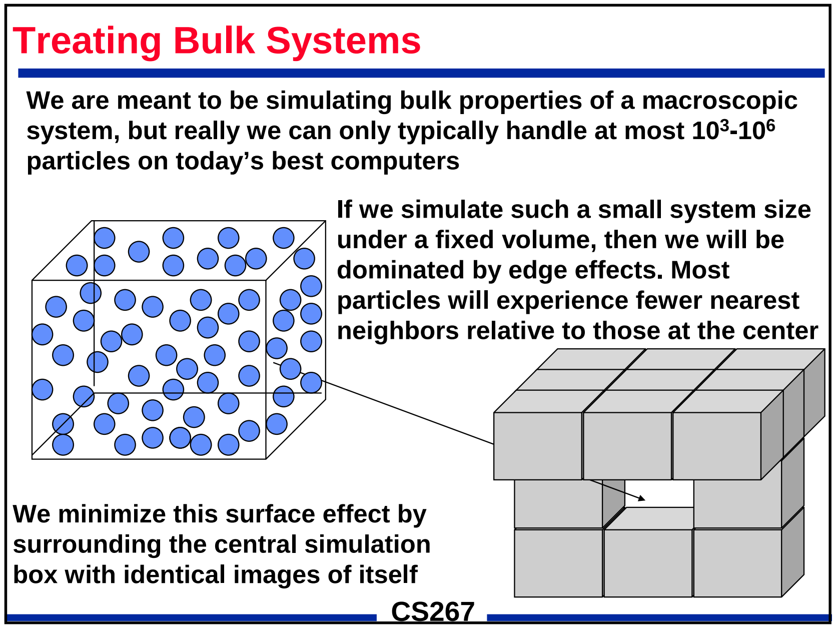
cut-off interactions at 8Å, poor by simulation standards

(2) severe time-scale problem

parallelization using spatial decomposition

(3) Statistics (1 trajectory is anecdotal) X

many trajectories required for kinetics and thermodynamics

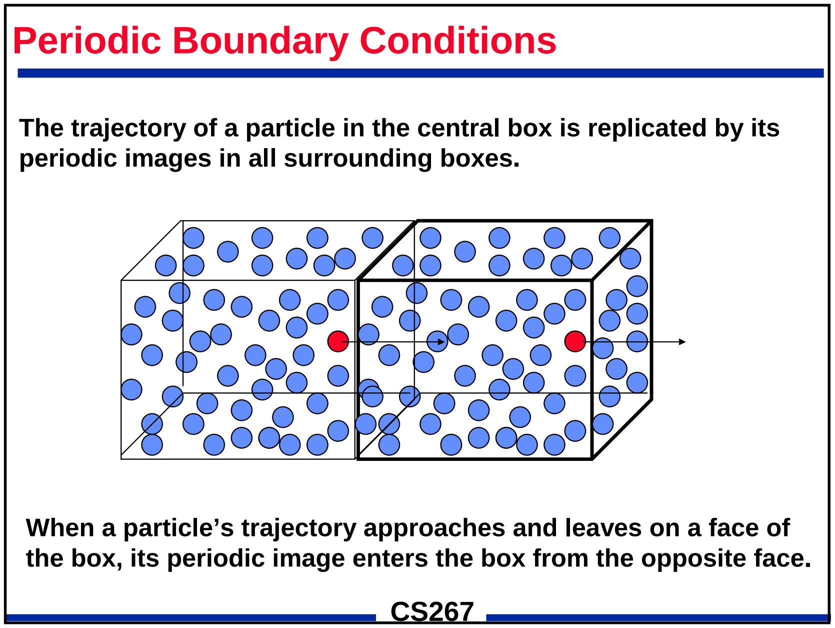


Treating Bulk Systems

We are meant to be simulating bulk properties of a macroscopic system, but really we can only typically handle at most 10^3-10^6 particles on today's best computers

If we simulate such a small system size under a fixed volume, then we will be dominated by edge effects. Most particles will experience fewer nearest neighbors relative to those at the center

We minimize this surface effect by surrounding the central simulation box with identical images of itself



Periodic Boundary Conditions

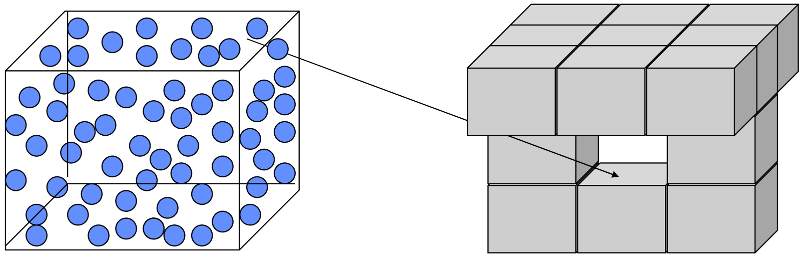
The trajectory of a particle in the central box is replicated by its periodic images in all surrounding boxes.

When a particle's trajectory approaches and leaves on a face of the box, its periodic image enters the box from the opposite face.

Treating Bulk Systems

But now we could in principle have an infinite number of interactions

$$V_{tot} = \sum_{\vec{n}} \sum_{i} \sum_{j} V\left(\vec{r}_{ij} + L\vec{n}\right)$$



This reintroduces the original problem of large N needed to simulate bulk systems, and with periodicity to boot!

Truncation for Short-ranged Potential

(1) Simple truncation

$$V = \sum_{i}^{\#atoms} \sum_{j}^{\#atoms} V\left(r_{ij}\right) \qquad r_{ij} \le r_{cut}$$

$$V = 0 \qquad r_{ij} > r_{cut}$$

(2) Truncation and Shift

$$V = \sum_{i}^{\#atoms} \sum_{j}^{\#atoms} V\left(r_{ij}\right) - V\left(r_{cut}\right) \qquad r_{ij} \le r_{cut}$$

$$V = 0 \qquad r_{ij} > r_{cut}$$

Suitable only for Monte Carlo. Not suitable for molecular dynamics since forces are discontinuous at $r_{cut,}$ and EOM become unstable

Truncation for Short-ranged Potential

(3) Truncation and Shift

$$V = \sum_{i}^{\#atoms} \sum_{j}^{\#atoms} V(r_{ij}) - V(r_{cut}) - \left.\frac{dV(r)}{dr}\right|_{r=r_{cut}} (r - r_{cut}) \qquad r_{ij} \le r_{cut}$$

$$V = 0 \qquad r_{ij} > r_{cut}$$

Where now discontinuity has been shifted to second derivatives

Now define a correction to the missing interactions as r_{cut}

$$V_{LJ} = \sum_{i}^{\#atoms} \sum_{j}^{\#atoms} V(r_{ij}^{cut}) + \frac{N\rho}{2} \int_{r^{cut}}^{\infty} V(r) 4\pi r^2 dr$$

Which assumes that the interaction is isotropic beyond r_{cut} with constant density ρ.

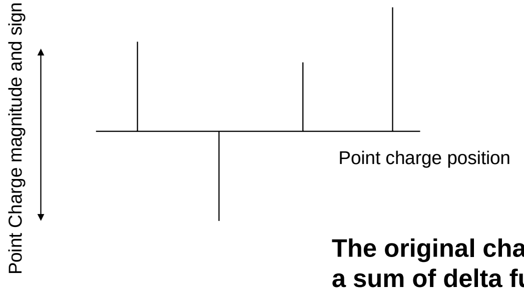
But note that correction becomes unbounded for potentials that are long-ranged: r^{-n} where n<3

Long-ranged potentials: Ewald Sum

$$V_{tot} = \sum_{i} \sum_{j} \sum_{\vec{n}}{}' \frac{q_i q_j}{\left|\vec{r}_{ij} + L\vec{n}\right|}$$

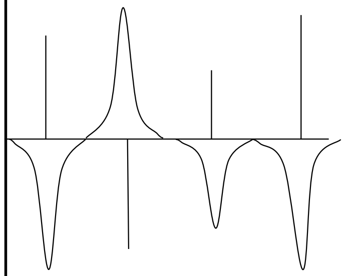
$$\vec{n} = (n_x L, n_y L, n_z L)$$

This sum is only conditionally convergent (depending on the order in which you add the terms).



The original charge distribution is a sum of delta function charges, and the interactions between charges decays as $1/r_{ij}$

Long-ranged potentials: Ewald Sum



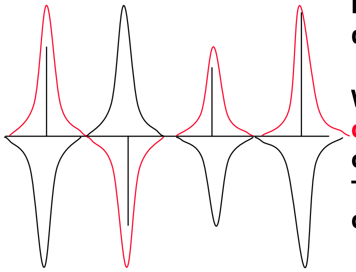
Instead we will introduce a diffuse charge distribution around charge i with opposite sign. For convenience we will make this a Gaussian charge distribution.

At large distances (near the tails) this screening charge value goes to zero rapidly.

$$\rho(r) \Rightarrow -q_i(\alpha/\pi)^{3/2}\, exp(\alpha r^2)$$

Therefore the original sum is more rapidly convergent than 1/r due to this screening.

Long-ranged potentials: Ewald Sum



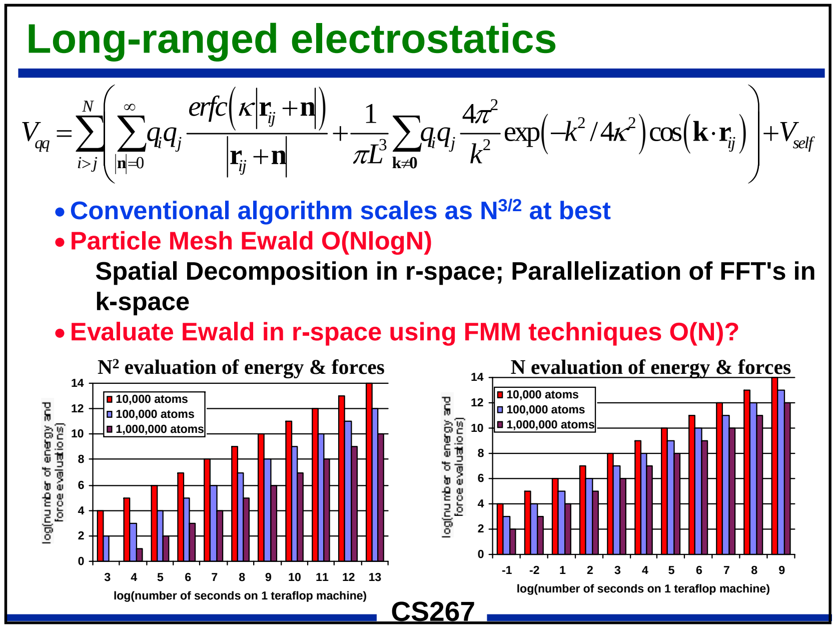
But this is not the true charge distribution itself.

We add back in a compensating charge distribution that will cancel out the screened charge distribution. This now will result in two fully convergent sums.

We will reformulate the original non-convergent sum

$$V_{tot} = \sum_i \sum_j \sum_{\vec{n}} \frac{q_i q_j}{\left|\vec{r}_{ij} + L\vec{n}\right|} = \sum_i q_i \Phi(\vec{r}_i)$$

with two sums: a real-space sum (r-sum: screened) and inverse-space sum (k-sum: compensating) which we can derive from Poisson's equation $-\nabla^2\Phi(\vec{r}) = 4\pi\rho(\vec{r})$



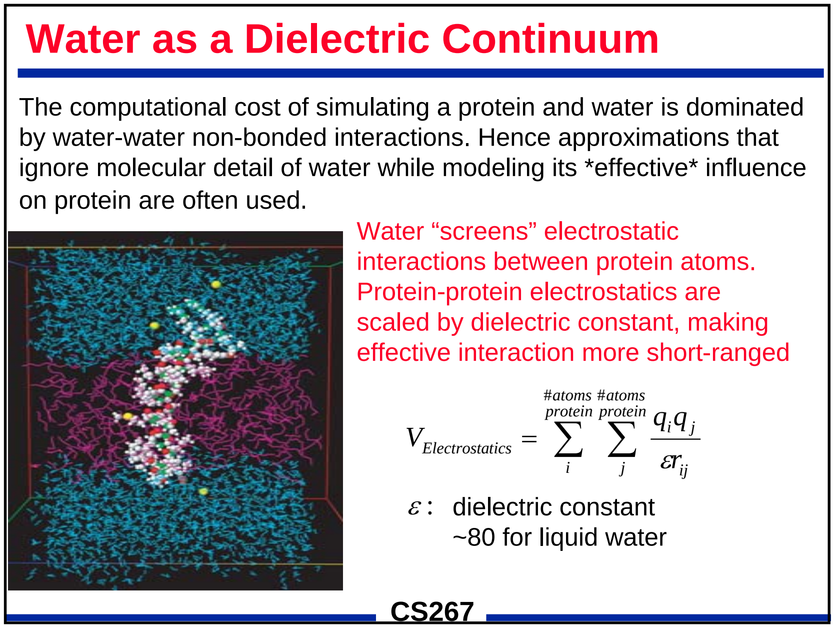
Long-ranged electrostatics

$$V_{qq}=\sum_{i>j}^{N}\left(\sum_{|\mathbf{n}|=0}^{\infty}q_iq_j\frac{erfc\left(\kappa\left|\mathbf{r}_{ij}+\mathbf{n}\right|\right)}{\left|\mathbf{r}_{ij}+\mathbf{n}\right|}+\frac{1}{\pi L^3}\sum_{\mathbf{k}\neq\mathbf{0}}q_iq_j\frac{4\pi^2}{k^2}\exp\left(-k^2/4\kappa^2\right)\cos\left(\mathbf{k}\cdot\mathbf{r}_{ij}\right)\right)+V_{self}$$

- Conventional algorithm scales as $N^{3/2}$ at best
- Particle Mesh Ewald O(NlogN)

  Spatial Decomposition in r-space; Parallelization of FFT's in k-space
- Evaluate Ewald in r-space using FMM techniques O(N)?

N^2 evaluation of energy & forces
log(number of energy and force evaluations)
log(number of seconds on 1 teraflop machine)
10,000 atoms
100,000 atoms
1,000,000 atoms
0
2
4
6
8
10
12
14
3
4
5
6
7
8
9
10
11
12
13
N evaluation of energy & forces
log(number of energy and force evaluations)
log(number of seconds on 1 teraflop machine)
10,000 atoms
100,000 atoms
1,000,000 atoms
0
2
4
6
8
10
12
14
-1
-2
1
2
3
4
5
6
7
8
9



Water as a Dielectric Continuum

The computational cost of simulating a protein and water is dominated by water-water non-bonded interactions. Hence approximations that ignore molecular detail of water while modeling its *effective* influence on protein are often used.

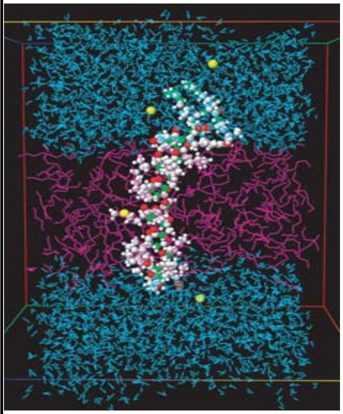
Water “screens” electrostatic interactions between protein atoms. Protein-protein electrostatics are scaled by dielectric constant, making effective interaction more short-ranged

$$V_{Electrostatics}=\sum_{i}^{\substack{\#atoms\\protein}}\sum_{j}^{\substack{\#atoms\\protein}}\frac{q_iq_j}{\varepsilon r_{ij}}$$

ε: dielectric constant
~80 for liquid water

Free Energy of Solvation

Protein-water interactions are most importantly manifested as the free energies of amino acid or protein solvation. We can qualitatively describe this as being composed of three separable terms



$$\Delta G_{solvation} = \Delta G_{electrostat} + \Delta G_{vdw} + \Delta G_{cavitation}$$

$$\Delta G_{electrostat} = -\frac{1}{2}\left(1-\frac{1}{\varepsilon}\right)\frac{q^2}{a}$$

$$\Delta G_{vdw} + \Delta G_{cavitation} = \gamma SA$$

$a:$ Effective cavity (Born) radius

$SA:$ Solvent accessible surface area

$\gamma:$ Parameter derived from transfer free energy data of alkanes from vacuum to water

http://amesnews.arc.nasa.gov/releases/2001/01images/512/512.html

Simplification of Protein Folding Simulations

Replace water molecules with Generalized Born/ Solvent Accessibility

$$\Delta G_{electrostat} = -\left(1-\frac{1}{\varepsilon}\right)\sum_{i=1}^{\#atoms}\sum_{j=1}^{\#atoms}\frac{q_i q_j}{r_{ij}} - \frac{1}{2}\left(1-\frac{1}{\varepsilon}\right)\sum_{i=1}^{\#atoms}\frac{q_i^2}{a_i}$$

$$\Delta G_{vdw} + \Delta G_{cavity} = \gamma SA$$

100 ops x (500 protein atoms)^2= 2.5 x10^7 ops per time step

2.5x10^7 ops/time step)x(10^9 time steps)=2.5x10^{16} ops

10^3 trajectories x (2.5 x10^{16} ops)=2.5x10^{19} ops

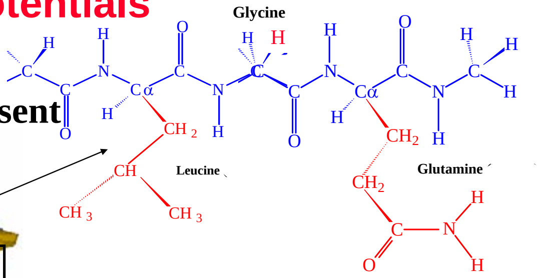
2.5x10^{19} ops /(10^{13} ops/second wall time)=2.5x10^6 seconds

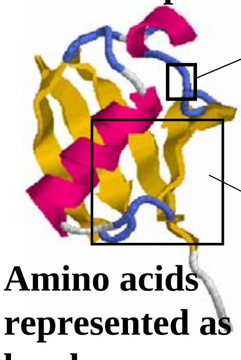
(2.5x10^6 seconds)/(8.64x10^4 seconds/day)~29days

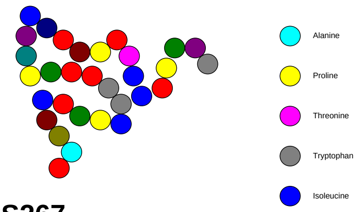
Folding@home

Molecular mechanics to Coarse-Grained Potentials

All atoms present



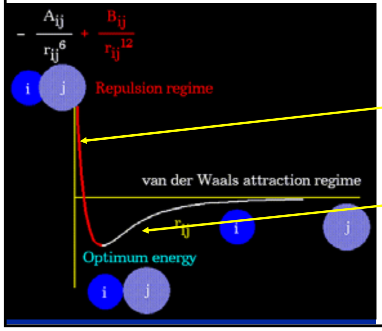




Amino acids represented as a bead

Simplifying Protein Interactions

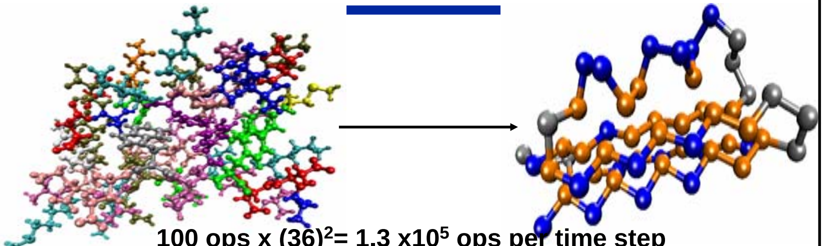
$$H = \sum_{bonds} k_b\left(b-b_0\right)^2 + \sum_{angles} k_\theta\left(\theta-\theta_0\right)^2 + \sum_{dihedrals}\{\, A[1+\cos\phi]+B[1-\cos\phi]+C[1+\cos 3\phi]+D\left[1+\cos\left(\phi+{}^{\pi}\!/_{4}\right)\right]\}+ \sum_{i,j\ge i+3} 4\varepsilon_H S_1\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - S_2\left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$



P-*, N-* interactions are repulsive
polar groups on protein surface

H-H interactions are attractive
mimics how hydrophobic groups segregate into core

Protein Bead Models



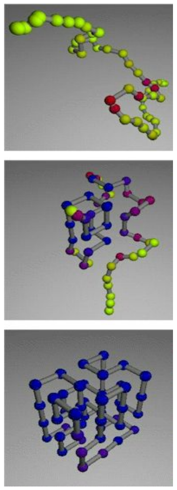
100 ops x $(36)^2$= 1.3 x10^5 ops per time step
1.3x10^5 ops/time step)x(10^9 time steps)=1.3x10^{14} ops
10^3 trajectories x (1.3 x10^{14} ops)=1.3x10^{17} ops
1.3x10^{17} ops /(10^{13} ops/second wall time)=1.3x10^4 seconds

(2.5x10^4 seconds)/(8.64x10^4 seconds/day)~3.5 hours

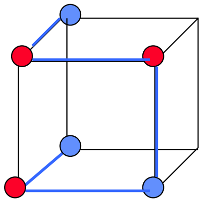
Now don’t need massive computing resources but more intermediate computing platforms are adequate

Protein bead models possible for those in class who are ambitious

Protein Lattice Models



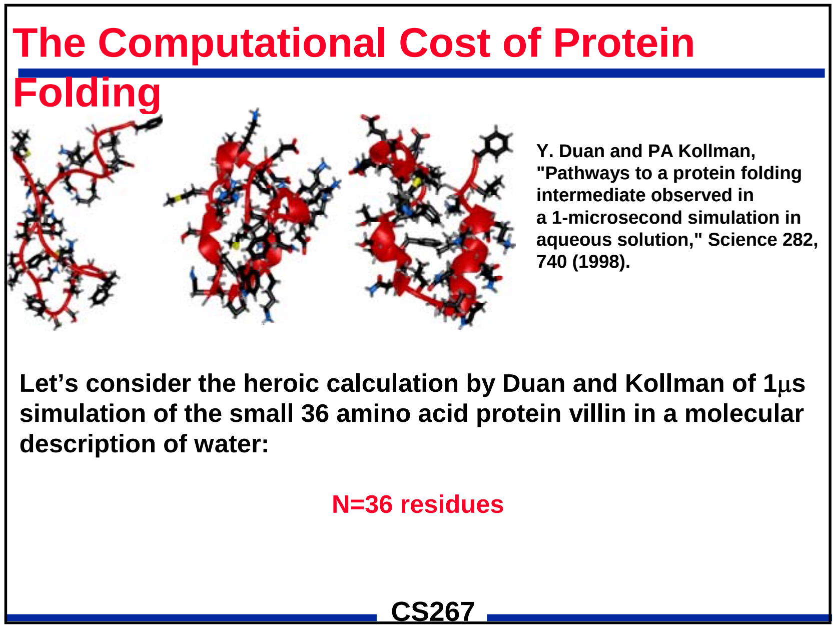
Protein lattice models: amino acids on a chain are restricted to points on some type of lattice



Example sequence PHHPHP

Greatly reduces the number of accessible protein states by restricting the continuous Cartesian space to discrete lattice points

http://www.lbl.gov/Science-Articles/Archive/model-protein-folding2.html

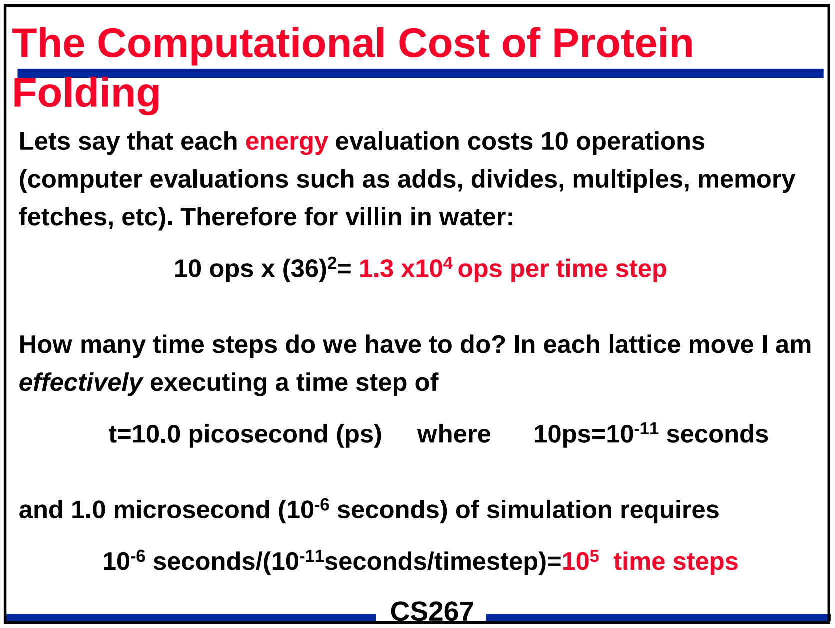


The Computational Cost of Protein Folding

Y. Duan and PA Kollman, "Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution," Science 282, 740 (1998).

Let’s consider the heroic calculation by Duan and Kollman of 1μs simulation of the small 36 amino acid protein villin in a molecular description of water:

N=36 residues



The Computational Cost of Protein Folding

Lets say that each energy evaluation costs 10 operations (computer evaluations such as adds, divides, multiples, memory fetches, etc). Therefore for villin in water:

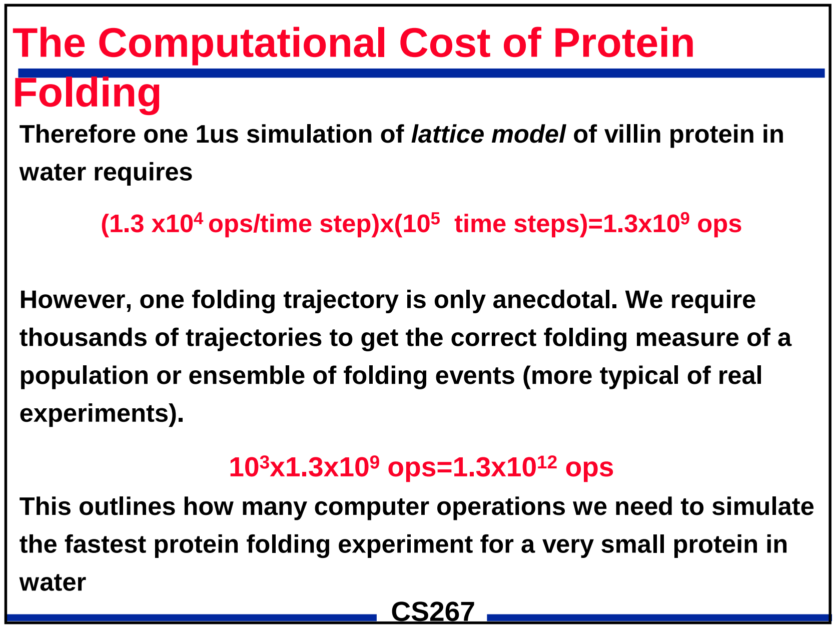
10 ops x $(36)^2$= 1.3 x10^4 ops per time step

How many time steps do we have to do? In each lattice move I am *effectively* executing a time step of

t=10.0 picosecond (ps) where 10ps=10^{-11} seconds

and 1.0 microsecond (10^{-6} seconds) of simulation requires

10^{-6} seconds/(10^{-11}seconds/timestep)=10^5 time steps



The Computational Cost of Protein Folding

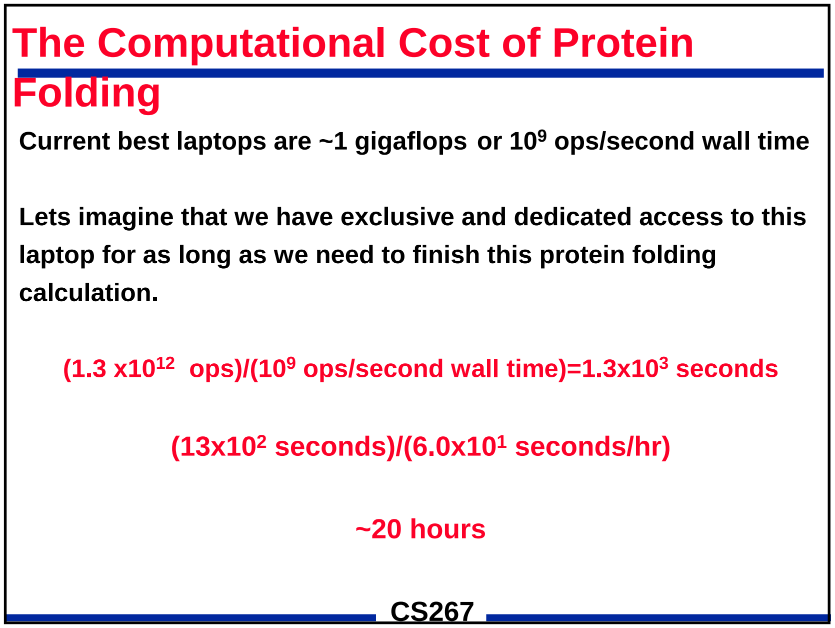
Therefore one 1us simulation of *lattice model* of villin protein in water requires

(1.3 x10^4 ops/time step)x(10^5 time steps)=1.3x10^9 ops

However, one folding trajectory is only anecdotal. We require thousands of trajectories to get the correct folding measure of a population or ensemble of folding events (more typical of real experiments).

10^3x1.3x10^9 ops=1.3x10^{12} ops

This outlines how many computer operations we need to simulate the fastest protein folding experiment for a very small protein in water



The Computational Cost of Protein Folding

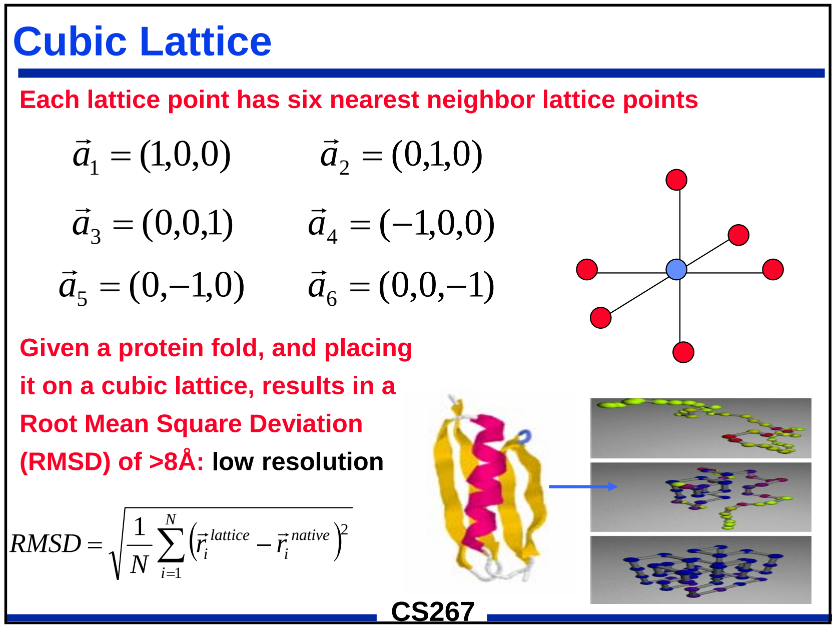
Current best laptops are ~1 gigaflops or 10^9 ops/second wall time

Lets imagine that we have exclusive and dedicated access to this laptop for as long as we need to finish this protein folding calculation.

(1.3 x10^{12} ops)/(10^9 ops/second wall time)=1.3x10^3 seconds

(13x10^2 seconds)/(6.0x10^1 seconds/hr)

~20 hours



Cubic Lattice

Each lattice point has six nearest neighbor lattice points

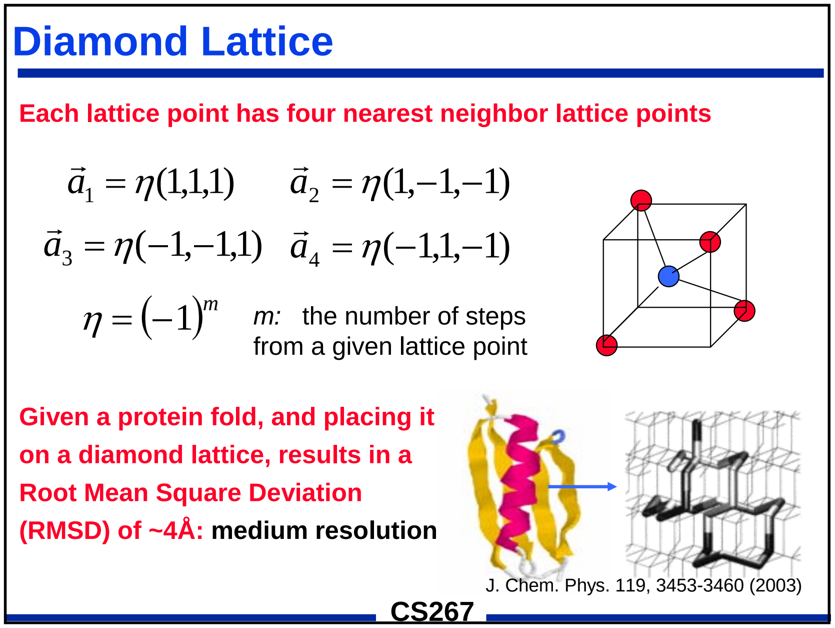
$$\vec{a}_1 = (1,0,0) \qquad \vec{a}_2 = (0,1,0)$$

$$\vec{a}_3 = (0,0,1) \qquad \vec{a}_4 = (-1,0,0)$$

$$\vec{a}_5 = (0,-1,0) \qquad \vec{a}_6 = (0,0,-1)$$

Given a protein fold, and placing it on a cubic lattice, results in a Root Mean Square Deviation (RMSD) of >8Å: **low resolution**

$$RMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\vec{r}_i^{\,lattice} - \vec{r}_i^{\,native}\right)^2}$$



Diamond Lattice

Each lattice point has four nearest neighbor lattice points

$$\vec{a}_1 = \eta(1,1,1) \qquad \vec{a}_2 = \eta(1,-1,-1)$$

$$\vec{a}_3 = \eta(-1,-1,1) \qquad \vec{a}_4 = \eta(-1,1,-1)$$

$$\eta = (-1)^m$$

m: the number of steps from a given lattice point

Given a protein fold, and placing it on a diamond lattice, results in a Root Mean Square Deviation (RMSD) of ~4Å: **medium resolution**

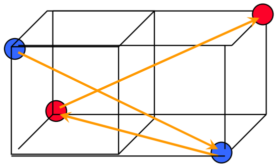
J. Chem. Phys. 119, 3453-3460 (2003)

Chess Knight (210) Lattice

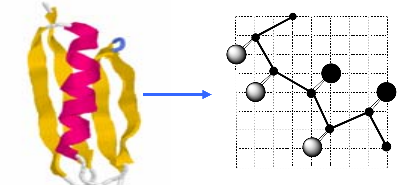
Each lattice point has up to 24 nearest neighbor lattice points

$$\vec{a}_{1-4} = (\pm 2, \pm 1, 0) \quad \vec{a}_{5-8} = (\pm 2, 0, \pm 1)$$

$$\vec{a}_{9-12} = (0, \pm 2, \pm 1)$$ Etc.

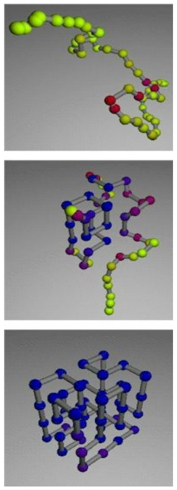


Given a protein fold, and placing it on a 210 lattice, results in a Root Mean Square Deviation (RMSD) of ~2Å: high resolution

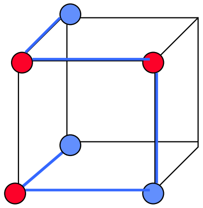


J. Chem. Phys. 119, 3453-3460 (2003)

Protein Lattice Model Interactions



HP models: amino acids on a chain are restricted to two flavors: Hydrophobic (H) and Polar (P)



Example sequence PHHPHP

$$V = \sum_{\substack{\langle i,j \rangle_{pair} \\ |i-j| \neq 1}} H_{ij}$$

Each amino acid bead interacts with only its nearest neighbors,

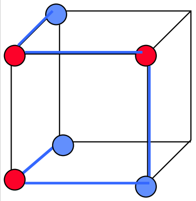
excepting its bonding partner

$$H_{ij} = \begin{array}{ccc} & H & P \\ H & -1 & 0 \\ P & 0 & 0 \end{array}$$

http://www.lbl.gov/Science-Articles/Archive/model-protein-folding2.html

Protein Lattice Model Interactions

Miyazawa-Jernigan (MJ) models: all twenty amino acids



Each amino acid bead interacts with only its nearest neighbors, excepting its bonding partner, but through PES:

$$V=\sum_{\substack{\langle i,j\rangle \\ |i-j|\neq 1}} H_{ij} \qquad H_{ij}=\begin{array}{c|cccccc} & C & T & G & Q & L & \ldots \\ \hline C & P_{CC} & P_{CT} & P_{CG} & P_{CQ} & P_{CL} & \\ T & P_{TC} & P_{TT} & P_{TG} & P_{TQ} & P_{TL} & \\ G & P_{GC} & P_{GT} & P_{GG} & P_{GQ} & P_{GL} & \\ Q & P_{QC} & P_{QT} & P_{QG} & P_{QQ} & P_{QL} & \\ L & P_{LC} & P_{LT} & P_{LG} & P_{LQ} & P_{LL} & \\ \ldots & & & & & & \end{array}$$

P_{xx}=probability of observing a residue-residue contact in the protein databank (PDB). These are known as “statistical potentials”

Bigger Time Steps

Time-Scale of motions bottlenecks (Δt)

Timestep limited by fastest timescale in your system

* bond vibrations: period of 10-14 seconds (10fs): Δt =1fs

*shake/rattle bonds (project out force along bond) Δt =2fs

Scales as N; fast timescales

$$U=\sum_{i}^{\#Bonds} k_b\left(b_i-b_o\right)^2+\sum_{i}^{\#Angles} k_\theta\left(\theta_i-\theta_o\right)^2+\sum_{i}^{\#Impropers} k_\tau\left(\tau_i-\tau_o\right)^2+$$

$$\sum_{i}^{\#dihedrals} k_\phi\left[1+\cos\left(n\phi+\delta\right)\right]+\sum_{i}^{\#atoms}\sum_{i<j}^{\#atoms}\left\{\frac{q_iq_j}{r_{ij}}+\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12}-\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right]\right\}$$

Scales as N^2; slow timescales

multiple timescale algorithms (~4fs to 10fs)

(active area of research)*

*Preserve symplectic,reversible properties!

Better Computers: IBM Blue Gene

Blue Gene will do

(1) Robust objective function

- All atom simulation with molecular water present
- Proper treatment of long-ranged interactions (Ewald)
- Part of the objective is to interrogate energy functions

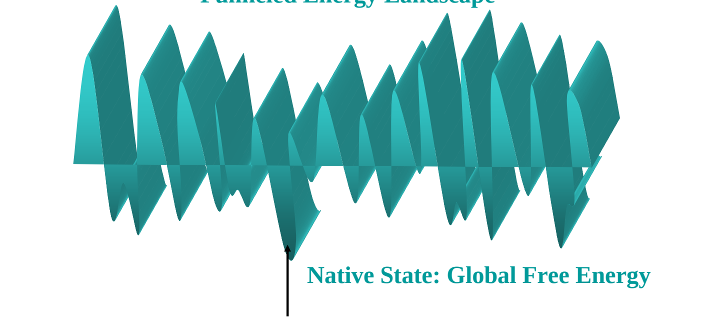
(2) Severe time-scale problem

- 10^9 energy/forces: parallelization (spatial decomposition)
- Blue Gene will simulate on the microsecond-millisecond

(3) Statistics (1 trajectory is anecdotal)

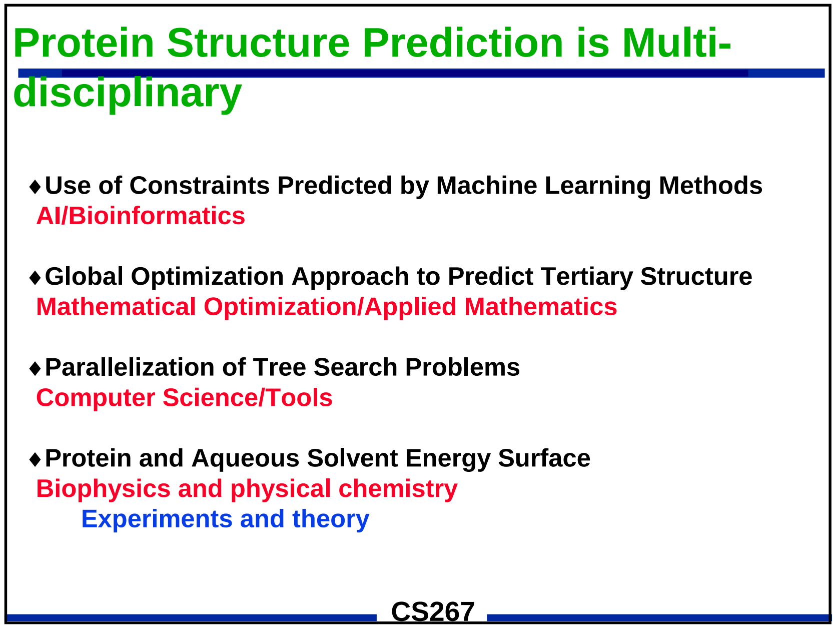
- Blue Gene can do 1000's

Global Optimization: Protein Structure Prediction



Sequence, an objective function, a search method → Tertiary Structure

- ♦Protein and Aqueous Solvent Energy Surface
- ♦Incorporation of Constraints Predicted by Machine Learning Methods
- ♦Global Optimization Approach to Predict Tertiary Structure
- ♦Parallelization of Tree Search Problems



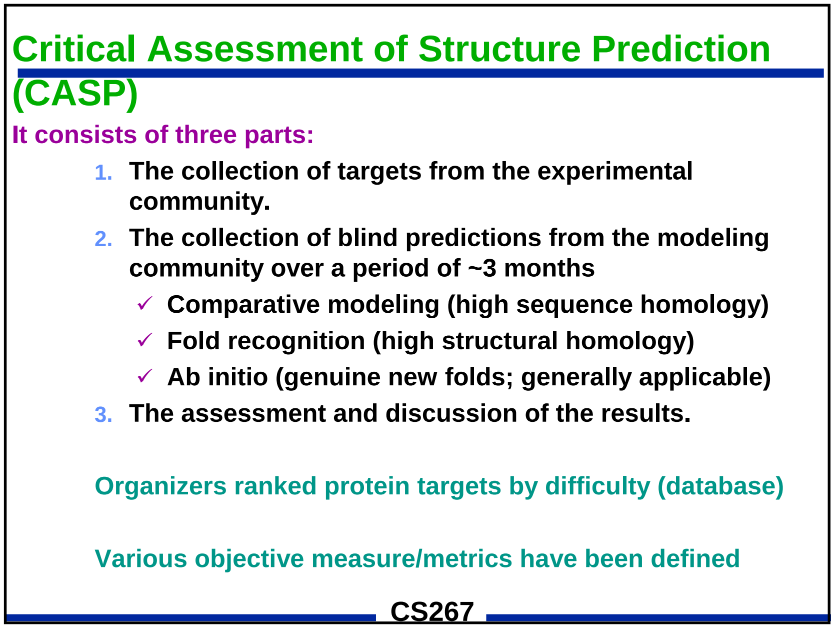
Protein Structure Prediction is Multi-disciplinary

- ♦Use of Constraints Predicted by Machine Learning Methods
  AI/Bioinformatics

- ♦Global Optimization Approach to Predict Tertiary Structure
  Mathematical Optimization/Applied Mathematics

- ♦Parallelization of Tree Search Problems
  Computer Science/Tools

- ♦Protein and Aqueous Solvent Energy Surface
  Biophysics and physical chemistry
  Experiments and theory



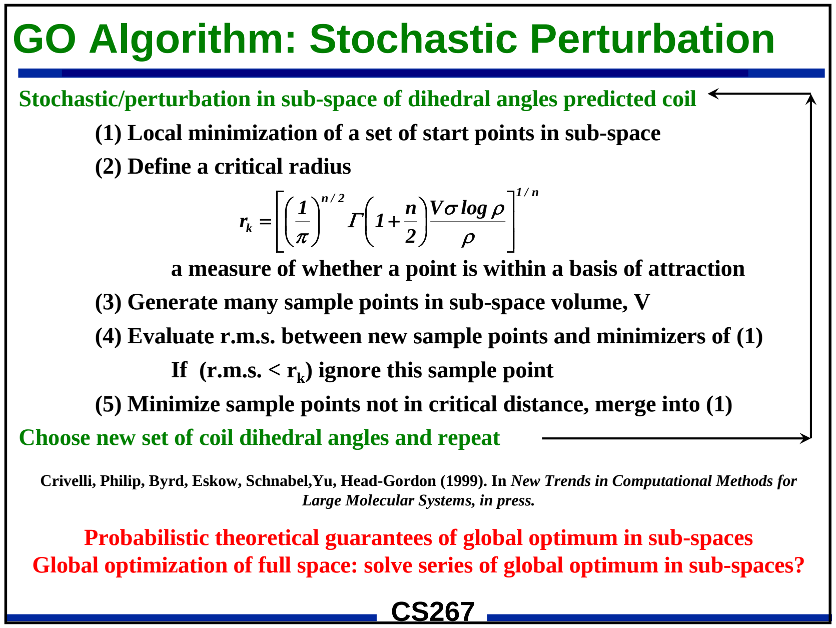
Critical Assessment of Structure Prediction (CASP)

It consists of three parts:

1. The collection of targets from the experimental community.
2. The collection of blind predictions from the modeling community over a period of ~3 months
   - ✓ Comparative modeling (high sequence homology)
   - ✓ Fold recognition (high structural homology)
   - ✓ Ab initio (genuine new folds; generally applicable)
3. The assessment and discussion of the results.

Organizers ranked protein targets by difficulty (database)

Various objective measure/metrics have been defined



GO Algorithm: Stochastic Perturbation

Stochastic/perturbation in sub-space of dihedral angles predicted coil

(1) Local minimization of a set of start points in sub-space

(2) Define a critical radius

$$r_k = \left[\left(\frac{1}{\pi}\right)^{n/2}\Gamma\left(1+\frac{n}{2}\right)\frac{V\sigma\log\rho}{\rho}\right]^{1/n}$$

a measure of whether a point is within a basis of attraction

(3) Generate many sample points in sub-space volume, V

(4) Evaluate r.m.s. between new sample points and minimizers of (1)

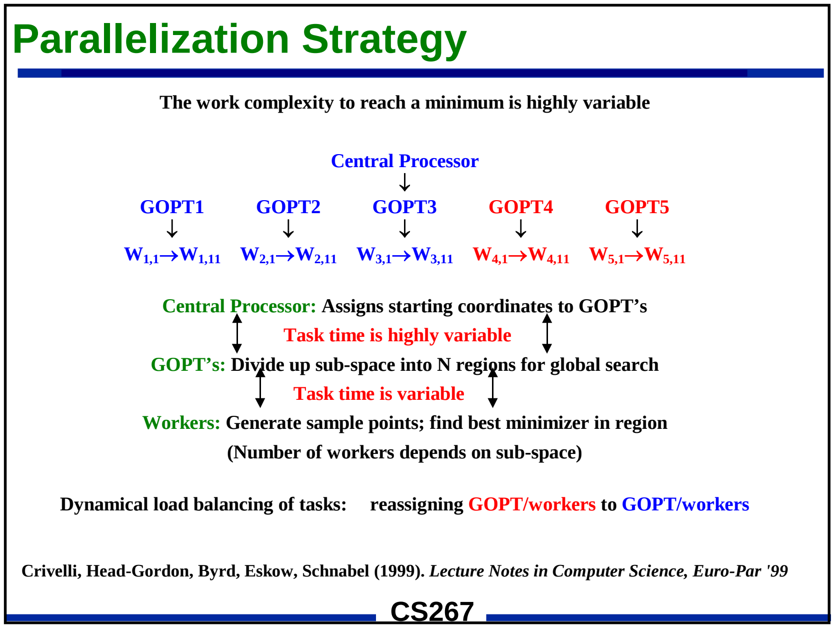
If (r.m.s. < r_k) ignore this sample point

(5) Minimize sample points not in critical distance, merge into (1)

Choose new set of coil dihedral angles and repeat

Crivelli, Philip, Byrd, Eskow, Schnabel,Yu, Head-Gordon (1999). In *New Trends in Computational Methods for Large Molecular Systems, in press.*

Probabilistic theoretical guarantees of global optimum in sub-spaces
Global optimization of full space: solve series of global optimum in sub-spaces?



Parallelization Strategy

The work complexity to reach a minimum is highly variable

Central Processor

↓

GOPT1	GOPT2	GOPT3	GOPT4	GOPT5
↓	↓	↓	↓	↓
$W_{1,1}\rightarrow W_{1,11}$	$W_{2,1}\rightarrow W_{2,11}$	$W_{3,1}\rightarrow W_{3,11}$	$W_{4,1}\rightarrow W_{4,11}$	$W_{5,1}\rightarrow W_{5,11}$

Central Processor: Assigns starting coordinates to GOPT's

Task time is highly variable

GOPT's: Divide up sub-space into N regions for global search

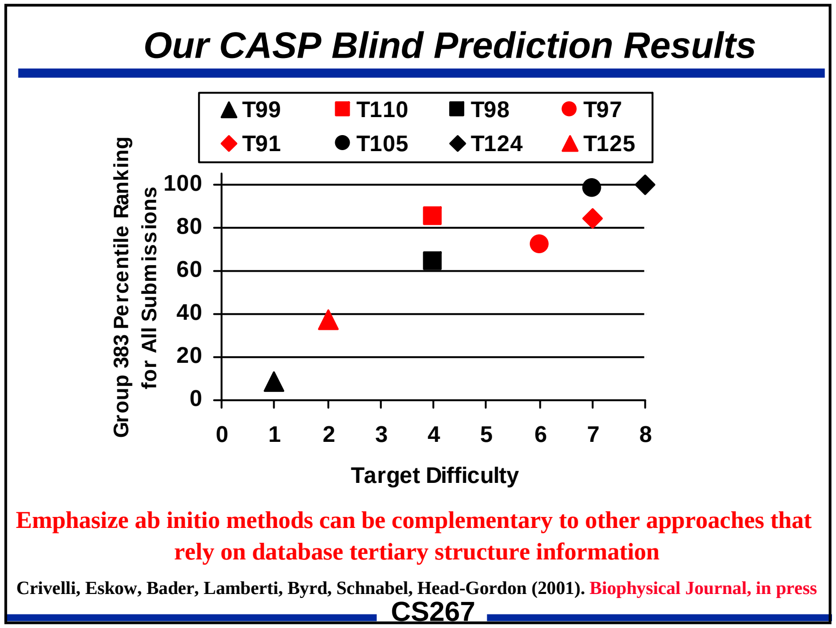
Task time is variable

Workers: Generate sample points; find best minimizer in region

(Number of workers depends on sub-space)

Dynamical load balancing of tasks: reassigning GOPT/workers to GOPT/workers

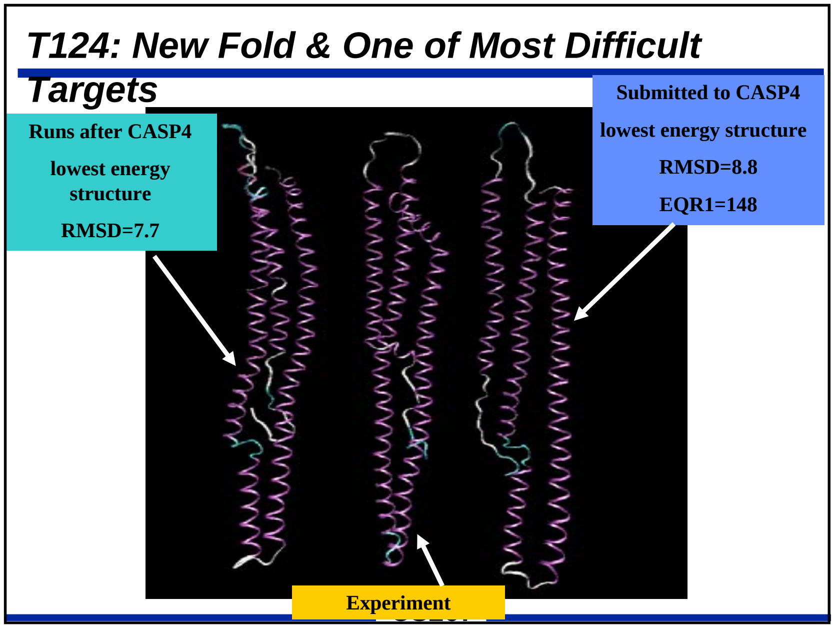
Crivelli, Head-Gordon, Byrd, Eskow, Schnabel (1999). *Lecture Notes in Computer Science, Euro-Par '99*



Our CASP Blind Prediction Results

Emphasize ab initio methods can be complementary to other approaches that rely on database tertiary structure information

Crivelli, Eskow, Bader, Lamberti, Byrd, Schnabel, Head-Gordon (2001). Biophysical Journal, in press



T124: New Fold & One of Most Difficult Targets

Runs after CASP4

lowest energy structure

RMSD=7.7

Submitted to CASP4

lowest energy structure

RMSD=8.8

EQR1=148

Experiment