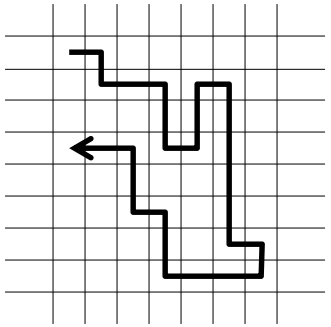
Monte Carlo and MD simulations

Andrew Torda, April 2018 strukt und sim

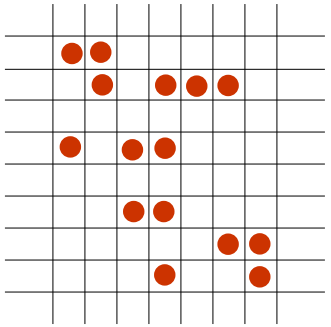
What we observe in any system ?

- averages of observables (pressure, energy, density)

Given enough time system will visit all states



time



random
hopping

My observable $\mathcal{A}$

$$\mathcal{A}_{obs} = \frac{1}{b-a}\int_a^b \mathcal{A}_t dt \qquad \mathcal{A}_{obs} = \frac{1}{N_{obs}}\sum_{i=1}^{N_{obs}} \mathcal{A}_i$$

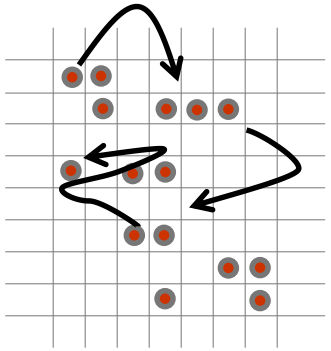
Time and space averages

If we believe $\mathcal{A}_{obs} = \frac{1}{N_{obs}} \sum_{i=1}^{N_{obs}} \mathcal{A}_i$

then

$$\mathcal{A}_{obs} = \sum_{j}^{states} p_j \mathcal{A}_j$$
$$\equiv \langle \mathcal{A} \rangle$$

and p_j is the probability of state j



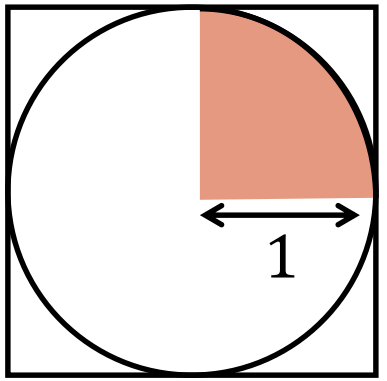
- $\langle \mathcal{A} \rangle$ is ensemble average and usually $\bar{\mathcal{A}}$ is time average
- if sample with correct probability, we can find $\mathcal{A}_{obs}$
- order of visiting states does not matter

Monte Carlo

How to calculate π with random numbers

$$\frac{points_{red}}{points_{square}} = \frac{{}^{1}/_{4}\,\pi r^2}{\text{area in square}}$$

$$\pi = 4\frac{points_{red}}{points_{square}}$$



while (not converged)

pick random x, y

n_{square}++

if $((x^2+y^2) < 1)$

n_{red}++

print $\frac{4\,n_{red}}{n_{square}}$

diagram loosely from Allen, MP and Tildesley, DJ

Generating distributions / Monte Carlo

Generating points in a circle ? (generating function)

$$p_{in\_circle} = \begin{cases} 1 & x^2 + y^2 \leq 1 \\ 0 & x^2 + y^2 > 1 \end{cases}$$

We could work out the area of a circle (integrate) by picking random numbers

What does Monte Carlo simulation mean ?

- generating points according to some distribution to find an average or integral
- what is our distribution in physical systems ?
  - Boltzmann distribution

Monte Carlo and Boltzmann distributions

Boltzmann probability distribution

$$p_i = \frac{e^{\frac{-E_i}{kT}}}{\sum_j e^{\frac{-E_j}{kT}}}$$ often written as $$p_i = \frac{e^{\frac{-E_i}{kT}}}{Z}$$ since we define $Z = \sum_j e^{\frac{-E_j}{kT}}$

If we could generate this distribution,

we could reproduce most properties of a system

Leads to a scheme (not possible)

correct, but not practical scheme

while (not happy)
    generate configuration $\mathbf{r}_i$ (conformation of protein, …)
    calculate p_i (number between 0 and 1)
    generate random number x
    if $(x < p_i)$
        accept $\mathbf{r}_i$
    else
        reject $\mathbf{r}_i$

$$p_i = \frac{e^{\frac{-E_i}{kT}}}{\sum_j e^{\frac{-E_j}{kT}}}$$

- Result ? a set of $\mathbf{r}_i$ with Boltzmann distribution
- Problem ? we do not know $\sum_j e^{\frac{-E_j}{kT}}$

a better scheme

We cannot generate points from $p_i = \frac{e^{\frac{-E_i}{kT}}}{\sum_j e^{\frac{-E_j}{kT}}}$

What if we have two configurations ?

$$\frac{p_i}{p_j} = \frac{e^{\frac{-E_i}{kT}}}{Z} \quad \frac{Z}{e^{\frac{-E_j}{kT}}}$$

$$= e^{\frac{E_j - E_i}{kT}}$$

$$= e^{\frac{-\Delta E}{kT}}$$

a better scheme

$$\frac{p_i}{p_j} = e^{\frac{-\Delta E}{kT}}$$

If we have one configuration to start

- we can work out the relative probability of a second

Convenient convention

- going from old→new $\Delta E < 0$
  - $E_{new} - E_{old} < 0$ energy is better / more negative

Does it matter where you start ? What is i ?

Metropolis Monte Carlo

- generating a distribution $\frac{p_i}{p_j} = e^{\frac{-\Delta E}{kT}}$
- if $\Delta E < 0$, new is likely (more than 1)
- if $\Delta E > 0$, old is p_{new} is possible

```
generate starting configuration r_o
while (not happy)
        generate r_new
        calculate E_new and ΔE
        if ΔE < 0
                set r_o to r_new
        else
                x = rand [0:1]
                if(x ≤ e^(−ΔE/kT))
                        set r_o to r_new
```

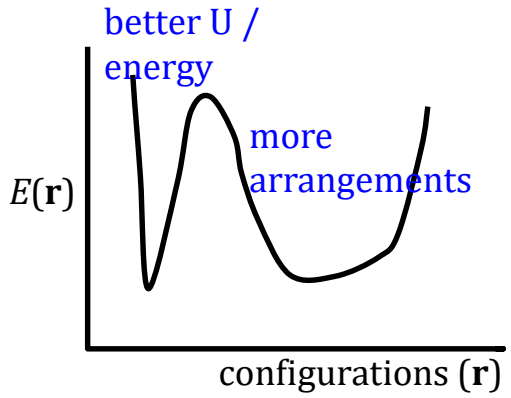
- what if ΔE slightly > 0 ?
  - 0.0000000001
- what if $\Delta E = 10^6$?
- small uphill moves are OK
- bigger moves are less likely

Properties of Monte Carlo

The set of $\mathbf{r}_o$ is a valid distribution (ensemble)

- for some property $\mathcal{A}$

$$\mathcal{A}_{obs} = \langle \mathcal{A} \rangle = \frac{1}{N_{visited}} \sum_{i}^{N_{visited}} \mathcal{A}_i$$



- $\mathcal{A}$ could be density, structural property, E, …
- only works for one temperature T

Look at picture.. could I calculate entropy / free energy ?

- for simple systems

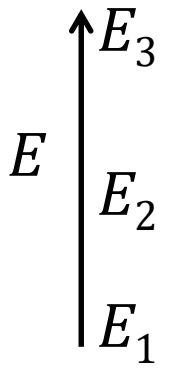
Equilibrium

MC results (observables / averages)

- only for system at equilibrium
- simulations generate system at equilibrium

What happens for a system out of equilibrium ?

- Toy system with 3 states
- for some *T,* at equilibrium
- $p_1 = {}^5/_8 \qquad p_2 = {}^1/_4 \qquad p_3 = {}^1/_8$
- if I have 80 copies of the system, most are in state$_1$



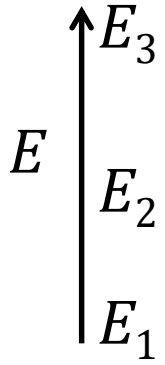
Reaching equilibrium

System wants $p_1 = {}^5/_8$ $p_2 = {}^1/_4$ $p_3 = {}^1/_8$

$50 : 20 : 10$

- start it with $5 : 70 : 5$

- all moves 2→1 are accepted (large flux)
- the flux from $1 \rightarrow 2$
  - $1 \rightarrow 2$ moves are not always accepted
  - there are less particles in state$_1$



Moving to equilibrium depends on

- population
- probability

Detailed balance

For any two states ($state_i$ and $state_j$)

Flow $i \rightarrow j$ must equal $j \rightarrow i$

- otherwise ?

Flow $i \rightarrow j$ depends on

- population N_i
- probability $\pi(i \rightarrow j)$

Detailed balance

$$N_i \pi(i \rightarrow j) = N_j \pi(j \rightarrow i)$$

- detailed balance must apply for any pair i, j

all textbooks use π for probability here

Ergodic

Assumptions

- I can do integrals because
  - I will visit every state
  - I can calculate p_i for all states
- I will visit every state

alternatively

For any i, j

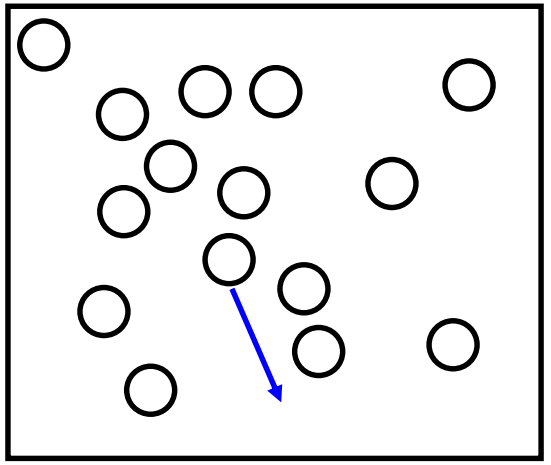
- $\pi(\, i \to j) > 0$
- may require a finite number of steps: $i \to k \to m \to j$
- must be satisfied

Moves

version 1

- decide on r_{max}
- pick a particle at random
- pick random $\Delta x, \Delta y, \Delta z$

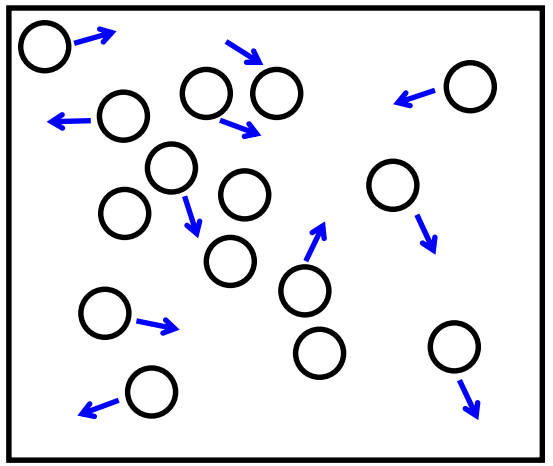
  $0 < \Delta\, a < \ r_{max}$
- apply move
- accept / reject move



version 2

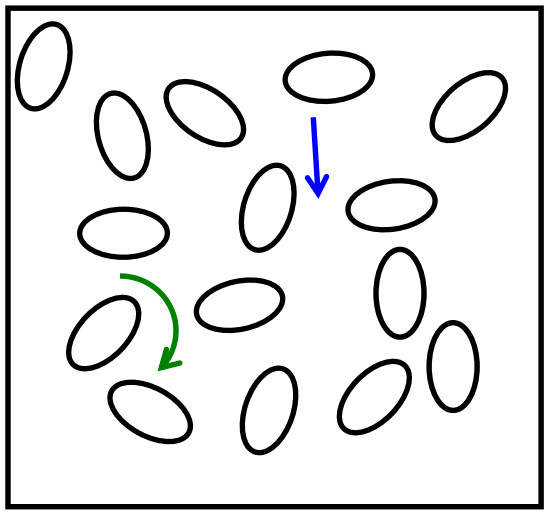
- decide on smaller r_{max}
- foreach particle
  - pick random $\Delta x, \Delta y, \Delta z$

    $0 < \Delta\, a < r_{max}$
- apply move
- accept / reject



Moves

- both kinds of move OK
- note
  - "accept / reject"



More generally,

- how big is r_{max} ?
- big
  - system moves faster
  - more moves rejected

What if my particles are not spheres ?

- rotations also necessary

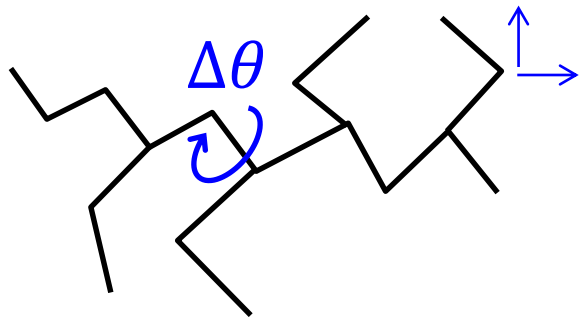
- time has no meaning

Bonded systems

Protein (lipid, polymer, ..)

Random Δx ?

- nearly all will stretch a bond
  - high energy : rejected move
- only feasible method
  - random rotations $\Delta\theta$



In general

- most kinds of simple moves OK
- must maintain detailed balance, ergodicity
- question of efficiency
  - high rejection rate means lots of wasted calculations

More moves – *N* particles

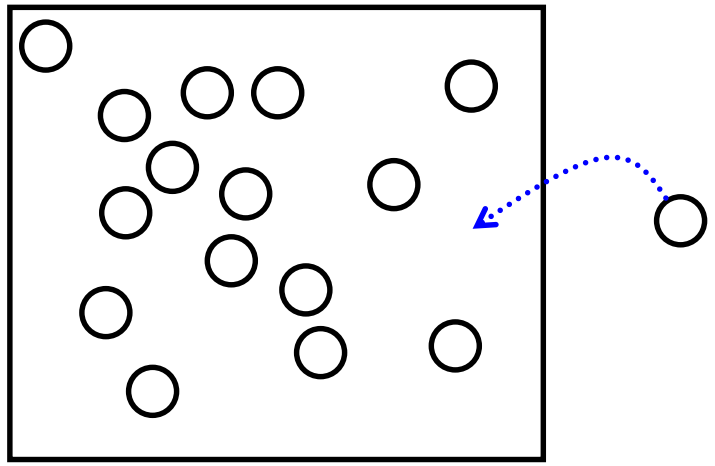
$$\frac{p_{new}}{p_{old}} = e^{-\Delta E/kT}$$

I have defined temperature

- and $N_{particles}$ and V
- called NVT simulation

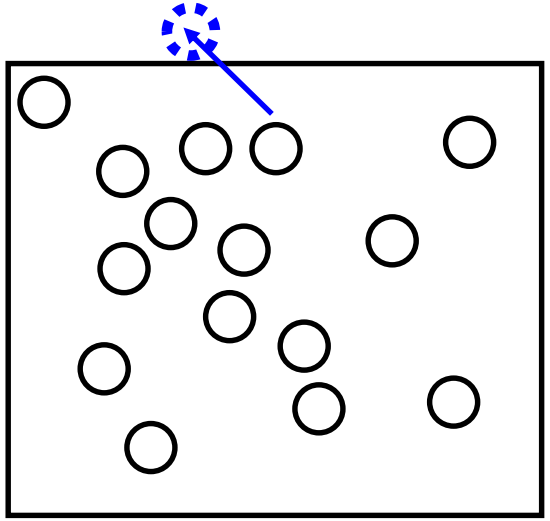
Could I have varied something else ?

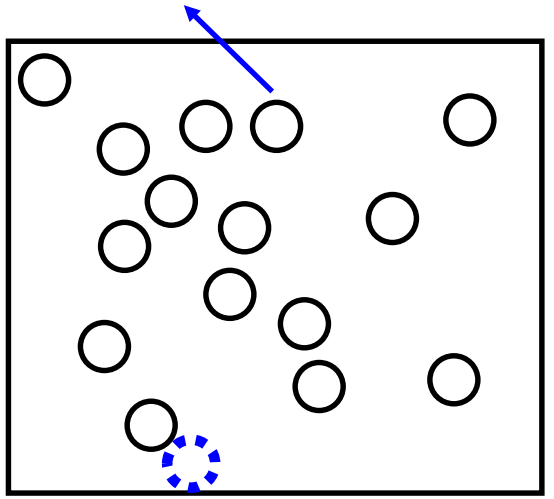
- what if I tried to put particles in / take out ?
  - sometimes energy ↑sometimes↓
- system will fluctuate around $\langle N \rangle$
- this would not be NVT



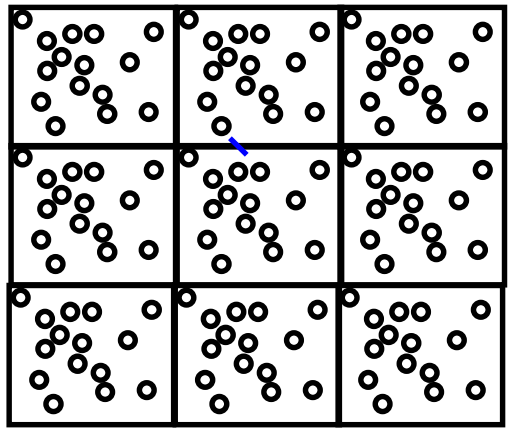
periodic boundary conditions

Relevant to gases, proteins in water, ..





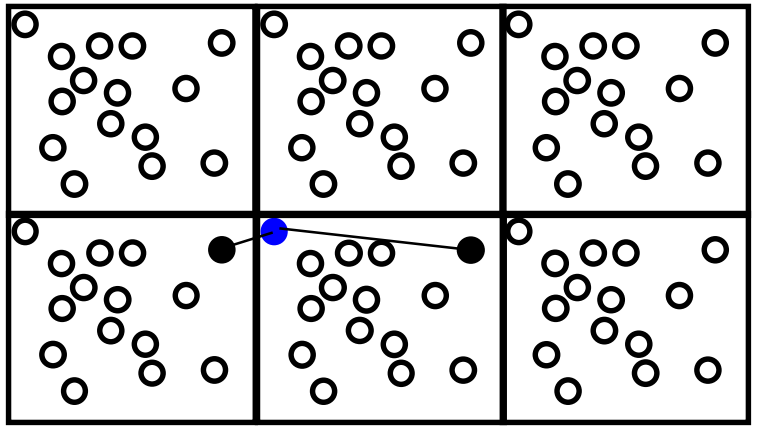
Behaves like an infinite system



Infinite interactions ?

Neighbours of blue particle

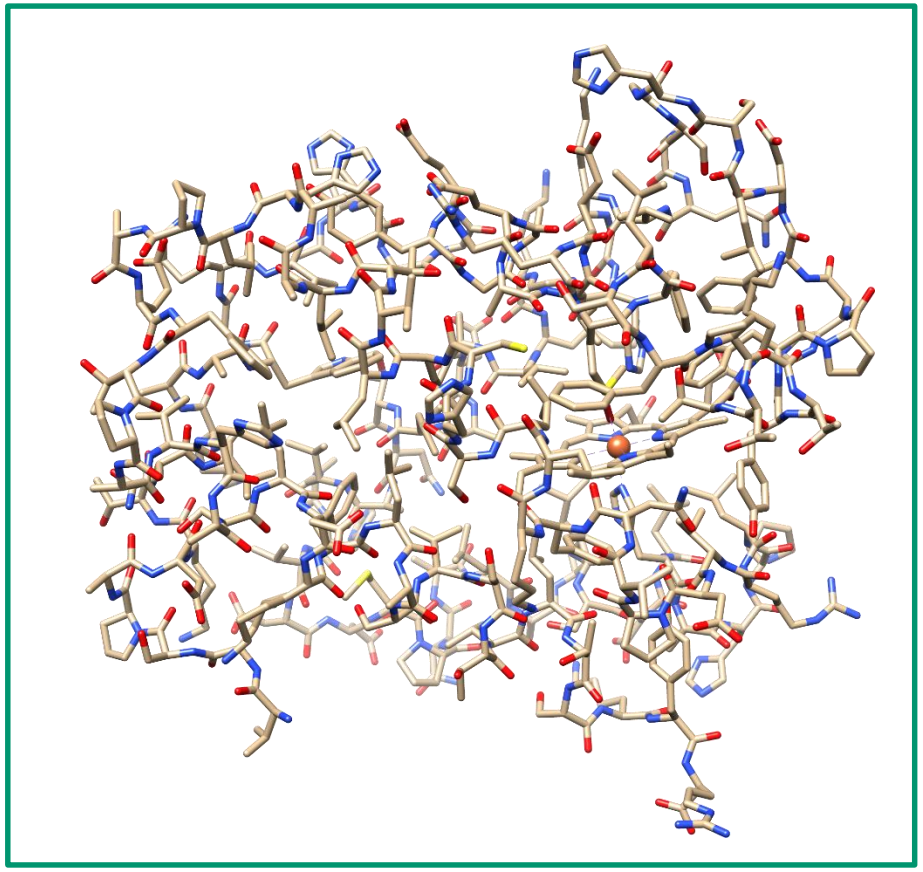
- only use the nearer
- not really an infinite system



- volume defined by box

- how many neighbours does blue particle have ?
- why do we need cutoffs ?

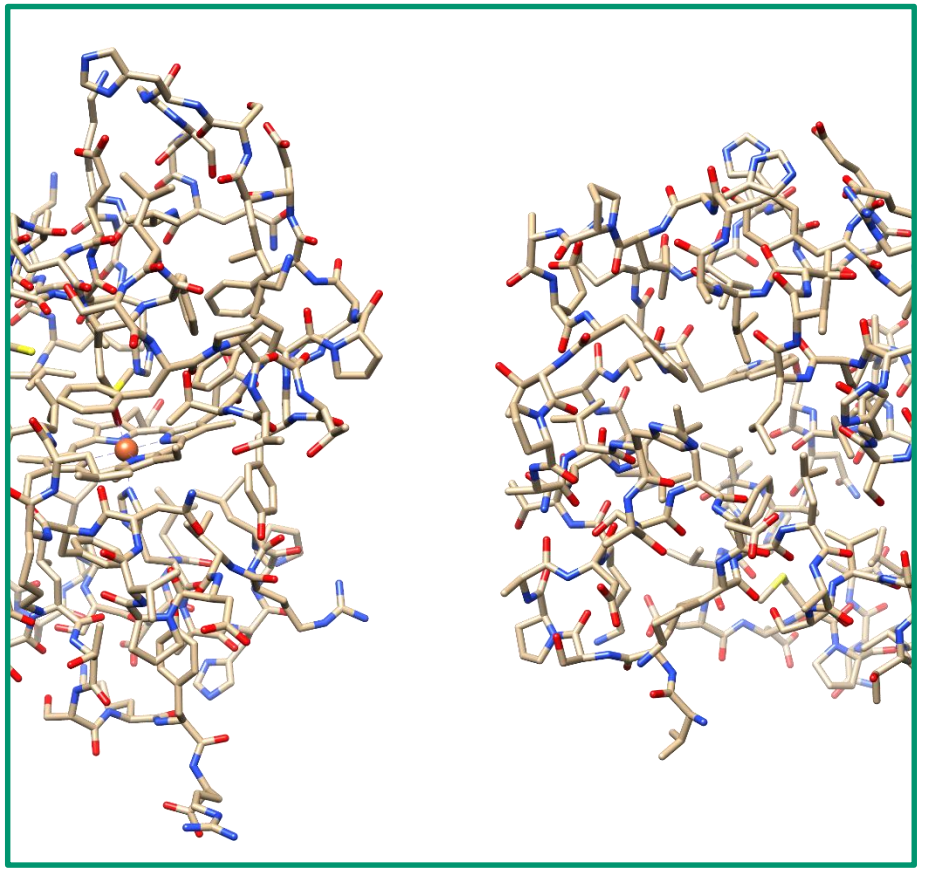
protein chopped up by periodic boundary conditions



start
coordinates

simulation

diffusion



later

Problems with Monte Carlo

```
while (not happy)
        propose move
        accept / reject move
```

Small steps ?

- system moves slowly: long time to visit all states

Big steps ?

- calculate energy
- reject move
  - no progress, wastes time

Dense Systems and Monte Carlo

Random moves ?

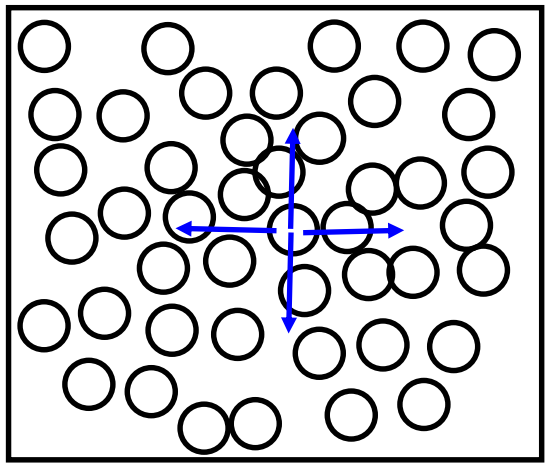
- most moves rejected

Dense systems ?

- liquids
- proteins, polymers, …

Solutions

- cleverer MC moves (later)
- MD



Why do molecular dynamics simulations ?

Real world

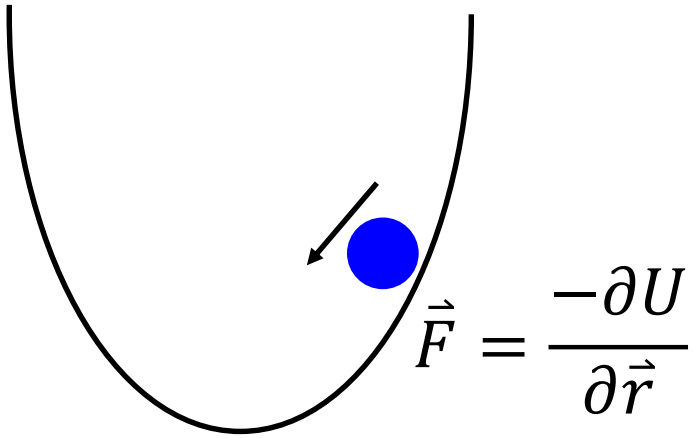
- box of gas, molecule in space, protein molecule in water
- atoms hit each other,
  - share energy, box expands/contracts, ..
  - soon reaches equilibrium
  - visits low energies (often), high energies (less often)
  - visits entropically favoured regions
- we stick in a thermometer
- measure density, …

What have the atoms done ?

- feel forces and move
- an MD simulation just copies this

What do we expect ? Molecular Dynamics

one particle in a well



$$\vec{F} = \frac{-\partial U}{\partial \vec{r}}$$

Unlike MC, particles have kinetic energy E_{kin}

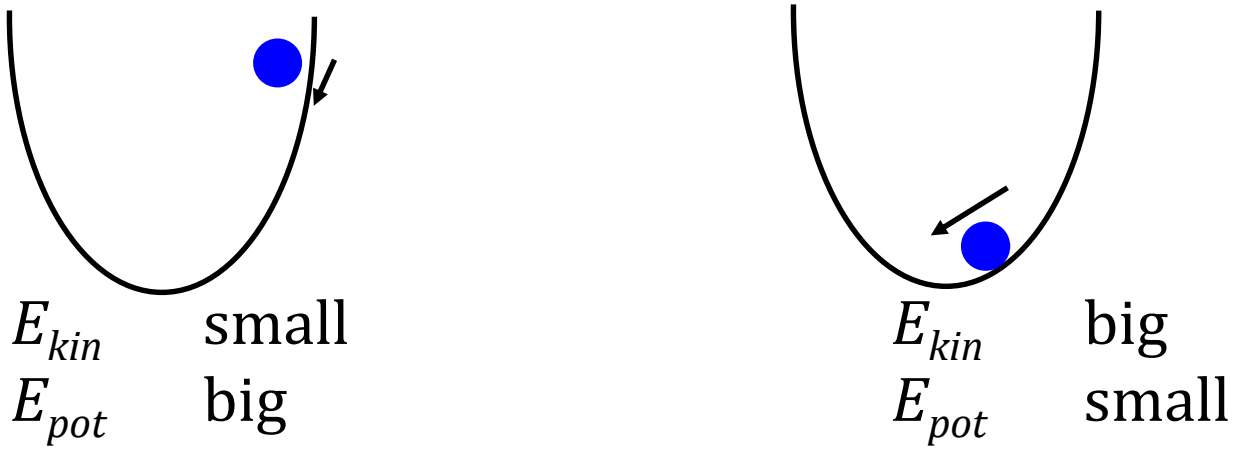
Kinetic and potential energy

Our system is isolated (no work done)

E_{tot} never changes

- conserves energy (no work done on system)

$$E_{tot} = E_{pot} + E_{kin}$$

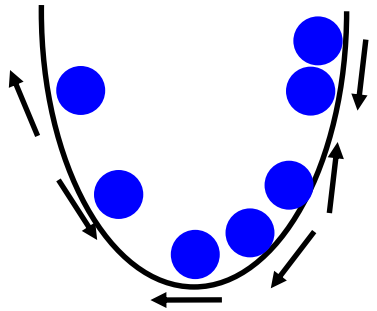


For one particle $E_{tot} = E_{pot} + E_{kin}$ = constant

Lots of particles

Particles hitting each other

- exchanging energy



Total system

- conserves energy

One particle ?

- maybe at bottom but moving slow (E_{kin} + E_{pot} small)
- per particle energy no longer conserved (may gain or lose)

Many particles

- distribution of velocities
- distribution of potential energies

Boltzmann distribution in real world

One version of real world (N, V, T)

- constant number of particles, volume, temperature
- today $E = E_{kin} + E_{pot}$
- Z is partition function
- earlier $Z = \sum_i e^{\frac{-\Delta E_i}{kT}}$

But now we have kinetic energy $E_{kin}(\mathbf{p})$

- where $\mathbf{p} = m\dot{\mathbf{x}}$
  - potential energy $E_{pot}(\mathbf{r})$
- if we write in continuous form …

Partition function for MD

Usually write $\mathcal{H}(\mathbf{p}, \mathbf{r}) = E_{kin}(\mathbf{p}) + E_{pot}(\mathbf{r})$

- "Hamiltonian"

All the states are defined by all possible momenta and coordinates

- sum over these: $Z(N, V, T) \propto \int d\mathbf{p} \int d\mathbf{r}\, e^{\frac{-\mathcal{H}(p,r)}{kT}}$

often see $\mathrm{H}(\mathbf{p}, \mathbf{r})$ or $\mathcal{H}(\mathbf{\Gamma})$

MD Method

For any particle we can calculate forces

Newtons law

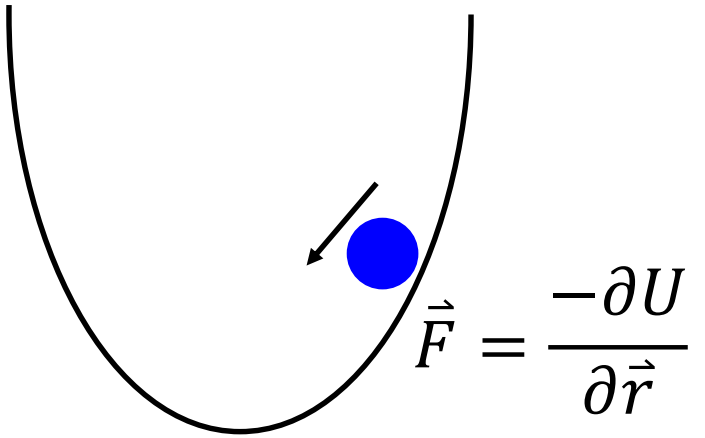
$F = ma$ often better written $\ddot{\vec{x}} = \vec{F}m^{-1}$

If we know acceleration

- we can get velocity

from velocity

- can get coordinates




```
while (nstep < max_step)
      calculate forces
      integrate to get new coordinates
      nstep ++
```

averaging,
sampling,
...

starting a system

Initial coordinates

- protein model
- protein from protein data bank (PDB)
- protein + proposed ligand
- box of liquid

Do initial coordinates matter ?

- in principle: no

  infinitely long simulation visits all configurations, reaches equilibrium

- in practice: yes
  - bad examples
    - no simulation is long enough to predict protein conformation
  - take water configuration and run at ice temperature

Initial velocities

First consider temperature – reflects kinetic energy

$$\left\langle \frac{1}{2} m v_\alpha^2 \right\rangle = \frac{1}{2} kT$$

where v_α^2 could be v_x, v_y, v_z

leads to definition

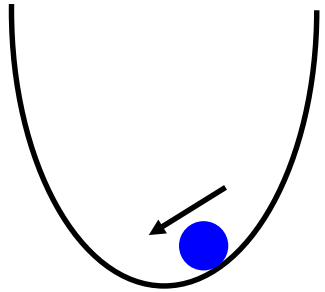
$$T(t) = \sum_{i=1}^{N} \frac{m_i v_i^2(t)}{k N_f}$$

- where N_f is number degrees of freedom $\approx 3N$
- we could use this to get initial velocities $\langle v_\alpha^2 \rangle = \frac{kT}{m}$

Initial velocities

Would one $<v^2>$ be OK ?

- not very good
  - E_{kin} correlated with E_{pot}



Either

- use more sophisticated distribution
- do not worry
  - system will go to equilibrium
    - velocities will reach sensible values

Getting new velocities / coordinates

constant acceleration

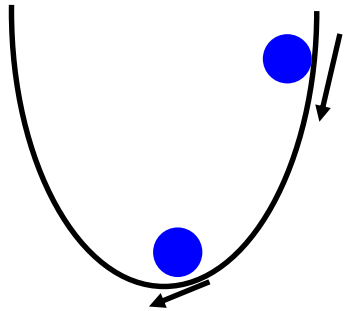
$$x_t = x_0 + vt + \frac{1}{2}at^2$$

or

$$x_t = x_0 + \dot{x}t + \frac{1}{2}\ddot{x}t^2$$

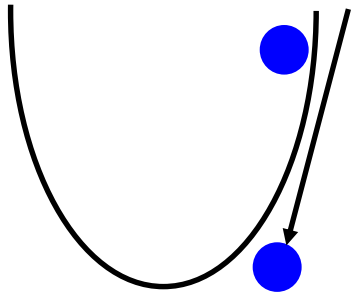
OK for constant acceleration

- try to use formula to predict future time

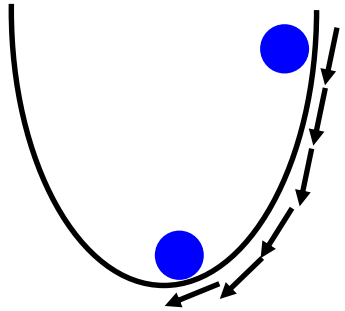


$\ddot{x}$ big

$\ddot{x}$ small



big Δt / step
big error



small Δt / step
small error
slow

Fundamental problem with integration

- We want to use big Δt (speed)
- We must use small Δt (accuracy)

All Δt will give us some error

- numerical integration is never perfect

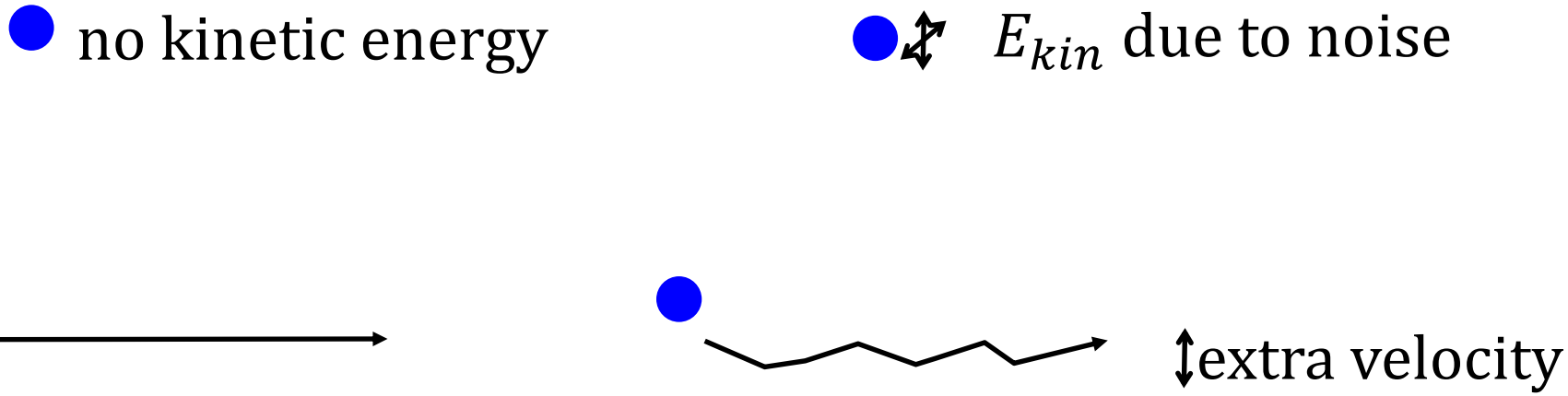
How small is Δt ?

- depends on fastest frequency / steepest walls in energy
  - usually bonds
- for proteins at room temperature
  - $\Delta t \approx 1$ fs (femtosecond 10^{-15} s)
- high temperature Δt should be smaller

Noise and heating

General rule

- noise heats the system
- formally difficult to prove
- $E_{kin} = \frac{1}{2}mv^2$



Noise-free Simulation

Energy conservation : Absolute rule $E_{pot} = f(\mathbf{r})$

- no time component
- invariant under translation, rotation

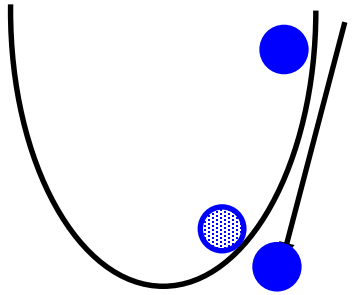
When violated ?

- $(\mathbf{r})$ does not change, but E_{pot} changes: E_{tot} changes

Noise Sources

Integrator

- coordinates do not match velocity
  E_{kin} wrong: $\left(E_{kin} + E_{pot}\right) \neq$ constant
- energy not conserved

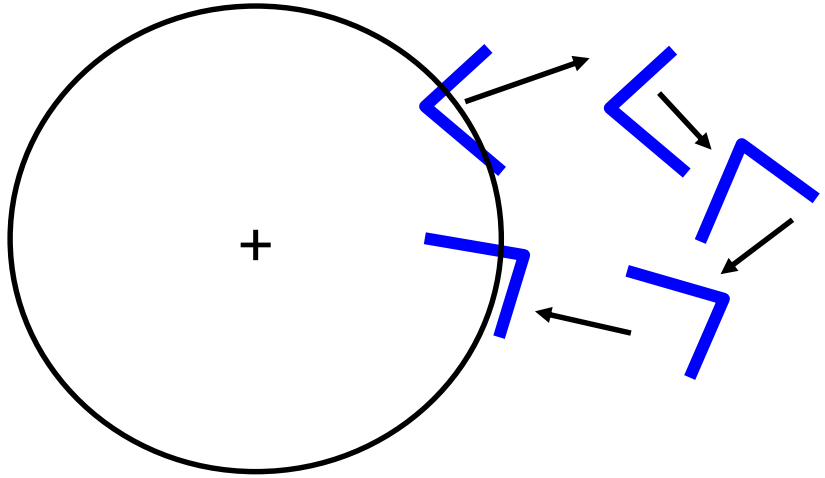


Numerical noise

- $E_{pot} = f(\mathbf{r})$
- initial coordinates ($\mathbf{r}$) quoted to 3 decimal places

Cutoffs

- within cutoff rotation restricted
- outside cutoff rotation suddenly free



Result

- heating

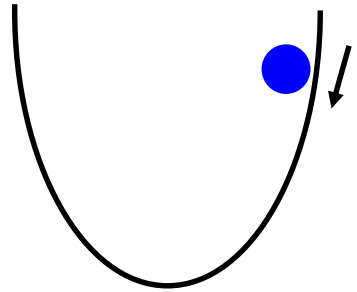
Equilibrium

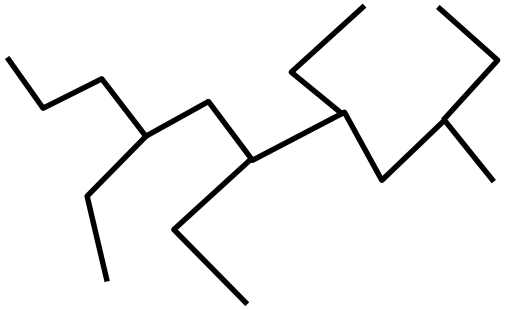
Remember MC story

- system not at equilibrium ? eventually equilibrates

MD

- start in high energy E_{pot}
- E_{pot} converted to E_{kin}





Some high energy conformation

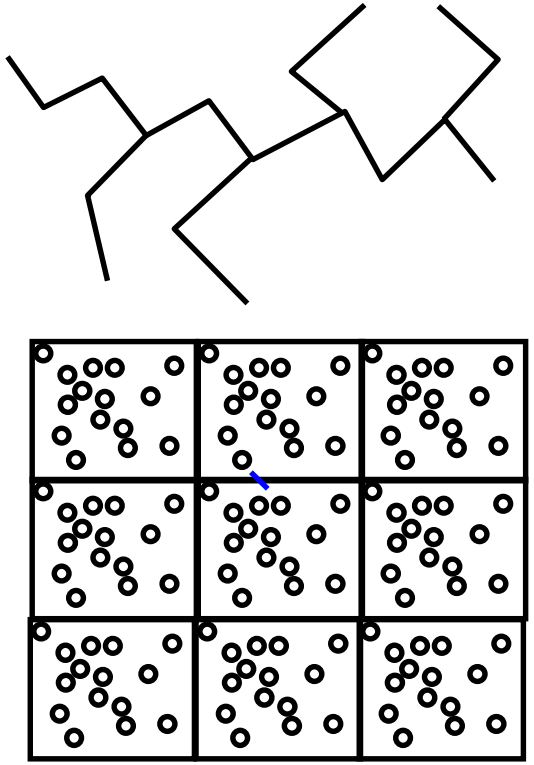
- relaxes
- E_{pot} converted to E_{kin}

MD system will not

- really find low energy
- known temperature

MD in a closed system

- An isolated molecule should not lose energy
- A repeated box will not lose energy
- Formally system is
  - NVE (constant $N_{particles}$, volume, energy)



Problems

- we want to set the temperature of the system
- we may have noise / heat creating energy

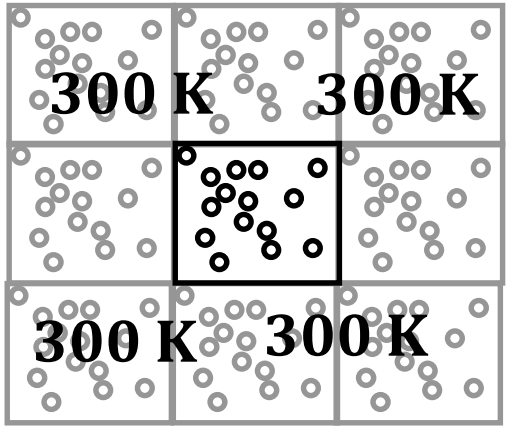
Cure

- thermostat

Bath

Imagine infinite bath at desired temperature

- heat will flow in or out
- at equilibrium no flow of heat
  - maybe removal of noise/heat



How to implement ? Many ways

Occasionally:

1. introduce a fake particle desired temperature / collide
2. pick a particle at random / give average *v* for temperature
3. Easy method –weak coupling…

Weak Coupling

Remember temperature* $E_{kin} = \sum_i^N \frac{1}{2} m_i v_i^2 = \frac{3}{2} NkT$

Goal: heat leaves system depending on how wrong temperature is

$$\frac{dT(t)}{dt} = \frac{T_0 - T(t)}{\tau_T}$$

- T_0 is reference temperature
- τ_t is a coupling / relaxation constant
  - τ_t tiny, heat moves fast. τ_t big, …
- to implement this idea ? Multiply velocities

*Slight simplification of formula

Classic reference: Berendsen, HJC, Postma, JPM, van Gunsteren, WF, DiNola, A, Haak, JR, "Molecular dynamics with coupling to an external bath", J. Chem Phys, 81, 3684, (1984)

Implementation of weak coupling

Scale velocities, $v_{new} = \lambda\, v_{old}$ and $\lambda = \left(1 + \frac{\Delta t}{\tau_T}\left(\frac{T_0}{T} - 1\right)\right)^{1/2}$

Intuitively

- Δt (time step) big ? temperature will change more
- what if $T_0 = T$?
- square root ?
  - wrong T reflects a difference in v^2

Importance of heat baths

Does not conserve energy

In principle

- bring a system to equilibrium for temperature

In practice

- avoid damage due to numerical errors / approximations

For a system at equilibrium

- heat bath should do nothing

Does allow artificial tricks

- gently heat a system and watch behaviour
- gently cool a system and "anneal" it (more later)

Extension to other properties

- analogous reasoning for pressure bath

comparison of Monte Carlo and Dynamics Simulations

MC	MD
any cost/energy OK	requires continuous $E_{pot}(\boldsymbol{r})$
time usually no meaning	gives time scales
most moves OK	physical trajectories
temperature from acceptance/rejection	has explicit E_{kin} and temperature bath
easy to program	difficult
both yield a Boltzmann distribution	
both include entropy	

Applications MC / MD

Basic tools

- Force field
- MD / MC

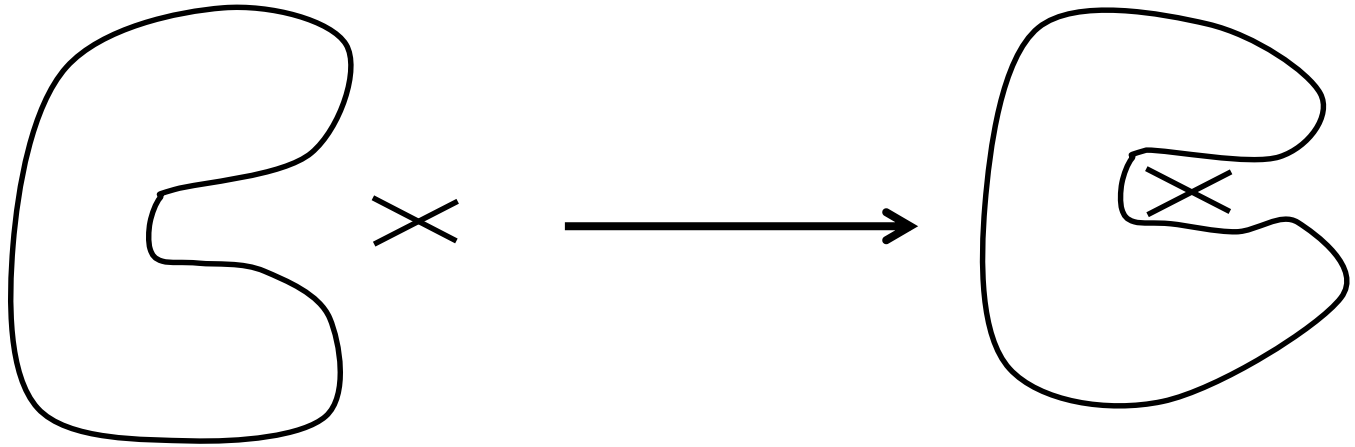
Some application areas

- timescales
- free energy calculations
- simulated annealing
- structure refinement

Simulating dynamics (optimistic / naïve)

Claim

- protein has a hinge which must open to bind ligand



Can one see rates ?

- rates for different ligands ?

Timescales

Most common quantity τ

- time to rotate by 1 radian
- time for decay in $A(t) = A(0)e^{\frac{-t}{\tau}}$
  - relaxation time
  - characteristic time

- times in proteins…

Typical times in proteins

	Amplitude (Å)	$\log_{10} \tau$(s)
bond vibration	0.01 – 0.1	–14 to –13
rotation of surface sidechain	5 – 10	–11 to –10
protein hinge bending	1 – 20	–11 to –7
rotation of sidechain in middle of a protein	5	–4 to 0
local loss of protein structure	5 – 10	–5 to +1

Numbers taken from McCammon, J.A. and Harvey, S.C., Dynamics of proteins and nucleic acids, Cambridge Uni Press, 1987

Timescales, simulations, statistics

Typical big simulation ≈ 100 ns = 10^{-7}s

- Imagine event with characteristic time 10^{-7}s - may or may not be seen

Consider time 10^{-8} s

- may be seen a few times

What you would like - 100's or 1000's of observations

fast events	$\tau \ll t_{simulation}$	OK
	$\tau < t_{simulation}$	poor statistics
slower events	$\tau \approx t_{simulation}$	no idea / very bad statistics

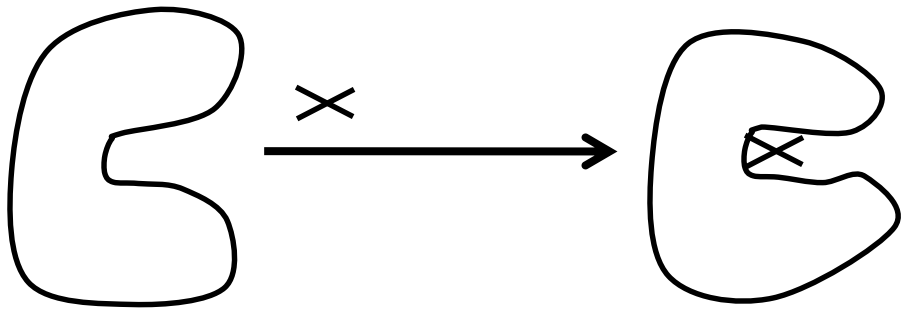
Previous example (drug binding)

- it is not enough to observe an event once (or few times)

Free Energy Calculations

$$k_d = \frac{[\text{drug}][\text{protein}]}{[\text{drug}-\text{protein}]} = \frac{[\text{D}][\text{P}]}{[\text{DP}]}$$

$$= e^{\frac{-\Delta G}{RT}}$$



Contributing terms ?

- ligand-water → ligand + water (many interactions, ΔS)
- ligand+protein
- ligand loss of entropy / water entropy change
  - simulate ?

Infinite time – free energy estimate

$$\mathrm{DP} \rightleftharpoons \mathrm{D} + \mathrm{P}$$

$$\Delta G = kT \ln \frac{[\mathrm{D}][\mathrm{P}]}{[\mathrm{DP}]}$$

Very simple - simulate for long time

- Ligand (drug) goes on and off protein
- Look at [D], [P] and [DP] - calculate ΔG directly from concentrations

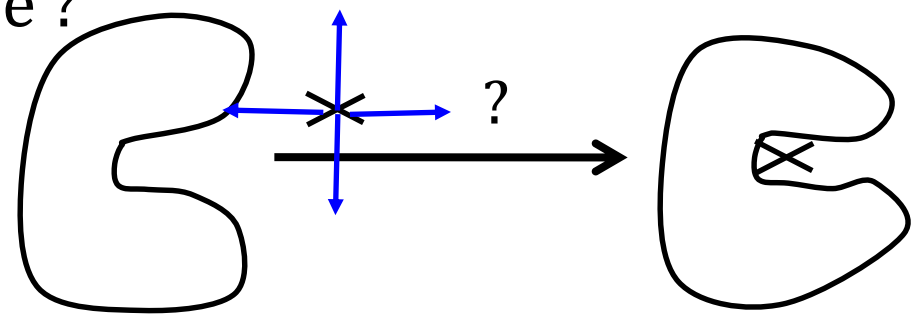
Will not work – cannot simulate long enough

Coming philosophy

- $\mathrm{DP} \rightleftharpoons \mathrm{D} + \mathrm{P}$ is too hard, find an alternative

Free simulation for binding

If we simulate, where will the ligand go ?
What is the shape of the energy landscape ?



May take years for ligand to find protein

Short cut ?

- force ligand to protein
  - artificial force + corrections
  - very difficult – still requires rearranging water
  - entropy estimation very difficult

Estimating free energy differences

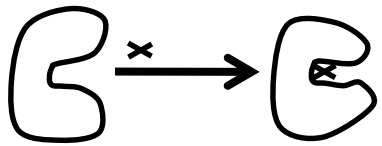
$G = U - TS$

but $S = -k \sum_{i=1}^{N_{state}} p_i \ln p_i$

- so we cannot really get S
- similar problem – especially visiting high energy regions

Forget absolute free energies

- concentrate on ΔG
- no problem – usually interesting property



Summarise free energy problem so far

- Sounds easy, just estimate [D], [P], [DP] – will not work – no simulation long enough
- Cheat – push ligand in ? System not at equilibrium, requires work
- Chemically difficult – lots of interations
  - requires completely changing water configuration
  - breaking ligand-water interactions, finding the correct ligand-protein binding
  - big change in solvent entropy, ligand entropy, protein entropy

How can one minimise the problems ?

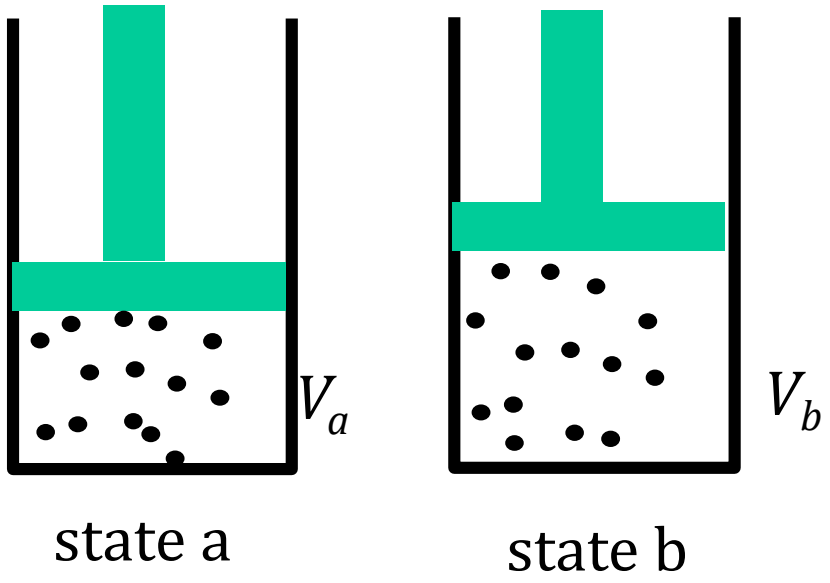
- do an easier problem (soon)

First - small detour on work

Work and free energy changes

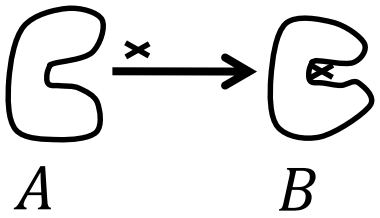
work done A to B

- free energy change
  - automatically includes entropy
    - go in either direction

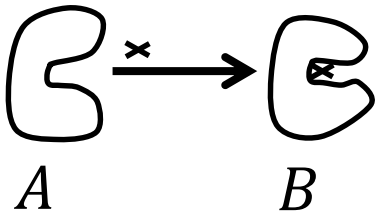


Work going from unbound →bound

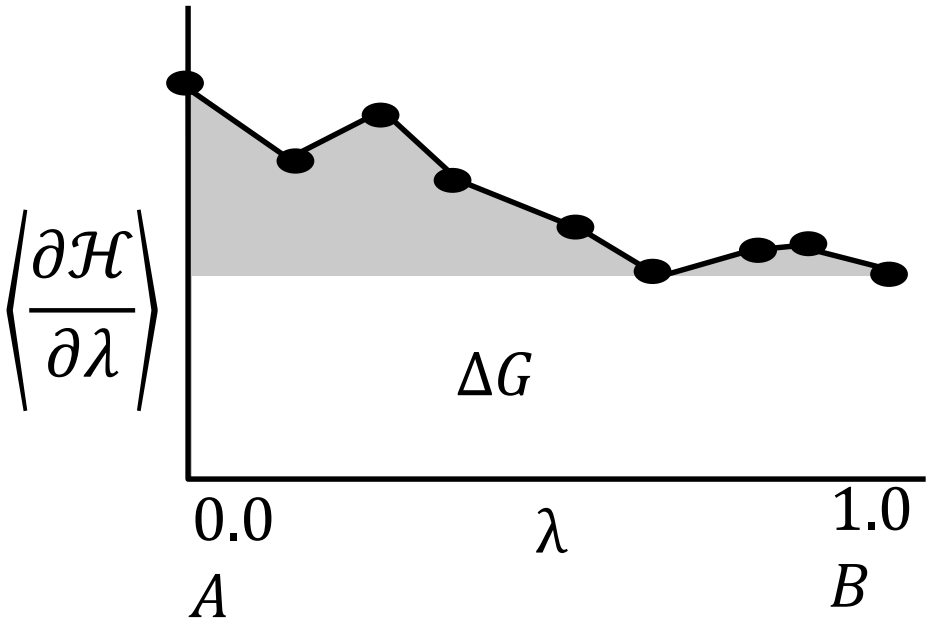
- ΔG_{AB}
- what is B ? what is A ?
  - more later
- measuring work ?



Work and free energy



Measure the work needed to move from A to B



$\mathcal{H}$ is Hamiltonian $(E_{kin} + E_{pot})$

$$\Delta G = \int_A^B \left\langle \frac{\partial \mathcal{H}(\mathbf{p},\mathbf{r})}{\partial \lambda} \right\rangle_\lambda d\lambda \quad \text{or} \quad \Delta G = \sum_{i=0}^{N_{step}} (H_{i+1} - H_i)$$

Binding energy - feasibility

Would this approach work ?

$\left\langle {}^{\partial \mathcal{H}}/_{\partial \lambda} \right\rangle$ must be a good average (lots of fluctuations)

must change λ slowly

Chemistry problems: your simulation would

- get averages with all water molecules
- gradually remove water molecules (high energy ?)
- find the correct binding
- get good averaging there

- states A and B are very different they must be well sampled
- intermediate (higher energy states) must also be sampled
- does not work well in practice

Paths / Energy differences (detour)

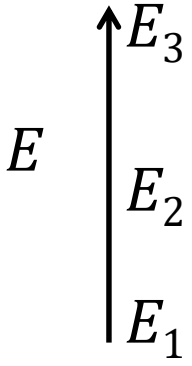
Problem – the path is too difficult – changes too big

- Energy differences depend on end states – not paths
- Look at $\Delta E_{1,2} = E_1 - E_2$
  - would it matter if we go $E_1 \rightarrow E_3 \rightarrow E_2$?

Can we take even stranger paths ?

- go through non existent E_4 ?
  - no problem

Same reasoning applies to free energies



Applying different paths

Originally wanted (ligand A or B, protein P)

A + P ↔ AP $\quad\quad \Delta G_{\mathrm{A}}$

B + P ↔ BP ? $\quad\quad \Delta G_{\mathrm{B}}$

If I know ΔG_{B}

$\Delta\Delta G_{AB}$ is easier

$\Delta\Delta G_{AB} = \Delta G_A - \Delta G_B$

$$\mathrm{A + P} \xrightarrow{\Delta G_{\mathrm{A}}} \mathrm{AP}$$

What would $\Delta\Delta G_{AB}$ mean ?

- relative binding strength

$$\mathrm{B + P} \xrightarrow{\Delta G_{\mathrm{B}}} \mathrm{BP}$$

Alternative routes

$\Delta\ G_A$ and $\Delta\ G_B$ too hard

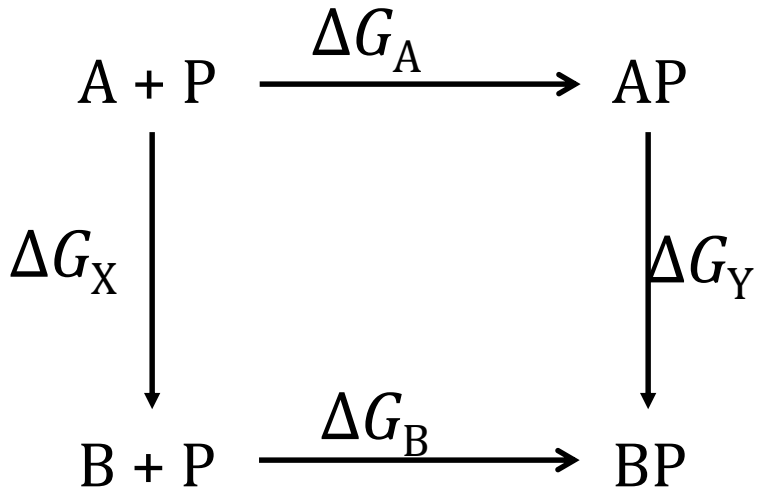
- we would be happy with $\Delta\ \Delta G_{AB}$

$$\Delta G_A + \Delta G_Y = \Delta G_B + \Delta G_X$$

$$\Delta G_A - \Delta G_B = \Delta G_X - \Delta G_Y$$ remember $\Delta\Delta\ G_{AB} = \Delta G_A - \Delta G_B$

So $\Delta\Delta G_{AB} = \Delta\Delta G_{XY}$

- why ΔG_X easier ?
- why ΔG_Y easier ?



Easier free energy changes

if A/B are rather similar

AP ↔ BP or

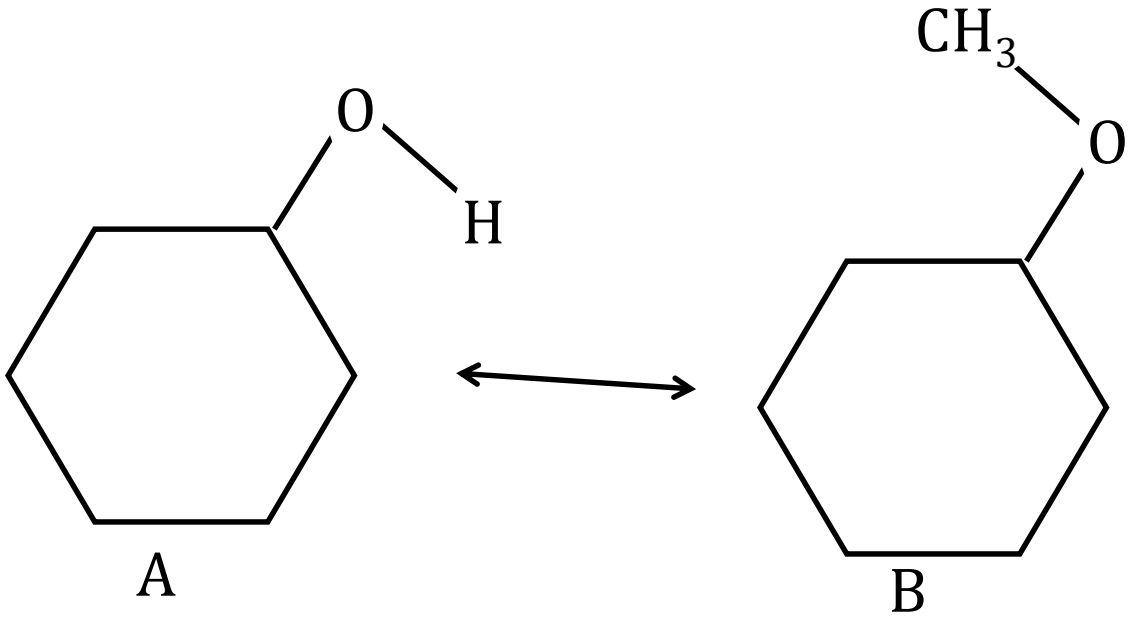
B + P ↔ A + P (free A ↔ B forget the protein)

are small changes – smaller than

- removing water order, removing water energy, finding protein…

Example

- small change

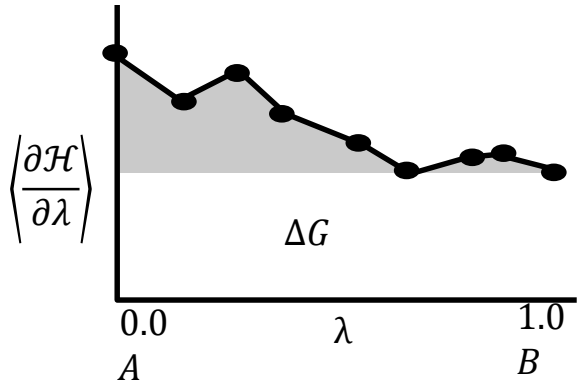


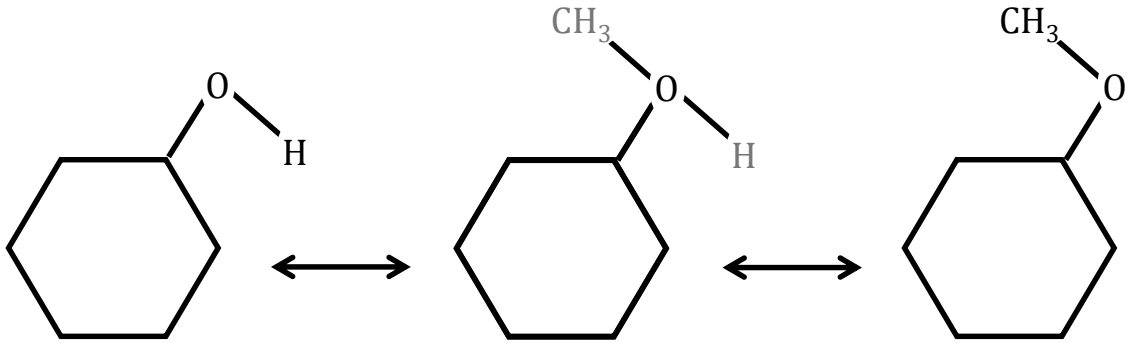
Fictitious states

Remember formulae

$$\Delta G = \int_A^B \left\langle \frac{\partial \mathcal{H}(\mathbf{p},\mathbf{r})}{\partial \lambda} \right\rangle_\lambda d\lambda \quad \text{and} \quad \Delta G = \sum_{i=0}^{N_{step}} (H_{i+1} - H_i)$$

make chemistry a function of λ





A
λ=0

A/B
λ=0.5

B
λ=1

λ dependence

λ = 0 an OH group

λ = 1 an OCH_3 group

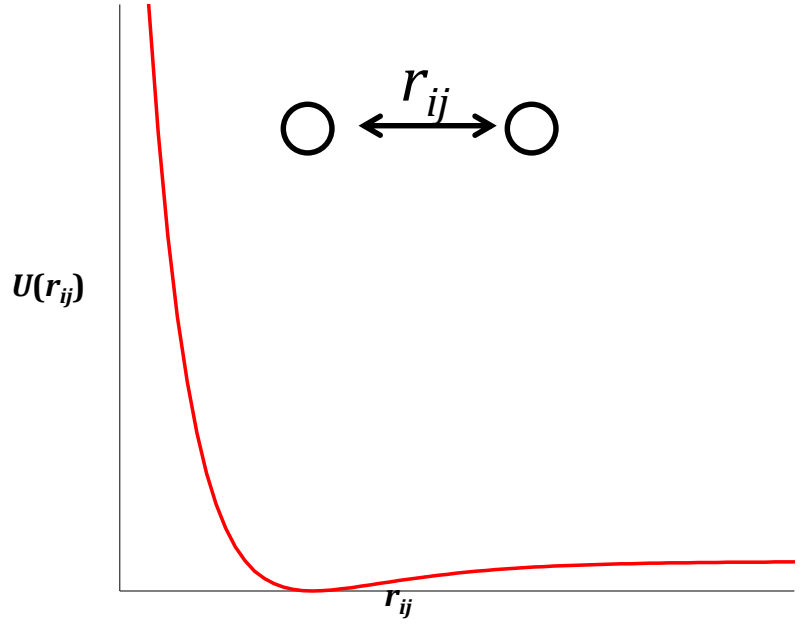
λ = 0.5

- charge of H – half of original charge
- radius / size (σ, ε) half of real value and so on

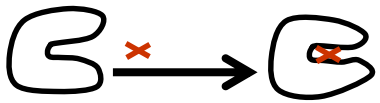
Atoms gradually

- appear in one direction
- disappear in other

Description of system is now function of λ

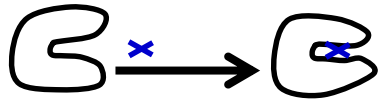


λ dependent simulations



Two simulations necessary

- λ from 0.0 ↔ 1.0 in protein
- λ from 0.0 ↔ 1.0 in water
- both from red ↔ blue



As λ slowly moves from 0.0

- water gradually feels more/less influence of some atoms
- system should not have to rearrange itself too much

When does method work best ?

- when changes are small
  - comparison of similar ligands in a protein

Summary of free energy calculations

From first principles: free energy differences, equilibria

- easy to calculate
- in practice impossible (sampling not possible)

Forget absolute free energies

- ΔG determine most phenomena in the world

Processes like binding still too difficult to simulate

- slow, too many conformations / states to visit

Most calculations use $\Delta\Delta G$

- aim to get relative binding strengths

Simulated Annealing

Classic reference – in stine

Basic tools

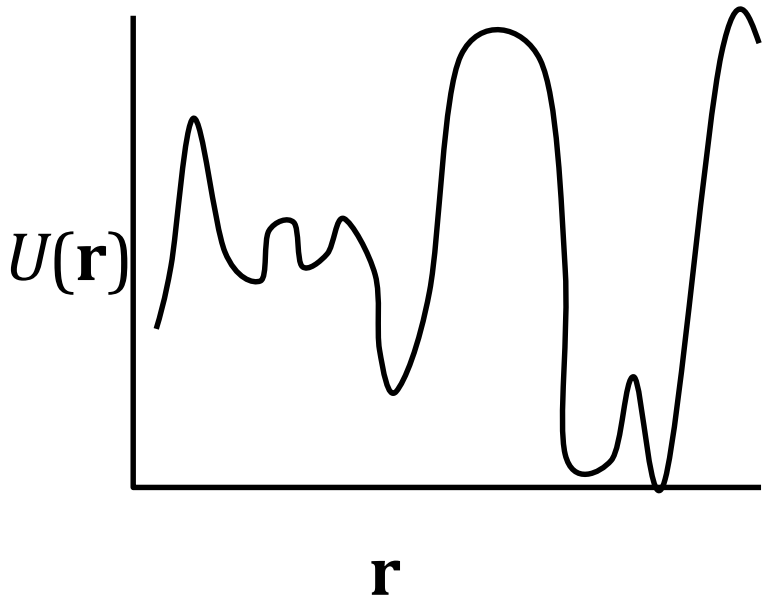
- MC or MD
  - with control of temperature (temperature bath)

Use : difficult optimisation problem

- chip layout
- travelling salesman problem
- protein structure

Optimisation problem

- several dimensional (2 to 2000)
- many local minima



Kirkpatrick, Gelatt and Vecchi, Science, 220, 671-680, "Optimization by Simulated Annealing" (1983)

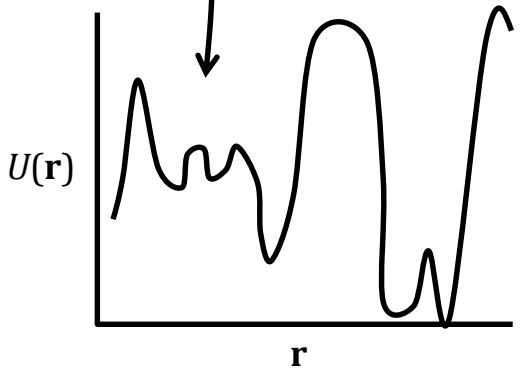
Procedure

```
while (T > T_end)
    T(t) = T_0 e^(-ct)
    move system (Monte Carlo)
```

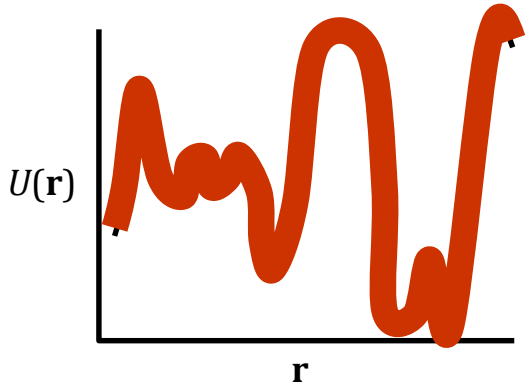
- T_0 initial temperature is hot
- c is decay rate (cooling of system)
- cost function is
  - E_{pot} in chemistry
  - path length in travelling salesman
  - board cost in chip layout problem …
- why may this work ?

Simulated Annealing concept

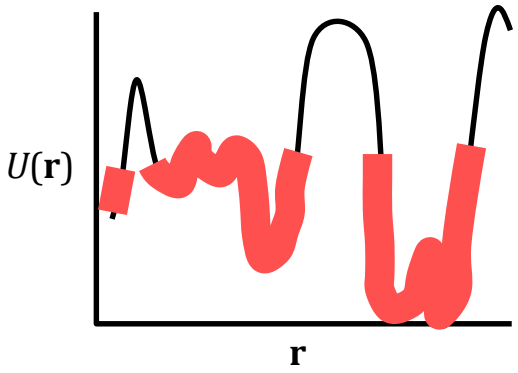
initial (poor) guess



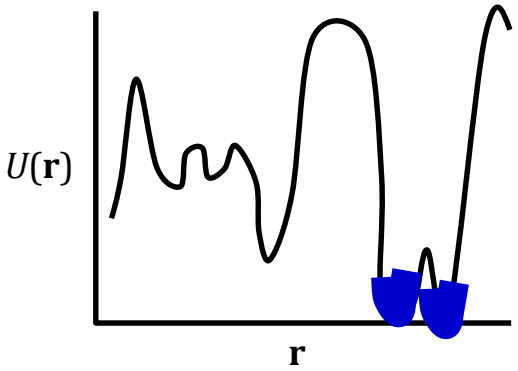
initial high T distribution



cooler T



cold T

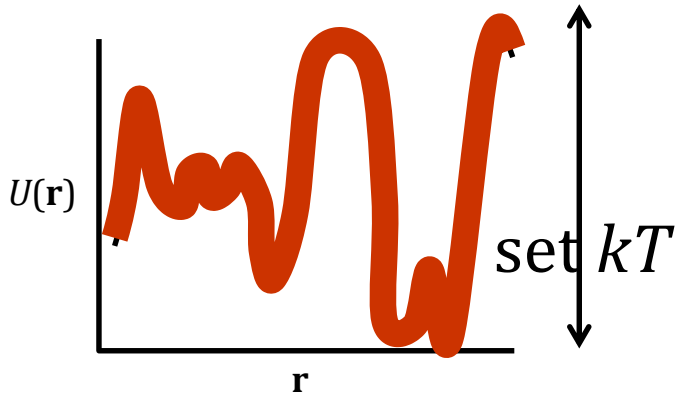


Properties, practical issues

Admit that there may not be a best solution

- not worth spending effort between many very good solutions

Some problems have "phase transitions"



How hot should T_0 be ?

- infinite ? No : look at barriers

How slow should cooling be (c) ?

- system should be at equilibrium
- very slow

Cool exponentially ?

- best first guess
- should certainly cool more slowly at transition points

Anneal with MC or MD ?

Historic use of Monte Carlo

- easiest to apply to many problems

Use MD ?

- provides expected advantages (efficiency)
- uses available gradient / derivative information

Implementation

- Couple to temperature bath (dynamics) / make T time dependent (MC)

Use in practice ?

- simulated annealing in
  - most MD codes, refinement packages, …

Refinement of Structures (NMR / X-ray)

Story from first semester

- problem : generate protein coordinates from NMR information (or X-ray)
- distance geometry gives an initial guess, but
  - distance geometry methods spread error across all distances
  - errors are spread across bonds, measured distances
  - chirality may be broken (causes distance problems)

Belief

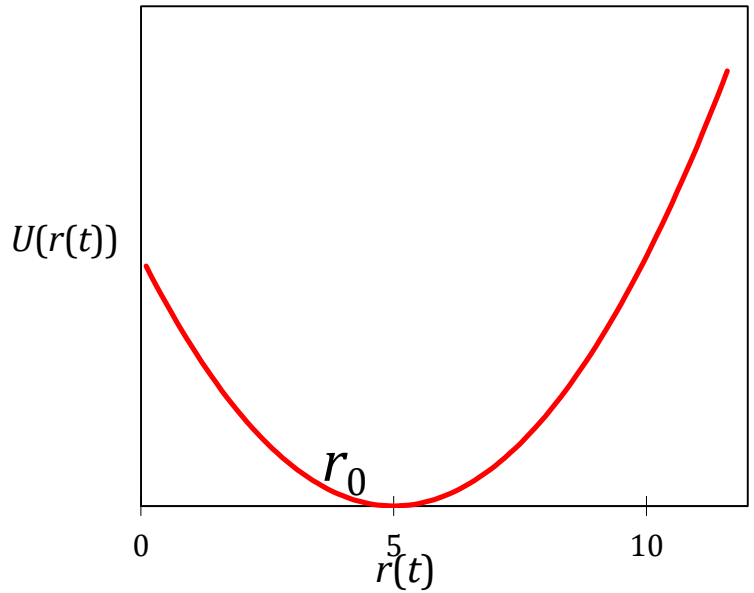
- coordinates are not bad, but could be improved

Pseudo – energy terms

For some distance measurement *i* between some pair of atoms

- r_0 measured distance
- $r(t)$ distance between particles at time (t)
- say $U_i(r) = c_i(r(t) - r_0)^2$
- add this to normal force field

$$U_{tot}(\mathbf{r}) = U_{phys}(\mathbf{r}) + \sum_{i=1}^{N_{restraints}} U_i(\mathbf{r})$$



$U_{phys}(\mathbf{r})$ normal force field - atomistic (bonds, electrostatics…)

result ?

System moves to low energy + low fake energy

- gradually moves to agree with experimental data

Practical issues $U_{tot}(\mathbf{r}) = U_{phys}(\mathbf{r}) + \sum_{i=1}^{N restraints} U_i(\mathbf{r})$

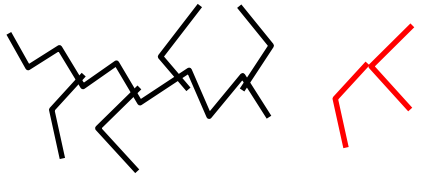
$U_i(r) = c_i\,(r(t) - r_0)^2$

- big c very artificial
- small c system will be slightly biased to agree with experimental data

Fake Energies - examples

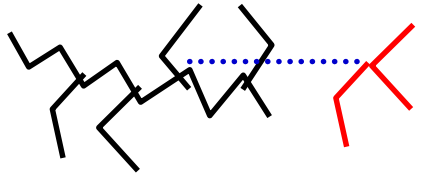
Refinement of

- X-ray structures (common)
- NMR (often)
- others: microwave spectroscopy, …



Modelling problems

- you want to put a bond in a model
  - putting it in directly
    - high energy bond
    - system stuck in minimum
  - introduce a distance restraint
    - gradually increase associated constant *c*



Summary

What one can do with related methods

- look at timescales of motions (very superficial)
- free energy calculations – important for problems such as binding of ligands
- simulated annealing – methods used as minimizers, not necessarily to get an ensemble
- pseudo-(potential) energies (X-ray, NMR, …)