

# Introduction to computer modeling of

Note Title

4/22/2011

## Soft matter systems.

- Complexity cannot be investigated analytically
- Often have large scale / long time correlated / collective dynamics phase behavior.  
⇒ use numerical / simulation methods to help understand / verify soft material properties
- Two Main Approaches

I). Monte Carlo (Metropolis 1953).

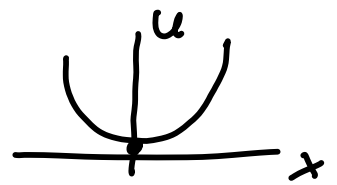
e.g. a particle in a harmonic potential

$$U = kx^2.$$

it will oscillate & make random displacements

What's the probability the particle will have a given energy.

$$P(u) \sim e^{-\beta u}$$



① start with random particle position  $r_0$

$$U_0 = U(r_0)$$

② make a random displacement

$$r_0 \rightarrow r_1$$

$$U_1 = U(r_1) \quad \Delta U = U_1 - U_0$$

probability of transition is  $e^{-\beta \Delta U}$

To test whether a move is made, we "throw a dice" or generate a random number  $n \in [0, 1]$

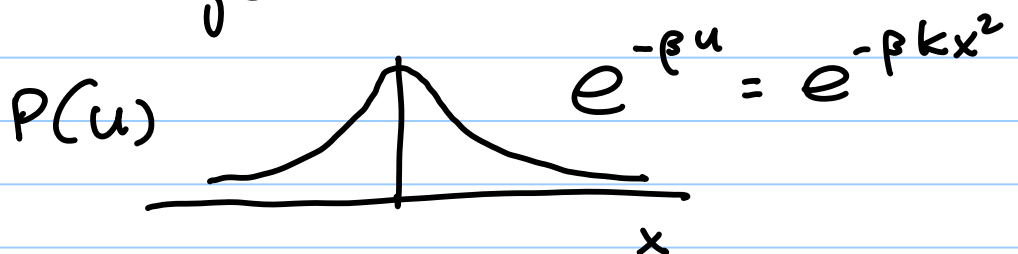
if  $\Delta U < 0$ ,  $e^{-\beta \Delta U} > 1$ , then always move

if  $\Delta U > 0$ , test  $n < e^{-\beta \Delta U}$  ?

yes, make a move

no, reject transition

③ make a large # of test moves.

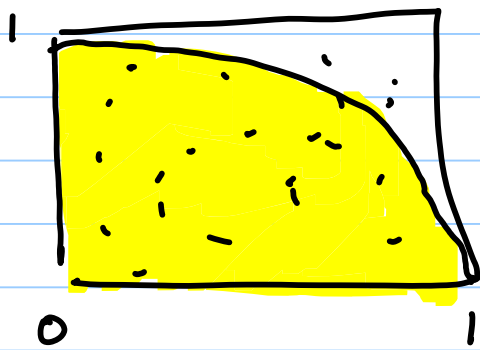


## 2nd example

use MC -method to do numerical calculations

- Calculate  $\pi$ .

Imagine a square board and we throw darts onto the board randomly



look at where the darts fall

Count how many darts fall in shaded region vs. total # of darts.

Shaded area =  
circle / 4

$$= \frac{\pi R^2}{4} = \frac{\pi}{4} = \frac{\# \text{ of darts in region}}{\text{total \# of darts.}}$$

Algorithm ① generate two random #s.  
 $x, y$

② Test  $x^2 + y^2 < R^2$  ?

yes  $\rightarrow$  add to count.

no  $\rightarrow$  reject.

③ repeat  $N$  times

$$\frac{\# \text{ accepted}}{N} \rightarrow \frac{\pi}{4} \text{ as } N \rightarrow \infty$$

exercise How does error change w/  $N$ ?

### Advantage

- If we have a complex energy landscape, we can sample the  $U(r)$  quickly.

### Limitations

- no/little information about barriers or dynamics  
problem with protein folding

## II. Molecular dynamics

know the pair potentials between particles  
three-body

We can determine the position & momentum from Newton's eqn. of motion or Langevin

① know  $U_{ij}(r) \Rightarrow F_{ij}(r)$

② Integrate

$$V_i(t + \Delta t) = V_i(t) + \sum_j \frac{F_{ij}(r, t)}{m_i} \Delta t$$

Euler  
method

$$X_i(t + \Delta t) = X_i(t) + \left( \frac{V_i(t + \Delta t) + V_i(t)}{2} \right) \Delta t.$$

Evolving particle trajectories through time

Truncation error  $\sim O(\Delta t^2)$

require small  $\Delta t$

For simulating atoms,

$$\Delta t < 10^{-12} \text{ s.}$$

To study proteins, typical  
time of entire simulation

$$\sim 1 \mu\text{s} \quad (2011).$$

real processes can be ms

Advantage :

- get dynamics

Limitation :

- hard to cross large barriers.

In the next 4 weeks, more details

on MC & MD.

Generally, we are doing simulations in boxes of finite size

for atomic simulations

$$x_m \approx O(\text{\AA})$$

large box size  $\approx O(100 \text{ nm})$ .

3D  $\rightarrow 1000^3$  particles.

maximum # within a reasonable real time.

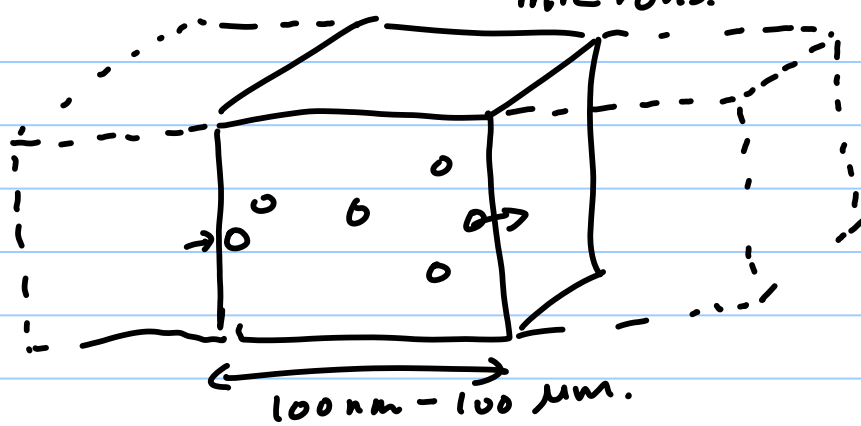
Coarse-grain: neglect atomic-level interactions

keep the physics on the length scale of interest.

For polymers, keep Kuhn length,  $R_g$

typically Kuhn length  $\approx 1-10 \text{ nm}$ .

box size can be hundreds of microns.



periodic boundaries.  
remove wall-induced artifacts.

To simulate large systems, we cannot get system properties that vary on a wavelength larger than the box size.

"finite-size" effect.

Shape of box can also introduce artifacts :

cubic box tend to bias the formation of cubic lattices unit cells

frustrate the formation of tetragonal or other shape of crystals.

WARNING:

To do a simulation, we need to choose box size, shape, particle density carefully.

simulation time

# Basic programming. (C)

Scheme of a program :

- To calculate a function value.

① Call/include libraries w/ fctns needed.  
e.g. `#include <stdio.h>`  
`#include <stdlib.h>`

② Declare variables  
To calculate  $f(x)$ , we need to  
tell computer if  $x$  is a integer  
or a float  
or a double.

integer : no decimal

float :  $10^{-16}$  precision

double :  $10^{-32}$  precision

e.g. `int x ;`  
`double x, f ;`  
`double x[10] ;` ← array w/ 10  
components.

③ declare our function.

e.g.  $f(x) = \sin(x) + \cos(x)$



$$f = \sin(x) + \cos(x) ;$$

Example program 1 :

```
#include <stdio.h>
#include <stdlib.h>

void main(void) {
    int i ;
    double x[10], y[10], r[10];


    for (i=1 ; i < 10 ; i++) {
        x[i] = 0.1 * i ;
        y[i] = 0.05 * i ;
        r[i] = x[i] * x[i] + y[i] * y[i];
        printf(" values = (%lf %lf %lf)\n "
            , x[i], y[i], r[i]);
    }
}
```

To compile :

```
gcc -o test_file test.c
```

                  ↑                                  ↑  
                  output file name                  source file.

To run :

 test\_file

↑ location is current directory

## Molecular Dynamics

Given initial particle positions & momentum.

Given intermolecular potentials/ forces

We calculate the particle trajectories using

Newton's Equation of motion

$$v_i = \dot{x}_i$$

$$a_i = \dot{v}_i = \left( \frac{f_i}{m} \right)$$

The basic setup of a  
molecular dynamics simulation

- ① Specify parameters (T, N, P, # of simulation steps)
- ② initialize - select initial positions & velocities.
- ③ compute forces on all particles
- ④ Integrate trajectories using EOM
- ⑤ Repeat ③ - ④ to calculate particle trajectory  
Collect system properties & output

## Notes

I. The most time consuming calculation is for the forces.

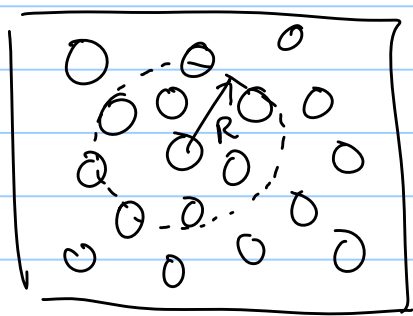
Since there are  $N$  particles, we need to calculate  $\frac{N(N-1)}{2}$  pair interactions

So typically, computational time  $\sim N^2$

There are some tricks to increase the speed so that time  $\sim N$

① truncate the potential (also shift).

② Create a list of interacting neighbors. (for "dense" systems)



- Only calculate the interactions with the neighbors.

- keep an <sup>updated</sup> list of neighbors.

For  $N > 200$ , time  $\sim N$

II. Integrate the EOM numerically

$$r(t + \Delta t) = r(t) + v(t) \Delta t + \frac{F(t)}{2m} \Delta t^2 + \frac{\Delta t^3}{3!} \ddot{r} + O(\Delta t^4)$$

There is always a truncation error. This always leads to instabilities at long times.  
(deviates from the real trajectory)  
total energy is not conserved

## - Integration Methods

### 1. Euler method

$$v(t + \Delta t) = v(t) + \frac{f(t)}{m} \Delta t.$$

$$r(t + \Delta t) = r(t) + v(t + \Delta t) \Delta t.$$

$$\text{error} \sim O(\Delta t^3)$$

generally not accurate.

### 2. Verlet method

$$r(t - \Delta t) = r(t) - v(t) \Delta t + \frac{f(t)}{2m} \Delta t^2 - O(\Delta t^3)$$

$$r(t + \Delta t) = r(t) + v(t) \Delta t + \frac{f(t)}{2m} \Delta t^2 + O(\Delta t^3)$$

$$r(t - \Delta t) + r(t + \Delta t) = 2r(t) + \frac{f(t)}{m} \Delta t^2 + O(\Delta t^4)$$

$$\text{or } r(t + \Delta t) = 2r(t) - r(t - \Delta t) + \frac{f(t)}{m} \Delta t^2$$

a. it is  $O(\Delta t^4)$  accurate.

b. does not use velocity.

Another variation

Velocity Verlet

$$r(t + \Delta t) = r(t) + v(t) \Delta t + \frac{f(t)}{2m} \Delta t^2$$

$$v(t + \Delta t) = v(t) + \frac{f(t + \Delta t) + f(t)}{2m} \Delta t.$$

This method "time reversible"

see Frenkel Smit Chapter 6

error is same as Verlet.

---

So we expect the total energy of a system to drift during a long MD simulation.

To correct for this, thermostats are used.

To calculate the kinetic energy or temperature

$$k_B T = m \langle v_x^2 \rangle$$

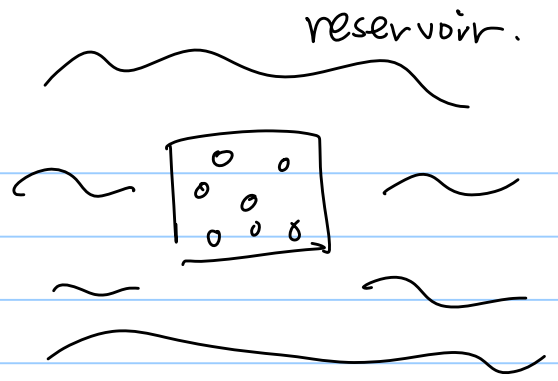
↑  
velocity of a given component.

Thermostats keep  $T$  constant on average.

The simplest thermostat is to rescale the velocities of all particles such that  $T$  is constant at every step. — neglects fluctuations in  $K.E.$

## Andersen thermostat

particles in system will collide with reservoir randomly.



On a collision, the chosen atom/molecule's momentum is generated from a Boltzmann distribution at temperature  $T$ .

Nosé - Hoover thermostat. (MD at constant  $T$ )  
(Frenkel Smit Appendix E).

use a "Lagrangian minimization" method to keep the temperature fixed.

Stochastic dynamics (Langevin Egn. of Motion).

add a thermal fluctuation to particle.

add a dissipation force through friction.

$$f^{\text{total}} = f^{\text{int}} + f^{\text{fric}} + \delta f$$

$f^{\text{fric}}$  &  $\delta f$  are coupled through fluctuation - dissipation theorem.

## Advantages of MD

① It gives us information about dynamic properties ( $\eta$ ,  $D$ , conductivity, etc.).

② With enough sampling (long time MD)

ergodicity:  $\overline{A} = \langle A \rangle$

We should also get the thermodynamic properties.

③ We can also examine non-ergodic systems.  
(relaxation time  $\rightarrow \infty$ )

glasses, gels, non-equilibrium phase transitions & solids.

## Disadvantages

① system state can be trapped in a local minimum (often depends on the initial condition).

② Hard to know when a system has reached equilibrium

e.g. protein folding, protein adsorption

In a fully atomistic simulation  
( $\Delta t \approx 10^{-15}$  s)

is 10 ns enough?

100 ns

1  $\mu$ s

?

When are the processes considered  
to be equilibrated states?

must always run longer to check  
(much)



# Week 12

Note Title

5/20/2011

Molecular dynamics model the trajectory of every molecule.

But if we want to study the properties of a colloidal particle with  $R = 1 \mu\text{m}$  in a solvent with  $a = 0.5 \text{ nm}$ .

If we use a box of  $10 \mu\text{m}^3$ , that means we need to fill it with

$$\approx \frac{10 \mu\text{m}^3}{a^3 \approx 0.1 \text{ nm}^3} \quad \# \text{ of solvent particles}$$

$$\approx 100 \cdot 10^9 \text{ solvent particles}$$

This is very inefficient for a few colloids if we want to study the dynamics of larger particles!!

So we need to consider "Mesoscale" methods in between molecular & continuum dynamics

One way to model colloids is with  
Stochastic dynamics or Brownian dynamics

include collisions with solvent as a  
fluctuation in force.

$$F_i^{\text{tot}} = F_i^{\text{pot}} + F_i^{\text{fric}} + \delta F_i^{\text{fluc}}$$

$$F_i^{\text{pot}} = - \sum_j \frac{\partial U_{ij}}{\partial r_i}$$

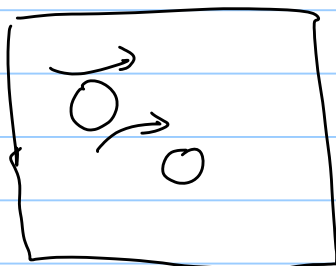
$$F_i^{\text{fric}} = - \gamma v_i \quad \gamma = \text{friction coef.}$$

$$\langle \delta F_i^{\text{fluc}} \rangle = 0$$

$$\langle \delta F_j^{\text{fluc}} \delta F_i^{\text{fluc}} \rangle = 2kT \gamma \delta_{ij}$$

We can model the motion of a colloid  
freely diffusing in a fluid.

We also need to consider hydrodynamic  
interactions.



The hydrodynamic interactions capture how the motion of one particle  $i$  affect the trajectory of another particle  $j$ .

$$D_{ij} = \frac{kT}{6\pi\eta a} I_{ij} + \Omega_{ij} \quad I \approx \text{identity matrix.}$$

In a quiescent fluid, this was solved by Stokes.

$$D_{ij} = \begin{cases} \frac{kT}{6\pi\eta a} \delta_{ij} & \text{for } i=j \\ \frac{kT}{8\pi\eta r_{ij}} \left( I + \frac{r_{ij} r_{ij}}{r_{ij}^2} \right) & \text{for } i \neq j. \end{cases}$$

"Oseen tensor"  $\Rightarrow$

$$\zeta = 6\pi\eta a$$

$a \equiv$  particle radius

$\eta =$  solvent viscosity

For one particle undergoing Brownian motion, we can write

$$\langle \Delta r^2 \rangle = 2D \Delta t = \frac{2kT}{\zeta} \Delta t$$

$$r(t+\Delta t) = r(t) + \underbrace{\sqrt{\frac{2kT}{\zeta} \Delta t}}_{\sqrt{2} \tilde{B} \cdot \tilde{dW}}$$

"Brownian dynamics"

For multiple particles  $\downarrow$  potential forces acting on a particle  $i$ :

$$\Delta \underline{r} \approx \left[ \underbrace{\frac{1}{kT} \underline{D}}_{\text{mobility effect between } i \neq j} \cdot \underline{F} + \nabla \cdot \underline{D} \right] \Delta t + \sqrt{2} \underline{B} \cdot d\underline{W}$$

mobility  
effect between  
 $i \neq j$ .

$$\underline{B} \cdot \underline{B}^t = \underline{D}$$

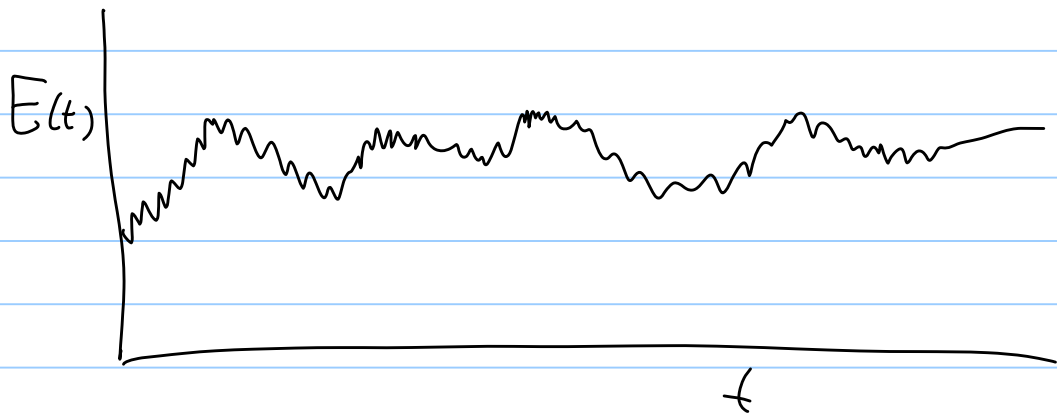
$d\underline{W}$  = fluctuation

$\nabla \cdot \underline{D}$  = spatial variation of particle diffusion. Usually 0 in free space (homogeneous diffusivity).

$$\langle \Delta r^2 \rangle = \frac{2kT}{J} \Delta t.$$

Brownian dynamics with hydrodynamic interactions

## Data analysis



Simulation is run for a total time  $\tau$   
the equilibrium average of  $A$

$$A_\tau = \frac{1}{\tau} \int_0^\tau dt A(t).$$

Variance in  $A_\tau$  :

$$\begin{aligned} \sigma^2(A) &= \langle A_\tau^2 \rangle - \langle A_\tau \rangle^2 \\ &= \frac{1}{\tau^2} \int_0^\tau \int_0^\tau dt dt' \underbrace{\langle [A(t) - \langle A \rangle] [A(t') - \langle A \rangle] \rangle} \end{aligned}$$

time correlation fcn :  $C_A(t-t') = \langle \delta A(t) \delta A(t') \rangle$

$$\begin{aligned} \sigma^2(A) &\approx \frac{1}{\tau} \int_{-\infty}^{\infty} dt C_A(t) \\ &\approx 2 \frac{t_A^c}{\tau} C_A(0) \end{aligned}$$

$$t_A^c = \int_0^{\infty} dt \frac{C_A(t)}{C_A(0)} \approx \text{correlation time.}$$

relative  
variance

$$\frac{\sigma^2(A)}{\langle A \rangle^2} \approx \frac{2 t_A^c}{\tau} \frac{\langle A^2 \rangle - \langle A \rangle^2}{\langle A \rangle^2}$$

$$\sigma^2 \sim \frac{t_A^c}{\tau} \sim \frac{1}{\# \text{ of independent samples}}$$

root mean square error  $\sim \sqrt{\frac{t_A^c}{\tau}}$

Our simulation time  $\tau$  should always be  $\gg$  the correlation time  $t_A^c$

For a Hamiltonian of  $N$ -particles

$$H(\underline{r}^N, \underline{p}^N)$$

$\underline{r}^N \equiv$  position,  $\underline{p}^N =$  momentum

$$P(\{\underline{r}^N, \underline{p}^N\}) \sim e^{-H/kT}$$

The partition fcn

$$Q = \int d\underline{p}^N d\underline{r}^N e^{-H/kT}$$

average property of  $A$   $\beta \equiv \frac{1}{kT}$

$$\langle A \rangle = \frac{\int d\underline{p}^N d\underline{r}^N A(\underline{p}^N, \underline{r}^N) e^{-\beta H}}{\int d\underline{p}^N d\underline{r}^N e^{-\beta H}}$$



At thermal equilibrium, there is  
reversibility:

detailed balance - # of accepted moves  
leaving state old  
= # of accepted moves  
from new  $\rightarrow$  old

$$N(o) \pi(o \rightarrow n) = N(n) \pi(n \rightarrow o)$$

$N(i)$  = # of configurations in  
state  $i$

$\pi(o \rightarrow n)$  = prob. of successful  
transition from  $o \rightarrow n$

$$= \alpha(o \rightarrow n) \text{acc}(o \rightarrow n)$$

$\alpha(o \rightarrow n)$  = prob. of total moves  
from  $o \rightarrow n$

$\text{acc}(o \rightarrow n)$  = prob. of acceptance  
from  $o \rightarrow n$



If  $\alpha(o \rightarrow n) = \alpha(n \rightarrow o)$ .

$$N(o) \text{acc}(o \rightarrow n) = N(n) \text{acc}(n \rightarrow o)$$

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \frac{N(n)}{N(o)} = e^{-\beta[u(n) - u(o)]}$$

Science  
Basic structure of M.C. algorithm

1. start with a random particle configuration.

calculate its  $U(\{r^n\})$ .

2. Give a random particle a random displacement and calculate  $U'(\{r^n\})$

3. generate a random number  $x$ .

accept the move from  $r^N$  to  $(r^N)'$

with probability

$$\text{acc}(o \rightarrow n) = \min\left(1, e^{-\beta(u' - u)}\right)$$

test if  $x < \text{acc}(o \rightarrow n)$

There is an art (engineering-type approaches) in generating new configurations.

There are several ways to do a Monte Carlo simulation

lets look at ideal gas

$$P V = N k T.$$

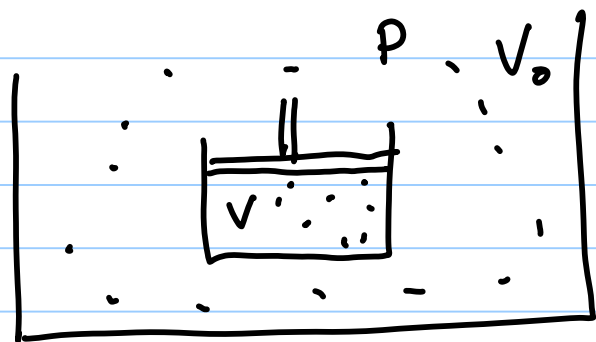
need to fix 3 variables.

Common MC (Canonical Ensemble).

fix  $N V T$ .

$$\mathbb{P} \sim e^{-\beta U(\{r^N\})}.$$

Isobaric - Isothermal (NPT)



measure the equation of state of a gas / fluid

$$\mathbb{P} \sim e^{-(U + PV) / kT}$$

$V$  is treated as an additional coordinate.

Make trial moves in  $V$

$$\ln V' = \ln V + d(\ln V).$$

$$\text{acc}(0 \rightarrow n) = \min\left(1, e^{-\beta \left[ U(\underline{s}^N, v') - U(\underline{s}^N, v) + P(v' - v) - \frac{N}{\beta} \ln\left(\frac{V'}{V}\right) \right]}\right)$$

rescaled  
coordinate

$$\underline{s}^N = \frac{\underline{r}^N}{L}$$

$$V = L^3$$

$$U(L') = \left(\frac{L}{L'}\right)^{\alpha} U(L) \quad \leftarrow \text{dimensionality.}$$

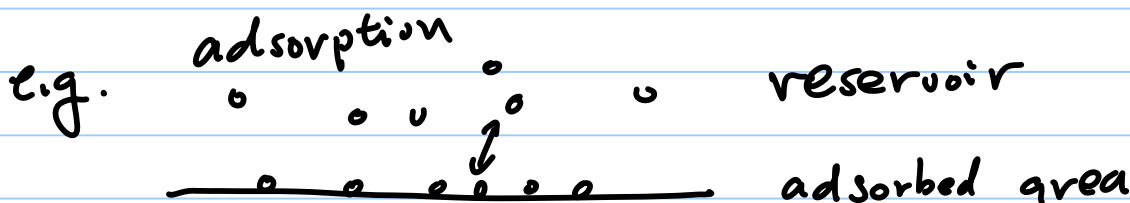
check if  $P = \frac{1}{3V} \left\langle \sum_{i=1}^N \underline{r}_i \cdot \underline{f}_i \right\rangle = P_{\text{input}}$

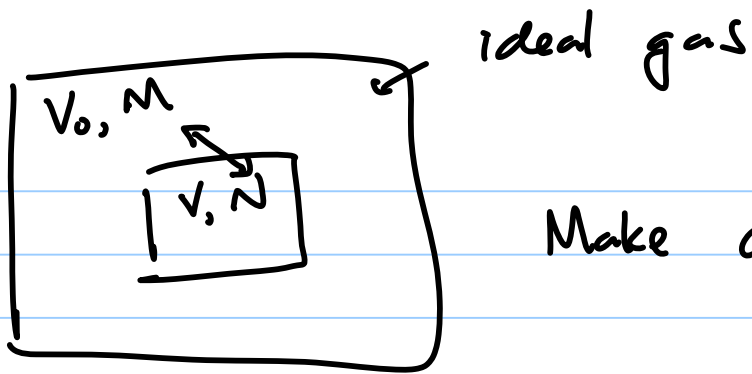
How about when  $N$  is not fixed?

Common for systems w/ many phases.

fix chemical potential  $\mu$ ,  $V$ ,  $T$

Grand-canonical ensemble.





Make changes in  $N$

$$P \sim e^{-\beta(U - \mu N)}$$

$$\alpha(N \rightarrow N+1) = \frac{V(M-N)}{V'(N+1)} e^{-\beta[u(\xi^{N+1}) - u(\xi^N)]}$$

$$\alpha(N+1 \rightarrow N) = \frac{V'(N+1)}{V(M-N)} e^{-\beta[u(\xi^N) - u(\xi^{N+1})]}$$

$$\mu = g kT \ln \Lambda^3 \quad \Lambda = \text{thermal wavelength}$$

$$= \sqrt{\frac{h^2}{2\pi m k_B T}}$$

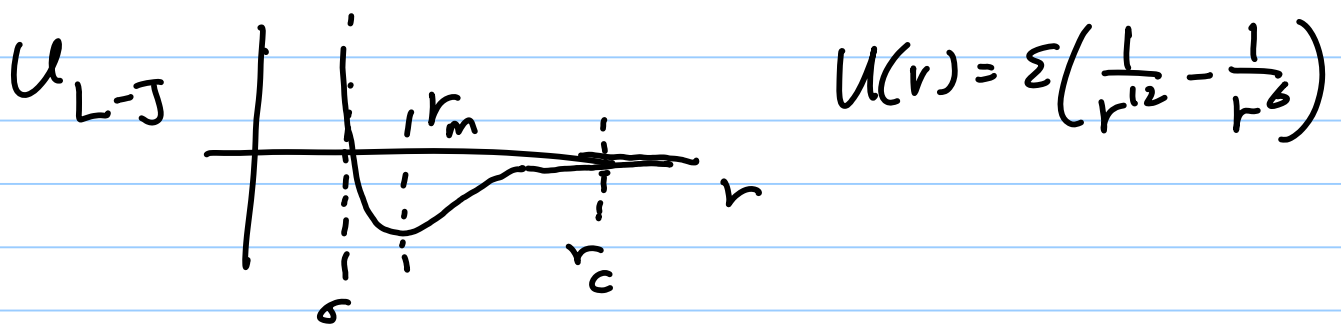
$$\text{acc}(N \rightarrow N+1) = \min \left\{ 1, \frac{V}{\Lambda^3(N+1)} e^{\beta[\mu - u(N+1) + u(N)]} \right\}$$

$$\text{acc}(N \rightarrow N-1) = \min \left\{ 1, \frac{\Lambda^3 N}{V} e^{\beta[\mu + u(N-1) - u(N)]} \right\}$$

Things we need to be careful about

Simulations are done in a fixed, finite box.

- properties that have long wave lengths ( $> L$ ) cannot be captured
- pair interactions that have long range need to be truncated.



truncate at  $r > r_c$

$$U'(r) = \begin{cases} U_{LJ} & r < r_c \\ 0 & r > r_c \end{cases}$$

leads to errors due to truncation.

$$\text{For } r_c = 2.5 \sigma \quad U(r_c) \approx U(r_m) / 60$$

$$\text{but } U^{\text{tail}} = \frac{1}{2} (4\pi\rho) \int_{r_c}^{\infty} dr r^2 U(r) \\ \approx 10 \% \text{ of total pot. } U / \text{atom}$$

Truncate & shift

$$U'(r) = \begin{cases} U^{LJ}(r) - U^{LJ}(r_c) & r \leq r_c \\ 0 & r > r_c \end{cases}$$

Matching simulations to real systems.

length :  $\sigma \equiv$  particle diameter

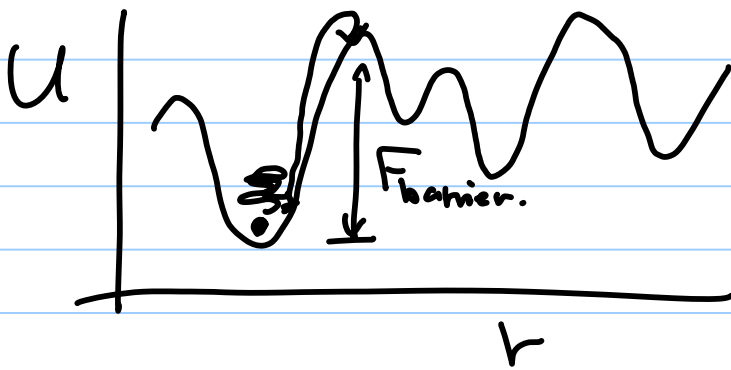
energy :  $\epsilon \equiv$  from L-J or other potentials

mass :  $m \equiv$  particle mass

time :  $\sigma \sqrt{\frac{m}{\epsilon}}$

temperature :  $\frac{\epsilon}{k_B}$

map to real particles.



Construct a full picture of a complex energy landscape

-- often obstructed by energy barriers

The probability of overcoming an energy barrier

$$P(F_{\text{barrier}}) \sim e^{-\frac{F_{\text{barrier}}}{kT}}$$

becomes very improbable for  $\frac{F_{\text{barrier}}}{kT} > 10$

Question : How do we make sure the entire landscape is sampled?

- There are methods to bias sampling or biased method to create new configurations

- umbrella sampling.
- Wang-Landau method
- configurational-bias Monte Carlo

Biased Sampling technique:

whatever we can imagine, but must satisfy detailed balance.

$$N(o) \pi(o \rightarrow n) = N(n) \pi(n \rightarrow o).$$

$$\pi(o \rightarrow n) = \alpha(o \rightarrow n) \text{acc}(o \rightarrow n)$$

$N(o)$ : population of state  $o$

$\alpha(o \rightarrow n)$ : prob. of total moves from  $o$

$\text{acc}(o \rightarrow n)$ : prob. of successful moves

Lets assume we have a MC scheme to generate trial configurations w/ probability

$$\alpha(o \rightarrow n) = f[u(n)]$$

$$\alpha(n \rightarrow o) = f[u(o)]$$



$f \equiv$  arbitrary biasing function

detailed balance  $\Rightarrow$

$$\frac{\text{acc}(0 \rightarrow n)}{\text{acc}(n \rightarrow 0)} = \frac{f[u(0)]}{f[u(n)]} \frac{N(n)}{N(0)}$$
$$= \frac{f[u(0)]}{f[u(n)]} e^{-\beta[u(n)] - u(0)}$$

So an acceptance rule is

$$\text{acc}(0 \rightarrow n) = \min\left(1, \frac{f[u(0)]}{f[u(n)]} e^{-\beta[u(n)] - u(0)}\right)$$

## ① Umbrella Sampling (Torrie & Valleau)

bias  
modify the random process that

samples the configuration space

-- replace the Boltzmann factor ( $e^{-\beta u}$ )

of the system by a non negative

weight function  $\pi(\tilde{r}^N)$

$\pi(\tilde{r}^N)$

$$\Rightarrow \langle e^{-\beta \Delta U} \rangle = \frac{\int dr^N \pi(r^N) e^{-\beta U_1(r^N)} / \pi}{\int dr^N \pi(r^N) e^{-\beta U_2(r^N)} / \pi}$$

$$= \frac{\langle e^{-\beta U_1} / \pi \rangle_{\pi}}{\langle e^{-\beta U_2} / \pi \rangle_{\pi}}$$

$\pi(r^N)$  promote sampling of rarely-accessed configuration space

It is usually better to perform several "umbrella" sampling runs in overlapping windows.

When we calculate the ensemble average of thermodynamic property  $A$

$$\langle A \rangle = \frac{\langle A / \pi \rangle_{\pi}}{\langle 1 / \pi \rangle_{\pi}}$$

## ② Wang - Landau Algorithm (2001)

- bias by the system energy  
density of states  $g(E)$ .

create a histogram of  $g(E)$   
with bin size  $\delta E$   
# of bins  $N$

$$N \delta E = E_{\max} - E_{\min}$$

① initially, we set  $g^o(E) = 1$

$$\ln[g^o(E)] = 0.$$

$\because g(E)$  is unknown

create a visit histogram  $H(E) = 0$ .

② create a new configuration

calculate  $E(\vec{r}_N^n)$

Add 1 to  $H(E_n)$

accept or reject a move to state  $n$   
by a probability  $P = \min\left(1, \frac{g(E_0)}{g(E_n)}\right)$

add to  $\ln[g(E_0)]$  by  $\ln[f^i]$ .

★ The goal is to have  $f^i \rightarrow 1$  for  
large # of iterations

usually  $f^0 = e \approx 2.72$ ;

★ The criteria for when to end an  
iteration is when  $H(E)$  is flat.

"flatness" i.e.  $\sum_{j=1}^N (H(E_n) - \langle H(E_n) \rangle)^2 < 0.1$

when sufficiently flat, we reset  
 $H(E) = 0$ , and reiterate with

$$\ln(f^{i+1}) = \frac{1}{2} \ln f^i.$$

④ repeat ② - ③ until  $\ln(f^{i+1}) < 10^{-6}$

- complex energy landscapes  
protein folding, glass transition  
problems with many body interactions.

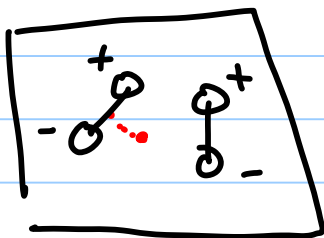
- doesn't satisfy detailed balance  
in the intermediate steps

## Configurational-bias trial moves

- we will  $\overset{\text{pre-}}{n}$  generate configurations that  
have a higher probability of being  
accepted.

Example : orientational bias

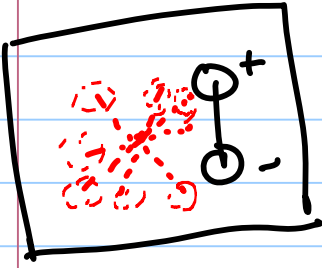
①



### Algorithm

1. move the center-of-mass  
of a molecule w a  
small random displacement  
& determine all interactions  
that do not depend on orientation.

②



$U^{pos}$

2. Generate  $k$  trial configurations

$\{\tilde{b}_1, \dots, \tilde{b}_k\}$  and calculate

the orientational energy of each

$$U^{OR}(\tilde{b}_i)$$

$$U^{total} = U^{pos} + U^{OR}$$

3. We define a factor (Rosenbluth) to weight the prob. of orientations

$$W(n) = \sum_{j=1}^k e^{-\beta U^{OR}(\tilde{b}_j)}$$

and randomly select one, say  $n$ ,

with probability

$$P(\tilde{b}_n) = \frac{e^{-\beta U^{OR}(\tilde{b}_n)}}{W(n)}$$

4. For the old configuration  $o$ , the

weighting procedure is also performed

$$W(o) = e^{-\beta U^{OR}(\tilde{b}_o)} + \sum_{j=2}^k e^{-\beta U^{OR}(\tilde{b}_j)}$$

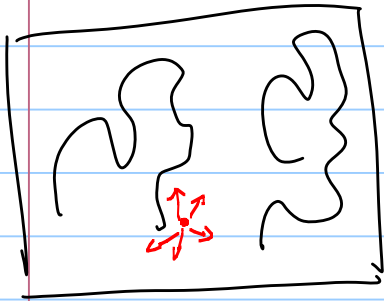
5. The move is accepted with prob.

$$\text{acc}(o \rightarrow n) = \min\left(1, \frac{W(n)}{W(o)} e^{-\beta[U(n)^{\text{pos}} - U(o)^{\text{pos}}]}\right)$$

### Example 2

Configuration-biased New configuration

for chain molecules with length  $l$



1) Generate a trial configuration using the Rosenbluth scheme

a) 1<sup>st</sup> atom is inserted at random its energy of  $U_1(n) \neq$

$$W_1(n) = k e^{-\beta U_1(n)}$$

$k$  is the coordinate #

in continuum, usually  $k \approx 6-10$ .

b) for the next segment, select from  $k$  possible trial directions with prob.

$$P_i(n) = \frac{e^{-\beta U_i(n)}}{W_i(n)}$$

$U_i(j)$  includes all interactions of  $i$

other molecules in the system & w/ segments 1 to  $i-1$ , but not  $i-1$  to  $l$ .

c) repeat (b) until the chain is grown. The total weight of  $n$  is

$$W(n) = \prod_{i=1}^l w_i(n)$$

We also calculate the weight of old configuration  $o$ . we

i) measure the energy of 1<sup>st</sup> atom

$$U_1(o) \quad \& \quad w_1(o) = k e^{-\beta U_1(o)}$$

ii) compute the Rosenbluth weights for the remainder of the chain

$$w_i(o) = e^{-\beta U_i(o)} + \sum_{j=2}^k e^{-\beta U_i(j)}$$

iii) retrace the entire chain w/ (ii)

$$W(o) = \prod_{i=1}^l w_i(o).$$

accept move from  $o \rightarrow n$  with prob

$$P = \min \left[ 1, \frac{W(n)}{W(o)} \right].$$