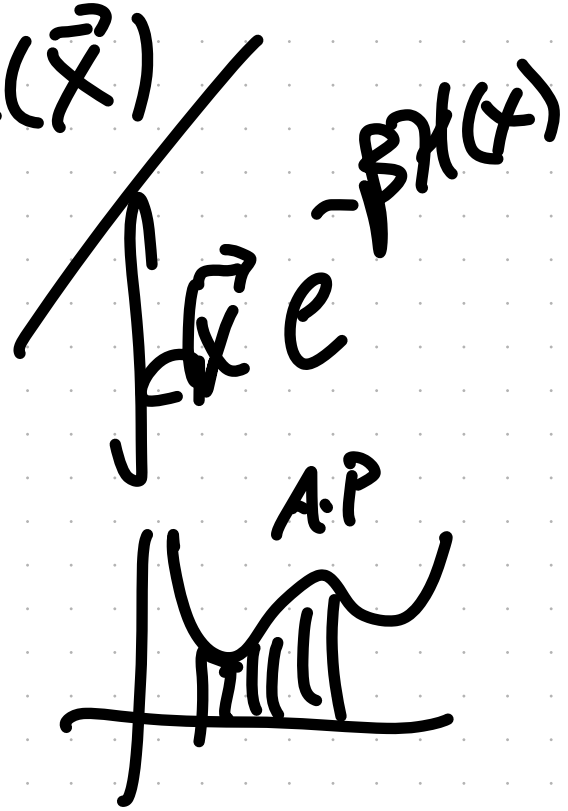
Intro to computer simulations

Goal: compute averages N, V, T

$$\langle A \rangle = \int d\vec{x}\, P(\vec{x})\, A(\vec{x})$$

mostly $P(\vec{x}) = e^{-\beta \mathcal{H}(\vec{x})}$

$$\beta = (k_B T)^{-1}$$

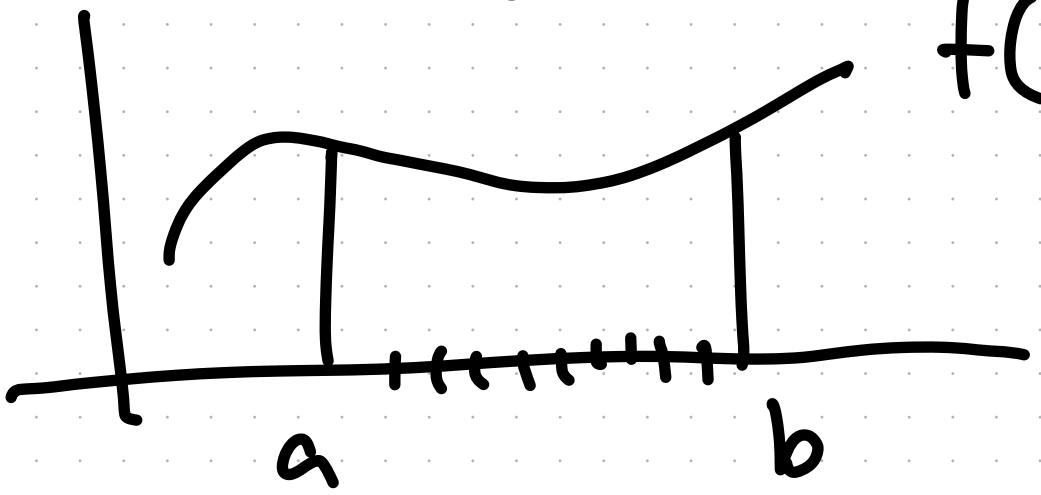


If we knew $P(\vec{x})$, in 1d

$$\langle A \rangle = \sum_{i=1}^{N} A(x_i)\, P(x_i)\, \Delta x_i$$

$\Delta x_i = x_{i+1} - x_i$

Solving by "quadrature"



$$\int_a^b f(x)\,dx$$

N^{th} points

$$\Delta x = \frac{b-a}{N} \qquad x_i = a + i\left(\frac{b-a}{N}\right)$$

$i \in 0$ to N

In not 1d but "d" dimensions

length L in each dimension $(b-a)$

$$\int dx \int dy \; P(x,y) A(x,y)$$

$$\approx \Delta x \Delta y \sum_{i,j} P(x_i, y_j) A(x_i, y_j)$$

$$x_i, y_j = \begin{cases} x_{min} + i \dfrac{x_{max} - x_{min}}{N_x} \\ y_{min} + i \dfrac{y_{max} - y_{min}}{N_y} \end{cases}$$

$$N_{bins} = \frac{L}{\Delta x}$$

$$N_{bins}^d = \left(\frac{L}{\Delta x}\right)^d =$$

$$e^{d\left(\ln\left(L/\Delta x\right)\right)}$$

In a stat mech problem

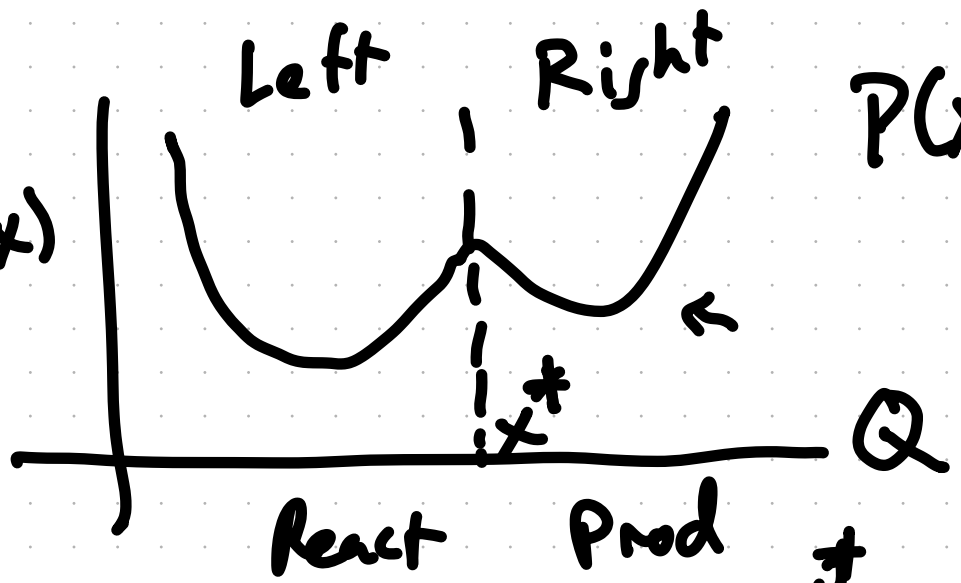
$$\langle A \rangle = \int d\vec{x}\, P(\vec{x})\, A(\vec{x})$$

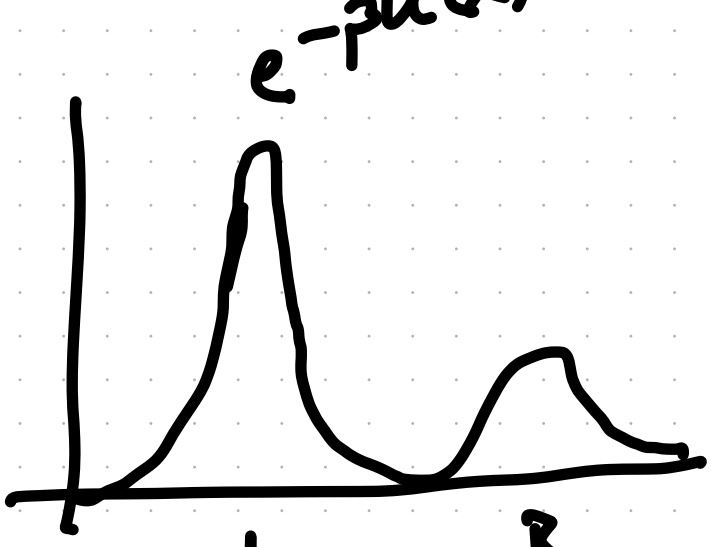
$\vec{x}$ is in dim $d \sim 6000 \cdot 6N_A$

$$\vec{x} = \{\vec{r}_1, \vec{r}_2, \ldots \vec{r}_N, \vec{p}_1, \vec{p}_2 \ldots, \vec{p}_N)$$

$6N$ degrees of freedom

in 1d





"A" $A(x) = \begin{cases} 1 & x < x^* \\ 0 & x > x^* \end{cases}$

"indicator function"

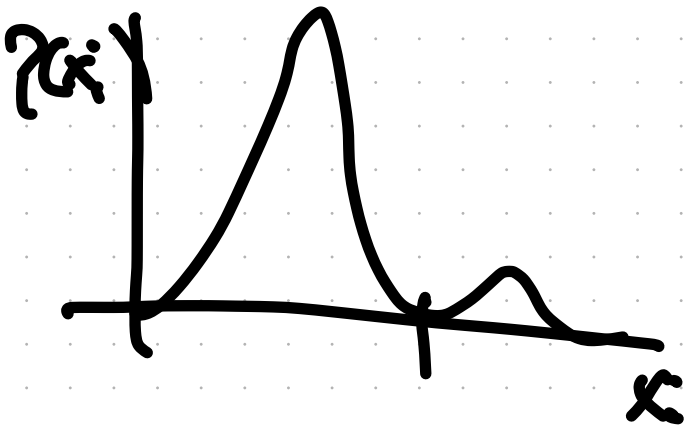
Simulations:

can't do integration by quadrature

instead: integration by sampling

$\langle A \rangle$ a bunch of samples of $\vec{x}_i$, if $\vec{x}_i$ appears with prob $P(\vec{x}_i)$ then

$$\langle A \rangle \approx \frac{1}{N} \sum_{i=1}^{N} A(\vec{x}_i)$$



Generate "Markov Chain"

Markov chain:

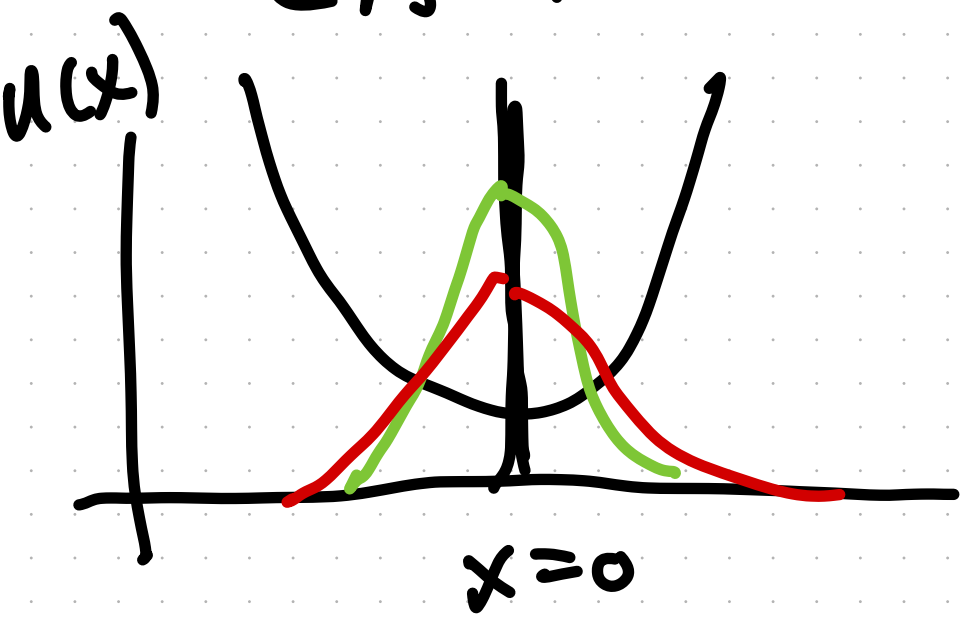
rule takes $x_i \to x_{i+1}$

only depends on x_i ← Markovian

To sample from $P(x)$, also ensure "detailed balance"

$$P(x_i)\,P(x_i \to x_{i+1}) = P(x_{i+1})\,P(x_{i+1} \to x_i)$$

If enforce detailed balance then distribution of $\vec{x}$ converges to $P(\vec{x})$



initial $P_0(x) = \delta(x)$

simulate

at long time $P(x) = \frac{e^{-\beta u(x)}}{Z}$

Detailed Balance

want: $P(x_i)\,P(x_i \to x_{i+1}) = P(x_{i+1})\,P(x_{i+1} \to x_i)$

$$P(x_i \to x_{i+1}) = P_{gen}(x_i \to x_{i+1})\,P_{acc}(x_i \to x_{i+1})$$

$$P(x_i)\,P_{gen}(x_i \to x_{i+1})\,P_{acc}(x_i \to x_{i+1})$$

$$= P(x_{i+1})\,P_{gen}(x_{i+1} \to x)\,P_{acc}(x_{i+1} \to x_i)$$

$$P_{acc}(x_i \to x_{i+1}) = \frac{P(x_{i+1})\,P_{gen}(x_{i+1} \to x)\,P_{acc}(x_{i+1} \to x_i)}{P(x_i)\,P_{gen}(x_i \to x_{i+1})}\, r(x_i \to x_{i+1})$$

Metropolis:

$$P_{acc}(\vec{x}_i \to \vec{x}_{i+1}) = \min(1, r(\vec{x}_i \to \vec{x}_{i+1}))$$

$$P_{acc}(x_i \to x_{i+1}) = \frac{P(x_{i+1})\, P_{gen}(x_{i+1} \to x)\, P_{acc}(x_{i+1} \to x_i)}{P(x_i)\, P_{gen}(x_i \to x_{i+1})}$$

$$r(x_i \to x_{i+1})$$

$$\frac{P_{fwd}}{P_{bckwd}} = \begin{cases} r/1 \\ 1/(1/r) \end{cases} = r = e^{-\beta \Delta H}$$

$$P(x) = e^{-\beta \mathcal{H}(x)}/Z \to \frac{P(x_{i+1})}{P(x_i)} = \frac{e^{-\beta \mathcal{H}(x_{i+1})}}{e^{-\beta \mathcal{H}(x_i)}}$$

for molecular system

$$x_{i+1} = x_i + \delta \cdot r$$

$$r \in (-1, 1)$$

another example

swap 2 positions

Algorithm:

0: initial configuration $= x_0$

1) propose x_{i+1} from x_i $\quad P_{gen}(x_i \to x_{i+1})$

2) random number $\tilde{r} \in (0, 1)$

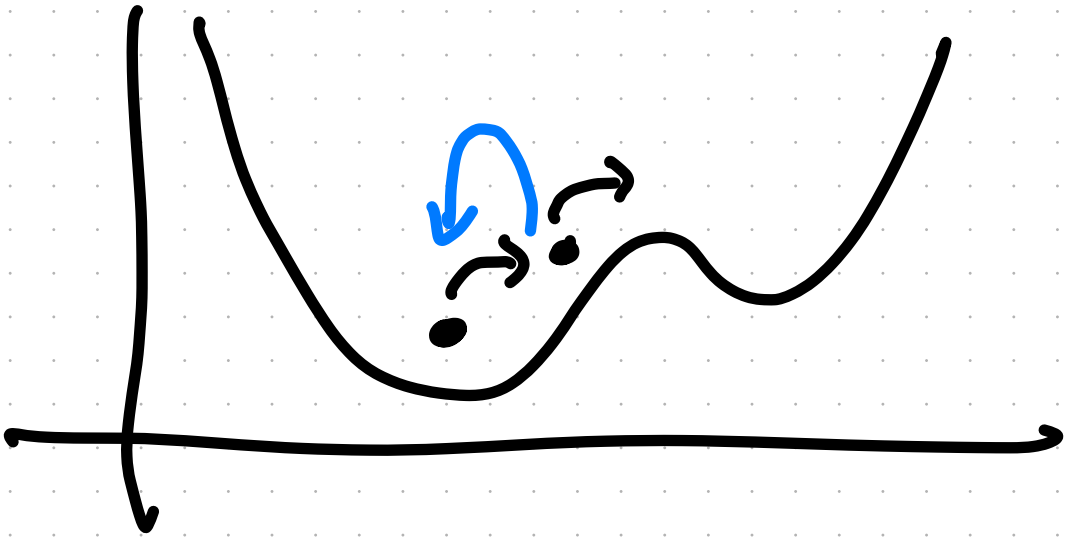
if $\tilde{r} <$ $\quad r(x_i \to x_{i+1})$: keep x_{i+1}

else: $\quad x_{i+1} = x_i$

$1 \to 2 \to 2 \to 2 \to 2 \to 3 \to 2 \to 2 \to 1$

for Boltzmann!

$$r(x_i \to x_{i+1}) = \min\left\{1, e^{-\beta(H(x_{i+1}) - H(x_i))}\right\}$$

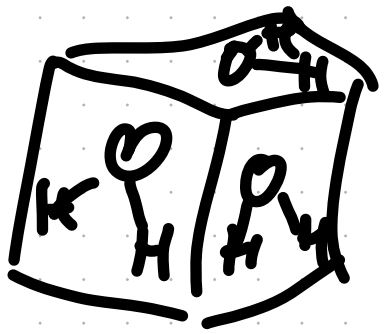


Summarize!

Monte is a powerful & general method

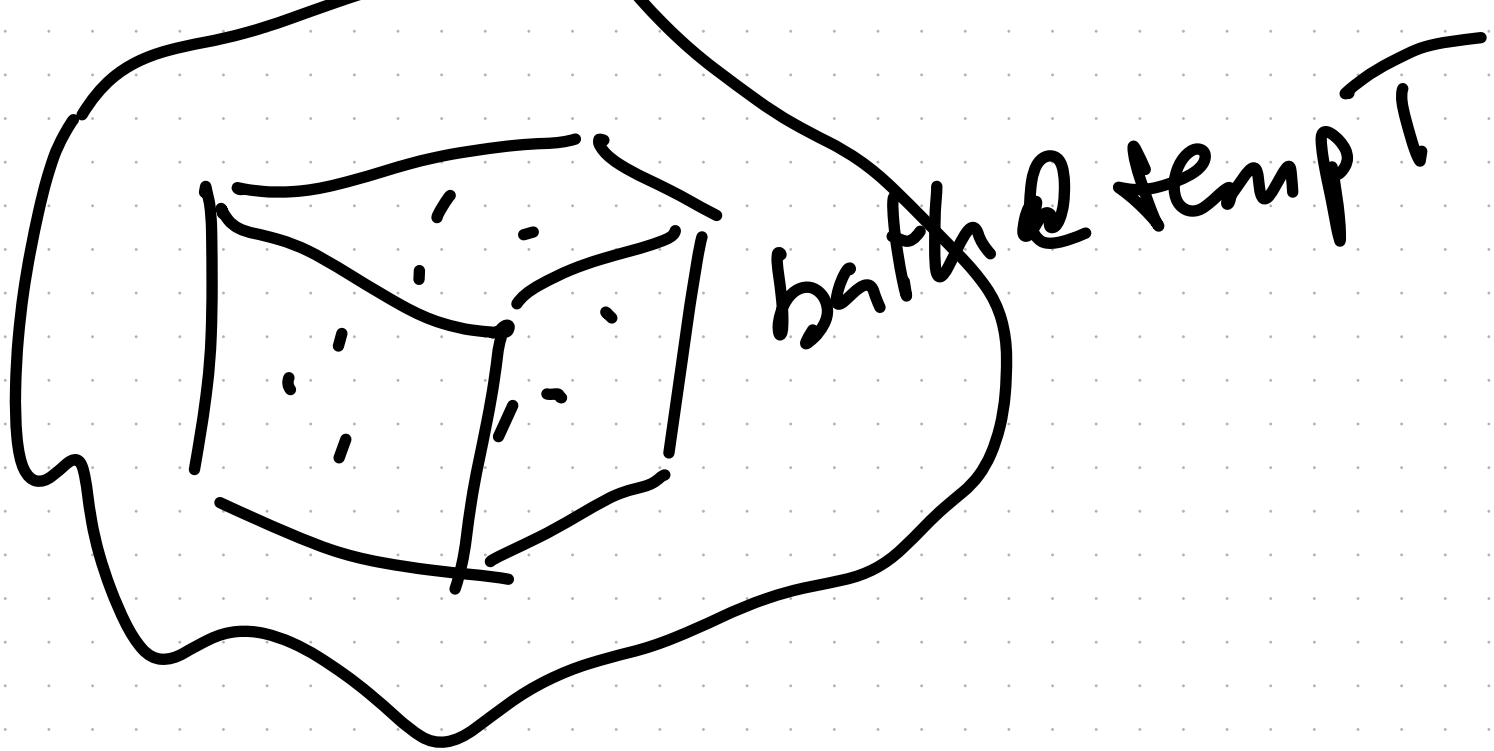
downsides:

① 1 thing happens at a time



reject a lot of moves

② gives static properties most of the time


bath @ temp T