• Sampling strategies

Goal: find the best (most efficient) way to obtain $N_{samples}$ independent configurations of the trajectory $\{x(\tau)\}$, distributed according to the probability $P[x(\tau)] = \frac{1}{Z} \exp\{-S_E[x(\tau)]\}$

Four basic strategies

① Heat bath algorithm → Useful when you can sample directly using a readily available random number generator (RNG).

Note: • People use this all the time in lattice QCD to generate "pseudofermions". More on this later!

• This method can be used for certain simple pure gauge theories (see Refs. in lecture 6)

In general the probability is a very complicated function, so we need more general strategies.

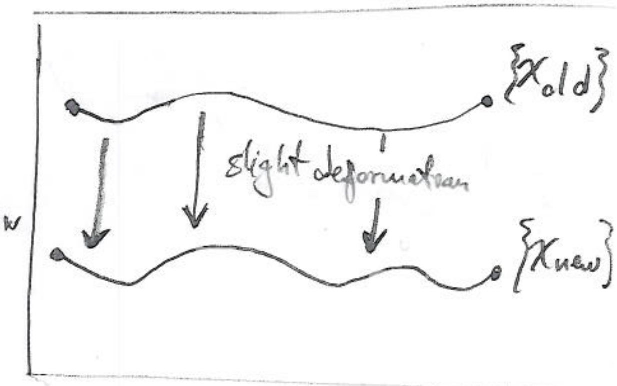
② Metropolis algorithm → This is so generic that almost every non-heat-bath method involves this algorithm.

If you've done an Ising model simulation, then you surely know about this.

Before going into the details of why this algorithm works, let's try to understand the recipe (i.e. how it works).

- Metropolis recipe (I mean... algorithm!)

0. Pick a random configuration $x_{old}$
1. Pick a tentative new configuration $x_{new}$




2. Compute

$$q = \exp\left\{-\left(S_E[x_{new}] - S_E[x_{old}]\right)\right\}$$

3. Pick a uniform random number $\xi \in [0,1]$.
4. If $\xi < q \longrightarrow$ set $x_{old} = x_{new}$ (updating!)

   If $\xi > q \longrightarrow$ retain $x_{old}$, discard $x_{new}$.
5. Go back to step 1.

- Claim: • After a large enough # of iterations the "current" $x_{old}$ configuration will be distributed according to $P = \dfrac{e^{-S_E}}{Z}$.

- Comments: (Ask first!)
  - When will $x_{new}$ be accepted regardless of the value of $\xi$?
    Answer: when $S_E(\text{new}) < S_E(\text{old})$
  - What happens if $S_E$ is not bounded from below? What are the consequences for $P$ in that case?
    Answer: • the simulation will "run away"; it becomes unstable
    • $P$ is not bounded from above in that case.

- Why does this algorithm work in practice?

It works because it creates a Markov chain that is

- Ergodic : a-priori there is no prefered or forbidden configurations, so all of config. space will be explored if we wait long enough.

- Reversible :

$$\frac{e^{-S_E[x_{old}]}}{Z} \cdot W(x_{old} \to x_{new}) = \frac{e^{-S_E[x_{new}]}}{Z} W(x_{new} \to x_{old})$$

With these two conditions one can show that $e^{-S_E}$ is an equilibrium distribution and that the equilibrium is stable. Let's prove the first.

- Equilibrium :

$$P[x_{new}] = \int \mathcal{D}x_{old} \frac{e^{-S_E[x_{old}]}}{Z} W(x_{old} \to x_{new}) \overset{\text{Reversibility}}{=} \frac{e^{-S_E[x_{new}]}}{Z} \underbrace{\int \mathcal{D}x_{old} \, W(x_{new} \to x_{old})}_{1} = \frac{e^{-S_E[x_{new}]}}{Z}$$

probability that we get configuration $\{x_{new}(\tau)\}$

assuming we are already sampling according to $\frac{e^{-S_E[x]}}{Z}$

by ergodicity

We obtain the same distribution!

- Is Metropolis reversible?

– If $S_E[y] < S_E[x]$, then the move $x \to y$ is accepted, and $y \to x$ is accepted w/ probability $q = e^{-(S_E[x] - S_E[y])}$ (Notice we are looking at $y \to x$, not $x \to y$!)

$$\therefore \quad \frac{W(x \to y)}{W(y \to x)} = \frac{1}{q} = \frac{e^{-S_E[y]}}{e^{-S_E[x]}} \qquad \text{QED.}$$

- Using the Metropolis algorithm

- Always remember, we need to wait for a # of steps so that we reach
  - Equilibration (a.k.a. thermalization, before taking the first sample)
  - Decorrelation (i.e. wait between samples so they are independent).

- How do we decide on the changes when going from $x_{old} \to x_{new}$?
  - Big changes $\to$ Big changes in $S_E$ $\longrightarrow$ Better decorrelation, but acceptance rate drops !
  - Small changes $\to$ Better acceptance rate, but decorrelation time increases... :(

Compromise: Make changes such that acc. rate $\simeq 1/2$.

- Problems of using Metropolis alone.

Perhaps the most important disadvantage of using the Metropolis algorithm without any extra concept is the following.

Actions tend to be complicated functions of the field variables, especially when fermions are present, in which case the action is <u>very</u> non-linear and <u>very</u> non-local.

As a consequence, only small localized changes are possible (if we do this randomly), or the acceptance rate will drop dramatically.

→ <u>We need algorithms that can perform global changes.</u>
More specifically, we need an updating strategy that does not destroy the acceptance rate, but makes changes everywhere.

(i.e. there is nothing wrong with Metropolis itself,
rather the updating strategy $x_{old} \to x_{new}$ is the problem)