Monte Carlo Simulations

Much of our previous discussion has referred to numerical methods to evaluate the path integral

$$\langle O \rangle = \frac{1}{Z} \int DU \; O[U] \exp -S[U]$$

Although in principle $\int DU$ is defined rigorously, in practise it consists of many many integrals ($O(N^4)$) and hence can't be evaluated.

One could proceed to evaluate the integral by randomly choosing configurations $\{U\}$, evaluating $S[U]$, and weighting the contributions to O:

$$\langle O \rangle = \frac{\sum_{\{U\}} O[U] \exp -S[U]}{\sum_{\{U\}} \exp -S[U]}$$

Such a procedure would not sample phase space efficiently, and would converge very slowly if at all.

A better plan is IMPORTANCE SAMPLING of phase space: ie. sample configurations with probability "built-in"

Suppose we generate configurations to form an ensemble with probability $\sim \exp -S[U]$

Then $$\langle O \rangle = \frac{1}{N_{config}} \sum_{\{U\}} O[U]$$

if probability normalised correctly, then $Z = 1$

$\Rightarrow$ generate estimates for $\langle O \rangle$ with errors $\delta O \sim N_{cor.}^{-\frac{1}{2}}$

How do we generate the ensemble? It is the end point of a MARKOV PROCESS, by which the configuration is evolved step by step

Define $P_{eq}(\{U\})$ to be desired equilibrium distribution $\propto e^{-S[U]}$

$\mathbf{P}(\{U'\}, \{U\})$ probability for config $\{U'\}$ to be generated from $\{U\}$

we require: positivity $P(\{u'\},\{u\}) \geq 0$

normalisation $\sum_{\{u'\}} P(\{u'\},\{u\}) = 1$

+ Ergodicity $P(\{u'\},\{u\}) > 0$

Now if $p_n(\{u\})$ is probability of $\{u\}$ after n steps of Markov process, then

$$p_{n+1}\{u'\} = \sum_{\{u\}} P(\{u'\},\{u\})\, p_n(\{u\})$$

is probability of $\{u'\}$ after $n+1$ steps.

A sufficient (not necessary) condition for the Markov process to converge on the correct distribution p_{eq} is DETAILED BALANCE:

$$P(\{u'\},\{u\})\, p_{eq}(\{u\}) = P(\{u\},\{u'\})\, p_{eq}(\{u'\})$$

Proof: define the "distance between distributions"

$$\|p - p'\| = \sum_{\{u\}} |p(\{u\}) - p'(\{u\})|$$

so: $$\|p_{n+1} - p_{eq}\| \equiv \sum_{\{u\}} |p_{n+1}(\{u\}) - p_{eq}(\{u\})|$$

$$= \sum_{\{u\}} \Big| \sum_{\{u'\}} P(\{u\},\{u'\}) \big(p_n\{u'\} - p_{eq}(\{u'\})\big) \Big|$$

see attached sheet

since normalisation

$$p_{eq}(\{u\}) = \sum_{\{u'\}} P(\{u'\},\{u\})\, p_{eq}(\{u\})$$

$$= \sum_{\{u'\}} P(\{u\},\{u'\})\, p_{eq}(\{u'\})$$ by detailed balance

Triangle inequality + positivity

$$\Rightarrow \|p_{n+1} - p_{eq}\| \leq \sum_{\{u\}\{u'\}} P(\{u\},\{u'\})\, |p_n(\{u'\}) - p_{eq}(\{u'\})|$$

inequality is strict from ergodicity unless $p_n = p_{eq}$

normalisation $\Rightarrow \|p_{n+1} - p_{eq}\| \leq \|p_n - p_{eq}\|$ $\Rightarrow$ converge to equilibrium probability distribution

www.qpt.tu.it

Union Flag (sorry!)

The Mountain
↳ Nature Walks
Winter Activities
Tridentin Alpinist Society

$$\| p_{n+1} - p_{eq} \| = \sum_{\{u\}} | p_{n+1}(\{u\}) - p_{eq}(\{u\}) |$$

$$= \sum_{\{u\}} \left| \left(\sum_{\{u'\}} P(\{u\},\{u'\})\, p_n(\{u'\}) \right) - p_{eq}(\{u\}) \right|$$

$$= \sum_{\{u\}} \left| \sum_{\{u'\}} \left(P(\{u\},\{u'\})\, p_n(\{u'\}) - P(\{u'\},\{u\})\, p_{eq}(\{u\}) \right) \right|$$

← normalization

← detailed balance

$$= \sum_{\{u\}} \left| \sum_{\{u'\}} P(\{u\},\{u'\}) \left(p_n(\{u'\}) - p_{eq}(\{u'\}) \right) \right|$$

$$| \alpha \vec{a} + \beta \vec{b} + \dots | \leq (\alpha, \beta, \dots)\ |\vec{a}| + |\vec{\beta}| + \dots$$

positivity + Triangle inequality

$$\leq \sum_{\{u\}\{u'\}} P(\{u\},\{u'\})\, | p_n(\{u'\}) - p_{eq}(\{u'\}) |$$

$$\circledast = \sum_{\{u'\}} | p_n(\{u'\}) - p_{eq}(\{u'\}) | = \| p_n - p_{eq} \|$$

ie $\| p_{n+1} - p_{eq} \| \leq \| p_n - p_{eq} \|$

Most practical algorithms implement changes to configurations by a succession of small changes, generated using pseudo-random numbers.

Detailed balance: $$\frac{P(\{u'\};\{u\})}{P(\{u\},\{u'\})} = \frac{p_{eq}(\{u'\})}{p_{eq}(\{u\})} = \exp(-\Delta S(\{u'\},\{u\}))$$

where $\Delta S(\{u'\},\{u\})$ is change in action when configuration $\{u\}$ is changed to $\{u'\}$

For small changes to local actions, ΔS is a "cheap" function to evaluate. (For LGT it's a sum of "staples" )

Eg. Metropolis algorithm: $P(\{u'\},\{u\}) \propto \min[1, \exp(-\Delta S)]$

"Try a change: if it reduces the action, accept it
if it raises the action, accept with probability $e^{-\Delta S}$"

To be efficient, an algorithm should have a high rate of acceptance of changes, and yet not "linger" in just one corner of phase space.

— A huge amount of progress in recent years, particularly in "non-local" updating schemes

Two things to think about: (eg. discussion of non-zero T)

(i) Finite volume effects.

Ideally we simulate in a regime $a \ll \xi a \ll La$, where L is the dimension of the box we simulate.

Clearly there are practical limitations on L — hence on approaches to criticality

(ii) Critical Slowing Down.

As we approach criticality, the simulation must run for longer to decorrelate the system (otherwise our estimates for $\langle O \rangle$ are not statistically independent)

It takes $\sim n^2$ steps for local updates to diffuse outwards and affect physics at n lattice spacings away

$\Rightarrow$ decorrelation "time" $\tau_d \sim \xi^2$ ← identify dynamical critical exponent. z

$\Rightarrow$ take configurations $\{U\}$ from an ensemble generated with probability $\frac{1}{Z}\exp(-S[U])$. Suppose we have N statistically independent configurations

$$\Rightarrow \quad \langle O \rangle_{est} = \frac{1}{N} \sum_{i=1}^{N} O[\{U\}_i]$$

$$\lim_{N\to\infty} \langle O \rangle_{est} = \langle O \rangle_{true} \pm \frac{\text{const.}}{\sqrt{N}}$$

↑ statistical error

The size of the constant depends in part on how successful we are in generating independent configurations, & in part on how intrinsically noisy the signal is — eg. we would expect much greater noise measuring $\langle \sigma_i \rangle$ in the Ising model if we are near the critical point.

Note that due to finite cpu & memory, all "simulations" are necessarily performed on finite systems

$\Rightarrow$ important to check this is not a source of systematic error ie. on a lattice of spatial size N, we need

$$a \ll \xi \ll Na$$

$\Rightarrow$ Usually use translationally invariant boundary conditions

either periodic boundary conditions $U_\mu(x+N) = U_\mu(x)$

or anti periodic " " $U_\mu(x+N) = -U_\mu(x)$