Monte Carlo simulations

Monte Carlo integration

$$\int_a^b dx\, f(x) \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

x_i taken from a uniform distribution $[a, b]$

error $\sim \frac{1}{\sqrt{N}}$

Why would that be useful?

Typical numerical integration routines: error $\sim h^k$ $\quad k \geq 1$

$N = \frac{L}{h} \Rightarrow$ error $\sim \frac{1}{N^k}$

(i.e. always better than $\frac{1}{\sqrt{N}}$)

Useful: high-dimensional integrals:

example L^d hypercube

error $\sim h^k$, but now $N = \left(\frac{L}{h}\right)^d \Rightarrow$ error $\sim 1/N^{k/d}$

Monte Carlo: still $\sim \frac{1}{\sqrt{N}}$, independent of d !!

Example of high-dimensional integral in physics

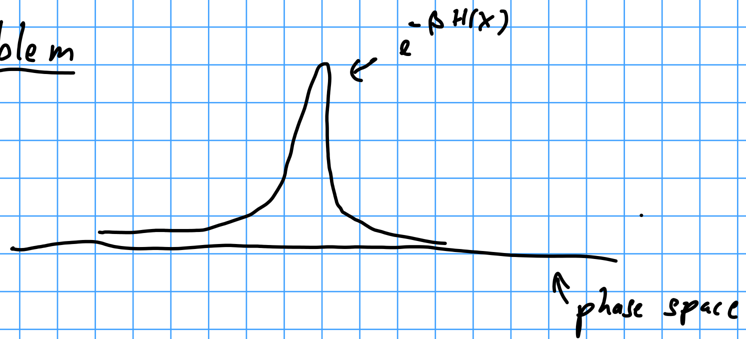
Statistical mechanics

$$\langle A \rangle = \frac{1}{Z} \int dR\, e^{-\beta H(R)} A(R) \qquad Z = \int dR\, e^{-\beta H(R)}$$

e.g. particle coordinates, spins, etc.

$\rightarrow$ huge number of integration variables

Problem



uniform sampling
very inefficient

Solution: importance sampling

$$\int dX \; p(X)\, A(X) \approx \frac{1}{N} \sum_{i=1}^{N} A(X_i)$$

$p(X)$: positive, normalized to 1

X_i sampled according to probability $p(X)$

How to sample $p(X)$?

Metropolis algorithm

Use a Markov chain to sample $p(X)$

Markov chain: set of random states $\{X_i\} = X_1, X_2, X_3, \ldots$ where the probability of X_{i+1} depends on X_i only:

$$T(X \to X'), \qquad \sum_{X'} T(X \to X') = 1$$

(you have to go somewhere from X)

We can thus write a rate equation for probabilities:

$$p(X, i+1) = p(X, i) - \sum_{X'} p(X, i)\, T(X \to X') + \sum_{X'} p(X', i)\, T(X' \to X)$$

We want a stationary probability $\to p(X_{,i+1}) = p(X, i) = p(X)$

$$\Rightarrow \sum_{X'} p(X)\, T(X \to X') = \sum_{X'} p(X')\, T(X' \to X)$$

Fulfilled by $$p(x)\,T(x \to x') = p(x')\,T(x' \to x)$$
detailed balance

$\Rightarrow$ to get $p(x)$, we need to choose $T(x \to x')$ such that detailed balance is fulfilled.

separate $T(x \to x') = \omega_{xx'} \cdot A_{xx'}$

probability to try to go from x to x'

probability to accept x' if system was in x

In many cases we have $\omega_{xx'} = \omega_{x'x}$ (this must be checked!)

$$\Rightarrow \quad \frac{A_{xx'}}{A_{x'x}} = \frac{p(x')}{p(x)}$$

solution: $A_{xx'} = 1$ if $p(x') > p(x)$

$A_{xx'} = p(x')/p(x)$ if $p(x') < p(x)$

Summary Metropolis algorithm:

1. Start with a state x_i
2. generate a state x' from x_i (such that $\omega_{x_i x'} = \omega_{x' x_i}$)
3. If $p(x') > p(x)$, $x_{i+1} = x'$

   If $p(x') < p(x)$

   set $x_{i+1} = x'$ with probability $\frac{p(x')}{p(x)}$ (accept move)

   $x_{i+1} = x_i$ with probability $1 - \frac{p(x')}{p(x)}$ (reject move)
4. Continue with 2.