Lecture 4

Metropolis Algorithm & Simulated Annealing

Now we can use stochastic methods to estimate the calculations of multidimensional integrals, and even reducing the statistical error of these approximations by means of importance sampling, where we generate random variables according to a specific PDF (Gaussian, uniform, etc.)

But, how can we draw samples from a **generic** probability distribution?

This can be done by generating a random walk with a specific transition probability $T(R \to R')$. For that, a *detailed balance condition* is sufficient. Namely,

$$P(R)T(R \to R') = P(R')T(R' \to R) \tag{4.1}$$

so that the probability of being in R and transitioning to R' is symmetric to the opposite flow. This generates a random walk $R_1, R_2, \ldots, R_{N_{iter}}$ according to PDF $P(R)$.

Which is a simple choice for the transition probability such that the detailed balance condition in Equation 4.1 is fulfilled? A simple choice is

$$T(R \to R') = min\left[1; \frac{P(R')}{P(R)}\right]$$

In other words, if we move to a place where the probability is higher, then we accept it with probability 1 (always). However, if $P(R') < P(R)$ then we will only accept this transition with probability $P(R')/P(R)$.

Metropolis Algorithm
Metropolis consists in generating a new trial position R' starting from the current position R. Then, we calculate the weight defined as

$$\omega = \frac{P(R')}{P(R)}$$

so that, when we generate a random number $0 < u < 1$ according to a uniform distribution,

- we accept the new state $R_{i+1} = R'$ if $int(\omega + u) > 0$, or
- keep the old one $R_{i+1} = R$ if $int(\omega + u) = 0$.

Some observations:

1. The first step is to generate a move, that is, displacing R by a random value:

$$R' = R + \xi$$

where ξ can be sampled from a uniform probability distribution, such that $\xi = a(2u' - 1),\ 0 < u' < 1$, a being the amplitude of the displacement.

But it can also be sampled from a normal distribution

$$p(\xi) = \frac{1}{\sqrt{2\pi}a} e^{-\frac{\xi^2}{2a^2}}$$

where now a is the standard deviation.

2. Which value of a should be used? For that, we can calculate the acceptance rate defined as

$$P_{\text{accept}} = \frac{N_{\text{accepted}}}{N_{iter}}$$

which will always be $0 < P_{\text{accept}} \leq 1$. For example, a $P_{\text{accept}} \lesssim 0.1$ is a low acceptance rate. And $P_{\text{accept}} \gtrsim 0.9$ is a high acceptance rate.

Simulated Annealing

This method is used to optimize the search for a global minimum. It can be used with a multivariable function where the function might have a large number of local minimum, where *steepest descent* Newton method might not be efficient.

A more efficient method should jump over the barriers that separate local minima, and jumping to "deeper" wells should be more probable than the opposite.

Annealing method

Consider the function $E(x_1, \ldots, x_N)$ that should be minimized. We can introduce a fictitious "temperature" T that we will use to define the following PDF

$$P(x_1, \ldots, x_N) = e^{-\frac{E(x_1, \ldots, x_N)}{T}}$$

Then, Metropolis algorithm is used to draw samples from it. However, every iteration the temperature decreases by a factor γ:

$$T_{i+1} = T_i \cdot \gamma$$

where γ is usually close to 1 (e.g., 0.999).

Essentially, when the temperature is high initially we are able to "jump" over wells easily, but as we perform more iterations, the temperature gradually decreases so the new trial states are less far apart until we settle to the closest minimum (as seen in Figure 4.1).

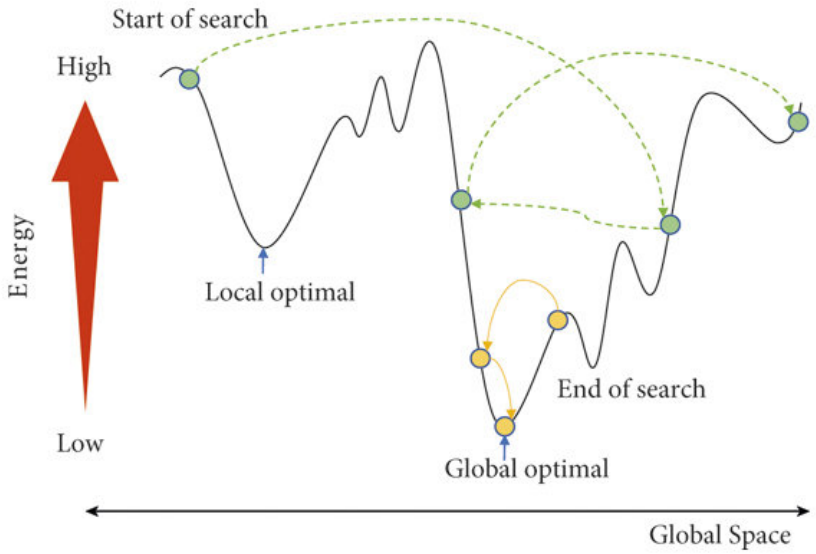

Figure 4.1: Simulated Annealing method diagram.

4.1 Exercise 4: Global minimum of Thompson atom

Consider the energy of an atom given by the expression

$$E = \sum_{i=1}^{N} \frac{1}{2} m\omega_i^2 r_i^2 + \sum_{i<j}^{N} \frac{e^2}{|\vec{r}_j - \vec{r}_i|}$$

with the left term corresponding to a harmonic term, and the latter the Coulomb potential one.

In dimensionless form, it can be written

$$\tilde{E} = \sum_{i=1}^{N} r_i^2 + \sum_{i<j}^{N} \frac{1}{|\vec{r}_j - \vec{r}_i|}$$

Defining a dimensionless temperature $\tilde{T}$, find the optimal positions for $N = 5, 20, 26$ charges in two dimensions.

To generate new states (set of positions for each particle) we have 2 options:

1. "Global" move, in which we move all the particles at once randomly.

$$x_i' = x_i + \xi_i^x$$
$$y_i' = y_i + \xi_i^y$$

   for $i = 1, \ldots, N$. And ξ can be generated according to Metropolis algorithm.

2. "Local" move, that involves moving a single particle when generating a new state.

$$x_i' = x_i + \xi_i^x$$
$$y_i' = y_i + \xi_i^y$$

only for the i-th particle, the rest are kept the same. To decide which particle to move, we can also generate a random variable between $[1, N]$.