

Tutorial

10: Monte Carlo Simulation of Polymers

Baofu Qiao*

July 5, 2007

SimBio group, FIAS, Frankfurt

Contents

1	Introduction	1
2	Monte Carlo Simulation of polymers	1
3	Links	3

1 Introduction

Because the random walk in one dimension and two dimensions have been discussed in the 2nd tutorial, and so is the self-avoiding walk, they are not repeated here. In the present tutorial, only the three dimensional case is discussed.

2 Monte Carlo Simulation of polymers

2.1 Simple Monte Carlo method

The sample Monte Carlo method involves the following steps:

1. Obtain a configuration of the system by randomly generating $3N$ Cartesian coordinates which are assigned to the N particles.
2. Calculate the potential energy of the configuration, $U(r^N)$, and the Boltzaman factor,

$$\exp\left(-\frac{U(r^N)}{k_B T}\right)$$

*qiao@fias.uni-frankfurt.de

3. Add the Boltzmann factor to the accumulated sum of Boltzmann factors and the potential energy contribution to its accumulated sum, and then return to step 1.
4. After a number, N , of iterations, the mean value of the potential energy would be calculated using:

$$\langle U(r^N) \rangle = \frac{\sum_{i=1}^N U(r^N) \exp\left(-\frac{U(r^N)}{k_B T}\right)}{\sum_{i=1}^N \exp\left(-\frac{U(r^N)}{k_B T}\right)} \quad (1)$$

However, this is not a feasible approach because there exist large number of high-energy configurations whose Boltzmann factor is extremely small. Some smarter methods have been proposed, such as Rosenbluth-Rosenbluth method, Metropolis Monte Carlo method, etc.

2.2 Metropolis Monte Carlo method

The main feature of the Metropolis Monte Carlo method is that it biases the generation of the configurations towards those that make the most significant contribution. It generates states with a probability $\exp\left(-\frac{U(r^N)}{k_B T}\right)$ and then counts each of them equally. While the simple Monte Carlo method generates states with equal probability (both high and low energy) and then assigns them a weight $\exp\left(-\frac{U(r^N)}{k_B T}\right)$.

If a new configurations is lower in energy than its predecessor, the new configuration is accepted as the starting point for the next iteration. If the new configuration is higher in energy than its predecessor, the Boltzmann factor, $\exp\left(-\frac{\Delta U}{k_B T}\right)$ is compared to a random number between 0 and 1. If the Boltzmann factor is bigger than the random number, the new configuration is accepted, if no, rejected.

$$rand(0, 1) \leq \exp\left(-\frac{\Delta U}{k_B T}\right) \quad (2)$$

where $\Delta U = U_{new} - U_{old}$

2.3 Tasks

1. Understand the codes. Use $SAW=0$ to study the case of random walk of polymer, and $SAW=1$ for self-avoiding walk of polymer. Calculate the parameter ν in the two cases according to the following equation:

$$\langle Re^2 \rangle \propto N^\nu \quad (3)$$

Table 1: Mean square end-to-end distance of 3D self-avoiding walk and random walk for varying polymer length N .

N	10	20	50	100	200	500	1000
$\langle Re^2 \rangle_{SAW}$							
$\langle Re^2 \rangle_{random}$							

2. Turn on the periodic boundary condition, $PBC=1$, to study the effect of periodic boundary condition.
3. In the present code, the simple Monte Carlo method is used. Turn on the subprogram, *neighbour.c* to calculate the nearest neighbour in the system. Expand the codes to implement the Metropolis Monte Carlo method.

3 Links

3.1 Self-Avoiding Walk

- <http://polymer.bu.edu/java/java/saw/saw.html>

3.2 Rosenbluth-Rosenbluth factor

- See: M. N. Rosenbluth and A. W. Rosenbluth, *J. Chem. Phys.* **1955**, 23(2):356-359.