Monte Carlo simulation of a polymer

Niels Kolen, 4037936, and Stefani Peters, 1513575

Technical University of Delft

April 9, 2016

Abstract

In this project a Monte Carlo simulation of a polymer was written. A population of polymers was grown by adding monomers. The interaction between monomers was modeled with a Lennard-Jones potential. Simulations were made using the pruned-enriched Rosenbluth method to investigate the scaling behavior of the end-to-end distance. This end-to-end distance is then compared with literature.

I. Introduction

The study of the behavior of polymers is a very interesting and important topic in statistical mechanics. These polymers can be studied as a melt or in a solution. Several computational methods based on Monte Carlo (short: MC) simulations already have been succesfully used, [1].

In this work the focus will lie on polymers that are situated in a dilute solution. This allows us to make a mesoscopic model of the polymer. This means that the polymer is a chain consisting of beads: the beads represent segments of the polymer; the segments in turn represent groups of atoms, [2]. These groups of atoms interact with each other through Van der Waals attraction. This interaction can be modelled by a Lennard-Jones potential, $V(r)$:

$$V(r) = 4\epsilon \left[(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^{6} \right] \tag{1}$$

where r is the distance between the particles, ϵ is the depth of the potential well and σ is the distance at which the inter-particle potential is zero. The parameters chosen for the Lennard-Jones potential are $\epsilon = 0.25$ and $\sigma = 0.8$, [2]. In this paper we want to compute the end-to-end distance.

In section II, we describe the steps taken to compute the end-to-end distance of a polymer. To calculate this a MC program in Python is written. In section III we give and discuss the results for the end-to-end distance that is computed in the program. In section IV the conclusion is drawn for the MC simulation of a polymer.

II. Methods

We consider a polymer of $N = 50$ beads in two dimensions. Because the beads interact according to the Lennard-Jones potential, equation (1), they will repel each other. And so we deal with a self-avoiding walk (SAW). To simulate a SAW there are a lot of different algorithms. The methods that we will discuss are the Rosenbluth method, section A and the pruned-enriched Rosenbluth method, section B. The Rosenbluth method can be used to generate the polymer in a certain configuration following a canonical distribution at a given temperature, T. For this experiment we chose $T = 1\,\mathrm{K}$. In section B we discuss the pruned-enriched Rosenbluth method. This method is an improvement on the Rosenbluth method. It removes bad configurations and replaces them with good ones. In section C we discuss how the end-to-end distance can be calculated.

A. Rosenbluth method

The Rosenbluth method consists of adding new segments or beads to the polymer, avoiding high-energy conformations. In this method we have two beads on positions (0,0) and (1,0). We start by adding new beads one by one. The position of the newly added bead can be defined by the angle, θ, with the x-axis and the position of the previous position. Before a bead

is placed, we discretise the space of possible θ-values to obtain a finite number of different values of θ. In our simulation we chose six different values for θ. We get a random offset spaced by $2\pi/6$. The weight, w, of these six θ-values numbered by j is $w_j^{(l)} = \exp[-E(\theta_j)/(k_B T)]$. Their sum is:

$$W^{(l)} = \sum_j \exp\left[-E(\theta_j)/k_B T)\right] \qquad (2)$$

where l is the bead added, $E(\theta)$ the interaction energy of the beads, k_B the Boltzmann constant and T the temperature. The scientific value of the parameter $k_B = 1.38 \cdot 10^{-21}\,\mathrm{J/K}$, but for this experiment we chose $k_B = 1\,\mathrm{J/K}$.

The final position of the bead is chosen with probability $w_j^{(l)}/W^{(l)}$. This creates a bias towards states with a lower energy. This process is then repeated for the remaining beads.

B. *Pruned-enriched Rosenbluth algorithm*

The problem with the Rosenbluth method is that the algorithm polymers produces with very low probability as well as with high probability. Especially with very long polymers. Grassberger, [3], offered a solution: the 'pruned-enriched Rosenbluth method' (PERM). Here he shows that populations evolve toward a more balanced distribution by removing the bad configurations and replace them with good ones which then evolve further individually. The removing of the bad configurations is called pruning and the copying of the good configurations is called enriching. Whether a configuration is good or bad is denoted with a weight factor. This weight factor is at the start of its growth 1. After every addition it is multiplied with the factor $W = \sum_j w_j$. When a configuration is removed or copied, the weight should not change. If the partially grown polymer gets a weight below a certain threshold, half of the time the polymer is removed and the other half of the time its weight is doubled and the growing continues. When a partially grown polymer gets a weight above a certain threshold, the polymer is split. Both these polymers can independently grow from this point on and their weights are halved.

C. *End-to-end distance*

For the pruned-enriched Rosenbluth method we will compute the end-to-end distance. As the name implies this is the distance between the first and last bead. This is a very useful quantity because the Rosenbluth method generates polymer populations for all lengths up to the maximum set in the simulation. In our case this is 50 beads. Because the polymer behaves like a SAW in two dimensions, the end-to-end length, R, scales with N as:

$$R \propto N^{\nu} \qquad (3)$$

where ν is an exponent and is 0.75 in two dimensions. The average squared end-to-end distance, $< R^2 >$, scales with N as in equation 3, where $\nu_{average}$ is an exponent and is 1.5 in two dimensions.

III. Results and discussion

A. *End-to-end distance*

In figure 1 we plot the end-to-end distance for the PERM method with respect to the number of beads. We also compare this end-to-end distance with the theoretical behavior.

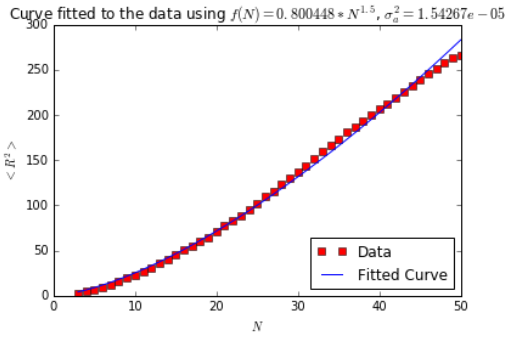

Figure 1: *A plot of the average squared end-to-end distance of a polymer population that grows to $N = 50$. At each step the population is pruned and enriched. The drawn line has the form $a \cdot N^{1.5}$, where a is chosen to fit the data.*

To see how many polymers are alive at each growth step in the simulation, we plot the number of alive polymers against the number of beads, figure 2. By changing the upper and lower limit, we can influence the population.

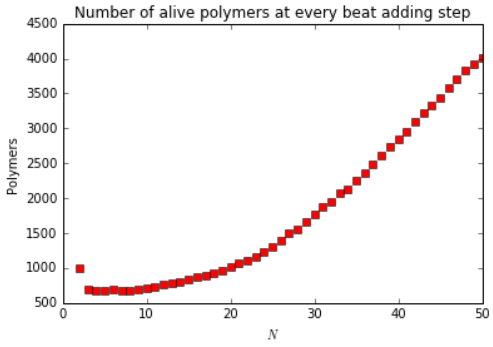

Figure 2: *A plot of the amount of polymer chains. At each step the population is pruned and enriched. By changing the upper and lower limit, we can influence the population.*

We can see that the PERM method scales the R according the theoretical result.

IV. Conclusion

The MC simulation written in python for the pruned-enriched Rosenbluth method for finding polymer configurations does find end-to-end data that corresponds with the theoretical relation.

References

[1] J. Baschnagel, J. P. Wittmer, H. Meyer, 'Monte Carlo simulation of polymers: coarse-grained models', *Computational Soft Matter: From Synthetic Polymers to Proteins. Lecture notes* (N. Attig, K. Binder, H. Grubmüller, K. Kremer, eds), vol. 23 of *NIC*, (Jülich), pp. 83-140, Johm von Neumann Institute for Computing, 2004.

[2] J.M. Thijssen, *Computational Physics*, Cambridge University Press, 2007.

[3] P. Grassberger, 'Pruned-enriched Rosenbluth method: Simulation of θ polymers of chain length up to 1 000 000,' *Phys. Rev. E.*, **56**, 3682 - 3693, 1997.