

Simulations of linear polymer chains in continuous space

Agustin Iñiguez and Eduardo Pavinato Olimpio*
ICCP - Delft University of Technology
(Dated: April 11, 2015)

In this work we discuss the computational study of mesoscopic linear polymer chain model under Lennard-Jones attraction potential. For this we used an off-lattice three-dimensional setup, implemented using the Rosenbluth method and the Prune-Enriched Rosenbluth Method (PERM), checking the Flory exponent ν . Furthermore we used the PERM implementation to obtain some preliminary results for the radius of gyration, comparing with those in literature.

INTRODUCTION

The simulation of the dynamics of linear polymer chains in solution has attracted a lot of interest over the years. For this purpose, several computational methods based on Monte Carlo simulations have been successfully used [1]. In general, these methods are applied in on- and off-lattice models, either in two or three dimensions and, depending on the conditions being studied, the potential energy included in the sampling changes. In this work, we focus our attention in off-lattice models for dilute polymers in bad solvent. This allows us to use a mesoscopic model in which we simulate the monomers (taken of unit length) as beads interacting through a Lennard-Jones potential:

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 + C_{\text{bias}} \right] \quad (1)$$

where we use $\sigma = 0.8$ following [2, 3]. The value of ϵ is defined through the non-dimensional variable $\epsilon/k_B T$ as in [2, 4]. Contrary to what is done in the previous cited references, we followed the idea in [1] and truncate the potential at twice the minimum of the potential, $r_{\text{cut}} = 2 \times 2^{1/6} \sigma$, including a small bias to make the potential zero at r_{cut} , $C_{\text{bias}} = 127/16384$.

In the following sections, how we implemented this model in an off-lattice three dimensional Monte-Carlo, comparing the results obtained using Rosenbluth method and Prune-Enriched Rosenbluth Method (PERM) for the conformational parameter ν (called Flory exponent). Furthermore, we discuss how we used PERM to analyse the behavior of the radius of gyration under different temperatures, comparing with the literature.

SIMULATION DESCRIPTION

Our simulation describes a polymer of 250 monomers in three dimensions off-lattice. Each monomer is represented by a bead with a fixed distance from their neighbors. Each bead is represented as a point in space. The beads have interactions with each other following the Lennard-Jones potential previously described. This will

generate a self-avoiding walk (SAW) due to the repulsion at short distances. To simulate the SAW effectively there exist many algorithms.

The first method we use is the Rosenbluth Method. This is based on the probability of finding the polymer in a specific configuration following a canonical distribution based on a given temperature. The second is the Pruned-Enriched Rosenbluth Method (PERM). This being the improvement of the first one in order to avoid polymers with low probability. From the latter one we calculate the gyration radius of the polymer as an average over the ensemble.

Rosenbluth Method

The Rosenbluth Method consist in adding beads one by one avoiding high energy configurations. For this we put the first bead on the origin and the second one on (1,0,0) since the fixed distance between beads is 1. For the next ones we need to discretize the available space. We divide it in 12 possibilities (4 different θ and 3 ϕ in spherical coordinates) and calculate the energy of all of them.

From the energy we get the weights of each configuration:

$$w_j^{(l)} = \exp[-E(\theta_j)/k_B T] \quad (2)$$

We accept the j th configuration with a probability of $w_j^{(l)}/W^{(l)}$ where $W^{(l)}$ is the sum of all the weights at a determine bead l . For this we generate a random number between 0 and 1 and check to which probability it corresponds.

With this method we generate a polymer with probability of:

$$P = \prod_{l=3}^N \frac{w_j^{(l)}}{W^{(l)}} = \exp[-E_{\text{total}}/k_B T] \prod_{l=3}^N \frac{1}{W^{(l)}} \quad (3)$$

As we can see we are off by a factor of $\prod_l W^{(l)}$ from the total Boltzmann weight of the polymer. So to have accurate properties we need to correct them by this weight factor. This is the “method A” described in reference [3].

PERM Method

The problem with the Rosenbluth Method is that polymers with very low probability will still be produced as well with extreme high probability. This problem will be deeper treated later in the comparison. To avoid this wide distribution of weights we have to control the polymer's population.

First we need to make an enrichment. This means that at any l step we compare the weight with a suitable chosen threshold $W_l^>$. If the polymer weight at l bead is bigger than this threshold we make an additional copy of it and we half the weight of both, the original and the copy. Then we let them grow until the final length. With this we are avoiding extreme high probability polymers.

On the other hand if the weight is less than another threshold $W_l^<$ we prune. The pruning is made through a random generated number. If this number is less than 0.5 we leave the polymer grow but with the double of the weight. On the contrary if it is bigger we cut it.

With this enrichment and pruning we are not introducing any additional bias so we can chose any thresholds for our convenience. However bad choices of these thresholds lead to inefficiency on the program. To have a small statistical advantage we run the program without any enrichment or prune for 10 times to after calculate the thresholds:

$$\begin{aligned} W_l^> &= 3 \times \frac{1}{l} \sum_{n=3}^l W^{(n)} \\ W_l^< &= 0.2 \times \frac{1}{l} \sum_{n=3}^l W^{(n)} \end{aligned} \quad (4)$$

Contrary to other evolutionary algorithms we use a depth first approach. This means that we follow a single polymer until it ends before we continue to another polymer instead of having a constant population and simultaneously grow all of them (breadth first approach). We implement this by means of recursion having a minimized storage use and a more efficient and elegant code [4].

COMPARISON OF THE METHODS

To compare successfully both methods we use the end-to-end distance. This is, as the name implies, the distance between the first monomer and the last one. We use this property because the Rosenbluth Method generates polymer populations for all the length up to the maximum, 250 beads in our case.

As discussed before the polymer behaves as a SAW. The end-to-end distance (R) in these walks scales with the length of the polymer (N) as:

$$R \propto N^\nu \quad (5)$$

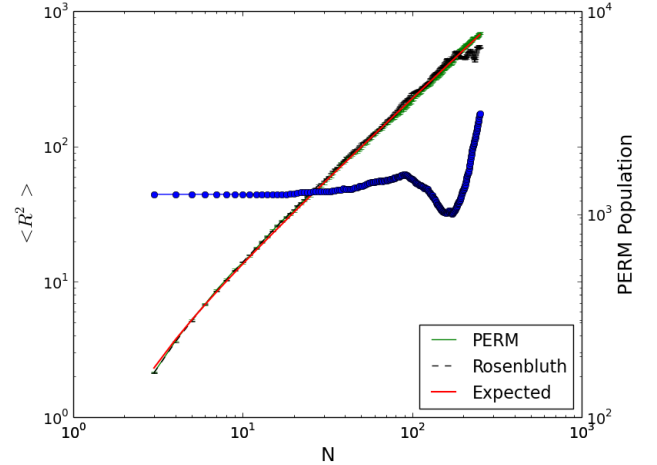


FIG. 1. Comparison of the squared end-to-end distance obtained using Rosenbluth and PERM methods for the three dimensional off-lattice model. The results are compared with the predicted line, in which $\langle R^2 \rangle = a(N-1)^{2\nu}$, with $\nu = 0.5876$ [5]. We used the data to obtain the parameter a . The blue circles show the population of polymers with different N using the PERM algorithm. The data was obtained using $\epsilon/k_B T = 0.25$

where ν is the Flory exponent and has a value for three dimensions of 0.5876 [5].

In Figure 1 we plot the end-to-end distance for the two methods with respect to the number of beads. For the Rosenbluth Method we grow 1000 polymers until the maximum length of 250. For the PERM the number of polymers for each length is also plotted. We compare both methods to the theoretical behavior.

The Rosenbluth Method reproduces the scaling behavior for short polymers, but after the 100th bead approximately the error starts to increase and deviates from the theoretical scaling. The reason behind this error is that the R begins to fluctuate a lot. These fluctuations are due to the acceptance of any polymer with weight higher than zero. So when increasing the number of beads, the distribution of the weight become wider, leaving just a few polymers dominate the statistical (polymers with extremely high probability).

To avoid this dominance, the PERM uses a genetic algorithm to enrich the high probability polymers and prune the low probability, leaving a narrower weight distribution. We can see that this method scales the R according to the theoretical result. The problem with it is the choice of the thresholds. They are very sensitive and when choosing them wrong we get either a lot of pruning (almost no polymer grow to 250 beads) or a lot of enrichment (the computational time increases exponentially). This is the reason why we do not get a highly constant population.

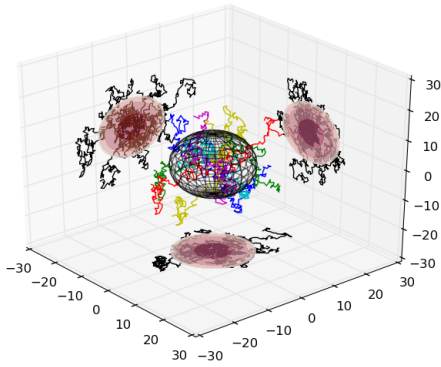


FIG. 2. Six samples of polymers obtained with the PERM algorithm for $N = 250$ and a sphere of radius $\langle R_g \rangle$, where the radius of gyration was obtained through the average of 10000 samples. The data is taken for $\epsilon/k_B T = 0.235$. On each plane we show the projection of the sphere and the shadow of the polymers.

RADIUS OF GYRATION

In this section we discuss some results obtained using PERM algorithm for three dimensional off-lattice chains. The idea is to verify some concepts developed in the literature regarding the radius of gyration.

The radius of gyration measures the average distance that the monomers are from the polymer center of mass (\vec{r}_{CM}). We can write the radius of gyration as:

$$R_g^2 = \frac{1}{N} \sum_{k=1}^N (\vec{r}_k - \vec{r}_{CM})^2 = -\vec{r}_{CM} \cdot \vec{r}_{CM} + \frac{1}{N} \sum_{k=1}^N (\vec{r}_k \cdot \vec{r}_k) \quad (6)$$

where N is the number of monomers in the polymer chain and \vec{r}_k is the k -th monomer position.

We average the calculated radius of gyration over all the polymer samples to obtain its value for each polymer chain size N . In Figure 2 we show six polymer samples with $N = 250$ obtained with the algorithm and we draw a sphere with radius $\langle R_g \rangle$, with the average taken from 10000 samples of $N = 250$ polymer chains. It is possible to verify that, as expected, the polymers are mostly contained inside the sphere.

According to the discussion in [6] the end-to-end distance and the radius of gyration must have the same dependence on N and, therefore, we expect the radius of gyration to grow as N^ν with $\nu = 0.5876$ as we obtained for the end-to-end distance. Indeed, for $\epsilon/k_B T = 0.245$ we obtained a behavior close to $\nu = 0.5876$ (Figure 3).

Finally, we briefly studied the behavior of the above results with the temperature. First, we should note that the radius of gyration is more sensible to changes in

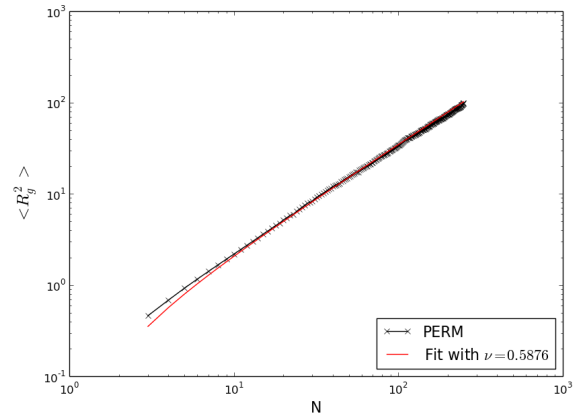


FIG. 3. Squared radius of gyration as a function of the number of monomers N for $\epsilon/k_B T = 0.25$. The results are compared with the predicted line, in which $\langle R_g^2 \rangle = a(N-1)^{2\nu}$ with $\nu = 0.5876$ as in Figure 1, according to [6].

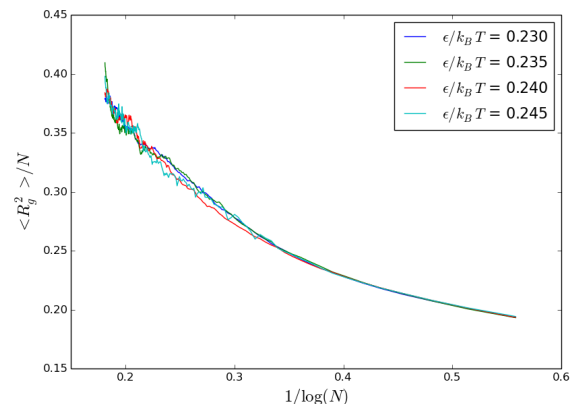


FIG. 4. Swelling factor $\langle R_g^2 \rangle / N$ as a function of $1/\log(N)$ for different temperatures as in [4]

the polymer configuration than the end-to-end distance, which means that if we have some kind of transition, it will manifest first on it. Indeed, as can be seen in [7], at $\epsilon/k_B T \approx 0.25$ we should have a transition from the ideal chain. For this matter, we try to reproduce the results of [4] for the swelling factor $\langle R_g^2 \rangle / N$ (Figure 4) and the ratio $\langle R^2 \rangle / \langle R_g^2 \rangle$ (Figure 5) for different temperatures. The values obtained in these graphs and the behavior for low N agree with those in [4]. However, for large N our result does not show the differences of $\langle R_g^2 \rangle / N$ obtained in that work (in which there is a swelling of the polymer), and for $\langle R^2 \rangle / \langle R_g^2 \rangle$ the behavior is similar but we lack statistics to be conclusive on that. Further work has to be done here to improve the results for large N . These differences can be due to the fact that in the previous cited papers, the authors use variable monomer length, with an unit variance Gaussian

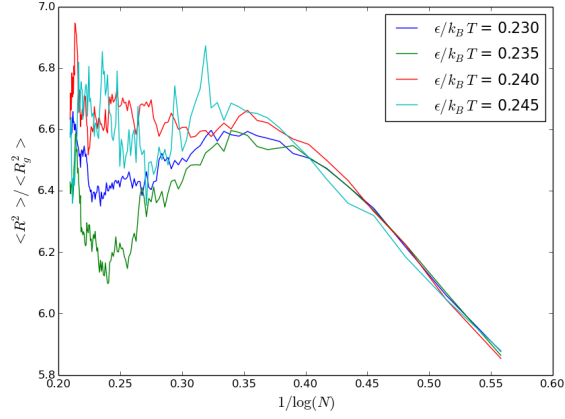


FIG. 5. Ratio $\langle R^2 \rangle / \langle R_g^2 \rangle$ as a function of $1/\log(N)$ for different temperatures as in [4].

behavior whereas in our work we use a fixed monomer length. Here we could also work further to include this Gaussian behavior.

CONCLUSION

Agustin Part!

-
- * All files on https://github.com/bofo90/Monte_Carlo
- [1] J. P. Wittmer and H. Meyer, *Monte Carlo Simulation of Polymers: Coarse-Grained Models*, Vol. 23 (2004).
 - [2] C. W. Yong, J. H. R. Clarke, J. J. Freire, and M. Bishop, *The Journal of Chemical Physics* **105**, 9666 (1996).
 - [3] J. M. Thijssen, *Computational Physics*, 2nd ed. (Cambridge University Press, 2007).
 - [4] P. Grassberger, *Physical Review E* **56**, 3682 (1997).
 - [5] N. Clisby, *Physical Review Letters* **104** (2010), 10.1103/PhysRevLett.104.055702, arXiv:1002.0494.
 - [6] N. C. Smith and R. J. Fleming, *Journal of Physics A: Mathematical and General* **8**, 938 (1975).
 - [7] M. Wittkop, S. Kreitmeier, and D. Gritz, *Physical Review E* **53**, 838 (1996).