Rosenbluth method and PERM simulations of a polymer chain

Stefan Zwinkels 4107896 & Mitchel Kokken 4063384,
Applied Physics, Delft University of Technology.

***Abstract*—In this paper two computational methods are proposed in order to simulate the growth of a polymer chain, the Rosenbluth method and the Prune-Enriched Rosenbluth Method (PERM). The resulting square average end-to-end distances as a function of polymer length were used to compare the methods and to judge their validity by comparing with literature. We found PERM to be significantly more accurate at higher polymer length. A least square error fit was performed on the PERM end-to-end distance data and the resulting parameters were in good agreement with values found in literature.**

***Index Terms*—Computational Physics, Polymer, end-to-end distance, Rosenbluth, PERM, Monte Carlo**

I. Introduction

In this paper two method for simulating the growth of a polymer chain in 2D are proposed and compared. The benefit of this simulation is to gain insight in the growing behaviour of polymer chains and to study whether Monte Carlo based methods for this problem is effective. In this simulation we treat our polymer as a chain of monomers with a fixed distance between two subsequent monomers. The monomers feel short range repulsion and a Van der Waals attraction at longer ranges. We simulate the interaction between the monomers with the Lennard-Jones potential:

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

For convenience σ and ϵ are scaled such that r is in the order of 1. When determining the position of the next monomer, we use a statistical mechanics approach with the Lennard-Jones potential as potential energy to the determine the probability distribution of the possible coordinates of the subsequent monomer.

II. The computation method

In this section two computation methods for the polymer chain simulation are explained, namely the Rosenbluth algorithm and the pruned-enriched Rosenbluth method. The basics of the two methods are the same, however the pruned-enriched Rosenbluth method compensates for the proliferation of very unlikely polymer configurations.

A. The Rosenbluth algorithm

In the Rosenbluth algorithm [1] we start modeling the polymer with two beads on positions $(0,0)$ and $(1,0)$, the distance between each monomer: l is 1. The coordinates of the third bead is chosen from a integer number of similar spaced points on a circle (with radius 1) around its previous monomer. See Figure 1.

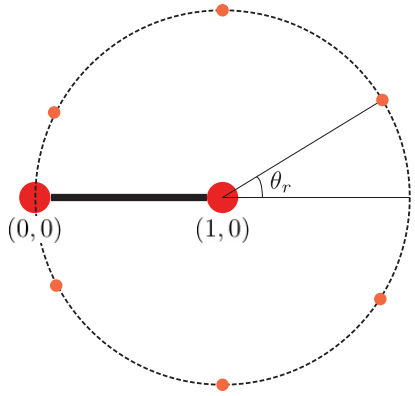

Fig. 1: In this Figure $n = 6$ similar spaced points starting with a random offset of θ_r are shown. The large red dots represent the two initial monomers, the smaller orange dots represent the possible coordinates of the third monomer. The random offset θ_r is chosen in the range: $0 < \theta_r < \frac{2\pi}{n}$.

In Figure 1 $n = 6$ similar spaced points are chosen (the orange dots), all with a random offset of θ_r. The random offset fluctuates random for each monomer between $0 < \theta_r < \frac{2\pi}{n}$. For each possible position a weight can be calculated using the following Equation:

$$w = e^{-\frac{U(r)}{k_b T}}, \tag{2}$$

where $U(r)$ is the Lennard-Jones potential from Equation 1, k_b the Stefan-Boltzmann constant and T the temperature. From the weighted distribution the positions of the next monomer is chosen. This algorithm is repeated recursively until the desired number of monomers is reached.

Due to computational limitations the weight from Equation 2 can return 0 as outcome, which means the simulation will not return the position of the next monomer. In order to compensate for this limitation the calculation for which a total weight $w = 0$ is returned, is executed a second time. In this second try the calculation starts with a new θ_r, which influences the outcome. This procedure is repeated until the criteria for total weight: $w \neq 0$ is met.

After the polymer is fully grown, the squared end-to-end distance R^2 is stored for each length and a new polymer is grown. After a large number of polymers are grown, the

average R^2 is found as well as its error from the standard deviation and the square root law.

B. The pruned-enriched Rosenbluth method

The Rosenbluth method starts to diverge from the expected end-to-end distance for increasing amounts of monomers. This is due to the fact that unlikely configurations are not properly denied: When adding a monomer in which all weights of the possible positions are small, one of these positions is still accepted (with the odds favoring the 'least bad' position). These low weights have no further effect on the polymer itself, and the method may result in growing very unlikely polymers. An attempt to improve on the Rosenbluth method is PERM (pruned-enriched Rosenbluth method) by Grassberger [2]. In this method, unlikely configurations that arise in the Rosenbluth method are removed (pruned), while very likely configurations are copied and allowed to grow further as 2 independent polymers (enriched).

To implement PERM, we track the weight of the entire polymer as it grows as a measure of the likelihood of the configuration of this polymer. This weight is calculated as the product of the sums of all the possible weights for all monomers up until the current length:

$$W_{polymer} = \prod_{i=3}^{L} \left(\sum_{j=1}^{NA} e^{\frac{-U_j^i}{K_b T}} \right) \tag{3}$$

Where L is the current length of the polymer in monomers and NA is the number of possible locations of a monomer. U_j^i is the potential of the monomer i at the possible location j, is given by Equation 1.

After a new monomer is added in the same manner is in the Rosenbluth method, this polymer weight $W_{polymer}$ is updated and the polymer is considered for enrichment. The condition for enrichment is given in Equation 4.

$$W_{polymer} > 2 \cdot \frac{\langle W_{polymer} \rangle}{\langle W_3 \rangle} \tag{4}$$

The current polymer weight is compared to the average of the polymer weights of the current length divided by the average of the polymer weights at the shortest length (in our case, length 3). The factor 2 was suggested by Thijssen [3], and is chosen as such for a chain of length 250 monomers the amount of pruning and enriching is balanced. If this condition is met, a copy of this polymer is made and is independently grown further. Import is that the weight of the original and it's copy is divided by two to maintain conservation of weight.

Similar to enrichment, polymers are also considered for pruning. The condition is similar to the enrichment condition in Equation 4, but has a different factor and the weight of the polymer should be worse than average:

$$W_{polymer} < 1.2 \cdot \frac{\langle W_{polymer} \rangle}{\langle W_3 \rangle} \tag{5}$$

If this condition is met, there is a 50% chance the polymer is not grown any further. If the polymer is grown further its weight is doubled to conserve weight.

In principle, the factors of 1.2 and 2 in Equation 4 and Equation 5 are dependent on the length of the polymer. As suggested by Thijssen [3], this dependence can be removed by multiplying the polymer weight $W_{polymer}$ with a constant each time a new monomer is added. This constant should be $\frac{1}{0.75NA}$ with NA the number of possible locations considered when adding a monomer.

III. Results

In this section the results from the Rosenbluth and PERM simulation will be presented and compared to each other and to similar results in literature. Results for both simulation algorithms are computed for polymer length $N = 250$ monomers for $k_bT = 1$ in reduced units. An example of one full grown polymer using the Rosenbluth method is shown in Figure 2

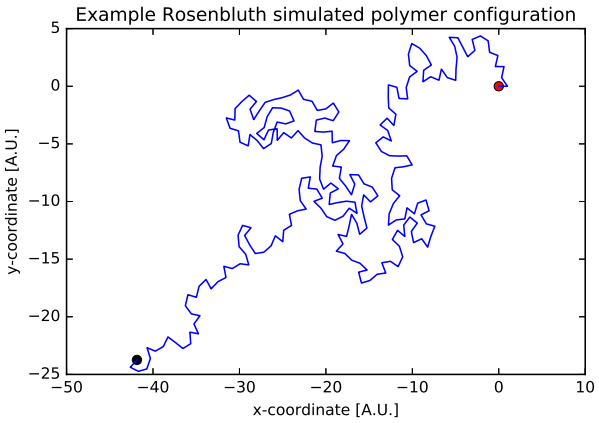

Fig. 2: Example of one full grown polymer using the Rosenbluth method for $N = 250$ monomers at $k_bT = 1$ in reduced units. For this configuration a choice of 8 different angles was offered (see Figure 1). The red dot indicates the start position and the black dot indicates the end position of the polymer.

We have simulated 10.000 polymers using the Rosenbluth method and calculated the average end-to-end distances for polymer lengths up to 250 monomers. Additionally we have simulated a population of PERM grown polymers with a start population of 4000 polymers. A weighted end-to-end distance was calculated. For comparison with the result found by Thijssen, we have fitted a curve through the weighted average PERM end-to-end distance using a least square method. The relation between the squared average end-to-end distance and the polymer length in monomers is shown in Figure 3

The parameters found for the fit to the PERM are: $a = 0.72$ and $v = 1.52$. It is visible that almost the entire fit lies within the margin of the error of the PERM results.

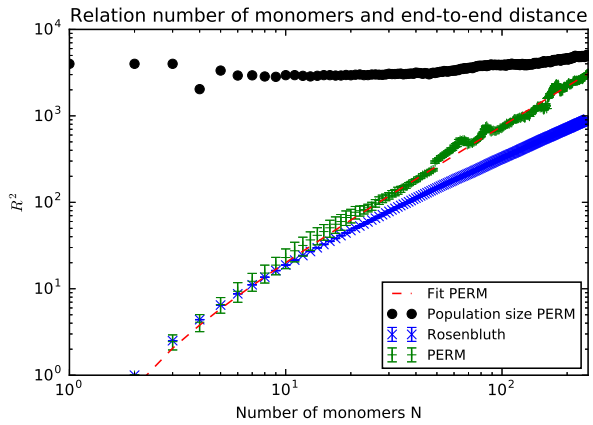

Fig. 3: Squared (Weighted) average end-to-end distance R^2 as a function of polymer length N up to 250. 10.000 polymers were grown using the Rosenbluth method, shown as the blue crosses with errorbars. Starting with a population size of 4000 polymers, the polymers grown using PERM are shown as the green dashes with errorbars. The population size in the PERM is monitored and shown as black dots. The red dashed line is the least square method fit to the PERM results for R^2 as a function of N. The fit has the form: $a(N-1)^v$. The found parameters are: $a = 0.72$ and $v = 1.52$.

IV. Discussion

OUR result of the polymer simulation in Figure 2 shows a self avoiding walk due to the interaction described by the Lennard-Jones potential. Since we are dealing with the Monte Carlo method sporadic self crossing should occur, of which we can see one example.

The curve of the Rosenbluth method shows smooth behaviour, but when compared to the curve found by Thijssen [3] our results for longer polymers appears lower. This difference is outside the margin of error. A possible explanation for the difference might be the computational limitations of the different programming languages (Python vs FORTRAN).

Unlikely polymer configurations are pruned in PERM, resulting in an expected larger average squared end-to-end distance at longer polymer lengths. This behaviour is visible in Figure 3, as the PERM curve rises above the Rosenbluth curve for increasing polymer length N. The fit agrees with the PERM results within the margin of error. The fit parameters also agree with parameters in Thijssen. The population size resembles the behaviour of the population size found by Thijssen. The initial dip is due to initial pruning of polymers with an relative unlikely start, while the immediate rise after the dip is due to enrichment of the relative likely polymers. After this the population size stabilises before increasing towards the end.

A noteworthy occurrence is the oscillations at longer polymer lengths for PERM in Figure 3. We see no reasons for these oscillations to occur. The oscillations might be reduced or removed when the population size is increased.

V. Conclusion

OUR computational implementations of the Rosenbluth method and PERM show expected behaviour and agree with similar implementations done by Thijssen [3]. In PERM the relative unlikely polymer configurations are pruned, while likely configurations are enriched, resulting in an expected larger average squared end-to-end distance at longer polymer lengths when compared to the Rosenbluth method. Our least square error method fit results in: $R^2 = 0.72(N-1)^{1.52}$, which within margin of errors agrees with our PERM results. The exponent of 1.52 agrees with the parameter in Thijssen. When comparing the results of the Rosenbluth method to the results found by Thijssen we see our result for longer polymer lengths appear below that of Thijssen. An explanation for the difference might be the computational limitations of the different programming languages used for the simulation. We can conclude that PERM leads to a significantly better result compared to the Rosenbluth method.

References

[1] M.N. Rosenbluth and A.W. Rosenbluth, *Monte Carlo Calculation of the Average Extension of Molecular Chains* ,The Journal of Chemical Physics 23, 356 1955

[2] P. Grassberger, *Pruned-enriched Rosenbluth method: Simulations of θ polymers of chain length up to 1 000 000*, American Physical Society, volume 56, issue 3, 3682–3693, 1997

[3] J.M. Thijssen, *Computational Physics*, Cambridge, second, 2013, *ch.10*