Monte Carlo polymer simulations with the Pruned-Enriched Rosenbluth Method

Sadiq van Overbeek, Lars Zondag

***Abstract*— In this paper a Monte Carlo simulation of 2D Polymers is presented. The Rosenbluth-Rosenbluth method and the Pruned-Enriched Rosenbluth Method have been used to determine the *end-to-end distance*, *polymer population* and the *gyradius* of a dilute polymer solution. A mesoscopic model of the polymer was considered, characterized by a chain consisting of N beads with a fixed mutual distance with their neighbours and a repulsive/attractive force between the non-bonded beads modelled by a Lennard-Jones potential. Using both methods, 10,000 polymers were grown up to a size of 250 beads with 6 possible angles for each new bead.**

I. INTRODUCTION

Monte Carlo (MC) methods are computational algorithms based on random numbers to solve deterministic problems. In this research we shall use MC to focus on an important topic in statistical mechanics: the behaviour of a dilute solution of polymers, flexible chain-like molecules. Although their chemical properties can be well described by chemical formulas, their physical properties, such as the end to end distance, are strongly related to their geometry. The numerical simulation in this research will focus on 2D polymers in a good solvent such that we can make a mesoscopic model of the polymer analogous to a self-avoiding walk (SAW). This was first introduced by Rosenbluth and Rosenbluth [4] in the 1950's and later on used to develop the Pruned-Enriched Rosenbluth Method (PERM) [2]. We will focus on both of them.

In section II the model & methods of the Rosenbluth algorithm and PERM are described, this can be used to determine the end-to-end distance, population and gyradius of the polymers presented in section III. Section IV will contain the conclusion of this study.

II. MODEL & METHODS

A. Rosenbluth and Metropolis algorithm

In this project we consider a mesotropic model of the polymer. A chain consisting of N beads that represent segments of the polymer that in turn represent groups of atoms. The neighbouring beads have a fixed mutual distance (taken to be 1) and the non-bonded beads interact with a Lennard-Jones potential with strength $\varepsilon = 0.25$ and range $\sigma = 0.8$ (in terms of natural units):

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

The repulsive force between the non-bonded beads makes this problem analogous to a self-avoiding walk (SAW) where a new bead is generated at each step whilst avoiding high-energy conformations. A method implementing such a generation of avoiding improbable configurations is the Rosenbluth-Rosenbluth (RR) method [4]:
(1) For simplicity the first link is placed from (0,0) to (1,0).
(2) The set of N beads reaching from the origin to the end is associated with a weighting function W_N calculated at each step.
(3) At any stage of N beads reaching to the end, four positions $(x \pm 1, y \pm 1)$ must be considered.
The Metropolis algorithm (each configuration is generated with a probability depending on the previous configuration) proposes to add each new bead with a probability distribution of:

$$\exp[-E(\theta)/(k_B T)] \tag{2}$$

where $E(\theta)$ is the interaction energy of the new bead with every previous bead given by the Lennard-Jones potential in equation 1 and discretized for a finite number, J, of equally spaced angles θ_j with a random offset, $d\theta$. An angle θ_j is then accepted with probability $w_j^{(l)}/W^{(l)}$ where the weight $w_j^{(l)}$ is given by the probability distribution introduced in equation 2: $w_j^{(l)} = exp[-E(\theta_j)/(k_B T)]$ and $W^{(l)}$ is the sum over these weights. The acceptation of an angle θ_j is done with the 'roulette-wheel' algorithm [5]:
(1) The interval [0,1] is divided into J segments of size $w_j^{(l)}/W^{(l)}$.
(2) A uniform random number between 0 and 1 is calculated and the corresponding segment and angle is chosen.
The final Boltzmann weight of the polymer is then the product of all N chosen weights:

$$W = \exp[-E_{total}/(k_B T)] = \prod_{l=3}^{N} w_j^{(l)} \tag{3}$$

B. Pruned-Enriched Rosenbluth Method

A more advanced algorithm for simulating flexible chain like polymers is the Pruned-Enriched Rosenbluth Method (PERM) introduced by Grassberger [2]. It combines the Rosenbluth method with recursive pruning (removal of polymers with a low weight) and enrichment (cloning polymers with a large weight). This is done as follows:
(1) If a polymer has a weight above an upper limit, the polymer is cloned and each of them is given half the weight of the original polymer keeping the total weight of the conformation in the population constant.
(2) If a polymer has a weight below a lower limit, the polymer is removed with probability 0.5, however, if the polymer is not removed, its weight is multiplied by 2.
The choice of the upper and lower limit determines whether

the population grows, shrinks or remains constant. A right choice is given by J. Thijssen [5] and depends on the average weight 'AvWeight' at step i and the shortest length of the polymer which is 3 beads, 'Weight3' as follows:

$$\begin{aligned} \text{Lowlim} &= \alpha_- \cdot \text{AvWeight}/\text{Weight3} \\ \text{Uplim} &= \alpha_+ \cdot \text{AvWeight}/\text{Weight3} \end{aligned} \tag{4}$$

where α_- and α_+ are chosen to be 1.2 and 2 respectively in accordance with the algorithm prescribed by J. Thijssen.

C. Overview of PERM code procedure

The scheme below constitutes the PERM algorithm and can be summarized in pseudo-code as follows:

```
Initialize the starting polymer (1st & 2nd bead)
loop for size of desired population:
function ADD_BEAD(polymer, W):
    Calculate the J possible positions for new bead
    Determine the interaction energies of the J positions
    Calculate the weights & probabilities of positions
    Roulette wheel → new bead position
    Store end-to-end value in array
    Add new bead to polymer and determine W
    Calculate Uplim and Lowlim
    if W > Uplim then
        Call add_bead(polymer, W/2)
        Call add_bead(polymer, W/2)
    if W < Uplim then
        if random in [0,1] ≤ 0.5 then
            Call add_bead(polymer, 2·W)
```

III. RESULTS & DISCUSSION

In total the script runs for 12 and 16 minutes, growing 10,000 polymers up to a size of 250 beads using the Rosenbluth algorithm or PERM respectively. Per bead there was a probability distribution over 6 equally spaced angles. The methods described in the previous section were used to determine the average end-to-end distance R, the population size N_{pop} and the gyradius as a function of the number of beads N.

A. End-to-end distance

The end-to-end distance as a function of the number of beads N was modeled for both the Rosenbluth algorithm and PERM, depicted in Fig. 1 and Fig. 2. A fit in the form of $y = \alpha(N-1)^{\beta}$ was made to compare the results with the theoretical end-to-end length scale of $R \propto N^{1.5}$ [5]. The values for α and β for the Rosenbluth method were 3.7 and 1.6 respectively. In Fig. 1 it can be observed that the simulation corresponds quite well with the theoretical scaling for small polymer sizes and shows unstable behaviour for polymers larger than about 100 beads. This can be explained by the fact that the Rosenbluth algorithm does not quench the high-energy configurations enough and thus having a bias towards dense configurations. This effect will be increasingly apparent for a larger polymer length as illustrated by the black circles representing the population size.

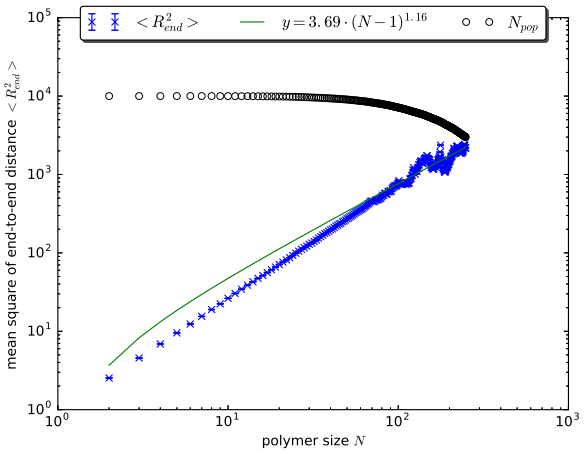

Fig. 1. Mean square of the end-to-end distance $< R^2_{end} >$ for 10,000 polymers with size N grown up to 250 beads and modeled with the Rosenbluth algorithm. The green line is a fit of form $y = \alpha(N-1)^{\beta}$ with the parameters α and β given in the figure. The black circles represent the population size (N_{pop}).

The values for α and β for the PERM were 0.87 and 1.48 respectively. This corresponds well with the theoretical exponential value of 1.5 and this behaviour can also be observed in Fig. 2. Now that more of the dense configurations have been removed and the stronger configurations enriched, the population moves to a more stable distribution.

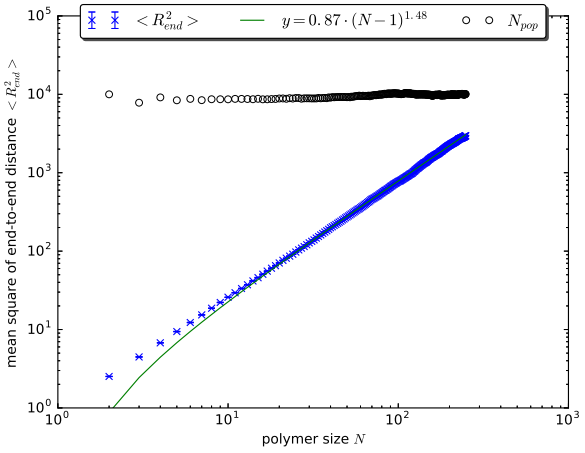

Fig. 2. Mean square of the end-to-end distance $< R^2_{end} >$ for 10,000 polymers with size N grown up to 250 beads and modeled with the PERM. The green line is a fit of form $y = \alpha(N-1)^{\beta}$ with the parameters α and β given in the figure. The black circles represent the population size (N_{pop}).

B. Gyradius

An alternative representation of the end-to-end distance is the radius of gyration, the root mean square distance of the

beads from the centre of gravity of the polymer:

$$R_g = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(\mathbf{r}_k - \mathbf{r}_{mean})^2} \tag{5}$$

This is an interesting property because it can be measured experimentally with light scattering experiments such that the theoretical model can be compared with reality [1]. The gyradius determined with the Rosenbluth algorithm and PERM is presented in Fig. 3 and Fig. 4 respectively and the exponential value of 1.51 found with the PERM corresponds well with the literature value of 1.5 [3]. The unstable behaviour in the Rosenbluth method for larger polymer sizes is also recognized here.

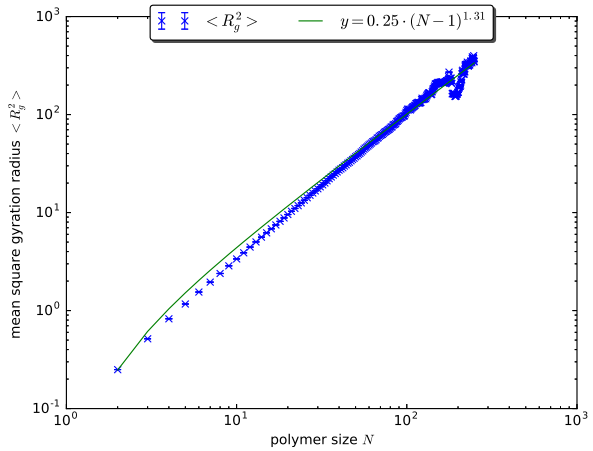

Fig. 3. Mean square gyradius $< R_g^2 >$ for 10,000 polymers with size N grown up to 250 beads and modeled with the Rosenbluth algorithm. The green line is a fit of form $y = \alpha(N-1)^{\beta}$ with the parameters α and β given in the figure.

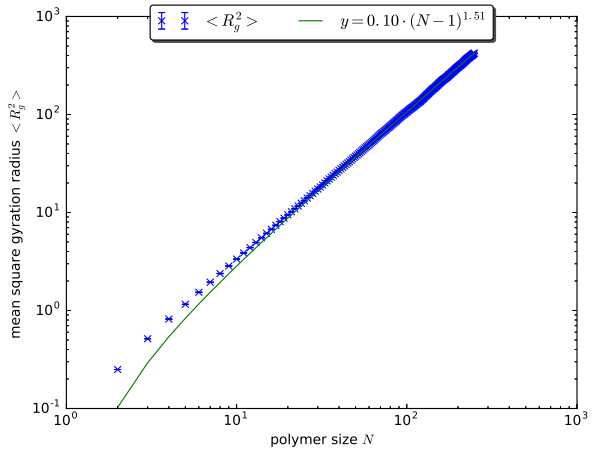

Fig. 4. Mean square gyradius $< R_g^2 >$ for 10,000 polymers with size N grown up to 250 beads and modeled with the PERM. The green line is a fit of form $y = \alpha(N-1)^{\beta}$ with the parameters α and β given in the figure.

IV. CONCLUSION

In the present study, the end-to-end distance, polymer population and the gyradius of 10,000 polymers grown up to a size of 250 beads have been investigated with the Rosenbluth-Rosenbluth (RR) method and the Pruned-Enriched Rosenbluth Method (PERM). The obtained result for the exponential value of the end-to-end length scale of 1.48 with PERM is in close resemblance with the literature value of 1.5. The RR method shows a similar behaviour for small polymer sized ($N < 100$) but does not perform well when the sizes exceed 100. This can be explained by the bias the RR method has towards dense (high-energy) configurations.

REFERENCES

[1] W. J. Briels. *Theory of Polymer Dynamics*. University of Twente, first edition, 1998.

[2] P. Grassberger. Pruned-enriched rosenbluth method: Simulations of θ polymers of chain length up to 1 000 000. *Phys. Rev.*, 56:3682–3693, 1997.

[3] H. Meyer J. Baschnagel, J.P. Wittmer. Monte carlo simulation of polymers: Coarse-grained models. *Computational Soft Matter: From Synthetic Polymers to Proteins*, 23:83–140, 2004.

[4] M.N. Rosenbluth and A.W. Rosenbluth. Monte carlo calculation of the average extension of molecular chains. 23:356, 1955.

[5] J. Thijssen. *Computational Physics*. Cambridge University Press, second edition, 2007.