Polymer Chain Monte Carlo

A. Lovell

March 28, 2015

Abstract Write the abstract here - give a brief overview about what you have done and what you're planning on showing.

1 Introduction

Polymer are chains atoms or molecules of the same type that interact directly between sequential atoms through a spring force and through other pairs via some other long- or short-range potential (given, for example, by hydrogen bonds, van der Waals interactions, or the Lennard-Jones interaction). The typical number of atoms in a polymer is between 10^3 and 10^5.

One would often think that a molecular dynamics calculation would be used to simulate a polymer chain. In this case, the movement of the chain would be calculated at each time step, the forces between each molecule calculated, and then used to update the chain's next position. However, because of the different types of motion involved in the problem - ranging from fast motion of the individual units to slow motion of the chain as a whole - this process can be inefficient, or even impossible, for long chains. [1] Instead, Monte Carlo simulations are used to discover properties of the polymer.

This report is broken up into the following sections. In Section 2, we will discuss the theory behind the problem, including the interaction between atoms and the algorithm that was used to run the Monte Carlo simulation. Section 3 gives a brief discussion of the initial conditions of the simulation and the different grids that were explored. The results of the simulation are discussed in Section 4. Finally, in Section 5, we provide a summary of our work along with progress that could be make to this project in the future.

2 Theory

In the following section, we will describe some of the concepts that will be needed to simulate a polymer chain.

2.1 Interaction

Polymer chains can be modeled in several ways. One of the most common ways is describing the interaction between sequential pairs of atoms as a stiff spring and putting in a Lennard-Jones potential between every other pair of atoms. **Reference this.** However, in this project we will take a somewhat simplified approach. Instead of stiff springs, we will fix the length between each successive pair of atoms, while keeping the interaction between all other pairs a Lennard-Jones interaction. There are, of course, more complicated that could be implemented to fold the polymer into more complex shapes (such as proteins or even origami ducks), but those will not be explored here.

The Lennard-Jones interaction is

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

where ϵ is the depth of the interaction, σ describes the particle size, and r is the distance between the interacting pairs. When the two atoms are far enough apart, they can be viewed as non-interacting. For this reason, we can introduced a cut-off radius, r_c, beyond which the potential between two particles is taken to be zero. Thus, we use the interaction

$$V_{LJ}(r) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] & r \leq r_c \\ 0 & r > r_c \end{cases} \qquad (2)$$

This form takes into account the short-range repulsive force between the two atoms due to Coulomb repulsion and the longer-range attractive forces from dipole-dipole and dipole-induced dipole forces. An example of the Lennard-Jones potential can be found in Figure 1.
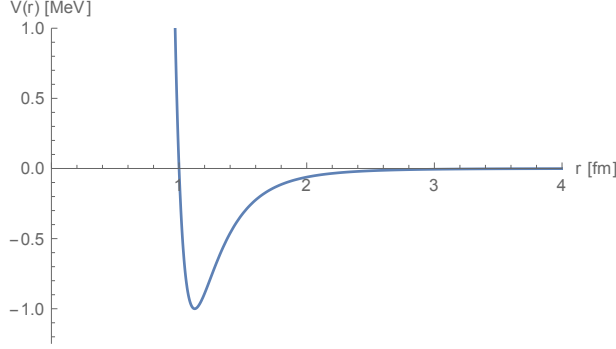
Figure 1: The shape of the Lennard-Jones potential. Here, $\sigma = 1$ fm (the distance where the potential crosses the r-axis) and $\epsilon = 1$ MeV (the maximum depth of the interaction).

2.2 Algorithm

Maybe this will be its own section instead of a subsection.

The algorithm that we use to pick the next point for this Monte Carlo simulation is the Rosenbluth algorithm proposed in 1956 by Marshall and Arianna Rosenbluth. As compared to simple sampling - which generates all configurations with equal probability - the Rosenbluth algorithm generates configurations with different probabilities. These probabilities cause the chain to be self-avoiding and make it so that it terminates when it is trapped in a dead end. [2]

On a square (or cubic) lattice, this means that each nearest neighbor is tested as to whether or not it is already occupied, and the the probability that it moves to that cite is based on the energy of the potential new configuration using the statistical factor

$$P_i = \frac{1}{Z} e^{-V_i/T} \tag{3}$$

where

$$Z = \sum_{i=1}^{N} e^{-V_i/T} \tag{4}$$

Here, V_i is the potential energy between the ith potential configuration and the rest of the chain, N is the number of unoccupied nearest neighbors, and T is the temperature of the system. These probabilities are mapped onto the range $[0, 1]$, and a random number is picked within that same range. If this random number falls within the range P_i, then the next atom is added at the position from which V_i was calculated.

However, we can also grow our polymer off of a grid. In this case we have to sample from a circle in two-dimensions or a sphere in three-dimensions. Do do this we sample from $\cos\theta \in [-1, 1]$ and $\phi \in [0, 2\pi]$ (taking $\theta = \pi/2$ for the two-dimensional case, to fix the third dimension). This samples over the circumference of a unit circle in two-dimensions and the surface of a sphere in three-dimensions. Some number, N, of the points are randomly chosen, and new positions are calculated based on

$$x_i^j = x^{j-1} + \sin\theta_i\cos\phi_i$$
$$y_i^j = y^{j-1} + \sin\theta_i\sin\phi_i \tag{5}$$
$$z_i^j = z^{j-1} + \cos\theta_i$$

The probabilities for each $\mathbf{r}_i$ to be chosen is the same as described above with equations (3) and (4).

2.3 Configuration

The main quantity that we want to calculate is the relationship between the size of the system and the number of atoms in the polymer, which should be related to the dimensionality of the simulation (e.g. two-, three-, or more dimensional space).

3 Initial Conditions

I don't know if this will actually be relevant.

4 Discussion

In this section, we present the main result of our calculation. We were able to vary the temperature of the polymer as well as the number of atoms to calculate the radius of gyration and see how well our simulation compares to (**??**). These results are summarized in Table **??**

5 Conclusion

Talk briefly about what we calculated. **Varying size, varying dimension, varying temperatures - find the critical temperature Theta.**

There is always future work that could be done regarding this project. While we calculated a polymer chain in three-dimensions, it would also be interesting to calculate higher dimensional chains and examine the scaling of the size of the chain with the number of atoms added to the chain. The same results as

(put reference to either equations or figures) should hold, but it would be an interesting project nonetheless to generalize the code. It would also be interesting to examine the difference between this free self-avoiding walk calculated here and a self-avoiding walk confined to a grid. In this way, we could see how the size of the polymer is related to the number of atoms and the dimensionality of the problem.

There are also several thermodynamical quantities that could be calculated from this system, including specific heat, thermal energy, and **many more**.

Although we used a fixed bond length between each successive atom and a Lennard Jones interaction between all pairs of atoms, there are many other interactions that could describe a more realistic polymer. This includes using a stiff spring to model the interaction between successive atoms. We also could have changed the strength of the interaction between various pairs to see what kinds of shapes we could form with our polymer chain. Still, we successfully modeled a polymer chain of **N** atoms in **two-** and three-dimensions.

References

[1] Hendrik Meyer Jörg Baschnagel, Joachim P. Wittmer. Monte carlo simulation of polymers: Coarse-grained models. *John von Neumann Institute for Computing*, 23:83–140, 2004.

[2] Thomas Prellberg. From rosenbluth sampling to perm - rare event sampling with stochastic growth algorithms. *Lecture given at hte International Summer School, Modern Computational Science 2012 - Optimization*, 2012.