Labreport for Simulation of Chain molecule statistics

Introduction: The purpose of this laboratory exercise is to give a simple illustration of Monte-Carlo methods and how these can be used to simulate equilibrium-properties in chain molecules. To achieve this, we will use a software tool called MATLAB. We will use MATLAB to create random (stochastic) numbers and use that to find the end-to-end distance of the ensemble of the chain molecules. We will also compare this to the theoretical value as a way of evaluating the quality of the simulation.

Theory: There are different parts of the theory that need further explanation. First I will begin with the actual Monte-Carlo methods, that have been given their name as a gesture to the capitol-city of Monaco, due to their probabilistic nature. The Monte Carlo methods are a wide class of conventional algorithms that are based on repetitive stochastic sampling to achieve numeric results. They are used in physical and mathematical problems that are impossible to use with other methods. The most famous method is the Monte-Carlo integration.

For regular numerical integration we use methods like Trapeze-methods and Simpsons method. They are deterministic and the user uses Riemann-integration to find values that are approximations. The Monte-Carlo method is non-deterministic; each situation gives a different outcome. We have the one-dimensional function f(x) from a to b with the Monte-Carlo method: $F = \int_a^b f(x)dx$

The Monte-Carlo estimator is then with N uniform variables x_i.

$$< F^N > = (b-a)\frac{1}{(N-1)}\sum_{i=0}^{N} f(x_i)$$

The random variable x_i can be created by taking a random number uniformly distributed. In Monte-Carlo integration the average of the function f(x) over the interval a to b is found and this is multiplied by the length of the interval (b-a), as explained by the formula above.

The other important part of the theory is conformations. We look at that next. Two groups that are bonded to each other with only a single bond can go through a rotation around the bond in relation to each other. The immediate molecular structure that is created as a consequence of that rotation is called the conformation of the molecule. Each possible structure is called a conformer. An analysis of the energy-changes that come about as molecule goes through the rotation is called conformation-analysis. The last important point with regards to conformations is that the structure to a chain molecule is the conformation that is the average through the time. In this laboratory assignment we are using the Monte-Carlo method to find the average of conformations, so that we can find the structure.

Now I will explain two other necessary terms that we need to fully understand the assignment. The first one is Kramers chain. The Kramers chain models a polymer molecule with N small particles bonded together by N-1 rigid rods. The rods are of the length a. The small particles and rods can move freely through each other, and the rods rotates freely around the small particles.

The other term is Riseman-Kirkwood polymer chain. It is a chainlike model where the polymer is presumed to be rigidly fixed in a large number of conformations. But in this model the angles between the rods is assumed to be fixed as opposed to in the Kramer chain.

Method and apparatus: The method is a computer simulation with the programme MATLAB. And the apparatus is a computer. We didn't really need anything besides the computer. The coding was already given and can be found in the attachment to this text.

Results and discussion:

Assignment a)

[a-i] = for xo= 1, a =8, c=2, m=32

Number sequence: 10,18,18,18,18,18

[a-ii] = for xo=1, a=5,c=3, m=32

Number sequence: 8,11,26,5,28,15

In part [a-i] we only have two different numbers created by the random number generator. To understand why this is so, we look at the algorithm manually:

For i=1: (8*1 + 2) mod 32 = 10

For i=2: (8*2 +2) mod 32 gives 32*2 = 64, 82-64=18

For i=3: (8*18+2) mod 32 gives 32*4 = 128, 146-128 = 18

And so on. We can see why the generator only gives two values. For [a-ii] we have a more diverse sequence and more numbers. We can check this one manually as well to understand how it works:

For i=1: (5*1 +3) mod 32 = 8

For i=2: (5*8+3) mod 32 gives 32 *1, 43-32 = 11

For i=3: (5*11+3) mod 32 gives 32*1, 58-32 = 26

We can see that we get the same sequence of numbers.

Assignment b)

We have these values inserted in the code: xo=1, a=75, c=0 and m=65537.

In this assignment, our task is to find the end-to-end ratio for the freely hinged Kramer chain. We were supposed to find this for two different chain lengths, [b-i]: N =5 and [b-ii]: N=10.

Assignment-number	Monte-Carlo value	Theoretical value	Error
[b-i]	3,9885	4	0,2875
[b-ii]	8,9305	9	0,7722
[b-iii]	10,7761		

We can see that we have a small deviation from the theoretical value. In the evaluation of the theoretical value I assumed Q=1. This implies that the program for finding the end-to-end

distance is quite good. In [b-i] we have an error of 0,2875 and in [b-ii] we have an error of 0,7722. We can see that the error becomes bigger as we get a longer chain. This is quite natural since there are no fixed angles between the molecules and we get a bigger variation of possible values.

The last point in this assignment was [b-iii], where we were supposed to use a simple number generator rather than the more complex one from this assignment. By using that generator we got the number 10,7761 for N=10. This is quite far from the theoretical value and gives that the quality of the random number generator gives that the bigger m-value you use, the closer you get to the correct answer. A high-quality number generator is very important for acquiring a correct value.

Assignment c)

For the last assignment, we were supposed to find the end-to-end distance for the Riseman-Kirkwood chain.

Assignment-number	Monte-Carlo value	Theoretical value	Error
[c-i]	55,5894	55,2982	0,5238
[c-ii]	16,7132	17,9803	7,0472

We can see from the table that the error between the Monte-Carlo value and theoretical value is quite small when the angle is 30 degrees, while it is unfortunately large when the angle is 70,5. I think this is largely due to an error in calculation from my side. After continuous attempts to get it smaller, I could not find a smaller value. I also considered some scenarios around the fact that the size of the angle could be a factor in the error, but I don't think that is the case. Its probably more due to a personal error somewhere.

Conclusion: The Monte-Carlo method used in the application MATLAB is a nice tool for finding an approximation for the end-to-end ratios of molecules. If you look at my results, you will find that overall the margin of errors are acceptable, with the slight exception of [c-ii]. This implies that the tool works quite well. But it is also clear that the programme has some limits since it is supposed to be non-deterministic and for a computer that is quite the hard

task since it is by nature deterministic (after all it is programmed to do certain tasks). However the Monte-Carlo method itself is a very good tool for complex questions.