Introduction to Polymer Physics
Prof. Dr. Prateek Kumar Jha
Department for Chemical Engineering
Indian Institute of Technology-Roorkee

Lecture-25
Practical Aspects of Monte Carlo Simulation

In the last class we have been discussing the Monte Carlo simulations of polymer chains and we discussed the basic algorithm and then we discussed some practical tips of how to perform a Monte Carlo simulation mainly in context of the polymer chains. Today I will take it further talk about some more practical tips and then I will briefly introduce some other computational methods apart from Monte Carlo that are used in polymer science.

The basic idea of the Monte Carlo simulation of the following- we start with a simulation box we always have been drawing the cubic box which need not be the case but cubic is the most common case, and in that box you will place your particles it can be your polymer chain like a bead spring model for example, but just for a general sake I can say it is a collection of particles which are at different positions in the box and let us say we refer to those positions as R_i that refers to say R_1 R_2 to R_n where n is a number is a number of particles and then a basic Monte Carlo step correspond to a trial move to a new position and then for the trial move we compute the energy change delta u and then we accept or reject the move with certain condition that we drive the metropolis algorithm. If we accept it then we set our $R_i = R_i'$ and we go back to making a trial again and continue until properties of interest converse.

$$\{\vec{r}_i\}\ \mathit{after\ trial}\ \{\vec{r}_i\,'\}$$

$$\mathit{compute\ energy\ change}\ \Delta U$$

$$\mathit{then\ accept} \vee \mathit{reject}$$

$$\mathit{if\ accept\ then}\ \{\vec{r}_i\} = \{\vec{r}_i\,'\}$$

$$\mathit{if\ reject\ then\ continue}$$

until properties of interest converge

So let us see like what do we mean when we say properties of interest, so in reality what we are doing is when I am doing these simulations to begin with can I start with state that is very far from equilibrium and then the hope is that affect continue doing it I will reach a state of thermodynamic equilibrium. So we will have some equilibrium or equilibration phase after which we will reach equilibrium then we get a particular positions of different particles in the system. Now those positions are not the only possible positions at equilibrium, that is one thing we have to be very clear because what do you mean by equilibrium. Equilibrium does not mean that the system is frozen. The particle still continue to move the only thing that is true is the number of forward transitions are equal to number of backward transitions. A particle is going this way it has probability also to go back in the opposite way such that the rates of both the forward and backward processes are constant.

So after we reach equilibrium then we do something known as a production run where you can average over those many confirmations which are at equilibrium. To begin with let us say I start with a polymer chain. The polymer chain would be very far from equilibrium that means it may have very higher energy the first bring it down to lower energy close to equilibrium and then look around that particular value and see like what is the ensemble average of say the end-to-end distance or the gyration radius that I am interested in.

So in the polymer chain then the property of interest can be for example R_g. So when I measure R_g for a particular Monte Carlo step I am getting the R_g for a particular confirmation. So in the beginning I can be very far off from actual values so in the equilibration phase I may get down to certain value and then production phase I will keep on fluctuating around that value which is some sort of a average value that we can refer to as say R_g^2 because you know for the ideal chain model R_g goes to 0 and I am doing like an ideal chain I should be looking at R_g^2 not R_g, but none the less when we get to equilibrium we always remember that it is not a frozen state, we will have fluctuations in the system but when we take an ensemble average we should get the value that we want at equilibrium and the other thing is that whatever fluctuations we have around the average it should not be biased on one side there must be positive fluctuations and negative fluctuations.

So as a practice we always discard the part of the run that is at equilibrium it is very important the equilibrium part, but that part is not used to get thermodynamic average because those confirmation correspond to confirmations which are far from equilibrium. We only care about the confirmation after we reach equilibrium.

Now if I look at say R_g^2 it may take a longer time to start fluctuating around an average it would feel like the property is converging after the longer time than compared to say some other property like pressure, volume or something else ok. So true equilibrium is only achieved at very long time but since we cannot go to like very long time since simulation the way to track whether we are gotten there or not just depends on how long we run, and earlier it used to be that how long it run dependent on how much the computer can do but now the computers have gotten faster. So now we can afford to do pretty long runs at least for system sizes which are not very large and first do a run long enough and start to see whether the R_g value or whatever I am interested in start to fluctuate around an average and then we start doing a production run and the key is we take many samples during that production run and then I take an average over those samples.

Now you will ask that why I am taking a sample that are somewhat farther from each other, why not take samples adjacent to each other, and the reason is if I take the points very close along the trajectory or if I take points just at the next step or the step after it these points corresponds to states which are highly correlated because with in one step I am only moving one particle in the system every other particle remain there right. So these samples which are like very close are basically highly correlated and for the purpose of getting an unbiased average we want to take average over uncorrelated samples. So this is average over uncorrelated samples. There are ways to measure correlations and we will discuss that later in the course, but the basic idea is that if I take samples at distant intervals then beyond the interval that we have chosen it must be such that the particle has lost the memory of the previous step. If I take a very close Monte Carlo steps the memory will remain there. We want to take samples distant enough that the memory is lost and will come back to this point when we discuss the Brownian motion later in the course.

So this is idea so we look at property of interest mean property I am interested in and I have to do long enough until that converse for equilibrium of course we have to take infinite time but since we cannot do infinite time will live with the fact that we first do a run long enough that we can do on a computer and try to see whether the property of interest is converging around an average value or not.

Now there is some problems with this algorithm related to the computational time, So if you think of this at every step I am computing all the pair interactions between all the pairs of particles and I am summing them over because I am interested in the energy change due to the move and energy at any particular state is a function of all the pair potential energy actually it is the sum of all of them and since a move will result in change in the position of particles will have to recomputed all those pair interactions at lease those data involving the particle that is moving.

So now if I start thinking this term then every step at every step we are doing of an order of N^2 computations of pair interactions simply because there are N^2 pairs. So now if I think in terms of computational cost the computational cost will be of the order of N^2 which means if I go from one particle to 2 particle my code will be 1 quarter slower or if I double the number of particles my code will be slower by one fourth. So the simplest scaling number systems largest size is will not really work from starting thinking of a thousand beat polymer and if I put 1000 and actually going to a cost that is 4 times higher.

So, there are some way to reduce this computational cost and one of the most common ways is using a cut-off scheme which basically works like this. If I go back to the simulation box I had like particles positioned everywhere and each particle is interacting with every other particle. However if you recall the definition of the pair interaction I use Lennard Jones function it was something like this that if I go very high in the distances the energy values are very small. In fact if you go to say R=2.5σ that one of the standard values the u of r is typically already less than the $k_B T$ that the thermal energy and I have earlier told you that any energy is which are less than thermal energy are typically less important and they can be ignored because if the thermal interaction energy is less than $k_B T$ the entropy will dominate in that particular case.

So what this means is I need not look at all the pairs of particles in the system we can only look at the particles which are within a cut-off distance which can be set as for example 2.5σ because beyond this cut-off in any case the interactions are very small right. For every particle I will draw a small volume R_c only those particles which are within that volume will be considered for the interaction energy with that particular particle, those which are like outside this volume will simply be ignored. Ok. So keeping that in mind now you can see that at every step I am not computing N^2 interaction because for every particle I am computing a much lesser number of pair interaction when compared to N^2. So what it means if my computer same code will be much faster depending on the cut-off distance subject to the fact that I will make an small error in the energy because I am cutting it off at some value.

So let us say if I cut it off there at 2.5 or whatever I choose, and let us say after this point my energy values are 0 and before this point it is positive. So I will think of a graph in which beyond R_C I would think it is going to 0 that means U of R is equal to that given by the Lennard Jones for R less than R_C and it is 0 for R greater than equal to RC. This is what is called a truncated interaction.

$$U(r) = U^{LJ}(r)\, where\, r < r_c$$

$$U(r) = 0\, here\, r \geq r_c$$

Now there is bit of a problem here that if I look at R=R_C at that point the energy is discontinuous. So this create some problems because in certain algorithms we are interested in the gradient of the energy. So in molecular dynamics I am interested in the gradient and in those cases even otherwise even in the Monte Carlo algorithm this discontinuity is known to give you some artefacts and it is better to have a continuous energy function. So it is a better practice than to shift the potential such that I will make 0 here and I will make that small correction corresponding to the u value at cut-off, it will barely make any difference here but make like somewhat larger differences towards the cut-off distance but by in large the advantage of this is I will not have any problem because of the discontinuity of the energy. So in that case what we use is known as a shifted and truncated potential which is given like-

$$U(r) = U^{LJ}(r) - U^{LJ}(r_c) here\, r < r_c$$

$$U(r) = 0\ when\, r \geq r_c (now\ it\ is\ continous)$$

So this is like one of the ways I can make the code faster because now I am looking at fewer pairs it will only work for the short range interactions or it will be more efficient for short range interaction and for which at certain cut-off distance the energy become very small. If you think of for example in long range interaction like coulomb that decay is less rapid and then we have to take either a very long cut off or maybe think of something else and there are advanced scheme that work in those cases. Apart from cutting it off like in this particular way there are smarter way of doing it which I will not going to details. But if we want to read up, there are more efficient schemes known as Verlet List, Cell list etc. Ok, that is the main thing that you have to keep in mind if I want to make my Monte Carlo code very efficient.

The next important thing in this kind of simulation is like what boundary conditions do I employ and this turns the boundary conditions that are sometimes very critical to the performance of the code and exactly more importantly what system we are trying to simulate. So one of the boundary conditions can be for example that I will keep my particles in a closed box.

So let us see like what is exactly we are modelling. If I am confining say a polymer chain in a closed box, let us think of a carton for example. So let us say that the sides of the box are closed in a carton and we have a polymer chains that is present in the box. Now as you can see now the polymer chain will keep on moving within the box and many an occasion it will hit the walls of the box and once it hits the wall with a box it may have for example an elastic collision and come back and then maybe it will reorient itself and maybe after sometime it some other boundary. So what this means is like it correspond to a system where a polymer chain is indeed contained in a very small box, it does not capture actual systems thinking to model where I had a beaker which contained a dilute solution of polymer for instance and as I was saying I will take a volume in the container that contains a polymer chain because that imaginary volume that we draw in the beaker will not have like a real walls, it was not like there was some wall there for every polymer chain it was going there hitting it come back. The walls of beaker are anyway not considered when will looking at a small volume around the polymer chain.

So in fact if you look at this particular volume or any volume within the beaker it will not be that only one single chain will confined there it can be that polymer chain moves away and some other chain present in the system comes in that volume. So the polymer chain that I am seeing in imaginary box inside the beaker is not the one which is confined it is a dynamic a polymer chain can come and go at one point and if I put in or impose an artificial box I will be introducing some artefacts because of the fact that now the polymer chain will collide with in the box and come back and they bound and do weird things.

So if I want to represent a bulk system an infinite bulk system a better boundary condition is what is known as a periodic boundary condition and what we do in this case is the following. So now I will think of my simulation box having replicas in three dimensions all over that means the polymer chain is contained in the box, what I will assume now is that this box repeats itself or have replicas in all the directions and just like a polymer chain is contained in the box I am simulating. So in the simulation box I only simulate what is happening in the simulation box, but then I assume that the same polymer chain at the same positions or relative positions are contained in all of those boxes.

So now the advantage of doing this is the following- now if any particle now hit the wall there is no elastic collision now if it is fitting a wall actually it will simply keep on going as if there is no wall present. ok. What we do then is the following, so let us say now the polymer chain during simulation crosses the wall then the bead comes on this end all I do is I will forget about this bead and I will bring in another bead from that side opposite side which was the image of the guy that is lost. So I will call all these polymer chains images of this chain every time a bead or a particle moves out of the box one of its images comes from the opposite end and in this way the wall effect is being modelled and the advantage now is like this entire polymer chain can then move out of the box and the chain containing its image can come into the box.
So there is actually no actual box being simulated, there is no wall effect at all actually the whole idea of the wall is just imaginary, I can think of that my simulation box is not this entire volume I can equivalently think of my simulation box as some other cubic volume. This is equivalently a simulation box because again we will have the polymer chain and its replicas. Everywhere I place this simulation box in the volume as long as the volume is being held fixed I am actually modelling the same system ok and this is the whole essence of periodic boundary condition that and this is the way it is helping you model an infinite system and this is the one that is most commonly employed in all Monte Carlo simulations.

So with this I want to stop here, in the next class I will briefly what is known as molecular dynamics methods and then will continue to talk about the thermodynamics of polymer solutions.

Thank you.