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

Lecture-11
Entropic Elasticity, Bead-Spring Model,
Simulations of Random Walk Models

In the last class we discussed about how to get the probability density of the end to end distance of a chain. First we did for a 1 dimensional random walk and then we generalise the idea for a 3 dimensional general random walk and finally we discussed that we can relate the probability density with the elastic energy of the polymer chain. So, I want to continue on that discussion and talk about some other models of polymer chains and then finally we will look at briefly how we can do the same things that we have derived by equations by some kind simulations.

So, just to recall what we said that the elastic energy of the polymer chain is given as

$$E_{elastic} = \frac{3}{2} k_b T \frac{R_e^2}{M b^2}$$

This has an entropic origin.

Now this particular result has a very special significance, so if you think of like a metal and you will heat it, normally see the metal expands on heating. On the other hand if you take a piece of rubber and you heat it what we do see is rubber shrinks on heating. This can be explained by the very idea that the elasticity in the case of a rubber is a cross linked polymer which has an entropic origin. So, when we heat it as we know as raise temperature the entropy of the system increases and it wants to go to a higher entropy state. And the higher entropy corresponds to the

case when we have more confirmations, so raising temperature we want to go to higher entropy and that means it wants to go to more number of confirmations.

The more number of confirmations will correspond to the case when the value of Re is small. So you think of an extreme case think of like a chain is completely stretched then in that case there is only one confirmation of a chain when the end to end distance is same as the contour length. But if I think of say Re=Lc/2 then that can be achieved in many confirmations. So, if I start thinking of lower values of Re we can think of more chain confirmations that means we are going to higher entropy this we have also from the probability density curve that we derived actually for Re=0 you have the highest number of confirmation. Because these are the ones which have most probable the probability actually decreases as I go to higher values of Re either way in the positive side or negative side. Thus this is one of the most important result that comes from the entropic origin of elasticity.

The other thing that we said is I can associate the polymer chain model with some kind of spring model with a spring constant that is given by:

$$k = \frac{3 k_B T}{M b^2}$$

Where K_b = Boltzmann Constant & T= Temperature in Kelvin.

So, there are couple of problems with this model but this works beautifully in many cases but there are some problems. One of the main problems is in this particular case we do not really limit the maximum stretching of the chain, the chain can really stretched indefinitely okay and if you think about it there has to be a maximum limit which is given by Re=Lc when the end to end distance is equal to contour length that is not presented in the model we discussed.

In our case what we do have is if I look at the elastic energy it does increase R_e as R_e increase but nowhere do we say that there has to be a physical limit in how for the R_e can go. So, it turns out that this particular approximations it is also known as the Gaussian chain approximation of the

ideal chain approximation only works for a small deformations of chain, if the R_e is really very high then in that case we have to have improved models that somehow capture the fact that there is a maximum value of R_e that is given by the contour length. So, those models for smaller values of R_e will be have something like a Gaussian chain but then it must become very steep other near the point when R_e approaches L_c on the other side. It must become very steep at this point. That means for that R_e higher than L_c E elastic should really go to infinity in improved models and in this case E elastic does not tend to infinity. So, one of models is known as the inverse Langevin model and you can read about it we will do not discuss in the class.

There is one thing we have to keep in mind that the approximation we have derived works only for a small extensions. If the chain extension is say more than say 50% of the contour length in that case you should be thinking of some other improved model this is by the way I would say an standard model in many of the cases of a in physics as well this is also called harmonic approximations that only works for a small deformations around the mean value.

the other point here is we can also associate a force that we require to hold back a chain at a particular value of R_e just like if I want to stretch spring I have to apply certain force to get it to certain length. In the same way if I want to maintain a certain value of R_e I do have to apply certain force to it. Okay, and the force is given by the derivative of the energy. Because we know if I have to look at the energy, energy is given by the force multiplied by the displacement the integral of that. In the same way the force to help the polymer at a particular value of Re will be given by the derivative of the elastic energy that we have derived. And in general we are looking at the vector R_e. so:

$$\vec{F}_{elastic} = \frac{\partial E_{elastic}}{\partial \vec{R}_e}$$

¿ $$\frac{\partial}{\partial \vec{R}_e}\left(\frac{3}{2}k_B T \frac{\vec{R}_e^2}{M b^2}\right)$$

$$¿\frac{3}{2}\frac{k_B T}{M b^2}\frac{\partial}{\partial \vec{R}_e}\left(\vec{R}_e \cdot \vec{R}_e\right)$$

$$¿\frac{3}{2}\frac{k_B T}{M b^2}\cdot 2\vec{R}_e = \frac{3k_B T}{M b^2}\vec{R}_e$$

And that force of course in the direction that force has a same direction as the end to end distance vector R_e in the same direction we have our force acting to hold the chain in it is position. So, now this is like where we have come is starting from the random walk models. And what we have seen here is we are replacing the polymer chain with 1 spring of a certain stiffness constant.

So, now we can extend this idea for bit further by noting that if I cut this polymer chain into pieces we still have polymer chains. So, we can think of those cuts also as effectively a Gaussian chain.

So, I mean to say is if I have a polymer chain like this and if I say cut the chain then these individual cuts also represent a polymer chain right because since the M is very large any particular if I do that M/2 what M/3 that is also very large. So, we can start thinking of not really a single spring but many of these springs that my polymer chain is composed off.

And the advantage of doing this is unlike the first case where I represent the polymer chain by simply 1 spring we now have some sort of internal flexibility inside the polymer chain. Now,I have is actually a collection of springs each of them will have in a spring constant k but each of them will have different extensions the model we thus form is known as the bead-spring model. It became especially useful. If I am trying to simulate the behaviour of a polymer chain that is one use of it, the other use of it is when I try to look at the dynamics of a polymer chain. If I used the earlier model there is no internal flexibility being considered it does capture the mean square end to end distance correctly. But it does not tell me like what is the internal structure of the polymer chain and if I go for this detailed model I can also account for the internal flexibility. But the idea simply extends from the model we discussed, so we have not completely derived this model, we simply state the relationships for this particular model. So, I can say that these beads have positions R_0, R_1, R_2 and so on.

So for any spring in the system, let us say look at the nth spring that is between Rn-1 and Rn this spring will correspond to certain energy U_n as like this. So, now we can write as:

$$U_n = \frac{3}{2}\frac{k_B T}{b^2}\left(\vec{r}_n - \vec{r}_{n-1}\right)^2$$

So, if I look at the probability distribution of this particular bead spring model. So, now we do not look at just end to end distance the entire chain I can look at the end to end distance of the individual springs or in general I can look at the positions of all the beads which I can write in shortcut notation as:

$$P\left(\left\{\vec{r}_n\right\}\right)$$

It means is basically all the positions R_0, R_1 to R_m and this will then be a product of the individual probability densities for each of the spring. So, for each of this spring now is Gaussian spring, for each spring we can associate a probability density and so at any given movement the probability density for the entire chain will be a product of the probability densities of the individual segments.

$$P\left(\left\{\vec{r}_n\right\}\right) \propto \prod_{n=1}^{M} \exp\left(\frac{-3}{2b^2}\left(\vec{r}_n - \vec{r}_{n-1}\right)\right)$$

This model becomes particularly useful in simulations as we will discuss later. So, now I want to briefly touch upon how we can we simulate these polymer models not specifically the bead spring model that we come to later but the other models that we have discussed earlier and see like there is an alternate way to get the results that we have derived and the alternate way also promises to be more versatile in the sense that it can make those models more complicated then what we can analytically derive.

So, let's say we consider the simulation of freely jointed chain, so now earlier we did the analytical derivations to get the mean square end to end distance R_e^2. So, now the idea would be that the ensemble average that I am getting here we will now get by simulations. So, essentially what we have to do we have to simulate many chain confirmations compute their R_e^2 and take the average of all of that, provided that we are generating a very large number of confirmations. We can hope that the simulations will give me the same results of course for a smaller number of confirmations this may not be accurate because the idea of the thermodynamics is that the ensemble size must be very large. So, but we can do that in typical simulations, so the model we are after is this freely jointed chain and we can generate this in the following way, so we can start from say origin.

So, let us say in the first step we start from origin 0, 0, 0 which becomes my first point that is right here or you can call it R_0 and the subsequent point I will generate at a distance b from this R_0 generate next point. And there is particular way of doing it, so let us say if I have started from this point this is my R_i then R_{i+1} has to be at a distance b but then it can be in any direction in the 3 dimensions. So, the way to think about it is we will draw sphere of radius b around R_i and then I will choose any point on sphere surface as R_{i+1} where the sphere has a radius b. So, you can use the idea of spherical coordinates and by that what it amounts to is we can choose randomly the spherical angles θ and ϕ where θ has to be between 0 to 2 πand ϕ has to be between 0 and π. And the position we can write as:

$$x_{i+1} = x_i + b\cos\theta\cos\phi$$

$$y_{i+1} = y_i + b\cos\theta\sin\phi$$

$$z_{i+1} = z_i + b\sin\theta$$

Here we are generating points randomly on the sphere surface that it at radius b with respect to Ri, I can get the new coordinates and then I can repeat until required number of points which for M segment case would be M+1 are generated.

Then we have to repeat for large number of confirmations and for example evaluate R_e^2 and then take ensemble average and then we have the result. So, this is the way of doing it there is only a small problem here that if I use this particular method of generating my random point on the sphere surface, it turns out to be inefficient, in the sense that it will not generate points uniformly over the entire surface you will tend to have more points near the poles and lesser points near the equator.

An alternative way is to do is to generate 2 random variables u between 0 and 1 and v between 0 and 1 and then we get:

$$u \in [0,1], v \in [0,1]$$

$$\theta = 2\pi u, \phi = a\cos(2v-1)$$

As an assignment you can try doing both these methods and see why I am saying that you will not get a uniform distribution in this case and why this will ensure uniform distribution.

So, once we have got this idea for a freely jointed chain now you can of course get the ensemble values of R_e^2. We can look at the probability density distribution of confirmations, we simply look at what is the probability of having confirmations within a range of R_e values and so on and we can reconstruct the probability density curve, the mean square ensemble average and all the quantities we have simply derived and actually we can also see how they structure of polymer chain is changing for different confirmations.

Now we can apply the same idea for other models particularly the one dimensional random walk is extremely simple. So, I want to do a walk of M steps like that where I can move left or right with equal probability. The way to do that again-

- Start from origin.
- Generate next confirmation to left or right with equal probability that is ½ and the way of doing it is you generate a random number say between 0 and 1.
- let us say I call this number r, then r>0.5 we move to the right that

$$x_{i+1} = x_i + b$$

- if r < 0.5 then we move to the left

$$x_{i+1} = x_i - b$$

- Repeat for M steps and
- Repeat for large number of confirmations.

If I am doing for example a random walk in 2 dimension or higher dimensions all that changes is my algorithm here. So, if I am doing for example a 2D walk. So, now we have 4 outcomes left, right, up and down, so now we can again generate R between 0 and 1 provided that it is uniformly distributed from 0 to 1 and now we can say:

- $r \in [0,1]$
- If r<0.25 then we move left
  $x_{i+1} = x_i - b$
  $y_{i+1} = y_i$
- If r is between 0.25 to 0.5 we do move right, then
  $x_{i+1} = x_i + b$
  $y_{i+1} = yi$
- If r is between 0.5 and 0.75 then it will move up. So we have:
  $x_{i+1} = x_i$
  $y_{i+1} = y_i + b$
- If r is between 0.75 and 1 then it will move down. So,
  $x_{i+1} = x_i$
  $y_{i+1} = y_i - b$

There are small detail here that of course r can take a value of 0.5 in this case that we not really make much difference here but it has to be r greater than equal to here. Similarly we can put greater than equal to sin at this intervals depending on also your generator for the unknown number. But that will not really make much difference because r is a floating variable that is changing in a continuous way from 0 to 1.

So, same idea we can also apply if I for the 2D walk if I am not allowing for the folding back. The only difference would be is, so I do not want to allow for the possibility of folding back. So, this is my positions R_i, R_{i+1}, R_{i+2}, so I have to make sure that R_{i+2} is not equal to R_i. So, we add constraint in the algorithm that we had earlier you can generate in the same way but now with a constraint that we will not allow $X_{i+2}=X_i$ and, so this is like the and operator. So, only when both of these are true we should disallowed both are not be true. Of course Xi+2 can be equal to Xi but yi+2 can be different and that will not amount to a folding back.

So, this is a way we can simulate the variety of models that we have discussed keep in mind that in all these cases we are simply enumerating confirmations there is no energy coming into this picture. We simply enumerate many possible confirmations and we take an ensemble average, we do not say that these confirmations have higher energy and these have a lower energy and so it turns out that we have to generate many confirmation irrespective of the actual scenario where some confirmations can be much less probable and they can be avoided.

So, more on that later but right now what we have done is actually some sort of a Monte Carlo we can think about it. But basically we are simply enumerating all possible confirmations and then try to see like what confirmations are more probable my simply this enumeration process. So, you can write this kind of codes in any language of your choice and try to visualise how the polymer chains look like that should give you the feeling of what we have covered in the class so far. That why are we saying that there are more confirmations for lower values of Re this we should see by doing this kind of simulations. We can also establish by scaling large R_e^2 proportional to M and you should see that this could hold for large values of M this also we can establish from these kind of small simulations and finally once we are in the habit of doing this the same scheme will go to into a more detailed Monte Carlo code except that we will also have an energy calculation in between that.

So I want to stop that this, thank you.