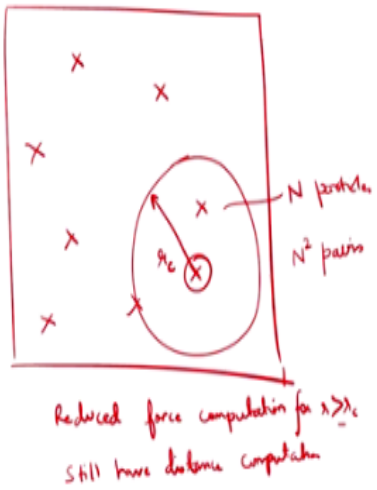
Advanced Thermodynamics and Molecular Simulations
Prof. Prateek Kumar Jha
Department of Chemical Engineering
Indian Institute of Technology, Roorkee

Lecture – 43
Saving CPU Time: Short Range and Range Interactions

Hello all of you, from the last class we have discussed the pair potentials and we touched upon the idea of cutoff and shifting and truncation of the Lennard Jones potential. So, in this lecture, I will first discuss some more efficient ways of handling with short range interactions, that will make computations faster. So, we already have discussed one of that in the previous lecture that the use of cutoff.

(Refer Slide Time: 00:51)



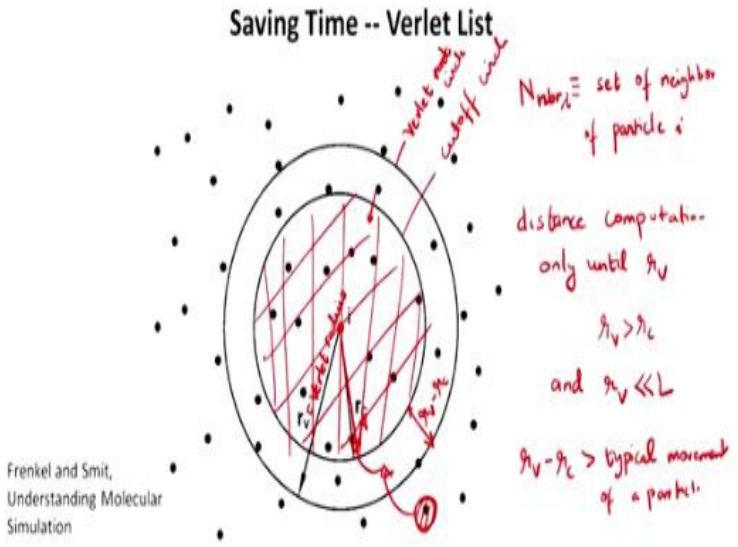
So, the idea is that if you have a system of some N particles, I am just drawing in 2D for convenience. Then instead of like looking at all the possible pairs, so you will have N square pairs. What will we do is we look at a cutoff distance from every particle and look at the region around that cutoff distance. And I only count the pairs that are falling within that cutoff stage.

So, in this particular case, let say only three particles are falling in this regime of centered around one of the particles. So, for that particle the particle here, I will compute only two pair interactions as opposed to total N -1 interactions that we would have computed.

Nonetheless if you would have noticed we still have to find distances between all the pairs of particles. So, although we have reduced the force computation for r greater than or equal to r_c, we still have the distance computation. Because using the distance only we can figure out whether we have to compute the force or not only after we compute distance and if the distance comes out to be more than cutoff then we say that the pair potential is 0, we will not compute it. But still we are doing a distance computation and distance computation is also expensive.

So, although we have reduced the number of force computation we still have N square distance computation in every step of the particle simulation. So, to go around it this particular problem or to make the code more efficient there are several schemes that are implemented in the Monte Carlo or molecular dynamics.

(Refer Slide Time: 03:17)



One of the most common scheme is the use of the Verlet list and the way it works is if I want to look at a particle P and the interactions due to other particles of this particular particle. So, of course I will look only within the cutoff distance. Now clearly the particles that will fall in this cutoff distance are going to be changing with time. So, particles are moving in the system and therefore at any at any given time the neighbors of the particle i are not fixed.

So, let us say for example there is something like $N_{neighbor}$, that is a set of neighbor, i the set of neighbors of particle i. So, this particular set is actually dynamic and it is changing with time. However what we can do and this is like a trick that we can apply is I can think of a slightly larger radius that I say r_v here, that is my verlet radius. And instead of looking at all the possible pairs containing particle i, I will look at pairs that are falling within this radius r_v. and for those

pairs I compute the distance and if the distance happens to be less than cutoff then we do a force computation. If the distance happen to be more than the r_c then we do not do force computation.

So, the advantage now clearly is that I am doing distance computation only until some radius or distance, r_v where clearly the r_v is higher than r_c. Now you may say that what is the point of doing it, just because the set of neighbors of particle i; The number of particles a set of particles lying in the range r_c, if that is dynamic then the neighbors lying in the range r_v are also dynamic. So, even if we took a larger radius then also the set of particles lying in there are going to be dynamic. So, what benefit do we have? We still have to figure out which particles lie in that larger radius r v. So, we still have to do distance computation.

Now here is the trick that works out here. If my r_v is significantly larger than r_c and my r_v is still less than the box size. I would say significantly less than the box size then in that case what is going to happen is that this verlet list, the list of particles lying within that r_v are going to be such that the smaller volume r_c contains by enlarge particles in that extended list r_v. Of course, this is going to change with time but for a significant period of time the particles are lying within that r v will be only the ones that can come to the smaller circle.

Now the reason why I say it will be true is because if I think of say some other particle that is lying outside that verlet radius or the larger circle in the drawing. For this to come to the region of the r_c, they first have to interrupt the bigger circle and from that bigger circle they have to come to the inner circle.

If this distance $r_v - r_c$ is significantly larger than the typical step size or movement that particle makes that is my $r_v - r_c$ is significantly larger than the typical movement of a particle then it will take a considerable time for those particles outside the verlet circle, let me call this a name verlet circle, let me call it a cutoff circle.

So, particles lying in the verlet circle can easily enter the cutoff circle so there is a movement like this they can come inside and they can go out but particles outside the verlet circle will not immediately jump inside the cutoff circle because they first have to enter the verlet circle and since there is considerable gap, $r_v - r_c$, it will take significant amount of time for the particles lying outside the verlet circle to enter the cutoff circle.

So, this means that I can create a list of particles in that larger verlet circle and update that list may be after certain number of steps. So, let us say, for example I say that I have to perform 10000 Monte Carlo steps then let us say every 100 steps we update verlet list of particles.

In verlet list, this case would mean a list of neighbors of all the particles within their own verlet circles. Every particle will have verlet circular rounded or particles lying within the distance r_v from every particle we create a list and we store in some sort of an array and now whenever I have to compute the forces we only look within the verlet list of every particle and within that if the distance is less than cutoff then we compute the force otherwise we do not.

So, essentially what we have been managed to achieve is instead of doing N square distance calculation in every step, we are doing distance computations only within the verlet circle of every particle, that is significantly smaller in comparison to the box size only when this is true this really makes sense. If my r_v is comparable to box size, it really makes no sense. Because I could have done computed over the entire box. What is the point of creating this verlet list and storing it in the memory of a computer?

But if that r_v is significantly less than the box size and r_v is larger than the r_c and if the gap r_v - r_c is significantly larger than the particle movement in every Monte Carlo step or in every MD step. So, we can make it applicable both for Monte Carlo or molecular dynamics and in that particular case, we can update the verlet list every so often. And within each of the Monte Carlo step I will work with the verlet list that has been prepared last.

So, from 0 to 100 step, I will work with the initial verlet list then I create a new list from 100 to 200 again. I work with the same verlet list I have created at 100 then I recreate the verlet list. So, I have to keep on updating the verlet list, but this will not happen in every Monte Carlo or molecular dynamics step. And this is where the advantage kicks in. So, although this is going to add some expense of creating those lists or arrays of neighbors of every particles and updating them so every so often in the simulation. This is going to reduce the number of distance computation in every step of the Monte Carlo.

So, I would say the larger good that we get from here is that the total number of computations that we have to perform becomes significantly smaller provided that the conditions I have

mentioned regarding that r_v and r_c are true. And indeed this happens in most of the scenarios that we are dealing with. Since we are dealing with very large number of particles and the interaction ranges are typically very smaller than comparison to the box size. The cutoff distances are still smaller and therefore we can make a reasonable large r_v and even then we can save some CPU time by creating those verlet list updating them yet within every particle simulation step or Monte Carlo or MD step within every of those steps we are simply searching over that verlet list of every particle instead of looking in that entire thing**.** This is one of that I would say older methods.

There is something called a neighbor list nowadays that has been there in many software's which also works in a similar idea. There are ways in which we create a list of neighbors of every particle in the system and we update the neighbors every so often.
Now it is quite possible that for whatever reason there was one particle which was outside that verlet circle somehow many to enter the cutoff circuit. So, there is some chance of error and the idea is not to be very rigorous in doing that because clearly we will have some error because of the presence of verlet list as long as that error is not so significant we are good to go.

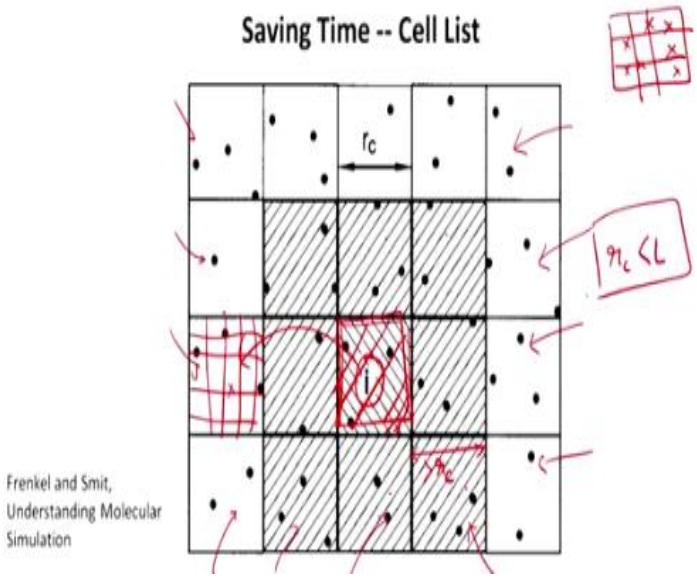
It is always a good practice to see how that cutoff scheme is working, how the verlet list is working to make sure that if that is a promising strategy or not. So, in more realistic cases you will do some preliminary runs with different verlet radius. If you are doing verlet list or different neighbor list conditions and make your that the results are not affected so much by the, implementation of the neighbor list or verlet list. At the same time this give rise to a significant gain in the computation time. This would have lesser computation time by doing that.

If for example, the computation time is larger this means that the creation and updating of this verlet list or any neighbor list is actually adding to the computation time. This is actually taking more time than the time gain we are getting by the implementation of that. So, therefore it is very much necessary to use these methods only when you see a computational gain from doing that. And even the parameters in these methods like the verlet radius in this case should be really optimized in a manner that it is accurate and fast as much as possible.

So, the next method that is even better than verlet list in many cases is to create something known as a cell list and here in the idea is the following so we have a simulation box where

the molecules are located. I divide the simulation box into cells like the what I am showing in here and then I count the list the particles in each of these cells. And it turns out that this particular calculation in which I identify which cell the particle will lie in happens to be pretty quick. I will not go into details here, but it happens to be pretty quick.

(Refer Slide Time: 15:43)



So, let say, for example if I had a simulation box and I choose to divide into nine cells then every particle in the box will be allocated to one of these cells and this will have to be updated after every particle simulation step. You can do it again less frequently as in the verlet list but that will not be very accurate. So, in most cases you want to update the cell list after every step or at least much frequently then compare to the verlet list.

Now what we do then is let us say, for example, if I am look interested in a particle i that is located in the cell right here and the cell length that I have chosen is either equal to r_c or larger than r_c. So, then we can only have to look at the cells adjacent to the particle i, cell of particle i. So, if I know i is in this cell I will look at all the shaded cells in the diagram which are adjacent to it because any cell that is far from here let us say for example this particular cell. Now this cell is clearly further than r_c, because the cell size itself is larger than the r_c.

So, in this case now we only have to look at so many cells within a very large simulation box. Again, this will work only when the cell size or the cutoff distance is significantly smaller than the box size. And again there can be instances where the creation of the cell list is actually giving rise to an increase in computer time, because we have to store it. But in many instances

and actually in most cases it turns out that cell list, reduces the computation time provided that the number of particles are significantly large.

If for example, you have only one particle in every cell in that case, it is not probably very efficient. On the other hand if you have many particles in every simulation cell then by just looking at fewer cells in the simulation box we are reducing the number of distance computations significantly and this will make our code more efficient.

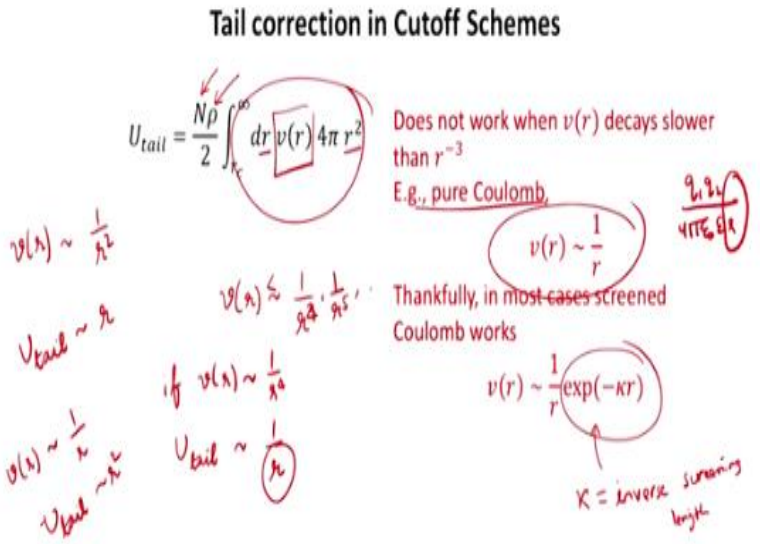
So, whenever we start coding any Monte Carlo or molecular dynamics scheme and once we have a basic cut of the code where once we have the basic algorithm in place. The first thing we should do before actually going to the results part of the thing is try to see how we can make our codes faster. And if we have short range interactions in the system, it is really wise to think of implementing some strategy like verlet list, neighbor list, cell list or something of that sort and this will give you a marginal games or dramatic gains in the computation time that it takes for the simulation.

So, now I want to discuss the idea of like, the how much error do we make when we choose any cutoff is key. So, whenever we are using cutoff clearly we are ignoring interactions beyond r greater than r_c. I was telling you that this is not so important as long as that energy at the cutoff.

Or the pair potential at cutoff is significantly less than the thermal energy. Nonetheless we want to evaluate how much error we make and that can be done by using something known as a tail correction.

$$U_{tail} = \frac{N\rho}{2} \int_{r_c}^{\infty} dr\, v(r)\, 4\pi r^2$$

(Refer Slide Time: 20:10)



So, what we essentially do is we integrate the mathematical form of v_r from the cutoff distance to the infinity because we are ignoring the v of r from cutoff distance to infinity. And this is before shifting and truncation, shifting and truncation were simply mathematical tricks. We were working with some original potential v of r that was without shifting and truncation and this was the error we were making in there. Shifting and truncation did not quite solve the problem it simply made the function look continuous. The error is still remains because we are ignoring the interactions between particles which are more than r_c distance.

In reality every particle in the system can interact with each other and that effect has to be included somehow. So, we have to measure how much is the error we are making by cutting off the potential or interaction at particular distance. So, this is the error we make, so n here is the number of particle, ρ is the system density. And it turns out that as long as the v of r is going like 1 over r^3 or less than that then in that case, we are probably okay, why is that? Because there is an r here and there is an r^2 here then total there is an r^3 here.

So, if v of r is like decaying like 1 over r^3 then you will have something like constant term here that will not increase with r even a constant even then it will diverse. If it is less than 1 over r^3 let us say for example, if it is 1 over r^4 or 1 over r^5 or whatever then in that case what we will have is my tail correction will go like something like 1 over r, if my v of r is like 1 over r^4 because you have r^2 from the integral so I am just doing a scaling argument here. So, you have r^2 , r^3 from the integral and my interaction is going like 1 over r^4 So, the U of tail should go like 1 over r and since this is pretty much going to 0 as r increases very smoothly, therefore we are good to go.

On the other hand if my v of r was going like say 1 over r^2 then in that case my U tail actually goes like r. And this means that this particular thing will not converse, it will diverse as r increases. So, U tail becomes larger and larger as r increases and in that situation it is problematic to use a cutoff interactions.

So, then when can this be one can be true? So this will clearly be true when you have for example a pure Coulomb interaction because then v of r will go like one over r, so Coulombic interaction is given by something like-

$$columb\ interaction = \frac{q_1 q_2}{4\pi\epsilon_0\epsilon_r}$$

So, basically should look at how it goes with distance it goes like 1 over r. So, U tail will go like, 1 over r multiplied with r^3 and that is like r^2 that is clearly a diverging integral. It becomes larger as r increases and therefore cutoff is not the correct way to look at it. Fortunately it does not have to be the case because in most cases the Coulomb interaction is not present in the pure form that is to say that you always have particles between the two particles and those particles do something known as a electrostatic screening in there.

So, if the particles were in vacuum separated then only you will have this pure Coulomb interactions. If there are particles in between these two particles then the in between particles will screen the interaction between these two particles and this is what is known as electrostatic screening and this given rise to a more rapid decay of the electrostatic interactions they go like one over r multiplied with exponential of minus kappa r, where kappa is known as the inverse screening length.

$$v(r) \sim \frac{1}{r}\exp(-\kappa r)$$

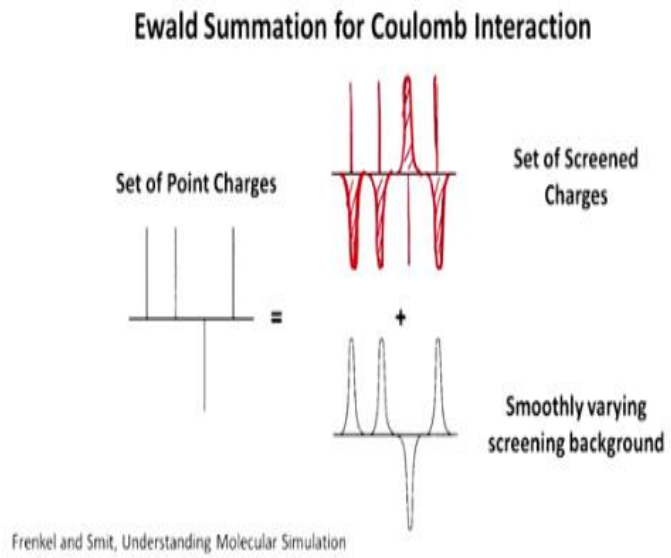
Particularly when we are studying interactions in salty solutions where you have some salt in addition to the species, we are interested in that salt happens to add lot of screening in the Coulomb interactions and therefore coulomb interactions start to behave in some limit as like a very short range interactions, the cutoff interactions is a valid approach to do and this is what is used in the Yukawa potential where you have essentially a coulomb interactions along with electrostatic screening.

But if this is not true, then we have to think of mechanisms by which we can incorporate long range interactions because electrostatics is one of the most common interactions if there is not enough screening in the substance in that case let us say for example, we are working in the lower salt concentrations or no salt concentration or dilute solutions. In those cases it becomes more critical to ignore the interactions beyond the cutoff distance and in that case is we have to think of something else. And more often than not whenever we are looking at charged systems, we really have to think of some mechanism to incorporate long range interactions. So, one of the ways to implement that in code is known as the method of Ewald summation.

So, the key idea of the Ewald summation is the following. If you think of any atomic system or molecular system, you may imagine that the simulation box containing the molecules or atoms will essentially contain a set of point charges because there can be charges located within each of the molecule and those charges essentially are point charges. So, simulation box will see so many point charges located throughout the simulation box corresponding to the number of charged species the simulation box.

Now those set of point charges can be approximated and this is a mathematical trick as a set of screened charges that means for every point charge I add an opposite compensating charge. But the charge that I add is added in somewhat smeared form, so we get somewhat like a continuous Gaussian like distribution of charges as opposed to point charges located at a point. So, simply a mathematical trick so for every charge I add a smeared charge which have the same magnitude but the charge is not I would say a delta function but it is some kind of a Gaussian function that goes to delta function in a limit, but a Gaussian function that is continuous in nature.

(Refer Slide Time: 27:26)



So, for every charge that we have whether it is positive or negative, I add an opposite charge that is smeared in nature. So, you add some Gaussian opposite charges at every point. And now clearly this is all the this is an addition that we have done we have to add an opposite distribution to compensate that and this is what it has to do, you simply add the opposite cloud and if you sum these two be pretty much get back the set of point charges. You may ask why we are doing that and I will not go in the detailed map here, but the key idea is that it is simply a mathematical trick that will make our job easier. If you are interested you can go to the book of Franken and Smith that is understanding molecular simulations and you can find a detailed derivation. I will not go in details here, but this is the trick that that is adopted in doing that.

The next thing we note is that when a system is under a periodic boundary condition, we have said that the simulation box essentially repeats itself although we are not modeling the images as such we simply simulate the simulation box all the image coordinates can be found in terms of the coordinates in the box but nonetheless less the assumption in periodic boundary condition is that that entire simulation box repeats itself throughout the large volume or infinite volume as we want to imagine.

So, in that particular case, what we can think of is the simulation box itself is like a unit cell of an infinitely large crystal. Because this is what is repeating throughout this. This is not to say the molecules from a crystal, molecules within the simulation box can be in any form. It can be amorphous can be liquid can be a solid. This crystalline behavior is appearing at a much larger scale that is at the scale of the simulation box itself. So, the whole simulation box there is containing billions of billions of molecules that hold simulation box repeat cell so, that

configuration is repeated that is an artifact of the periodic boundary condition this should not have been there in the first place but it is better than for example putting a hard wall that would create more problems than compared to the periodic boundary conditions.

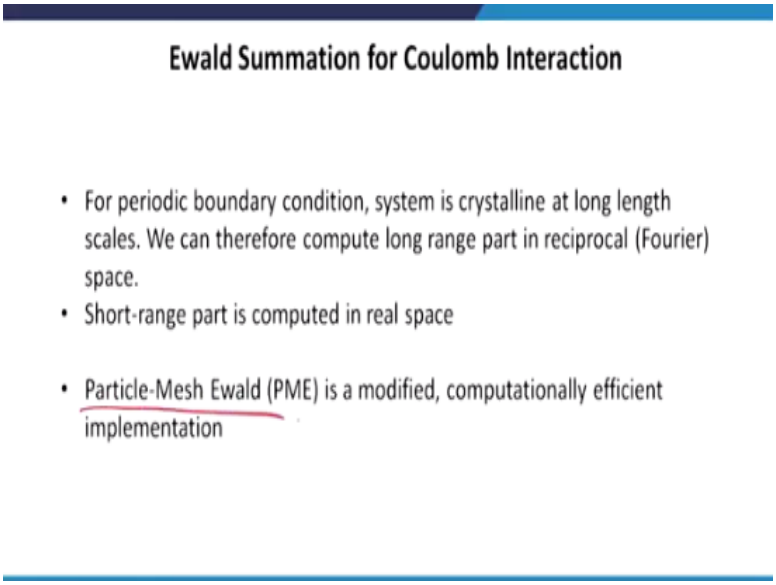
But this really helps me in some other way. So, when I assume a crystalline behavior of the simulation box we can use the math that is used to analyze crystals and it turns out that the crystalline systems can be understood in terms of representation in terms of unit cells which can be easily represented using the idea of Fourier series. And therefore for interactions larger than the box size or larger than half the box size to be more precise we can use the idea that the simulation box is crystalline and in that limit we can solve for the equations of electrostatics in Fourier space as opposed to real space.

Now the equations remain the same instead of writing in the real coordinates x, y, z. I am writing in terms of Fourier space coordinates k_x, k_y, k_z, just symmetrical manipulation. But that really makes computation much faster for the interactions happening beyond half the box size. For short range interactions for less than the box size, we still compute interactions like the pure Coulomb interactions, in the real space that is in the x, y, z space but at larger distances like more than half the box size or it can be even smaller than the box size and some cases, we use the math of the Fourier space and we represent the interactions in the Fourier space.

Finally we get the energy value as a scalar quantity but the computation is performed in a Fourier space. We create some sort of a grid in Fourier space that is in k_x, k_y, k_z in Fourier space where k_x, k_y and k_z are the wave numbers in the x y z dimensions of the Fourier space and that really becomes I would say computationally more convenient.

So, I have pretty much skipped all the mathematical details in the Ewald summation part, but just to give you some motivation of like how exactly we handle the longer range interactions. I give you these two tricks that are employed if you are interested to learn more about it I recommend you to read the Frankel book and go through the detailed derivation of this.

(Refer Slide Time: 33:26)



Now it turns out that there is another scheme that is called the particle Mesh Ewald scheme that is somewhat modified and computationally more efficient implementation of the Ewald summation. If you look at any particular software of molecular simulation it is more than likely that we have implemented a particle Mesh Ewald scheme. So, we can pretty much use that for granted in most situations but keep in mind that you will have model parameters of that scheme or the numerical parameters of that scheme, because in that scheme corresponding to the Fourier space variables because we are representing the long range electrostatics beyond some small cutoff by in the Fourier space and as you introduce more grid in the Fourier space, you will get more computational efficiency you only have to think of not in the Cartesian coordinates but in the Fourier space coordinates.

So, with this particular idea, I want to conclude the discussion of the pair potentials. Keep in mind the two basic messages that I wanted to convey one is that in most cases we pick a pair potential that fit ab-initio or experimental data. The second is we try to make the computation of distances and forces computationally efficient. So, not only we want to use cutoffs, but we also want to use schemes like verlet list or cell list whenever we have a short range interaction, when we have a long range interaction, if we really have to include that meaning that there is not enough screening in the interactions in that case, we should go for the particle Mesh Ewald scheme or the Ewald scheme. There are several other methods out there but these are the ones which are most commonly used in molecular simulations.

So, with that I want to conclude the discussion today, thank you.