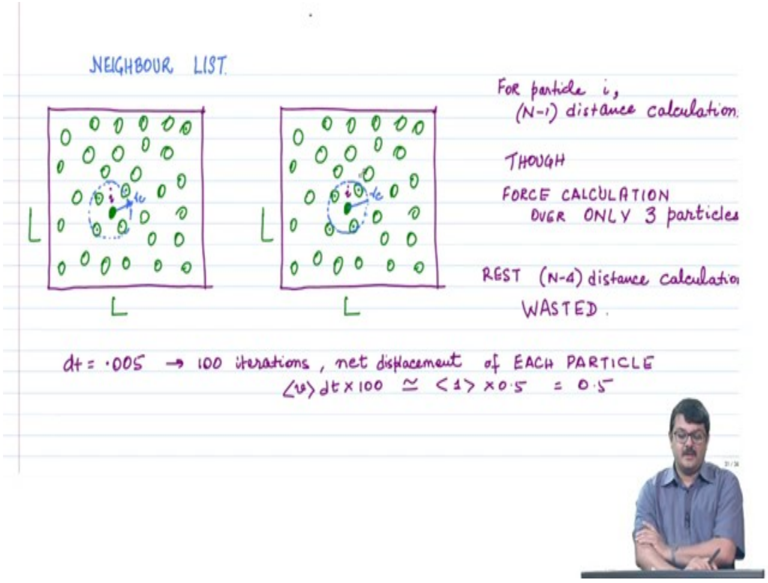
# Computational Physics
**Dr. Apratim Chatterji**
**Dr. Prasenjit Ghosh**
**Department of Physics**
**Indian Institute of Science Education and Research, Pune**

## Lecture - 53
## Molecular Dynamics Neighbours Lists Part 01

(Refer Slide Time: 00:17)


NEIGHBOUR LIST.

For particle $i$,
$(N-1)$ distance calculation.

THOUGH
FORCE CALCULATION
OVER ONLY 3 particles

REST $(N-4)$ distance calculation
WASTED.

$dt = \cdot 005 \rightarrow 100$ iterations, net displacement of EACH PARTICLE
$\langle v \rangle dt \times 100 \simeq \langle 1 \rangle \times 0{\cdot}5 = 0{\cdot}5$

So welcome back and today we are going to discuss the idea of Neighbour Lists. So, in the last class just to remind you we discussed about thermostats and what do units mean and today with the idea of neighbour lists we shall be able to significantly speed up our computer simulations of molecular dynamics. Before even we start neighbour lists let me give you a bit of background; now as you know in molecular dynamics you are basically accessing different microstates and you are integrating Newton's equations of motions.

Now, to get relevant statistical quantities you really need to run the simulations over extremely large number of iterations. Typically the minimal numbers would be something like 1 million because even with the simple Lennard Jones interactions the particles have to move around each microstate has to be accessed which is significantly different from the previous microstates and only then can you take a snapshot or you can take that microstate and from that extract relevant statistical quantities.

And of course, to get a relevant and good average you need to access large number of microstates so that your averaging is good and you can compare with your thermodynamic quantity. So, you are basically calculating thermodynamic quantities by you taking suitable statistical averages and for that you have to really run the simulation for extremely large number of iterations. And if you have suppose around 5000 particles interacting with each other in a simulation box, the way I have discussed in the past where each particle basically you check for the distance between each pair of particles.

So, if there are N particles then you are basically checking whether the distance whether the distance between a particular pair of particles is lesser than r c or not and this distance checking you have to do for N into N minus 1 by 2 numbers and only if the distance is less than r c then you calculate the potential.

So, whereas, it might seem that you are saving the calculation of potential for a large number or the force um, so, saving the calculation of the potential of the force for a large number of particles if suppose this is your particle i and this is your range of the interaction r c the potential is cut off at r c and what you are doing for this particle you are also checking the distance between this particle.

And this particle say right and this particle and this particle say in this particle and this particle say and only if it is less than r c value are calculating the potential, but this calculation of distances between this particle and this particle is not of a any much use right. I mean your distance calculation is important I mean you have to do it, but if somehow the number of pairs over which you are checking for the distance between this particle i and the other particles if the number of checkings of distances whether it is less than r c or not could be reduced significantly that would result in a significant speed up of your computational code right.

And to explore this idea I would like you to focus on this. And suppose this is a particle i and it is surrounded. So, there are only suppose only around 3 particles this and this within your cutoff distance. Now after 1 iteration this particle would have moved right. How much you did I have moved?.

Now if suppose dt is 0.005 the integration with which you are integrating the Newton’s equation is dt equal to 0.005, if your k b t was 1 that is what we have fixed till now you can change it of course, then the average speed of this particles would be order of 1 for half m v square equal to kbt; so, its order of 1. So, in one iteration the average displacement the expected displacement of say particle i here would be around 0.005 right which is significantly smaller than the diameter.

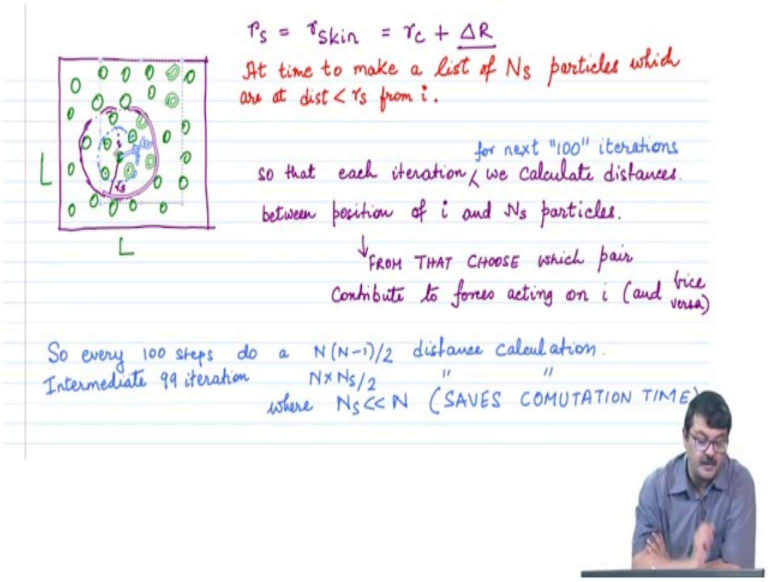
And if you consider 100 iterations even 100 iterations each particle is moving, but if you just focus on particle i, it would have moved from here to around a distance of 0.5 v d t in 200 iterations at 0.005 into 100 which is around 0.5. This is not much its not much compared to the its basically half a diameter all right if sigma equal to 1 significantly smaller than the size of the box.

So, this particle would have moved some small distance from here to here right. So, this particle; so, it was this particle was within its radius cut off. So, it was interacting there was a finite force that it was feeling from this particle after 100 iterations it might have moved half its radii where this particle i th particle is not interacting with this particle.

But now it is interacting with this particle right its a consequence of the motion, but still particle i is not interacting with this all these particles and however, you are calculating the distance between this i and all the other particles and similarly each particle and all other particles to figure out whether it is experiencing some force it will experience some force if the distance between particle i and its neighbours are less than r c right.

So, the question is since in even 100 iterations a particle i moves relatively small distances can we somehow do away with the calculation of distances with all these particles which are much farther away from the particle of interest here i and if we do not need to calculate over the distance of particle i with this large number of particles we are basically saving computation time and neighbour list allows you to do exactly that. So, that is the motivation for neighbour list or how one should think of the neighbour list.

(Refer Slide Time: 07:36)



$r_s = r_{skin} = r_c + \underline{\Delta R}$

At time to make a list of $N_s$ particles which are at dist $< r_s$ from i.

so that each iteration for next "100" iterations we calculate distances between position of i and $N_s$ particles.

↓ FROM THAT CHOOSE which pair Contribute to forces acting on i (and vice versa)

So every 100 steps do a $N(N-1)/2$ distance calculation.

Intermediate 99 iteration $N \times N_s/2$ " "

where $N_s << N$ (SAVES COMUTATION TIME)

Now, let us get down how a neighbour list is calculated ok. Now, suppose this is particle i th basically taken the same figure which was in the previous slide and this is your r c and these 3 particles are within this are within the distance r c at time t equal to 0 say right. Now at time t equal to 0 you make you draw another circle or sphere if you are in 3D you go off radius r s; r s this s stands for the word skin and the radius of this larger circle which is definitely larger than r c is r c plus a certain delta R.

So, from here to here is delta R ok. So, you basically make a list of all particles which are within this distance r s you do it at time t equal to 0 and for the next 100 iterations when you are calculating forces on particle i when you are calculating forces on particle i you do not calculate the distance of all the particles which are in the simulation box and particle i instead you calculate the distance of all the particles with which are within distance r s.

And if the particle if the neighboring particle if a neighbour of i is within distance r c then you calculate the force what are you saving you are checking you are calculating the distance between a much smaller subset of particles only those particles which are within distance r s right and not all the particles in the box and within that small subset you are choosing those particles which are actually interacting which are actually within the distance r c.

So, you have saved a lot large number of calculations of course, you are going to ask that you know this particle i and each particle in turn on which you are calculating the forces how do move around in space. So, if this particle at time t equal to 0 was here it is going to move here after some time and similarly all the other particles are going to move around.

So, the particles which are within distance r s which are basically called the list of particles which are the neighbours at time t equal to 0 this the neighbour lists list of neighbours which are in close proximity you have to update that list say every 100 iterations or every 500 iterations on every 50 iterations depending on both the speed and depending upon the interaction.

So, you have a neighbour list at time t equal to 0 after time t 0 plus 500 iterations you are going to make another neighbour list again calculate which particles are at a distance r s within the distance r s from particle i, but now particle i is in a new position after 100 iterations and of course, all the other particles are also moved around.

So, you have a new list of particles and for the next 100 iterations again instead of calculating the distance between particle i and all other neighbours you just calculate the distance between particle i and the neighboring particles which are at time t plus 500 iterations. That after 5000 iterations you again call the neighbour list you have a new list of particles which could be neighbours.

So, for each particle i and I can run from 1 to N; N being the total number of particles and part of you like and a number of particles you will have a list of particles which are in its immediate neighbours and you do not expect this list of neighbours to change within the next 500 iterations 500 iterations is just an example depending upon the physics of the problem it could be 100 iterations or 50 iterations.

But you do not expect a particle which is sitting here right they are not moving at relativistic speeds or anything there are classical particles you know their temperature you know their Maxwell Boltzmann velocity distribution and you and if you have 5000 or 10000 particles you do not expect extremely high velocities or speeds and you do not expect a particle which is sitting here to come suddenly to become very sharply and straight towards this particle and start interacting in the next 500 iterations.

Of course this particle could interact with this, but then this has to move in this direction; in this direction say and this has to move in this direction and it would take a few thousand iterations before these 2 particles could come in proximity and start interacting with each other right. So, this is all that I have discussed here this is.

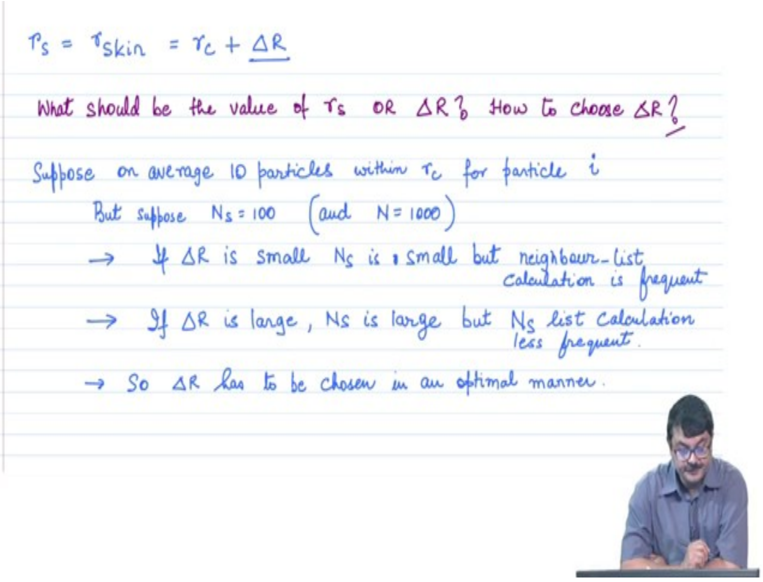
So, basically from a larger list of particles from N particles you are making for each particle i, i stands for 1 to N you make; you make draw a circle which is larger than r c and makers list of particles with which it could interact in the next of few iterations and every say 500 iterations you actually check the distance of a particle i with all the particles in the box and from there make a sub subset call it N s if you like corresponding to skin.

So, a smaller list of particle and the next 500 iteration do not do a distance calculation over all the particles just do a distance calculation over the small list and figure out which particles it is interacting with even. So, as it moves in the next 500 iterations it could move from here to here. So, it starts interacting with this monomer and does not start interact with this monomer, but it is unlikely that it is going to interact with this monomer.

So, every 500 iterations you are basically updating the neighbour list now if N s. So, what do you get? So, every 100 step you were doing N into N minus 1 by 2 distance calculation, now you have a smaller list of particles N s and for the next 500 iterations say you are actually the number of distance calculations you are doing is N into N s by 2 instead of N into N minus 1 by 2.

And if N s is significantly less than N you save computation time right; now the question comes is ok. So, what should be the value of r s or how big are you going to choose r s right?

(Refer Slide Time: 15:46)



$r_s = r_{skin} = r_c + \underline{\Delta R}$

What should be the value of $r_s$ OR $\Delta R$? How to choose $\Delta R$?

Suppose on average 10 particles within $r_c$ for particle i

But suppose $N_s = 100$ (and $N = 1000$)

→ If $\Delta R$ is small $N_s$ is small but neighbour-list calculation is frequent

→ If $\Delta R$ is large, $N_s$ is large but $N_s$ list calculation less frequent.

→ So $\Delta R$ has to be chosen in an optimal manner.

And for that to ask this question what should be the value of r s or delta R and how to choose that delta R? Here let me tell you as a motivation of how to choose what should be the logic to choose delta R. Now suppose for a particular particle i you have 10 particles into a within r c for particle i write at a certain time and; however, you have chosen a relatively large value of delta R.

So, that r s is significantly large I mean it is much smaller than the size of the simulation box, but still its large and so, that N s is 100 suppose there were 1000 particles in the simulation box and instead of doing a check of N into where N is 1000 N into N minus 1 by 2 you are now 499 iterations you can live with N into N s where N is 100 distance calculation. So, you are already saving a factor of 10 distance calculations right.

So, that is good you have saved a bit, but if N s is 100. So, even when. So, in this figure, so, from N s you are actually the particle is a particle i is interacting with only 10 particles. So, even you for the next 500 or 499 iterations you are checking for the distance for these extraneous 90 particles right I mean of course, some of them.

So, right now it is interacting with certain 10 particles after say 100 iterations it could interact with a different set of 9 particles because they would have moved in space because all the particles would have moved in space, but still you are calculating the distance between around 90 particles extra right.

So, that is costly; however, if you chose delta r to be relatively smaller. So, that r s is smaller right then N s could have been. So, N s could have been say 40 and that is again a factor of 2.5 speed up from where N s equal to 100 right. So, basically you have chosen a smaller value of r s which would in turn lead to a smaller value of N s say 40 right. So, it is this case, but in that case since r s is small within your 500 of iterations right this particle i could itself move and come here. So, that it would start interacting with this set of particles right.

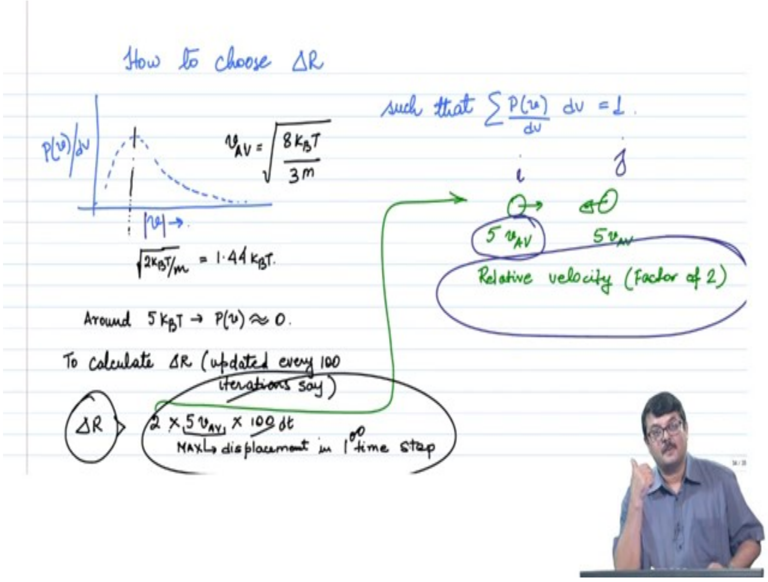
So, if you chose r s to be small then you have to update your neighbour list the list of neighbours more frequently instead of 500 iterations you have to do it every 100 iterations. Now calculating the neighbour list where you have to do I N into N minus 1 by 2 distance calculation pair and. So, that is the number of pair of neighbours and you have to calculate the distance that is an extremely costly step as well.

So, if N s is too small you save on your each force calculation you do not have to calculate over 100 neighbours you can calculate over just 40 neighbours maybe and actually the particle might be at a certain point of time might be interacting with only 8 or 9 neighbours it depends upon the value of r c of course, but then your neighbour list calculation becomes more frequent on the other hand if delta R is large. So, that r s is large here.

Then you can call or update the neighbour list less frequently, but then each in each force calculation you are basically calculating over a large number of neighbours extra neighbours whereas, the particle is actually interacting with only 10 and that is an extra cost message. So, delta R cannot be too small delta R cannot be too large both will. So, of course, having neighbour list will speed up, but it would not speed up optimally.

So, you will get the maximum speed up or the optimal speed up if delta R is chosen in an optimal manner right. So, if it is not too large and if it is not too small and for every problem and density and d t and r c one has to check out how much should one choose r s. So, that your calculation or your speed up is the maximum; so that that is first part of the thing ok.

(Refer Slide Time: 21:07)



How to choose $\Delta R$

$v_{AV} = \sqrt{\frac{8 K_B T}{3m}}$

such that $\sum \frac{P(v)}{dv} dv = 1$.

$\sqrt{2K_BT/m} = 1.44 K_BT$.

Around $5 K_BT \rightarrow P(v) \approx 0$.

$5 v_{AV}$ $5 v_{AV}$

Relative velocity (Factor of 2)

To calculate $\Delta R$ (updated every 100 iterations say)

$\Delta R > 2 \times 5 v_{AV} \times 100 dt$

MAX $\rightarrow$ displacement in $100$ time step

So, the question is how to choose the value of delta R ok. So, that it is optimum suppose you have already decided that you want to update your neighbour list every 100 iterations 500 is too much because the particles are moving because basically you will have a large value of r s, so, sorry r s and thereby N s. Now if you are updating your neighbour list 100 iterations how should you choose the value of delta R?

For that we basically take the help of statistical physics Maxwell Boltzmann distribution. So, here I have plotted the probability density of velocity or sorry speed right. So, it is a probability speed distribution Maxwell Boltzmann speed distribution p v d v by d v and on the x axis you have speed right.

So, this is the magnitude of speed and of course, there is a peak like this and you might know already that the position of the peak if you just revise your kinetic theory of gases the position of the peak is at route to K B T by m. So, this revise of statistical mechanics and you will know that the position of the peak is at root K B T by m reminding you that K B T we have chosen to be 1.

We are measuring every energies in units of K B T and m is also chosen to be 1 and if you have any confusion about this aspect K B T what does commit equal 1 means just refer to the previous class where we have discussed units. So, the, so, if K B T is 1 then basically the this velocity this the velocity the speed at which you have the peak is at around 1.44 K B T and what is the v average just to remind you from just to remind you from your kinetic theory of gases it is it was 8 K B T by 3 m.

So, 8 by 3 is approximately equal to 3 and square root of that is 1.7 order of magnitude of course, only if 8 was there was 9 you would get square root 3, but that is what the order of magnitude is that is what the approximate value is right. And if you have around 1000 to 5000 particles you would see that you hardly have any particle which has peace of greater than 5. You might find a few very few, but of course, as you increase the number of particles it is instead of say 1000 or 5000 particles you had a million particles.

Then you would find some particles for very rare occasions to be moving with speeds of 5 I did not mention the units you should refer back to the units to mean to understand what speed of 5 means coming back. So, if you had million particles maybe you will find some very few particles moving with a speed of 5, but hardly any particles moving with a speed of say 10 in these units; in these units were sigma equal to 1 and I told you how time it what time equal to 1 means in terms of K B T m and l right in these units. So, what is the message some message is the maximum speeds that you are going to see if you have around 5000 particles is around 5 or 6 not more than that. Now, we are updating your neighbour list every 100 iterations say. So, you do not want your particle to have moved so, much in your 100 iterations that the particle which is another particle which is at a distance greater than delta R would have come and start interacting.

So, basically delta R is the maximum distance is an estimate of the maximum distance that a pair of particles could move relative to each other in the 100 iterations. So, that they start interacting; interacting means less than distance becomes less than r c. So, basically r should be the maximum value that the relative speeds if suppose 2 particles were moving absolutely at very high speeds towards each other right.

And this particle was not. So, this is your particle i and this is your particle and under consideration. So, this particle should not in 100 iterations if it is not within the neighbour list if it is not in the neighbour if this particle is not in the neighbour list of this particle i at time t equal to 0 then in the 100 iterations it should not come close to each other in the next 100 iterations.

So, that it starts interacting right of course, if suppose this moves here in 100 iterations and this moves here then this particle will come within the neighbour list then your distance calculation will be there and if it is if they are interacting by a certain force that would be calculated, but if this particle was here very far away and not within the distance r s at time t equal to 0 this should not start interacting within the 100 iterations because you are not calculating for the distances for the for the distance between this and this for the next 100 iterations of this particle is not within the neighbour list right.

It is only after that you are going to do a recount that which particles are close to me which particles have a potetial to interact with particle i in the next 500 iteration see. So, basically delta R is a measure of the maximum displacements which are possible in the 100 iterations and you should choose delta R significantly large.

So, that within the 100 iterations no other particle which is not accounted for which is not in the neighbour list starts interacting with the particle i and the way you would do it is suppose one of the particles say i as let us take the maximum velocity has around 5 times v average right beyond that you are not going to find particles has such high speeds if you have a 1000 to 1000 particles 5000 particles in your simulation box.

And suppose another particle j was also moving at an extremely high speed 5 v average and they were moving towards each other. So, the relative velocity would be essentially a factor of 2 so, 10 v a v. So, in 100 iterations with 10 v a v being the average velocity what is the maximum displacement what is the maximum change in relative positions that can happen that is that will be 2 in 2 because 2 particles relative velocity 5 v average in 200 iterations right into dt because v into dt is displacement right this is for the maximum velocity this is relative and in 100 iterations what is the maximum displacement that can happen.

So, this is what would happen in one time step right or sorry this I made a mistake here. So, there should be dt also here and since there is a factor of 100 that is what how much it would move in 100 time steps and delta R should be greater than this distance. So, the point is delta R keeps an enough margin. So, that a particle which is far away does not come in and start interacting within the number of iterations over which you update your neighbour list.

So, you should update your neighbour list suppose every 100 iterations and then this is the maximum distance that the fastest pair of particles can travel can reduce the distance between each other and delta R should be at least greater than this number.