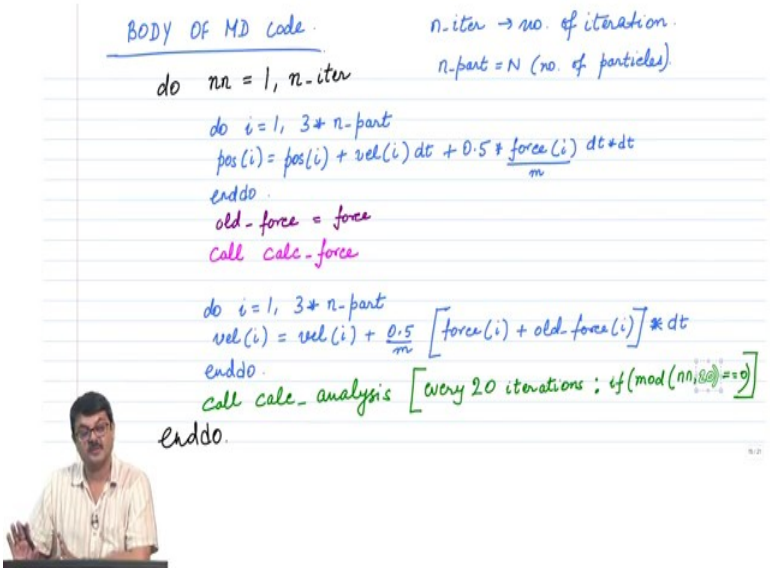
Computational Physics
Dr. Apratim Chatterji
Dr. Prasenjit Ghosh
Department of Physics
Indian Institute of Science Education and Research, Pune

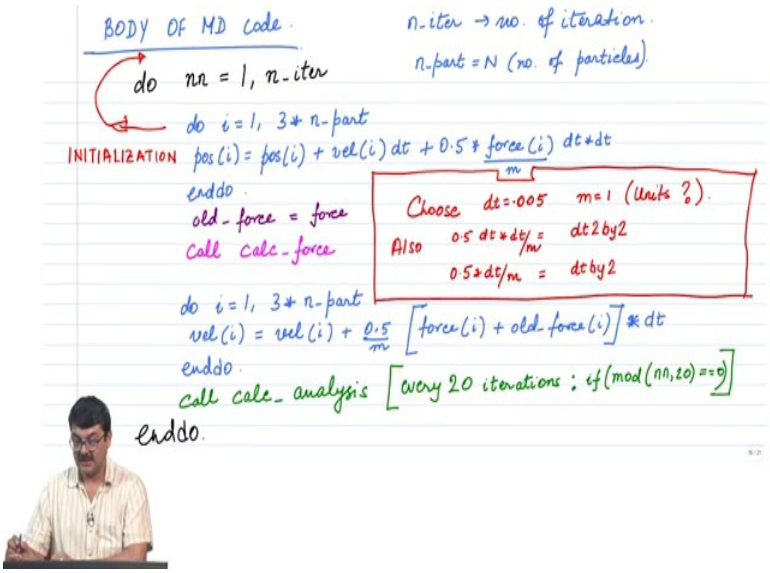
Lecture - 49
Molecular Dynamics Details and Algorithm Part 02

(Refer Slide Time: 00:16)



So, this is what is the main body of the MD code, but and there are few parts, there are some more points to discuss.

(Refer Slide Time: 00:22)



First thing you want to say that typically for the Lenard Jones interaction you can choose dt equal to 0.005 and by convention in most cases one can also take m to be 1. So, what is this gt equal to 0.005 mean; I mean, what is the unit of time, we will discuss it has to be it I cannot discuss all the things at once.

So, we will be discussing units, but typically if you have Lenard Jones with epsilon equal to 1 dt equal to 0.005 works pretty good and what does pretty good mean, we have to run the code look how well energy is conserved. If momentum is conserved and so on so forth even typically chooses m equal to 1. But they are these quantities like dt into dt into 0.5 by m. So, there is no point; So, you are when you are doing this calculation right there is no point in writing it like this, because this calculation dt into dt by m into 0.5.

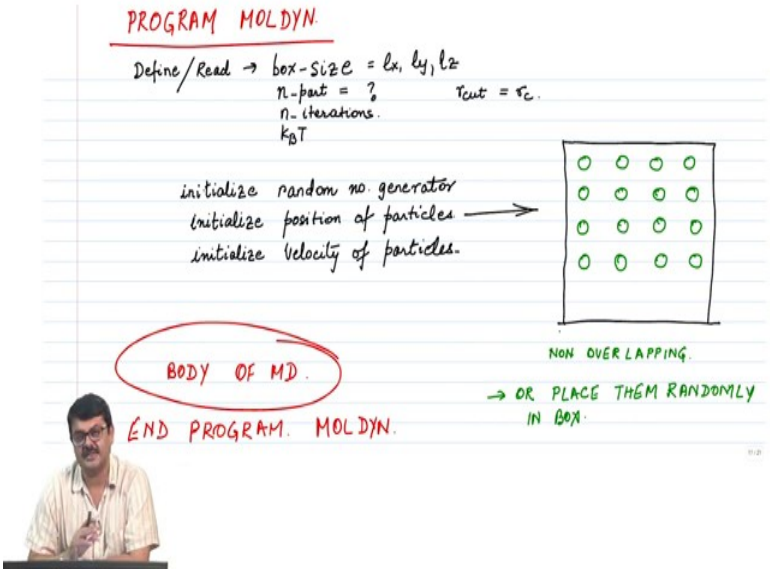
We are doing it all the time in each of these iteration and all iteration when I mean i equal to like 3 into n part. Every time it is doing this algebra and you are doing it so many times over n iterations over every time step. Instead if you basically replace this quantity by some other variable dt to dt square by 2 say dt. So, you can write it like this dt square by 2 ok, you can at least remember it, right at the beginning of the code. And here like you have 0.1;0.5 half by m into dt.

If you basically write this as dt by 2, right at the beginning of the code outside this loop essentially, right. Then the you save the computer time, because the computer does not have to do this multiplication every time, you might think to do a multiplication it take it

takes with giga Hertz like one nanosecond or something like that. But, then you have a large number of particles, you are going to calculated distances you are going to do this for large number of particles so multiple iterations. So finally, this takes times and every time it is doing a calculation this is like three multiplications, one division it takes same, so it should just replace these by suitable variables.

So, these details the molecular dynamics are important you do not write it like this, right. The other thing is even before we go to the body of the so called MD code when n equal to 1 that is the first iteration, you need the velocity of the particles you need the force of the particles and you need the position of the particles.

(Refer Slide Time: 03:42)



So, what have we not discussed here, initialization. How do we initialize the position of the particles, the velocity in of the particles? If you know the position you can calculate the force, because force depends upon the distance between the particles, right. So, even before we come to the body of the MD code, we need to discuss initialization, right.

So, here is essentially the body of the code, here basically the program MOLDYN starts here it ends. But, before that we have to define we have to define suitable variables and these variables can be read, you can give them on screen or you can write them on a small separate file. And you have to define definitely lx ly lz typically at least for the beginning you can choose to be a cubic box. So, all three directions they have the same size and so lx equal to l y equal to l z. You have to choose n part, the number of particles

you would choose the value of r cut or r c which is the cut off distance of the potential write it, you have to explicitly mention it.

And once you have your value of the r c, then you can calculate the so called f c what is the value of the force at the cut-off and then you have to calculate also v dash c v. If the potential the mod the value of the modified potential at the cut-off, those are quantities which you do not need to calculate every time. You calculate the force because those are fixed values you want to save computing calculation time, right.

So, as soon as you have access to this quantity you calculate the value of f c, you calculate the value of v dash c which you are going to use later, whenever you calculate the force other than that you have to mention the how many iterations. You want to run your code for that will depend, of course upon the physics that you want to want to extract from the system.

The how many iteration, how many independent runs you need to run that once you are running in empty in a regular manner, you get a hang of that. But anyway for testing you can choose it to be 1000, not 1000; 10000 iterations 1 lakh iterations, 10 lakh iterations just as we were doing for molecular for the ising model. Then you should and you could initialize the random number generator, you might need it and you will can need it even to initialize the position of the particles and the velocity of the particles.

So, how does one initialize the position of the particles, suppose this is your simulation box of size l x cube assuming l x equal to l y equal to l z. Your final physics, your final distribution of particle positions and the velocities of these particles is not going to depend upon your initial position, just like an ising model you can start with any initial configuration.

The easiest would be do you put all the particles you put all n particles at equal distances from each other and once the force starts acting and since they have initial velocities of as soon as you start the molecular dynamics. Of course, they are going to move around in space and how they are distributed whether they are clumped, whether they are all far apart from each other on an average will this will be decided by the potential, the density, the temperature, the thermodynamic quantities.

But just to start off you can just put them all at equidistant from each other say at a distance of I do not know 2.3 or depends upon the density also if it is highly packed you cannot fit all the particles of the box. If you start with distance of three each particle is surrounded by neighbours which are at a distance of 3 sigma.

So, here distances as I said in the last class is measured in units of sigma, so you set sigma equal to 1, the diameter of a particle equal to 1. And when you mention lx you say my simulation box is 50 times the diameter of a particle right, so that has been assumed. So, when I say that the distance between any pair of particles in the initial condition is 3, I actually mean 3 sigma where sigma has been set equal to 1, ok.

So, that is a possible initial position it is basically sitting on a lattice, of course the lattice is going to melt or not melt depending upon the potential and the density and the temperature so, or it could go into some other crystalline form. Here I am just talking about a simple cubic lattice one could always start Lenard Jones simulations from a lattice simulation or you could choose well.

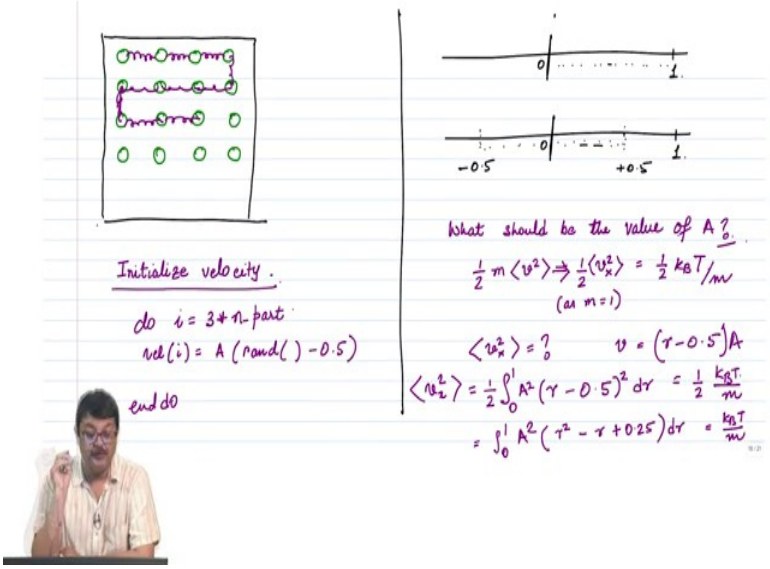
You know that you do not find that comfortable, you can choose put particles randomly in the box, you generate a random number generate a multiply it in to l x it will be some number between 0 and lx and you say ok. I shall place the first particle at this x coordinate and then you generate another random number multiplied by l y and that will be another some number between zero and l y which is basically the size of the box.

So, you can do similar things for the z coordinate of the particle and you put you say that you ok, I put it randomly somewhere inside the box. But, when you put the second particle you should ensure that the distance of the second particle and the first particle is greater than sigma. Because, if you by chance you put it right on top because they are using a random number generator, now put them right on top of each other right that would lead to extremely large forces as soon as this simulation starts.

So, typically when you put the second particle you do not have any problem, typically random numbers which has chosen you can put it some other point in the simulation box. But, as you put more and more particles there is larger and larger probability that one of the particles will sit on the top of an already introduced particle. So, if you want to basically introduce particles randomly we want to place particles randomly in a simulation box.

Then, you have to take suitable precautions you have to write a code and check that when you are introducing a new particle, one of those n part particles that they do not overlap with a previously already introduced particle right. You can do it or just put all the particles on a lattice at suitable distances, anyway the lattice will melt at least if you have Lenard Jones particles and you have significantly high temperature, right.

(Refer Slide Time: 10:31)



So, that is how you initialize the position of particles if you are introducing Lenard Jones particles is relatively easy, but you might also be ending up introducing or wanting to study more complicated systems. Suppose a rod like molecule right or a polymer. What is a polymer? You have a polymer base where basically the monomers the particles which constitute the polymer, right, that the monomers assume them to be spherical and they can be connected by springs.

So, you cannot necessarily in this case choose the position of n particles randomly, because the constraint that the distance between neighbouring particles on a polymer along the polymer chain, they should be at a distance of approximately the bond length that has to be maintained, right. You cannot say that over bond is extremely highly stressed that is unphysical.

So, typically you say the first particle is here, the second particle is here the third particle is here you put them basically keep on intervening the position here, but then you have to be sure that the fifth particle basically sits here and not here. Because then this; because

then basically this bond would be extremely stretched and if it is extremely stretched then the force acting between these two particles this is extremely high.

And then when you start the simulation or you run the iterations, the extremely high forces can lead to a breakdown of the integration unless you choose extremely small values of dt which you do not want to do, alright. So, for Lenard Jones introducing particles is simple, if you have more complicated systems you might have to take care how you want to introduce the particles. But, let us stick to the simple case I just gave you a caviet up warning that one has to be careful Lenard Jones is relatively easy.

Introducing the velocity you could say that you know each particle is moving in random directions right at the beginning. So, you do not want to give the all the particles the same velocity, they cannot be moving in all in a perfectly the same direction that would be again unphysical that is something like all particles moving as a whole. So, there is it is moving it is like a river all the particles are moving, unless you want to actually simulate that. But, let us first of all consider a static liquid or a gas, how would you model that because that is simpler. If you are modelling flows, that is more complicated we will leave it for future class or basically if and when you need it you can always learn that aspect.

So, in general you want the velocity of all the particles to be in random directions, more like what you have the gas particles in this room, they are all moving in all possibility directions. But, remember they have also in equilibrium the Maxwell Boltzmann distribution of speed and velocity, right. For simplicity you can say that you know what I want particles in random directions, but I am choosing the velocity from a uniform random number generator. And as the particles move around in space they collide with each other and do if they are doing the right physics, if you have model the right physics.

Then, we are going to generate basically a Maxwell Boltzmann distribution. In fact, that is going to be one of our tests that we are doing our molecular dynamics simulations correctly, ok. For simplicity case you say that I am using a random number generator here right, which generates random numbers between 0 and 1, ok. The velocity of each particle is some amplitude which we will discuss how you choose that amplitude into random number minus 0.5. Random number minus 0.5 what does it do? So, random

number would choose a uniform random number between 0 and 1 and if you subtract 0.5 from it, then basically the range of that random number becomes minus 0.5 to 0.5.

So, you choose any random value, which you choose that to be v x, you choose another one to be v y we choose generate a third random number from that generate v z. So, some particles will be moving in the positive direction, some particles will have velocities negative values it is moving in the negative direction.

So, on an average you essentially have equal number of particles moving right, equal number of particles on an average moving left that is how you start off. Then you have to ensure that your momentum in the x y and z direction goes to 0, because if it is not exactly equal to 0. That means, you have a box you have a simulation box which has a centre of mass velocity, which means it is moving you want to look at a stationary liquid, right.

So, after introducing the velocity of all the particles you calculate the centre of mass velocity of all the particles and subtract a certain small amount that exact amount from the velocity of each particle. So, that the centre of mass velocity of the n particles becomes exactly equal to 0, right.

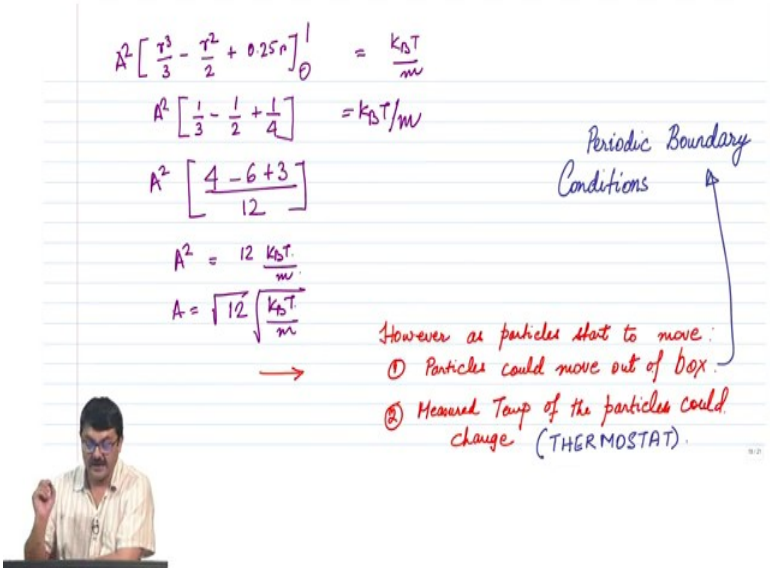
You have n particles half of them are moving to the left half of them to the right some on, it towards the camera, some away from the camera, some upwards and some downwards vx vy vz and it goes from minus f minus 0.5 to plus 0.5. The question is what should be the value of e some amplitude, there is no reason that it should be just simply you choose it from minus 0.5 to plus 0.5.

In fact, the way you have to choose it is once you have specified the temperature of the system right and you know that the equipartition theorem holds half m v square average equal to half k B T for each degree of freedom. Basically, the value of a should be such so, the average v square of n particles for each degree v x square average over n particles, v y square average over n particles that should be equal to close to half k B T right, that is what I have written here. So, half m v square or rather if you just take one particular degree of freedom half m v x square. It should be half k B T by m, I mean m could be here and then and m you can choose even to be 1, right.

So, you want to choose a v x, so that the average kinetic energy of the system is approximately half k B T, and this condition essentially determines the value of A. So, how do we do it? So, we have to calculate the expectation value of v x square. What is v? v x say I mean 1 component is add minus 0.5 into and some amplitude A and that is exactly what we have to determine. And if you have to calculate v x square average because where n is large we are generating so many different random values of v x and similarly for v y and similarly for v z, that if you average over all particles you will have fairly good distribution. So, you can take average v x square over the particles, right.

So, basically this v x square can be written as half; half m v square half 0 to 1, because that is the range of minus 0.5 whole square into A square and that should be equal to half k B T. So, half of this factor cancels and what you have is 0 to 1 A square, I have just expanded this r square minus r plus 0.5 whole square is 0.25 dr. Basically, you are integrating over the entire distribution of random number, so that is why 0 to 1 that is your range. So, r being the random number and that should be equal to k B T by m, right.

(Refer Slide Time: 19:29)

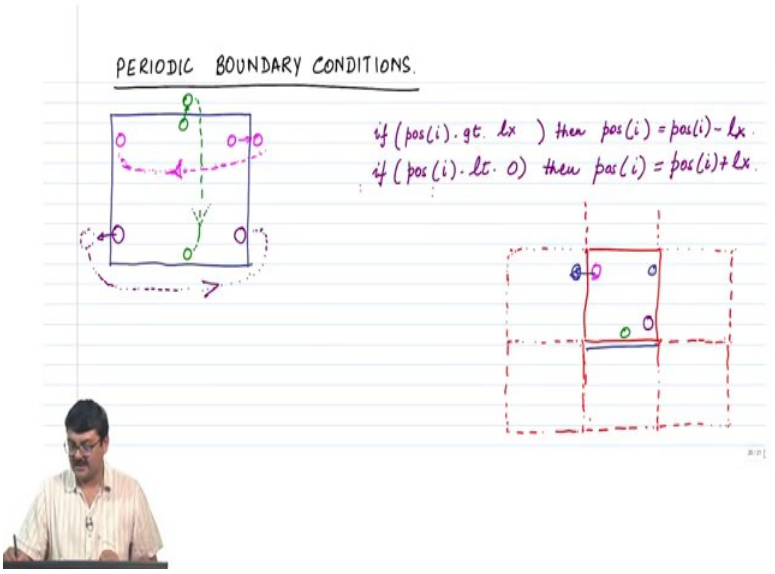


And if you work out the algebra here and I have worked it out from here for you, if you integrate it becomes r cube by 3 minus r square by 2 plus 0.25 r 0 to 1 and that should be equal to k B T by m. And if you work this out you essentially get A equal to root 12 root k B T by m and that is how you choose your amplitude of A. Amplitude of the velocity

so that when you start out your average velocity over n particles remains close to k B T, whatever value of k B T you have chosen, ok.

So, now you have we have discussed initialization of position, they have discussed initialization of velocities. Since, you have positions you can call calc force call calculation of force and you have the initial forces acting on each of these particles and now you are all set and nearly ready to basically go into the so called body of MD.

(Refer Slide Time: 20:50)



Except that you still have some things to learn and that is the following as the particles start to move you have a simulation box, you have to either implement walls. So, that once the particle suppose is moving and this is the end of the simulation box, one it goes it hits it should come back into the box. But, often when we are studying liquids we are basically studying practically an infinite number of particles and equal to 10 to the power 23.

We are studying the system in bulk if you like and if you implement walls and you have some 1000 particles, then there will be strong wall effects. Rather you want to study a system where all the particles are surrounded in all possible directions by identical copies of the system; so that you do not see confinement effects I think we discussed this similar stuff even for the ising model, right. You want to basically simulate the system in bulk, so that in each direction you have an identical condition. If there is an endless chain of endless number of particles whichever direction you see, and that basically brings up

the discussion of periodic boundary conditions. Where you say even if the particles go out from here they come in from the other side of the box, right.

Or else you could model walls and modelling walls is a pretty detailed topic by itself, we then by having explicit more walls you could basically confine all the simulated particles within the box and people do it. Especially when I want to study nanofluidics or you want to see systems in a nano confinement and so on so forth. However, if you want to compare with classical statistical mechanics, where you do not have confinement then you would rather be well off studying systems in bulk and then one would be better off using periodic boundary conditions. So, we have to discuss that.

The other point is basically that as particles start to move; as particles start to move from the initial position and velocities the depending upon the interaction between particles. They would increase the potential energy of the system or decrease the potential energy and similarly correspondingly, the kinetic energy of the system also changes.

The average kinetic energy because the average kinetic energy of the system also changes because, if the potential energy decreases suppose there is an attractive interaction all the particles come close together and set at the potential minima. Then the potential energy decreases to keep the energy constant the average kinetic energy of the system has to increase, which means that the particles are moving around with higher speeds and half m v square average is equal to half k B T.

So, if the average kinetic energy of the particles have increased then what you have is essentially that the temperature of the system has increased, but you wanted to study the system at a particular temperature which you want to decide, right. So, what we need is the concept of a thermostat. Thermostat is something which will keep the average kinetic energy at a value so that the temperature is maintained.

So, that half (Refer Time: 24:47) by 2 and k B T equal to half m v square your equipartition theorem; which means the thermostat will either take energy out from the system readjust the kinetic energy of the system or put in energy into the system right, that is exactly what you have in a canonical ensemble, right. So, energy is lost on an or energy is gained their energy fluctuations.

So, we must introduce a thermostat, if you want to study a system at a particular n v t. we do not have a thermostat then what you are essentially studying is n v e ensemble, where the number of particles, the volume of the particles and the energy of the particles is constant. And the physics of the problem of course decides, which instability will choose we shall discuss both n v e and n v t, ok.

In today's class we will basically discuss the periodic boundary conditions and thermostat shall be discussed in the next class and so how do we implement periodic boundary conditions. So, that so when; why do we need suppose this is a particular particle, you have updated to this position there are some forces acting on it and it is position gets modified it gets updated and such that the position of the particle is larger is it is goes out of the box. And then you must put it back you should basically put it back here, right.

So, because it was out of the box and that is basically should put it back here, similarly if a particle goes out from the top. So, it is again the position of the particle is such that the value of suppose the y is y coordinate value is greater than l y, then it should be put back into the box here. Because this is basically put it back from the other side.

And similarly suppose a particle is a position gets updated. So, that it position it moves to the left position x coordinate decreases and becomes greater less than 0 even. You imagine the boxes from 0 to lx 0 to l y 0 to lz, but suppose it is position is such that it becomes less than 0, it depends upon the value of the force and the velocity, of course the magnitude and the sign of velocity and force.

If it becomes less than zero then it should be put it back here. How do you implement that? How do you implement PBC? You basically you if the position after the position gets updated let us look at that, so here. So your position update was happening here in the main body of the code and after it is position gets updated here you write down the conditions to implement PBC and what do you write down, you write down precisely this.

If position is greater than l x which means the position of the particle after update is here greater than l x, then pos i equal to pos i means the i th particle really you are also mentioning whether it is x or y or z. It is pos i minus l x and if you are doing position y if you choose the size of the box, so that in 3 different directions l x and l y and l z are

different. Then you have to be careful whether you want to subtract l x or l y or l z, but if you take a cubic box you are safe. It is just for the sake of learning and then basically lx equal to l y equal to l z then it does not matter right.

And if on the other hand some particle goes out here or goes out here or rather this way. Then if pos i is less than 0, then pos i equal to pos i plus l x which means it has gone here you are subtracting the length of the box and it will sit here. Whereas, if it goes out from here you are adding the length of the box, so you are basically adding the length of the box and it is going to set here.

So, as a consequence of periodic boundary conditions, what you have? You essentially have your main simulation box which is drawn here and as the particle goes out, right. So, maybe from here right, so that it can sit here or you could just put it back here. So, this putting it back here is the same as putting I mean, if it goes out you are putting it back here, that is what the PBC essentially does is that the main simulation box is surrounded by identical copies of the same box in all possible direction.

So, hence you are seeing this system in basically bulk because, imagine in any direction you have essentially statistically similar systems. Of course, the size of the box should be significantly larger than the diameter of the box and especially if you are near critical point, there could be correlations and these correlations could extend to infinity.

So, one has to be careful of course, but if you are away from the critical point, which is the case we shall study in this case. Basically, correlations between positional correlations die down between 4 5 6 times the diameter of the particles and if size of the box is supposed 20 or 30, 50 sigma. Then essentially statistically you have self similar system surrounding the main simulation box, right. With that we shall end today's lecture and we shall discuss the other point of thermostat in the next class, ok.

Thanks.