Computational Chemistry & Classical Molecular Dynamics
Prof. B. L. Tembe
Department of Chemistry
Indian Institute of Technology – Bombay

Lecture – 30
Classical Molecular Dynamics – 3: Periodic Boundary Conditions and Molecular Dynamics Algorithms

Let us continue our discussion that we started in the last lecture. In the last lecture we discussed several aspects of molecular dynamics and we ended the last lecture with this particular slide which talks of the equations of motion.

(Refer Slide Time: 00:31)

$$m_i a_i = F_i$$

$$F_i = \sum_{j \neq i}^{N} f_{ij}$$

Newton's Equations of Motion

$$\frac{d}{dt}\left(\frac{\partial L}{\partial v_i^{\alpha}}\right) = \frac{\partial L}{\partial r_i^{\alpha}}$$

Lagrange's Equations of motion

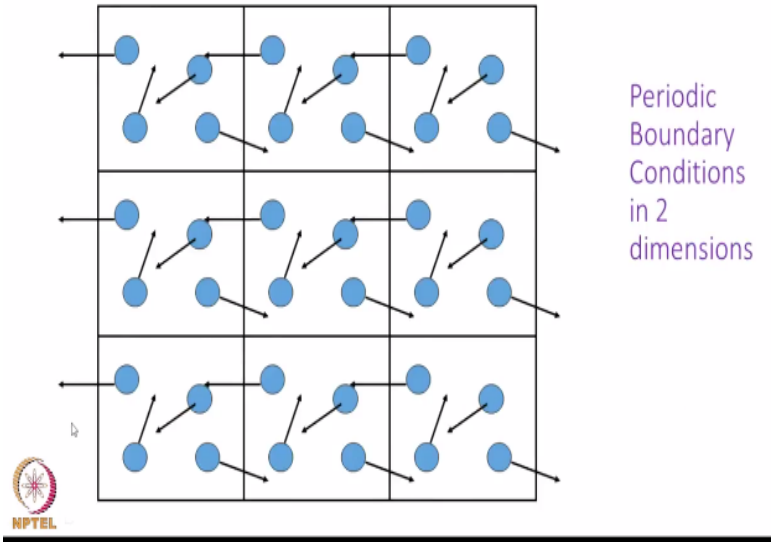
$$f_{ij} = -\nabla_i V(r_{ij}) = 24\frac{\varepsilon}{r_{ij}^2}\left\{2\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right\} r_{ij}$$

Lennard-Jones forces

So the top line shows the Newton's equations of motion and this force is nothing but gradient of the potential. What is the meaning of the gradient? You plot the potential along a distance and take the slope along that direction, so that is my force. Force is the gradient of the potential and this is the formula for the force and to find this formula for the force you need that model for the potential.

In this case the model for potential was the Lennard-Jones potential. So this would be the force between any 2 particles i and j. So now suppose we consider the whole system.

(Refer Slide Time: 01:06)



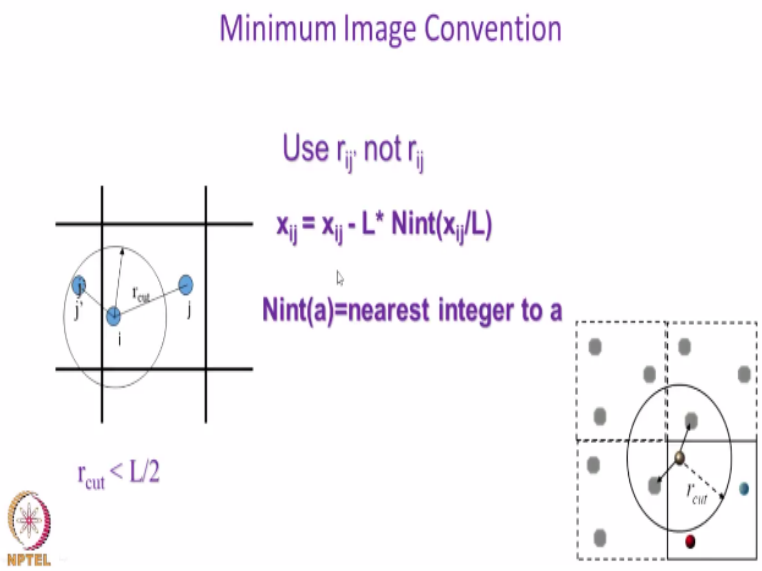
So next slide this shows my system. So my original system consist of only 4 particles, it could be 4, it could be 4000, but each of this box is surrounded by similar boxes on all sides. Now why do I surround this box on all sides by similar boxes, because if I continue this repetition in all directions so I have an infinite system. So an infinite system will really behave like a liquid, not like just a box.

So that is what is the meaning of periodic boundary conditions. The meaning of periodic boundary condition is that whatever happens in this central box is repeated in all other boxes. See for example suppose this molecule here moves into this box. So if this molecule moves in this box, this molecule move in the box to the next side, okay. So if one molecule moves to the box to the right, another molecule will move to the box to the original box.

So what periodic boundary conditions guarantee is that the density of my system will not change. So periodic boundary condition ensures that the density will not change and in addition to that what we do is called a minimum image convention. So before I go to this minimum image convention. So let us understand this properly again. Each box is surrounded on all sides by boxes of a similar type.

So in 2 dimensions it will be surrounded by 8 boxes. What are those 8 boxes? 1, 2, 3, 4, 5, 6, 7, 8. So in 2 dimension each box is surrounded by 8 boxes, in 3 dimensions the central box will be surrounded by 26 boxes, make sure you get this right because not only there will be these 8, there will be 9 boxes behind the screen and there will be 9 boxes in front of the screen. So 9 + 9 that is 18 + these 8 so there will be 26 boxes surrounding the central box.

(Refer Slide Time: 03:16)



So that is my periodic boundary condition and the advantage of periodic boundary condition is that each particle in my periodic system will interact with all the particles around itself okay. So this is called a minimum image convention. So what is the meaning of a minimum image convention. Each particle interacts with all it is neighbours in such a way that this box, this particle is treated as a center of the box.

So for example if this molecule is the end of this box it will not interact with molecule j in the same box, it will interact with molecule j prime because that j prime is closer to i than j. So that is the meaning of minimum image convention. Minimum image means around each particle I construct a box of the same size as the original box and the particle will interact with all the particles in that box with this particle as the center.

So when you, what it means for particle i in this case I will use rij prime and not rij okay. So when I do a computer program how will I know whether I should calculate j prime or j. So this is an algorithm for that okay. This is an algorithm, so that algorithm says xij that is the distance between i and j will be actual xij okay. So this is the original xij - box length times Nint xij/L xij is this distance and L is the box length.

So when I divide xij/L, so I get an integer, so if I take the integer part of that and multiply by L I will get the real xij. So what is Nint a, nearest integer to a. So nearest integer to a. For example, this if xij is more than the half box length. Suppose xij is 0.7 times L so then the

integer is 1. So if this xij is 0.7 times L so I will not take the original xij, but I will subtract that L because now it has become 1. So I will take xij prime.

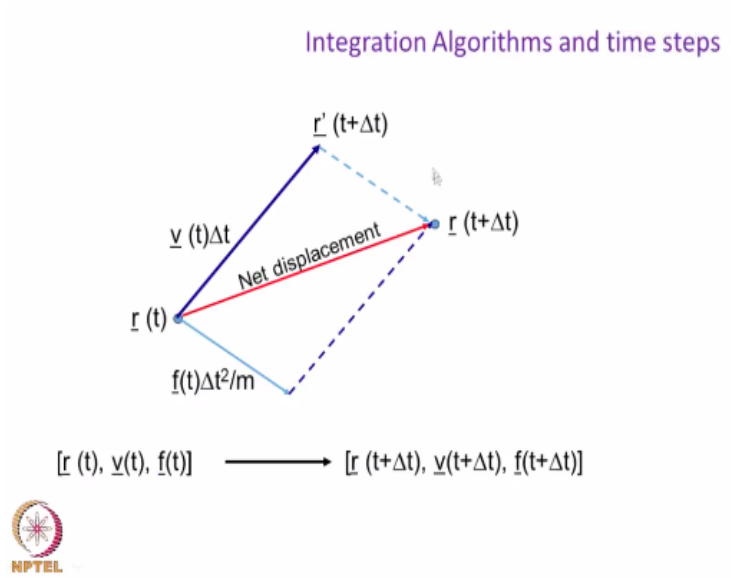
So this ensures that I interact with only those particle which are within the half box length about this particle okay. So the same thing is illustrated here okay. I have a particle in this box so this will not interact with any particle outside that cut, cut means L/2 is a cut off length. All molecules whose length is more than L/2 are not considered at all, okay. So that is the meaning of saying that each molecule is treated as the center of the box in which it is located.

So each molecule behaves as if it is in the center of that box. So and the great advantage is that the boundary effects are taken care. So you know suppose I did not have periodic boundary conditions. If I did not have periodic boundary condition this particle will interact only with these particles and this will behave as if it is in the edge of the box whereas if I have periodic boundary condition, there is no edge any more.

Each particle is surrounded by a similar box around itself. So the edge effects are gone. So it is a very standard practise to do molecular dynamics with periodic boundary conditions and minimum image convention, the only problem would be if there are coulombic interactions this situation is not adequate, because in a coulombic situation there are long range interactions. So therefore I have to consider things like Ewald sum or reaction filed.

I have already mentioned this in my last class and when you do GROMAC simulations you will have more occasion to use these particular ways of doing things.

(Refer Slide Time: 06:53)



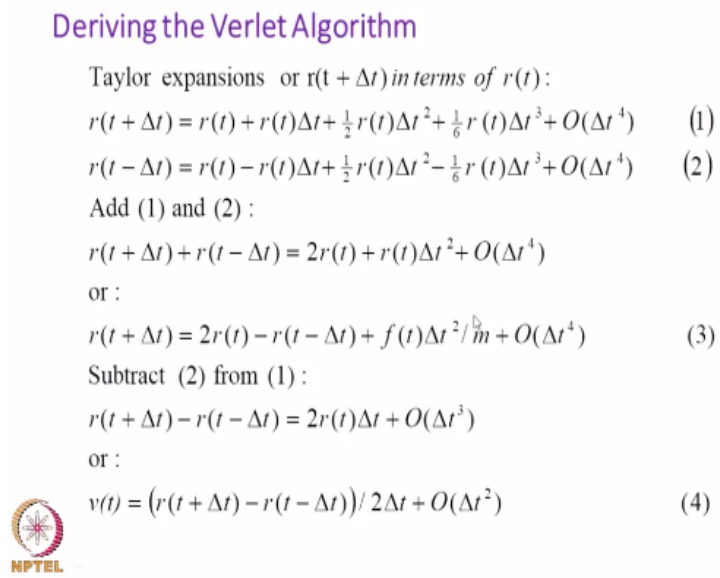
So now let us do this integration algorithms, what is an integration algorithms, I have a position r and in the next, suppose I have a time step, in all our simulations delta t will be the time step. So after this time delta t where will the particle be located. So to find that out, the particle will be acted upon by velocity as well as the acceleration. So the new position will be velocity * delta t.

So this is by the velocity term okay, and this is due to the acceleration term, okay, this is the acceleration term. So the net displacement would be a sum of velocity term and the acceleration term. So the new position will be r at t + delta t. So that is how I have to do my simulations. I have original position r, original position t, original force f of t at this particular position. So when I calculate the new position okay.

I have calculated the new position by adding these 2 components to the movement. So that is my new position. Now how do I determine velocity? So that is the question now. How do I determine velocity, that is the challenge then once I have the new position, in that new position there will be new force, so what will be the new force at this point? I need the new velocity which is already given here.

I have to calculate the new force then I will have the new position from this. So if I keep on going from one position to the next, next position to the other one, so that constitutes the trajectory. Trajectory means all the positions in a sequence that is my trajectory okay.

(Refer Slide Time: 08:32)



So as I mentioned how to calculate velocity so that is the slightly tricky. So one of the oldest and the most used algorithm is called the Verlet algorithm. So let me describe here the steps of a Verlet algorithm. So what is done in Verlet algorithm, I do a Taylor expansion, this should be of not or, Taylor expansion of r at t + delta t in terms of rt. So I want to do a Taylor expansion new position in terms of old position.

So how do I do the Taylor expansion r at new position will be r at the current position + r.t, this should be r.t, I do not know whether that dot is available okay. So this should be okay, so this should be new position + velocity times delta t + acceleration times delta t square + third derivative, third derivate * delta t cube and so on. So remember that this is the velocity, not r, this is acceleration, okay.

So this is r at t + delta t. At t – delta t it will be rt – velocity * delta t + acceleration * delta t square – third derivative of r * delta t cubed and so on. So this is the new position. This is the old position. If I add the two, what I get is r at t + delta t + r at t – delta t is 2 rt + this should be force * delta t square. This should be r double dot t * delta t square + order t fourth or the new position is given by 2 times the old position – the one position before the old position t – delta t + force * delta t square.

So remember this r, it should be r double dot t, so that is force * delta t square/m okay. So that is the acceleration * + a correction of the fourth order. So this particular algorithm so this is our main result here, the new position is given in terms of 2 times the current position - the

old position + acceleration * delta t square okay, + a correction term. So now just I got the new position, I can also get velocity now.

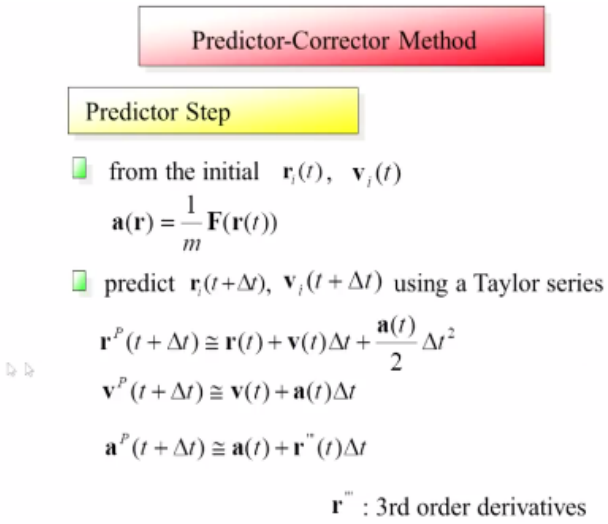
How do I get velocity, I subtract 2 from 1, when I subtract new position – the previous position will be 2 times this should be again velocity * delta t + correction so the velocity now will be r at t + delta t – r at t – delta t/2 delta t. It is really like a change in position in 2 steps divided by the time for 2 steps so that is my velocity. So only problem of this is that I can get the new position in terms of the old position.

But I cannot get the new velocity, what I got really is the current velocity. Because this velocity is now at the current position and not the new position. So that is the problem, but when we need only forces suppose I need forces, in terms of that force I have the new position if I do not need velocity for any specific purpose this is an okay algorithm. Now there is also a refined algorithm for this it is called the velocity Verlet algorithm.

I urge you to look it up in the websites, so that is more symmetric with respect to position of velocities. This is not symmetric between positional velocities because new position is predicted in terms of old positions okay whereas there is no formula for new velocity. So that is the lacuna here, but it does not matter because many times we just want new position as a function of time.

I also get new velocity when I reach a new position. For the new velocity I can always get after I have gone to the new position. I can get this in the next step. So in principle I have velocities at each step and positions at each step okay.

(Refer Slide Time: 12:30)



So now just as I described the Verlet algorithm there is another algorithm called the predictor-corrector so this is a slightly improved algorithm okay, so there are many algorithm there. So I will just discuss 2 so that you know that there is no unique algorithm here and what was good thing about Verlet algorithm the corrections to positions, look at here. The corrections position to position is the fourth order.

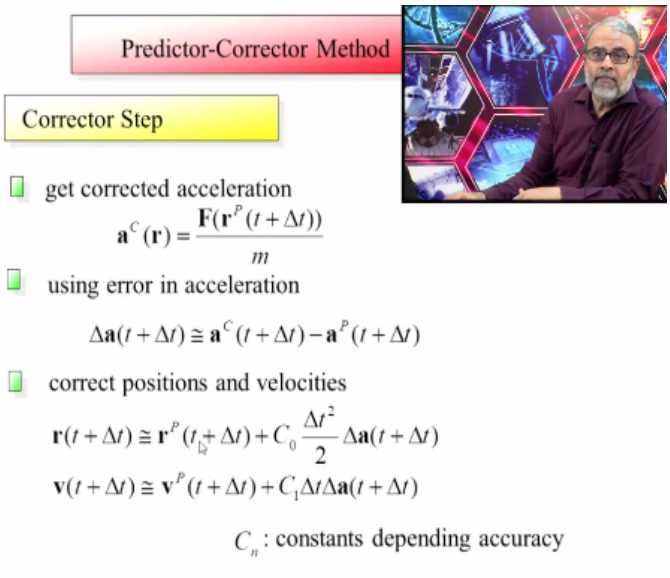
So if your delta t was 0.1, delta t to the 4 will be 0.0001. So the corrections are to the fourth order. There is no correction to the third order. Whereas for velocity there is correction to the second order. So now coming to the predictor-corrector. I have initial position, initial velocity, I calculate the initial acceleration. How do I calculate the initial acceleration? calculate the force, divided by mass that is my initial acceleration.

So once I know the initial acceleration I can predict new position and new velocity using a Taylor series okay. Here there are no correction in the slide in the previous slide I did not write velocity in the acceleration properly this is better here. Predicted position at t + delta t current positon + velocity * delta t + acceleration/t * delta t square. So this is my predicted position. This is my predicted velocity at t + delta t is the new velocity, current velocity + acceleration * delta t.

What is the predicted acceleration, present acceleration + third derivative * delta t. So to get the new acceleration I need the third derivative of position with respect to time. So velocity is the first derivative, acceleration is the second derivative, the third derivative of r normally we

do not use it so we do not come across. So here to know the new acceleration I need the third derivative of r with respect to t. So these are my predicted position.

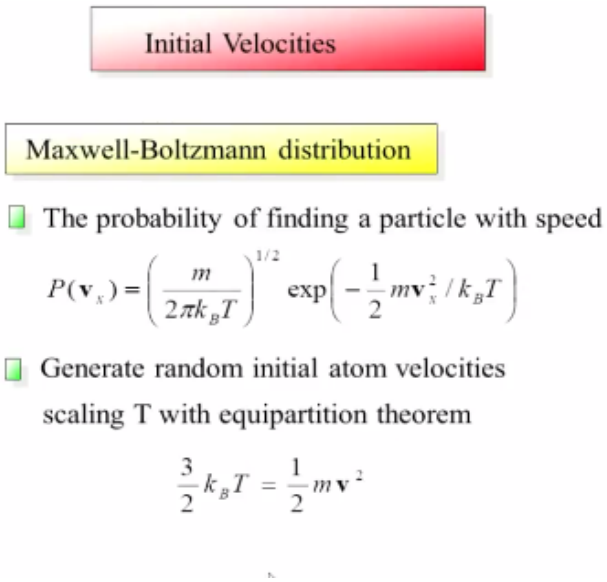
(Refer Slide Time: 14:31)



How do I correct them? to correct them I calculate a correction. So corrected acceleration is force at the new predicted position/mass okay, this is my correction. So what is the change in the acceleration now. Corrected acceleration - the predicted acceleration okay. So this is my correction. So how do I correct positions now. The corrected position is predicted position + a constant times delta t square/2 * correction in acceleration.

Corrected velocity will be predicted velocity + a constant times delta t * some correction in acceleration. So these constants will now determine the accuracy. So there are ways to determine these constants so one can do the predicted corrected algorithm okay.

(Refer Slide Time: 15:19)

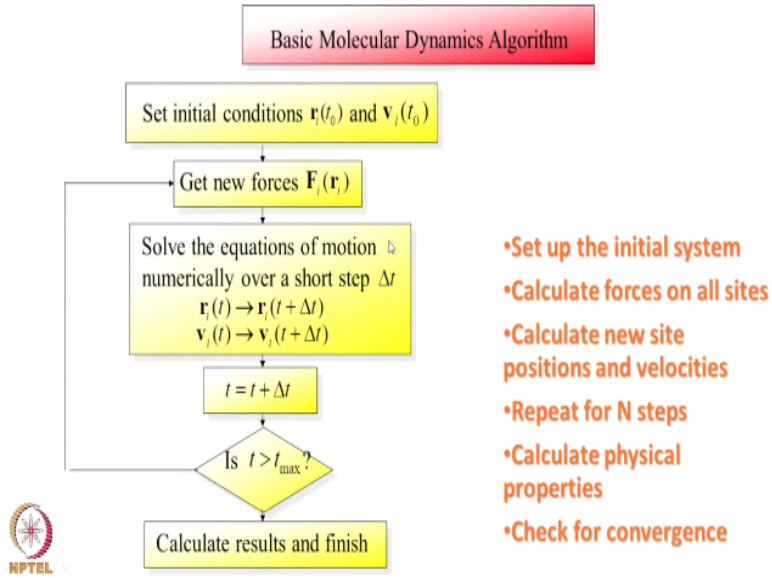


So now so far we have told you how to do the dynamics, but when I am starting my simulation that is initial step how will I know the velocities okay. So I do not know my velocities when I start my simulations. So one way to get those velocities is through a Maxwell-Boltzmann distribution. So remember we studied our random numbers one of the greatest application of random numbers is to calculate these velocities.

How do I determine these velocities? you know that they are distributed according to a Maxwell-Boltzmann distribution. So I determine the velocities in this manner using Gaussian random numbers and once I obtain these velocities I determine the temperature to those velocities. How do I determine the temperature? you know that when I take the average of kinetic energy.

So the average kinetic energy is nothing but 3/2 kT, so that is the equipartition theorem. So I determine a set of velocities. I determine the average kinetic energy and equate it to 3/2 kT. So if the average kinetic energy is not equal to 3/2 if it is less I increase all the velocities such that I get the average temperature. So when you do a simulation you want to do the simulation at a given temperature and you can obtain a gaussian distribution at that temperature and assign all the initial velocities, okay.

(Refer Slide Time: 16:45)



So now let us come to the basic algorithm of this molecular dynamics. So we know now the equations of motion, okay, we know the algorithm, so now we are in a good position to study the structure of this program. So how does this program work now? Set initial condition that is I determine all the initial positions and all the velocities though these velocities could be though the Maxwell-Boltzmann distribution or if you have already done the simulation earlier.

So at the end of the simulation you will have some velocity. So I can use the last point of the previous simulation as the starting point for the new simulation. We will do this through a practical session as well. So set up the initial conditions, calculate the new forces. So once you calculate the new forces solve the equations of motion okay we have chosen a time step, once you have chosen time steps you have new position and new velocity after a time step delta t.

So once you do that you increment the time again. So I determine at a new time step. So now I want to increment time further. So how long will I increment the time, suppose I want to do a simulation for 100 picoseconds. So if my time, time has been incremented because I did simulation for delta t. So now t has become incremented by delta t, so if this time is greater than the maximum time that I want to simulate such as 100 nanosecond.

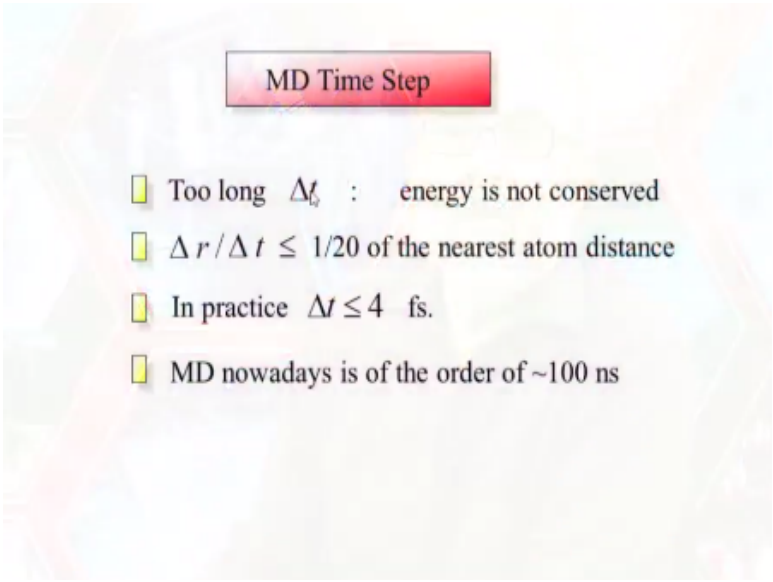
If it is greater than I stop, so if it is less I go back calculate new forces, calculate new velocities, new positions and keep on repeating the simulation okay. So once I reach a time step which is the maximum time I conclude my simulation. So that is the same thing written

on the right side, set up the initial system, calculate the forces on all sides, calculate new site positions and velocities.

Repeat for N steps, this N will be determined by the maximum time t max, calculate the physical properties and the other thing is you have to check for convergence. So we will have more to discuss on convergence because one of the very important ideas in simulation is that all the results you do should converge to a given value because if they do not converge there is no meaning in this simulation.

So I shall discuss that as we go along. So the next thing I want to comment what time step do I use for molecular dynamics okay.

(Refer Slide Time: 19:18)

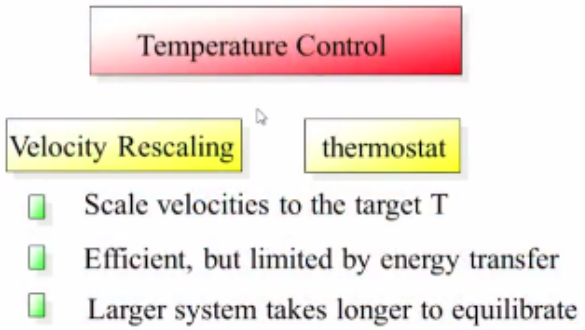


So if I use a very large time step okay then energy is not conserved there will be problem because particle will start moving too fast, they approach other particles then there will be lot of problems with conserving energy okay. So standard practise will be that delta r/delta t should be 1/20th of the nearest atom distance. So there are distances between atoms so typically they are 3 angstroms.

So velocity should be typically 1/delta r/delta t should if it is typically less than 1/20 in magnitude then that is a good time step. So typically in practise delta t of less than 4 femtosecond is very good. Femtosecond is 10-15 seconds. So ideally one uses around 0.1 femtosecond or 2 femtosecond these are very good time steps for complex molecules. For very simple molecules you may even use up to 5 femtoseconds.

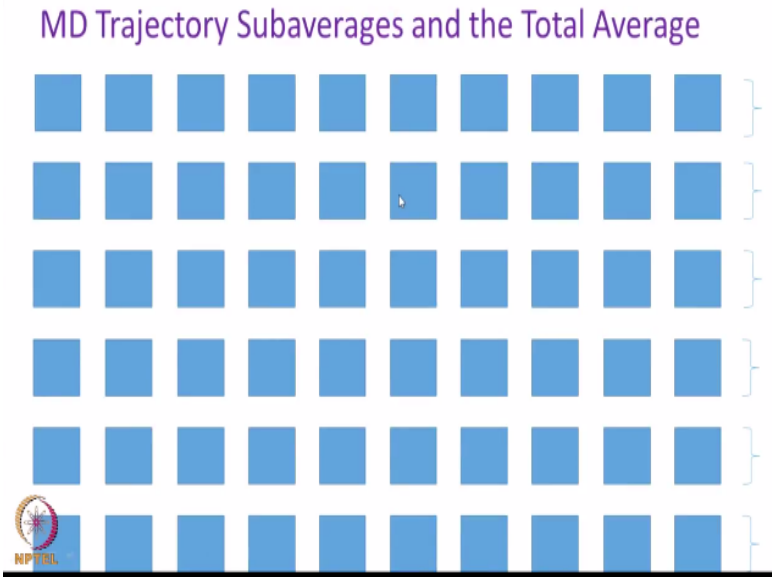
So this is the time step and nowadays because we have fairly good and fast computers, molecular dynamics is done for about 100 nanosecond. Remember I had a t max in the last slide, typically one does simulation for 100 nanoseconds to get convergence okay. So now when we do simulations another thing that is important is temperature control okay.

(Refer Slide Time: 20:40)



Usually this is not necessary but it is always a good idea to rescale velocities to the target velocity every now and then okay. So this when you do rescaling it makes the system a lot stable okay. Ideally in molecular dynamics one should not do rescaling, but if you do not do rescaling many times there will be a problem so if you do rescaling once in a few steps you get a better simulation result.

(Refer Slide Time: 21:08)



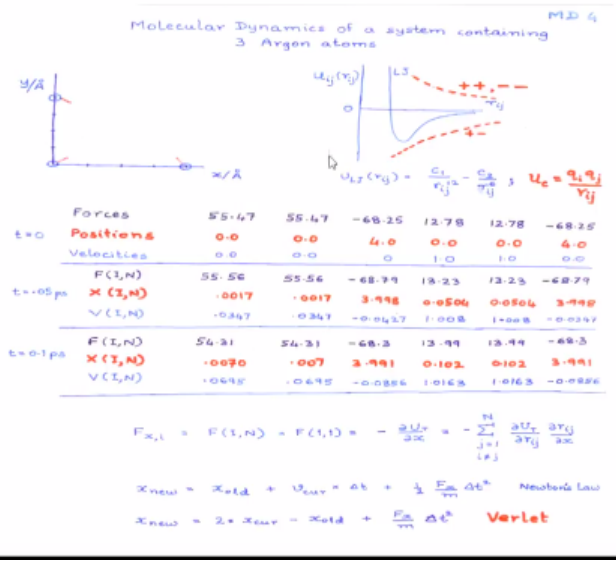
So now the next thing I want to do now we now discussed that in simulations you will be determining a trajectory. What is the meaning of trajectory step 1, step 2, step 3, step 4, all the way up to let us say 100 nanosecond. There will be 1000s and 1000s of steps. Let us typically, there will be 100,000 steps. So when I average over all the 100,000 steps I am going to get only one average and when I get this one average I do not know whether it is a good average or not.

So one way to solve the problem would be suppose I am doing 1 lakh steps, so I take one set to be 10,000 steps, one block, next block will be next 10,000 steps. So this way I generate 10 blocks of 10,000 steps each, so which means for each block I have a sub average okay and the total average will be average of these sub averages. Now if I know that this is average over 10,000 steps one sub average, next 10,000 step another sub average.

If all these sub averages are very near each other then I know that it is a converge systems okay. So that means my sub average does not change from one subset to another subset. So that is one way to confirm that there is convergence. So if there is no convergence the first sub average maybe x, the next sub average maybe 1.1x, the third one will be 1.3x. So if the sub averages keep on changing then you know that you have not converged.

So a very good way to check whether there is convergence is to calculate sub averages then the average overall the sub averages, okay.

(Refer Slide Time: 22:51)



So now this is a typical example of a molecular dynamics. This is just to give you a feel now what I have done here. I have taken a 3 particle system. So, so far we have only talked of concepts now let us take real example. So in this there are 3 particles one at the center, one along the y axis, one along the x axis. So I will calculate so this is my initial position. At initial position the first position is 00, second position is 40 this is my second particle.

Third particle is 0 and 4. I calculate the forces on each of these particles using this formula. I calculate the forces using this formula of course these are Lennard-Jones particles so there are no charges. So I am just going to calculate the forces now. So you will see that I have printed the forces now. On the first particle it is 55 55, second particle - 55 12, third one 12 and – that is on the first particle both the x and y components are in this direction okay.

Whereas on the second particle the force on the x component is negative and the y component is positive. Whereas for the third particle the x component is positive and the y component is negative see this arrow. This is the direction of the force. So I calculate the force use the Verlet algorithm and calculate new position. So I have calculated new positions and new velocity.

At the new position and velocity, I calculated the fourth again and I calculate again new position and velocity, okay, then again calculate the forces, new position and velocities. So this really illustrates the use of the Verlet algorithm numerically and an important point I want to make here what are the units we are using to use in this particular simulation program. Usually we use time in picoseconds, distance in angstroms, these are standard practise.

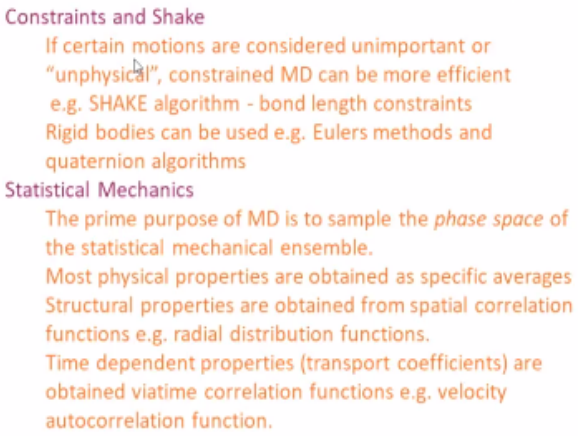
So if you use time in picoseconds and distance in angstroms the force, the units of the force will be 10 joule per mole/angstrom. So I repeat again, if the distance is in angstroms and the time is in picosecond energy will come into 10 joules per mole, that is the unit of energy. So if the energy is in 10 joules per mole, force will be energy per length, okay. So my force will be 10 joule per mole per angstrom. So that is my unit of force.

So it is extremely important to know all the units in your calculation because if you use a wrong unit there is no meaning in your result okay. So this is how I have illustrated here a typical example of a molecular dynamics of a 3 particle system. I illustrated 3 steps, you can

actually calculate this by hand okay. So make sure you understand everything okay. I have given you Verlet algorithm okay.

And I have also calculated velocity using the Verlet algorithm. So now before I conclude I want to talk about this constraints, what is the meaning of constraints. Suppose I have a molecule like water.

(Refer Slide Time: 26:07)



As it keeps on moving the bond lengths and bond angle should not change. Because if the bond lengths and bond angle change the water will become something like a linear water or a right angled water, that is not a realistic water. All water molecule should have the same bond lengths and same bond angles. So for that I use an algorithm called a SHAKE algorithm that is to keep the bond length constraints fixed.

I want to fix the bond lengths and bond angles. So I will describe in the next class how to use the SHAKE constraints and basically what we have done this simulation constitutes an example of a statistical mechanics. So what is the main purpose of molecular dynamics. The main purpose is to generate a trajectory. So what is a trajectory, position and momentum as a function of time that is called a phase space in statistical mechanics, okay.

And once you generate a trajectory then you can calculate physical properties as specific averages okay. In particular, structural properties can be obtained from spacial correlation functions that is the radial distribution function and the time dependent properties these are

the transport coefficients they can be obtained via time correlation function that is the velocity autocorrelation function.

So what I shall do in the next class I will define the radial distribution function as well as the time correlation function. I will describe algorithms to calculate it and once we describe the algorithm to calculate it we will also use them to calculate average quantities of lot of interesting properties. So we have many interesting properties in liquids and all of them can be obtained as some integrals over either your radial distribution functions or time correlation functions.

So in the next class we will define both these and we will give the formulae for calculating all the properties and then give an example of a simulation program. So once I give an example of a simulation program so that will complete our discussions on argon then we will discuss GROMACS and once we discuss GROMACS that will be towards the end of the course. So you should be able to do MD simulations not only for simple system, but also rather complicated systems using GROMACS.

So I will conclude my lecture here, in the next lecture there will be radial distribution functions and velocity autocorrelation functions. Thank you.