Basic Statistical Mechanics

Prof. Biman Bagchi
Department of Chemistry
Indian Institute of Technology, Bombay
Indian Institute of Science, Bangalore

Lecture - 60
Computer Simulation Methods in Statistical Mechanics Part 2

Welcome back to the course on statistical mechanics. We shall continue with our study of computer simulations as a primary method to implement the principles of statistical mechanics. In the part 1 which was in the last lecture. We discussed not just the importance of computer simulations in statistical mechanics, but how this allows as discussed highly complicated systems including biological systems like proteins and DNA and other complexes.

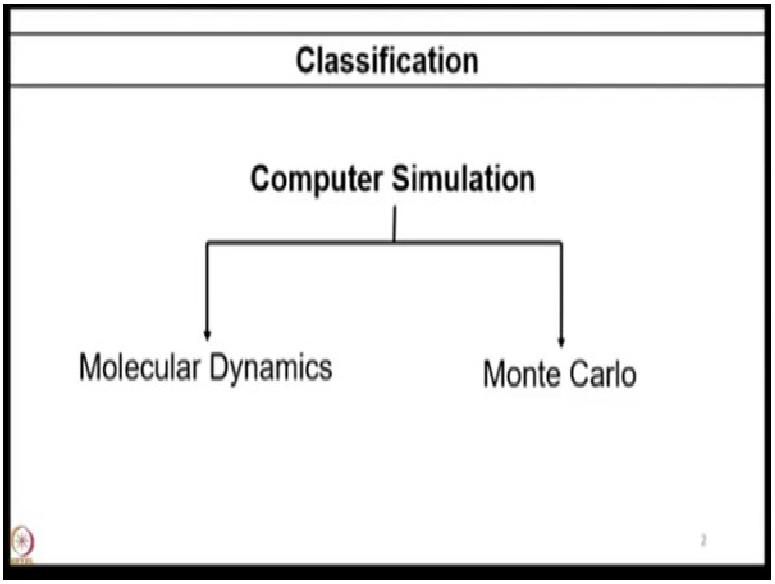
And we mentioned there that, there statistical mechanics has two major techniques; molecular dynamics simulation other is Monte Carlo. They are entirely different approaches and tied to our two postulates of statistical mechanics. One invokes the time averaging that is molecular dynamic simulation the other invokes the ensemble averaging which is the Monte Carlo simulations. In this sense they are very nicely tied together.

And again the same thing as Ergodic hypothesis also comes here because sampling of both the two will determine the outcome at the accuracy of the result. In this class, we will continue with that. I mentioned in the part I that this is the vast course, that many books on computer simulations and there is also a large number of lectures included.

I strongly recommend students read some of the books. In my book, you go to the chapter 29 (on basics of computer simulation) and chapter 30 is more advanced topic of computer simulation. And we have done it somewhat differently from other people but this is really tied very well with the course. I have already explained how we use Ising model explained the necessity and the implementation of periodic boundary condition.

And we use Ising model also the technique of minimum image convention and that we use to count the interactions and minimize load on the computer.

(Refer Slide Time 3:01)



Classification

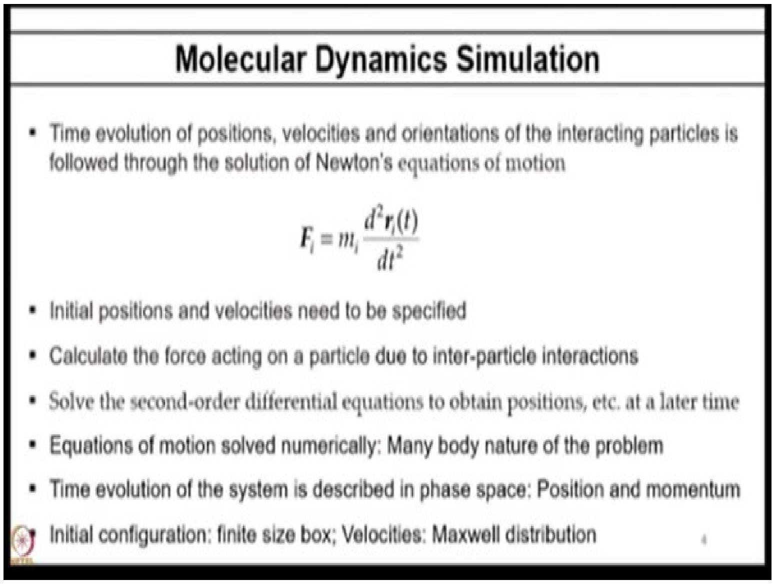
Computer Simulation

Molecular Dynamics

Monte Carlo

Today we update you again is same classification, molecular dynamics and Monte Carlo but today we will go detail now into the methods.

(Refer Slide Time 3:13)



Molecular Dynamics Simulation

- Time evolution of positions, velocities and orientations of the interacting particles is followed through the solution of Newton's equations of motion

$$F_i = m_i \frac{d^2 r_i(t)}{dt^2}$$

- Initial positions and velocities need to be specified
- Calculate the force acting on a particle due to inter-particle interactions
- Solve the second-order differential equations to obtain positions, etc. at a later time
- Equations of motion solved numerically: Many body nature of the problem
- Time evolution of the system is described in phase space: Position and momentum
- Initial configuration: finite size box; Velocities: Maxwell distribution

Now, as I told you is the time averaging, we have to generate temporal trajectory. The way we generate the temporal trajectory is by solving Newton's equation in classical mechanics and the Newton's solution equation force and mass time acceleration so the method is just solving numerically propagating a second order differential equation you go to each particle finds it is position, velocities it is orientation if it is like water.

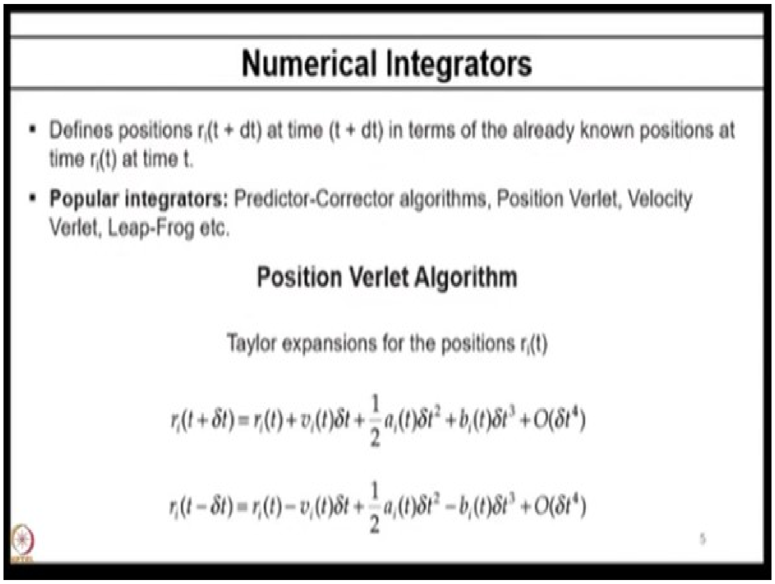
Then you find the force and torque on the molecule and then from time t to next time $t + \delta t$ you propagate it by integrating. This is called the method of quadrature which is solved numerically and there are over the 100 of years many techniques have been developed and all of them came handy in solving because it is very important that we propagate the equation for a very long time.

As I told you we propagated what may be 100 nanoseconds sometimes in microsecond, however, individual time steps really many of you have done a finite difference method of solving the differential equations. The difference here is that a) that you have a large number of degrees of freedom like we have even if you have 1000 argon atoms then we have this 2000 position coordinate and 3000 momentum coordinates and these forces change very quickly in a liquid.

So the time step that we have to use it is a bit small and time steps have been in femtoseconds. So now you see the enormity of the problem even less than femtoseconds. Let us say that we take half a femtosecond. Now, I have to propagate this equation with the 6N coupled degrees of freedom by using a time scale of half a femtosecond to microsecond.

So it is less than 10^{-15}, 10^{-6}. So I have to do something of the 10 orders of magnitude. So this is huge and that is the real difficult.

(Refer Slide Time 5:47)



Numerical Integrators

- Defines positions $r_i(t + dt)$ at time $(t + dt)$ in terms of the already known positions at time $r_i(t)$ at time t.
- **Popular integrators:** Predictor-Corrector algorithms, Position Verlet, Velocity Verlet, Leap-Frog etc.

Position Verlet Algorithm

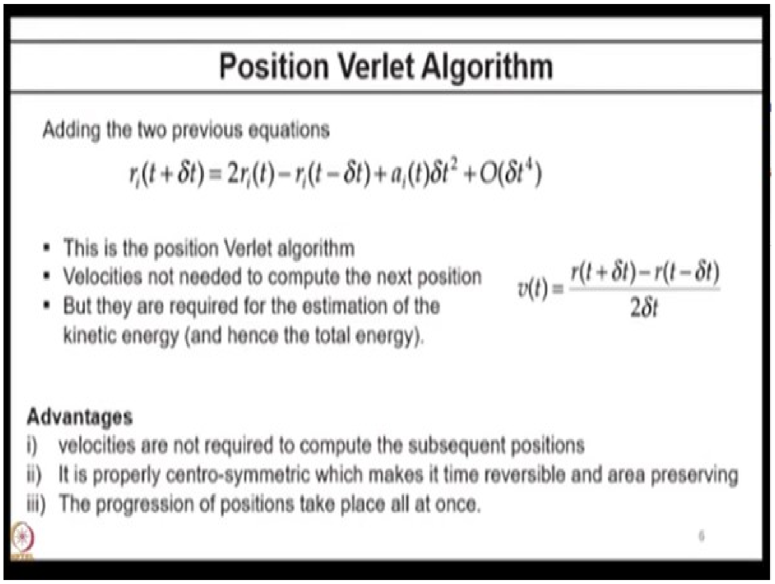
Taylor expansions for the positions $r_i(t)$

$$r_i(t+\delta t) = r_i(t) + v_i(t)\delta t + \frac{1}{2}a_i(t)\delta t^2 + b_i(t)\delta t^3 + O(\delta t^4)$$

$$r_i(t-\delta t) = r_i(t) - v_i(t)\delta t + \frac{1}{2}a_i(t)\delta t^2 - b_i(t)\delta t^3 + O(\delta t^4)$$

How do we integrate well you know as I say this is a finite difference method there is fine one of the popular one is a position Verlet algorithm where you both propagate. First you propagate the position $t + \delta t$ and you write this Taylor expansion when you write $t - \delta t$. The whole idea is to add them and get rid of the linear term.

(Refer Slide Time 6:17)



Position Verlet Algorithm

Adding the two previous equations

$$r_i(t+\delta t) = 2r_i(t) - r_i(t-\delta t) + a_i(t)\delta t^2 + O(\delta t^4)$$

- This is the position Verlet algorithm
- Velocities not needed to compute the next position
- But they are required for the estimation of the kinetic energy (and hence the total energy).

$$v(t) = \frac{r(t+\delta t) - r(t-\delta t)}{2\delta t}$$

Advantages

i) velocities are not required to compute the subsequent positions

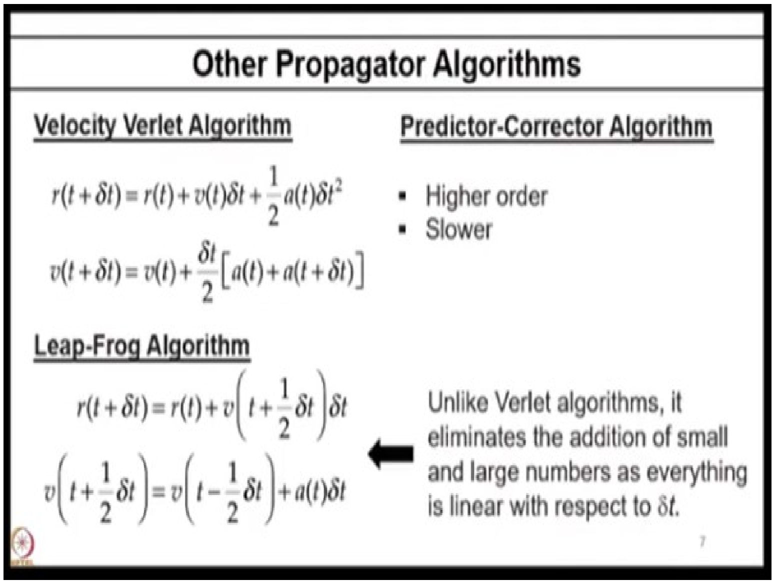
ii) It is properly centro-symmetric which makes it time reversible and area preserving

iii) The progression of positions take place all at once.

So I did $r_i(t+\delta t)$ and then $2r_i(t) - r_i(t-\delta t)$ just adding the 2. This is the position algorithm this is I also get rid of the first term linear term I get rid of also the velocity but I do have acceleration in $a_i(t)$. This is called positional Verlet algorithm very neat technique. In this technique velocity is obtained after you get the trajectory.

Since you know the positions at $t + \delta t$ and $t - \delta t$ you know the velocity as it is given on the right hand side of the equation and you have to store the velocity not only because you have to check the kinetic energy and you have to get many properties you need the velocity. So by this way I get the full phase space information the motion of the trajectory in the velocity but the advantage is that I do not have to store the velocity immediately and so I do not need it so that I has certain simplification.

(Refer Slide Time 7:31)



Other Propagator Algorithms

Velocity Verlet Algorithm

$$r(t+\delta t) = r(t) + v(t)\delta t + \frac{1}{2}a(t)\delta t^2$$

$$v(t+\delta t) = v(t) + \frac{\delta t}{2}\left[a(t) + a(t+\delta t)\right]$$

Predictor-Corrector Algorithm

- Higher order
- Slower

Leap-Frog Algorithm

$$r(t+\delta t) = r(t) + v\left(t+\frac{1}{2}\delta t\right)\delta t$$

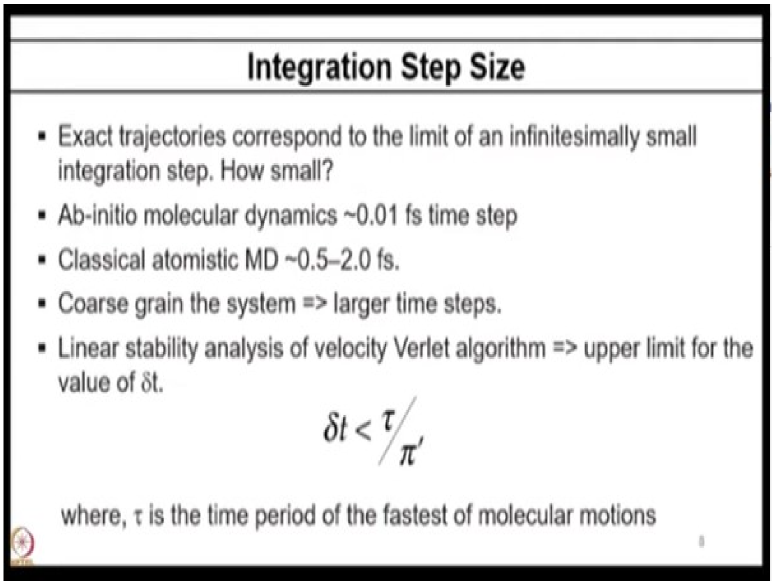
$$v\left(t+\frac{1}{2}\delta t\right) = v\left(t-\frac{1}{2}\delta t\right) + a(t)\delta t$$

Unlike Verlet algorithms, it eliminates the addition of small and large numbers as everything is linear with respect to δt.

Then there is this Velocity Verlet Algorithm. This is very clever way of doing thing again now you propagate both the position and the velocity. There is a cleverness in putting the velocity term. The whole idea is essentially to minimize the error of propagation. In the Leap-Frog algorithm another interesting thing calculation of the velocity is the tricky thing.

So you use $v\left(t+\frac{1}{2}\delta t\right)$ this in terms of $v\left(t-\frac{1}{2}\delta t\right)$. So it is like velocity and position staggered by $\frac{1}{2}\delta t$. And that has the advantage that the by doing that your accuracy increases and if I remember correctly the $(\delta t)^2$ term is eliminated. So everything is linear in δt up to means $(\delta t)^2$ is eliminated and that has a numerical accuracy. At one time when computers was not that powerful these techniques were exceedingly important because success of your simulation dependent on what technique we use.

(Refer Slide Time: 9:11)



Integration Step Size

- Exact trajectories correspond to the limit of an infinitesimally small integration step. How small?
- Ab-initio molecular dynamics ~0.01 fs time step
- Classical atomistic MD ~0.5–2.0 fs.
- Coarse grain the system => larger time steps.
- Linear stability analysis of velocity Verlet algorithm => upper limit for the value of δt.

$$\delta t < \tau / \pi'$$

where, τ is the time period of the fastest of molecular motions

So then an integration step size, I said the classical molecular dynamic simulation is half femtosecond to 2 femtosecond. Ab-initio molecular dynamics simulation, where you solve time dependence the time is because of quantum degrees of freedom, because of quantum coherence, is exceedingly small time steps. So these simulations cannot be done easily remove the ratio even now, if you cause again, few tens of a femtosecond.

Many times when we talk of very big systems we do coarse grain the system. Coarse grain is essentially what you do? You kind of have lens, you make the lens opaque that means you now do not want the information which are very detailed. For example, we did Landau's theory of order parameter and you wrote down the free energy in terms of a parameter expansion that is an example of coarse-grained description.

That means we are not going fully to position and momentum details but we kind of setting the grid to the system. Let us say Q be grid to the 3 dimensional system and now we have average over the grid and with averaging, we find out say, for example, order parameters. Now we derive an equation of motion for the order parameter and then we propagate that order parameter.

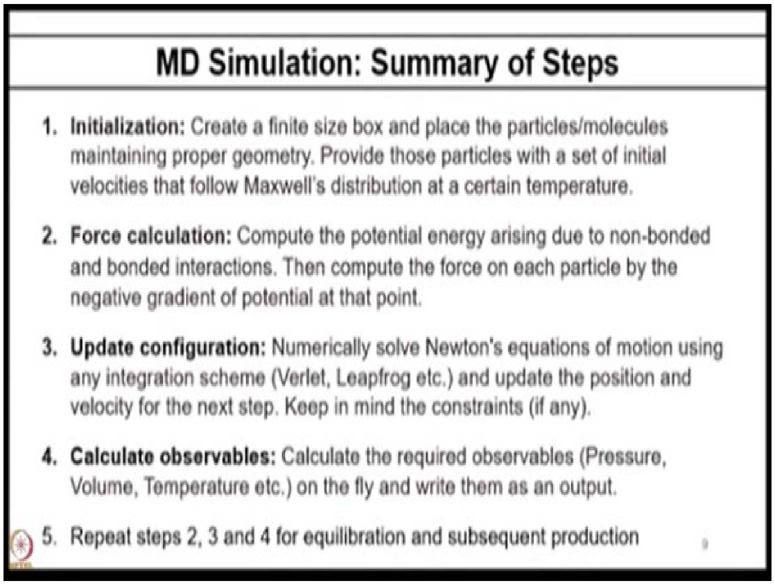
So, that also has a great merit and it is used in systems where we do not want the atomistic details many times. Sometimes these days, coarse grain system is also done when you eliminate, say about 4 or 5 degrees of freedom like I want to do a polymer and then polymer I have, the

CH3- CH2- CH2- CH2-CH2 then I can coarse grain it by first CH2 is one carbon atom. So I eliminate hydrogen then I can then take the for example, in an protein the amino acid side chains become a sphere.

We are then, of course, you have to place somewhere, so when you develop a coarse grain description, finding the force field or intermolecular potential undergoes lot of change. Because, no longer you have the interaction in terms of pair atoms, so you have to also develop the force field, so that has a lot of effort goes into that. However, coming back, so these are the kind of things that we do the different integration, different quadrature technique, different propagation techniques by this kind of short time step we generate a trajectory.

Once we generate the trajectory these statistical mechanics tells us what to do with it because you know the time everything you can calculate the pressure. Pressure by calculating the specific heat so we get by energy fluctuation many other properties radial distribution function all these things we can calculate and then our job is done.

(Refer Slide Time: 12:10)



MD Simulation: Summary of Steps

1. **Initialization:** Create a finite size box and place the particles/molecules maintaining proper geometry. Provide those particles with a set of initial velocities that follow Maxwell's distribution at a certain temperature.
2. **Force calculation:** Compute the potential energy arising due to non-bonded and bonded interactions. Then compute the force on each particle by the negative gradient of potential at that point.
3. **Update configuration:** Numerically solve Newton's equations of motion using any integration scheme (Verlet, Leapfrog etc.) and update the position and velocity for the next step. Keep in mind the constraints (if any).
4. **Calculate observables:** Calculate the required observables (Pressure, Volume, Temperature etc.) on the fly and write them as an output.
5. Repeat steps 2, 3 and 4 for equilibration and subsequent production

So, the summary of these steps, first you create a finite size box since that is what we discussed in the last lecture and then I give him but the example of the ice, calculate the force on each atom and molecule and then you go through each atom and molecule and depending on the force is experiencing and the Newton’s equation.

And the some quadrature like Verlet algorithm new move in position momentum of each molecule this is called the updating configuration then you have the trajectory in calculation.

(Refer Slide Time 12:45)

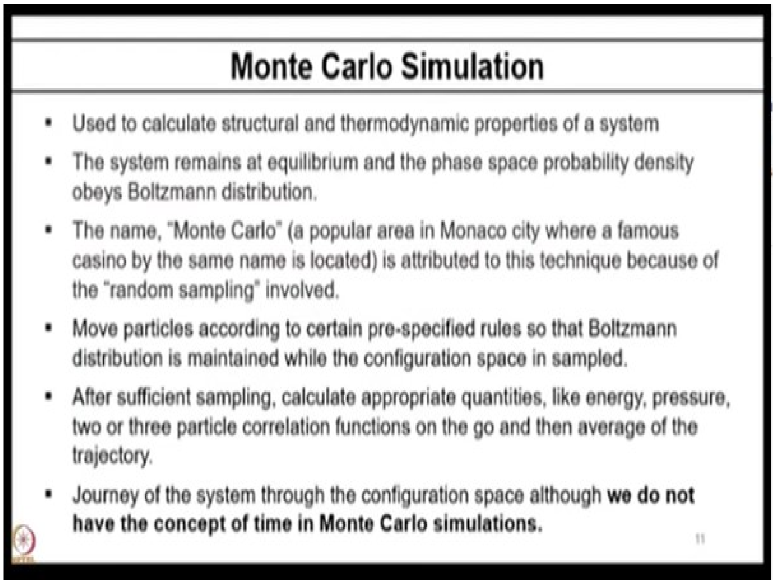
Monte Carlo Simulation

This is our Monte Carlo Simulation, which is straightforward and often the method of choice and most of the packages that exist in literature like gromacs, charms and all these they essentially they are all molecular dynamic solutions because molecular dynamic solution as I explained you these are straightforward in classical simulation solution of the Newton's equations of motion.

Then I can propagate it very easily; however, the next powerful technique and often the method of choice in complex systems if you do not want the dynamical properties is the Monte Carlo simulation. As I discussed last days Monte Carlo simulation placed in with a camping, it has to do with the lot of random number generator and trap it is based on ensemble average principle.

But, we have to relate the system, go to all different microscopy states remember; when we did the ensemble average is equal to time average basic idea is that in time trajectory has to go to all corners of the phase space and in ensemble I have to take sufficient number ensembles that all the microscopic states are captured all of number or fair amount of microscopy states, so both have their demand. So we now discussed the Monte Carlo simulation.

(Refer Slide Time 14:23)

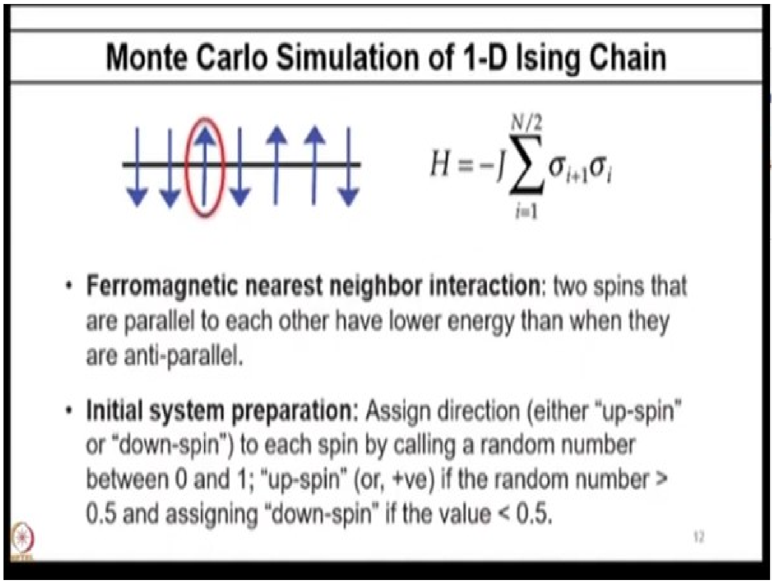


Monte Carlo Simulation

- Used to calculate structural and thermodynamic properties of a system
- The system remains at equilibrium and the phase space probability density obeys Boltzmann distribution.
- The name, "Monte Carlo" (a popular area in Monaco city where a famous casino by the same name is located) is attributed to this technique because of the "random sampling" involved.
- Move particles according to certain pre-specified rules so that Boltzmann distribution is maintained while the configuration space in sampled.
- After sufficient sampling, calculate appropriate quantities, like energy, pressure, two or three particle correlation functions on the go and then average of the trajectory.
- Journey of the system through the configuration space although **we do not have the concept of time in Monte Carlo simulations.**

Again, the Monte Carlo Simulation more common to calculate structural and thermodynamics properties of the system as you can read here it is situated here at a popular area in Monaco City. I thought from France but I think Monaco is situated by France anyway and it is an algorithm and Monte Carlo algorithm which was used in metropolis algorithm by this name was introduced throughout the theory. It is probability theory that guarantees many of these things.

(Refer Slide Time 15:04)



Monte Carlo Simulation of 1-D Ising Chain

$$H = -J\sum_{i=1}^{N/2}\sigma_{i+1}\sigma_i$$

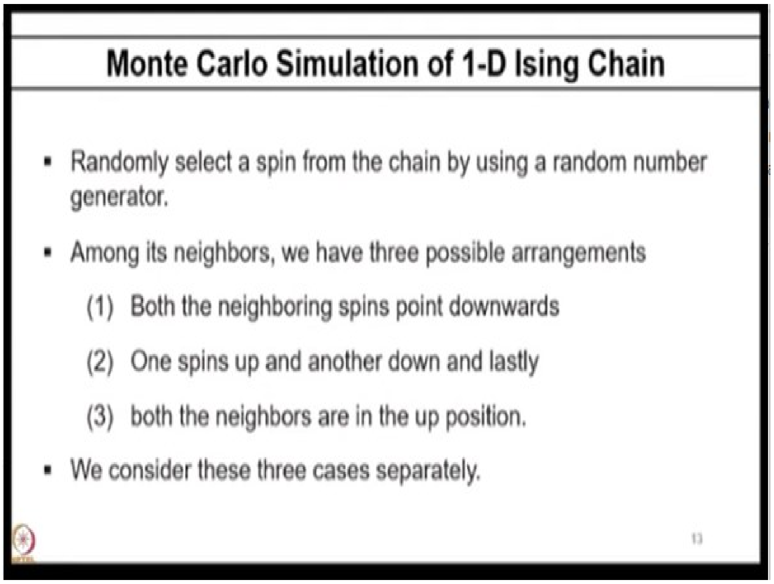
- **Ferromagnetic nearest neighbor interaction:** two spins that are parallel to each other have lower energy than when they are anti-parallel.
- **Initial system preparation:** Assign direction (either "up-spin" or "down-spin") to each spin by calling a random number between 0 and 1; "up-spin" (or, +ve) if the random number > 0.5 and assigning "down-spin" if the value < 0.5.

Now, as before I bring the things in terms of one dimensional Ising model, so I wrote down the one-dimensional Ising model with this Hamiltonian that we have done and if you have a ferromagnetic when two spins are parallel energy is less. Now, it is a wonderful thing to explain Monte Carlo simulation of all these.

So first they generate a initial configuration by go to each lattice side. I call it random number and that has value between 0 and 1 and if it is less than 0.5 that make it up and if greater than 0.5 then I make it down. So, I can arbitrarily create a configuration is the beauty of Ising model. Now, I wanted to Monte Carlo simulation from there I, of course, sample configurations. In one dimensional Ising model we have additional advantage of the analytical expression.

So whatever I get by computer simulation I can check it. There is a very good starting point and I recommend all of you to do it this simple calculation here that I have a point outlined to you.

(Refer Slide Time: 16:30)

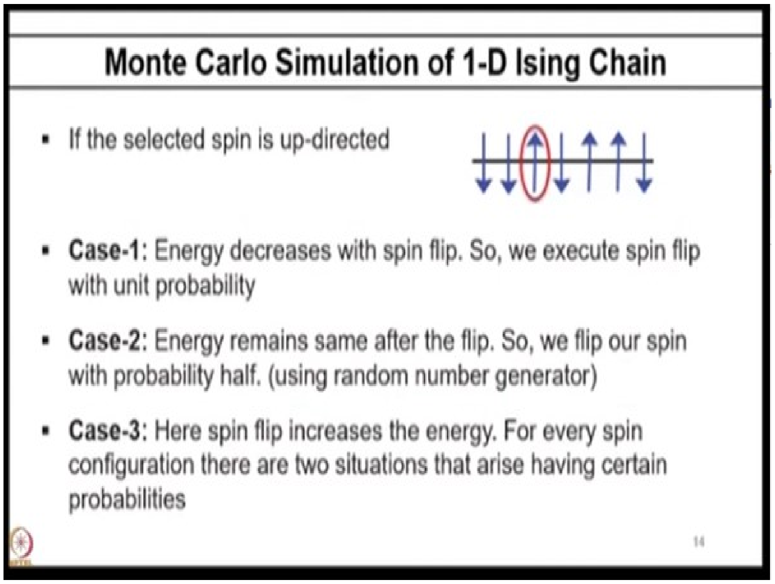


Monte Carlo Simulation of 1-D Ising Chain

- Randomly select a spin from the chain by using a random number generator.
- Among its neighbors, we have three possible arrangements
  (1) Both the neighboring spins point downwards
  (2) One spins up and another down and lastly
  (3) both the neighbors are in the up position.
- We consider these three cases separately.

Randomly selected a spin by using random number generator. Then you find that amongst neighbours we have three possible arrangements, with the neighbouring spins point downwards, then is an unfavourable configuration one spin up and among its neighbours and one spin down, putting up neighbours are in up position. Now,we consider the energy of these three cases are different.

(Refer Slide Time: 16:58)



Monte Carlo Simulation of 1-D Ising Chain

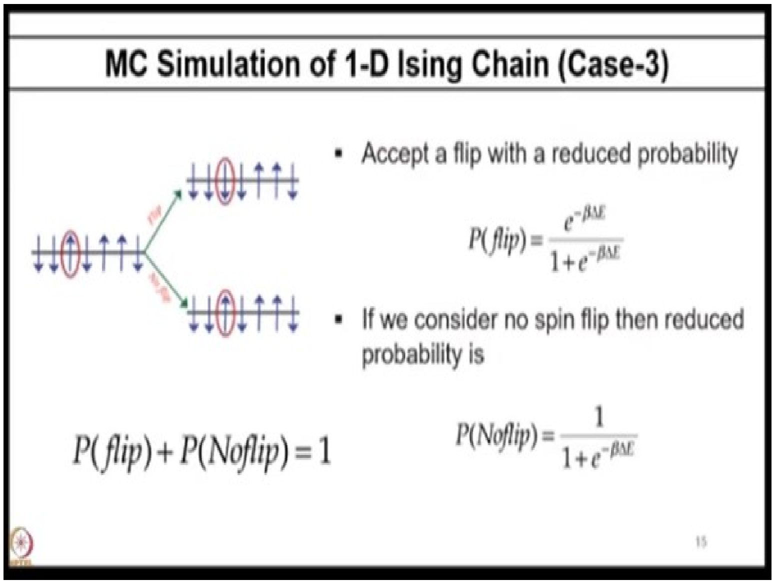
- If the selected spin is up-directed
- **Case-1:** Energy decreases with spin flip. So, we execute spin flip with unit probability
- **Case-2:** Energy remains same after the flip. So, we flip our spin with probability half. (using random number generator)
- **Case-3:** Here spin flip increases the energy. For every spin configuration there are two situations that arise having certain probabilities

This selects the up-directed, as we have done energy decreases, when you spin flips, by the way this is exactly the example which has been done in my book and I did it by all these things in terms of Ising model. So that it is easy for students to understand. In Monte-Carlo simulation is very important that you do a relative movement that means, you favour those moves those who lower the energy.

So you can easily know now that, if I am in ferromagnetic interaction, if the tagged spin, if I flip down then energy decreases and that is accepted with unique probability. Now, energy remains the same after the flip not in this case but if my one spins up and the spin is down that configuration this will happen and then I take that configuration the change configuration with half probability.

However, if increases then there are two situations and it is very interesting, if it increases then you accept but we accept with the penalty. By doing these choices you bias the system to low energy conformation. Why? Because this Boltzmann told us that these properties of the system are weighted by $e^{-\beta E}$.

(Refer Slide Time: 18:39)



MC Simulation of 1-D Ising Chain (Case-3)

- Accept a flip with a reduced probability

$$P(flip)=\frac{e^{-\beta\Delta E}}{1+e^{-\beta\Delta E}}$$

- If we consider no spin flip then reduced probability is

$$P(flip)+P(Noflip)=1$$

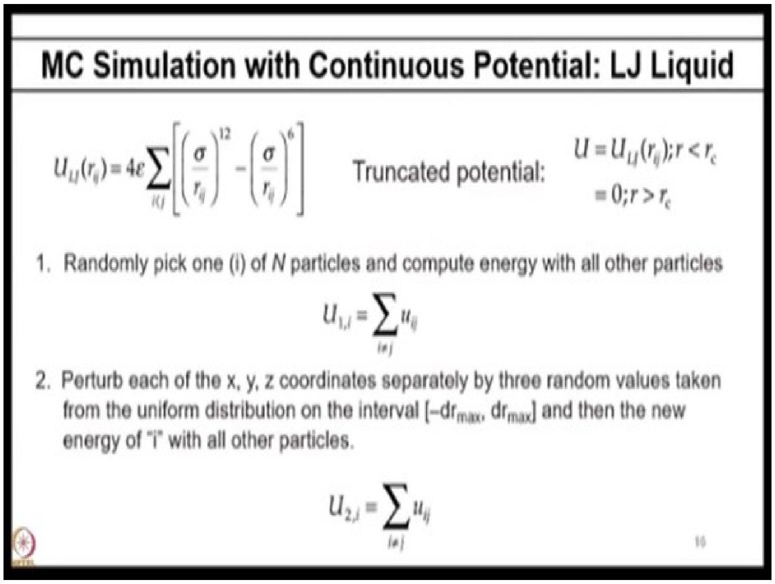
$$P(Noflip)=\frac{1}{1+e^{-\beta\Delta E}}$$

So here, I have told you, that if it have two condition it can flips, and if flips then it is accepted within probability. However, if it flips this picture is not 100% correct it should be it would be down and down going up like actually on the upper one. Then energy going up then we accept with the probability such that probability is conserved but with the penalty.

So flip probability that means from less spin down to going up that will be the change of energy is ΔE then $1+e^{-\beta\Delta E}$. Most people say no, I now open the option that I will have no spin flip then that will have a probability. Probability must be conserved, so if energy goes up then I had the option that I am going to stay in my same configuration.

So that is given by my bottom equation, so these two must be added to unity, as I show you in the left. Again to remind you there is mistake in the figure here, on the extreme should actually be down.

(Refer Slide Time: 20:34)



MC Simulation with Continuous Potential: LJ Liquid

$$U_{LJ}(r_{ij}) = 4\varepsilon\sum_{i<j}\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

Truncated potential: $U = U_{LJ}(r_{ij}); r < r_c$
$= 0; r > r_c$

1. Randomly pick one (i) of *N* particles and compute energy with all other particles

$$U_{1,i} = \sum_{i\neq j} u_{ij}$$

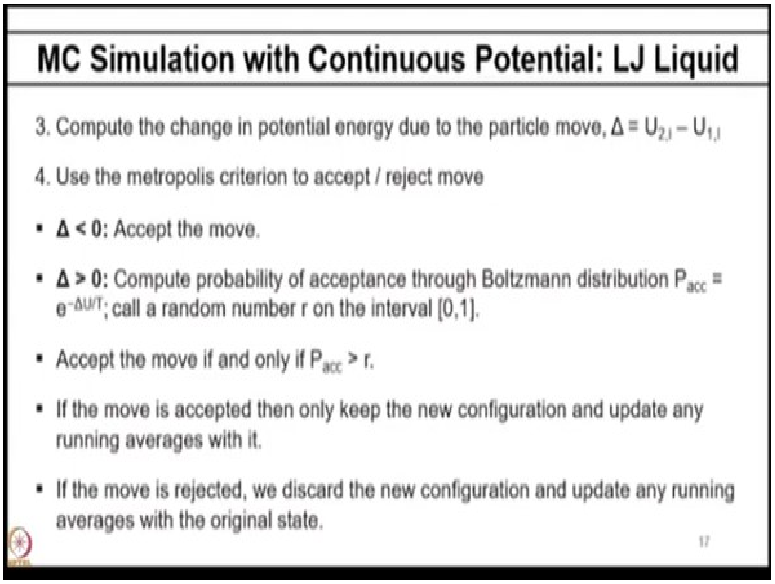
2. Perturb each of the x, y, z coordinates separately by three random values taken from the uniform distribution on the interval [$-dr_{max}$, dr_{max}] and then the new energy of "i" with all other particles.

$$U_{2,i} = \sum_{i\neq j} u_{ij}$$

Now this do what I described right now to you is like 1, spin discrete dynamics, but many times as you know Lennard-Jones water will have a potential that is continuous in that case what the way we do is very interesting. We take one of the N particles and compute all the interaction; you can easily guess what will happen compared with the interaction to all the other particles. Then, we now alter coordinates by random variables move three particles a little bit.

And then we calculate the energy, if the energy goes down, we accept it but if energy goes up we exactly do the same thing.

(Refer Slide Time: 21:23)



MC Simulation with Continuous Potential: LJ Liquid

3. Compute the change in potential energy due to the particle move, $\Delta = U_{2,i} - U_{1,i}$
4. Use the metropolis criterion to accept / reject move

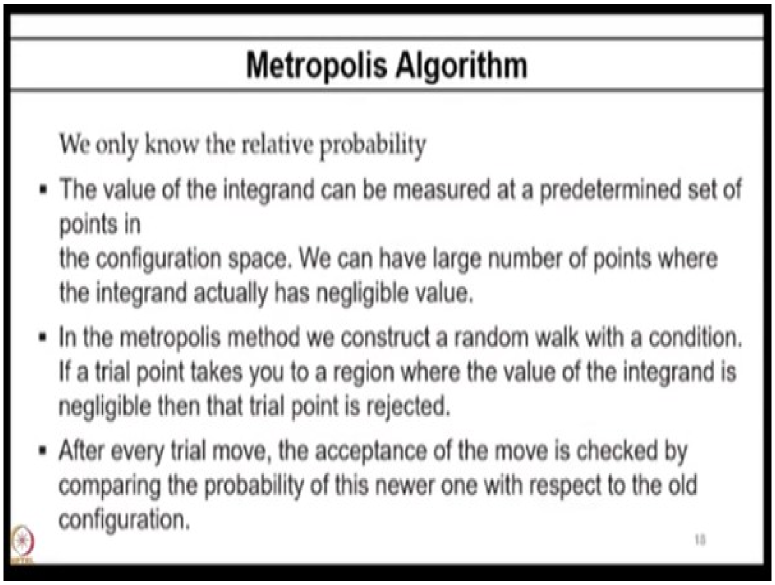
- **Δ < 0:** Accept the move.
- **Δ > 0:** Compute probability of acceptance through Boltzmann distribution $P_{acc} = e^{-\Delta U/T}$; call a random number r on the interval [0,1].
- Accept the move if and only if $P_{acc} > r$.
- If the move is accepted then only keep the new configuration and update any running averages with it.
- If the move is rejected, we discard the new configuration and update any running averages with the original state.

Again we do, if energy goes down then you accept to move it goes 0, then again we do the same thing except to move. Now one more thing we took mostly no flip in this case no flip means that we do not make these changes of my 3N position coordinates. But here you call it random number r and if my $e^{-\beta\Delta E}$ is greater than r then only I accept it.

That means when the energy increase is not very large why we do that? Because if we keep on neglecting all the moves then it becomes very demanding and we do not get. So many times we have to go up the hill then go down so but if I reject all the moves where energy is going up then that will be might some time my traping in a very small minima and also my statistics become poor.

So in this case even when energy goes up we do the Boltzmann factor and if Boltzmann factor is $e^{-\beta\Delta E}$ is not very large and positive.

(Refer Slide Time: 23:00)



Metropolis Algorithm

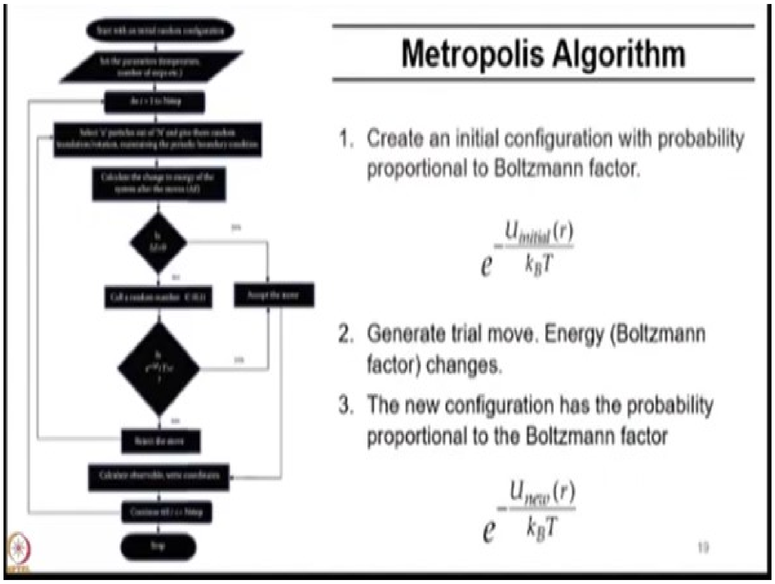
We only know the relative probability

- The value of the integrand can be measured at a predetermined set of points in
  the configuration space. We can have large number of points where the integrand actually has negligible value.
- In the metropolis method we construct a random walk with a condition. If a trial point takes you to a region where the value of the integrand is negligible then that trial point is rejected.
- After every trial move, the acceptance of the move is checked by comparing the probability of this newer one with respect to the old configuration.

So this is the technique that I just explained to you is essentially the metropolis algorithm. Now what we do need, in essentially the calculation of the partition function or calculation of average that is what statistical mechanics is we calculate the average is over Boltzmann factor. So in this case can be measured peredetermined set of points in the configuration space and then the metropolis method construct a random walk, where the value of the integrand is negligible. So what we do? That when energy goes up very high then we neglect that and one more move rather than Markov chain, then we continue. So it is as if we propagating a trajectory but the proper propagation of the trajectory is not always Newton's laws of motion it is following Boltzmann factor and calling a random number again and again.

So the random number again and again as I said it has to be a Markov chain i.e., there to be independent of each other and these random numbers will allow me to move every atom in its own three-dimensional space random directions and by random amount and then I calculate the energy.

(Refer Slide Time: 24:35)



Metropolis Algorithm

1. Create an initial configuration with probability proportional to Boltzmann factor.

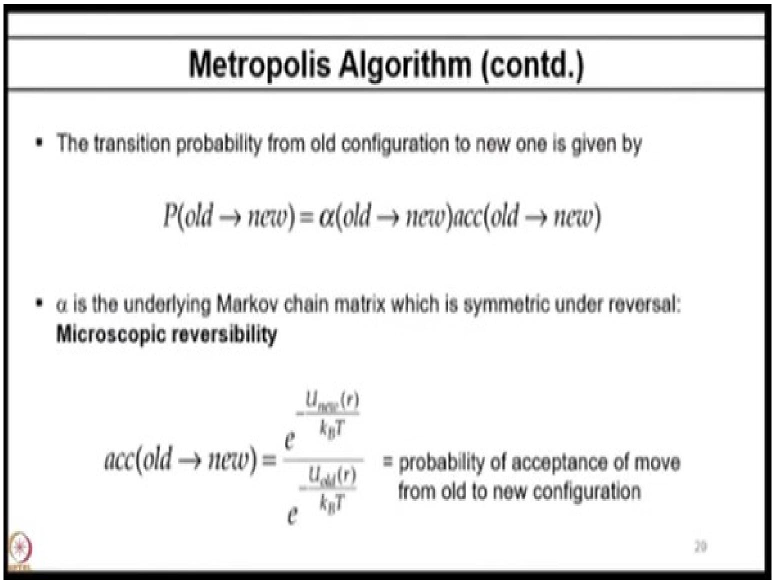
$$e^{-\frac{U_{initial}(r)}{k_B T}}$$

2. Generate trial move. Energy (Boltzmann factor) changes.
3. The new configuration has the probability proportional to the Boltzmann factor

$$e^{-\frac{U_{new}(r)}{k_B T}}$$

So this is the metropolis algorithm as I said it is completely controlled, this is what in flowchart I described little bit, small but you will be able to read young guys. So you generate the move calculate the energy except that energy. If energy goes up there except it little bit.

(Refer Slide Time: 24:56)



Metropolis Algorithm (contd.)

- The transition probability from old configuration to new one is given by

$$P(old \to new) = \alpha(old \to new)acc(old \to new)$$

- α is the underlying Markov chain matrix which is symmetric under reversal: **Microscopic reversibility**

$$acc(old \to new) = \frac{e^{-\frac{U_{new}(r)}{k_B T}}}{e^{-\frac{U_{old}(r)}{k_B T}}}$$ ≡ probability of acceptance of move from old to new configuration

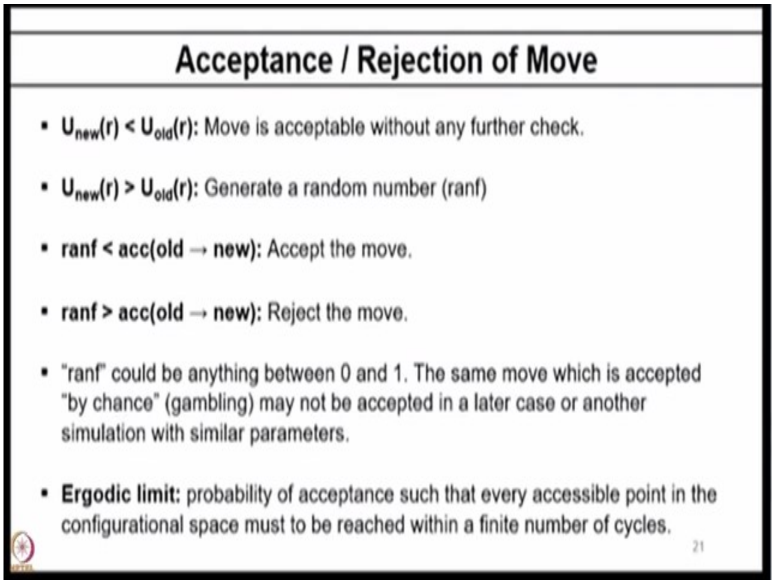
So this is goes on there little bit more detail because one thing that you want to implement is what is called microscopic reversibility. That means you are going to a place and you should be able to come back and since we are in equilibrium is extremely the essential concept that introduced by Onsager and in chemistry we already did in detail balance.

Detail balance is thought has that it is not microscopic reversibility is more detail and much more profound. Therefore, the microscope irreversibly means the probability of going to your place and the probability of in the initial position and probability of going to new position is equal to probability of the new position into probability back the rate of coming back. So these things is microscopy reversibility is essentially what we use in chemistry all the time K12, P1 K1 is P2 K21 and that going back and forth that has to be maintained.

So that is a very important quantity, so that has to be implemented in choosing this scheme and the in doing the transition probability from one state to other state because that you are in equilibrium because microscopy reversibility is the condition of equilibrium. So every move must obey microscopic reversibility, otherwise you can go for an equilibrium.

And Monte Carlo algorithm is something we study on the equilibrium system. Yes, we can start equilibrium system but that is required far more demanding that we are not going to discuss. What we are discussing here is grouping systems and where microscopic reversibility must be enforced. Many studies went wrong because microscopy reversibility both in analytical and numerical must not to obey.

(Refer Slide Time: 27:02)

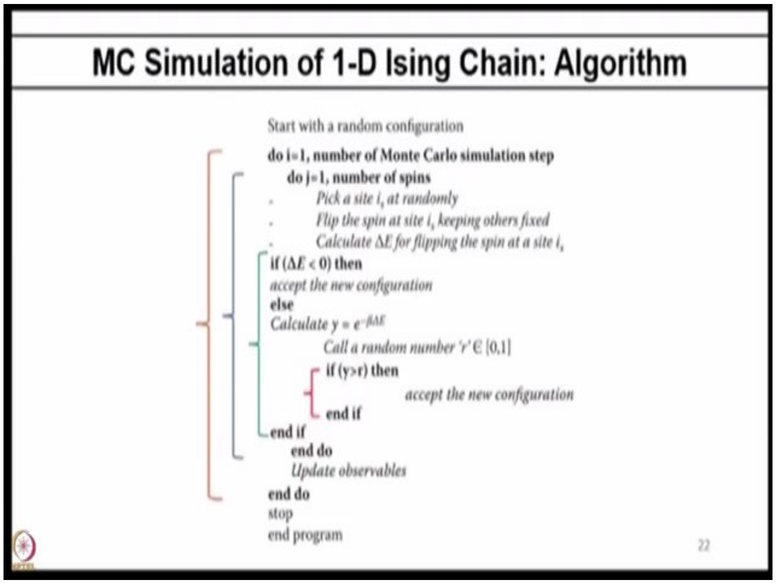


Acceptance / Rejection of Move

- **$U_{new}(r) < U_{old}(r)$:** Move is acceptable without any further check.
- **$U_{new}(r) > U_{old}(r)$:** Generate a random number (ranf)
- **ranf < acc(old → new):** Accept the move.
- **ranf > acc(old → new):** Reject the move.
- "ranf" could be anything between 0 and 1. The same move which is accepted "by chance" (gambling) may not be accepted in a later case or another simulation with similar parameters.
- **Ergodic limit:** probability of acceptance such that every accessible point in the configurational space must to be reached within a finite number of cycles.

So here is the acceptance rejection ratio very similar to what I need in case of Ising model. So I am not going into detail but two random number generator same thing I described a couple of times but one has to make sure that you are in equilibrium. And you are also increase the rate of the exploration of the configurational space. You do not want to do it so slow that you do not go, if you do not accept going up for example you are just not going anywhere much of the time.

(Refer Slide Time: 27:47)

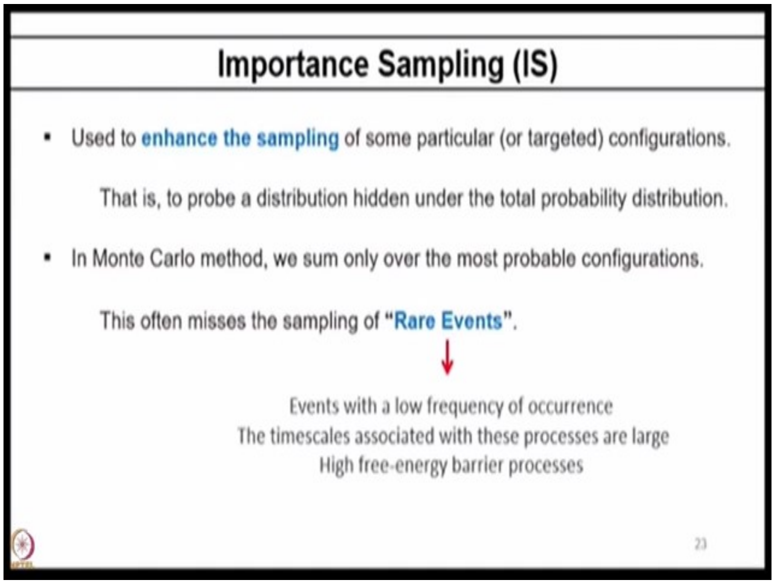


MC Simulation of 1-D Ising Chain: Algorithm

```
Start with a random configuration
do i=1, number of Monte Carlo simulation step
    do j=1, number of spins
.           Pick a site $i_s$ at randomly
.           Flip the spin at site $i_s$ keeping others fixed
.           Calculate $\Delta E$ for flipping the spin at a site $i_s$
if ($\Delta E < 0$) then
accept the new configuration
else
Calculate $y = e^{-\beta \Delta E}$
            Call a random number 'r' $\in [0,1]$
        if (y>r) then
                        accept the new configuration
        end if
end if
    end do
    Update observables
end do
stop
end program
```

We are again in flowchart, Monte Carlo simulation in 1 dimensional Ising model we did that.

(Refer Slide Time: 27:57)



Importance Sampling (IS)

- Used to enhance the sampling of some particular (or targeted) configurations.

  That is, to probe a distribution hidden under the total probability distribution.

- In Monte Carlo method, we sum only over the most probable configurations.

  This often misses the sampling of "Rare Events".

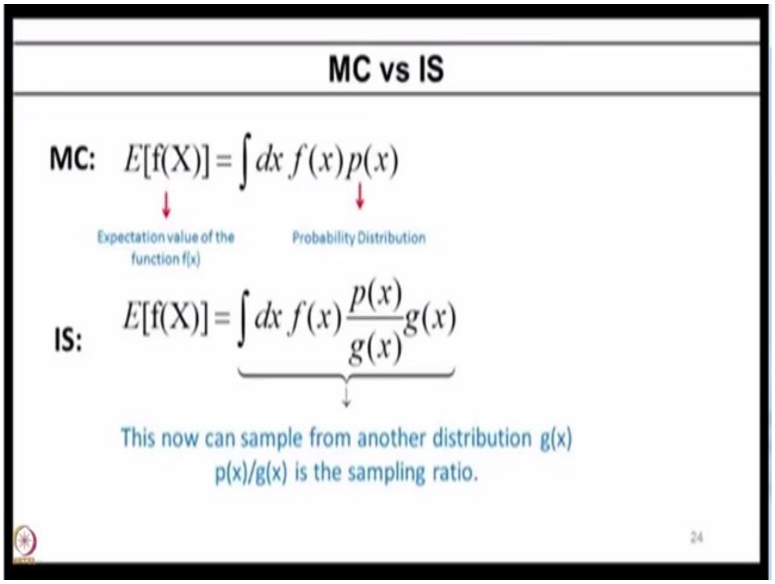
↓

Events with a low frequency of occurrence
The timescales associated with these processes are large
High free-energy barrier processes

There is an important thing that you often use is called important sampling. Important sampling is a name. I told you that problem of Monte Carlo Simulation, also a molecular dynamic simulation is that once, we system goes into a trap or kind of deeper minimum the system does not come out of that. This is a terrible problem in molecular dynamic simulation and many molecular dynamics simulation went wrong because the system that was simulated was stuck in minimum.

However, this minimum might be only a small part of the that particular minimum where the system is stuck because there many other domains which need to be explode to get another average and these the kind of things are done by important sampling. Most important is that is some of events like the chemical kinetics when the reactant the product are separated by a high free energy barrier.

(Refer Slide Time: 29:20)



MC vs IS

MC: $$E[\mathrm{f(X)}] = \int dx\, \underset{\downarrow}{f(x)}\underset{\downarrow}{p(x)}$$

Expectation value of the function f(x)

Probability Distribution

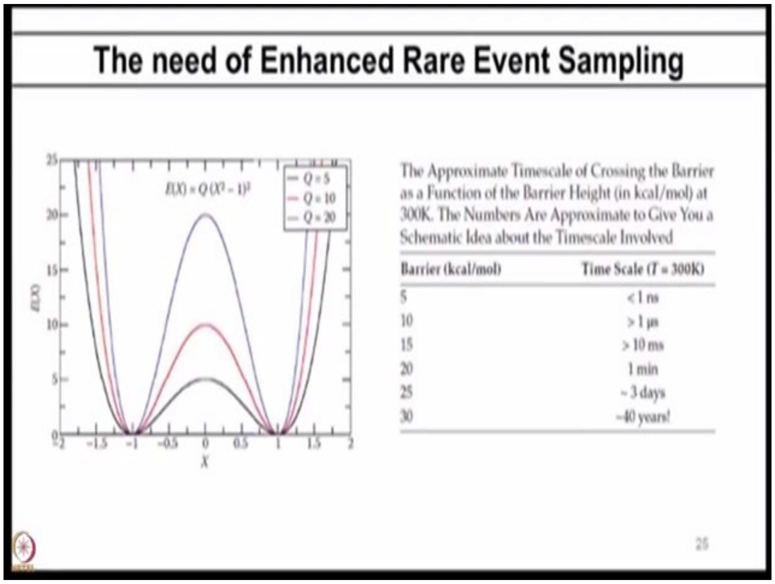
IS: $$E[\mathrm{f(X)}] = \underbrace{\int dx\, f(x)\frac{p(x)}{g(x)}g(x)}_{\downarrow}$$

This now can sample from another distribution g(x)
p(x)/g(x) is the sampling ratio.

As we have emphasized the Monte Carlo using metropolis algorithm essentially ensures the Boltzmann averaging over configurations and Boltzmann sampling of the configurations. Important sampling on the other hand tries to go where Boltzmann probability distribution is very small. So, conventional metropolis algorithm does not work. As we already said these are the real processes like in phase transitions which one has to go over a large free energy barrier as obtaining if the barrier is very important.

The way it is done is actually quite smart. You define another a new sample from the other distribution. So you write like here the expectation value as an integration and you define $p(x)/g(x)$ and now as a result of that $g(x)$ that you have, this gives you an average over the term is effects $p(x)g(x)$ and $g(x)$ now you have the freedom to choose and so choose the value $g(x)$ that it is it is not small in the region that you want to probe.

(Refer Slide Time: 30:38)



The need of Enhanced Rare Event Sampling

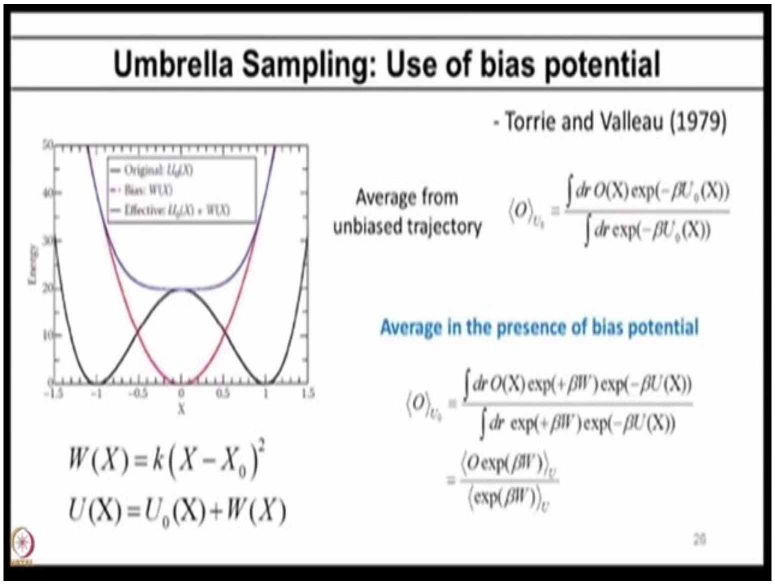
The Approximate Timescale of Crossing the Barrier as a Function of the Barrier Height (in kcal/mol) at 300K. The Numbers Are Approximate to Give You a Schematic Idea about the Timescale Involved

Barrier (kcal/mol)	Time Scale (T = 300K)
5	< 1 ns
10	> 1 μs
15	> 10 ms
20	1 min
25	~ 3 days
30	~40 years!

25

So this is most important is the rare event sampling where you do not get to the barrier region and it is shown on the table right side that if you go to our large barrier then you can go days to years to sample them.

(Refer Slide Time: 30:58)



Umbrella Sampling: Use of bias potential

- Torrie and Valleau (1979)

$$W(X) = k(X - X_0)^2$$

$$U(X) = U_0(X) + W(X)$$

Average from unbiased trajectory

$$\langle O \rangle_{U_0} = \frac{\int dr\, O(X) \exp(-\beta U_0(X))}{\int dr \exp(-\beta U_0(X))}$$

Average in the presence of bias potential

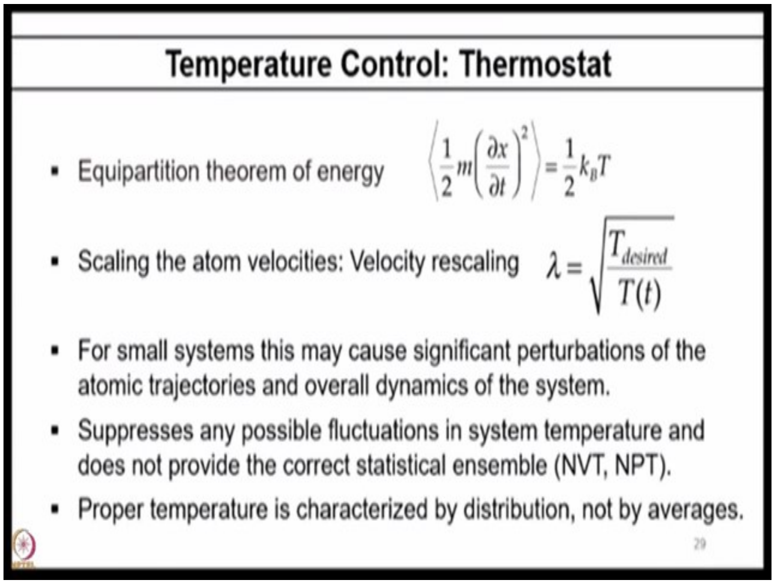
$$\langle O \rangle_{U_0} = \frac{\int dr\, O(X) \exp(+\beta W) \exp(-\beta U(X))}{\int dr\, \exp(+\beta W) \exp(-\beta U(X))} = \frac{\langle O \exp(\beta W) \rangle_U}{\langle \exp(\beta W) \rangle_U}$$

26

However, you can now devise a distribution like it is defined here the $e^{\beta W(x)}$ and that pulls you towards the distribution and is shown here that biases $W(x)$ is shown so bias is a harmonic placed exactly at the maximum. So that now the minimum is placed exactly at the barrier, so coupled to this added together allows you sample the barrier region and I am not going to go through the mathematics.

But you can easily see, how a new term $e^{\beta W(x)}$ is to introduced and then you get the sampling over this region of importance. These are rather advanced techniques but it is good to know. So there is a drawback of the importance sampling.

(Refer Slide Time: 32:13)



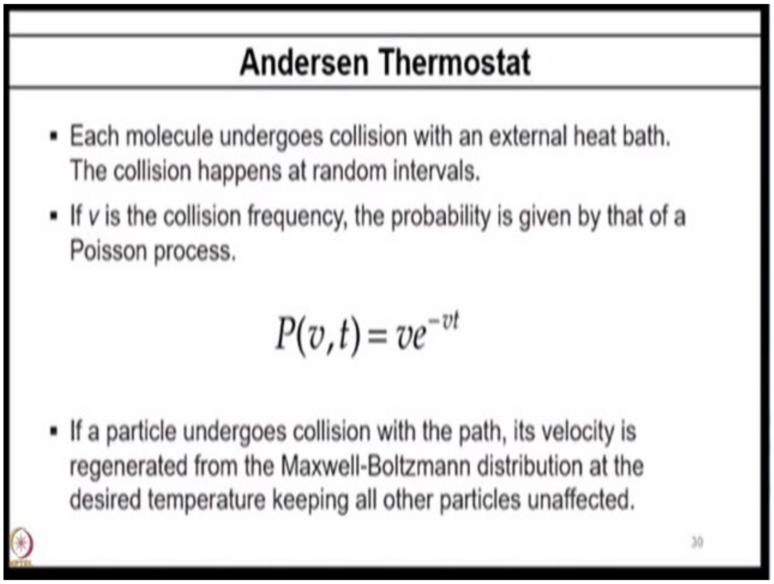
Temperature Control: Thermostat

- Equipartition theorem of energy $\left\langle \frac{1}{2} m \left(\frac{\partial x}{\partial t} \right)^2 \right\rangle = \frac{1}{2} k_B T$
- Scaling the atom velocities: Velocity rescaling $\lambda = \sqrt{\frac{T_{desired}}{T(t)}}$
- For small systems this may cause significant perturbations of the atomic trajectories and overall dynamics of the system.
- Suppresses any possible fluctuations in system temperature and does not provide the correct statistical ensemble (NVT, NPT).
- Proper temperature is characterized by distribution, not by averages.

One is that how do you keep the temperature and pressure because these things are completely fluctuating and these are small systems. Temperature is easy to understand temperature you do by descaling the velocity like this described here. I know it is Equipartition theorem and I know that velocity square average of squared gives temperature see the temperature goes up.

When the temperature goes up, scale back. There are many ways of do that. So rescale the temperature can be done by velocity is carrying. Similarly, temperature falls down so I can upscale the velocity. However, it is easier said than done because we have to do it systematically and you have to go without perturbed the trejectory.

(Refer Slide Time: 33:08)



Andersen Thermostat

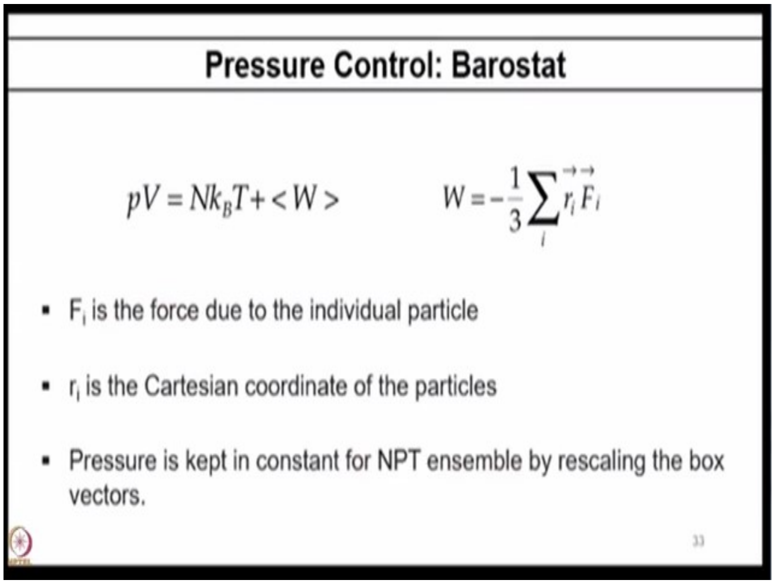
- Each molecule undergoes collision with an external heat bath. The collision happens at random intervals.
- If v is the collision frequency, the probability is given by that of a Poisson process.

$$P(v,t) = ve^{-vt}$$

- If a particle undergoes collision with the path, its velocity is regenerated from the Maxwell-Boltzmann distribution at the desired temperature keeping all other particles unaffected.

There is a bit done first one was done by Anderson well Anderson thermostat so he considered that there is an external bath and each molecule is undergoing a condition. So the bath permits the system and the heat bath collision frequency given by probability and that maintains Maxwell-Boltzmann velocity probability distribution of the particles.

(Refer Slide Time: 33:42)



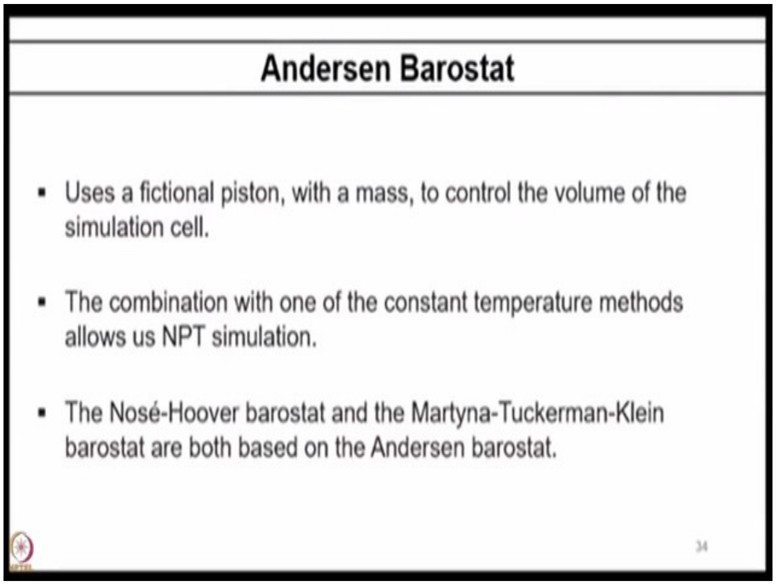
Pressure Control: Barostat

$$pV = Nk_BT + \langle W \rangle \qquad W = -\frac{1}{3}\sum_i \vec{r}_i \vec{F}_i$$

- F_i is the force due to the individual particle
- r_i is the Cartesian coordinate of the particles
- Pressure is kept in constant for NPT ensemble by rescaling the box vectors.

Then how do we control pressure? Pressure is a little bit more tricky and little bit more dangerous. When you do NPT simulations even I do NVT simulations like the canonical ensemble or grand canonical ensemble simulations or microcanonical much of the time, we do microcanonical. But we need to get the pressure correct. So many times what we do we keep a pressure and run the trajectories and stabilization system and then you see what is the volume then we remove the pressure but before that we need pressure. Pressure is fixed by using Barostat as you know from experiment. In this case pressure is kept constant for NPT ensemble buy rescaling the box vectors.

(Refer Slide Time: 34:29)

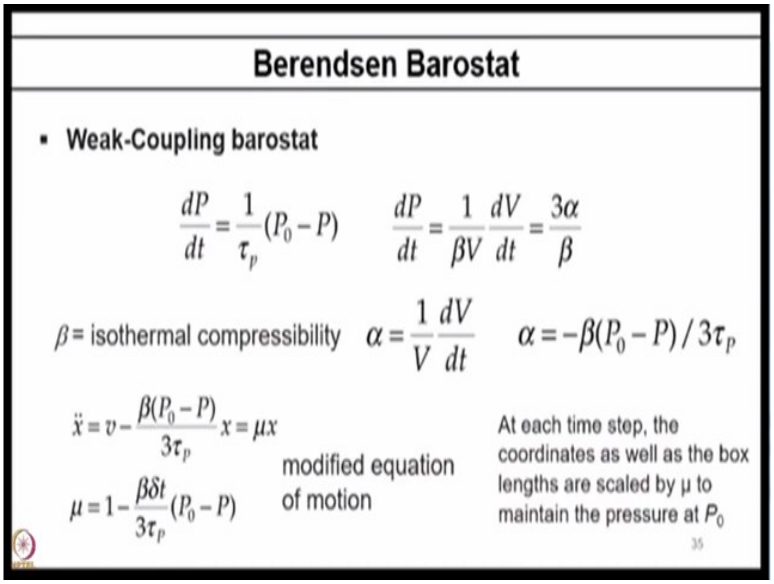


Andersen Barostat

- Uses a fictional piston, with a mass, to control the volume of the simulation cell.
- The combination with one of the constant temperature methods allows us NPT simulation.
- The Nosé-Hoover barostat and the Martyna-Tuckerman-Klein barostat are both based on the Andersen barostat.

So you just like pressure is where you having a piston here you are control the volume of the simulations shell.

(Refer Slide Time: 34:41)



Berendsen Barostat

- **Weak-Coupling barostat**

$$\frac{dP}{dt} = \frac{1}{\tau_p}(P_0 - P) \qquad \frac{dP}{dt} = \frac{1}{\beta V}\frac{dV}{dt} = \frac{3\alpha}{\beta}$$

β = isothermal compressibility $\quad \alpha = \frac{1}{V}\frac{dV}{dt} \quad \alpha = -\beta(P_0 - P)/3\tau_p$

$$\ddot{x} = v - \frac{\beta(P_0 - P)}{3\tau_p}x = \mu x$$

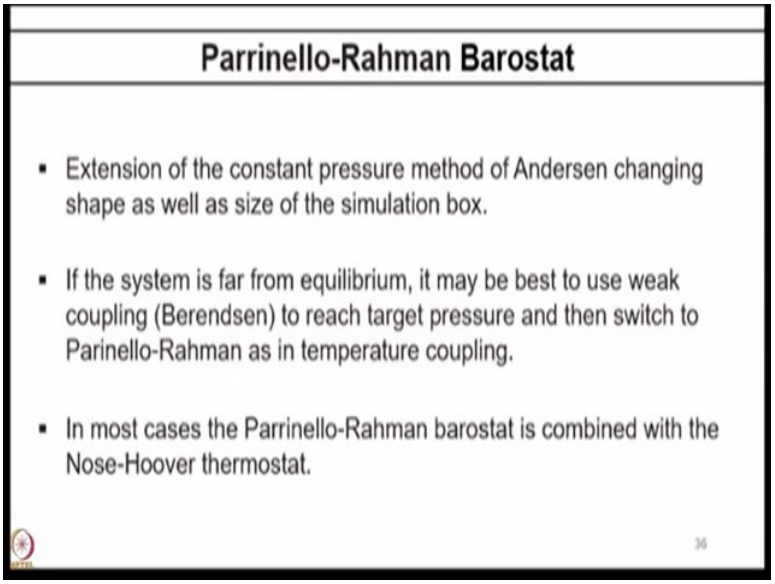
$$\mu = 1 - \frac{\beta\delta t}{3\tau_p}(P_0 - P)$$

modified equation of motion

At each time step, the coordinates as well as the box lengths are scaled by μ to maintain the pressure at P_0

And they have again Berendsen Barostat that is again use very similar technic that used here that you have pressure weak coupling barostat and at each time step now instead of atom or molecules you are changing the coordinate of the box.

(Refer Slide Time: 35:02)

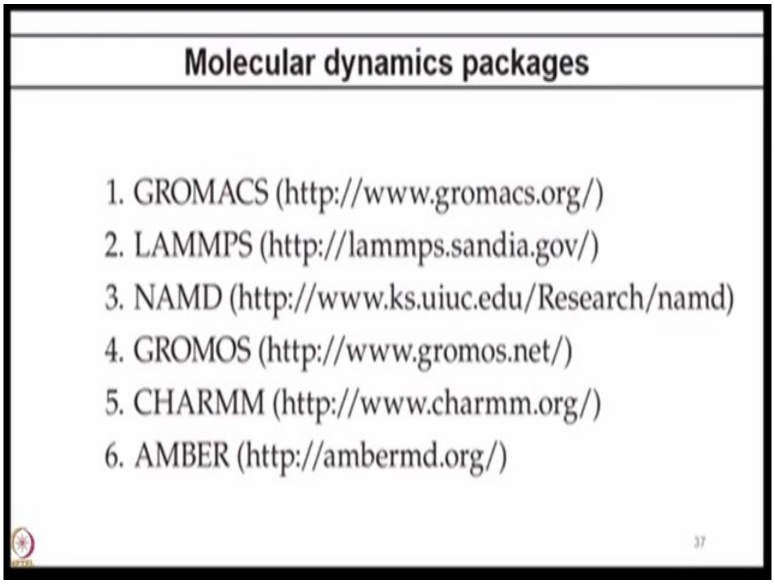


Parrinello-Rahman Barostat

- Extension of the constant pressure method of Andersen changing shape as well as size of the simulation box.
- If the system is far from equilibrium, it may be best to use weak coupling (Berendsen) to reach target pressure and then switch to Parinello-Rahman as in temperature coupling.
- In most cases the Parrinello-Rahman barostat is combined with the Nose-Hoover thermostat.

Has been done by Parrinello-Rahman Barostat that you not only change its shape, you also change the size. Actually these are very nice fast computer simulation of solid state phase transition was done by Parrinello-Rahman, where each change the shape and that was then they were not much more by Parinello. They work more extensively with Rahman, so you are now changing the sheet shape, changing the volume. That means your system is responsive but doing that you can keep the target pressure.

(Refer Slide Time: 35:48)

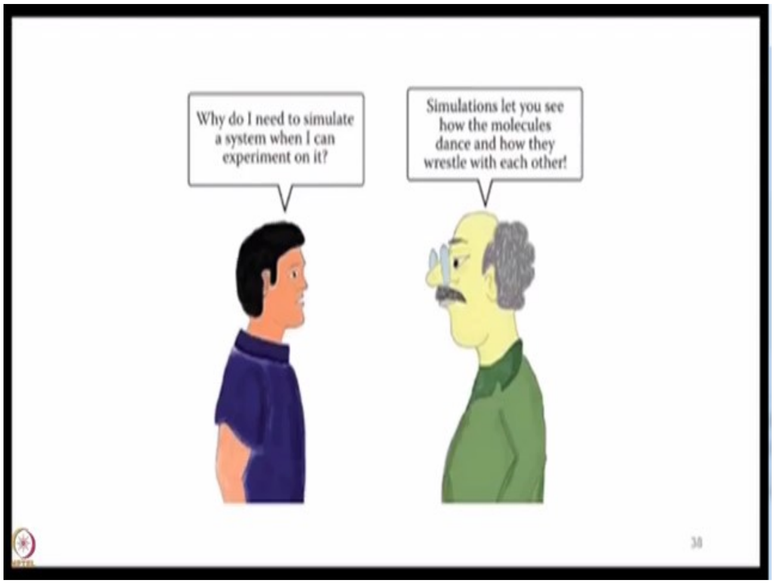


Molecular dynamics packages

1. GROMACS (http://www.gromacs.org/)
2. LAMMPS (http://lammps.sandia.gov/)
3. NAMD (http://www.ks.uiuc.edu/Research/namd)
4. GROMOS (http://www.gromos.net/)
5. CHARMM (http://www.charmm.org/)
6. AMBER (http://ambermd.org/)

So there as I say there are lots of packages, most of the packages are, molecular dynamic packages, you know I, GROMACS, LAMMPS, NAMD. I told you that in Monte Carlo simulations most of the time you had to write your own code, though there are some codes that are available in different books and in papers but then you do not have packages. So, and one usually has to be more creative in doing Monte Carlo simulation.

(Refer Slide Time: 36:28)



These are the questions I have been asked often when I give talk by experimentalist doing watching some stage simulation replace the experiment that probably never happen. He does one thing is that, yes simulations are getting more and more powerful. More importantly simulations gives you information that many times experiments not yet that gives with microscopic information and if you know how molecules are interacting.

So it gives you by comparing theory with simulation experiments. We know whether force field are directly interacting but that is the law. Once you know how the molecules will interact. You can change the interaction with some substitution and you can create new materials. So now I will stop here, but I strongly recommend to see the books and I did it and many other books are there.