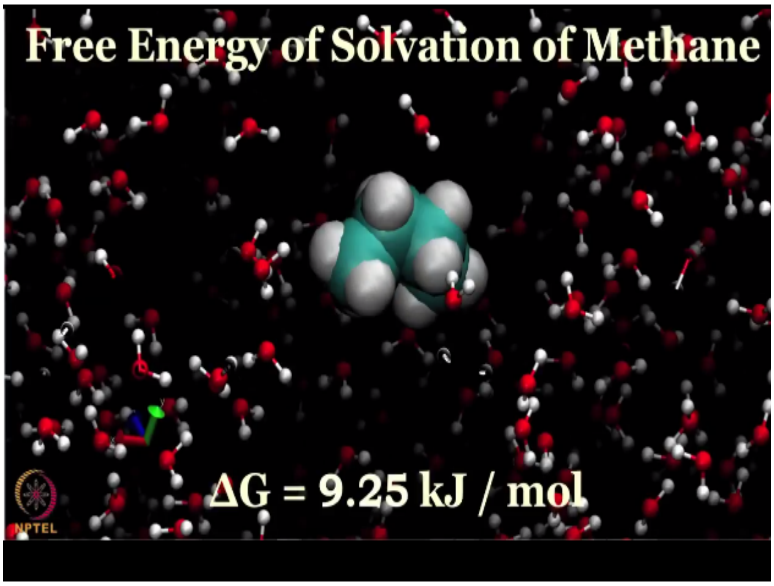
Computational Chemistry & Classical Molecular Dynamics
Prof. B. L. Tembe
Department of Chemistry
Indian Institute of Technology – Bombay

Lecture – 41
Molecular Dynamics Using GROMACS-9: Free Energy of Solvation of Methane, Concluding Remarks

(Refer Slide Time: 00:15)



Free Energy of Solvation of Methane

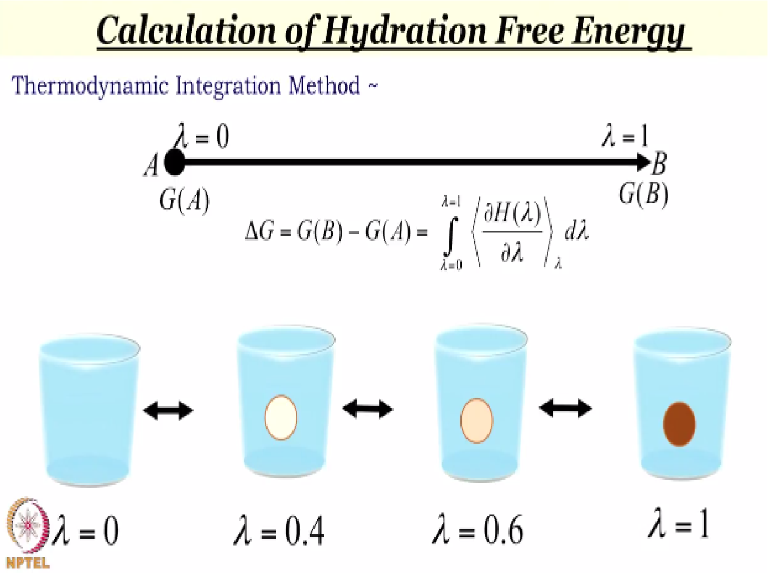
$\Delta G = 9.25$ kJ / mol

Our next simulation example will be the solvation free energy of methane. Normally when you have a single liquid and add a second liquid to it when the 2 liquids mix completely that means each molecule is surrounded by not only it is own neighbours but also neighbours of a different kind. So if all possibilities are allowed then mixing is possible, but suppose A and B are not interacting with each other strongly, say for example oil and water.

Oil and water will not mix because there is a lot of repulsion between the 2 objects. So before we want to know whether something is miscible or not one idea would be to take a single molecule and see what is the free energy of solvating that molecule in a liquid that is what we call as free energy of solvation and in this particular lecture we will see how to solvate a molecule in a liquid.

So the first slide tells you that there is one molecule there and you try to solvate it and if the free energy is positive that means it is in not a favorable arrangement. So to find out how that delta G comes about this whole lecture is about. So can we have the next slide. So in this slide we will see how to hydrate a particular molecule. Go to the next few okay.

(Refer Slide Time: 01:40)



Calculation of Hydration Free Energy

Thermodynamic Integration Method ~

$$\Delta G = G(B) - G(A) = \int_{\lambda=0}^{\lambda=1} \left\langle \frac{\partial H(\lambda)}{\partial \lambda} \right\rangle_{\lambda} d\lambda$$

So this is called a thermodynamic integration method, what we do in this is insert a cavity in a liquid and that cavity is initially not interacting with the medium. So and in that cavity you put a molecule so when lambda = 0 that cavity does not interact with the rest of the medium. Now slowly when lambda goes from 0 to 1 the interaction is completely established. See for example suppose which is this molecule we are considering methane? Methane, yeah.

So suppose I have a methane molecule we know that there are 5 sites in it, 1 carbon and 4 hydrogen. So when they do not interact with the solvent that lambda is 0 and if I suddenly put lambda = 1 there will be lot of repulsion and the system will not like it very much. Therefore, what do we do say lambda = 0, no interaction, when lambda is a small value. Here we have shown 0.04, but I will actually put 0.01 or 0.1.

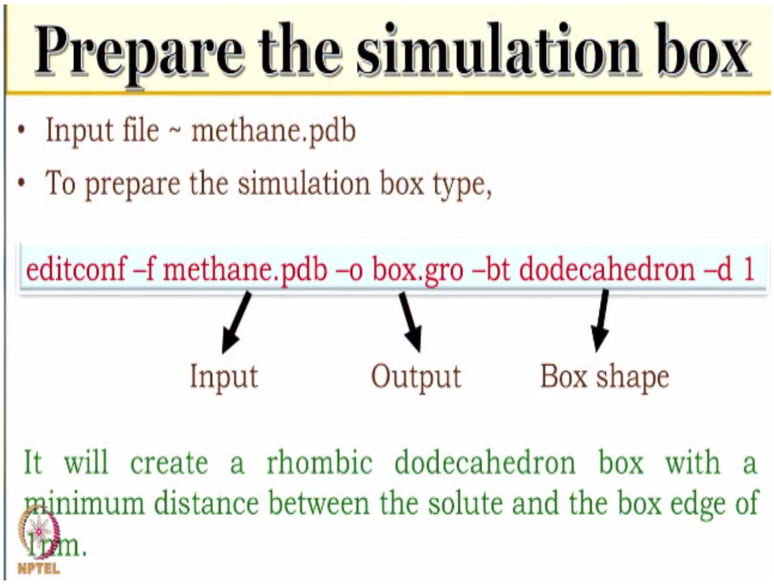
When I put lambda = 0.1, the entire interaction is not turned on, only if it is 0.01 only 1% of the interaction is turned on and when 1% of the interaction is turned on this 1% interacts with the rest of the solvent and slowly aligns that solvent in a favorable arrangement. So next time I make lambda = 0.02, so that is I enhance it a little more then, more and more interaction will occur and eventually in the end lambda =1 so my entire interaction is put on then that is my final simulation.

To find out the free energy from 0 to 1 first I have to take, at each step there is an interaction. So my Hamiltonian now or the interaction it depends on these parameters lambda. So I take a derivative of H with respect to lambda that is called DH lambda/D lambda, I take an ensemble average. What is the meaning of ensemble average? I am going to do a large number of steps in my simulation.

Say 100,000 or 1 million steps so when I average that quantity I am getting DH lambda/D lambda at each point in the simulation. So once I have that for several lambda points let us say 100 lambda points. I start with 0.01, 0.02, 0.03 and so on. And when I integrate all from 0 to 1 that entire integral gives me the free energy of state B-free energy of state A that is called GB-GA that is your delta G, that is the free energy of solvation.

This is what we want to calculate using the GROMACS software. So all the steps needed will now be introduced to you. You can start with the next step. So you see even in the picture initially there is nothing lambda = 0, lambda = 0.4 it is a light white ball then it becomes more coloured and eventually when lambda = 1 the entire interaction is switched on. So this is the switching off, switching on procedure and we will see how GROMACS enables you to do this. So first we have to prepare the simulation box right.

(Refer Slide Time: 04:43)



Prepare the simulation box

- Input file ~ methane.pdb
- To prepare the simulation box type,

```
editconf –f methane.pdb –o box.gro –bt dodecahedron –d 1
```

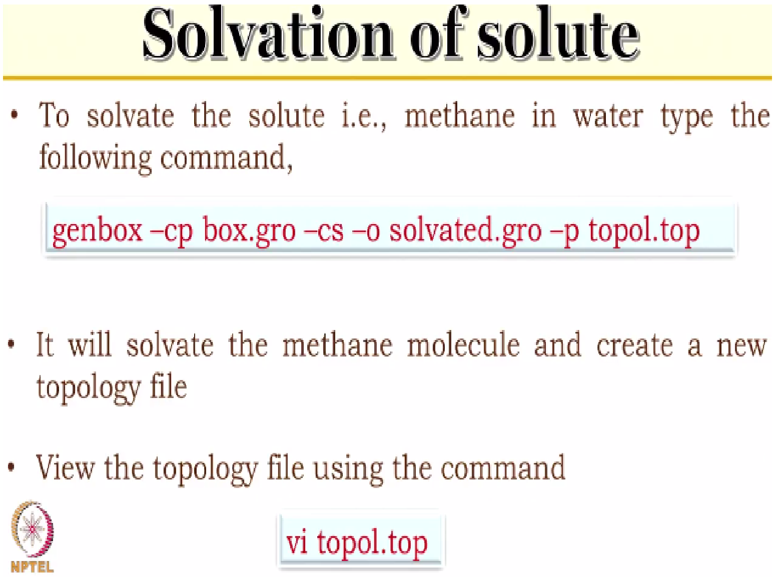
Input
Output
Box shape

It will create a rhombic dodecahedron box with a minimum distance between the solute and the box edge of 1 nm.

We want to calculate the free energy of hydration of methane. So the input file that is required is methane.pdb. Now we have the pdb file of methane. Now what we do we create a box around the solute molecule and to prepare the box we will use the command editconf, so editconf-f methane.pdb-o box.gro that is my output file, gro is the gromos structure file and –bt dodecahedron, bt is the box type, dodecahedron is my box shape and –d1.

So what will it do? It will create a dodecahedron box with the minimum distance between the solute and the box edge being 1 nanometer.

(Refer Slide Time: 05:30)



Solvation of solute

- To solvate the solute i.e., methane in water type the following command,

```
genbox –cp box.gro –cs –o solvated.gro –p topol.top
```

- It will solvate the methane molecule and create a new topology file
- View the topology file using the command

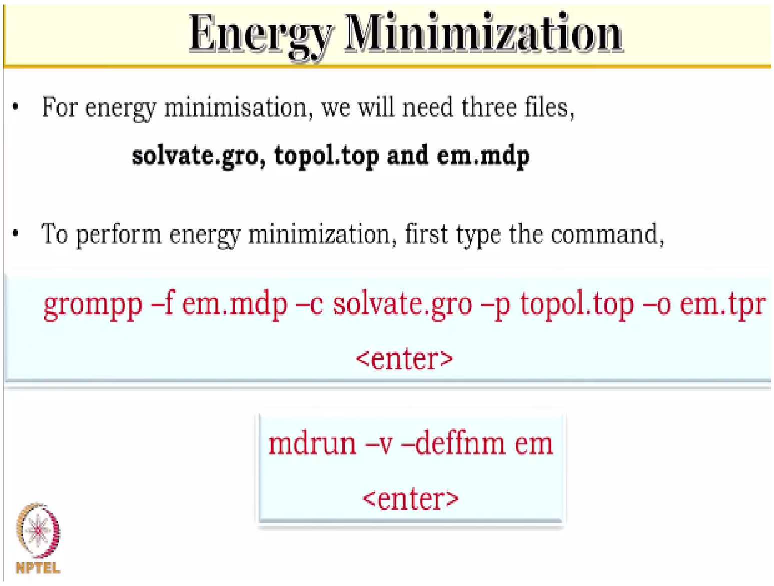
```
vi topol.top
```

So after we have prepared the simulation box now what we have to do we have to solvate the solute molecule. Now to solvate the solute molecule we will use the command genbox as we have used in case of s-peptide. So we will type the command genbox space –cp box.gro space –cs space –o solvated.gro space –p topol.top. Why do you have 2 of these next to each other –cs and –o? –cs is used for the solvent input and –cs solvent input when you download GROMACS there is already spc.gro downloaded automatically.

So it will take the spc.gro file from GROMACS and –cp is for the solute size. So it will add the solvent from spc.gro and then it will create my output file that is solvated.gro and it will also modify my topology file. Initially my topology file I will only have the coordinates of methane. The force filed parameter or methane molecule, but now since I also have water molecules.

So it will add water molecule in my topology file as we have already seen that in case of s-peptide also it modifies the topology file right. Okay. So it will solvate my methane molecule and it will create a new topology file. So we can view the topology file by using the command vi that is vi topol.top.

(Refer Slide Time: 07:00)



Energy Minimization

- For energy minimisation, we will need three files,

  solvate.gro, topol.top and em.mdp

- To perform energy minimization, first type the command,

```
grompp –f em.mdp –c solvate.gro –p topol.top –o em.tpr
<enter>
```

```
mdrun –v –deffnm em
<enter>
```

And now we will perform energy minimization. So what we have? we have solvate.gro which is methane solvated by water. Then we have topol.top and we have em.mdp, em.mdp is my energy minimization, molecular dynamics parameter file. What about the sizes of the boxes? Where will now, in this case you have a dodecahedron, what size will it choose or we have to specify that in some file?

You can either specify it or it basically what it does it will take I have specified my distance between the solute and the box edge that is 1 nanometer. So it will create a box around it. So when you type the command editconf it will display on the screen what is the box length that it has chosen right. Now when it will solvate it, it will slightly modify the box length. So it will also be displayed in this screen.

You can open the file solvate out gro by using the command vi and then you go to the last line it will display you the box length, what is the box length and the box length will be according to the type of box that you have chosen. Suppose in this case I have taken a dodecahedron box so it will give me the box length according to a dodecahedron box. It can be done using a cubic box also?

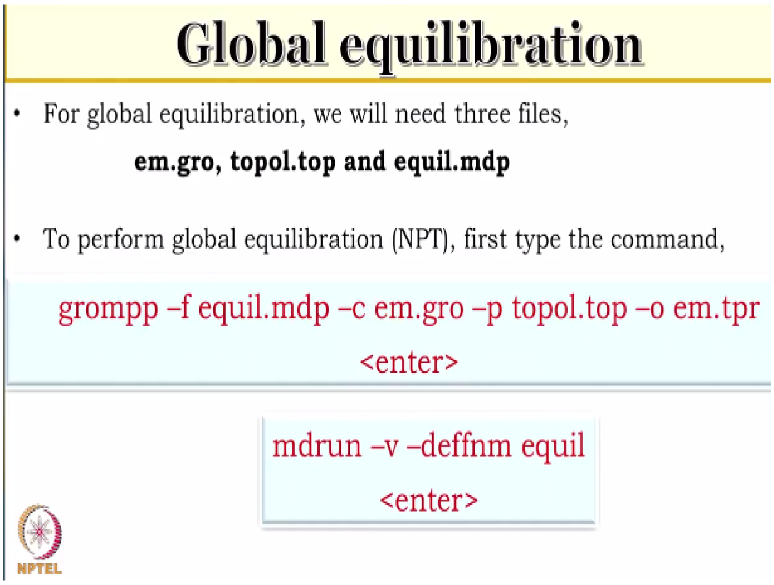
Yeah it can be done using a cubic box so what you do there is –bt then instead of dodecahedron you type cubic. So it will create a cubic box. Okay, so now we have solvate. So the advantage of the dodecahedron is that there are less edges, see in a cube the edges are very sharp. So when you have a dodecahedron there are less sharp edges and the periodic boundaries are not as strong because cubic is a very strong periodicity whereas the dodecahedron it is much smoother ideally spherical boxes will be nice.

But a sphere cannot be completely surrounded by all the sphere there will be gaps. So you need only those edges where infinite medium can be created by replication, okay. Yeah go ahead. So now we already have solvate.gro file which is my methane solvated by water. We have the topology file and we have energy minimization file. This em.mdp file is same for all the case like we have done in argon, water, water methanol, s-peptide, the formate does not change.

So m is the final energy minimized configuration? No this is the molecular dynamics parameter file. Okay. This parameter file it remains almost the same, the only thing we can change here is the cut off length. That will be according to my box length right. So now to perform energy minimization we will first need a tpr file. So to generate a tpr file we will use the command grompp space –f space em.mdp which is my energy minimize molecular dynamics parameter file.

Then –c space solvate.gro which is the gromos structure file and – space p space topol.top that is the topology file and then – space o em.tpr that is my output file. So now we will use the tpr file to generate all my energy minimized configuration. So what we will do, we will use the command mdrun. So mdrun space –v space –deffnm space em. So I will get my energy minimize configuration that is em.gro and it will also write the minimum that is the minimum value of potential energy.

(Refer Slide Time: 10:40)



Global equilibration

- For global equilibration, we will need three files,

  em.gro, topol.top and equil.mdp

- To perform global equilibration (NPT), first type the command,

```
grompp –f equil.mdp –c em.gro –p topol.top –o em.tpr
<enter>
```

```
mdrun –v –deffnm equil
<enter>
```

So after this what we do we proceed towards the equilibration. So it depends which type of simulation you want to do. Whether you want to do in NVT or you want to do in NPT. Where is this specified? This is specified in the equil.mdp file, right. So for equilibration again we will need the gro file from energy minimization. We will need the topology file and the mdp file for equilibration.

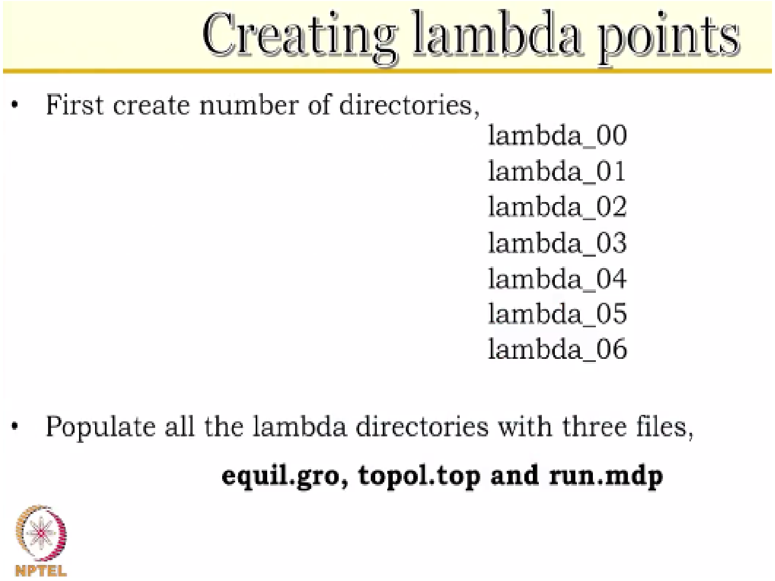
So here how do we specify suppose I want to perform an NPT ensemble. So what is that required? We need to keep the pressure constant, temperature constant and number of molecules constant. So number of molecules constant is being taken care of by the periodic boundary condition that we are using for pressure being to keep constant we will use the a barostat to keep the pressure constant.

And to keep the temperature constant we will use a thermostat in the mdp file. Since there are many ways of keeping pressure constant and temperature constant does the GROMACS give you different options for them? Yeah, it does give different options for them you have to choose accordingly that what is the kind of barostat or thermostat that you want to choose. Are there options like 1, 2, 3 or you have to give the name of the procedure?

You have to write the name; you have to write the correct name of the thermostat or barostat that you want to use. Okay. So once you have the mdp file em.gro and topology file we will again generate a tpr file here. So the command we will use is grompp space –f space equil.mdp where it is the molecular dynamics parameter file for my equilibration in NPT ensemble.

Then –c space em.gro that is my energy minimized configuration, then space –p topol.top, the topology file then space –o it will be equil.tpr. So after I get my equil.tpr file I will run md run using the command md run space –v, space –deffnm equil. So it will create all files with the starting name equil? Yeah. How many will be, which are the files? I will need the gro file for my next step. Okay.

(Refer Slide Time: 13:00)



Creating lambda points

- First create number of directories,

lambda_00
lambda_01
lambda_02
lambda_03
lambda_04
lambda_05
lambda_06

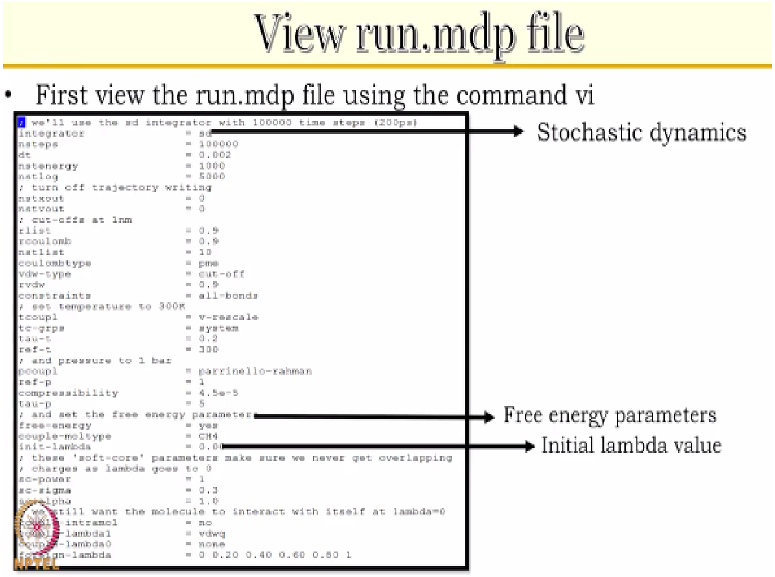
- Populate all the lambda directories with three files,

equil.gro, topol.top and run.mdp

So after it is done so as you have already said earlier that we need to create a lambda point here. Lambda it specifies the path between A and B state. So for our sake we will create lambda point say first point will be 0 right. Then we will have 0.1, 0.2, 0.3, 0.4, 0.5, 0.6 till 1. Correct, 0.01, 0.02, 0.03 yeah that depends on the user that how many steps he or she wants to take between these two.

So you are not showing all the 100 directories, you have shown only 6 or 7 of them. Yeah. Okay. And now we have to populate all this lambda directories what we need is we need the gro file from the equilibration. We need the topology file and we need the run.mdp file. Now this mdp file is different from the equil.mdp or em.mdp file right.

(Refer Slide Time: 13:52)



View run.mdp file

- First view the run.mdp file using the command vi

```
; we'll use the sd integrator with 100000 time steps (200ps)
integrator               = sd
nsteps                   = 100000
dt                       = 0.002
nstenergy                = 1000
nstlog                   = 5000
; turn off trajectory writing
nstxout                  = 0
nstvout                  = 0
; cut-offs at 1nm
rlist                    = 0.9
rcoulomb                 = 0.9
nstlist                  = 10
coulombtype              = pme
vdw-type                 = cut-off
rvdw                     = 0.9
constraints              = all-bonds
; set temperature to 300K
tcoupl                   = v-rescale
tc-grps                  = system
tau-t                    = 0.2
ref-t                    = 300
; and pressure to 1 bar
pcoupl                   = parrinello-rahman
ref-p                    = 1
compressibility          = 4.5e-5
tau-p                    = 5
; and set the free energy parameters
free-energy              = yes
couple-moltype           = CH4
init-lambda              = 0.00
; these 'soft-core' parameters make sure we never get overlapping
; charges as lambda goes to 0
sc-power                 = 1
sc-sigma                 = 0.3
sc-alpha                 = 1.0
; we still want the molecule to interact with itself at lambda=0
couple-intramol          = no
couple-lambda1           = vdwq
couple-lambda0           = none
foreign-lambda           = 0 0.20 0.40 0.60 0.80 1
```

Stochastic dynamics

Free energy parameters

Initial lambda value

So do you have to load all the directories in the 3 files? Yeah, we have to populate it because these 3 files need to be there in each lambda directory. But suppose I want to do lambda 06 calculation for that will I not need the configuration from lambda 05? No you have the equilibrated configuration right. So once you have the equilibrated configuration you can start your calculation at each lambda point separately.

You do not need file from lambda 05 or 06. Okay. So this is the run.mdp file what we do here we use your stochastic dynamics. So in integrated we have to write sd which specifies stochastic dynamics. So which is this file you are showing the mdp file, run.mdp file and then we also have to specify free energy parameters, right. Here it is shown the free energy parameters where free energy which is = s that means you are calculating free energy.

Then couple molecule type here, here you have to type the residue name of your solute, which you want to couple, then you will have the innate lambda that is the initial value of lambda. So initial value is 0 right. Then you will have softcore potential that is sigma value, then power and all this free energy stuff like you want to couple intramolecular interaction. Then there will be lambda.

Then there will be foreign lambda that is rest of the lambda values. So these are the free energy parameters that you have to specify in your mdp file. If you go in the GROMAC side it gives you a list of parameters that you want to, you can keep for your free energy calculations. So it is already available there, it is written in detail that what are the parameters that you need for an mdp file.

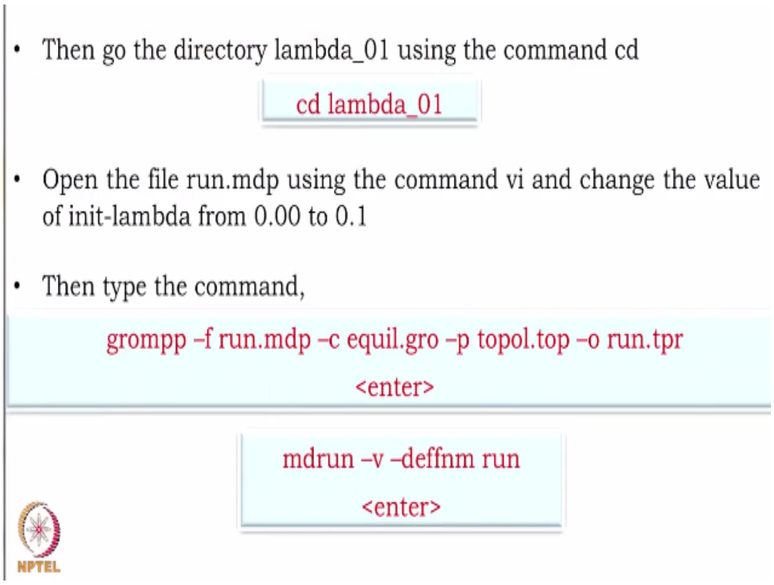
You just have to add all this line at the end of your mdp file that you are already using for your previous calculations. Have they given some examples in the? Yeah, there are examples given in there. So methane is one example or we are creating separately. Methane is an example. Ethanol is an example, so they already have the mdp file. Okay. So once we have the mdp file in each of the lambda directories.

So now what we do we go to the directory first we have lambda 0 right, so we go to the directory by using the command cd. So now we will in the lambda_00 directory we will have the gro file from the equilibration, we will have the topology file and we will have the mdp file for free energy calculation. So here also the same command, we first have to generate a tpr file and then we have to do an mdrun.

So for generating a tpr file we will use the command grompp, so grompp space –f space run.mdp this is the molecular dynamics parameter file for free energy calculation then space –c, space equil.gro that is my equilibrated structure file then space –p then space topol.top which is my topology file, then space –o space run.tpr that is my output file. So now I will run the tpr file using the command md run.

So mdrun then space –v space –deffnm space run. So the output file which is of my interest in calculating free energy is the xvg file. So we will have this run.xvg file which will contain the potential energy as the function of time. Suppose you are performing for 200 picosecond. So from 0 to 200 picosecond it will write the potential energy at each time frame. So similarly you have done calculation in lambda 00 directory.

(Refer Slide Time: 18:00)



- Then go the directory lambda_01 using the command cd

```
cd lambda_01
```

- Open the file run.mdp using the command vi and change the value of init-lambda from 0.00 to 0.1
- Then type the command,

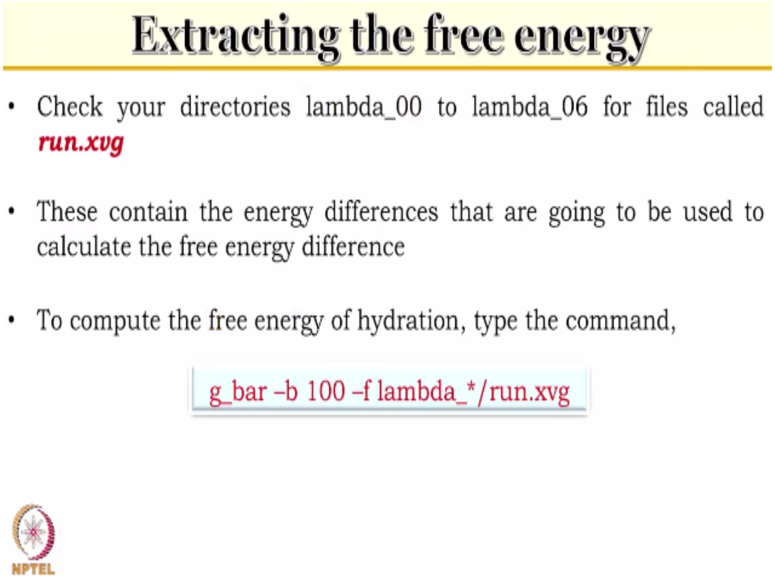
```
grompp –f run.mdp –c equil.gro –p topol.top –o run.tpr
<enter>
```

```
mdrun –v –deffnm run
<enter>
```

So now you will go in a lambda 01 directory. So here also you then change your init-lambda value in the run.mdp file from 0.00 to 0.1. So you have to change that in each of the directory? In each of the directory. Whatever actually lambda I want. Yeah, because starting lambda point is 0, but as you change the directory your lambda point will also keep on changing. So you have to change the lambda value and then again you have to generate the tpr file.

And then using the tpr file you will run mdrun and then you will get the xvg file that is the run.xvg file. So this is the procedure that you have to follow till lambda =1. Okay. And at each lambda directory you have to change the init-lambda value in the mdp file. So once you have all the xvg file in each of the lambda directory what you do is.

(Refer Slide Time: 18:51)



Extracting the free energy

- Check your directories lambda_00 to lambda_06 for files called ***run.xvg***
- These contain the energy differences that are going to be used to calculate the free energy difference
- To compute the free energy of hydration, type the command,

```
g_bar –b 100 –f lambda_*/run.xvg
```

This is the file that contain the energy differences and this is the energy difference that we will use to calculate the free energy difference. So now to compute the free energy of hydration of methane we will just type the command g_bar –b100, -b100 is it will discard the initial 100 picosecond.

It will consider it as further equilibration. So –f lambda_star that is it will go to all the lambda directories and then it will take the run.xvg file from each of the lambda directories it will calculate the free energy difference and then it will finally display it on the screen whatever the value of your free energy of hydration is. So the final output delta g will be given after this command?

Yeah, it will write on the screen. On the screen? Yeah. It will also write in some other file, log file? Yeah, you have to give an option for that –o. Okay. And then you can also get a histogram of DH D lambda as a function of lambda how it looks. So is that contained, where is that contained in that log file? That is the several output options that you will get, once you type g_bar space –h it will give you an n number of options for your output file whatever output file you want to have.

So suppose I want DH/D lambda at every step, there is an option to see that? Yeah, there is an option to see that. Then you can also plot that DH/D lambda as a function of lambda? Yeah, that you can plot. And sometimes the DH/D lambda is it always having one sign or it can be different? It can be different depending on your system, what is the solute, what is your solvent, like free energy it is different for different solute and solvents.

So now before running this command should we not check whether every lambda value calculation is done properly? Yeah definitely, that we have to check that every lambda directory has to have this dot xvg file. Right because even if one is missing this will not run? This will run, this will give you the value, but it may not give you the correct value right, because if one lambda point is missing.

Right then this will have a problem know it will not have that file at all? It will give you the value of 0 at that point. So it will take 0 and do the calculation. So we have to make sure that all the lambda points at each lambda directory we have the xvg file which contain the free energy differences right. Supposing lambda = 02 0.2 there is no you have not done calculation in lambda = 0.

So from 0 to 0.2 it will take some arbitrary value, it will not give you the correct value right. It will start from 0.1 to 0.2, but it would not give you the value what you would expect from 0 to 0.2. But it will not give an error saying that file missing or anything? No it would not give an error. So one has to be extra careful in running this thing? Yeah, one have to make sure that at each lambda point we have run the simulation, we have the xvg file at each lambda point.

Because when you write this command g_bar it will display on the screen at each lambda like from 0 to whatever you are giving the difference of lambda you are taking. Suppose from 0 to 1 you have 0.2 you are taking right. So 0.2, so 0 to 0.2 then 0.2 to 0.4, 0.4 to 0.6, it will write the value of delta G in this thing. So one can also get an idea from there, whether it has.

So now the example we showed here we were saying lambda = 0.01, 0.02,0.03, it was almost linear from 0 to 1, but it is not necessary like that know because initial values the lambda has to be very close to each other. Later values it can be very different know. Yeah, later values you can. Because once I have switched on 80% of the potential then the differences need not be 0.01?

No the differences can be small. Smaller or larger you may need larger. Initially we have taken the difference very small because you will get a converge result. Otherwise from 0 to 0.4 you take suddenly it would not give you the correct result, you suddenly switch on the interaction from 0 to 0.4. So now the way you have described here you have to do simulations in each of the 100 directories? Yes.

Is there a way I can give a command where it will simulate in all the directories? Can you create some file like that? Then you will have to have a script file for that. Okay. So then it will go to each lambda directory, it will perform the simulation then it will go to other lambda then it will perform the simulation. So it will take some time to do that. So maybe we can in our website we can put examples of script files also.

Yeah, there are script file which you can use to run, you need not go to each and every lambda part. Does the manual also recommend this or the manual does not say about script files? In manual there are like only 5 lambda points that they are showing for practise purpose right. So there they really do not show the script file, but while we do free energy of solvation of ethanol they do use a script file then.

They have given a format of the script file and while you use a script file the possibility that you miss one lambda point is also reduced. Very large correct. Suppose there is some crash in between you will not know at all. No but then it will also give me an error file where I can see that what is my error, where my simulation stopped and if I run through a script file the chances that I miss one lambda directory will also be less.

Because it will cover all the lambda points. That is right. And in this case does it write many coordinates or just write the potential energy? Coordinate it depends on you, whether you want to write the coordinates or not. Because we do not really need the coordinates, because unnecessarily it will just fill up the disc. Yeah, we just need the energy differences, if you do not need the coordinate you just write in the mdp file that nxtout you write it 0, it would not give you any coordinate. Okay.

So when you, you said there are 2 different minimisation, one is the thermodynamic integration there is another method for small molecules know? That is the test particle insertion method. Right, this is doing the thermodynamic integration. Yeah. Is there a easy way to do the test particle insertion? Is there an algorithm given or there an option in GROMACS or there is separate? Yeah there is an option in GROMACS like here while we do free energy using a thermodynamic integration method we use stochastic dynamics right.

We use as an integrator we give the SD, so in test particle insertion method you have to write tpi. Okay. So it will perform simulation according to the test particle insertion method. So there also. There we do not need different lambda points. Okay, so and is there some limit to the size because? Yeah, there is a limit you cannot go beyond like neopentane or something. Only small particle can be done, test particle integration.

So can you do the free energy of very large molecules with this or there is some limitation, suppose I want to know the solvation free energy of a protein is it possible to do this? Yeah, maybe I mean. Are there some examples in literature or not too many? There might be some example I have to look after that. Okay, small peptide they must have solvated easily? Yeah, small peptide they must have because peptides even fulleries they have solvated, c180 they have solvated, they have calculated the free energy of hydration.

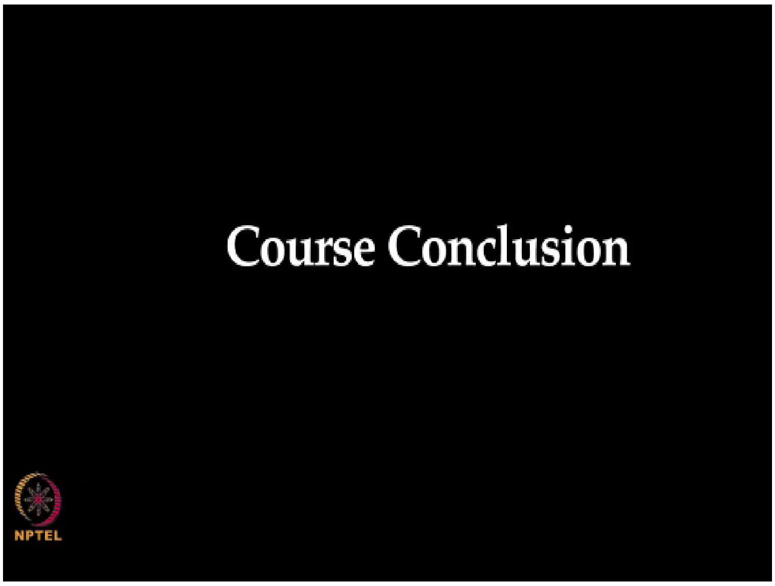
So I guess small peptides and c180 the size may be more or less similar. Okay, so, so far what we have done, we have done a simple simulation of liquid argon, then we did simulation of water, then water methanol mixtures then we consider an s-peptide and then we considered this free energy of solvation. So I think these 5 examples give you a good spectrum of different types of things that you can do with GROMACS simulations.

So in our demonstration we will demonstrate all these 5 things and we will give you assignments to do simulations on different molecules which are similar to this. So once you do this you will have a fairly good knowledge of what an MD simulation means and what are the kind of things you can do with the simulations. So with this although these programs are very involved maybe in our earlier discussion we will try to do a very simple liquid argon simulation using our own program.

That also we will try to do so that you know all the details of the algorithm of the simulation. So with this kind of a background now you have already done simple Fortran programming for several situations like matrix multiplication, matrix inversion, Fourier transforms and so on. And with molecular dynamics programs you have done detail calculations which can have a real impact and also a comparison with experimental data.

So this will really give you good practise on computational chemistry and I hope you will enjoy all these calculations and if you need any help you have our email addresses and you can write to us so that whatever difficulties you have we will be able to solve it.

(Refer Slide Time: 28:15)



Course Conclusion

With this lecture, we have now come to the end of our course computational chemistry and classical molecular dynamics. Let me summarise what we have done so far and what is expected from you. After an introduction to the methods of computer programing using a common language such as Fortran we have studied many common numerical methods such as interpolation, curve fitting, matrix methods, finding roots of equations integration and differential equations and also the use of random numbers.

While programming skills take a long time to develop there are excellent public domain software such as Scilab and we have also used Scilab software to supplement our understanding of numerical methods. While Scilab is very easy to use it is good to have a knowledge of the programs and algorithms for at least some of the numerical methods such as matrix inversion, roots of equation and Fourier transforms.

Because if we do not have basic understanding of the algorithms just using computer should not be a technique like clicking a mouse and getting a result. So algorithms are at the very root of programming and you should be able to appreciate as well as use algorithms to write your own simple programs. Now after programming Fortran and the use of Scilab we took up the problem of classical molecular dynamic simulations.

These are now being used to understand a lot of macroscopic chemistry as well as biology. We illustrated molecular dynamics by using a simple program on liquid argon using our own program and then use public domain software such as GROMACS to do molecular dynamic simulations on simple liquids, liquid mixtures, ions in solution as well as peptides in water.

An exposure to this course will help you to appreciate computational chemistry through first-hand experience and then enable you to explore more advanced computational chemistry.

Currently the scope of computational chemistry is increasing every day. The strength of the present course is the hands on experience in the use of numerical methods and other public domain softwares to perform chemical computation. After every lecture you will be required to complete one or two assignments so that you get a practical and first-hand knowledge of what has been described in the lectures.

If you need assistance, it will be available through TS as well as through email. Hope you enjoy participating in this course and we appreciate our association with you. Thank you very much and our best wishes.