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

Lecture - 33
Molecular Dynamics Using Gromacs-1

Today, we will start our discussions on this software Gromacs which allows us to do molecular dynamic simulations for a variety of systems. So before we start let me summarize what we have done so far. First, we started with basics of computer programming, then discussed several numerical analysis methods like interpolation, roots of equations, matrices, solving differential equations and so on.

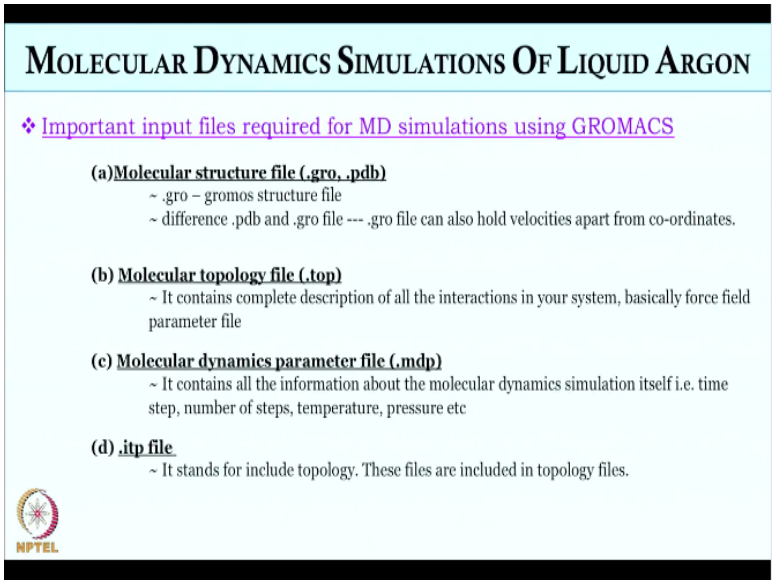
That gave you an idea of how to write program statements for all these issues. Then, we took up a very nice software Scilab in which all these numerical methods can be done almost instantly, so it is an extremely powerful software. So you have seen how to execute things using your own program, which is your dot slash a.out. In Scilab, you have a Scilab environment which is really a program that is running.

So now we want to do molecular dynamic simulations. These simulations involve a large number of molecules or that of the order of 100 to 1000. Then, since you have a different, you can have different types of molecules, so the input data from all these systems are different. So therefore the software has to cater to a variety of possibilities, so therefore rather than just 1, dot slash a.out you will have to do several operations, so that your data set is ready.

And so today we will discuss all the kinds of data sets, all the kinds of commands and finally we will also see how to install Gromacs. So my student and colleague Sonanki Keshri who has been working on this problem for many years. She will give us the detailed information about all these software. Now we will ask her to start introducing about the molecular dynamic simulation on liquid argon using Gromacs software.

So to perform molecular dynamic simulations of liquid argon or any other molecule, there are 4 basic input file that is required.

(Refer Slide Time: 02:17)



So the first file that is required is the molecular structure which is in format of gro or pdb file. That is the structure file of the molecule, pdb is the protein data bank file so which is easily available from protein RCSB site and we can as well create a pdb file also and the Gromacs input structure for pdb file is the gro file which we have to convert. The second file which is required in Gromacs for MD simulation is the topology file which contains the complete description of all the interactions in your system.

Basically, it contains the force field parameter like the Lennard-Jones potential, sigma, epsilon, charge and all. The third file that is required is the molecular dynamics parameter file. That is the mdp file which contains information about the molecular dynamic simulation itself that it contains the information about which is the algorithm which is required to simulate.

Or which is required to integrate the Newton's equation of motion, the total number of step, what is the temperature, what is the pressure that is required or what is the type of cut off you are using and all the stuff and the fourth file that is required is the itp file which is the include topology file it stands for and in that we write all the force field parameters. All the detailed information is given in the itp file.

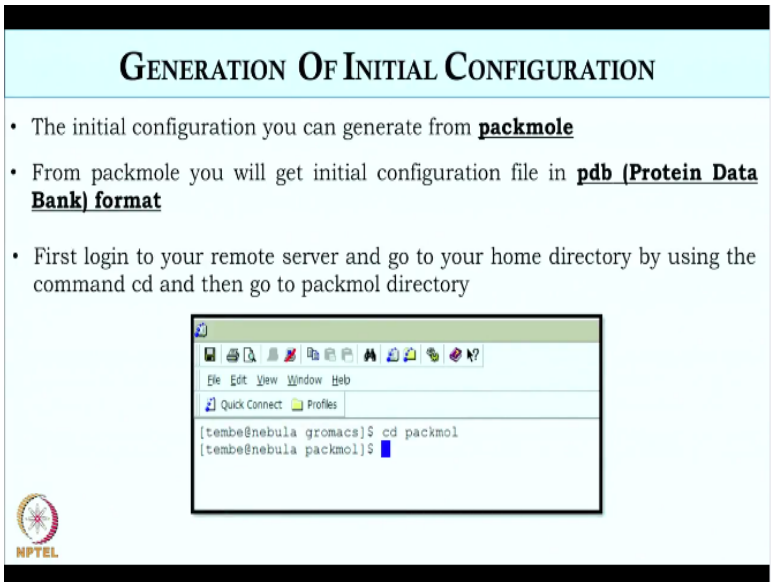
So can you comment on advantages of putting data in different files like that? Yeah, because we cannot put all the file in all the like pdb, top and mdp file in one single file. If you have different, different file, it is easier for a user to understand each and every file in detail like if

you have a separate gro file, we can know that what this file do, what this file contains and all thing, what is the information that is contained in a topology file.

And if we combine it together, it will be a huge file and then the user. And also do not you think that since these files are separate I can always change parameters in these files and then make my starting correct so. Yeah definitely. Keeping it in this different modular format allows user to change the parameters their own way, so they can have their own parameters other than what are given by the standard force fields.

Yeah, because anyway all these parameters we have to change according to each and every molecule. They will have different parameters, so all these are user-defined. Okay, yeah please go ahead. So you have basically pdb, top, mdp and itp file. Yes, these are the 4 input file that is required. Yeah okay. So now to perform an MD simulation, we have to have an initial configuration of the molecule of which MD simulation is to be performed.

(Refer Slide Time: 05:05)



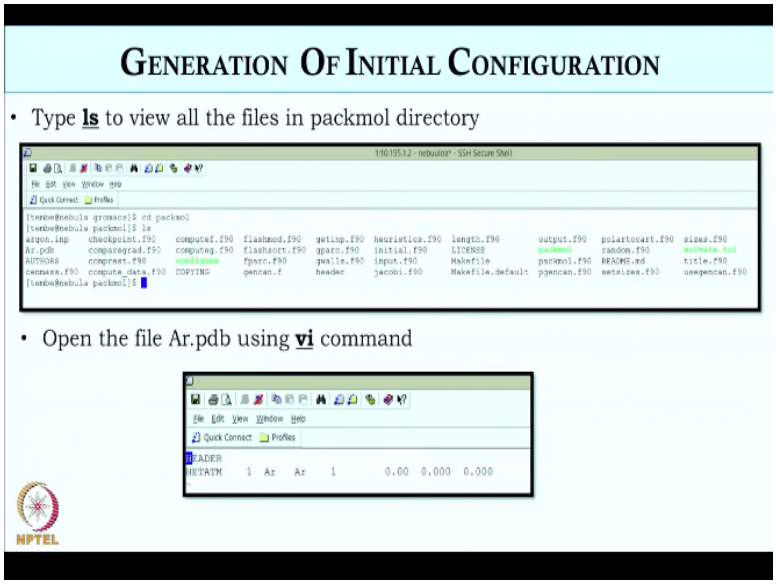
And to generate initial configuration, we use a software packmol. It is also a free software and one of the things I want to say that everything that we are using here are all free software, freely available and so that you do not have to pay anything, no license fees and these are very, very good softwares. In fact, Scilab that we discussed, they are using the Scilab software even to launch satellites.

You know launching a satellite is much more than doing the simple simulation in a lab. The simulation can break down but if a satellite breaks down, it will be a loss of huge amount of

money. So many of these softwares are really top world class softwares, so we should feel good that several groups in the world are really putting effort, so that all these softwares are available to the public.

Okay yeah, so now we are discussing how to generate initial configurations using a software Packmol. Yeah, so from packmol the configuration which we will get the format will be in a pdb format that is the Protein Data Bank format and so to first we have to login to our remote server and then we have to go to the Packmol directory using the command cd. So if I type cd packmol, it will go to your directory packmol.

(Refer Slide Time: 06:14)

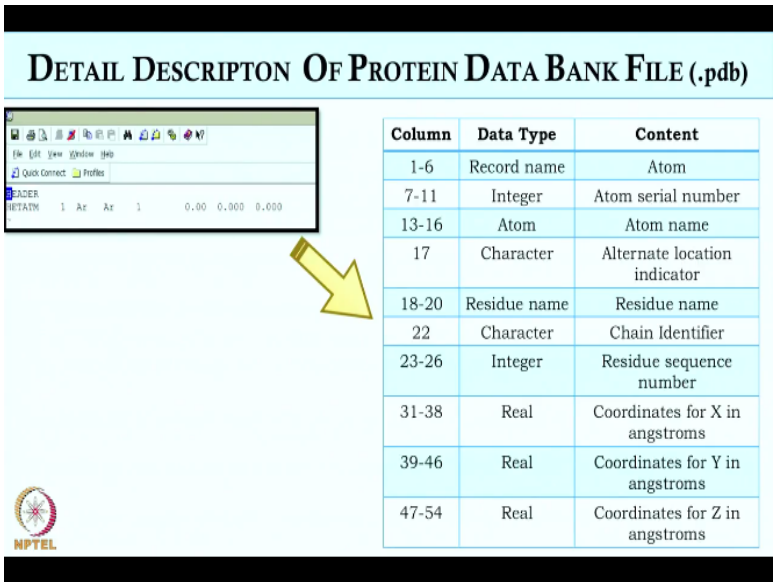


And we can type ls to view all the files that are present in the packmol directory. So you mean to say all these files now they are all from packmol directory? Yeah, when we install packmol, these are all the files that are already present in packmol. Okay. So there are two files that we have put in, first is the argon.pdb file which is the pdb file of liquid argon. So we can open the file using vi command.

So now we want to start with liquid argon because argon is one of the simplest fluids, something like water or CO2 or any molecular liquid will have several atoms in a molecule whereas argon is just a single atom, so if we have say 100 or 200 argon atoms, you just have to specify the coordinates of those 100 atoms. So atom simulation is far easier than molecule simulation, so we are starting with argon, so that you get a good feel for this.

Now lower part is the structure of the pdb file? Yeah, this is the structure of the pdb file. So I will get the detail of the pdb file in my next slide okay.

(Refer Slide Time: 07:17)



Column	Data Type	Content
1-6	Record name	Atom
7-11	Integer	Atom serial number
13-16	Atom	Atom name
17	Character	Alternate location indicator
18-20	Residue name	Residue name
22	Character	Chain Identifier
23-26	Integer	Residue sequence number
31-38	Real	Coordinates for X in angstroms
39-46	Real	Coordinates for Y in angstroms
47-54	Real	Coordinates for Z in angstroms

So this is the pdb file of liquid argon, so here we have given that in column 1 to 6, it represents the record name of the atom right. From column 7 to 11, it is a integer that is the serial number of atom since it is a monoatomic molecule, so there is only one atom like argon, so there is serial number there will be only 1, but if we have water or CO2 then it will have oxygen and 2 hydrogen sides, so there will be 3 separate line for each atom.

And column 13 to 16 is the atom name for each molecule. Since here we are dealing with liquid argon, so the atom name is Ar that is argon. So you mean to say that that Ar which is typed on the left side, it can be anywhere between 13 and 16? Yeah, it can be anywhere between 13 and 16. So you need not exactly start with 13, it can be 14 or 15 also. Okay. But it has to be in between 13 and 16. Okay, 17th is the character which is the alternate location indicator.

Then, 18 to 20 is the residue name since here it is just a monoatomic molecule, so here we will have argon as a residue name, it is user-specified. User can change the residue name according to their convenience. So when we will deal with water or CO2 or methanol molecules, then we will have a different residue name. For water, we will have say SOL as a residue name.

And then the atom name will be different from the residue name for a polyatomic molecule. So like H2O will be HOO things like that. Exactly. There will be 3 residues for water, maybe 5 for you know. No, no, there will be atom name for water. The residue name will be the same. The residue name is solvent or methanol, for in case of methanol it will be MEOH and there are 3 sides in methanol, one is CH3, one is oxygen and one is hydrogen.

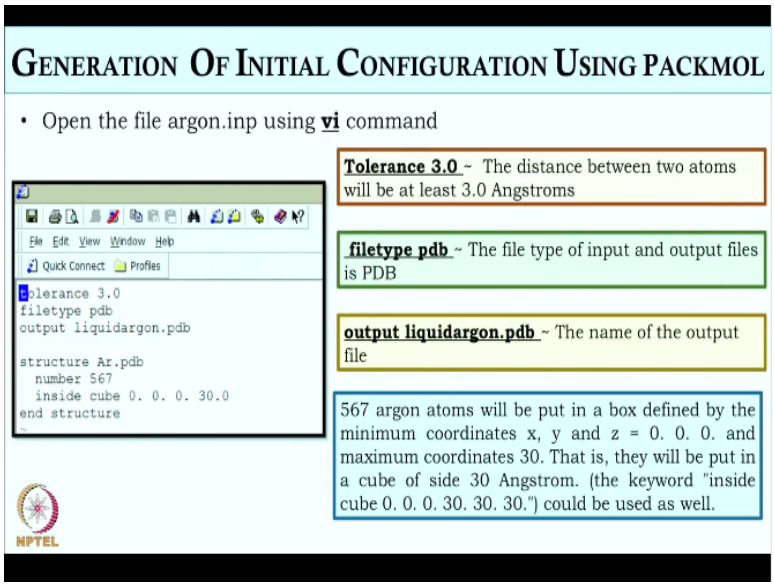
Though the atom name will be different but the residue name will be the same, it has to be the same throughout. Okay. So residue is like more like solute and solvent. Yeah, it is more like solute and solvent and it has to be for solvent there has to be 1 residue name or for solute there has to be 1 residue name, it cannot have different, different residue name for each and every side.

Then, column 23 to 26 is the integer which is the residue sequence number. Since here it is only monoatomic molecule, so the sequence number is 1. Then, column 31 to 38 is real type data and it is a coordinate for x axis in angstrom. Similarly, column 39 to 46 is the coordinates for y axis in angstrom and 47 to 54 is the coordinate for z axis in angstrom. So now you are saying that all these coordinates are given in angstroms?

Yes, in pdb. In pdb format? Yes. But Gromacs needs it in nanometers is not it? Yeah, it needs it in nanometers. So at some stage you have to convert. Yeah, when we will convert this pdb file into a gro file in Gromacs, then will convert it into nanometer. So now these coordinates are given as 000? Yes. Instead can it be like 222? Yeah, it can be, it is allowed, it can be 222. So I can give any, it can be any coordinate x y z.

Yeah, it can be any coordinate x y z. Since it is a monoatomic molecule, you can define it according to your convenience. Yeah and since our simulation boxes are of the size of say 30 to 40 angstroms, we expect these coordinates to be between 0 and 40. Is that correct? Yes. So I cannot give more than the length of the box? No. Okay yeah. So the 000 could be 111, 457, any combination? Yeah, any combination. Less than the box length. Less than the box length yes. Okay. Yeah, go to the next one.

(Refer Slide Time: 10:50)



Okay, so now to generate initial configuration using packmol, we have to have an inp file where we will specify the box length of the or which kind of box we are going to use in simulation and what is the coordinate of the box, how many molecules of liquid argon has to be there, what is the name of the output file (()) (11:04).

So even if there is no inp file to begin with you can create a new file argon .inp using the vi command. Yeah, we can create, no, no. We can download it in the packmol site. There are examples given there and which is freely available and you can download it. There they have commented each and every line like which line stands for what, so you have to have this inp file which you can get.

No, this inp is it in packmol or created by the user? No, no, it is created by the user. It is not in the packmol. Okay. And so this is the format of an inp file. So here the tolerance which is given as 3.0, it is the distance between two atoms that will be at least 3 angstrom. This is defined by the user. It can be 2, 3, 4 according to the size of the molecule. So what you are saying is that I have defined one coordinate using 000. Yes.

Now for liquid simulation, I have to generate a large number of coordinates. Yes. So to generate many, many coordinates, I need to make sure that the second atom is further from the first atom by a certain distance. Yeah. That is this tolerance. Yes, otherwise it will crash the simulation. Correct and we may have already discussed what is the Lennard-Jones potential. In that potential, there is a size parameter called sigma.

So typically tolerance should be of the order of sigma or larger. Yes. Okay. Depending on the size of the atom or molecule. Right. Then, there is file type pdb because in packmol we are using the pdb file format. Okay. And then there is output liquid argon.pdb, it is user-defined, user can give any name according to their convenience, according to the name like simulation, they are performing.

But this file type.pdb this is not there in your inp file? This is there in the inp file, file type.profile type pdb, it means that the file type. That the second statement. Yeah, there is a second statement. What it means is that the input file which we are providing like argon.pdb, it is in the format of pdb and the output which we will get it is in the format of a pdb file. So it is the file type of input and output file which is a pdb file. Right.

And then output is liquid argon.pdb, it is a name of the output file. User can change it according to their convenience or the molecule which they are going to simulate. Since here we are simulating liquid argon, so I have given the file name as liquid argon.pdb. Okay. And then there is structure argon.pdb. It is the name of the pdb file that we are using and there is number 567 that is the total number of argon molecule in the simulation box.

And what is meant by inside cube 0.0.0.30 is that it is a cubical box of length 30 angstrom in x y z axis and instead the keyword 000 30.0 we can as well use 0.0.0 30.30.30 to specify each x y and z axis and then you have to end this file using the command end structure. So what does the structure mean there? Structure is the input file that you are providing that is argon.pdb.

When you will simulate water, then you have to use water.pdb or the pdb file of water molecule or the pdb file of the molecule which you are going to simulate. Here we are doing simulation of liquid argon, so we have to provide the pdb file of argon molecule. So maybe the software know that when you say structure Ar.pdb, it starts that file containing all the coordinates. Yes. It generates all the file 67, then it ends that.

Probably, is that the possibility? Yes, it will go to the argon.pdb file then it will take another since we are giving here 567 molecules, so it will take another argon molecule and the distance between them will be minimum 3 angstrom since we have specified it and then it

will fill the box of, fill the cubical box of 30 angstrom by 567 molecules. So I think for the learners you have seen now that there are 567 atoms in a box of size 30 by 30 by 30.

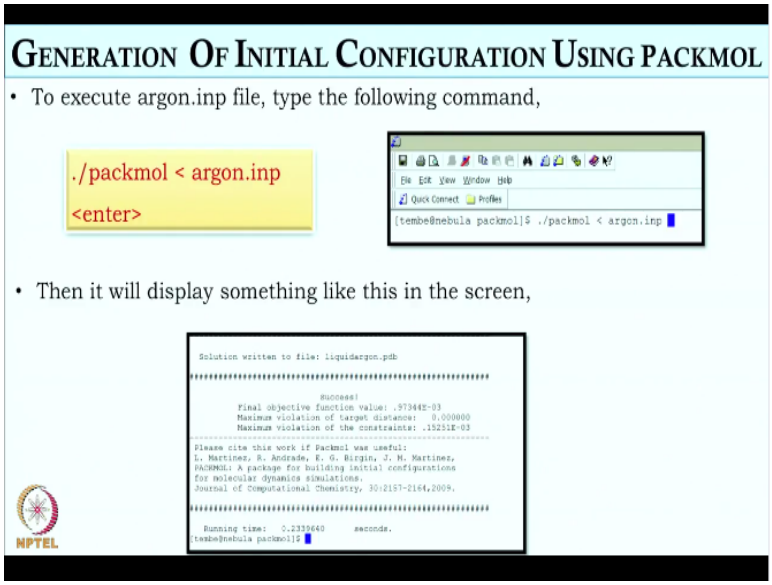
So using this 567 you can find out what is the mass of these 567 atoms and this much mass is containing a box of 30 by 30 by 30, take density at density is mass/volume, do this mass/volume and see whether the numbers you get corresponds to density of liquid argon . So we have chosen this 567 so that you get the density of liquid argon correctly which is say around 1.39 grams per cc.

Right, so you please check that this uses the correct density of liquid argon because if the density is wrong, all your simulations will be wrong. So the first thing when you do any simulation, you should identify the number of molecules in your box. We have chosen 30 because it is a good number. So even 25 would be good, 35 will be good but suppose you use 100 by 100 by 100 just calculate how many argon atoms you will have.

And the argon atoms will be so many that you will not finish the simulation in several days. Is that right? Yeah, it is right. Okay, so. It depends on the box length and number of molecules as well. So nowadays typical box say about 50 years ago or 30 years ago, we would use box length of 15 or 20 angstroms. Today 30 and 40 is common, maybe after another 10 years they may all be using 100 angstrom because your computers are much faster and more memory is available.

Even 70 angstrom box people have used. Okay, right, okay yeah please go ahead.

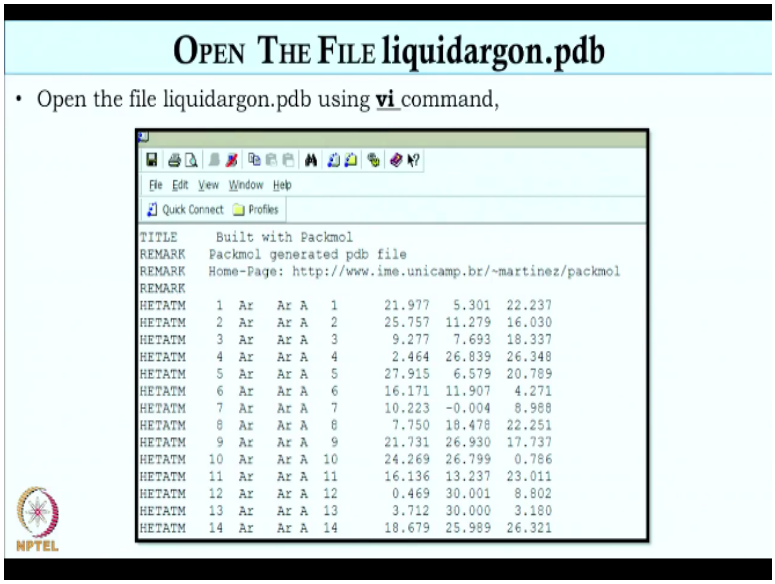
(Refer Slide Time: 16:41)



So now to execute the file argon .inp, we have to just type the command dot slash packmol space argon .inp and then enter. So this is the command that we have to use to execute the file and to get the output liquid argon.pdb and then after performing it, it will display something like this in the screen that success and it is built in this much time is used to build the file liquid argon.pdb.

So this is the result of your execution what is shown below? Yeah, this is the result of my execution. So how many lines will be there? In liquid argon.pdb in the output file? Yeah.

(Refer Slide Time: 17:18)



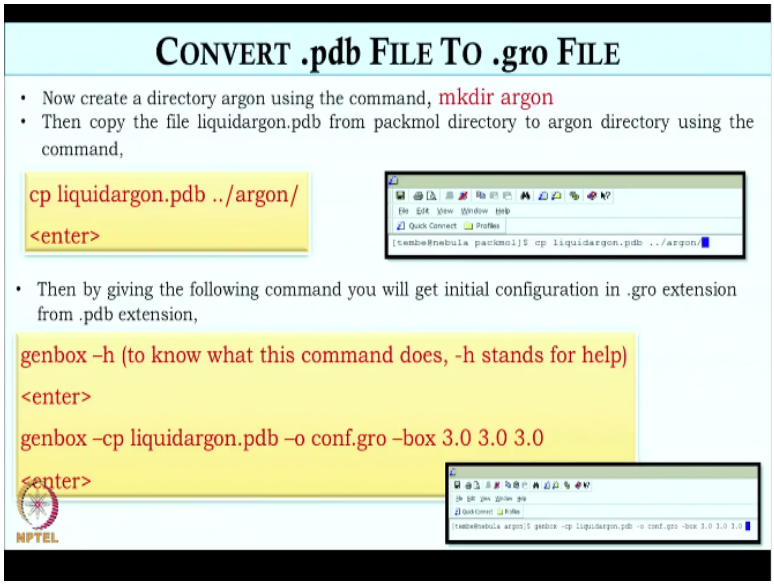
The output file will have, so to open the output file we will use vi command and this will be the output file. Since we have taken 567 argon molecule atoms, so it will have 567 argon atoms in the liquid argon.pdb file. Here I have just shown 1 to 14 and the last 3 columns are

the coordinates of x y and z axis in angstrom and it will similarly have more 560 lines. Total 567? Total 567 lines since we have 567 atoms.

And you should all see that none of the coordinates is more than 30 30 30 right. That means all these atoms are inside that cubic box. Is that correct? Yeah, because the box length is 30 angstrom, so it cannot have coordinates more than 30. Correct, now suppose for this box size 30 by 30 by 30, I had given 6000 atoms rather than 567, will the program execute properly? No, the program will not execute properly and it will crash the simulation.

Because you cannot fit yeah, you cannot fit so many in this small box. Yeah, it will explode. Right, so it is like putting 1000 people in a classroom of 100 seats. Exactly. Okay. So now we have a pdb file for the liquid argon. Yes, now we have a pdb file which is the first input file which is required. Right. So now we will take this pdb file and will convert it to gro file because in Gromacs the input file which is taken in Gromacs is the .gro file and not the .pdb file.

(Refer Slide Time: 18:55)



So first will create a directory using the command mkdir. Since we are simulating liquid argon, so we will create a directory mkdir argon. Now we will copy the liquid argon.pdb file from packmol directory to argon directory using the command cp, cp liquid argon.pdb to the directory argon. So we will use this command. So what do those 2 dots mean? Does it mean it goes to a higher directory or?

The two dots mean that it is going out from the directory packmol. Right. And then it is going to the directory argon. Right, so the directory argon is at the same level as the packmol? Yeah. Okay. So now to convert the pdb file into a gro file, we will type the following command Gromacs command that is genbox. So if we type genbox -h, so we can know that what this command does, -h it stands for help.

It will display several things where it shows that which is the input file that is required, which is the output file that we need and it will give detail of the command and how it was. If whatever the genbox does there will be many, many options is not it? Yeah, that there will be many options. So all those will be shown okay. So what we will do? Today we are going to describe all the commands and after one or two days when all the commands are described we will actually execute it on the computer.

So that you would have heard all the details and then you will actually execute, so that whatever you said you actually see in real life okay. So then we will type the command genbox –cp liquid argon.pdb which is the pdb format we have got from packmol. Right. Then –o conf.gro, -o stands for output file and conf.gro is the configuration file conf.gro then –box 3.0 3.0 3.0, –box is the box cubical box and 3.0 3.0 3.0 which in nanometers. It is in nanometers.

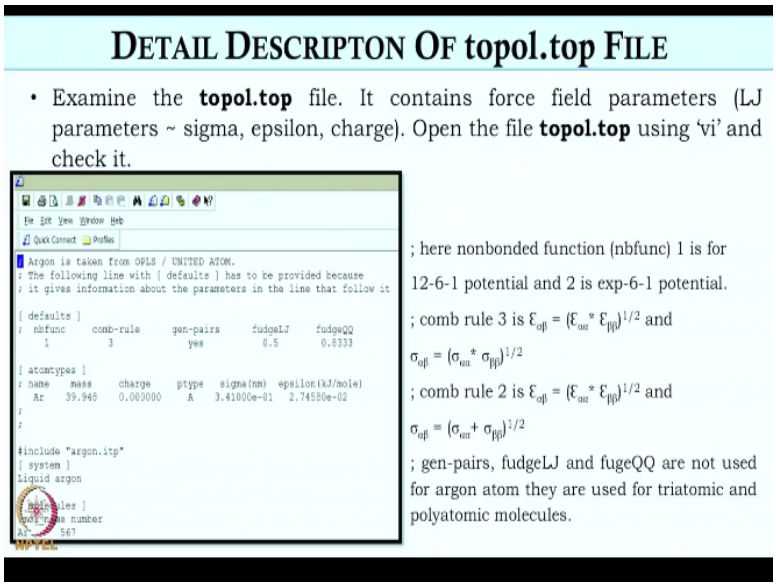
Since packmol we have used 30 angstrom, so here in Gromacs all the distance, the unit of distance is in nanometer. The energy units will be in kilojoule per mol. So earlier we had 30, 30, 30. Yes, now we have 3, 3, 3. That is in nanometer. Right. Can this be instead of 3, 3.3 or will it give an error? It would not give an error; it can be 3.3. But it cannot be<3. No, it cannot be<3.

Okay, so if you instead of 3, suppose it is 3.5 then there will be more space than what is required for the current density. Exactly. So you will do a simulation at a lower density than what you started with. Exactly. But if you do 2.5 2.5, it will give an error because it is not fitting in. It is not fitting in and there might be some argon atoms which will have coordinates of 2.5. Okay.

Maybe because the box length we have given is 30 angstrom, so some of them may have coordinates 2.5, 2.6. Right. So it will not take. Okay fine. So it can be maximum the box

length or greater than that. Okay. The box length. Right. Now you have got a gro file. Yes. Okay, so what is the next step? So we have got the gro file. Now if we can view the gro file using the command vi. Okay. It will show the coordinates of argon atom in nanometer. Right.

(Refer Slide Time: 22:16)



Then, we will need a topology file which will contain the force field parameters that is the Lennard-Jones parameters, sigma, epsilon, charge, mass. So all these are from the Gromacs force field? No, these are from OPLS force field and OPLS force field when we install Gromacs, there is a top folder that is installed automatically after successful installation on Gromacs.

And in that top folder, we will have all the force field parameters Lennard-Jones parameters, sigma, epsilon, charge. There will be a file folder of each and every force field parameters like for OPLS force field there is a separate directory. Okay. Then, for AMBER force field there is a directory, for CHARMM force field there is a directory, for GROMOS there is directory.

So now suppose I am using OPLS force field, yes and after some time I decide to use CHARMM force field. Okay. So what do I have to do for that? So you just go to CHARMM, they will be in top folder. Okay. You will have all these directories, so you just go to CHARMM directory, there you will have ffbonded.itp and ffnonbonded.itp file. Okay. And since in all like if you use OPLS force field, we will have a different set of parameters.

If we go to CHARMM force field, we will have some other set of parameters. Okay. So we just have to change the parameters. We just have to change the charge, sigma or epsilon values. So you do not have to reinstall if you have to use different force fields? No, we do not have to reinstall it. We just have to change the parameters. Okay. We just have to go to the specific directory.

Yeah. We just have to see what is the sigma or epsilon value in OPLS force field and in CHARMM force field and we have to change it accordingly. Okay. So now, and if we knew those parameters also we can change by hand also in tha pdb file? No, it has to be done in the topology file. Okay yeah. So it is user-specified, user can change it according to the force field they are using.

So this topology file is a normal file not a binary file. No, it is a normal file. It can be viewed using the vi command. So which are the binary files? The binary files which will be created after we run the energy minimization command or MD run which will come later. So what is the extension of that binary? It is, the extension is tpr. Okay. So other than tpr are all other files normal text files?

Tpr and trr are the binary files which can be viewed using a different command. Okay. Not by using simple command. Not the vi command. Yeah, not the vi command. Yeah. So you can open the topology file using vi command. So this is the topology file of argon atom. So here we will see that there is a nb function defined which is defined as 1. So nonbonded nb function stands for.

Is that semicolon for a comment? Yeah, the semicolon is for comment, so which means the first 3 lines from the left are all comments. Yeah, the first 3 lines are comment. Okay. So whereas the default it set in and below default there is nb function, nb function is nonbonded function. Okay. And it is 1. Okay yeah. So 1 stands for the type of potential model that we are using.

So here we will be using 12 6 1 potential that is the Lennard-Jones plus Coulomb potential. Okay. There are other potential models as well, so we have to change it accordingly okay. So 1 will refers to Lennard-Jones? Yeah, 1 will refer to Lennard-Jones. Okay and that 3 second

one? Second one is the combination rule. In Lennard-Jones potential, we have sigma and epsilon right.

And the combination rule 3 is if the sigma if epsilon a b is a geometric mean of epsilon alpha, which means that combination rule is if we have different types of atoms? Yes, if we have different atoms. If we have single atom like argon then it does not matter? No, it does not matter. Because all atoms have the same sigma. Exactly. So there is nothing to do okay. Yeah. And what is the next one?

The next one is generate pairs that will it is not used in case of a monoatomic molecule but these are used in case of polyatomic molecule. When we come to water and methanol and other complicated molecule, we will have to use generate pairs. It will generate pair of interactions. Okay. And then we have atom types. In atom types, we have to define. And what are those other two lines fudge?

FudgeLJ and fudgeQQ, these are not used for argon atoms. These are used for triatomic and polyatomic molecules. Okay. Then, we have atom types. In atom types, we have to define the name of the atom like here it is argon. Right. Then, we have to define the mass of argon atom. Okay. Then, charge, charge is 0. Then, there is p type that is paired types. Then, there is sigma and epsilon.

Sigma is the distance, so it is in nanometer and epsilon is the energy that is the depth of the potential well it is in kilojoules per mole. Okay. The value of sigma and epsilon and charge will get it from the ffbonded and nonbonded.itp file which is in the OPLS directory. Right. So now after that you are including an itp file? Yeah, after that I am including an itp file argon .itp.

Right. And then I have to also specify the total number of argon atoms that is being simulated. Okay. And the residue name. Okay. So the residue name that we have already given in the pdb file, we have to use the same residue name and then we have to give the total number of molecules that are being simulated. Here we are using 567 argon atoms. Okay. So we have to specify it. Okay. So, so far we have considered pdb file, gro file and top file, is not it?

Yeah. So now what we will do, we will conclude this discussion now and next time we will start the remaining files. Which are the remaining files now? The remaining files are itp file and the mdp file that is the molecular dynamics parameter file. Okay. So we will conclude this lecture here and the next one we will summarize what we have done so far and then take up the remaining files and then show how to execute the Gromacs package. Thank you.