BT5420 – Computer Simulations of Biomolecular System

Assignment 2

Submitted by Sahana G (BE17B038)

Question 1

How to build the missing hydrogen atom co-ordinates in GROMACS? Give the syntax and the flag indicating its usage.

We first remove H2O before everything, because we don't want those extra water molecules in our computation. More molecules mean higher computation.

GROMACS includes hydrogen coordinates that is not present in PDB. Sometimes, hydrogen could be present and sometimes, it might be absent in the PDB file. The pdb2gmx command, while converting the file to .gro extension, with a flag (-ignh) ignores H atoms in PDB, and the GROMACS commands adds H atoms automatically when it converts PDB to gro.

```
gmx pdb2gmx -f 1aki_nowaters.pdb -o 1aki.gro -ignh
```

The flag used is -**ignh.**

Question 2

What is the difference between the input PDB file and the output GRO file generated from step 1 using `gmx pdb2gmx`. Is there any change in the co-ordinates? (mention the units wherever necessary)

In GRO files, hydrogen atoms are absent. Therefore, GROMACS automatically fills hydrogen coordinates for you. In PDB file, all the atom's coordinates are present.

The GRO file that is created has a very compact format, but it also has limited precision.

The purpose of pdb2gmx is to produce a force field-compliant topology; the output structure is largely a side effect of this purpose and is intended for user convenience. GROMACS can handle many different file formats, with GRO simply being the default for commands that write coordinate files.

In **PDB** – The units are given in **Angstroms** and in **GROMACS** – The units are **Nanometres**.

Question 3

Solvate your system using `gmx solvate`.

The 1aki.pdb file has crystal waters in it. To reduce the time for simulation, water crystals were removed in VMD application using the following command in Tk console.

```
set a [atomselect top "protein"]
$a writepdb "1aki_nowaters.pdb"
```

These commands save a PDB file named '1aki_nowaters.pdb' after removing all water crystals from 1aki.pdb.

After doing this, when you create the GRO file for the protein, it reports the box's dimensions at the end.

(a) Report the box length along X, Y, Z needed for solvating your system.

```
gmx pdb2gmx -f 1aki_nowaters.pdb -o 1aki.gro -ignh
```

Choose CHARMM27 and then TIP3P for water model. This will create the required GRO file.

Box size -

5.90620 6.84510 3.05170

You'll notice that the protein is not centred in the box. For this purpose, use the following command –

```
gmx editconf -f 1aki.gro -o 1aki_box.gro -c -d 1.0 -bt cubic
```

Here -c : centring the protein, -d : padding distance – set at 1 nm, and -bt : box type – set as cubic.

Once, this command is executed, the box size then changes to:

Box size -

7.01008 7.01008 7.01008

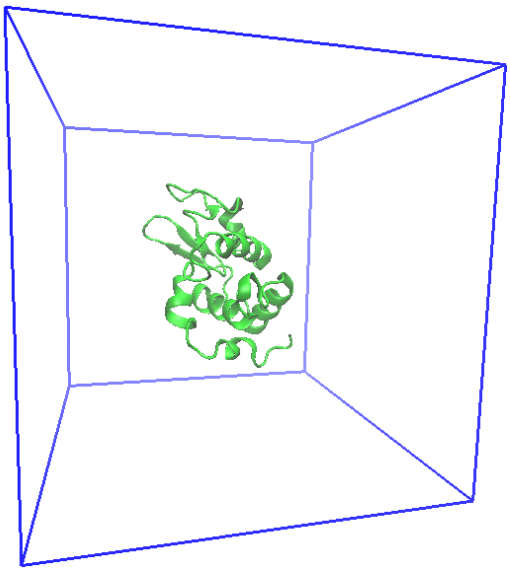
Command for solvating the system -

```
gmx solvate -cp 1aki_box.gro -cs spc216.gro -o solv.gro -p topol.top
```

b) Can we visualize the PBC box in VMD. If yes, state the command.

In the VMD application– Open solv.gro using the 'Load new molecule' option. Go to Graphics -> Representations -> Newcartoon

Click on Extensions -> Tk Console, and type the following commands – **`draw pbcbox`**



Question 4

What is the resname identifier for the added ions in your system during neutralization step `gmx genion`? Show using topol.top screenshot.

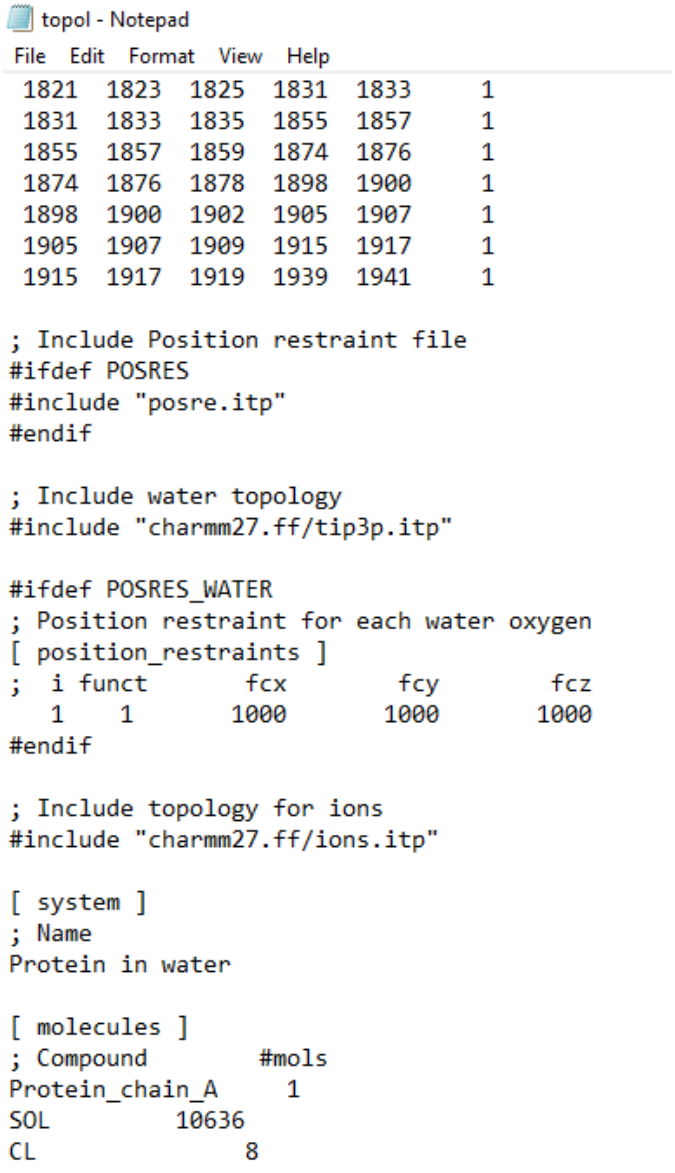
The last line in your [atoms] directive of TOPOL.TOP file. – qtot = 8

For neutralizing these charges,

```
gmx grompp -f ions.mdp -c solv.gro -p topol.top -o ions.tpr
gmx genion -s ions.tpr -o ions.gro -p topol.top -nname CL -pname NA -neutral
```

Choose 13 (SOL) -> which means, that we are replacing corresponding SOL atoms with CL/NA based on qtot.

Since my protein has 8 positive charges, to neutralise it GROMACs had added 8 Chloride ions. It is represented with RESNAME - CL



Question 5

Perform energy minimization of the neutralized system using minim.mdp

Direction –

```
gmx grompp -f minim.mdp -c ions.gro -p topol.top -o em.tpr
gmx mdrun -v -deffnm em &
```

a) Report the number of steps required to converge potential energy

Overall, it took 700 steps to converge to minimal potential energy, since the variation between energy corresponding to steps in 600-700, didn't vary much, the process converged withing 700 steps and not the assigned 50000 steps.

b) Plot the potential energy curve using plotting tool.

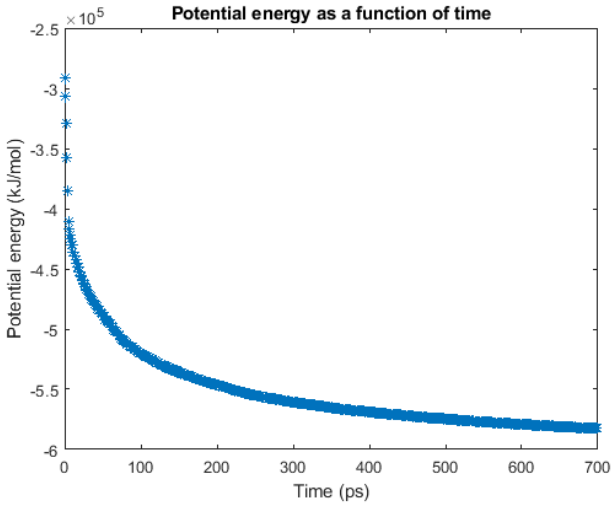
The data was stored in pote.xvg using gmx energy command

```
gmx energy -f em.edr -o pote.xvg
```

Following this, the pote.xvg file was parsed to obtain only the values of time and energy. The data was plotted using MATLAB.

Code –

```
% Generate Potential energy plots
%Load the table.

pe = pote{:,:};

time = pe(:,1);
energy = pe(:,2);

plot(time,energy,'*');
title('Potential energy as a function of time');
xlabel('Time (ps)');
ylabel('Potential energy (kJ/mol)');
```

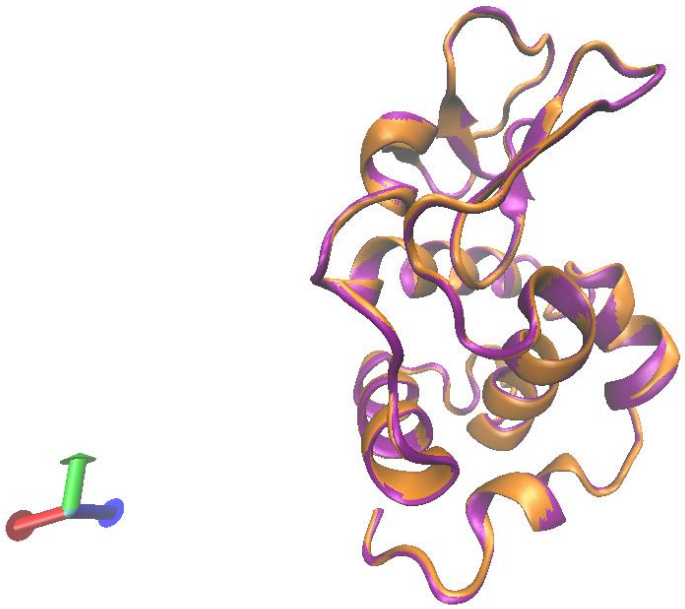


c) Load both the input and output .gro files used in minimization step in VMD. Represent in New Cartoon in different colors. Comment on the difference observed in their structure.

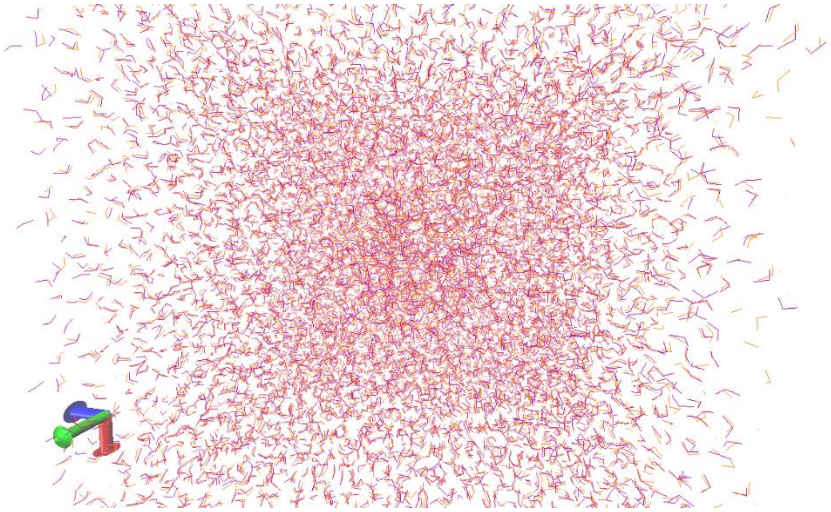
Input GRO file – ions.gro (coloured – purple)

Output GRO file – em.gro (coloured – orange)

Not much difference is observed in the main chain of the protein's main chain. Also, since the NewCartoon representation only shows us the main chain of the protein, this does not allow us to observe difference in positions of the solvent ions with respect to each other in both the GRO files.



To observe if there is any change at all between both the GRO files were represented as Lines.



This image shows that there are some deviations between the solvent ions of both the files, however, no significant change is observed within the main protein chain.

Question 6

State the syntax and purpose of a checkpoint file.

Syntax = -cpt

```
grompp -f new.mdp -c old.tpr -o new.tpr -t old.cpt
mdrun -s new.tpr
```

-cpt means check point file which restarts the simulation from previous run.

If gen_vel=yes, then system will again use a random velocity. This will take a longer time to equilibrate.

If a simulation crashes, make use of the state.cpt file that is written; it contains all of the information necessary to continue the simulation. In order to pick up from where the simulation stopped, simply use the -cpi and -append options to mdrun.

Question 7

Why is it necessary to use atleast 1 nm padding for water during solvation? What do you think would happen if we use 0.5 nm and 2 nm padding?

The hydration layer around a protein has been found to have dynamics distinct from the bulk water to a distance of 1 nm. Hence a padding distance of 1 nm is necessary for water during solvation. Also, minimum image convention is satisfied when the padding distance is set as 1 nm. That is, a protein should never coincide or meet with its periodic image; if it happens, the energy calculations will be erroneous. Specifying a solute-box distance of 1.0 nm will mean that there are at least 2.0 nm between any two periodic images of a protein. This distance will be sufficient for just about any cut-off scheme commonly used in simulations.

Padding of 0.5 nm can cause artifacts with its neighbouring image, while padding of 2 nm is a pretty big box that might demands computational cost and time. Hence padding of 1 nm is appropriate.

Question 8

What is PBC and why is it necessary to remove PBC before analysis?

PBC stands for Periodic Boundary Conditions. This is the box that holds 1 molecule of the protein. Multiple such molecules exist in the system and each molecule has its own PBC box. PBCs are unit cells used to represent the dynamics and properties of a small individual molecule which can be replicated in other cells. When any atom crosses the PBC, the neighbouring atoms enter the unit cell to maintain constant NPT or NVT ensemble.

It is important to remove a PBC box because, during the simulations, the atoms tend to move. But the PBC doesn't move along with it. When only a part of the whole protein is present in one box, it also affects the whole dynamics of the entire system and as a result, the calculations are erroneous. Hence it is advised to remove the PBC box from the trajectory (follows the motion of the atoms). If analysis is performed without removing PBCs, the molecules might remain broken and yield erroneous results. This is done by the following command –

```
gmx trjconv -f md.xtc -o md_nopbc.xtc -s md.tpr -pbc mol -ur compact
```

Question 9

Plot the temperature graph from 100 ps of NVT simulation. Provide screenshot of the syntax used and the average temperature printed in the terminal.

Directions –

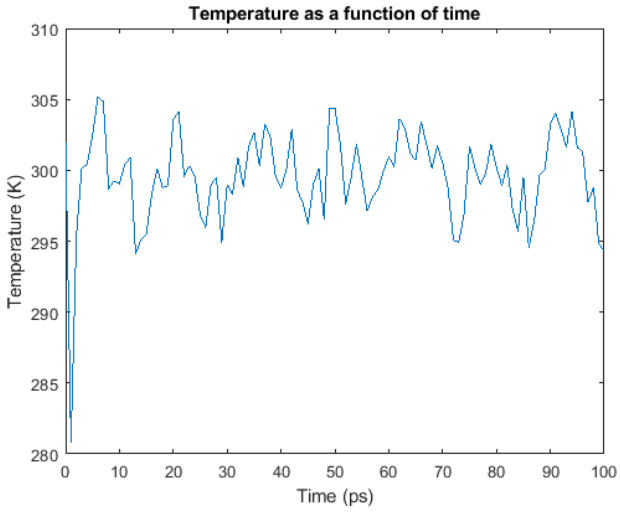
Run the following commands and choose (16 – Temperature) to be tracked.

```
gmx grompp -f nvt.mdp -c em.gro -r em.gro -p topol.top -o nvt.tpr
gmx mdrun -nt 8 -v -deffnm nvt
gmx energy -f nvt.edr -o temp.xvg
```

Following this, the temp.xvg file was parsed to obtain only the values of time and temperature. The data was plotted using MATLAB.

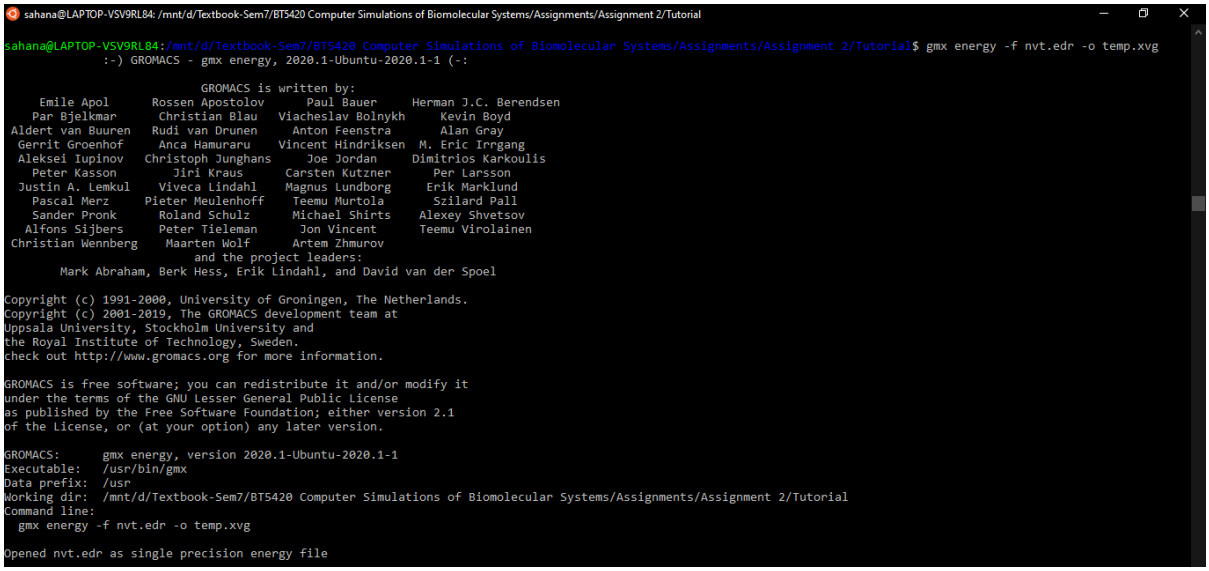
Code –

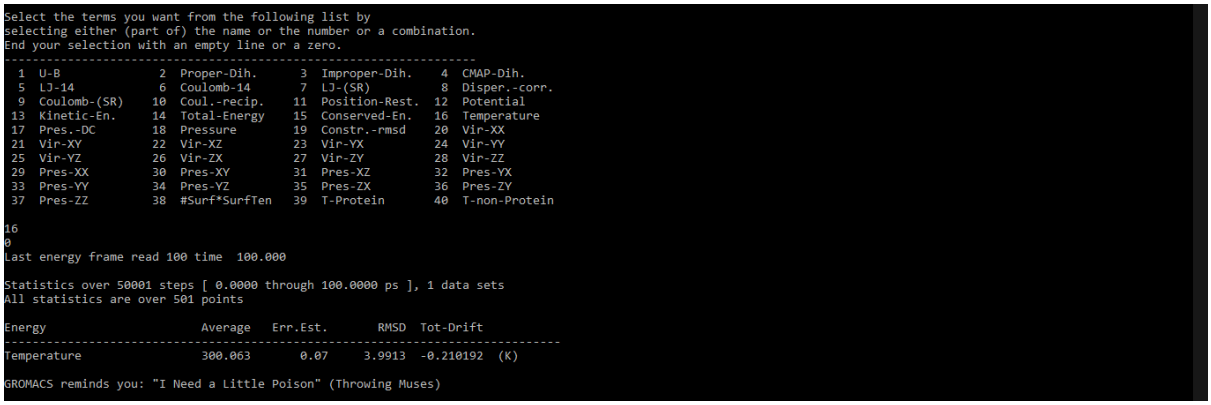
```
% Generate Temperature plots
%Load the table.

Temp = temp{:,:};

time = Temp(:,1);
T = Temp(:,2);

plot(time,T);
title('Temperature as a function of time');
xlabel('Time (ps)');
ylabel('Temperature (K)');
```



It is observed that the overall temperature fluctuates around 300K.

The screenshots for the code are provided here –





Question 10

Perform NPT for 150 ps using force constant of 1000 kJ/ (mol nm2). Provide screenshot of the mdp file highlighting the change in run parameters.

For a total of 150 ps, the npt.mdp file should be modified.

; Run parameters
integrator = md ; leap-frog integrator
nsteps = 75000 ; 2 * 75000 = 100 ps
dt = 0.002 ; 2 fs

The time steps are kept constant and the total number of steps are increased. The force constant is set as 1000 already in the posre.itp file. Therefore, no change needs to be done.

Directions –

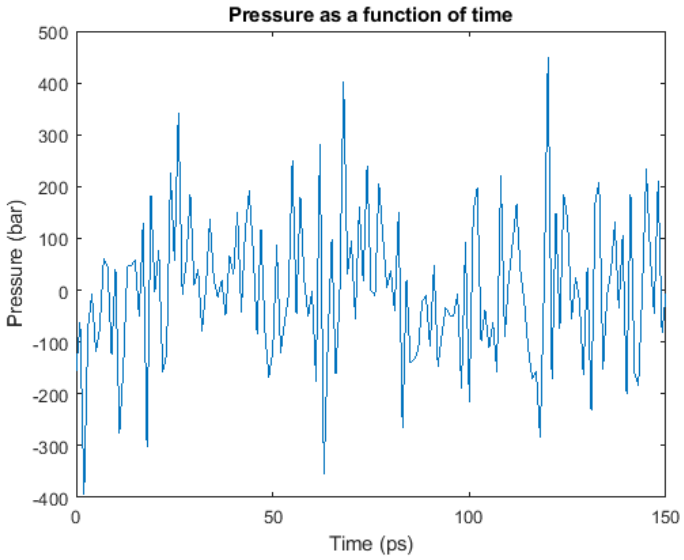
Run the following codes in GROMACS –

```
gmx grompp -f npt.mdp -o npt.tpr -c nvt.gro -r nvt.gro -p topol.top
-t nvt.cpt
gmx mdrun -v -deffnm npt &
gmx energy -f npt.edr -o pressure.xvg
```

Following this, the pressure.xvg file was parsed to obtain only the values of time and pressure. The data was plotted using MATLAB.

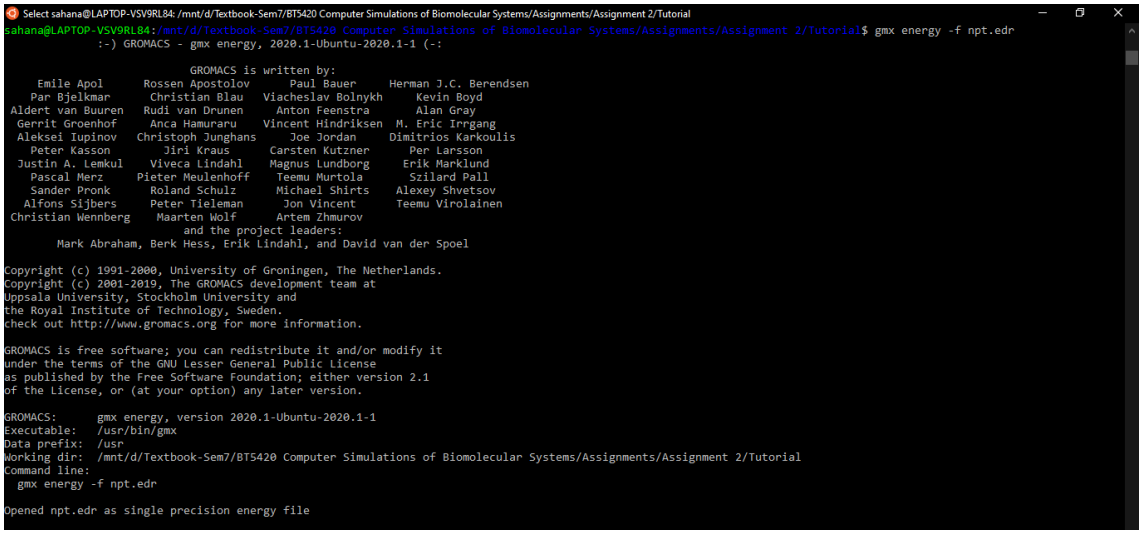
Code –

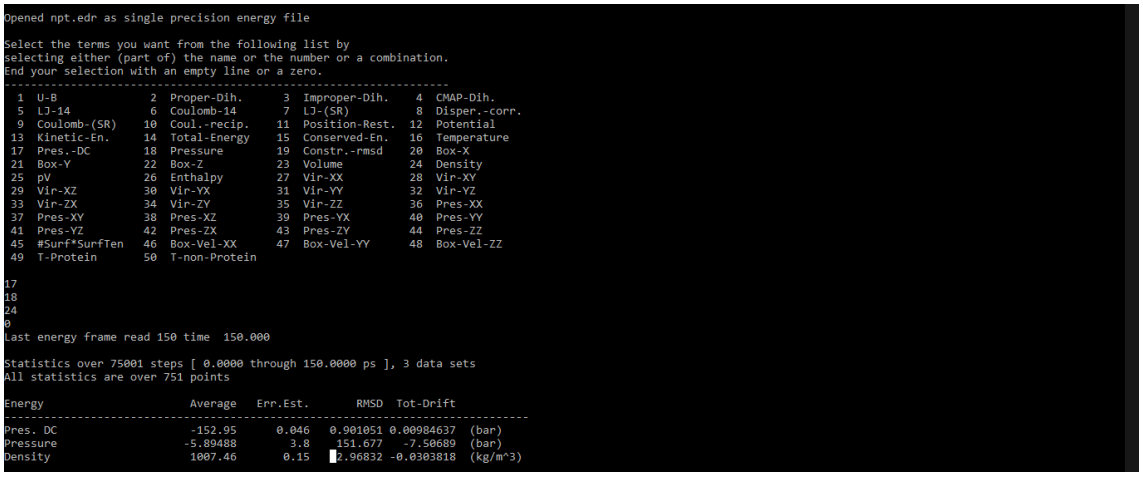
```
% Generate Pressure plots
%Load the table.

Pres = pressure{:,:};

time = Pres(:,1);
P = Pres(:,2);

plot(time,P);
title('Pressure as a function of time');
xlabel('Time (ps)');
ylabel('Pressure (bar)');
```



It is observed that the average pressure in the system is -5.89488 bar. However, since the RMSD ranges to 151.677 bar, the overall pressure of the system is greater than 0.

The screenshots for the code are provided here –





Question 11

Perform MD run for 1 ns.

Directions –

```
gmx grompp -f md.mdp -o md.tpr -c npt.gro -r npt.gro -p topol.top -t npt.cpt
gmx mdrun -v -deffnm md &
gmx energy -f md.edr
```

Here, choose to analyse the temperature, density and the pressure of the system by specifying appropriate numbers at the GROMACS.

a) Remove PBC from the trajectory. State the command used.

As mentioned earlier, it is important to remove PBC from the trajectory. The command used to achieve the same is –

```
gmx trjconv -f md.xtc -o md_nopbc.xtc -s md.tpr -pbc mol -ur compact
```

Here, choose (0 – System) for output as mentioned in the online tutorial link.

b) Report the distance between N and C terminal atoms of the trajectory (without PBC) as a time vs distance plot (gmx distance)

Directions –

Use the gmx command to compute the distance between both the atoms. Here, an index file is created for the atoms of interest, i.e., N terminal of first residue and C terminal of end residue. From the md.gro file, the atom's identifier/number is identified as 1 and 1958 respectively. Hence, a new file is generated – named as dist_inde.ndx which has the atom's index specified.

Once that is done, run the following command –

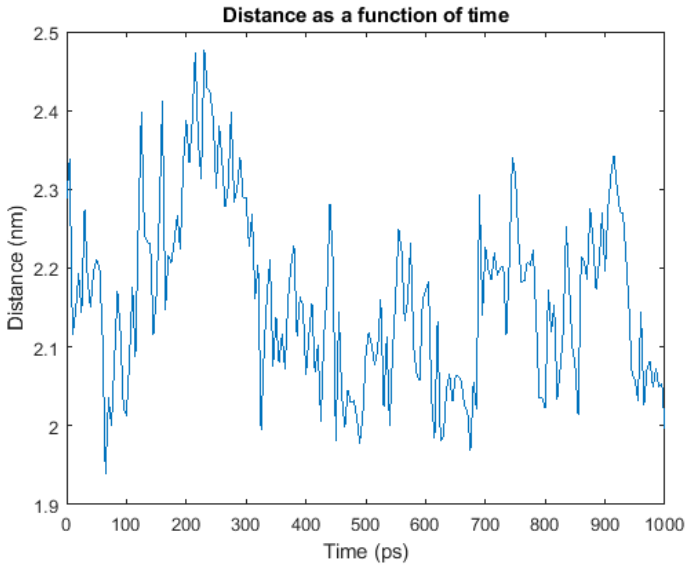
```
gmx distance -f md.xtc -s md.tpr -oall dist.xvg -n dist_inde.ndx
```

The program will ask you to specify the number of groups for which you would want to identify distance. Since we have only 1 pair, we enter 0 (denoting the basal/first value).

This will save the corresponding distance between the N and C atoms throughout the trajectory of our simulation. Following this, the dist.xvg file was parsed to obtain only the values of time and distance. The data was plotted using MATLAB.

Code –

```
% Generate Distance plots
%Load the table.

dist = distNC{:,:};

time = dist(:,1);
D = dist(:,2);

plot(time,D);
title('Distance as a function of time');
xlabel('Time (ps)');
ylabel('Distance (nm)');
```



It is observed that the average distance between the atoms of interest is 2.16319 nm with a standard deviation of 0.11380 nm.

c) How did you generate the index file? Discuss your observation on the distance plot.

The index file for the two atoms specified in the question was created manually. A new notepad was opened and the following content was typed in.

The content within the square bracket is called as the directive. The atoms correspond to the N terminal of first residue and the C terminal of last residue in the protein's mainchain.

However, you can also automate the process of creating the index file. The command is as follows:

```
gmx make_ndx -f md.gro
```

The program will analyse all the groups and sub-groups of atoms present in the md.gro file. For eample, if you would like to create an index file for all atoms in the protein, then mention

>keep 1, followed by **>q,** to save and exit.

This command can be used to create the index file of all the atoms that are present in the md.gro file. The created file has a .ndx extension and the content looks like this –