**BT5420**

# Computer Simulations of Biomolecular Systems
# Assignment 2

Shreeharsha G Bhat | BE21B037
Department of Biotechnology
Indian Institute of Technology Madras

**Question 1**

**How to build the missing hydrogen atom coordinates in GROMACS? Give the syntax and the flag indicating its usage.**

To reduce computational complexity, we first remove all water ($H_2O$) molecules. We do not consider them in our computation. Command for which is

grep –v HOH 1aki.pdb > 1aki_nowater.pdb

Hydrogen coordinates that are not present in the PDB are sometimes represented in GROMACS. The hydrogen atoms could be present or absent in the PDB file. The command **pdb2gmx** command, while converting the file to .gro extension, with the flag -ignh ignores H–atoms in PDB.

The GROMACS command adds the H atoms while converting the PDB to gro, automatically.

gmx pdb2gmx –f 1aki_nowater.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)?**

GRO files do not contain the Hydrogen atoms, GROMAC automatically files those coordinates for you. The PDB Files contain the coordinates.GRO has limited precision, but has a very compact form.

The pdb2gmx produces the force-field complaint topology, the output structure is largely a side effect of this purpose and is intended for the user's convenience. GRO file format is the default for commands that write coordinate files.

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

---

**Question 3**

**Solvate your system using gmx solvate.**
First we remove the HOH, crystal water molecules to reduce the time for simulations.
The command used is

grep –v HOH 1aki.pdb > 1AKI_clean.pdb

This command saves the PDB file named '1AKI_clean' after removing all the water crystals from 1aki.pdb

**a) Report the box length along X, Y, Z needed for solvating your system.**
Convert the pdb file to GRO file, this will report the box's dimensions at the end

gmx pdb2gmx –f 1AKI_clean.pdb –o 1AKI_processed.gro –water spce

**Box Size:**
**5.90620 6.84510 3.05170**

The Protein is not centered in the box, we now center it using the command –

gmx editconf –f 1AKI_processed.gro –o 1AKI_box.gro –c –d 1.0 –bt cubic

Padding distance set at 1 nm with a cubic box set

The box size is now:

**Box Size -**
**7.01008 7.01008 7.01008**

Finally to solvate the system we run the command –

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**
VMD Application→Load Molecule(open solv.gro)→Load.
VMD Application→Graphics→Representation→NewCartoon

Extensions → Tk Console

Entre command :

```
Main console display active (Tcl8.5.6 / Tk8.5.6)
(VMD) 1 % draw pbcbox
>Main< (VMD) 2 %
```

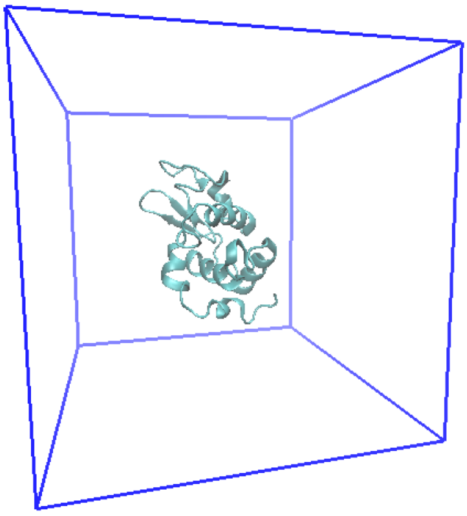


Fig 1: PBC Box around molecule ID: 1aki

---

## Question 4

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

Check the [atom] directive in the topol.top file, we see that the net charge is 8, given by qtot = 8.

Neutralizing these charges requires two commands

gmx grompp –f ions.mdp –c solv.gro –p topol.top –o ions.tpr

This creates an atomic-level description of the system in a binary file called ions.tpr

To neutralize the system we add ions to the system using the genion module

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 bases on qtot

Since the protein has 8 positive charges. To neutralize this GROMACS has added 8 chloride ions. Represented with RESNAME –CL

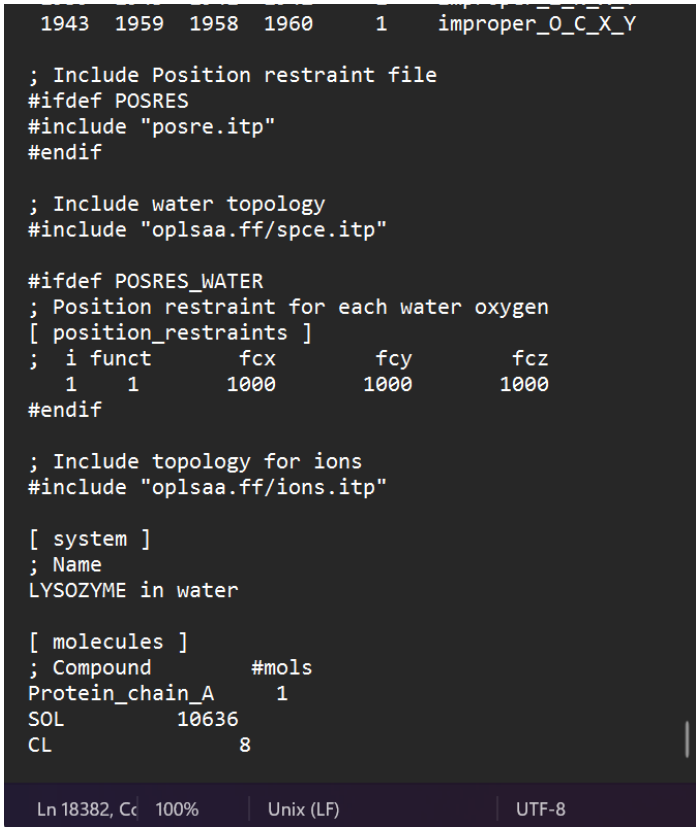



Fig 2:topol file showing the 8 CL ions added to neutralize the protein

## Question 5

**Perform energy minimization of the neutralized system using minim.mdp**
The following are the commands to perform energy minimization using minim.mdp
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**
It took 802 steps to converge to minimal potential energy, since the variation between energy corresponding to steps in 700-800, did not vary much, the process converged within 700 steps, and 50000 steps that were assigned.

**b) Plot the potential energy curve using plotting tool**

gmx energy –f em.edr –o pote.xvg

The data is now stored in pote.xvg file using the gmx energy command
When running this command, you will be prompted to choose the terms you wish to analyze. Select 10 and hit enter to analyze the potential energy. Type 0 and hit enter to stop input.
The data from the .xvg file is loaded into Microsoft Excel and plotted.
Alternatively a matlab code and/or python can be used to plot the curve

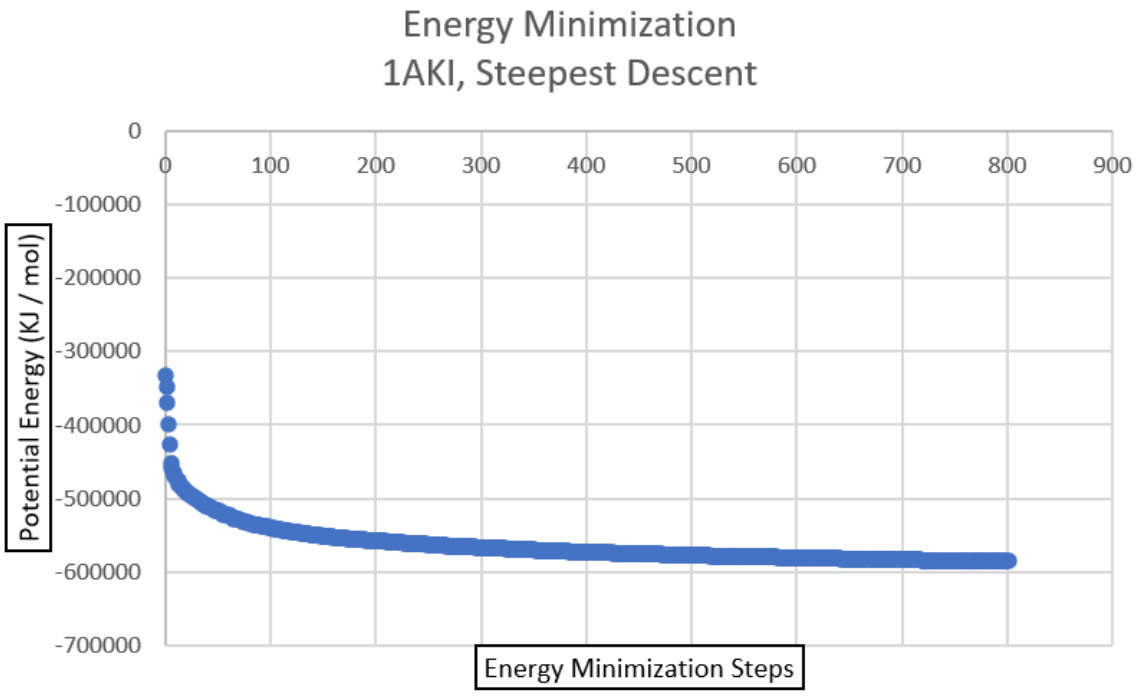



Fig 3: Potential Energy vs steps graph for energy minimization

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

Input GRO file - ions.gro (coloured - Blue)
Output GRO file - em.gro (coloured - Red)

Fig 4: main chain of the protein before and after energy minimization

Minor differences are observed in the main chain. The new cartoon representation does not allow us to observe the solvent ions and their position, it only shows the main chain
To observe the changes between the solvent ions, we represent the two files using Lines. We still observe no significant changes within the main protein.

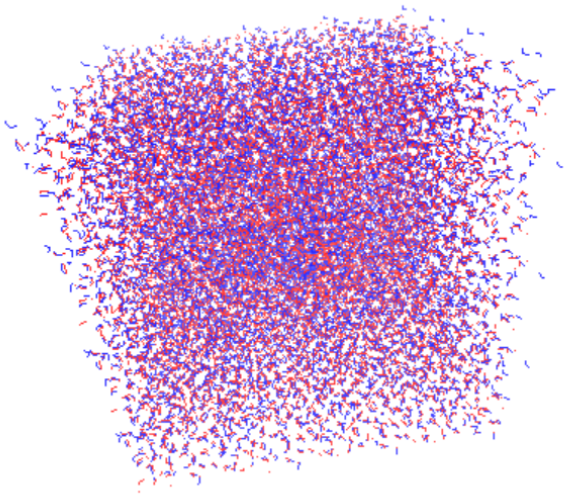


Fig 5: solvent ions before and after energy minimization

**Question 6**

**State the syntax and purpose of a checkpoint file**
The syntax -cpt is used.

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

–cpt is checkpoint file, which restarts the simulation from the previous run

gel_vel = yes, implies the system will use random velocity, hence it will take longer time to equilibrate.

If simulation crashes, use the state.cpt file that is written. Use the –cpi and –append options to mdrun inorder to pick up from where the simulation stopped

---

**Question 7**

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

The padding distance of 1 nm is necessary for water during solvation because the hydration layer around the protein has been found to have dynamics distinct from the bulk water to a distance of 1 nm. Additionally the minimum image convention is satisfied when the padding distance is set as 1nm.
The protein will never coincide with its periodic image, there will be a distance of at least 2 nm between them. Proteins coinciding would imply the energy calculations will be inaccurate.
Padding of 0.5 nm can cause artifacts with its neighboring image, while padding of 2nm is a pretty big box that might demand computational cost and time.
Hence the padding distance of 1nm is the most appropriate.

---

**Question 8**

**What is pbc and why is it necessary to remove pbc before analysis?**
PBC - Periodic Boundary Conditions. This is the box that holds 1 molecule of the protein.

Each molecule has its own PBC and there are many such molecules in the system. They are unit cells used to represent the dynamics and properties of a small individual molecule which can be replicated in other cells. When an atom crosses the PBC, the neighboring atoms enter the cell to maintain constant NPT or NVT ensemble.

Removing PBC before analysis is key because during the simulations the atoms tend to move while the PBC box doesn't. If analysis is performed without removing PBCs, the molecule might remain broken and yield erroneous results.

The command used in this case is
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 a screenshot of the syntax used and the average temperature printed in the terminal.**

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 – v – deffnm nvt
gmx energy – f nvt. edr – o temperature. xvg

The temp.xvg file was parsed to obtain only the values of time and temperature and the data was plotted on excel.

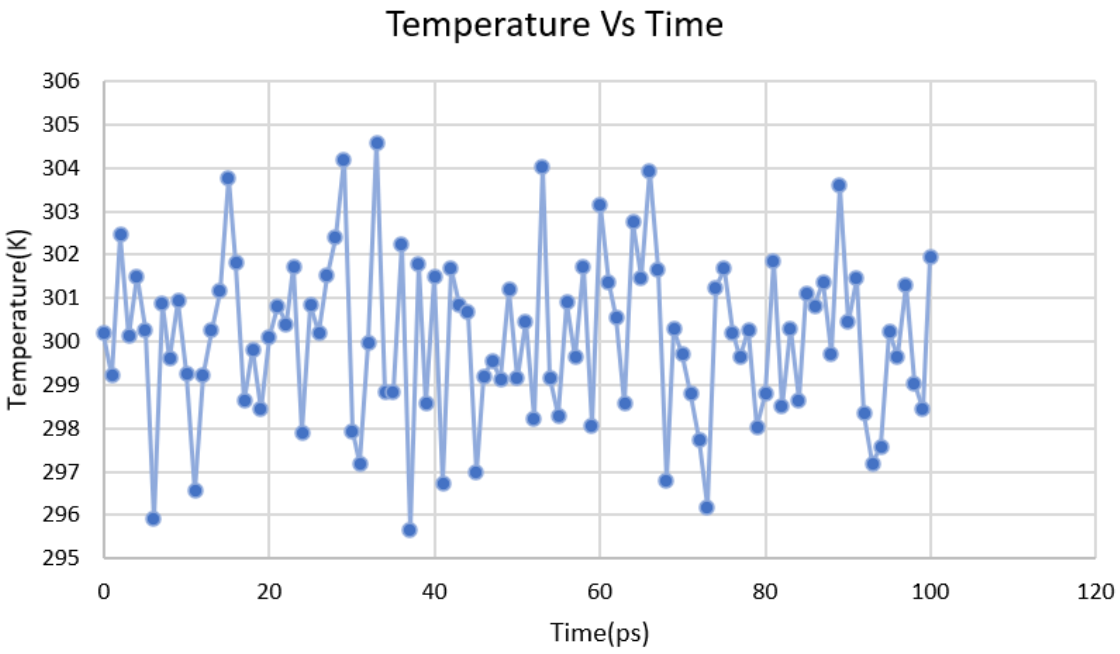



Fig 6: Temperature vs Time graph from 100ps of NVT simulation

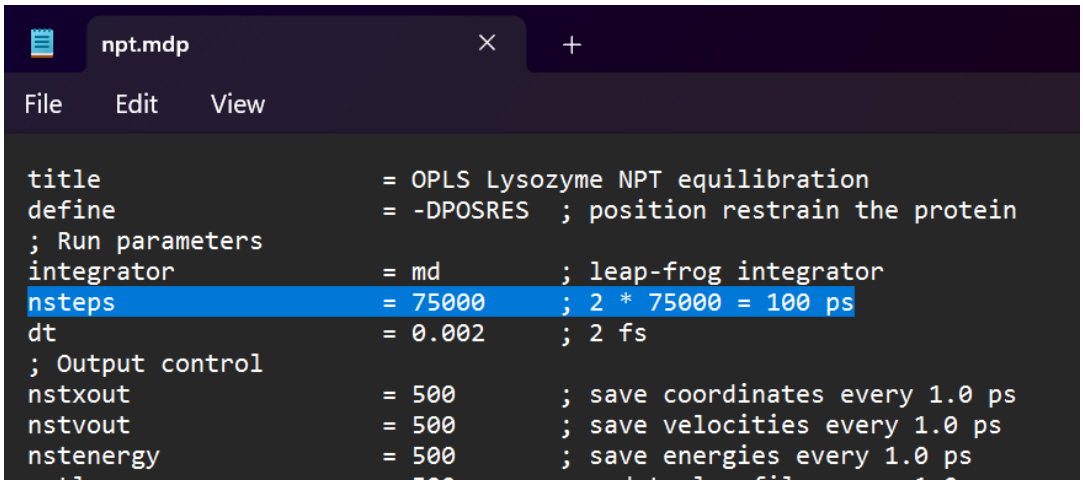
It is observed that the temperature fluctuates around 299.927K, close to 300 kelvin.

The screenshots are shown below-

```
harsha@LAPTOP-LE18RADI:~/GROMACS$ gmx energy -f nvt.edr -o temperature.xvg
                :-) GROMACS - gmx energy, 2020.1-Ubuntu-2020.1-1 (-:

                            GROMACS is written by:
     Emile Apol      Rossen Apostolov      Paul Bauer     Herman J.C. Berendsen
    Par Bjelkmar      Christian Blau   Viacheslav Bolnykh     Kevin Boyd
 Aldert van Buuren   Rudi van Drunen     Anton Feenstra       Alan Gray
  Gerrit Groenhof     Anca Hamuraru    Vincent Hindriksen  M. Eric Irrgang
  Aleksei Iupinov   Christoph Junghans     Joe Jordan     Dimitrios Karkoulis
    Peter Kasson        Jiri Kraus      Carsten Kutzner      Per Larsson
  Justin A. Lemkul    Viveca Lindahl    Magnus Lundborg     Erik Marklund
    Pascal Merz     Pieter Meulenhoff    Teemu Murtola       Szilard Pall
    Sander Pronk      Roland Schulz      Michael Shirts    Alexey Shvetsov
   Alfons Sijbers     Peter Tieleman      Jon Vincent      Teemu Virolainen
 Christian Wennberg    Maarten Wolf      Artem Zhmurov
                           and the project leaders:
        Mark Abraham, Berk Hess, Erik Lindahl, and David van der Spoel

Copyright (c) 1991-2000, University of Groningen, The Netherlands.
Copyright (c) 2001-2019, The GROMACS development team at
Uppsala University, Stockholm University and
the Royal Institute of Technology, Sweden.
check out http://www.gromacs.org for more information.

GROMACS is free software; you can redistribute it and/or modify it
under the terms of the GNU Lesser General Public License
as published by the Free Software Foundation; either version 2.1
of the License, or (at your option) any later version.

GROMACS:      gmx energy, version 2020.1-Ubuntu-2020.1-1
Executable:   /usr/bin/gmx
Data prefix:  /usr
Working dir:  /home/harsha/GROMACS
Command line:
  gmx energy -f nvt.edr -o temperature.xvg

Opened nvt.edr as single precision energy file

Select the terms you want from the following list by
selecting either (part of) the name or the number or a combination.
End your selection with an empty line or a zero.
-------------------------------------------------------------------
  1  Bond             2  Angle            3  Proper-Dih.      4  Ryckaert-Bell.
  5  LJ-14            6  Coulomb-14       7  LJ-(SR)          8  Disper.-corr.
  9  Coulomb-(SR)    10  Coul.-recip.     11  Position-Rest.  12  Potential
 13  Kinetic-En.     14  Total-Energy     15  Conserved-En.   16  Temperature
 17  Pres.-DC        18  Pressure         19  Constr.-rmsd    20  Vir-XX
 21  Vir-XY          22  Vir-XZ           23  Vir-YX          24  Vir-YY
 25  Vir-YZ          26  Vir-ZX           27  Vir-ZY          28  Vir-ZZ
 29  Pres-XX         30  Pres-XY          31  Pres-XZ         32  Pres-YX
 33  Pres-YY         34  Pres-YZ          35  Pres-ZX         36  Pres-ZY
 37  Pres-ZZ         38  #Surf*SurfTen    39  T-Protein       40  T-non-Protein
 41  Lamb-Protein                         42  Lamb-non-Protein

16
0
Last energy frame read 100 time  100.000

Statistics over 50001 steps [ 0.0000 through 100.0000 ps ], 1 data sets
All statistics are over 501 points

Energy                      Average   Err.Est.       RMSD  Tot-Drift
-------------------------------------------------------------------------------
Temperature                 299.927       0.18    3.02154   0.731745  (K)

GROMACS reminds you: "Rat-tat-tat Ka boom boom" (The Smashing Pumpkins)

harsha@LAPTOP-LE18RADI:~/GROMACS$
```

## 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.**




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

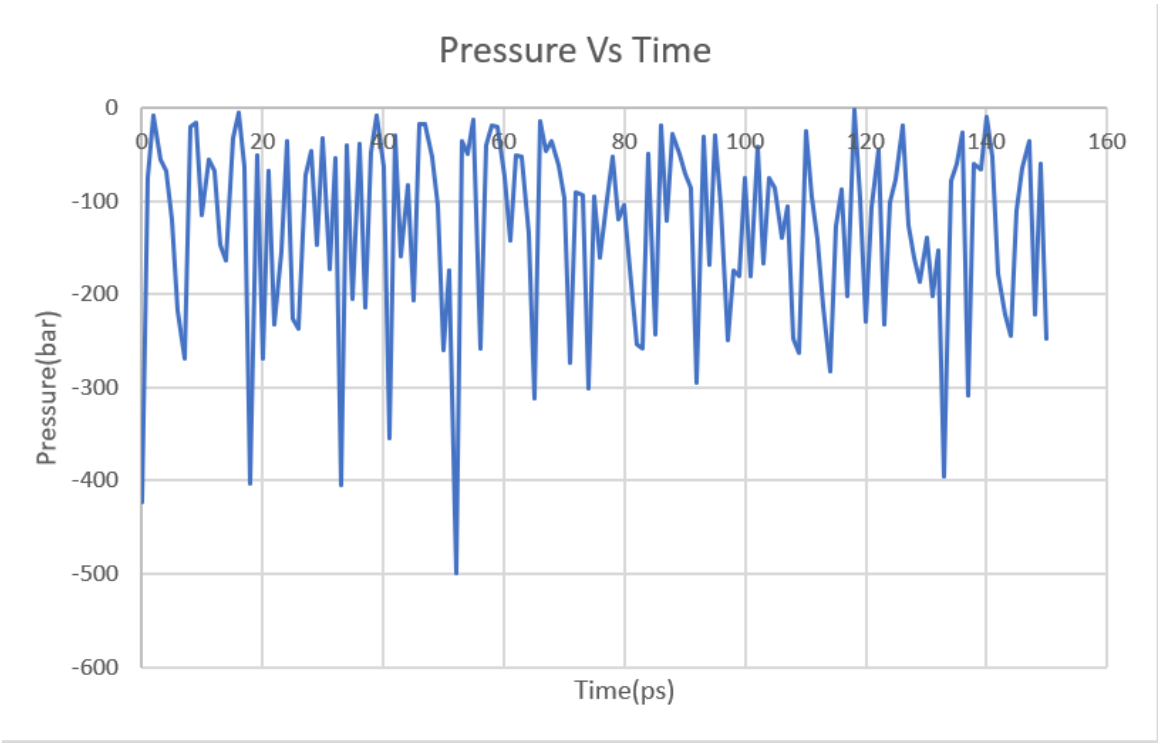
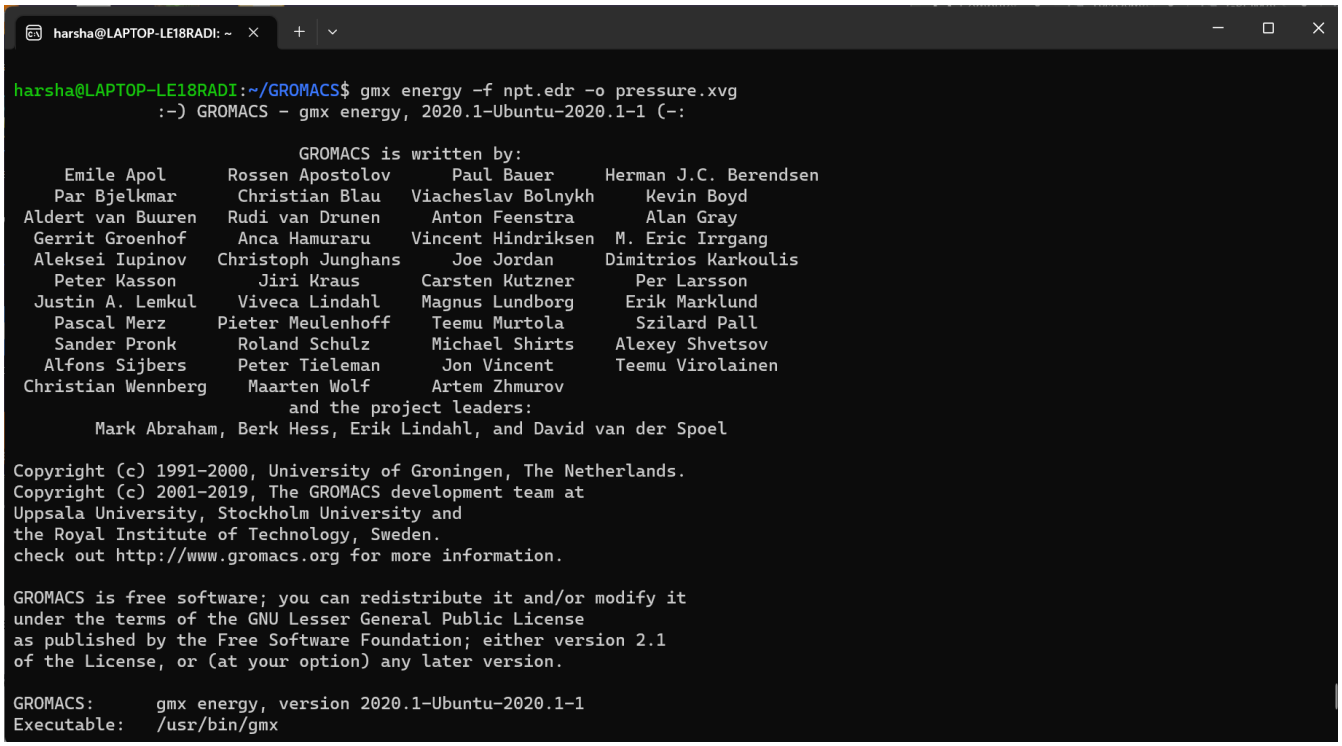
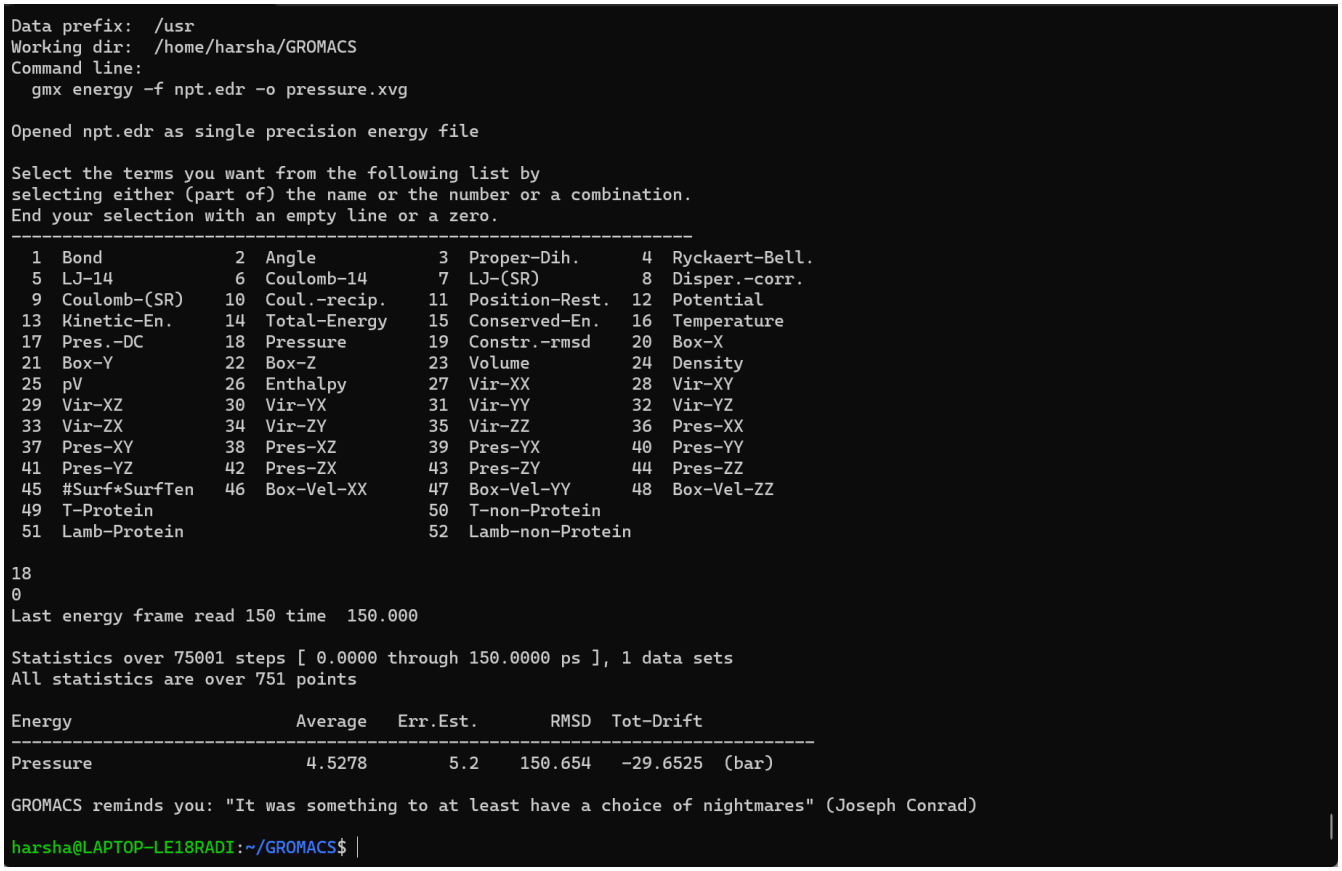



Fig 7: Pressure Vs Time

It is observed that the average pressure in the system is 4.5278 bar. Likewise the RMSD ranges to 150.654 bar, giving us the overall pressure of the system to be greater than 0.

The screenshots are attached below:



**Question 11**

**Perform MD run for 1 ns.**

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

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

```
15 23 17
0
Last energy frame read 100 time 1000.000

Statistics over 500001 steps [ 0.0000 through 1000.0000 ps ], 3 data sets
All statistics are over 5001 points

Energy                      Average   Err.Est.       RMSD  Tot-Drift
-------------------------------------------------------------------------------
Temperature                 299.945      0.027     1.7928 -0.0550579  (K)
Pressure                    2.38102        1.1    150.739   -3.58587  (bar)
Density                      1018.4       0.12    2.59411   0.240712  (kg/m^3)

GROMACS reminds you: "Take Dehydrated Water On Your Desert Trips" (Space Quest III)
```

Fig 8: Analysis of temperature, pressure and density of the system

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

As stated earlier, it is essential to remove the 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.

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

The GMX command is used to compute the distance between both the atoms. Here an index file is created for N and C, i.e, N terminal of first residue and C terminal of end residue. From the md.gro file, the atom's identifiers 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

When asked to specify the number of groups for which you would like to identify distance. Since we have one pair, we enter 0 (the base value)

This will save the corresponding distance between the N and C atoms throughout the trajectory of our simulation. We then plot the distance using excel by parsing through the dist.xvg file.

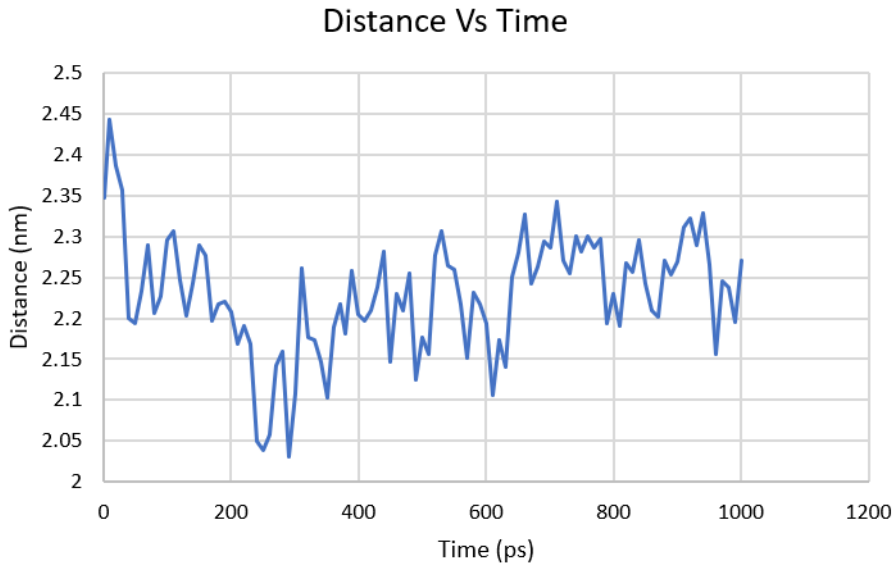



Fig 9: Distance Vs Time

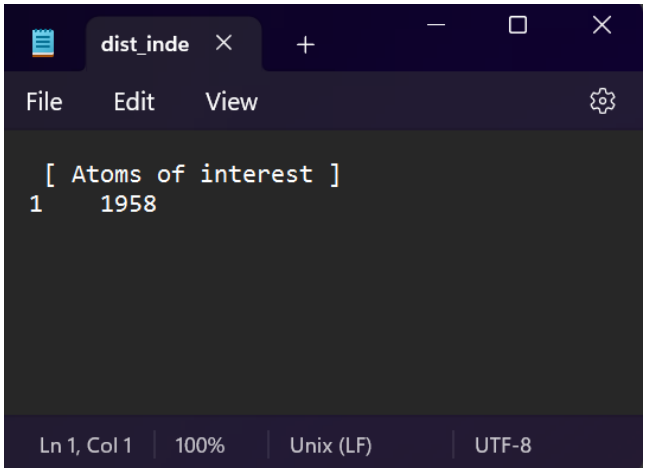
```
Analyzed 101 frames, last time 1000.000
Atoms:
  Number of samples:  101
  Average distance:   2.22936  nm
  Standard deviation: 0.07215  nm

GROMACS reminds you: "Encountered Subspace Anomaly" (Star Trek)
```

Fig 10: Analysis of the Distance between C and N

**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 using a new notepad.

The following content was typed in.




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

We can also automate the process.
The following command
gmx make_ndx –f md.gro
The program will analyze all groups and sub-groups of atoms present in the md.gro file.

**********