Molecular Dynamics Report

Name: Kaifeng Zheng

USC ID: 5051921484

E-mail: Kaifengz@usc.edu

1.Introduction

1.1 The Purpose for the Project

The aim for this project is to simulate Argon crystal behavior as temperature goes up and to study the physical properties of the system. The basic simulation starts when temperature is 0.3 and increases by every 0.3 for each continue simulation. For this project, I selected temperature ranging from 0.3 to 3.6. After running simulations, plots for physical properties can help us to understand the phenomenon vividly. The simulation works utilized USC-HPC supercomputer with LINUX system. Python were used to analyze Data and to obtain plots(Python plot codes: https://github.com/kaifengz007/Molecular-Simulation/tree/master/Molecular%20dynamics).

1.2 Files Configuration

The original files (before **make**) shows as figure 1.1 where md. f90, module. f90 are Fortran source files; Makefile is a compile file telling computer how to compile programs using **make** command; md.in files are inputs for simulation; md.sh submit the simulation job to HPC supercomputer.

```
Makefile  md.in~     md.in-01   md.in-02~  md.in-04   md.in-05~  md.sh
md.f90    md.in-00   md.in-01~  md.in-03   md.in-04~  md.in-06   md.sh~
md.in     md.in-00~  md.in-02   md.in-03~  md.in-05   md.in-06~  module.f90
```

Fig 1.1 files configuration

1.3 Inputs for Simulation

The inputs for one simulation include one initialization step (md.in-00), three heat up steps (md.in-01, md.in-02, md.in-03), two thermalization steps (md.in-04, md.in-05) and one measure step (md.in-06).

For the first time running, it is supposed to initialize a new crystal utilizing initialization step which tells program a new simulation beginning. Therefore, we need set the ini_new value in md.in-00 to 1(see figure 1.2).

```
1000  500  50      # istep, istable, iprint
1   0   1          # ini_new, iscale_x, iscale_v
0.30  1.557         # req_temp, xlcd
6   6   6          # ncell(3)
0   100  0.5       # isquench, iquench, coeff_quench
0.0                # lfreeze
```

Fig 1.2 Initialization step configuration

In heat-up step, there are three input files which contains same inputs configuration. That is because executing three times simulations with same configuration will correct the calculated temperature to the setting temperature.

```
1000  500  50      # istep, istable, iprint
0   0   1          # ini_new, iscale_x, iscale_v
0.30  1.557         # req_temp, xlcd
6   6   6          # ncell(3)
0   100  0.5       # isquench, iquench, coeff_quench
0.0                # lfreeze
```

Fig 1.3 Heat-up step configuration

At thermalization process, it is nothing but setting the time-step variable **istep** to 20000, which includes two input files.

The last step is measurement. In this step, the velocity should be the physical value instead of the value after scaling. Therefore, the scale option for velocity should be turned off, setting **iscale_v** variable to 0.

```
20000  500  50      # istep, istable, iprint
0   0   1          # ini_new, iscale_x, iscale_v
0.30  1.557         # req_temp, xlcd
6   6   6          # ncell(3)
0   100  0.5       # isquench, iquench, coeff_quench
0.0                # lfreeze
```

Fig 1.4 Thermalization step configuration

```
20000  500  50      # istep, istable, iprint
0   0   0          # ini_new, iscale_x, iscale_v
0.30  1.557         # req_temp, xlcd
6   6   6          # ncell(3)
0   100  0.5       # isquench, iquench, coeff_quench
0.0                # lfreeze
```

Fig 1.5 Measurement step configuration

After Setting up those input files correctly, the next step is to link the input files to feed the Fortran source files sequentially. It will construct a onetime simulation. This link works is done by md.sh. **Sbatch** command is utilized to submit simulations to the supercomputer from HPC. After the first run of the simulation with the temperature 0.3, the following simulation processes need to increase the temperature by every 0.3. Note that, in those following simulation processes the initialization step never works anymore, therefore we need to substitute the value of ini_new to 0 and also comment the initialization step in md.sh.

```
## Initialization
#cp -v md.in-00 md.in
#srun -n 1 ./a.out
```

Fig 1.6 Comment Initialization step for following steps

2. Data analysis

2.1 Introduction

After the whole simulation finished (Fig 2.1 shows the result files after simulation), I utilized the data from results to analyze the system physical properties. The tasks include:

(1) Plot the relationship between energy and temperature

(2) Plot the relationship between pressure and temperature

(3) Plot Heat capacity versus T

(4) Plot the relationship between Mean Square Displacement (MSD) over different temperature

(5) Plot Thermal Motion(B) versus temperatures before melting

(6) Plot diffusion coefficient(D) versus temperature after melting

(7) Plot Pair distribution function g(r) versus distance(r)

(8) Plot Coordination number n(r) versus distance(r)

(9) Calculate Thermal Expansion coefficient(α)

```
a.out      md.in      md.in-01~  md.in-04   md.in-06~   md.xyz      xvconf
epotke     md.in~     md.in-02   md.in-04~  md.out      meansqdisp
gr         md.in-00   md.in-02~  md.in-05   md.sh       module.f90
Makefile   md.in-00~  md.in-03   md.in-05~  md.sh~      nr
md.f90     md.in-01   md.in-03~  md.in-06   md_vars.mod properties
```

Fig 2.1 the results list for simulation

2.2 The relationship between temperature and energy

The figure 2.2 shows that when temperature goes higher, energy arise too, and also displays a small jump between 2.1 and 2.4. It is reasonable to conclude that the crystal begins to melt nearby 2.1, for this work, I select the melting points at 2.2.

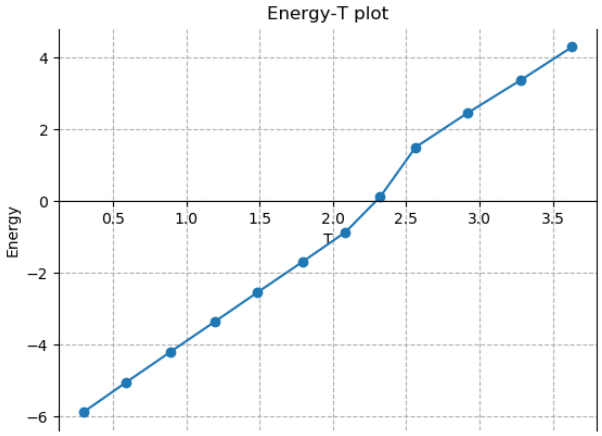

Fig 2.2 Energy VS T plot

2.3 The relationship between pressure and T

Fig2.3 shows when temperature increases, the pressure increases. There is also a jump nearby melting point as Energy-Temperature plot shows us.

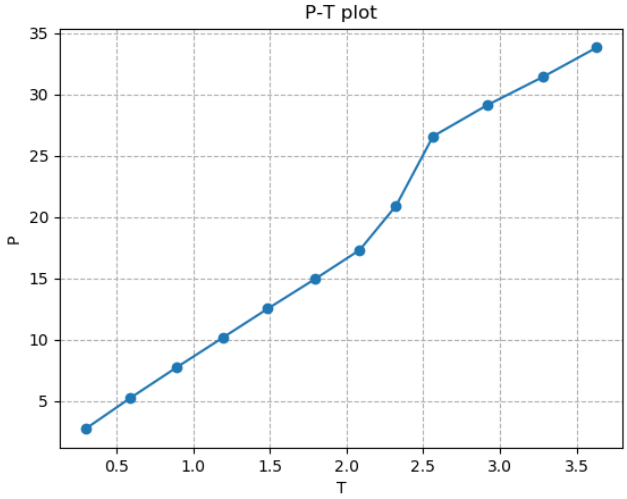

Fig2.3 Pressure VS Temperature plot

2.4 Plot Heat capacity versus T

Heat capacity for this simulation is supposed to be a constant. The value of Cv equals 3R(R is the gas constant). in my calculation, it is 2.696.

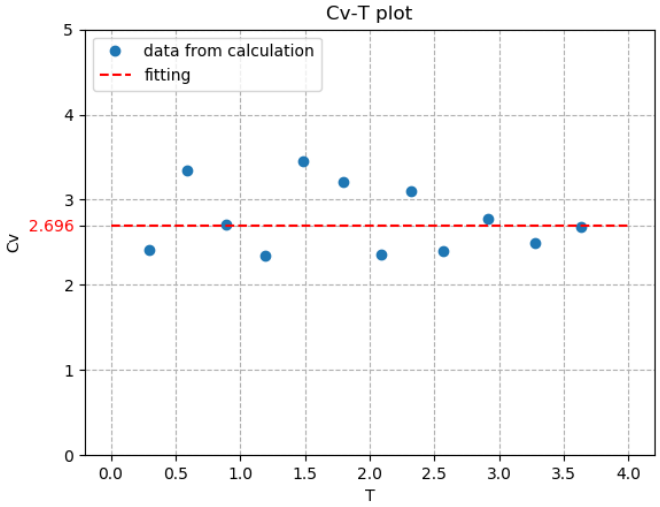

Fig2.4 Cv-T plot

2.5 the relationship between Mean Square Displacement (MSD) over different temperature

Before melting, MSD shows small variance which can considered as a constant. However, after melting (shows in Fig2.6) MSD versus time reveals a linear behavior.

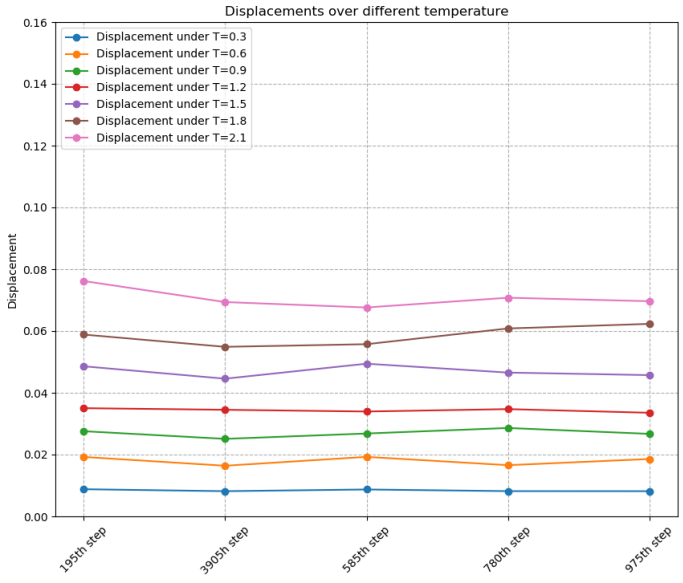

Fig2.5 MSD changing over time with temperature before melting

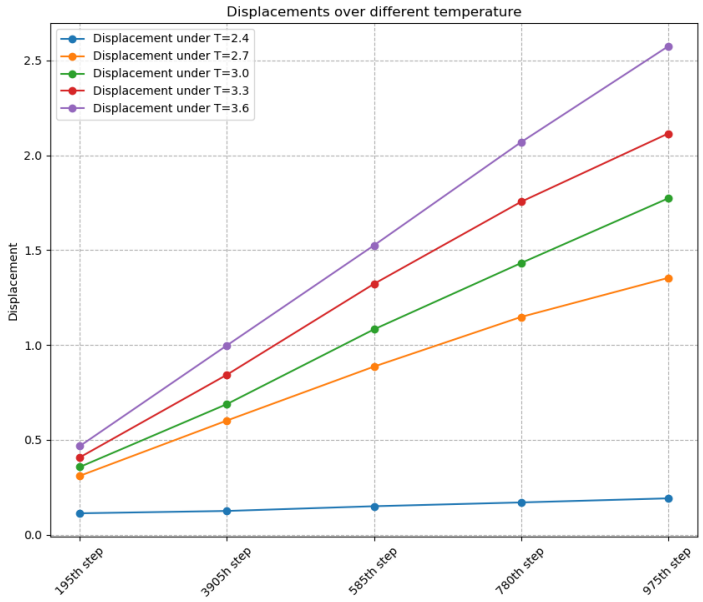

Fig 2.6 MSD changing over time with temperature after melting

2.6 Thermal motion(B) and Diffusion coefficient(D) versus Temperature(T)

Thermal motion for crystal is the MSD before melting. In this case, I averaged the MSD over calculated steps for each temperature lower than 2.2, the relationship between MSD and T showed by left subplot in figure 2.6.

Diffusion Coefficient has correlations with the slopes on the MSD-T plot after melting. The function uses to reveal the relationship between D and slope:

$$MSD = 6DT + constant$$

Obtaining slope from MSD-T plot, I calculated D by using:

$$\mathrm{D} = \frac{\mathrm{slope}}{6}$$

Through some calculations, the relationship between D and T presents on right subplot in fig 2.7.

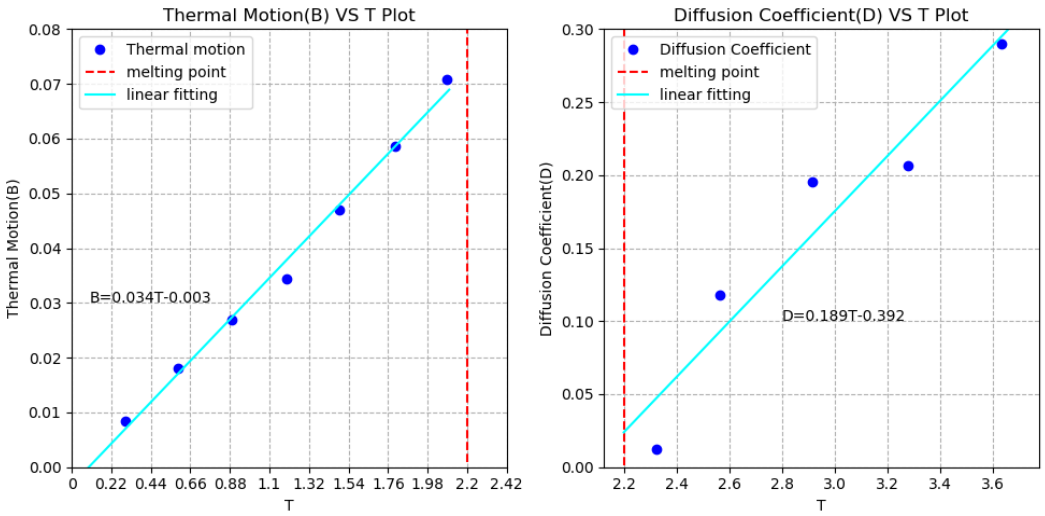

Fig 2.7 B and D versus temperature

2.7 Pair distribution function g(r) versus distance(r)

Pair distribution function reveals the probability distribution of particles around with certain distance from kernel particle. During heating process, there is a phase changing process from solid to liquid. Fig2.8 shows pair distribution function before melting, which can be identified as the discontinue nature between peaks.

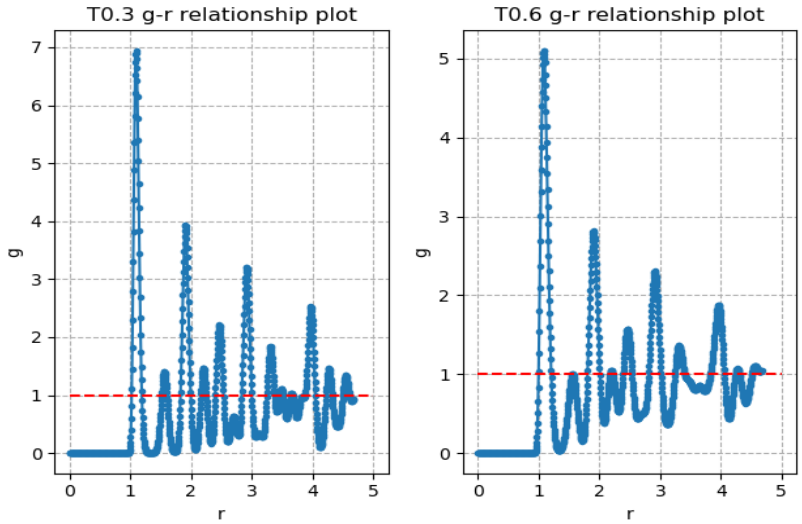

Fig2.8 pair distribution function versus distance plots T=0.3, T=0.6(before melting)

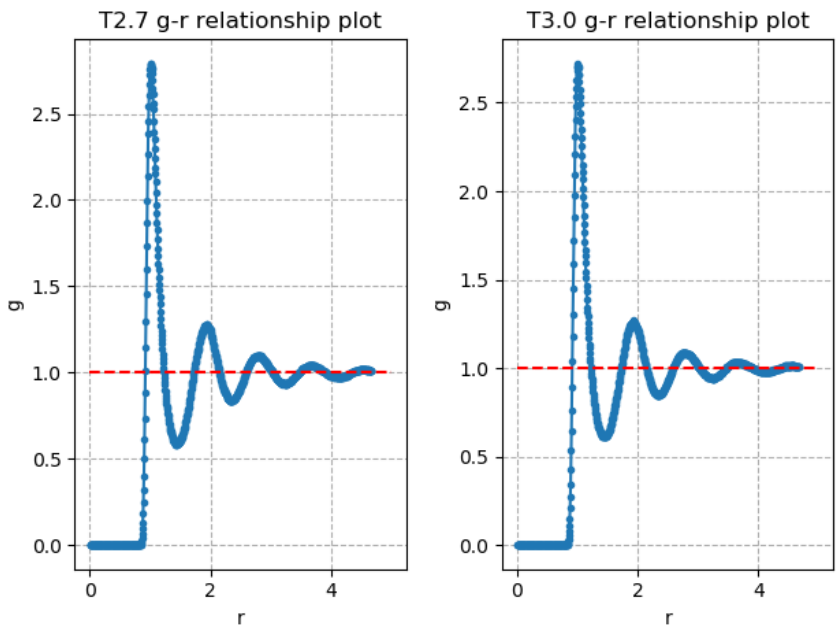

Fig2.9 pair distribution function versus distance plots T=2.7, T=3.0(after melting)

Fig2.9 real the continuous nature, and also when distance goes infinity, the g function goes to 1. It also reveals that the highest peak after melting is much lower than the highest peak before melting.

2.8 Coordination number n(r) versus distance(r)

Coordination number is an integral of g(r), which can be calculated using the equation as following:

$$n = 4\pi \int_{r_0}^{r} r^2 g(r) \rho dr$$

Fig2.10 shows the coordination number changes versus distance before melting, the cascades phenomenon as an identification for before melting.

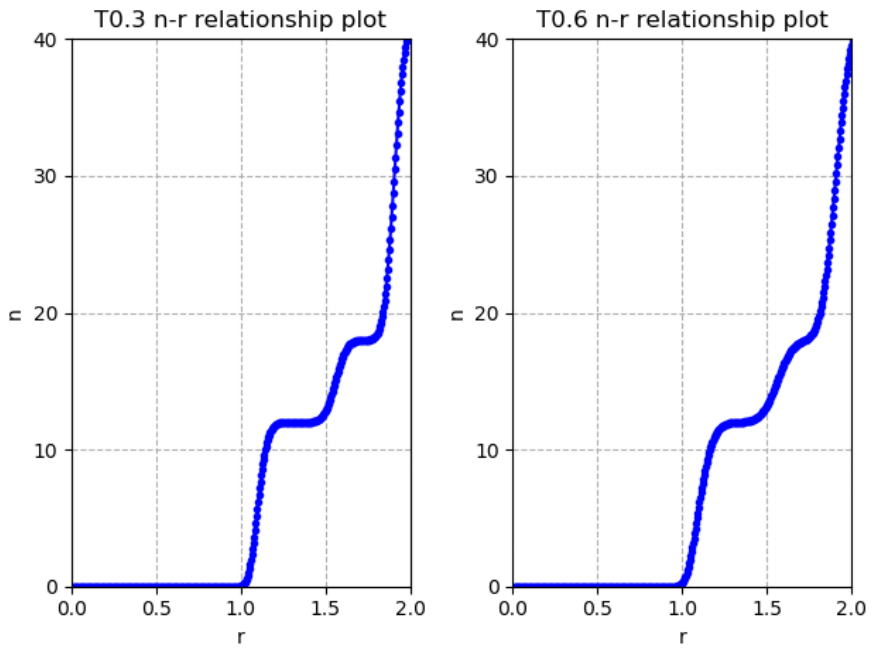

Fig 2.10 Coordination number versus distance plot T=0.3, T=0.6(before melting)

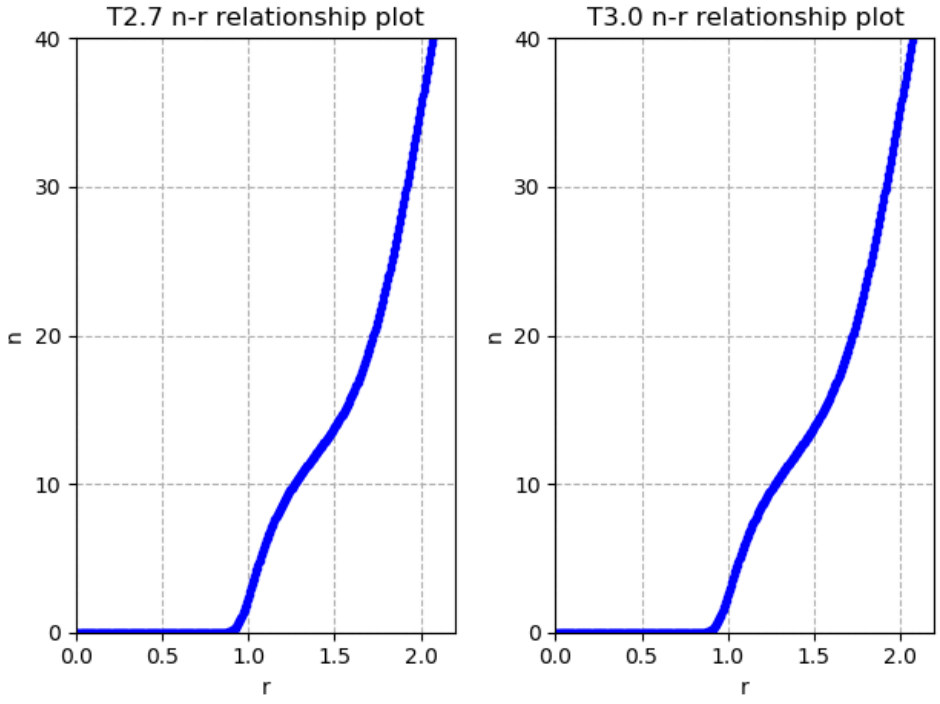

Fig 2.11 Coordination number versus distance plot T=2.7, T=3.0(after melting)

In Fig 2.11, the figures reveal the situations after melting, the smoother shape instead of the cascades shape.

2.9 Thermal Expansion Coefficient

Thermal expansion coefficient is a material property to measure a material expands during heating process. It can be calculate using equation as following:

$$\alpha = \frac{1}{a}\frac{da}{dT}$$

where a is lattice constant, T is temperature and α is thermal expansion coefficient.

However, we can't calculate the derivative directly in silico. According some fundamental knowledge from math and physics, scientist derived a numerical way to calculate this derivative.

The first step is to process simulations with different lattice constants and different temperatures before melting. For this project, I start lattice constant at 1.557 and multiply 1.01,1.02,,.1.09 for each time with temperature from 0.3-1.8. The data from calculation can be used to generate a P-T plots for each lattice constant over different temperature. This P-T relationship plot shows highly linear property, therefore We can obtain the group of linear function over the lattice constant set using linear fitting.

The second step is to decide the optimal temperatures, T_optimal (I calculated optimal temperature in python using Sympy package to solve the linear equation in loop). Those are temperatures at zero pressure with each lattice constant. Then, we can plot a figure shows the relationship of lattice constant(a) versus Temperature and calculate thermal expansion coefficient.

According to the data from simulations, the crystal is melting when the temperature around 2.1. In this case, we need to choose a temperature lower than 2.1. Actually, when lattice constant becomes bigger, the melting point decreases. The figure 2.12 shows the changes over different lattice constants. In this case, I selected the first four lattice constants and temperatures, lattice constants from 1.557-1.604, temperature from 0.3-1.2, for fitting and calculation. Figure 2.13 present the fitting plot for lattice constants and optimal temperature. Finally, the thermal expansion coefficient is 0.061. The equation of calculation is given by:

$$\text{Therm\_exp}(\alpha) = \frac{a_i - a_0}{(T_i - 0) \times a_0}$$

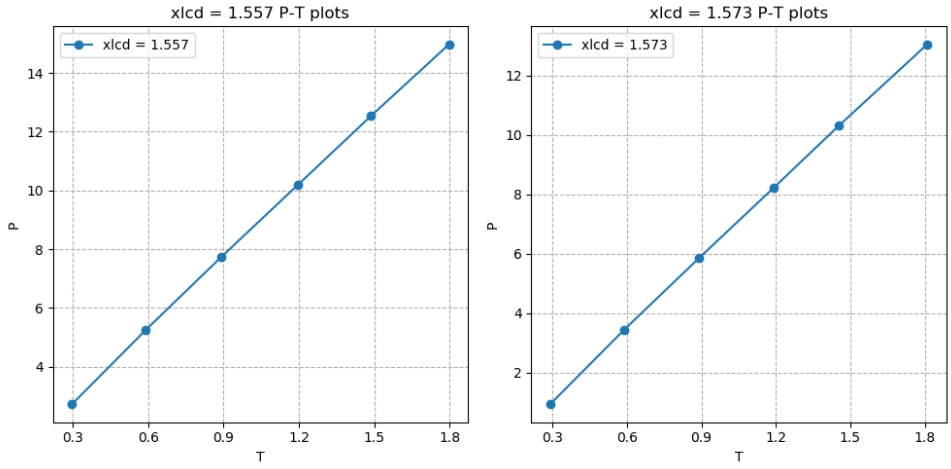

Fig 2.12(1) The P-T plots over different temperatures and lattice constants

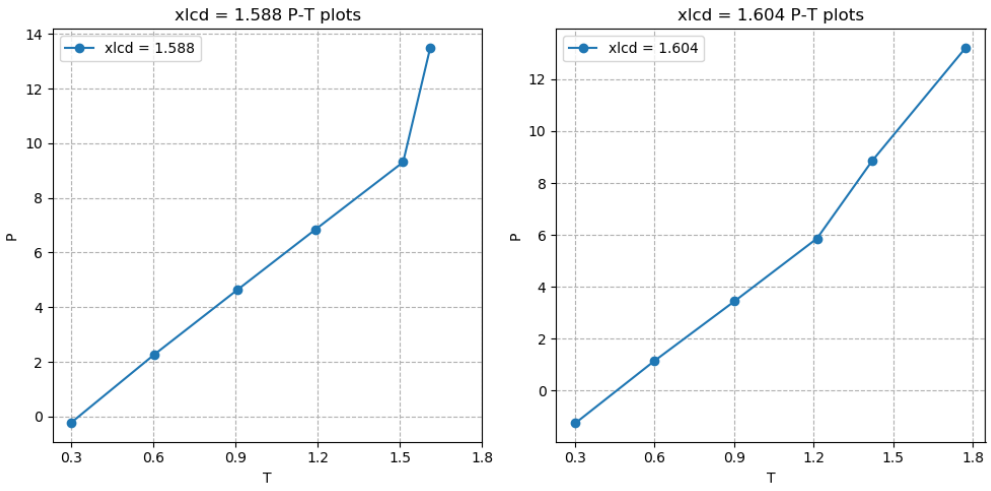

Fig 2.12(2) The P-T plots over different temperatures and lattice constants

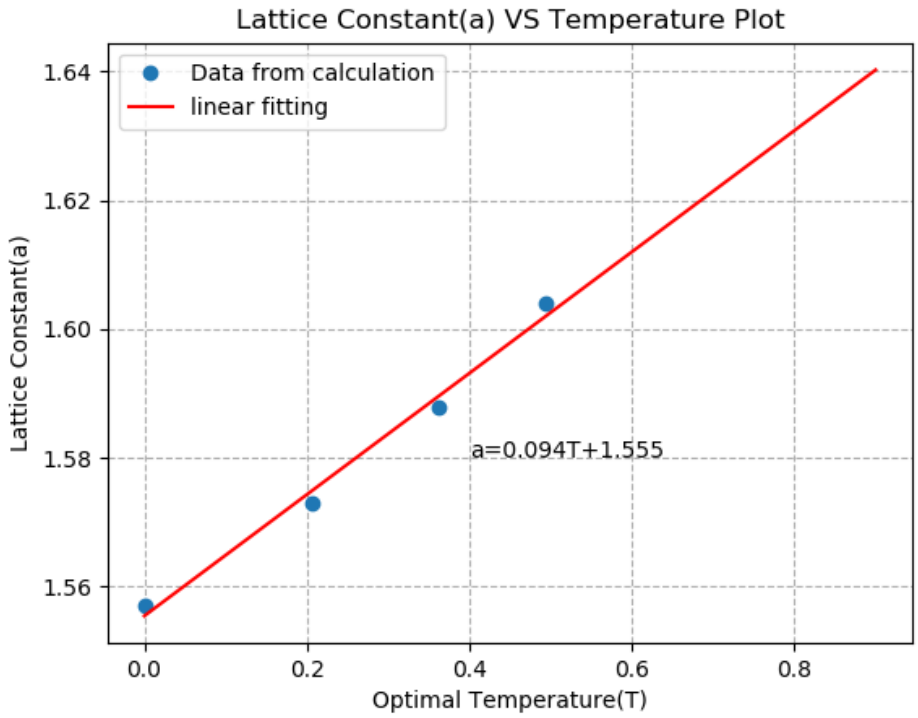

Fig 2.13 The linear fitting plots of lattice constants and Temperature

The fitting results on lattice constant versus temperature plot shows that when the optimal temperature is zero, the lattice constant is 1.555. In the real situation, it is supposed to be 1.557 when the optimal temperature is zero. However, we can accept the result, because the relative error is only 0.13%.

3. programs and work descriptions

3.1 Unix files construction

In the project, there are two parts of works with different file construction, the first is to simulate a process one crystal melting and another is to simulate some crystals with different lattice constants finding the thermal expansion coefficient in the end.

For the first part, I constructed a **T0** directory to do the simulations continuedly and each result copied in different directories with the name of setting temperatures. The file construction shows as below:

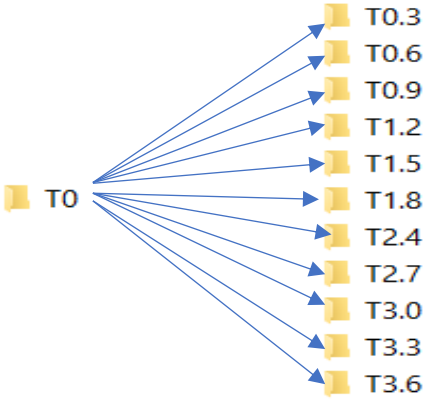

Fig 3.1 files construction for melting simulation

For the second part, it is supposed to simulate under situations of every temperature below the melting point with different lattice constant. The configuration here is totally different than the first one. I built the parallel file structure. The file structure I build as following.

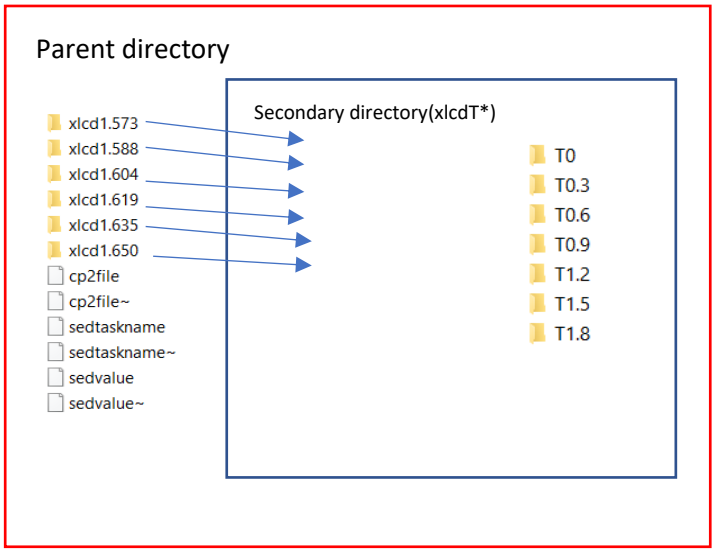

Fig 3.2 files construction for thermal expansion simulation

3.2 Interesting codes for speed-up process by hands

(1) **sed** command is for substitution of the variable values. For example, I want to change the temperature 0.3 to 0.6, under the directory containing md.in files, I can write down:

```
sed -i 's/0.30/0.60/g' md.in-*
```

(2) If we want to copy all files and directories from a cretain directory to a branch of other directories, we can use pipe manner, for example, I want to copy all files and directories from xlcdT0.3 to both directories: xlcdT0.6 and xlcdT0.9. I can write the codes as following:

```
echo xlcdT0.6 xlcdT0.9 | xargs -n 1 cp -Rv xlcdT0.3/*
```

3.3 Problems on HPC

If the files occupied too much space, the system will prompt an error said data exceeded. For example:

```
kaifengz@hpc-login3:~/MD/Tnew/xlcd/xlcdT0.3/xlcd1.557$ sed -i 's/0.30/0.50/g' md.in-00
sed: couldn't close ./sedFE1BTd: Disk quota exceeded
```

In my consideration, it is happened when the capacity of files reaches the system quota limitation. That means there is no room for new files. **sed** can't work correctly because it will generate a backup file(such as the file as following) after running.

```
sedoSc87g
```

The quota for every user should be 1 gigabate. However, the quota is less or equal than 978M for each user. The following picture shows the storage when I got the limitation.

```
kaifengz@hpc-login3:~$ du -sh
978M    .
```

One way to prevent from meeting the limitation is to download the finished files and delete them on HPC account to free the space, then there are no error prompt anymore.

4. Conclusion

Viewing the overall work of this molecular dynamics project, some basic ideas of MD simulation were covered in the work, and the behaviors of Argon atoms during heating up were been reported.

Part 1 talked about the file configuration and how those input files working during the project.

Part 2 focused on the data analysises after simulation. During heatting up, a melting process was happened and the vibarition of atoms become more dramatical. The melting point is around 2.1(In this project, I select 2.2) The obvious evidences for obtaining this data can be found in P-T, Energy-P and MSD-t plots: we can easily see a jump on P-T and Energy-P plots when there is a phase changing process. On the MSD plot, a behavior altered after temperature 2.1. Before melting, a termal motion can be calculated by used MSD plot, and Diffusion coefficients can be obtained using the part of MSD plot when temperature higher than melting temperature (see part2.6). The obvious difference on pair distribution function(g) and coordination function has been report in part 2.7, the discontinuous nature of g(r) under the melting point can be obversed in the plot, however a continuous property shows after melting. We can also see the limitation when distance(r) goes to infinity equalling 1. The n(r) function shows cascades nature before the melting process, and become smooth after melting. In 2.9, the thermal expansion were obtained using a bunch of parellel simulations over different Argon crystals with different crystal constans and temperature. The value of thermal expansion is 0.06.

Part 3 was talking the codes I used to submit the jobs and some situations I meet and possible solutions for the work during using USC-HPC system.

Reference

[1] https://en.wikipedia.org/wiki/Coordination_number

[2] https://en.wikipedia.org/wiki/List_of_Unix_commands

[3] https://matplotlib.org/users/pyplot_tutorial.html