

# Project 4: Molecular Dynamics Simulations

CHEN 4880

Jack Vogelgesang  
5-3-2022

### Problem 1: Equilibration, Energy Conservation, and Fluctuations

- A. a. In order to minimize computational resources used per simulation the integration time step should be chosen as large as possible. If the integration step size is too large energy conservation will be violated. In order to conserve energy, the maximal timestep is limited by the fastest frequency in our system which is the bond vibrations.
- b. In order to find the largest deviations from the average total energy for the final 90 picoseconds (ps) of our system the code in Figure 1 of Appendix A was used. A 4 x 4 x 4 supercell was used for each run as well as an initialized temperature of 5000 degrees Kelvin. Five different time steps were used which were 2, 5, 10, 16, and 20 femtoseconds (fs). We wanted to simulate a total time of 100 picoseconds so each time step used a different number of steps which were 50,000 for 2 fs, 20,000 for 5 fs, 10,000 for 10 fs, 6250 for 16 fs, and 5000 for 20 fs. The simulation was ran for each time step and the resulting data was plugged into Excel where the average of the total energy was calculated. Then we used the max and min function of Excel to find the two values that were furthest away from the average total energy and then selected the one that was the largest deviation from the average. The results can be seen in Table 1 below.

Table 1: Average Total Energy and Largest Deviation per Time Step

Time Step (fs)	Average Total Energy (eV)	Largest Deviation (Actual Value of Total Energy (eV))
2	-5.12594	6.13E-05 (-5.1260019)
5	-5.12521	.000262 (-5.12547)
10	-5.12348	.000627 (-5.12411)
16	-5.11845	.00159 (-5.11686)
20	-5.11126	.00472 (-5.10654)

- c. Using the data from part b for each time step a graph of time vs total energy was plotted for the final 90 picoseconds of the simulation. Then a trendline was fitted to said graphs and the slope of this trendline gives the systemic drift of the total energy. The slope of each trendline which is also the magnitude of the drift is listed below in Table 2. The graphs with their corresponding trendlines can be seen in Figures 2 through 6 in Appendix A. It can be seen that all time steps experience a small increase in total energy drift as time moves forward.

Table 2: Magnitude of Total Energy Drift for Each Time Step

Time Step (fs)	Slope of Trendline / Magnitude of Drift of Total Energy
2	$1.1 \times 10^{-6}$
5	$2.6 \times 10^{-6}$
10	$2.0 \times 10^{-6}$
16	$1.1 \times 10^{-5}$
20	$5.3 \times 10^{-5}$

- B. a. Using our data from the simulation of the system with a time step of 2 fs the mean temperature of the final 50 ps was 2378.793 degrees Kelvin. This differs from the velocity initialization temperature of 5000 degrees Kelvin because the total energy and more specifically the kinetic energy is much less after the simulation has ran for 50 ps. When initialized the starting kinetic energy is .644 but by time 50 ps the kinetic energy is 0.329 which will cause the temperature to be greatly reduced since temperature is correlated with the kinetic energy of a system.
- b. In order to estimate how long it takes for the system to reach equilibrium a graph of the temperature vs time for the first 5 ps of the 2 fs timestep data was created which can be seen in Figure 1 below.

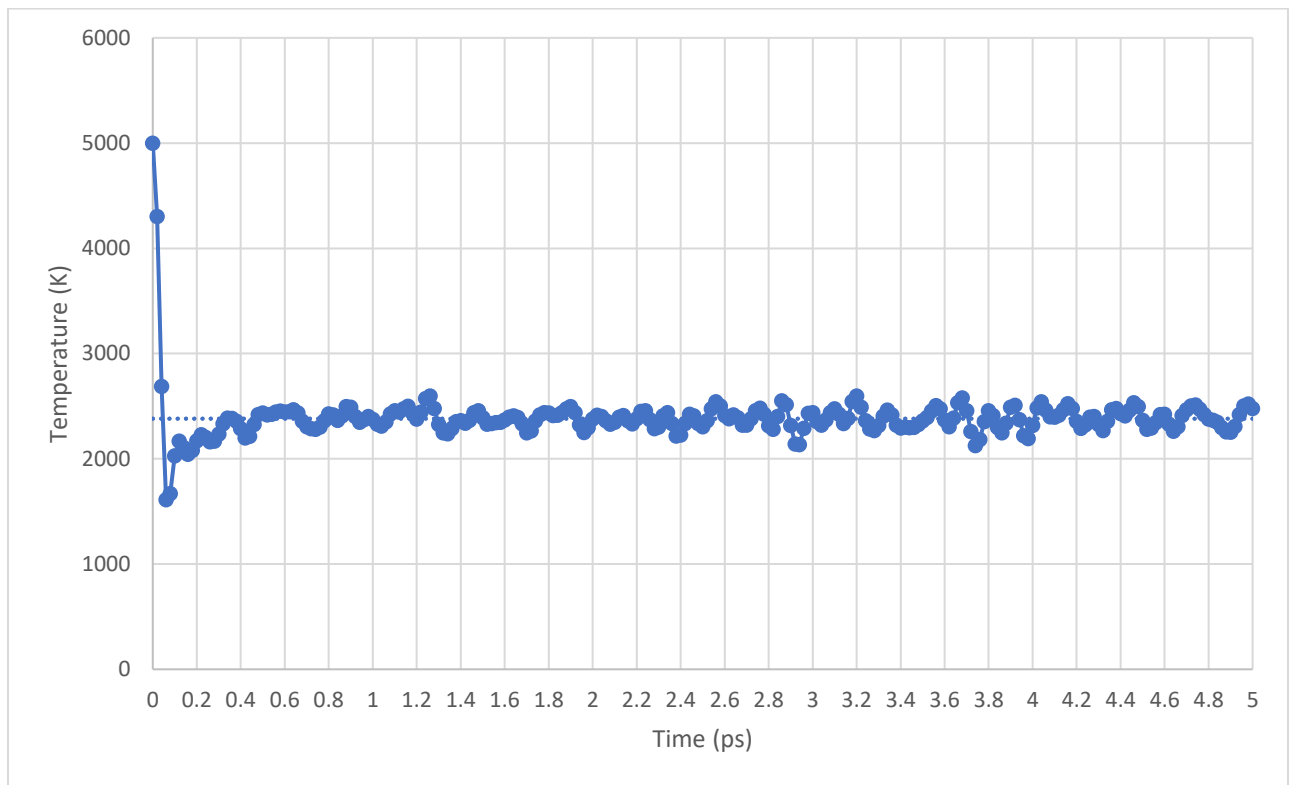


Figure 1: Time vs Temperature of Platinum System Using 2 fs Time Step

It takes approximately 0.5 to 0.6 picoseconds for the system to reach equilibrium as that is when the temperature begins to fluctuate around the mean temperature of 2378.793 degrees Kelvin which is where the line on the graph is placed.

- C. a. In order to investigate how the magnitude of the temperature fluctuations is affected by cell size the 20 fs time step was used over 20 ps for supercell sizes of 4 x 4 x 4, 8 x 8 x 8, and 16 x 16 x 16. For each cell size the temperature was plotted versus the time which can be seen below in Figure 2. It can be seen qualitatively that as the cell size increases temperature fluctuations decrease.

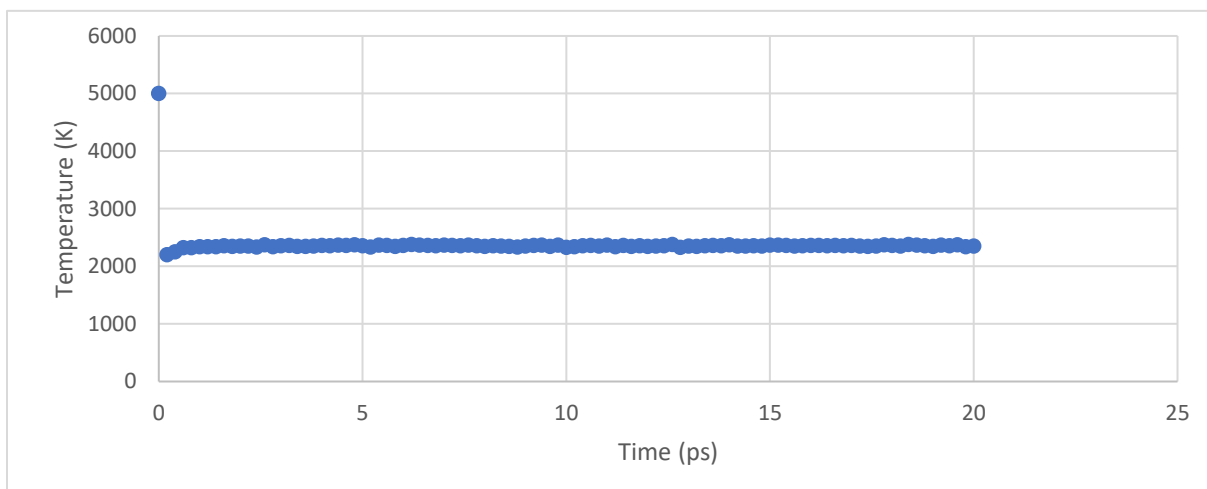
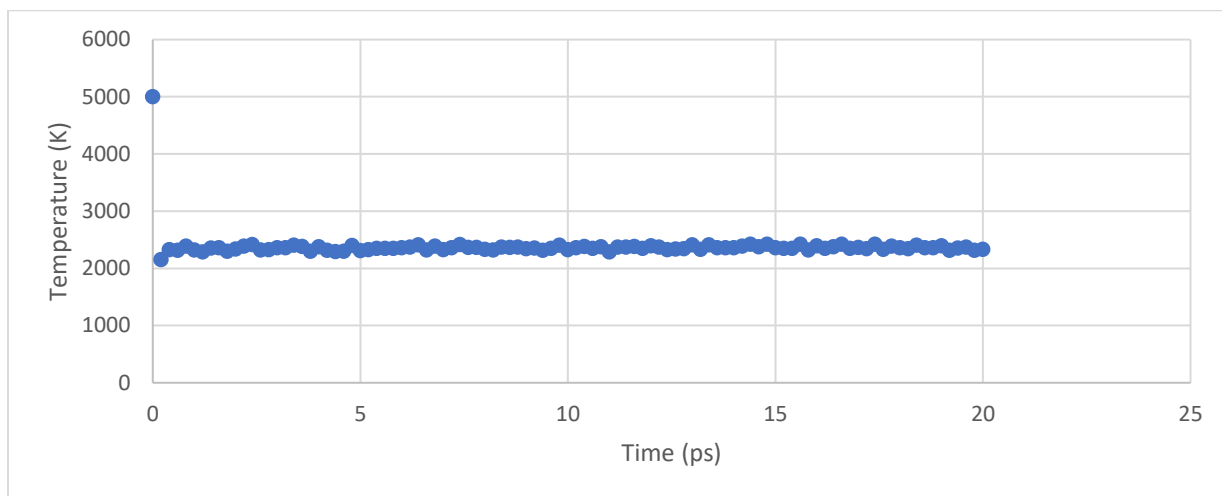
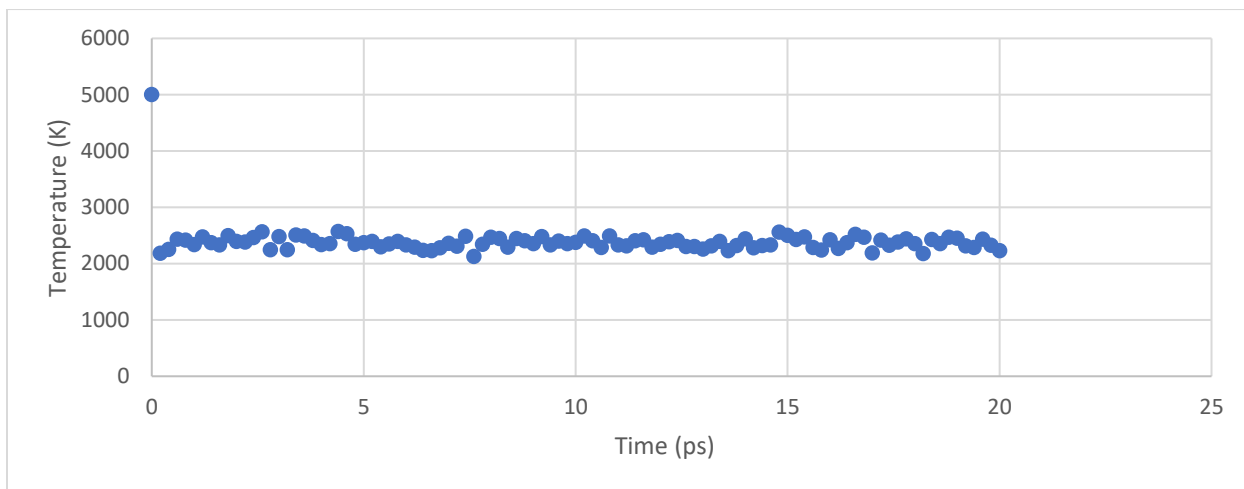


Figure 2: Graphs of Time vs Temperature using a 1) 4x4x4 cell 2) 8x8x8 cell and 3) 16x16x16 cell size

b. As the thermodynamic limit of N goes to infinity the fluctuations will vanish but be more computationally expensive as there are more elements to simulate.

## Problem 2: Structure of Molten Platinum

- A. a. To show that a 2 fs time step is appropriate for simulating molten platinum at a temperature of 2300 degrees Kelvin and pressure at 1 bar the code seen in Figure 7 of Appendix A was used. A 10 x 10 x 10 supercell was used with 100,000 MD steps and an output frequency of 5000 steps which produced the data seen in Table 3 below.

Table 3: Data from Simulation of Molten Platinum

Step	"Time"	"Temp"	"TotEng"	"PotEng"	"KinEng"	"Press"	"Lx"	"Ly"	"Lz"
0	0	4600	-5.17543	-5.76988	0.594448	33442.33	39.24	39.24	39.24
5000	10	2242.71	-5.15353	-5.44335	0.289821	142941.3	39.25304	39.25304	39.25304
10000	20	2339.453	-5.14159	-5.44391	0.302322	135923	39.29233	39.29233	39.29233
15000	30	2336.035	-5.14179	-5.44367	0.301881	122356.2	39.35637	39.35637	39.35637
20000	40	2293.718	-5.13569	-5.4321	0.296412	110950.6	39.4404	39.4404	39.4404
25000	50	2325.084	-5.12528	-5.42575	0.300466	95371.58	39.53636	39.53636	39.53636
30000	60	2188.95	-5.11616	-5.39903	0.282873	85177.44	39.63445	39.63445	39.63445
35000	70	2045.143	-5.07372	-5.33801	0.264289	92879.34	39.72763	39.72763	39.72763
40000	80	2291.399	-5.02019	-5.31631	0.296113	88575.87	39.81437	39.81437	39.81437
45000	90	2348.911	-4.99551	-5.29905	0.303545	82706.33	39.89551	39.89551	39.89551
50000	100	2263.25	-5.00423	-5.2967	0.292475	69269.27	39.97235	39.97235	39.97235
55000	110	2280.274	-4.99119	-5.28586	0.294675	62827.77	40.04537	40.04537	40.04537
60000	120	2315.785	-4.98504	-5.2843	0.299264	51894.51	40.11553	40.11553	40.11553
65000	130	2232.366	-4.98863	-5.27712	0.288484	42559.46	40.18274	40.18274	40.18274
70000	140	2299.173	-4.97213	-5.26924	0.297117	38000.33	40.24634	40.24634	40.24634
75000	150	2263.243	-4.97292	-5.2654	0.292474	28005.93	40.30607	40.30607	40.30607
80000	160	2331.003	-4.96094	-5.26217	0.30123	22015.76	40.36118	40.36118	40.36118
85000	170	2269.851	-4.96444	-5.25777	0.293328	15195.72	40.41142	40.41142	40.41142
90000	180	2253.098	-4.96043	-5.25159	0.291163	11099.47	40.45624	40.45624	40.45624
95000	190	2295.061	-4.94829	-5.24488	0.296586	7274.932	40.49562	40.49562	40.49562
100000	200	2291.289	-4.94719	-5.24329	0.296098	5620.676	40.53017	40.53017	40.53017

If the time step of 2 fs is truly appropriate to use then energy will be conserved which we can check by finding the largest deviations from the average total energy since it is a conserved quantity. The average total energy of the final 180 ps of Table 3 is -5.02344 eV. The largest deviation from this mean is .1183 eV which occurred at step 30 (-5.14179 eV). This is quite a large deviation so we will also check the systemic drift of the total

energy. When the total energy vs time for the final 180 ps is plotted the graph in Figure 3 is obtained.

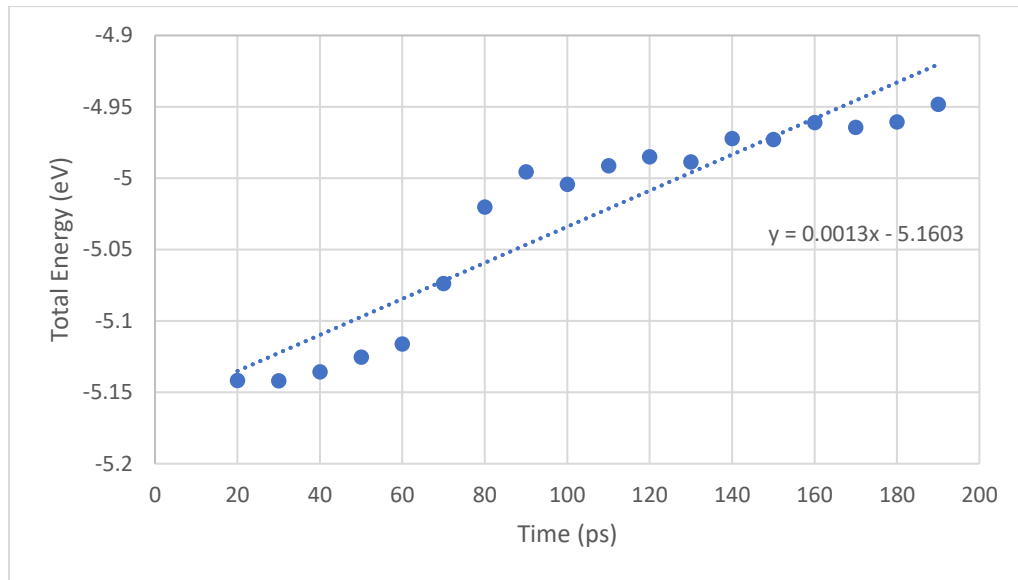


Figure 3: Total Energy vs Time for Molten Platinum

Plotting the trendline gives us a drift of .0013 which is small enough to say that energy is conserved and that a time step of 2 fs is acceptable to use.

b. In order to tell when the system reaches equilibrium, we can look at the temperature fluctuations around the mean temperature of the final 100 ps of the simulation. Using the temperature values in Table 3 the mean temperature is 2287.285. Plotting the temperature vs time give us the graph seen in Figure 4 below.

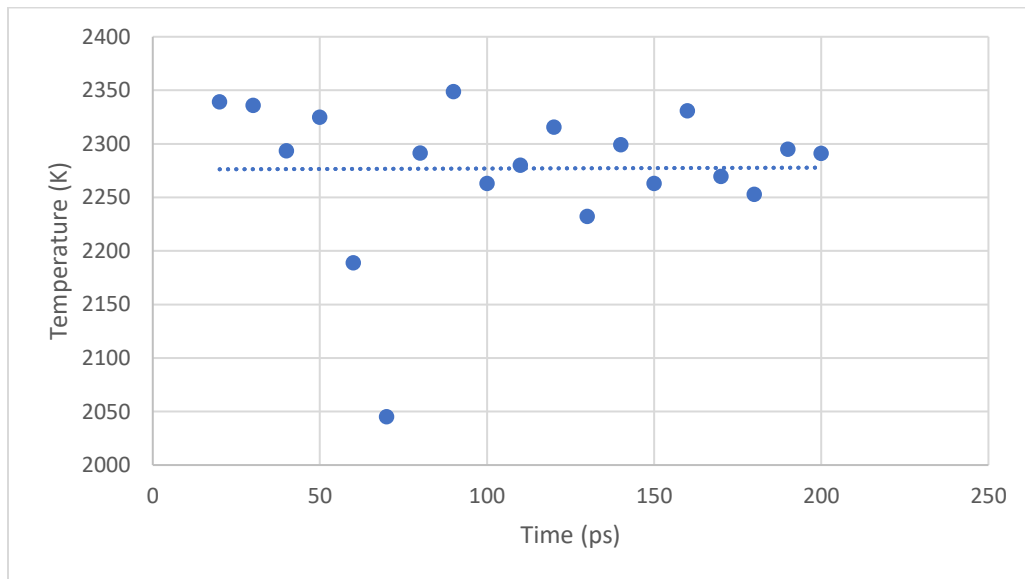


Figure 4: Temperature vs Time for Molten Platinum

Looking at the graph the temperature begins to fluctuate around the mean temperature of 2287.285 at around 80 ps so it takes about 80 ps to reach equilibrium.

- B.
- a. The RDF above the melting point should have smoother peaks as it is no longer in a crystalline form and thus atoms will fluctuate in distance from another in much greater distances and decrease the probability of seeing one another. RDFs below the melting point should have very pronounced peaks as the atoms will form a much more rigid structure with more defined positions and thus a higher probability of finding an atom.
  - b. Increasing the temperature causes the RDF peaks to be smeared out (or broadened) because the molecules will have more kinetic energy due to the rise in temperature and thus move around more so that means the distances are no longer as static and decreasing the probability that they see one another. Also, the correlation lengths are shorter and the functions approach 1 much faster. Decreasing the temperature will cause more peaks to occur and the probability of finding an atom within the set distance will increase so the peaks will be greater than those of higher temperatures.
  - c. Increasing the pressure of the RDF makes the peaks narrower/sharper as it will force the molecules closer together and thus a higher probability of being within the RDF shell.
  - d. An RDF of a crystal at 0 K would have very sharp peaks at each coordination shell as the atoms are held in place due to having no energy to move. An RDF of crystal at infinite Kelvin would look almost if not completely flat as the molecules have so much energy and move so fast that there would be no structure and thus very little probability of finding an atom within the measured distance  $r$ .
- C. Using the code in Figure 7 of Appendix A the RDFs were calculated for the temperatures of 1500, 1600, and 1700 Kelvin which are all below the melting temperature of 2041 Kelvin and 2300, 2750, and 2900 Kelvin which are all above the melting temperature. All RDFs simulations were calculated from a simulation with 100,000 MD steps, 1 bar of pressure, and a time step of 2 fs. The radial distribution had 1000 bins increasing the distance by .01 angstrom to cutoff of 10 angstrom. Figure 5 shows the RDFs of the three temperatures below the melting point and Figure 6 shows the RDFs of the three temperatures above the melting point.

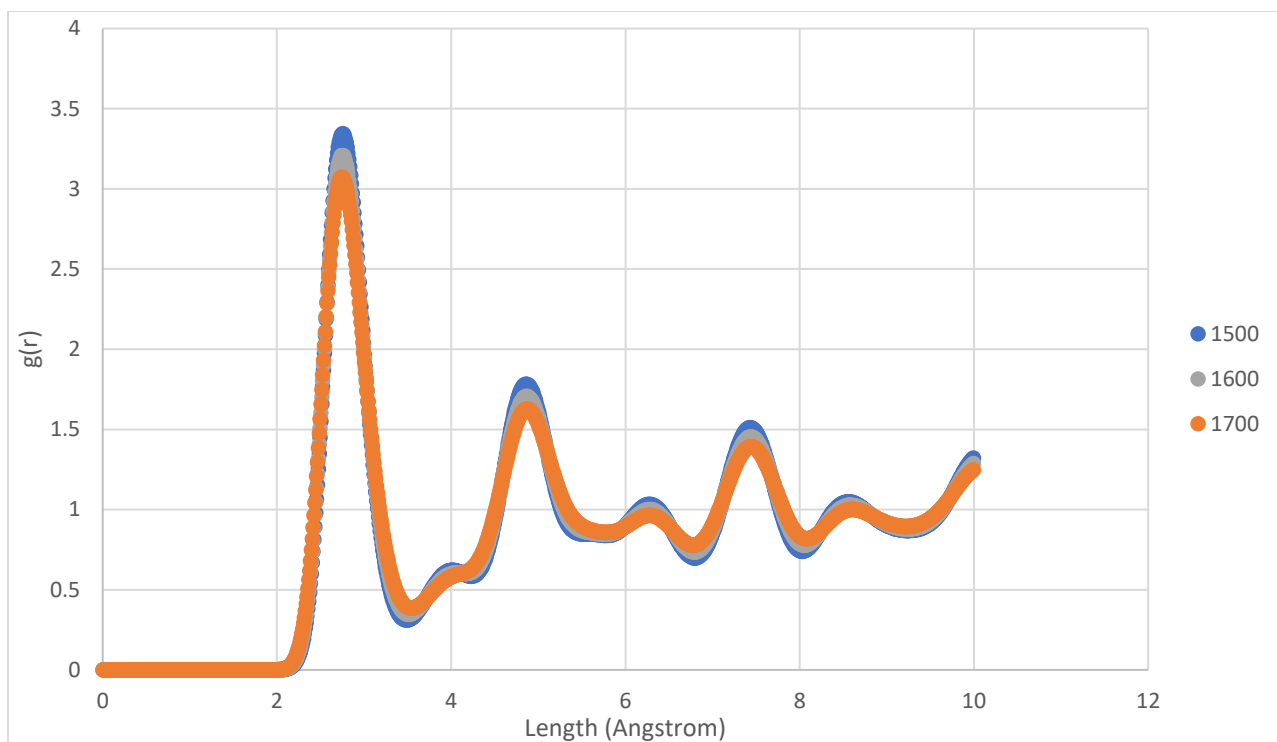


Figure 5: RDF Plot of 1500,1600, and 1700 K MD Simulations

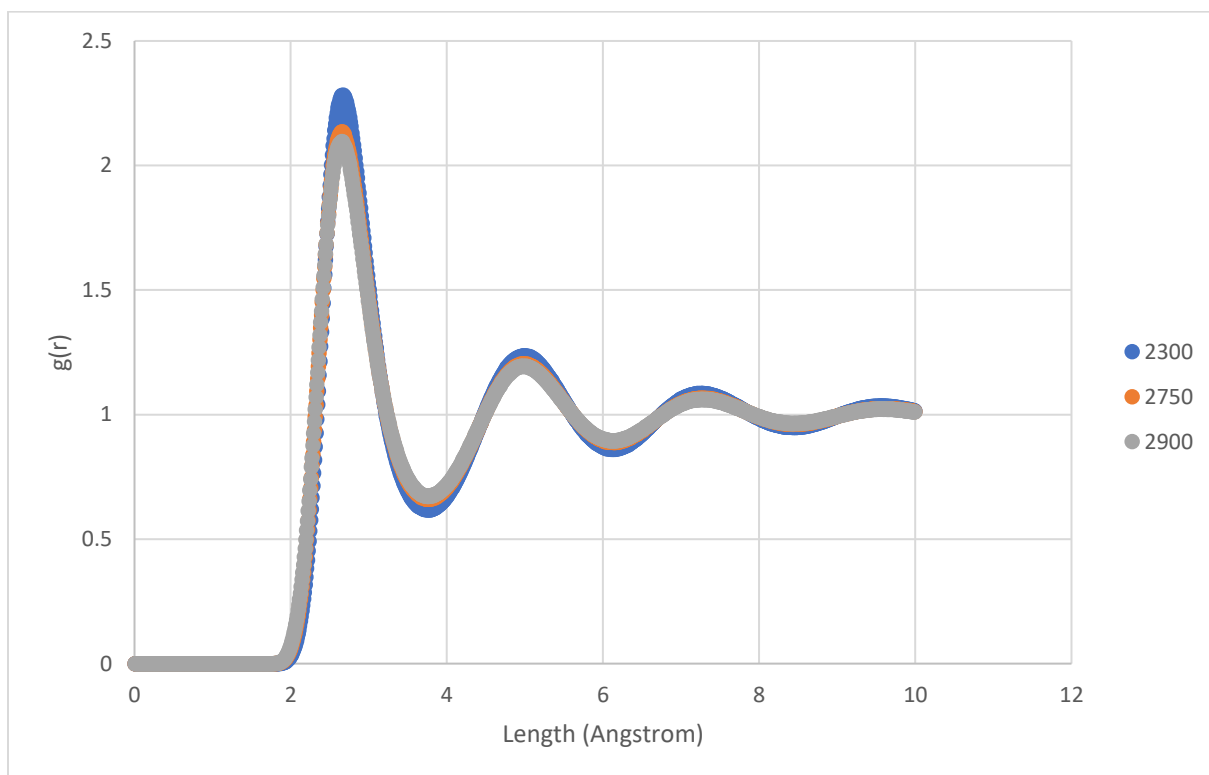


Figure 6: RDF Plot of 2300, 2750, and 2900 K MD Simulations



These graphs confirm our expectations from part B as the colder the temperatures in both graphs the sharper the peaks are and the hotter the temperature the more smoothed out the peaks become. Also, all of the RDFs that were calculated below the melting point have much more pronounced peaks than the RDFs above the melting point.

### Problem 3: Self Diffusivity of Platinum

- A. a. To compute the average mean square displacement (MSD) the code seen in Figure 8 of Appendix A was used. Each simulation was ran with a supercell size of 10 x 10 x 10, a time step of 2 fs, and 100,000 MD steps. Three simulations were ran at 2300 degrees Kelvin and three were ran at 2500 degrees Kelvin. Finally, a different seed was used for the velocity initialization for each simulation so that we do not get the same results each time. The average MSD for each run was then calculated after the simulation finished as well as the total MSD average for the different temperatures. Using this total MSD average the error for each simulation was also calculated using the percent error formula where the total MSD average is assumed to be the true value. The average and total average of the MSD for each temperature can be seen in Table 4 and the errors of each run can be seen in Table 5.

Table 4: Average and Total Average Mean Square Displacement of 2300 K and 2500 K Simulations

	MSD Average of Run			Total Average
	1	2	3	
Temperature (K)				
2300	.425	.955	.559	.646
2500	109.325	153.435	140.033	134.264

Table 5: Percent Error of Each MSD Simulation

Run	MSD Average	Error
2300 Run 1	.425	.342 (34.2%)
2300 Run 2	.955	.478 (47.8%)
2300 Run 3	.559	.135 (13.5%)
2500 Run 1	109.325	.186 (18.6%)
2500 Run 2	153.435	.143 (14.3%)
2500 Run 3	140.033	.043 (4.3%)

Finally, the MSD values obtained for each simulation were plotted versus time in ps and these graphs can be seen in Figures 7 and 8 below.

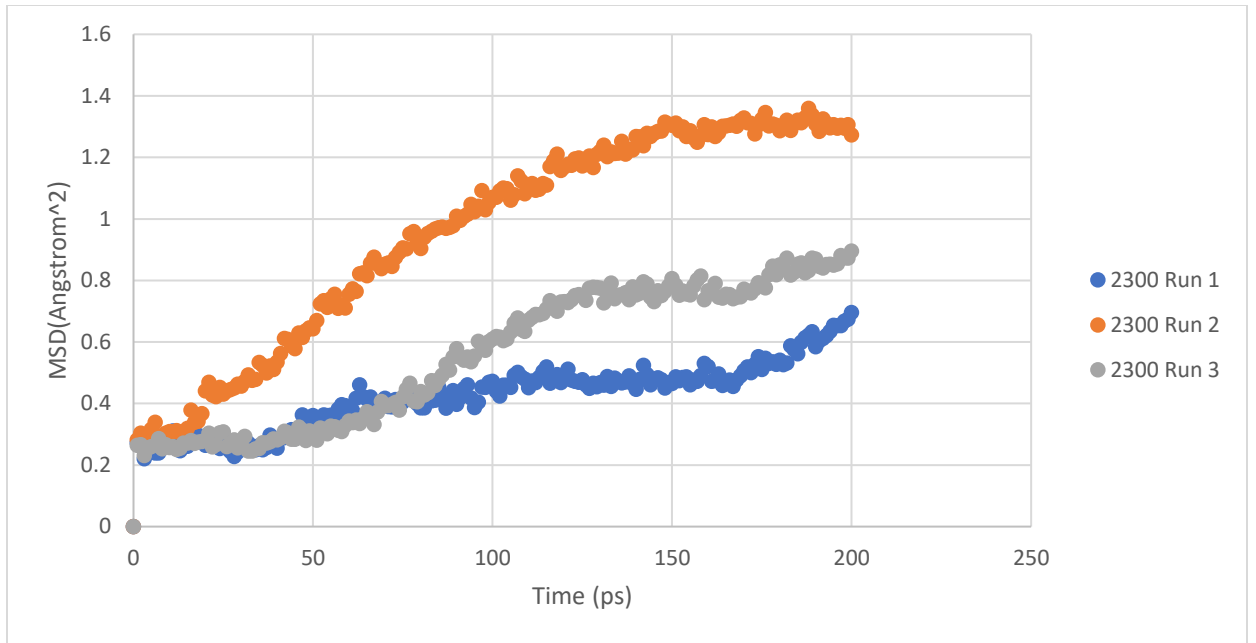


Figure 7: MSD vs Time for T=2300 K Simulations

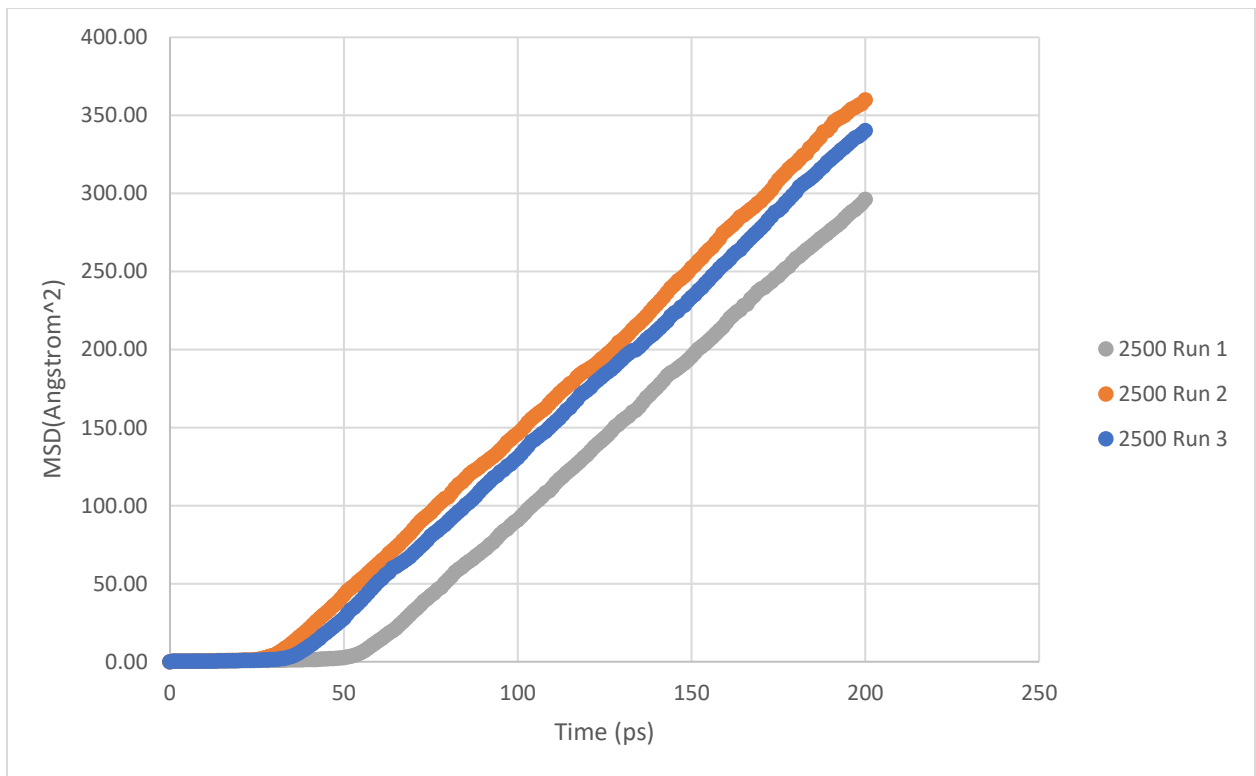


Figure 8: MSD vs Time for T=2500 K Simulations

b. When the temperature is equal to 2300 K equilibrium is reached approximately at 1 ps and when the temperature is at 2500 K equilibrium is reached on average at 40 ps. This can be determined from the data by looking at when the MSD begins to fluctuate around

the mean value which occurs around 1 ps for T=2300K and around 40 ps when T=2500K.

- B. a. Part A's calculations for MSD were repeated but this time the temperature was set to 1000 degrees Kelvin. Table 6 shows the average MSD of each run as well as the error. Also, the MSD calculations were plotted versus time for each of the runs and this can be seen in Figure 9.

Table 6: MSD Average for T=1000K and Total Average MSD Over All Runs

Run	MSD Average	Error
1	.0789	.005 (.5%)
2	.0796	.0038 (.38%)
3	.0793	0 (0%)
Total	.0793	n/a

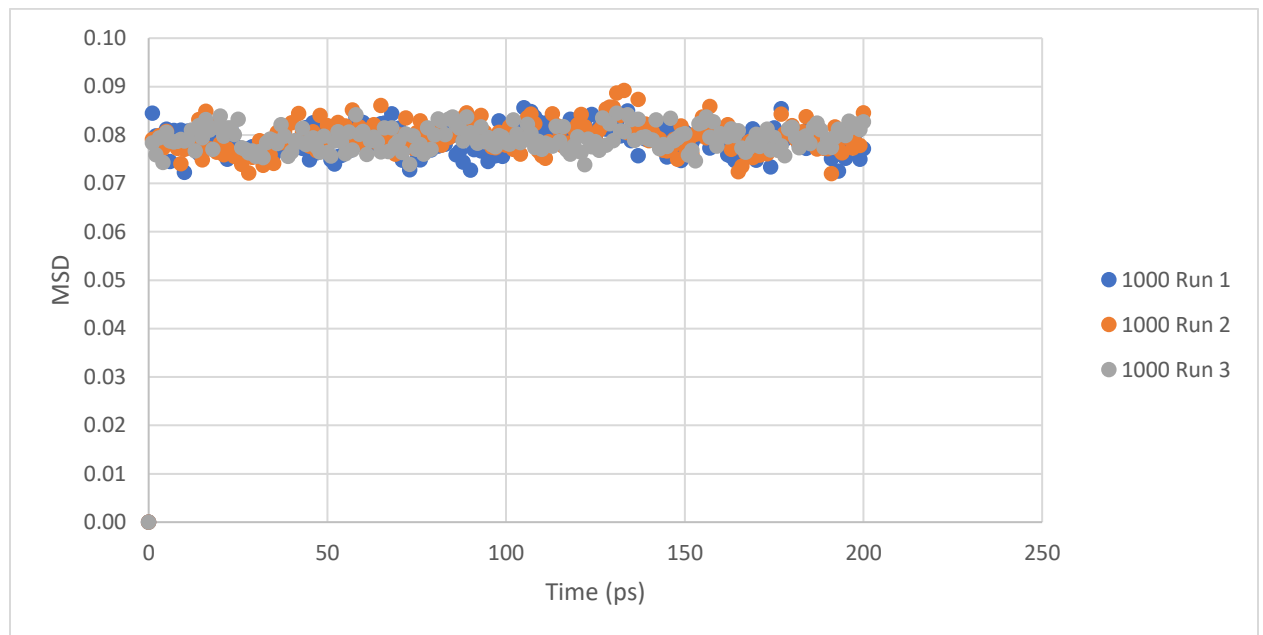


Figure 9: MSD vs Time for T=1000 K Simulations

- b. The MSD values when temperature is set to 1000 degrees Kelvin is essentially a line with no slope and only minor fluctuations where as both the T=2300 K and T=2500K MSD plots show that the MSD increase as time moves forward. This makes sense as at T=1000 K we are below the melting point and thus in a solid state so it is very hard for atoms to move around and displace themselves. Where as T=2300 and 2500 K are both above the melting point so atoms can move around move more freely and displace at much greater rates.

- C. a. Using the Einstein relation which states that the self-diffusion coefficient is the time derivative of the MSD values we can find the slope of the equilibrium part of our MSD vs time plots and that will give us the time derivative of our MSD calculations. The average slope of the T=2300K plots was  $4.092 \times 10^{-6}$  and the average slope of the T=2500K plots was .0042. In order to find  $D_{self}$  we then need to divide the slope by 1/6. This gives us  $D_{self} = 6.82 \times 10^{-7}$  when T=2300K and  $D_{self} = 7 \times 10^{-4}$  when T=2500K
- b. No its not possible to use the MSD results when T=1000K to determine the self-diffusion coefficient because there is no slope in the MSD vs time plot and thus the time derivative of MSD would be zero which cancels out the whole equation and makes  $D_{self} = 0$ .
- c. The general idea of the Arrhenius extrapolation is that we can not find room temperature diffusivities very easily so we instead find the diffusivity of the material at a high temperature and assume that the diffusion mechanism did not change. We can then use the Arrhenius law which states  $D = Ae^{-\frac{E_a}{k_B T}}$  to extrapolate the diffusivity from high temperatures to low temperatures. Through this method it would be possible for us to determine the diffusion coefficient of platinum at T=1000K from our simulation data at T=2300K and 2500K.

#### Problem 4: Questions Related to Molecular Dynamics

- A. In this project we used the mean square displacement method to compute diffusion coefficients but an alternative option would have been to use the velocity auto correlation functions also known as the Green-Kubo functions.
- B. Time correlation functions allow us to compare the similarity of our structure at two different points in time. One example of where time correlation functions are used is in the simulation of bond frequencies in which velocity auto correlation functions can be used to find the frequency of bond vibrations. Another example is in infrared spectroscopy where time correlation functions and more specifically velocity auto correlation functions can be used to simulate the vibrational modes and tell whether it has a dipole moment and thus is infrared active or not.
- C. Neighbor lists are used for finding atoms that are near/close by our central atom. The two examples we talked about in class were the Verlet list model and the linked cell list model. The Verlet list model uses a circular shell of radius  $rc$  which is defined as the cutoff potential. Another shell layer is added with width  $rs$  to the cutoff potential to create the full shell. The simulation then keeps track of where the atoms are and where they have moved. If they have moved a distance of  $rs$  or greater than they no longer are near by and thus do not need to be factored into the simulation. The linked cell list model cuts the simulation into cells with an edge length greater than or equal to  $rc$  where  $rc$  is the interaction potential. Each atom is then assigned a cell and we

then we know that the most important atoms for our simulations are in the neighboring cells/ which are within the interaction potential range  $r_c$ .

#### Problem 5: Equivalence of the Velocity and Position Verlet Methods

- A. The Verlet algorithm is symplectic so that means the Hamiltonian is conserved close to the Hamiltonian that describes the actual physical system and it is time reversible. Euler and Runge-Kutta methods are not time reversible nor symplectic which leads to large errors in energy conservation and thus are not suitable for molecular dynamic simulations.
- B. The integration error of the Verlet methods is on the magnitude of the fourth order  $\Delta t$ .
- C. We start with the position Verlet equation of:

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{F(t)}{2m} * \Delta t^2$$

$$v\left(t + \frac{\Delta t}{2}\right) = v(t) + \frac{1}{2}a(t)\Delta t^2$$

$$a(t + \Delta t) = -\left(\frac{1}{m}\right)\nabla V(r(t + \Delta t))$$

$$v(t + \Delta t) = v\left(t + \frac{\Delta t}{2}\right) + \left(\frac{1}{2}\right)a(t + \Delta t)\Delta t$$

$$v(t + \Delta t) = v(t) + \frac{F(t) + F(t + \Delta t)}{2m} \Delta t$$

The final equation is the velocity Verlet algorithm so this shows that they are equivalent

Collaborators: None

## Appendix A:

```
# -----  
# Columbia University - CHEN E4880 - Project 4  
# LAMMPS input file for problem 1  
# Molecular Dynamics Simulations  
# -----  
  
# Initial set-up  
clear  
units      metal      # eV, Angstrom, g/mol, ps, K, bar  
dimension  3  
boundary   p p p  
atom_style atomic  
  
# Create atoms using the optimized lattice parameter from project 1  
lattice     fcc 3.924 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1  
region      simulation_box block 0 1 0 1 0 1 units lattice  
create_box  1 simulation_box  
create_atoms 1 box  
  
# Create supercell  
replicate   4 4 4  
  
# Atomic mass (in amu)  
mass 1 195.084  
  
# EAM potential  
#   required file:  
pair_style  eam  
pair_coeff  * * Pt-Adams1989.eam  
  
# Configure output of thermodynamic quantities  
thermo      10  
thermo_style custom step time temp etotal pe ke  
thermo_modify norm yes # normalize extensive quantities by number of atoms  
  
# EoM integration  
run_style   verlet  
timestep    0.020 # ps  
  
# Neighbor list  
neighbor 1.0 bin  
neigh_modify once no every 1 delay 0 check yes  
  
# NVE simulations  
velocity all create 5000.0 123456 rot yes mom yes dist gaussian  
fix 1 all nve  
run 5000
```

Figure 1: LAMMPS Code Used for Problem 1

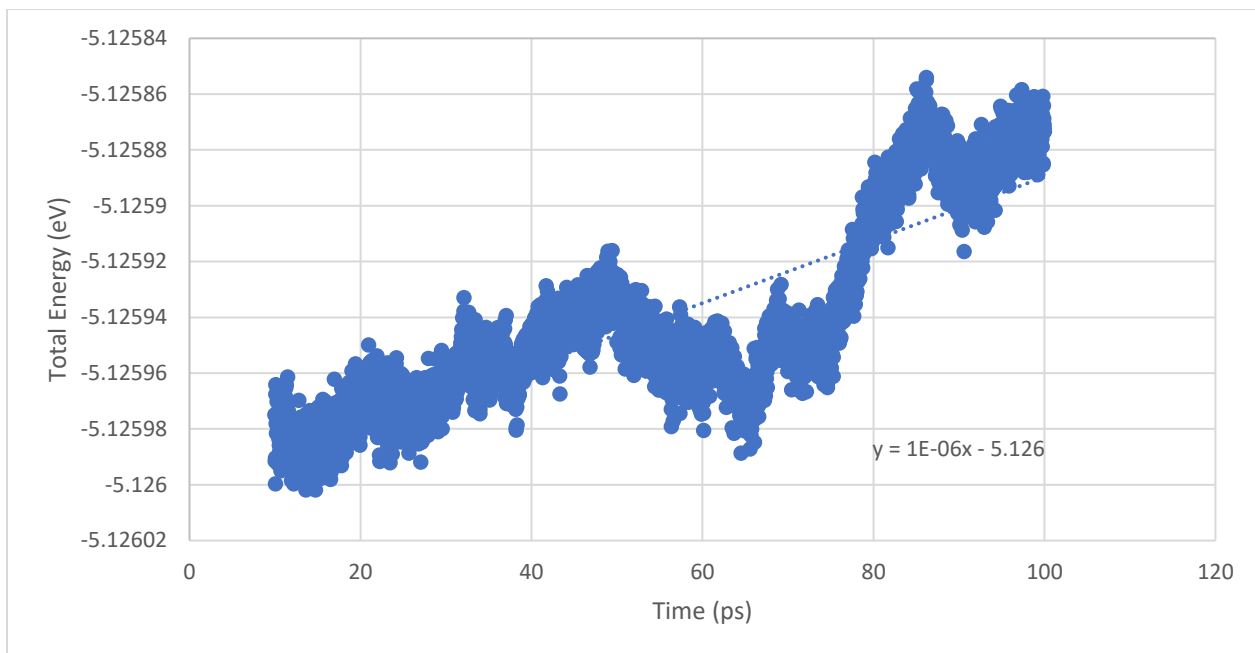


Figure 2: Time vs Total Energy for 2fs Time Step (Problem 1Ac)

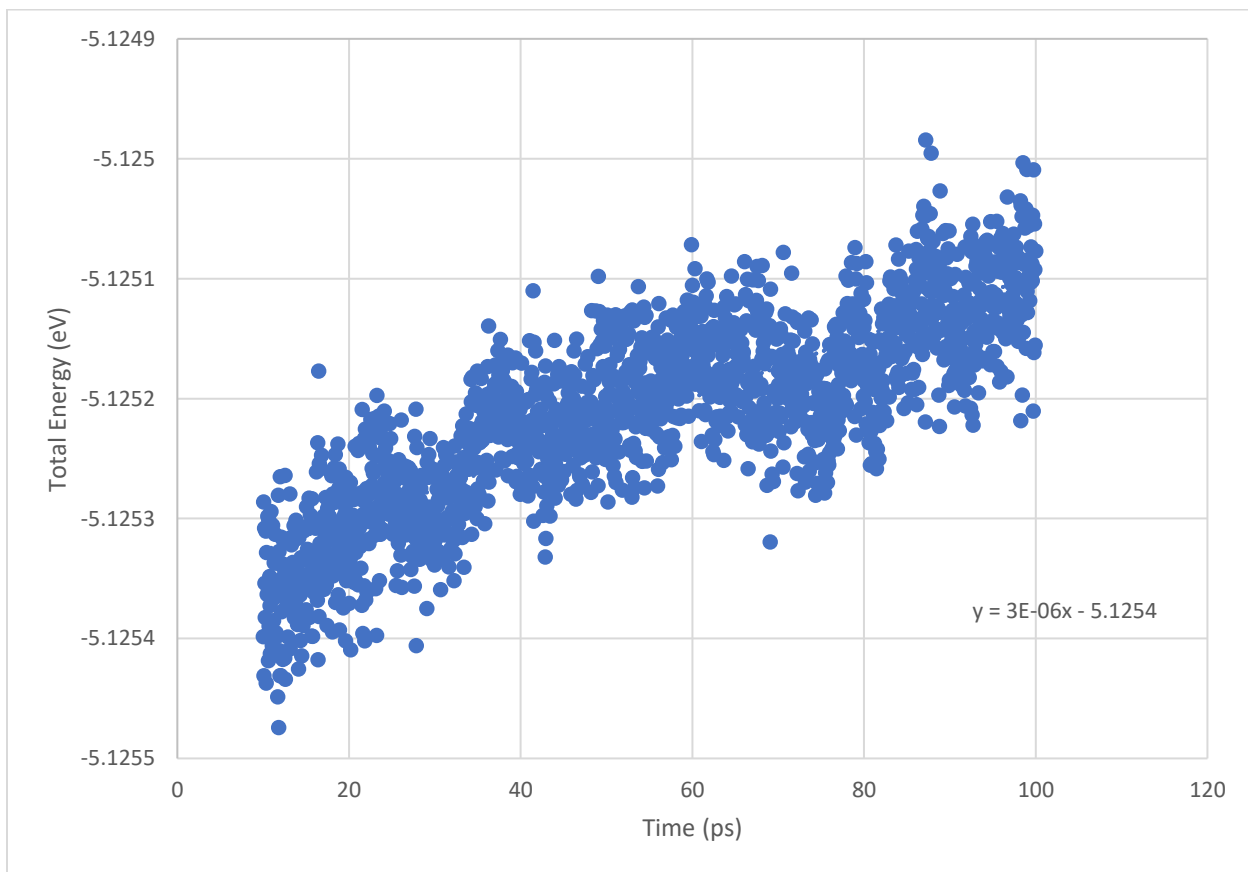


Figure 3: Time vs Total Energy for 5fs Time Step (Problem 1Ac)

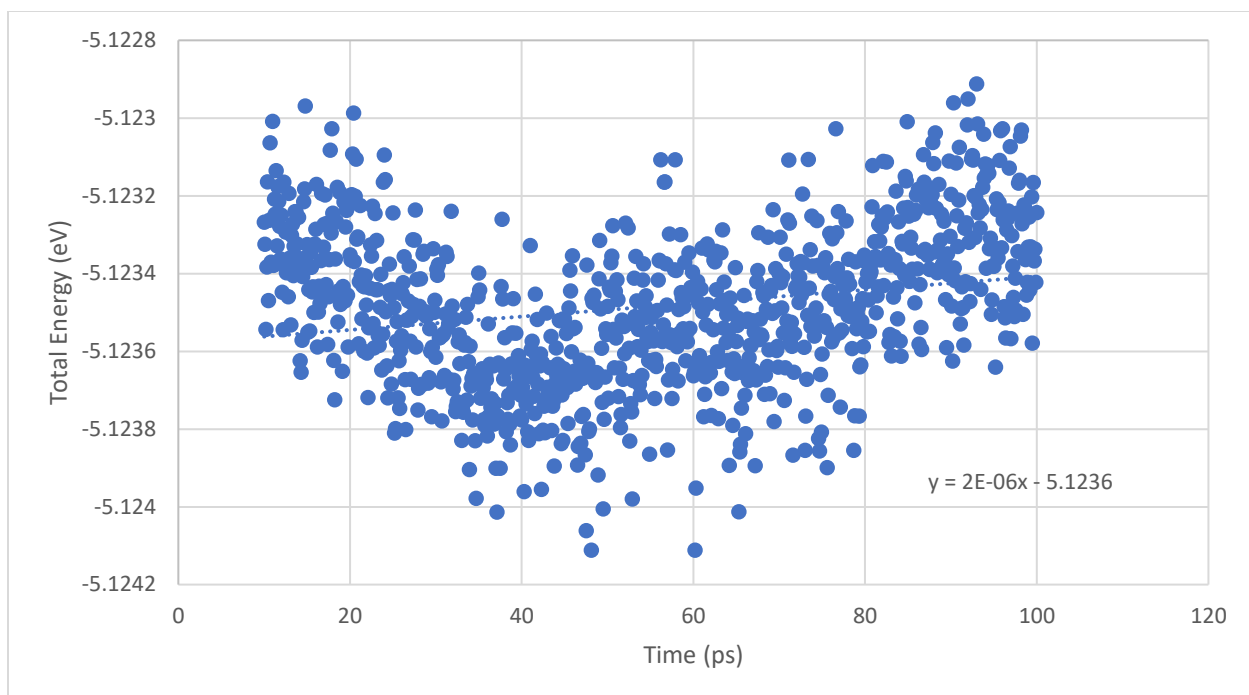


Figure 4: Time vs Total Energy for 10fs Time Step (Problem 1Ac)

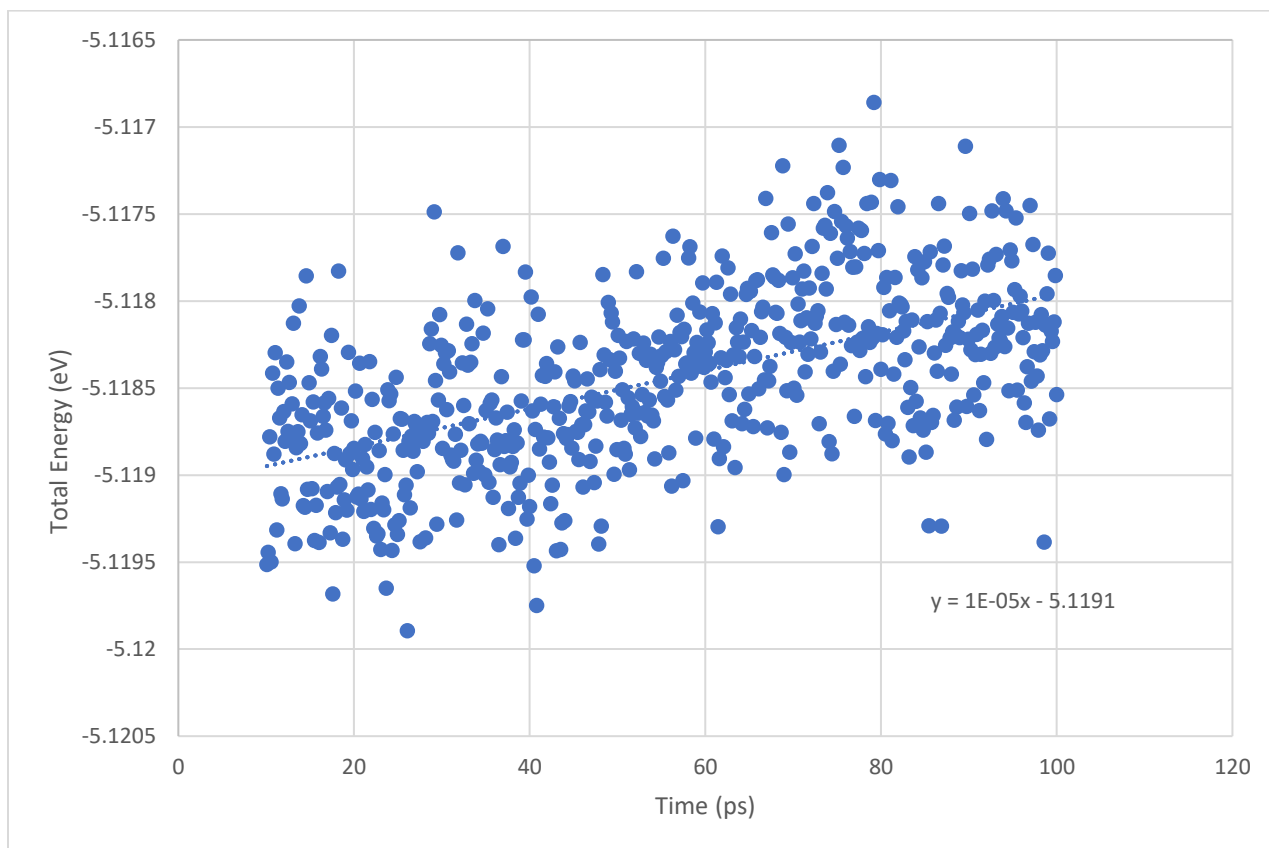


Figure 5: Time vs Total Energy for 16fs Time Step (Problem 1Ac)



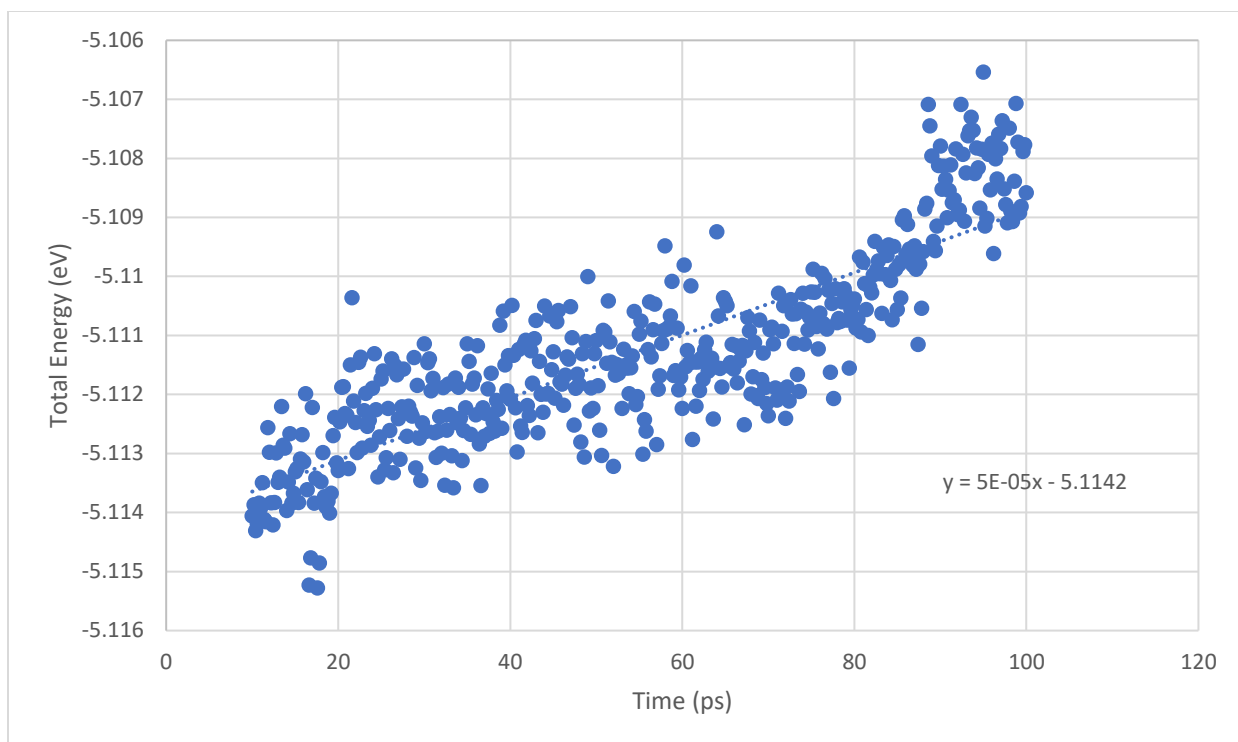


Figure 6: Time vs Total Energy for 20 fs Time Step (Problem 1Ac)

```

# -----
# Columbia University - CHEN E4880 - Project 4
# LAMMPS input file for problem 2
# Molecular Dynamics Simulations
# -----

variable Nrun      equal 100000
variable Ne        equal 10
variable T_ext     equal 2300.0 # Kelvin
variable P_ext     equal 1.0 # bar
variable seed      equal 123456

#-----

# Initial set-up ...
clear
units metal
dimension 3
boundary p p p
atom_style atomic

# Create atoms ...
lattice fcc 3.924 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1
region simulation_box block 0 1 0 1 0 1 units lattice
create_box 1 simulation_box
create_atoms 1 box

# Create supercell ...
replicate 10 10 10

# Atomic species ...
mass 1 195.084

# EAM potential ...
pair_style eam
pair_coeff * * Pt-Adams1989.eam

# EoM integration ...
run_style verlet
timestep .002 #ps

# Neighbor list ...
neighbor 1.0 bin
neigh_modify once no every 1 delay 0 check yes

# Configure output of thermodynamic quantities.
# This time include pressure and box lengths.
thermo_style custom step time temp etotal pe ke press lx ly lz
thermo_modify norm yes
variable Nf      equal ${Nrun}/20
thermo ${Nf}

# Compute Pt-Pt RDF during simulation
# Have to increase the cutoff including neighbor list skin radius
comm_modify cutoff 11.0
compute rdf all rdf 1000 1 1 cutoff 10.0
variable Nr_rdf equal 0.5*${Nrun}/${Ne}
fix rdf all ave/time ${Ne} ${Nr_rdf} ${Nrun} c_rdf[*] file rdf.dat mode vector

# Run NPT simulation
variable T2      equal ${T_ext}*2
velocity all create ${T2} ${seed} rot yes mom yes dist gaussian
fix 1 all npt temp ${T_ext} ${T_ext} 4.0 iso ${P_ext} ${P_ext} 1000.0
run ${Nrun}

```

Figure 7: LAMMPS Code Used For Problem 2

```

# -----
# Columbia University - CHEN E4880 - Project 4
# LAMMPS input file for problem 3
# Molecular Dynamics Simulations
# -----

variable Nrun      equal 100000
variable Ne        equal 10
variable T_ext     equal 1000.0 # Kelvin
variable P_ext     equal 1.0 # bar
variable seed      equal 967345

# -----

# Initial set-up ...
clear
units metal
dimension 3
boundary p p p
atom_style atomic

# Create atoms ...
lattice fcc 3.924 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1
region simulation_box block 0 1 0 1 0 1 units lattice
create_box 1 simulation_box
create_atoms 1 box

# Create supercell ...
replicate 10 10 10

# Atomic species ...
mass 1 195.084

# EAM potential ...
pair_style eam
pair_coeff * * Pt-Adams1989.eam

# EoM integration ...
run_style verlet
timestep .002 #ps

# Neighbor list ...
neighbor 1.0 bin
neigh_modify once no every 1 delay 0 check yes

# Configure output of thermodynamic quantities.
# This time include pressure and box lengths.
thermo_style custom step time temp etotal pe ke press lx ly lz
thermo_modify norm yes
variable Nf      equal ${Nrun}/20
thermo ${Nf}

# Compute the MSD of Pt atoms
compute msd all msd com yes
fix msd all ave/time 1 1 500 c_msd[4] file msd.dat

# run NVT simulation
variable T2 equal ${T_ext}*2
velocity all create ${T2} ${seed} rot yes mom yes dist gaussian
fix 1 all nvt temp ${T_ext} ${T_ext} 4.0
run ${Nrun}

```

Figure 8: LAMMPS Code Used for Problem 3