TABLE OF CONTENTS

1.PROBLEM

It was aimed to analyze vibrations of a material in a Lennard-Jones potential and calculate linear expansion coefficient of the material. In order to get approximate real values, hydrogen molecule was chosen as material since its Lennard-Jones potential formula is known well enough. Thus, behaviour of hyrogen molecule in Lennard-Jones potential was determined, and expansion coeffiecient was found in the project.

2.THEORY

2.1 Lennard-Jones Potential

The potential energy of system of two atoms is described by

$$U(r) = -\frac{A}{r^m} + \frac{B}{r^n}$$

where r is the distance between atoms, and A and B are constants.

For diatomic molecules, the potential energy is

$$U(r) = -\frac{A}{r^{12}} + \frac{B}{r^6}$$

This kind of potential energy functions called as Lennard-Jones potential [1]. The graph of Lennard-Jones potential is shown below.

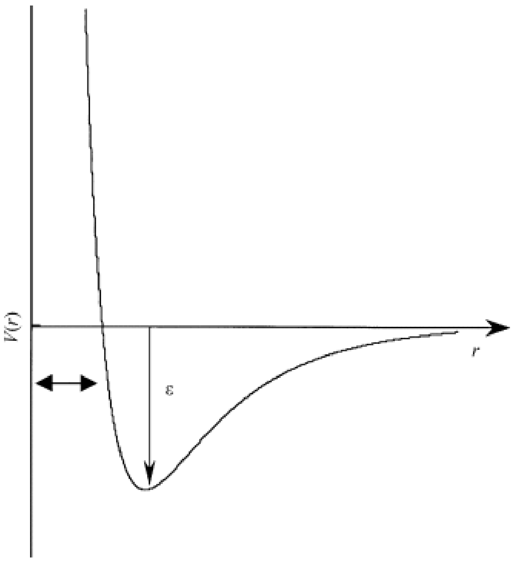


Figure 1: Lennard-Jones Potential

2.2 Thermal Expansion

When temperature of a material is increased, it expands. This expansion is due to increase in average spacing between atoms or molecules of material. The shift in average spacing can be detect easily from the graph of potential energy, Lennard-Jones potential, versus distance which is bond length as temperature rises [2].

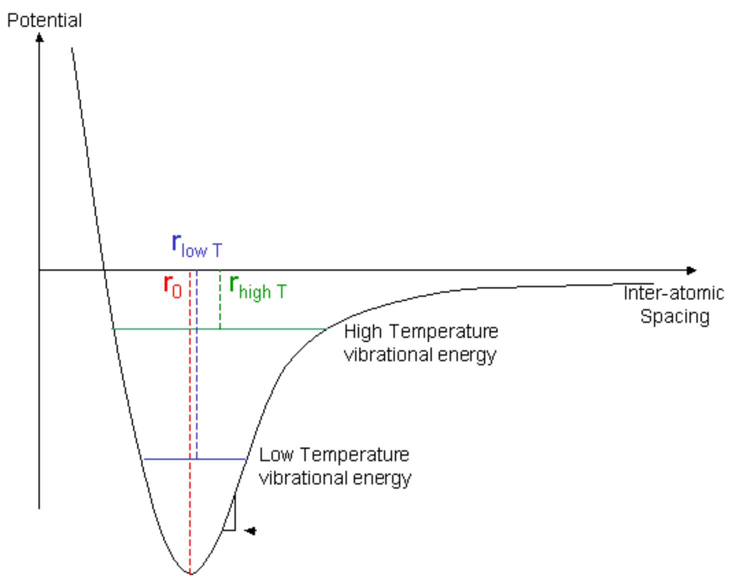


Figure 2: LJ Potential versus Bond Length [2]

In Fig 2, r_0 is the bond length at equilibrium. At r_0 the material has minimum energy, also. As the thermal energy of the system rises, bond length shifts to the right. The atoms of material vibrates at distance r_{highT} .

When temperature is increased by an amount T Kelvin, given energy is

$$K = \frac{1}{2} m v^2 = \frac{3}{2} k_b T$$

where k_b is Boltzmann's constant.

For expansion coefficient,

$\alpha \Delta T = \Delta L/L$

$\alpha = \Delta L/(L*\Delta T) = \frac{1}{r} \frac{drave}{dT}$ where r is the r_0 in Lennard-Jones potential..

r = 0.074 nm bond length between hydrogen atoms.

2.3 Hydrogen Molecule in LJ Potential

As a material Hydrogen molecule which is diatomic was chosen. For H_2 Lennard-Jones potential is given by

$$U(r) = \frac{0.124x10^{-120}}{r^{12}} - \frac{1.488x10^{-60}}{r^{6}} \text{ eV} \quad [1]$$

Constants A and B are in the units of eV. m^{12} and eV.m^{6} . If they are converted to nanometer,

$$U(r) = \frac{0.124x10^{-12}}{r^{12}} - \frac{1.488x10^{-6}}{r^{6}} \text{ eV}$$

Now, the distance at minimum energy, where T=0K, can be calculated from derivative of the LJ potential, which is also force.

$$r_0 = \frac{dU(r)}{dr} = -F(r) = 0 = 0.07 \text{ nm}$$

At U(r) = 0, r is approximately 0.064 nm (r_{min})

It is also needed to determine a limit meaning maximum energy since the LJ potential goes to infinity at higher distances. For maximum energy, I chose distance where second derivative of potential energy is equal to 0.

$$r_{max} = \frac{d^2U(r)}{dr^2} = 0 = 0.079 \text{ nm}$$

Bolztmann's constant is 8.6173×10^{-5} eV K^{-1} [3]

3. NUMERICAL METHODS

In order to calculate the expansion coefficient, it is needed to find average spacing in time as temperature increases. To find position of the molecule at minimum energy, at zero potential, and at limit potential and initial and final positions of the molecule in given energy Bi Section Method was used. After finding the all required positions, 4th Order Runge-Kutta Method was used to solve the differential equation of the motion. By appliying the method, positions in time and velocities were calculated.

3.1 Bi Section Method

This method uses Intermediate Value Theorem of Calculus. For continuos function f(x), if f(a) * f(b) < 0 or f(a) * f(b) > 0 then there is at least one root between a and b [4]. In the method, first interval is chosen by guess. Then it is checked whether there is a root between the interval. If there is a root, midpoint is calculated. Values of function for each interval points are multiplied by value of function for midpoint. According to results, interval is changed. The process continues like this until finding the root.

For r_0, a = 0.06 b = 0.073 were chosen since r0 is theoretically equal to 0.07 nm.

For r_{min}, a = 0.06 b = r_0 were chosen since r_{min} is between this interval theoretically.

For r_{max}, a = r_0 b = 0.08 were chosen since rmax is between this interval theoretically.

Then for initial and final positions when thermal energy is given to the system,

a = rmin, b = 0.07 for r_i and a = 0.071, b = 0.078 for r_f

with precision 0.01.

3.2 4th Order Runge-Kutta Method

This method is useful in order to solve second order differential equations. First, it is necessary separate 2nd order differential equation into two separate 1st order differential equations. Then constants k_i are calculated in the order. Using constants in a related relationship, variable value can be found in given time.

For H_2 molecule, the equation is

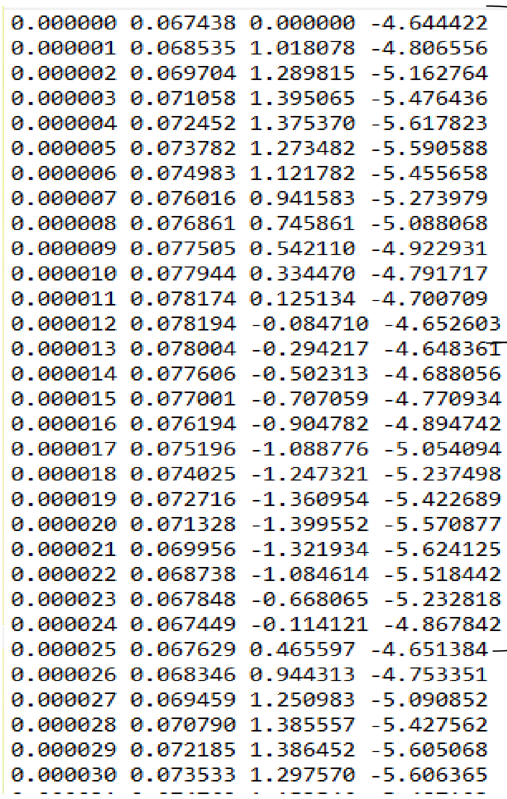
$F(r) = m\ddot{r}$

Now separating the equation,

$v = \frac{dr}{dt}$, $F(r) = \frac{dv}{dt} = -\frac{dU(r)}{dr}$ also

4.RESULTS

Data taken showed oscillations of hydrogen molecule. Below is the some data of hydrogen molecule at 0.1 Kelvin.



Time (s)	r (nm)	v (nm/s)	U (eV)
0.000000	0.067438	0.000000	-4.644422
0.000001	0.068535	1.018078	-4.806556
0.000002	0.069704	1.289815	-5.162764
0.000003	0.071058	1.395065	-5.476436
0.000004	0.072452	1.375370	-5.617823
0.000005	0.073782	1.273482	-5.590588
0.000006	0.074983	1.121782	-5.455658
0.000007	0.076016	0.941583	-5.273979
0.000008	0.076861	0.745861	-5.088068
0.000009	0.077505	0.542110	-4.922931
0.000010	0.077944	0.334470	-4.791717
0.000011	0.078174	0.125134	-4.700709
0.000012	0.078194	-0.084710	-4.652603
0.000013	0.078004	-0.294217	-4.648361
0.000014	0.077606	-0.502313	-4.688056
0.000015	0.077001	-0.707059	-4.770934
0.000016	0.076194	-0.904782	-4.894742
0.000017	0.075196	-1.088776	-5.054094
0.000018	0.074025	-1.247321	-5.237498
0.000019	0.072716	-1.360954	-5.422689
0.000020	0.071328	-1.399552	-5.570877
0.000021	0.069956	-1.321934	-5.624125
0.000022	0.068738	-1.084614	-5.518442
0.000023	0.067848	-0.668065	-5.232818
0.000024	0.067449	-0.114121	-4.867842
0.000025	0.067629	0.465597	-4.651384
0.000026	0.068346	0.944313	-4.753351
0.000027	0.069459	1.250983	-5.090852
0.000028	0.070790	1.385557	-5.427562
0.000029	0.072185	1.386452	-5.605068
0.000030	0.073533	1.297570	-5.606365

1st oscillation, as change in the sign of the velocity and position indicate.

2nd

Table 1: Data at 0.1 K

Data of other temperature values ranging from 0.1 K to 2.0 K show oscillations of hydrogen molecule, as well.

It was found that hydrogen molecule provides Lennard-Jones potential. When first back-and forth movement of hydrogen molecule is plotted, this fact is seen clearly.

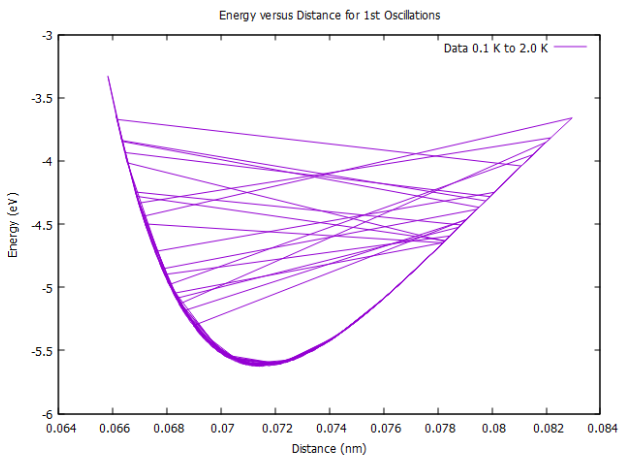
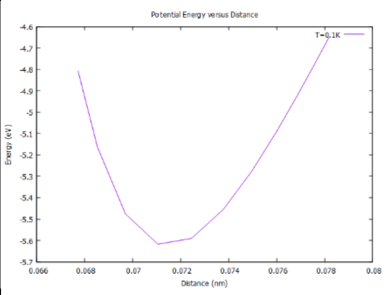
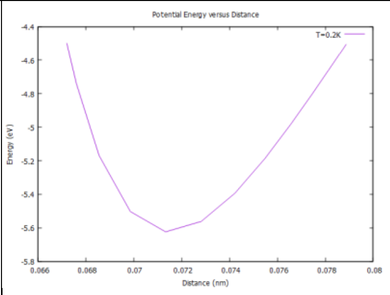


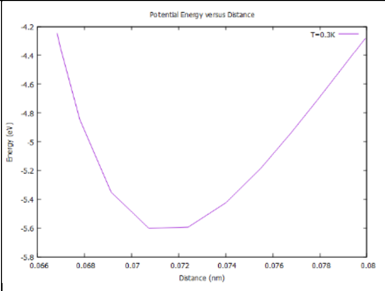
Figure 3: Distance vs Potential Energy of 1st Oscillations at T=0.1 K to 2.0 K



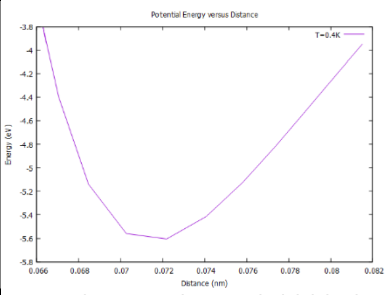
At T=0.1 K, t=0 to t=0.000012 s



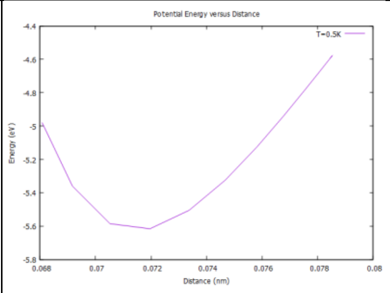
At T=0.2 K, t=0 to t=0.000012 s



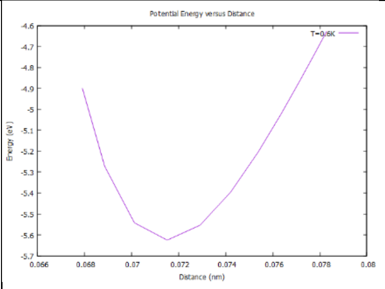
At T=0.3 K, t=0 to t=0.000012 s



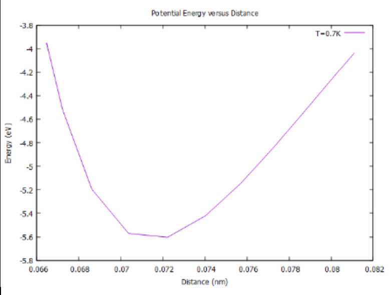
At T=0.4 K, t=0 to t=0.000012 s



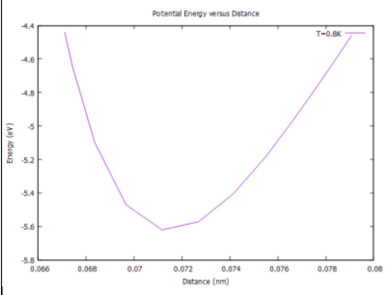
At T=0.5 K, t=0 to t=0.000012 s



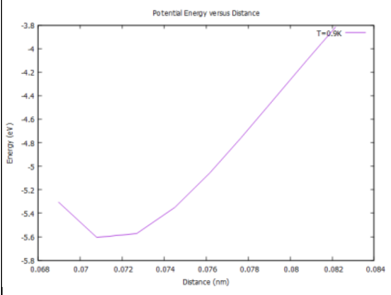
At T=0.6 K, t=0 to t=0.000012 s



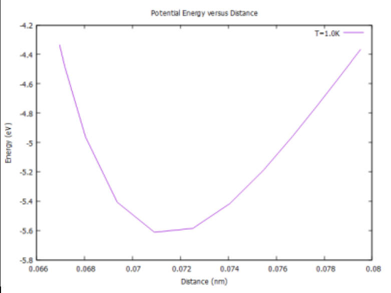
At T=0.7 K, t=0 to t=0.000012 s



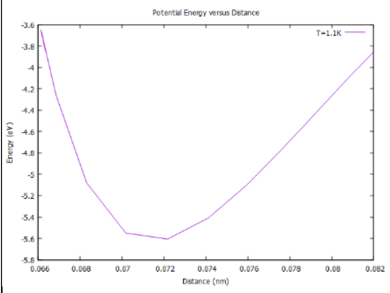
At T=0.8 K, t=0 to t=0.000012 s



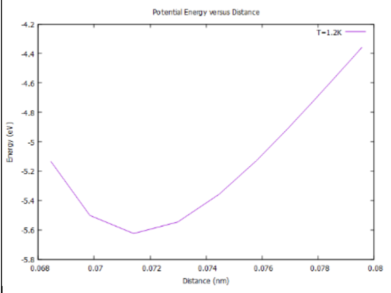
At T=0.9 K, t=0 to t=0.000012 s



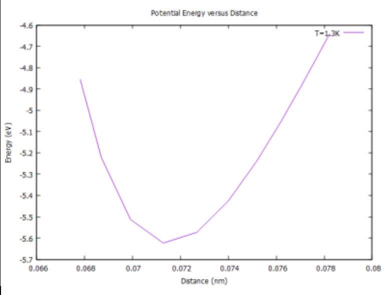
At T=1.0 K, t=0 to t=0.000012 s



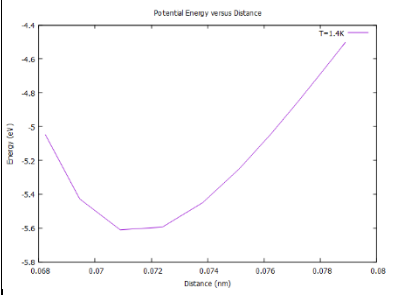
At T=1.1 K, t=0 to t=0.000012 s



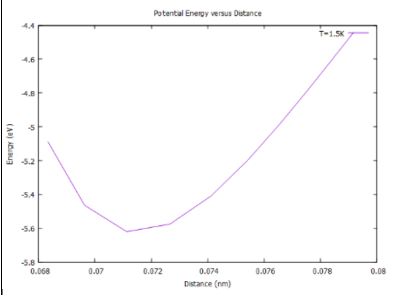
At T=1.2 K, t=0 to t=0.000012 s



At T=1.3 K, t=0 to t=0.000012 s



At T=1.4 K, t=0 to t=0.000012 s



At T=1.5 K, t=0 to t=0.000012 s

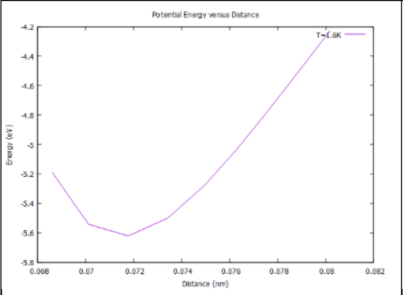
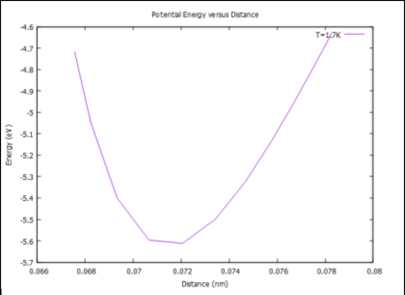
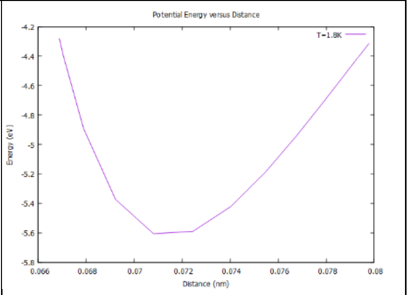
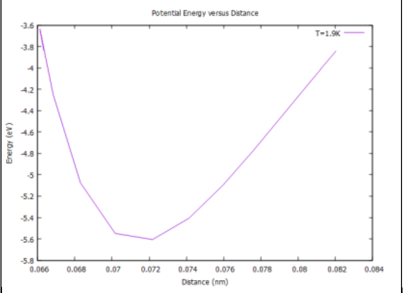
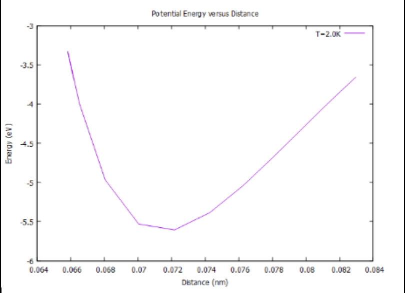
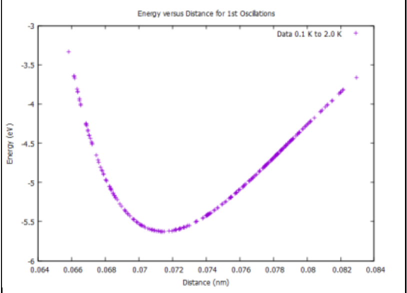
 At T=1.6 K, t=0 to t=0.000012 s	 At T=1.7 K, t=0 to t=0.000012 s	 At T=1.7 K, t=0 to t=0.000012 s
 At T=1.9 K, t=0 to t=0.000012 s	 At T=2.0 K, t=0 to t=0.000012 s	 T=0.1K to 2.0 K

Table 2: Graphs for 1st Oscillations at Temperatures 0.1 K to 2.0 K

When all data gathered from given temperature plotted as distance versus energy, the graph proves bound motion of hydrogen molecule in Lennard-Jones Potential. Here some graphs proving bound motion.

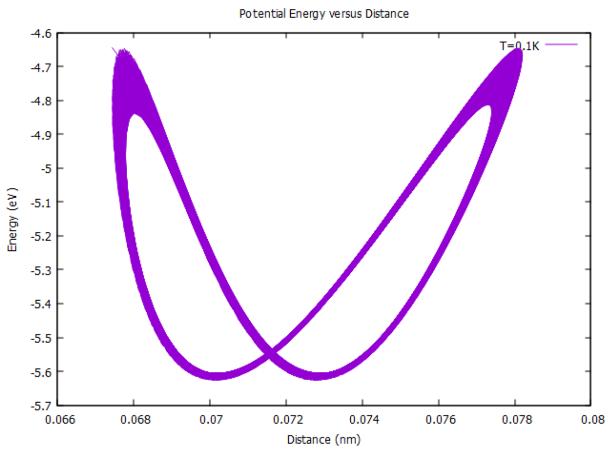


Figure 4: r vs U at T=0.1 K

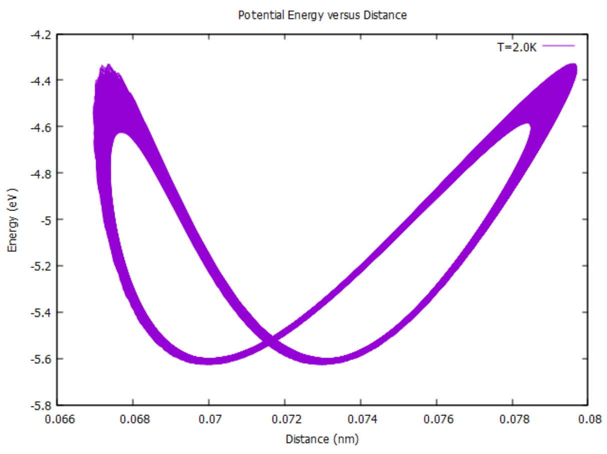
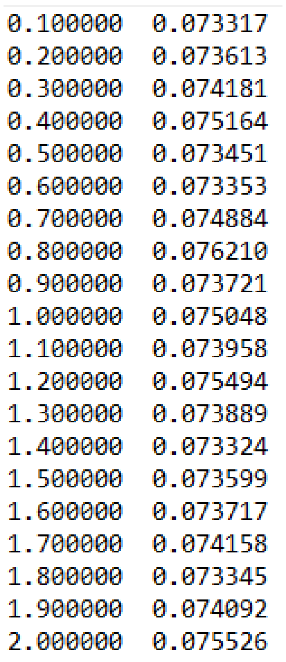


Figure 5: r vs U at T=2.0 K

Data for average spacing at different temperatures ranging from 0.1 K to 2 K,



Temp. (K)	r_{ave} (nm)
0.100000	0.073317
0.200000	0.073613
0.300000	0.074181
0.400000	0.075164
0.500000	0.073451
0.600000	0.073353
0.700000	0.074884
0.800000	0.076210
0.900000	0.073721
1.000000	0.075048
1.100000	0.073958
1.200000	0.075494
1.300000	0.073889
1.400000	0.073324
1.500000	0.073599
1.600000	0.073717
1.700000	0.074158
1.800000	0.073345
1.900000	0.074092
2.000000	0.075526

Table 3: T vs r_{ave}

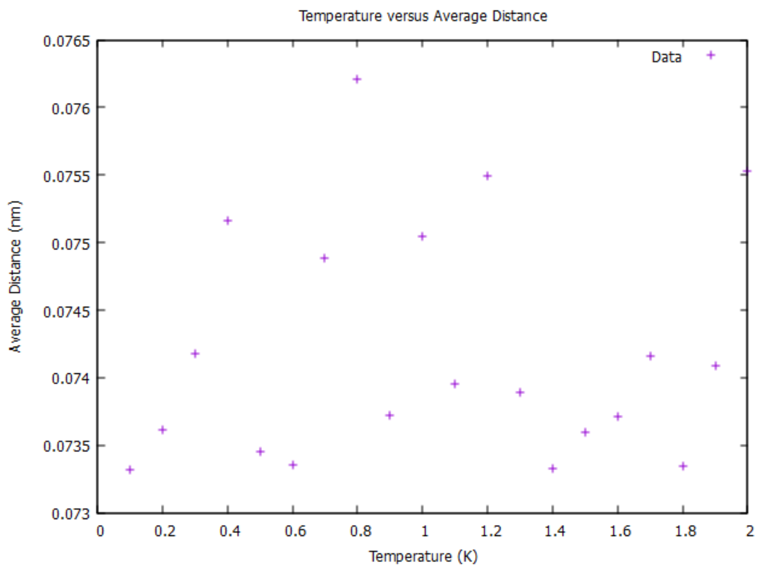
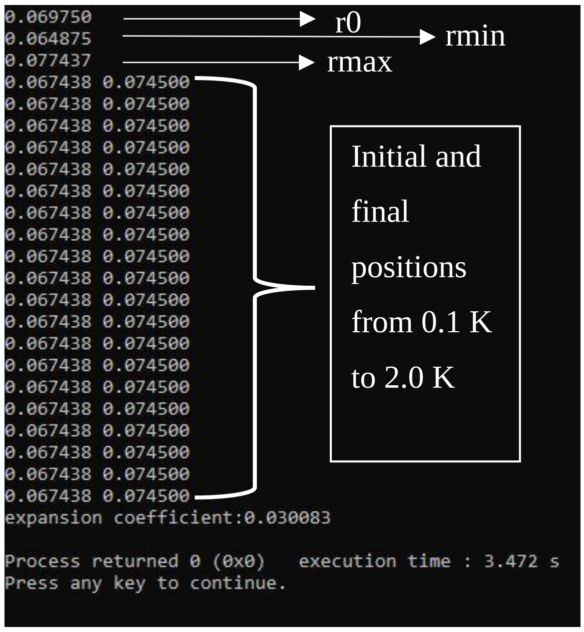


Figure 6: Temperature versus Average Spacing



```
0.069750
0.064875
0.077437
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
0.067438 0.074500
expansion coefficient:0.030083

Process returned 0 (0x0)   execution time : 3.472 s
Press any key to continue.
```

Figure 7: Output of the Program

Since Boltzmann constant is small, which is 8.6173×10^{-5} eV K^{-1} , code gives initial and final positions same at different temperatures. Due to this fact there are some shifts in average positions at different temperatures indicated by Table 3. This can be understood also from images in Table 2. For instance, at T=0.5 K, left hand side starts a less value than expected. This affect the result of average spacing. To deal with the problem, average of average spacing is calculated. Then change in temperature, r0, and theoretical bond-length were used to determine expansion coefficient.

By Fig 6, $\alpha = 0.030083\ K^{-1}$

5. ALGORITHM

- Define U(r), F(r), $\ddot{U}$, v(t, v), and F(r, t, v)
- Find r0 where U is minimum
- Find rmin where U=0
- Find rmax where $\ddot{U}$=0
- Open loop for Temperature increments
  - Find roots of U for given temperature, ri&rf
  - Do time steps from ri to rf
  - Calculate rave for given temperature
- Calculate expansion coefficient

6. CODE

```c
#include <stdio.h>

#include <stdlib.h>

#include <math.h>


double LJ_Potential(double r); // For H2 Molecule

double Force(double r);

double Force_RK(double t, double r, double v); // For Runge-Kutta calculations

double Velocity(double t, double v);

double D_Force(double r); // 2nd derivative of potential to determine a limit potential
value


int main(){


FILE *fp1 = fopen("Tvsrave.txt", "w");
```

```c
if (fp1 == NULL)
{
 puts("Error opening the file");
 return -1;
}

FILE *fp2 = fopen("FP_Data.txt", "w");
if (fp2 == NULL)
{
 puts("Error opening the file");
 return -2;
}

FILE *fp3 = fopen("One_Vib.txt", "w");  //To see LJ Potential graph shape in one period of vibraton of the molecule.
if (fp3 == NULL)
{
 puts("Error opening the file");
 return -3;
}

double S=(8.62e-5); //Boltzmann constant in unit eV/K
int i,j,l, N, T=10000, p=0;
double a, b, c, d, eps, bahalf, dchalf, r0, rmax, rmin, ri, rf, ri_nu, rf_nu, rsum, rave[20], Temperature[20], Umin, U_limit, U;
double r, rave_sum=0, dT, drave, ec, h, vn=0, rn, tn[T], vn_plus1[T], rn_plus1[T], k1v, k2v, k3v, k4v, k1r, k2r, k3r, k4r;


//Finding r0 by calculating Umin, at equilibrium
a = 0.06;
b = 0.073;
```

```
eps = 0.01;
if(Force(a)*Force(b) >= 0){
 printf("Choose a different interval");
   return 1;
 }
while( b-a > eps){
 bahalf = (a+b)/2;
 if(Force(bahalf)*Force(a) < 0){
   b = bahalf;
 }
 else if(Force(bahalf)==0){
   break;
 }
 else{
   a = bahalf;
 }
}
r0 = (a+b)/2;
printf("%lf\n", r0);


//Finding rmin by calculating root of potential, at U=0
a = 0.06;
b = r0;
if(LJ_Potential(a)*LJ_Potential(b) >= 0){
 printf("Choose a different interval");
   return 1;
 }
while( b-a > eps){
 bahalf = (a+b)/2;
 if(Force(bahalf)*LJ_Potential(a) < 0){
    b = bahalf;
  }
  else if(LJ_Potential(bahalf)==0){
    break;
  }
  else{
    a = bahalf;
  }
}
rmin = (a+b)/2;
printf("%lf\n", rmin);


//Finding rmax by calculating root of second derivative of potential
a = r0;
b = 0.08;
if(D_Force(a)*D_Force(b) >= 0){
  printf("Choose a different interval");
    return 1;
  }
while( b-a > eps){
  bahalf = (a+b)/2;
  if(Force(bahalf)*D_Force(a) < 0){
    b = bahalf;
  }
  else if(D_Force(bahalf)==0){
    break;
  }
  else{
    a = bahalf;
  }
```

```
}
rmax = (a+b)/2;
printf("%lf\n", rmax);


Umin = LJ_Potential(r0);
U_limit = LJ_Potential(rmax);
for(l=0; l<20; l++){   // T=0 Kelvin to 2K by 20 steps with 0.1 increments
  Temperature[l] = 0.1*(l+1);
  U = Umin + (3/2)*S*Temperature[l];
  rsum=0;
  if(U < U_limit){
    //Root finding for ri & rf values
    for (j=0; j<2; j++){
      if(j==0){
        c=rmin;
        d=0.07;
      }
      else{
        c=0.071;
        d=0.078;
      }

      if((LJ_Potential(c)-U)*(LJ_Potential(d)-U) >= 0){
        printf("Choose a different interval");
        return 1;
      }

      while( c-d > eps){
        dchalf = (c+d)/2;
        if((LJ_Potential(dchalf)-U)*(LJ_Potential(c)-U) < 0){
```

```
        d = dchalf;
      }
      else if((LJ_Potential(dchalf)-U)==0){
        break;
      }
      else{
        c = dchalf;
      }
    }
    if(j==0){
      ri=(c+d)/2;
    }
    else{
      rf=(c+d)/2;
    }
  }
  printf("%lf %lf\n", ri, rf);


  //Time steps for r values
  h=0.001;
  N = 10/h;

  tn[0]=0;
  rn=ri;

  for(i=0; i<=N; i++){
      k1v = Velocity(tn[i], vn);
      k1r = Force_RK(tn[i], rn, vn);
      k2v = Velocity(tn[i]+(h/2), vn+(k1r*h/2));
      k2r = Force_RK(tn[i]+(h/2), rn+(k1v*h/2), vn+(k1r*h/2));
```

```c
     k3v = Velocity(tn[i]+(h/2), vn+(k2r*h/2));
     k3r = Force_RK(tn[i]+(h/2), rn+(k2v*h/2), vn+(k2r*h/2));
     k4v = Velocity(tn[i]+(h/2), vn+(k3r*h));
     k4r = Force_RK(tn[i]+(h/2), rn+(k3v*h), vn+(k3r*h));

     vn_plus1[i] = vn + (k1r+2*k2r+2*k3r+k4r)*h/6;
     rn_plus1[i] = rn + (k1v+2*k2v+2*k3v+k4v)*h/6;

     if(i==0){
       fprintf(fp2, "%lf %lf %lf %lf\n", tn[i], rn, vn, LJ_Potential(rn));
     }
     else{
       fprintf(fp2, "%lf %lf %lf %lf\n", tn[i], rn_plus1[i], vn_plus1[i], LJ_Potential(rn));
     }

     tn[i+1] = tn[i]+h/1000;  // t=0 sec to 0.1 sec in 100 steps for chosen h=0.1
     vn = vn_plus1[i];
     rn = rn_plus1[i];
     rsum = rsum + rn;

     if(i<13){ // After 12th iteration, molecule goes backward
       if(i==0){
        fprintf(fp3, "%lf %lf %lf %lf\n", tn[i], rn, vn, LJ_Potential(rn));
       }
       else{
        fprintf(fp3, "%lf %lf %lf %lf\n", tn[i], rn_plus1[i], vn_plus1[i], LJ_Potential(rn));
       }
     }
  }
```

```
    rave[l] = rsum /(N+1);
    fprintf(fp1, "%lf  %lf\n", Temperature[l], rave[l]);
  }
  else{
    break;
  }
}

//average expansion coefficient calculation
for(i=0; i<20; i++){
  rave_sum = rave_sum + rave[i];
}
dT = Temperature[19]-0; //change in temperature which is 0 to 2.0 K
drave = (rave_sum/20)-r0; //average change in average distances
ec = (1/0.074)*(drave/dT);
printf("expansion coefficient:%lf\n", ec);

fclose(fp1);
fclose(fp2);
fclose(fp3);
return 0;
}

double LJ_Potential(double r){
  return ((0.1*pow(10, -12)/pow(r,12))-(1.5*pow(10, -6)/pow(r,6)));
}

double Force(double r){
  return ((12*0.1*pow(10, -12)/pow(r,13))-(6*1.5*pow(10, -6)/pow(r,7)));
}

```

```
double Force_RK(double t, double r, double v){
 return ((12*0.1*pow(10, -12)/pow(r,13))-(6*1.5*pow(10, -6)/pow(r,7)));
}

double Velocity(double t, double v){
 return v;
}

double D_Force(double r){
 return ((-156*0.1*pow(10, -12)/pow(r,14))+(42*1.5*pow(10, -6)/pow(r,8)));
}
```

7.REFERENCES

[1] Serway, R. A., and Jewett, J. W. (2019). Physics for Scientists and Engineers with Modern Physics (10th edition). Cengage. 1145, 1172

[2] The Lennard-Jones Potential. Dissemination of IT for the Promotion of Materials Science (DoITPoMS). https://www.doitpoms.ac.uk/tlplib/stiffness-of-rubber/lennard-jones.php

[3] Physical Constants. https://www.physics.rutgers.edu/grad/541/constants.html

[4] Program for Bisection Method. GeeksforGeeks. https://www.geeksforgeeks.org/program-for-bisection-method/