Exploring Potential Energy Surfaces of nanoclusters

M.Sc. Thesis

By

Shubham Rawat

(1703131018)

DISCIPLINE OF CHEMISTRY

INDIAN INSTITUTE OF TECHNOLOGY INDORE

June, 2019

Exploring Potential Energy Surfaces of nanoclusters

A THESIS

Submitted in partial fulfillment of the
requirements for the award of the degree
of
Master of Science

by
Shubham Rawat

DISCIPLINE OF CHEMISTRY
INDIAN INSTITUTE OF TECHNOLOGY INDORE
June, 2019

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Exploring Potential Energy Surfaces of nanoclusters** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF SCIENCE** and submitted in the **DISCIPLINE OF CHEMISTRY, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from July,2018 to June,2019 under the supervision of Dr. Satya S. Bulusu, Associate Professor, Indian Institute of Technology Indore.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

(Shubham Rawat)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(Dr. Satya S. Bulusu)

Shubham Rawat has successfully given his M.Sc. Oral Examination held on **_/_/2019.**

Signature of Supervisor of MSc thesis
Date:

Convener, DPGC
Date:

Signature of PSPC Member #1
Date:

Signature of PSPC Member #2
Date:

ACKNOWLEDGEMENTS

I would like to thank my supervisor, ***Dr. Satya S. Bulusu***, for the guidance, encouragement, and advice he has provided throughout my time as his student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly. I would also like to thank my PSPC members, ***Dr. Anjan Chakraborty,*** and ***Dr. Chelvam Venkatesh***.

I want to acknowledge the assistance given to me by my lab senior Ms. Shweta Jindal. She really helped me a lot by giving me guidance and encouragement whenever I was stuck in some problem. Her contribution was invaluable to me in this project.

I would like to thank my MSc batchmates for the friendly environment especially Aniket. Finally, I would like to thank my parents and sister who have always been a constant source of support for me throughout my life.

Shubham Rawat

Dedicated to my family

Abstract

Molecular Dynamics is an important technique for exploring Potential Energy Surfaces at finite temperatures. We have software packages for running Molecular Dynamics Simulations but these are generalized programs and hence more computationally complex. In this project, we developed a standalone Molecular Dynamics program using the guidance from the software called Tinker. This program runs Molecular Dynamics Simulations using our in-house potentials and specifically targets nanoclusters. We tested this program using the results already reported for Au147 and then ran Molecular Dynamics simulations for Au309 using two different initial structures.

TABLE OF CONTENTS

LIST OF FIGURES

SYMBOLS/UNITS

K - Kelvin

ps - picoseconds

fs - femtoseconds

Å - Angstroms

Au - Gold

ACRONYMS

MD – Molecular Dynamics

PES – Potential Energy Surfaces

BO – Born-Oppenheimer

RMSD – Root Mean Square Distance

NN – Neural Network

Chapter 1

Introduction

1.1 General Introduction:

The Concept of PES is a key concept in Quantum Chemistry. These are energy landscapes which relate the energy of a polyatomic system to its geometry. The concept of PES originates from the BO treatment of polyatomic systems. The key idea behind BO approximation is that the nuclei are so much heavier than the electrons that they move very very slowly compared to electrons so we can assume that electrons are moving in a field of fixed nuclei[1,2]. If the nuclei are fixed, then their kinetic energy is zero and the total Hamiltonian reduces to Electronic Hamiltonian.

The total Hamiltonian is as follows:-

$$\hat{H} = (K.E)_e + (K.E)_N + V_{eN} + V_{ee} + V_{NN}$$

Since, $(K.E)_N = 0$, therefore, we obtain the following:-

$$\hat{H} = (K.E)_e + V_{eN} + V_{ee} + V_{NN}$$

Since nuclei are fixed, V_{NN} becomes constant and therefore it can be discarded from the above equation. The equation then modifies to

$$\hat{H} = (K.E)_e + V_{eN} + V_{ee}$$

If this Electronic Hamiltonian is solved then it will give the total energy of the polyatomic system with a fixed geometry (fixed positions of the nuclei). On plotting the energy (obtained from the Electronic Hamiltonian**)** against nuclear coordinates, we get the PES.

The significance of PES lies in the fact that the stationary points on PES represent different species. A minima corresponds to a stable chemical species (reactants, products) while a saddle point represents a transition state. Thus, PES finds use in the study of molecular geometry and chemical reaction dynamics.

In MD, we simulate the atoms using Newtonian dynamics. The key idea is that using an appropriate potential energy function we calculate the forces acting on atoms and then move them in accordance with Newton's Laws of motion with time.

When we perform MD we are actually simulating the real dynamics of the system under study and in doing so we can keep track of each individual atom in an incredibly detailed way. In this way, MD simulations can help us to gain new insight into important processes taking place at the atomic and molecular level.

1.2 Organization of the Thesis:

Chapter 2: Theory

Chapter 3: Fortran Code

Chapter 4: Results and Discussion

Chapter 5: Conclusions

Chapter 2

Theory

2.1: Methodology for Molecular Dynamics

Step 1: Assigning positions and velocities to atoms

We have the initial positions for all atoms. For initial velocities, random velocities are assigned to each of the atoms.

Step 2: Calculating the force acting on each atom

Trained NN1[3] and NN2[4] are used to calculate the potential for an atom.

$$\mathbf{E_i = \Sigma w^{23} \times f_2(b_2 + \Sigma w^{12} \times f_1(b_1 + \Sigma w^1 D_{ij}))}$$

where E_i is the energy of atom and D_{ij} stands for input descriptor functions. w^{23} and w^{12} are weights connecting the different layers of NN. b_1 and b_2 are bias weights in different layers of NN. f_2 and f_1 represent the functions for activation of the network[5].

The total energy (E) of a nanocluster is computed by summation of all the atomic energies.

$$\mathbf{E = \Sigma\ E_i}$$

The Force(F) on an atom is given by the negative gradient of total energy of the nanocluster.

$$\mathbf{F = -\ \delta E/\delta r = -\ \Sigma(\delta E_i/\delta D_{ij}) \times (\delta D_{ij}/\delta r)}$$

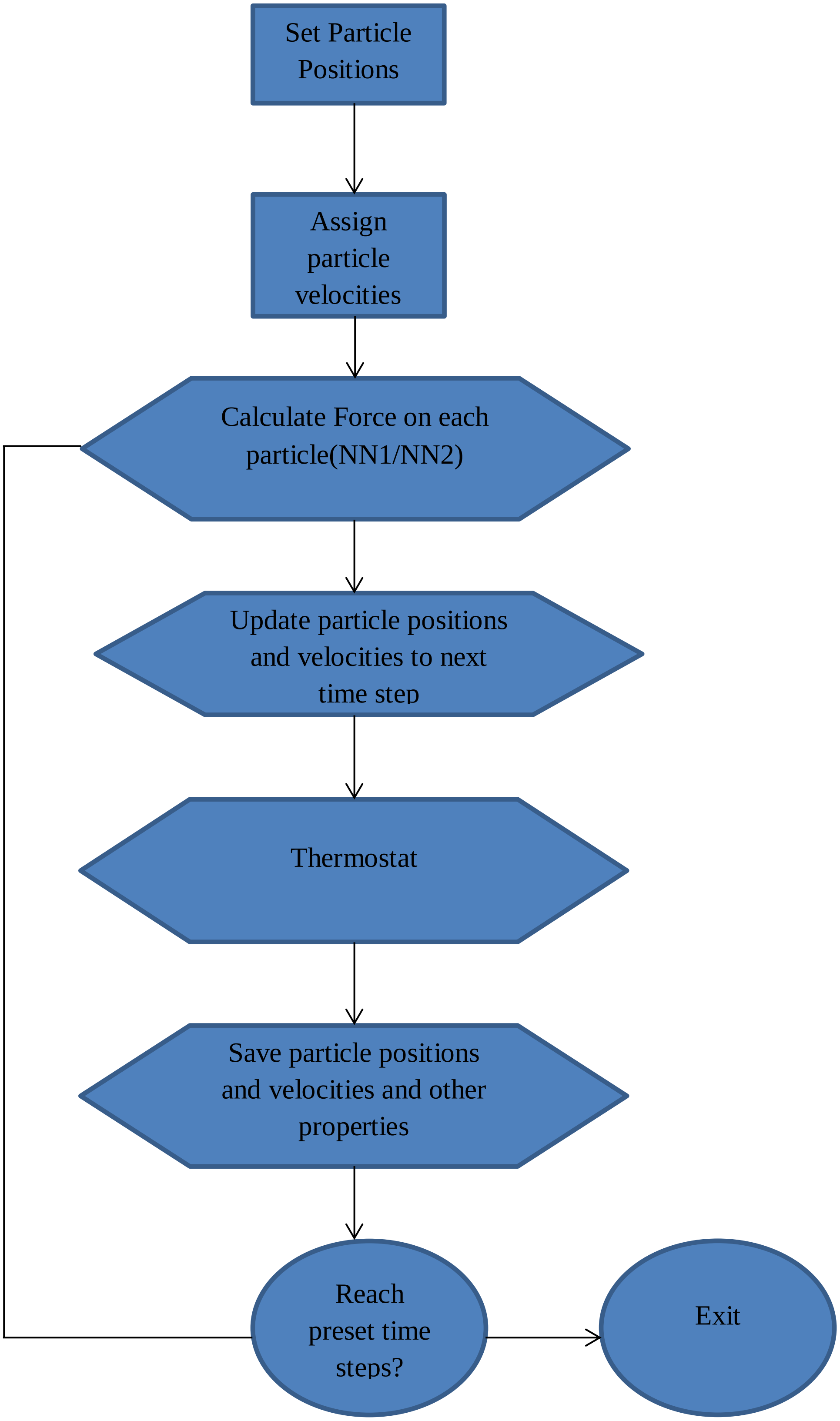


Figure 1: Flowchart for MD

Step 3: Updating positions and velocities of atoms

Using Newton's second law of motion, acceleration is calculated.

According to Newton's Second law of motion:-

$$\mathbf{F = m \times a}$$

If we know the value of force we can get acceleration:-

$$\mathbf{a = F/m}$$

Once acceleration is known, the velocity-verlet algorithm[6] is used to update positions for each atom and the velocities at mid-step

$$\mathbf{r(t+\delta t) = r(t) + \delta t \times v(t) + \tfrac{1}{2}\, \delta t \times \delta t \times a(t)}$$

$$\mathbf{v(t + \tfrac{1}{2}\, \delta t) = v(t) + \tfrac{1}{2}\, \delta t \times a(t)}$$

The forces and accelerations at time t+δt are then computed, and the velocity move is completed using the velocity-verlet algorithm**.**

$$\mathbf{v(t + \delta t) = v(t + \tfrac{1}{2}\, \delta t) + \tfrac{1}{2}\, \delta t \times a(t + \delta t)}$$

Step 4: Applying Thermostat

The updated velocities are scaled using the Andersen Thermostat algorithm to maintain constant Temperature condition.

Step 5: Saving current data in a file

Current velocities and positions of each atom are saved in a file. The current velocities and positions become previous velocities and positions.

Step 6: Check if the number of preset steps reached

If the number of preset steps is reached then the MD is stopped otherwise steps 3 to 5 are repeated until the required number of steps are reached.

2.2: Order Parameters

In order to study the movement of atoms in the structures obtained from Molecular Dynamics, some parameters can be calculated. Two such parameters are RMSD[4] and Volume Variation[4].

2.2.1: RMSD

$$\mathbf{RMSD_{confirmation}} = \sqrt{\frac{\sum_{\mathbf{i=1}}^{\mathbf{atoms}}(\mathbf{x_i} - \mathbf{x_{initial}})^2 + (\mathbf{y_i} - \mathbf{y_{initial}})^2 + (\mathbf{z_i} - \mathbf{z_{initial}})^2}{\mathbf{N}}}$$

where $x_{initial}$, $y_{initial}$ and $z_{initial}$ are the coordinates of the initial structure and x_i , y_i and z_i are the coordinates of the i^{th} atom of the confirmation obtained from the MD trajectory. N is the total number of atoms in the confirmation.

The significance of the RMSD is that the change in the atomic positions can be observed by comparing the current atomic positions with the atomic positions in the initial structure. Therefore, we can observe how the structure is evolving as the simulation progresses.

2.2.2: Volume Variation

We can find the variation in the volume of the structure by calculating the volume of each structure obtained from the MD. Small volume variations indicate that no major movements took place during the simulation while large volume variations mean that there is a lot of movement by the atoms during the simulation.

Chapter 3

Fortran Code

3.1 Main Code

```
      program dynamic
      implicit none
      include 'md.i'
      include 'atmtyp.i'
      include 'atoms.i'
      integer istep,nstep
      integer mode
      real*8 dt,dtdump
c
c
c     information levels within the program
c
      verbose = .false.
      debug = .false.
      abort = .false.
c
c     default input/output unit numbers
c
      input = 5
      iout = 6
c
c     default parameters used by optimizations
      iprint = -1
      iwrite = -1
c
c     type of coordinates file
c
      coordtype = 'NONE'
c
c     Reading key and xyz file
c
      call getxyz
c
c     initialize the temperature, pressure and coupling baths
c
      kelvin = 0.0d0
      isothermal = .false.
c
c     Opening .dat file
      open(input,file='Au309_ICO.dat',status='unknown')
```

```
c
c    initialize the simulation length as number of time steps
c
       read (input,340)
 340     format()
      read (input,220)  nstep
 220     format (i10)
c
c    get the length of the dynamics time step in picoseconds
c
      read (input,240,err=50)  dt
 240     format (f20.0)
  50    continue
     dt = 0.001d0 * dt
c
c    set the time between trajectory snapshot coordinate dumps
c
      read (input,260,err=70)  dtdump
 260     format (f20.0)
       if (dtdump .le. 0.0d0)  dtdump = 0.1d0
 70    continue
     iwrite = nint(dtdump/dt)
c
c    use constant energy or temperature for nonperiodic system
c
        read (input,280)  mode
 280       format (i10)
       if (mode .eq. 2 .or.mode.eq.1) then
         isothermal = .true.
           read (input,300,err=110)  kelvin
 300          format (f20.0)
           if (kelvin .le. 0.0d0)  kelvin = 298.0d0
 110          continue
       end if
c
c    initialize setup dynamics
c
     call mdinit
c
c    integrate equations of motion to take a time step
c
      do istep = 1, nstep
         call verlet (istep,dt)
     end do
c
c    print a final status message before exiting
```

```
c
      if (debug) then
         write (iout,320)
  320    format (/,' Exiting following Normal Termination',
     &              ' of the Program',/)
      end if
c
c     may need a pause to avoid closing the execution window
c
      if (holdup) then
         read (input,330)
  330    format ()
      end if
c
c     Closing .dat file
      close(input)
c
c
      end
```

3.2 Initializing Molecular Dynamics

Code for initializing Molecular Dynamics

```
     subroutine mdinit
      implicit none
      include 'md.i'
      include 'atmtyp.i'
      include 'atoms.i'
      integer i,j
      real*8 e,maxwell,speed
      real*8 vec(3)
      real*8 energy
      real*8 derivs(3,maxatm)
      logical exist
      real*8 rho,beta
      real*8 random,erfinv
      real*8 xspeed,yspeed
      real*8 zspeed
      external erfinv
      integer idyn,freeunit
      character*120 dynfile
```

```
c
c
c     set default parameters for the dynamics trajectory
c
      nfree=0
      integrate = 'BEEMAN'
      velsave = .false.
      frcsave = .false.
      iprint = 100
c
c     set default values for temperature and pressure control
c
      thermostat = 'BUSSI'
      collide = 0.1d0
c
    Assigning Parameters for MD
      integrate='VERLET'
      thermostat='Andersen'
      velsave=.true.
      frcsave=.true.
c
c     Setting degrees of freedom
      nfree=3*n
      if (nfree .eq. 0) then
         write (iout,50)
   50    format (/,' MDINIT  --  No Degrees of Freedom for Dynamics')
         call fatal
      end if
c
c     try to restart using prior velocities and accelerations
c
      dynfile = filename(1:leng)//'.dyn'
      inquire (file=dynfile,exist=exist)
      if (exist) then
         idyn = freeunit ()
         open (unit=idyn,file=dynfile,status='old')
         rewind (unit=idyn)
         call readdyn (idyn)
         close (unit=idyn)
      else
```

```
c
c     calculation of energy and gradient by using NN potential
      open (unit=30, file='tmp.xyz', status='unknown')
      write(30,*) n
      do i = 1, n
         write (30, '(3f15.8)') x(i), y(i), z(i)
      end do
      close(30)
      call system ('/home/satya/Shubham/Au309_MD_ICO/md309ICO_NN.exe &')
!      call system ('/usr/mpi/gcc/mvapich2-2.0/bin/mpirun
!     &-np 8 /scratch/sblusu/SHWETA/MD_T300Au58_constructed/NN.exe &')
      call wait_NNPES()
!      call sf_feedforward_neuralnetwork
      open (unit=30, file='force.txt', status='unknown')
      read (30,*) energy
      do i = 1, n
         read(30,*) derivs(1,i),derivs(2,i),derivs(3,i)
      end do
      close(30)
      call system("rm force.txt tmp.xyz")
c
c     Initializing velocities
c
         do i = 1, n
c
c     set normalization factor for cumulative velocity distribution
c
      beta = sqrt(mass(i) / (2.0d0*boltzmann*kelvin))
c
c     pick a randomly distributed velocity along each of three axes
c
      rho = random ()
      xspeed = erfinv(rho) / beta
      rho = random ()
      yspeed = erfinv(rho) / beta
      rho = random ()
      zspeed = erfinv(rho) / beta
c
c     set the final value of the particle speed in 3-dimensions
c
```

```
      maxwell = sqrt(xspeed**2 + yspeed**2 + zspeed**2)
            speed = maxwell
            call ranvec (vec)
            do j = 1, 3
              v(j,i) = speed * vec(j)
              a(j,i) = -convert * derivs(j,i) / mass(i)
              aold(j,i) = a(j,i)
            end do
        end do
        call mdrest
      end if
c
      return
      end
c
c
!     editing according to our potential requirement ends
      subroutine wait_NNPES()
      implicit none
      integer numline
      logical test
      character filen*80
      filen ='force.txt'
1      continue
      inquire(FILE=filen, exist=test)
      if (test) then
        call system('wc -l force.txt >ntmpline')
        open(unit=30,file='ntmpline',status='unknown')
        read(30,*) numline
        close(30)
        call system('rm ntmpline')
        if (numline.GT.6 ) then
          goto 3
        else
          goto 1
        endif
      else
        goto 1
      endif
3      continue
```

```
      return
      end
```

3.3 Velocity- Verlet algorithm

Code for Velocity-Verlet Method

```
       subroutine verlet (istep,dt)
      implicit none
      include 'md.i'
      include 'atmtyp.i'
      include 'atoms.i'
      integer i,j,istep
      real*8 dt,etot
      real*8 dt_2,dt2_2
      real*8 eksum,epot
      real*8 temp,pres
      real*8 ekin(3,3)
      real*8 xold(maxatm)
      real*8 yold(maxatm)
      real*8 zold(maxatm)
      real*8 energy
      real*8 derivs(3,maxatm)
c
c
c     set some time values for the dynamics integration
c
      dt_2 = 0.5d0 * dt
      dt2_2 = dt * dt_2
c
c     store the current atom positions, then find new atom
c     positions and half-step velocities via Verlet recursion
c
      do i = 1, n
           xold(i) = x(i)
           yold(i) = y(i)
           zold(i) = z(i)
           x(i) = x(i) + v(1,i)*dt + a(1,i)*dt2_2
           y(i) = y(i) + v(2,i)*dt + a(2,i)*dt2_2
           z(i) = z(i) + v(3,i)*dt + a(3,i)*dt2_2
```

```
            do j = 1, 3
               v(j,i) = v(j,i) + a(j,i)*dt_2
            end do
      end do
c
c     calculation of energy and gradient by using NN potential
      open (unit=30, file='tmp.xyz', status='unknown')
      write(30,*) n
      do i = 1, n
         write (30, '(3f15.8)') x(i), y(i), z(i)
      end do
      close(30)
      call system ('/home/satya/Shubham/Au309_MD_ICO/md309ICO_NN.exe &')
!      call system ('/usr/mpi/gcc/mvapich2-2.0/bin/mpirun
!     &-np 8 /scratch/sblusu/SHWETA/MD_T300Au58_constructed/NN.exe &')
      call wait_NNPES()
!      call sf_feedforward_neuralnetwork
      open (unit=30, file='force.txt', status='unknown')
      read (30, *) energy
      epot=energy
      do i = 1, n
         read(30,*) derivs(1,i),derivs(2,i),derivs(3,i)
      end do
      close(30)
      call system("rm force.txt tmp.xyz")
c
c     use Newton's second law to get the next accelerations;
c     find the full-step velocities using the Verlet recursion
c
      do i = 1, n
            do j = 1, 3
               a(j,i) = -convert * derivs(j,i) / mass(i)
               v(j,i) = v(j,i) + a(j,i)*dt_2
            end do
      end do
c
c     accumulate the kinetic energy and its outer product
c
      call kinetic (eksum,ekin)
c
```

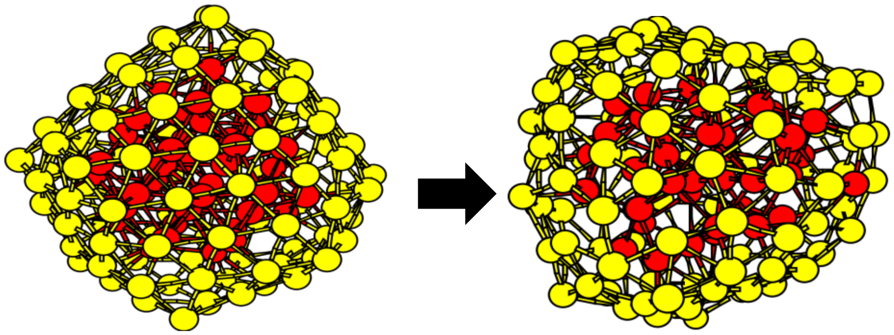
```
c     make full-step temperature and pressure corrections
c
      call temper2 (dt,eksum,temp)
c
c     system energy is sum of kinetic and potential energies
c
      etot = eksum + epot
c
c     compute statistics and save trajectory for this step
c
      call mdstat (istep,dt,etot,epot,eksum,temp,pres)
      call mdsave (istep,dt,epot)
      return
      end
```

Chapter 4

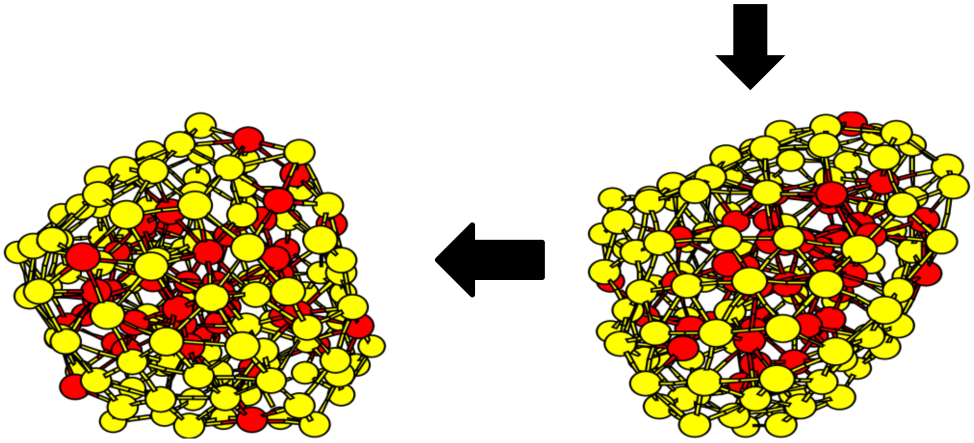
Results and Discussions

4.1 Testing on Au147

We tested our program by running MD simulations for Au147 at 700K with a time step of 3 fs. The Program reproduced the MD trajectory and RMSD data which are already reported in the literature [4].



Initial structure **At 24 ps**



At 480 ps **At 210 ps**

Figure 2: MD Trajectory for Au147 at 700K

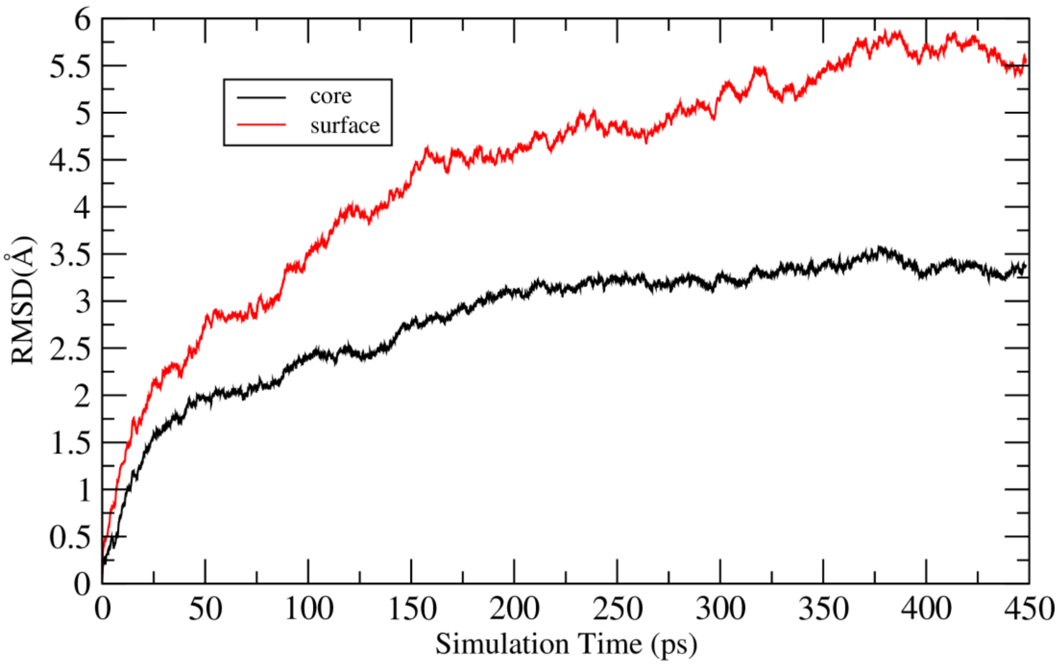
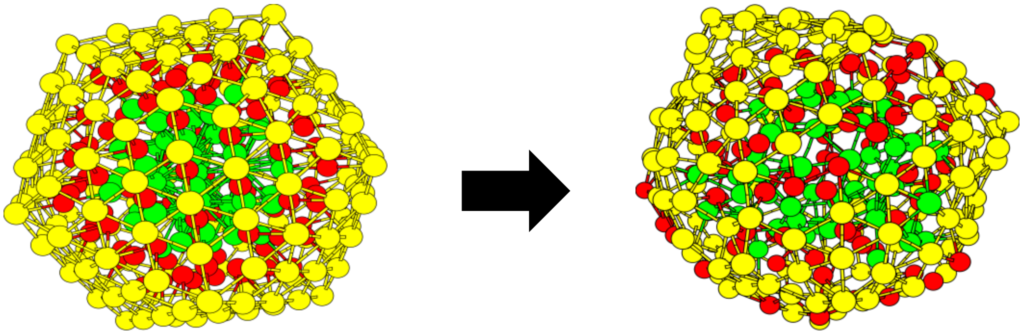


Figure3: RMSD for Au147 at 700K

4.2 Applying on Au 309

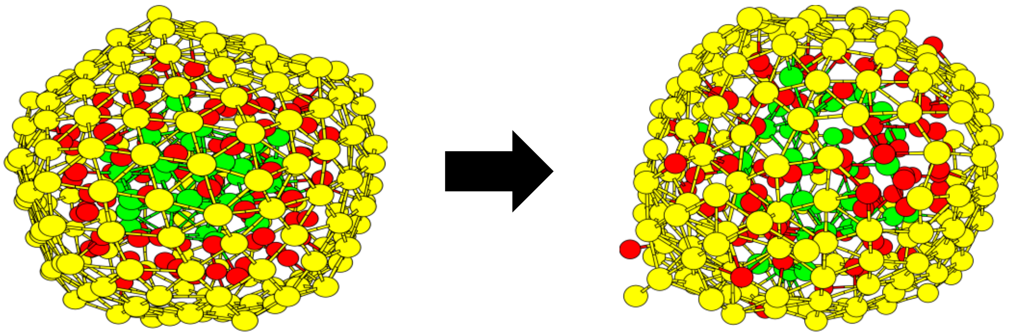
4.2.1: Molecular dynamics using NN1 potential

We ran MD simulations for 2 different initial Au309 clusters (icosahedral and amorphous) using our program. The simulations were run at a temperature of 500K with a time step of 3 fs. It was observed that the number of atoms on the surface increased for both the cases as the simulation progressed. In both cases, there was a change from a four-layered structure to a three-layered structure. Moreover, in the case of the icosahedral structure, all symmetry was lost and we obtained an amorphous structure.



Initial structure **At 42 ps**

Figure4: MD Trajectory for Au309 (Icosahedral) using NN1 potential at 500K



Initial Structure **At 42 ps**

Figure5: MD Trajectory for Au309 (Amorphous) using NN1 potential at 500K

In order to study the movement in the structures, two parameters were calculated. These are RMSD and Volume Variation.

RMSD

We calculated the RMSD of the clusters corresponding to the two initial structures we took.

From the RMSD plots for the two cases, we can say that core, surface and middle layer atoms are continuously moving during the simulation. From the plot, we can see that in the case of icosahedral, there is more movement of the middle layer atoms and core atoms. Indeed in the case of icosahedral, the increase in the no. of surface atoms is more as indicated from the final structures (at 42 ps) obtained from the MD. It is probably due to more no .of atoms from the middle layer and core moving outwards.

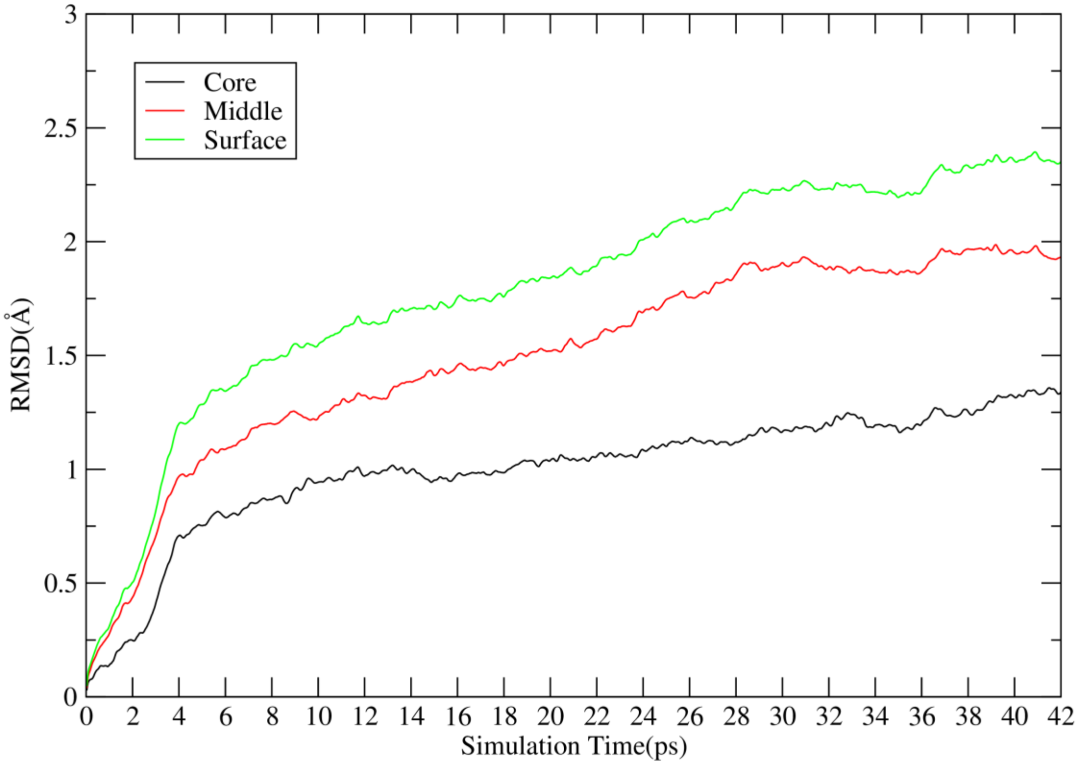


Figure 6: RMSD for Au309 (Icosahedral) with NN1 potential at 500K

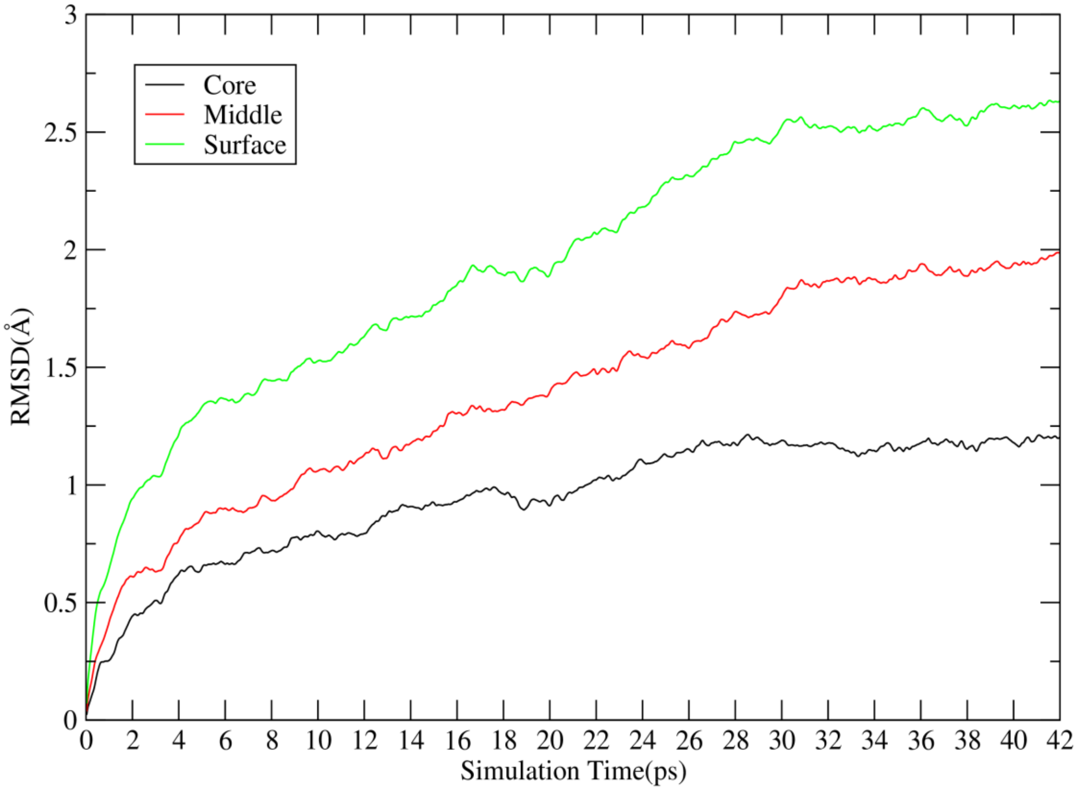


Figure 7: RMSD for Au309 (Amorphous) with NN1 potential at 500K

Volume Variation

We observed the change in the volume of the structure as the simulation progressed. In both the cases initially there was a large change in the volume but not so much change later on. One thing to be noted is that in the case of icosahedral, there is a drastic change in the volume compared to the case of amorphous which suggests more disturbance in the initial structure for the icosahedral case.

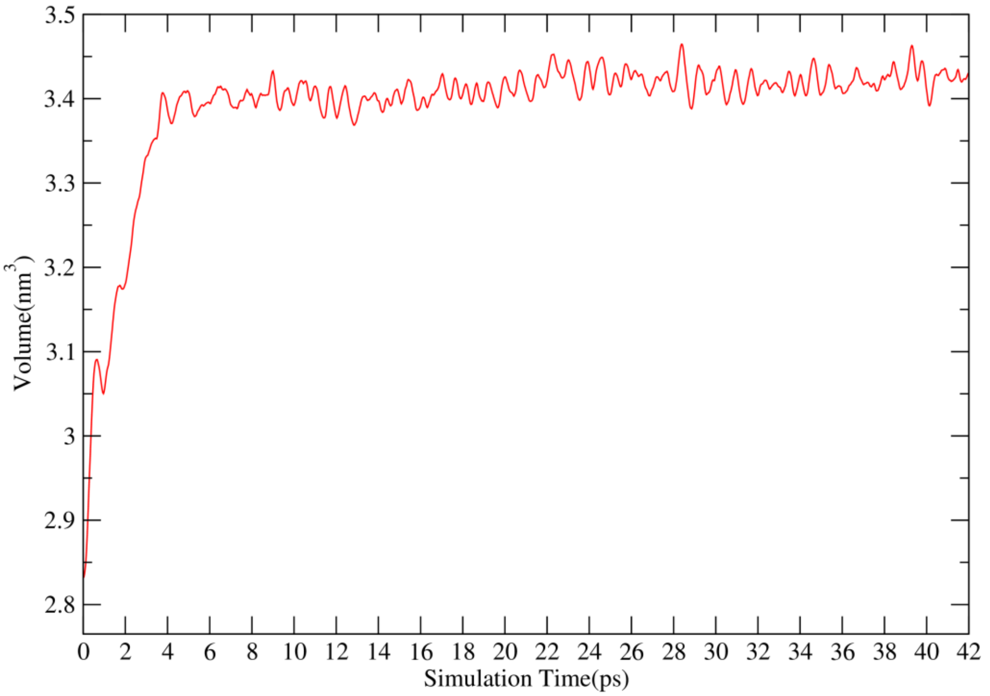


Figure 8: Volume Variation for Au 309 (Icosahedral) with NN1 potential at 500K

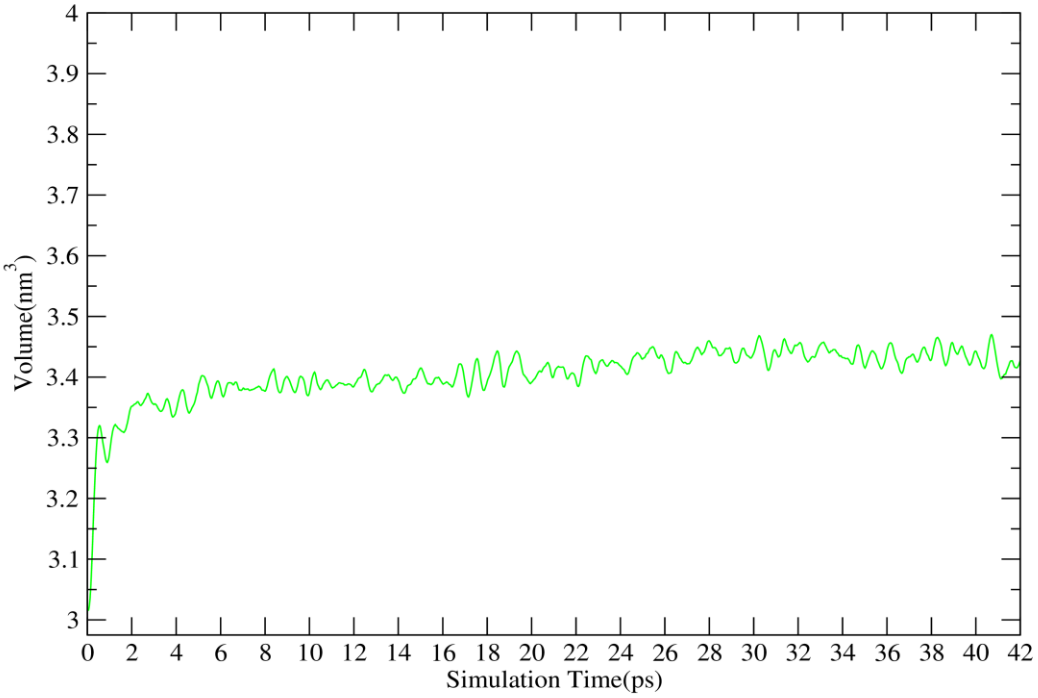
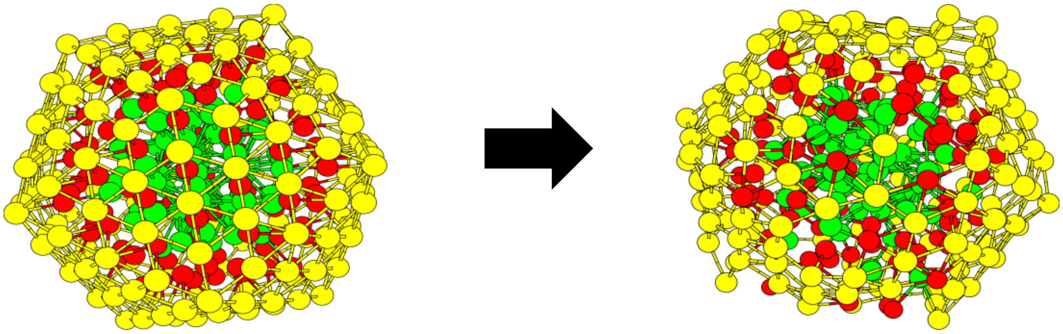


Figure 9: Volume Variation for Au 309 (Amorphous) with NN1 potential at 500K

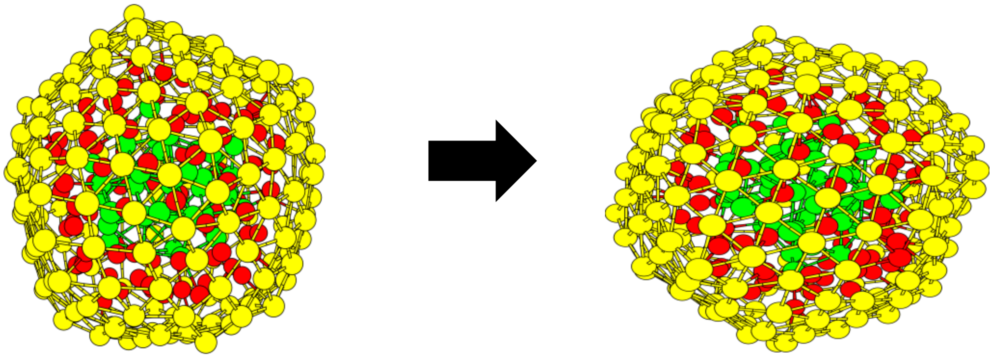
4.2.2: Molecular dynamics using NN2 potential

We again ran MD simulations with the same initial Au309 clusters (icosahedral and amorphous) using our own program. The simulations were run at a temperature of 500K and at 300K respectively with a time step of 3 fs. In the case of Icosahedral initial structure, it was observed that the number of atoms on the surface increased as the simulation progressed. Moreover, all symmetry was lost and we obtained an amorphous structure. In the case of Amorphous initial structure, the atoms in the 2 innermost layers increased as the simulation progressed. Unlike the previous NN 1 case, the structures remained four- layered.



Initial Structure **At 42 ps**

Figure 10: MD Trajectory for Au309 (Icosahedral) using NN2 potential at 500K



I **Initial Structure** **At 42 ps**

Figure 11: MD Trajectory for Au309 (Amorphous) using NN2 potential at 300K

In order to study the movement in the structures, two parameters were calculated. These are RMSD and Volume Variation.

RMSD

We calculated the RMSD of the clusters corresponding to the two initial structures we took. From the RMSD plots for the two cases, we can say that core, surface and middle layer atoms are continuously moving during the simulation. From the plots, we can see that in the case of icosahedral, there is more movement of the middle layer atoms and core atoms but in the amorphous case, there is more movement of surface atoms.

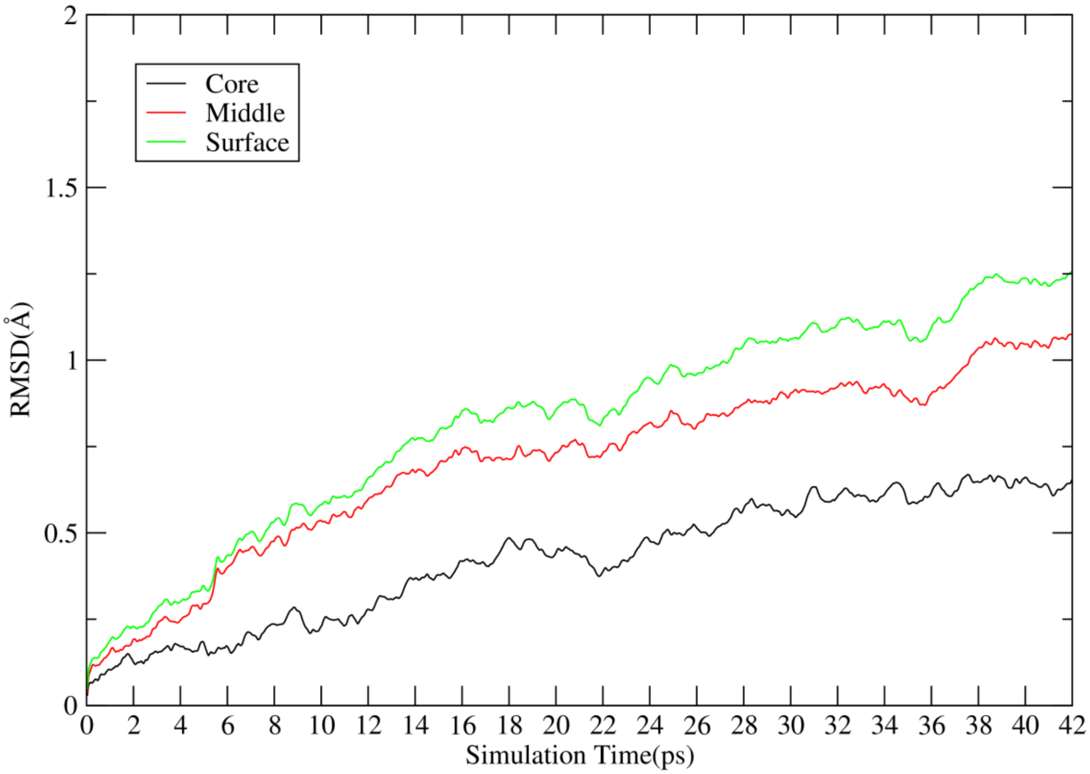


Figure 12: RMSD for Au309 (Icosahedral) with NN2 potential at 500K

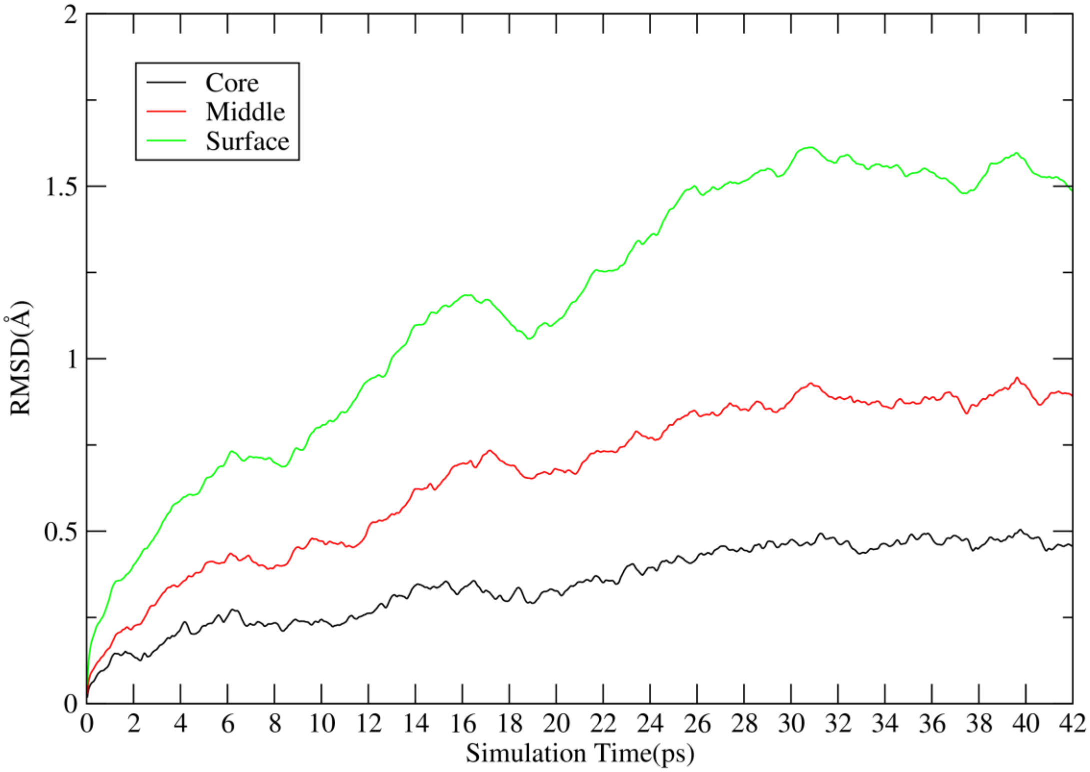


Figure 13: RMSD for Au309 (Amorphous) with NN2 potential at 300K

Volume Variation

We observed the change in the volume of the structure as the simulation progressed. In both the cases initially there was an appreciable increase in the volume but not so much change later on. One thing to be noted is that in the case of icosahedral, the increase in the volume is more compared to the case of amorphous which suggests more disturbance in the initial structure for the icosahedral case.

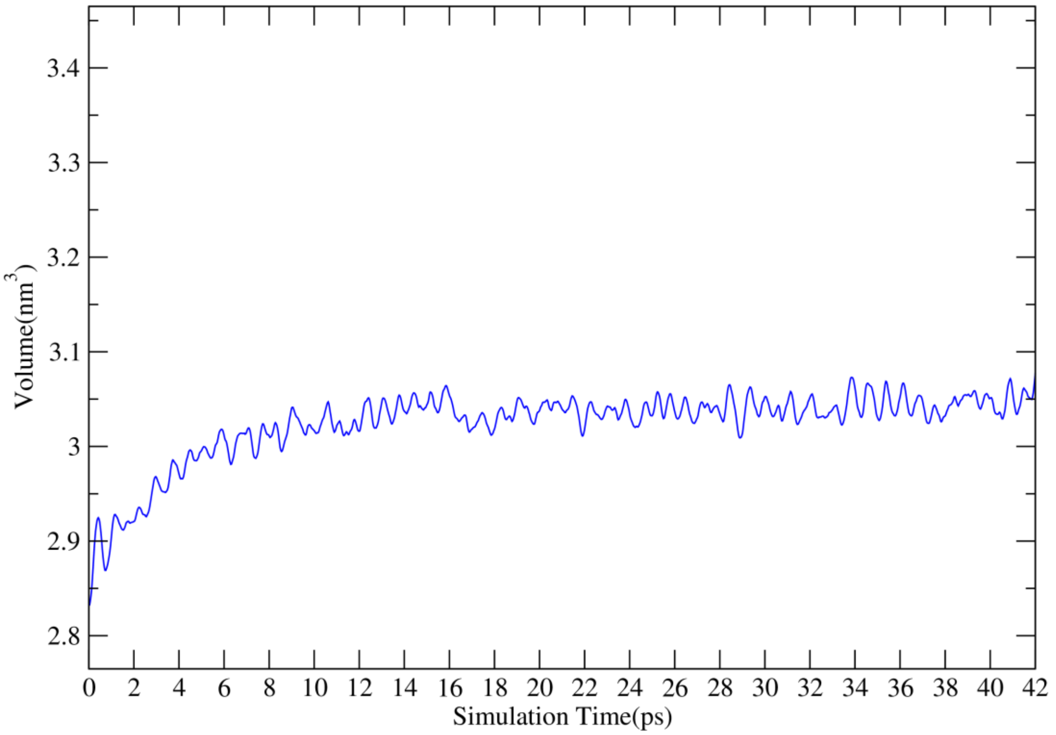


Figure 14: Volume Variation for Au 309 (Icosahedral) with NN2 potential at 500K

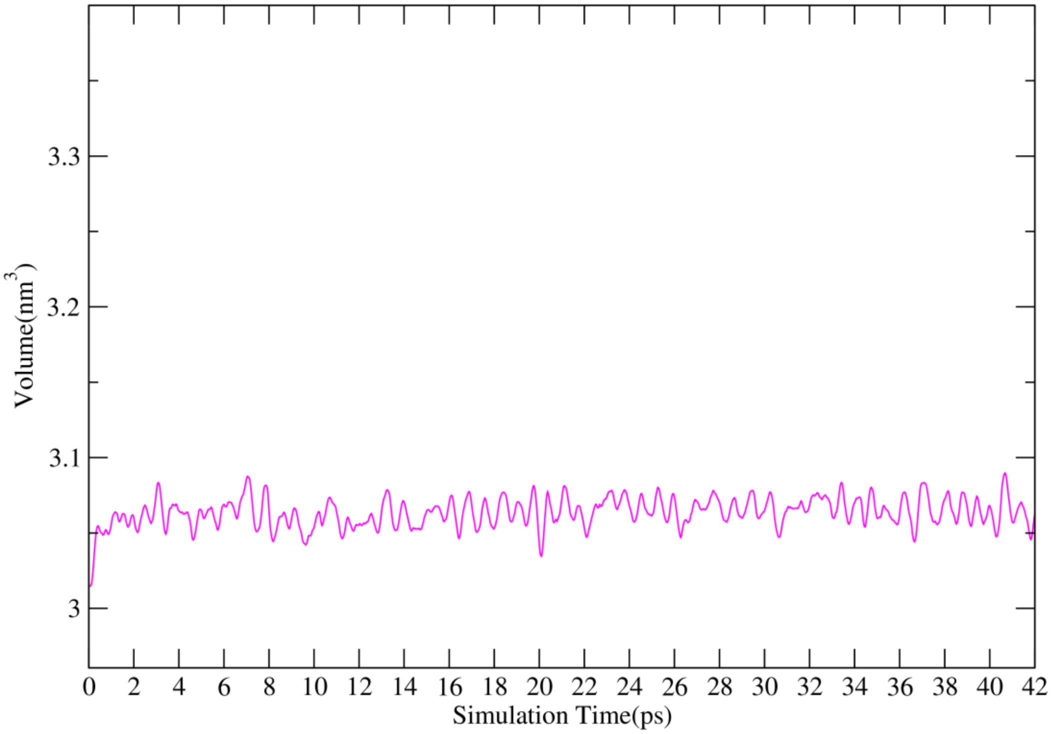


Figure 15: Volume Variation for Au 309 (Amorphous) with NN2 potential at 300K

Chapter 4

Conclusions

- The aim of the project was to write a standalone MD program which uses our in-house potentials for running MD simulations for nanoclusters to explore potential energy surfaces of these nanoclusters.
- The MD program was written and it was tested successfully on Au147 nanocluster. It reproduced the MD trajectory and RMSD data for Au147 nanocluster as reported in the literature.
- This program offers less computational complexity. Unlike generalized MD programs, it is specialized to work for nanoclusters.
- Using this program, we obtained the MD trajectories for Au309 nanocluster. The MD was initiated with two different initial structures ((a) Icosahedral and (b) Amorphous) using two different NN Potentials (NN1 and NN2). We analyzed the trajectories by calculating the RMSD and Volume Variation in the clusters with respect to the simulation time.
- The structures of Au309 nanocluster evolved from four-layered to three-layered when NN1 potential was used to calculate potential energy and forces during the dynamics. On the other hand, structures remained four-layered when NN2 potential was used.
- As part of future work, further modifications can be done in the program in order to incorporate other possibilities.

REFERENCES

[1] Mustroph H. (2016), Potential-Energy Surfaces, the Born–Oppenheimer Approximations, and the Franck–Condon Principle: Back to the Roots. ChemPhysChem, 17, 2616-2629.

[2] Sutcliffe B. T. and Woolley R. G. (2012), On the quantum theory of molecules. J. Chem. Phys., 137, 22A544.

[3] Jindal S. and Bulusu S. S. (2018), An algorithm to use higher order invariants for modelling potential energy surface of nanoclusters. Chem. Phys. Lett., 693, 152-158.

[4] Jindal S., Chiriki S., and Bulusu S. S. (2017), Spherical harmonics based descriptor for neural network potentials: Structure and dynamics of Au147 nanocluster. J. Chem. Phys., 146, 204301.

[5] Chiriki S., Jindal S., and Bulusu S. S. (2017), Neural network potentials for dynamics and thermodynamics of gold nanoparticles. J. Chem. Phys., 146, 084314.

[6] Allen M. P. and Tildesley D. J. (2017), Computer simulation of liquids, Oxford University Press Inc., New York, pp. 78-82.