Multicanonical Monte Carlo simulations combined with neural network potentials for structure evolution of core-shell gold nanoclusters

M.Sc. Thesis

By

Mukti Jadon

(1803131013)

DISCIPLINE OF CHEMISTRY

INDIAN INSTITUTE OF TECHNOLOGY INDORE

June 2020

Multicanonical Monte Carlo simulations combined with neural network potentials for structure evolution of core-shell gold nanoclusters

A THESIS

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

by
Mukti Jadon

DISCIPLINE OF CHEMISTRY
INDIAN INSTITUTE OF TECHNOLOGY INDORE
June 2020

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Multicanonical Monte Carlo simulations combined with neural network potentials for structure evolution of core-shell gold 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 2019 to June 2020 under the supervision of Dr. Satya S. Bulusu, Associate Professor, Department of Chemistry, 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.

Mukti Jadon

(Mukti Jadon)

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

(Dr. Satya S. Bulusu)

MUKTI JADON has successfully given his/her M.Sc. Oral Examination held on **___/___/2020.**

Tushar Kanti Mukherjee

Signature of Supervisor of MSc thesis
Date:

Convener, DPGC
Date:

Signature of PSPC Member
Date:

Signature of PSPC Member
Date:

Acknowledgments

I would like to thank my supervisor, Dr. Satya S. Bulusu, for his constant support, guidance, and encouragement. He has assisted me throughout my one-year masters project. 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, Prof. Suman Mukhopadhyay, and Dr. Manavendra Mahato.

I would like to acknowledge this assistance given to me by my lab senior Ms. Shweta Jindal and Mr. Abhishek Ojha. They helped me a lot by giving me guidance and encouragement whenever I was stuck in some problem. Their contribution was precious to me in this project.

I would like to thank all my classmates and friends. Finally, I would like to thank my parents who have always been a constant source of support for me throughout my life.

Mukti Jadon

Dedicated to My Family

and

Respected Teachers

Abstract

Evolution of gold nanoclusters have attracted great attention in the past decade due to their remarkable catalytic activities, size- and support-dependent CO oxidation, and structure-activity relationships. Here, we performed a global structure search of the low-energy cluster of Au_n (n = 59, 60, 68, 70, 92, 106, 112, 138, 156, 166, 168) exhibit core-shell type structures by using multicanonical Monte Carlo simulations combined with artificial neural network potentials. We found that the most stable structure of gold nanoclusters exhibit icosahedral type core structure.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS/UNITS

eV	Electron volt
Au	Gold
n	Number of gold atom
K	Kelvin

ACRONYMS

MCMC	Multicanonical Monte Carlo
MC	Monte Carlo
ANNPs	Artificial neural network potentials
MD	Molecular Dynamics
LM	Local Minimum
NN	Neural network

Chapter 1

Introduction

1.1 General Introduction:

Global optimization of a function is that finds the global minima or maxima of a complex system that has continuously attracted attention. For searching the global minimum, potential energy taking as a function while complex variables as the coordinates of atoms. Global minimum can be taken as a technique to find the ground state of a physical system.[1] MC techniques are valuable for simulating systems with numerous degrees of freedoms, for example, liquids, disorder materials, strongly coupled solids, and cellular structures. These methods are used when a large number of minima exist in a complex system.

Mathematically functions are estimated by MC simulation using random sampling and statistical modeling. These sampling experiments contain the generation of arbitrary numbers observed by using a restricted number of arithmetic and logical operations, which are often the same at each step. At the start of an MC move, a random atom is selected which gives a uniform random displacement along each of the coordinate instructions. The most displacement is an adjustable parameter that governs the dimensions of the region and controls the convergence of the Markov chain.[2]

The MC technique is time-consuming and meanwhile, most researchers are progressively keen on new outcomes rather than the methodology. There has been little work on the optimization of parameters, for example, maximum displacement and the choice of the transition matrix. Metropolis method one randomly selected atom is moved to produce another state. The underlying stochastic matrix can be changed so a few or the entirety of the particles are moved at the same time. Chapman and Quirk found that equilibration was accomplished more rapidly by utilizing multi-particle and single-particle moves, as estimated by their ability to sample phase space in a given measure of computer time, has not been exposed to systematic study. A typical practice in MC simulation is to pick out the particles to move sequentially instead of randomly. This cuts down the amount of arbitrary number generation and is a similarly valid method of producing the correctly weighted states. The length of an MC simulation is conveniently estimated in ‘cycles’, i.e. N trial moves whether selected sequentially or randomly. The computer time engaged with an MC cycle is similar to that in an MD time step. Of fundamental significance to all Monte Carlo techniques, the statistical weight relates the energy potential to the coordinate system.[2]

MCMC simulations are a Markov chain Monte Carlo approach for the simulation of bodily models, which yields canonical expectation values over various temperatures. MCMC simulation is the most normally applied algorithm wherein a Boltzmann weight has been followed in which temperature performs a pivotal role in determining the thermodynamic properties.

$$T = \beta^{-1}$$

$$w(a) = e^{-\beta E_a}$$

A perfect choice of the weight is the inverse of the density of states of energy, which might result in the favoured uniform random-walk trajectory over the whole energy space. However, the density of states is unknown a priori, and the MCMC method generates an approximation to it by enhancing the estimate iteratively. The process is repeated until no further significant development of the multicanonical weight is found, and the approximate position of the energy minimum is resolved. MCMC method has already shown achievement in their applications to finding stable systems of crystalline clusters and predicting protein native systems.[1]

Gold nanoparticles can show particular physical and chemical properties. A lot of these unique properties can be attributed to both strong relativistic properties and finite-magnitude quantum effects [3,4]. MD or MC simulations are used to achieve the finite-time and finite temperature properties of chemical systems. These simulations use classical interatomic potentials or ab initio techniques to calculate the energy and forces of the system. In the present study, we performed a LM structure search of low-energy clusters for size selected Au_n nanoparticles using MCMC simulations combined with ANNPs.

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 Multicanonical Monte Carlo simulations

To evaluation of gold nanoclusters by using MCMC simulations with ANNPs and starting with different random geometries of various sizes Au_n (n = 59, 60, 68, 70, 92, 106, 112, 138, 156, 166, 168). ANNPs was used for generating data using the MCMC simulation for the structural evolution of gold nanocluster.

Step 1: Assigning positions of atoms

We assigned random initial position for each of the atom of various sizes Au_n (n = 59, 60, 68, 70, 92, 106, 112, 138, 156, 166, 168).

Step 2: Calculate the potential of nanoparticles

Trained NN is used to calculate the potential for an atom. A two-layer feed-forward NN architecture are shown as follows:

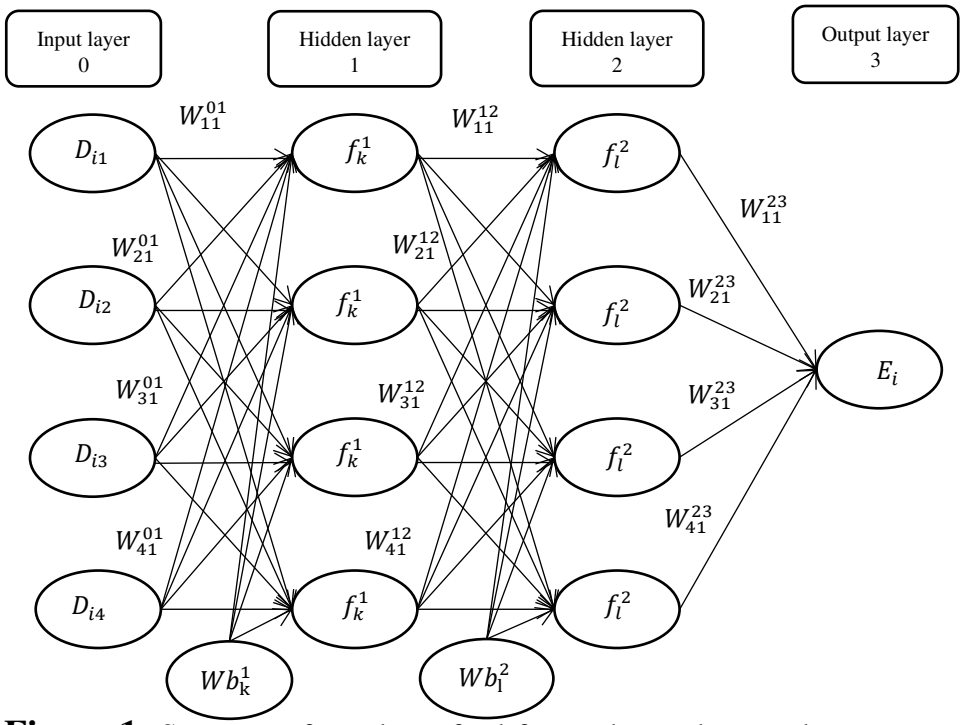


Figure 1: Structure of two-layer feed-forward neural networks

$$E_i = \sum_{l=1}^{N_{hn}} W_{l1}^{23} . f_l^2 \left[Wb_l^2 + \sum_{k=1}^{N_{hn}} W_{kl}^{12} . f_k^1 \left(Wb_k^1 + \sum_{j=1}^{N_D} W_{jk}^{01} . D_{ij} \right) \right]$$

Where E_i is the energy of i^{th}atom and N_{hd} is the number of hidden nodes. D_{ij} stands for input descriptor functions. W_{jk}^{01}, W_{kl}^{12}, and W_{l1}^{23} are weights connecting the different layers of NN. Wb_l^2 and Wb_k^1 are bias weights in different layers of NN. f_l^2 and f_k^1 represent the sigmoid function for activation of the network.[5,6]

The total energy (E) of a nanoparticle is computed by adding all individual atomic energies,

$$\mathrm{E} = \sum_i E_i$$

Step 3: Multicanonical Monte Carlo simulations

We study a physical system described by a ANNPs Energy. MC simulations with a-priori unknown weight factors are feasible and need to be considered. To sample the significant configurations of a canonical ensemble at temperature $T = \beta^{-1}$ with Boltzmann weights [7]

$$w(a) = e^{-\beta E_a}, \tag{1}$$

Where E_a is the energy of configuration a.

Let us first discuss the connection of the weight factors with microcanonical temperature $\beta(E)$,

$$w^{n+1}(a) = e^{-S(E_a)} = e^{-\beta(E_a)E_a + \alpha(E_a)} \tag{2}$$

Where $S(E)$ is the microcanonical entropy and, by definition,

$$\beta(E) = \frac{\partial S(E)}{\partial E}.$$

This determines the fugacity function $\alpha(E)$ as much as an (irrelevant) additive constant. We take into account the case of a discrete minimal energy ϵ and indicate

$$\beta(E) = [S(E+\epsilon) - S(E)]/\epsilon \tag{3}$$

And the identity $S(E) = \beta(E)E - \alpha(E)$ implies

$$S(E) - S(E-\epsilon) = \beta(E)E - \beta(E-\epsilon)(E-\varepsilon) - \alpha(E) + \alpha(E-\epsilon).$$

Inserting $\epsilon\beta(E-\epsilon) = S(E) - S(E-\epsilon)$ yields

$$\alpha(E-\epsilon) = \alpha(E) + [\beta(E-\epsilon) - \beta(E)]E \tag{4}$$

And $\alpha(E)$ is fixed through defining $\alpha(E_{max}) = 0$. In precis, once $\beta(E)$ is provide, $\alpha(E)$ follows for free.

A convenient beginning condition for the initial (n = 0) simulation is

$$\beta^0(E) = \alpha^0(E) = 0. \tag{5}$$

The energy histogram the n^{th} simulation is given through $H^n(E)$. To keep away from $H^n(E) = 0$ we replace for the instant

$H^n(E) \to \widehat{H}^n(E) = \max[h_0, H^n(E)]$,

Where h_0 is a variety of $0 < h_0 < 1$. Our final equations allow for the restriction $h_0 \to 0$. Subscripts zero are used to indicate that the ones quantities aren't yet the very last estimators from the n^{th} simulation. Let

$$w_0^{n+1}(E) = e^{-S_0^{n+1}(E)} = c\frac{w^n(E)}{\widehat{H}^n(E)},$$

Where the constant c is introduced to ensure that $S_0^{n+1}(E)$ maybe an estimator of the microcanonical entropy. It follows $S_0^{n+1}(E) = -lnc + S^n(E) + ln\widehat{H}^n(E)$. Putting this relation into (3) gives

$$\beta_0^{n+1}(E) = \beta^n(E) + \frac{[ln\widehat{H}^n(E+\epsilon) - ln\widehat{H}^n(E)]}{\epsilon} \tag{6}$$

As estimator of the variance follows

$$\sigma^2[\beta_0^{n+1}(E) = \frac{c'}{H^n(E+\epsilon)} + \frac{c'}{H^n(E)},$$

Wherein c' is an unknown constant and we have re-presented the original histograms to emphasize (and use) that the variance is endless while there is zero statistics. The statistical weight for $\beta_0^{n+1}(E)$ is inversely proportional to its variance. Selecting a convenient constant, we get

$$g_0^n(E) = \frac{H^n(E+\epsilon)H^n(E)}{H^n(E+\epsilon)+H^n(E)}. \tag{7}$$

Be aware that $g_0^n = 0$ for $H^n(E+\epsilon) = 0$ or $H^n(E) = 0$. The n^{th} simulation changed into done using $\beta^n(E)$. It's miles now sincere to mix $\beta_0^{n+1}(E)$ and $\beta^n(E)$ in keeping with their respective statistical weights into the preferred estimator

$$\beta_0^{n+1}(E) = \hat{g}^n(E)\beta^n(E) + \hat{g}_0^n(E)\beta_0^{n+1}(E), \tag{8}$$

Where the normalized weights

$$\hat{g}_0^n(E) = \frac{g_0^n(E)}{g^n(E)+\hat{g}_0^n(E)} \text{ and } \hat{g}^n(E) = 1 - \hat{g}_0^n(E)$$

Are determined by the recursion

$$g^{n+1}(E) = g^n(E) + g_0^n(E), g^0(E) = 0. \tag{9}$$

We can eliminate $\beta_0^{n+1}(E)$ from Eqn. (8) by inserting its definition (6) and get

$$\beta^{n+1}(E) = \beta^n(E) + \hat{g}_0^n(E) \times \left[ln\widehat{H}^n(E+\epsilon) - ln\widehat{H}^n(E)\right]/\epsilon \quad (10)$$

Finally, Eqn. (10) can be converted into a recursion for ratios of weight factor neighbors. We define

$$R^n(E) = e^{\epsilon\beta^n(E)} = \frac{w^n(E)}{w^n(E+\epsilon)} \qquad (11)$$

And get the recursion

$$R^{n+1}(E) = R^n(E)\left[\frac{\widehat{H}^n(E+\epsilon)}{\widehat{H}^n(E)}\right]^{\hat{g}_0^n(E)}. \qquad (12)$$

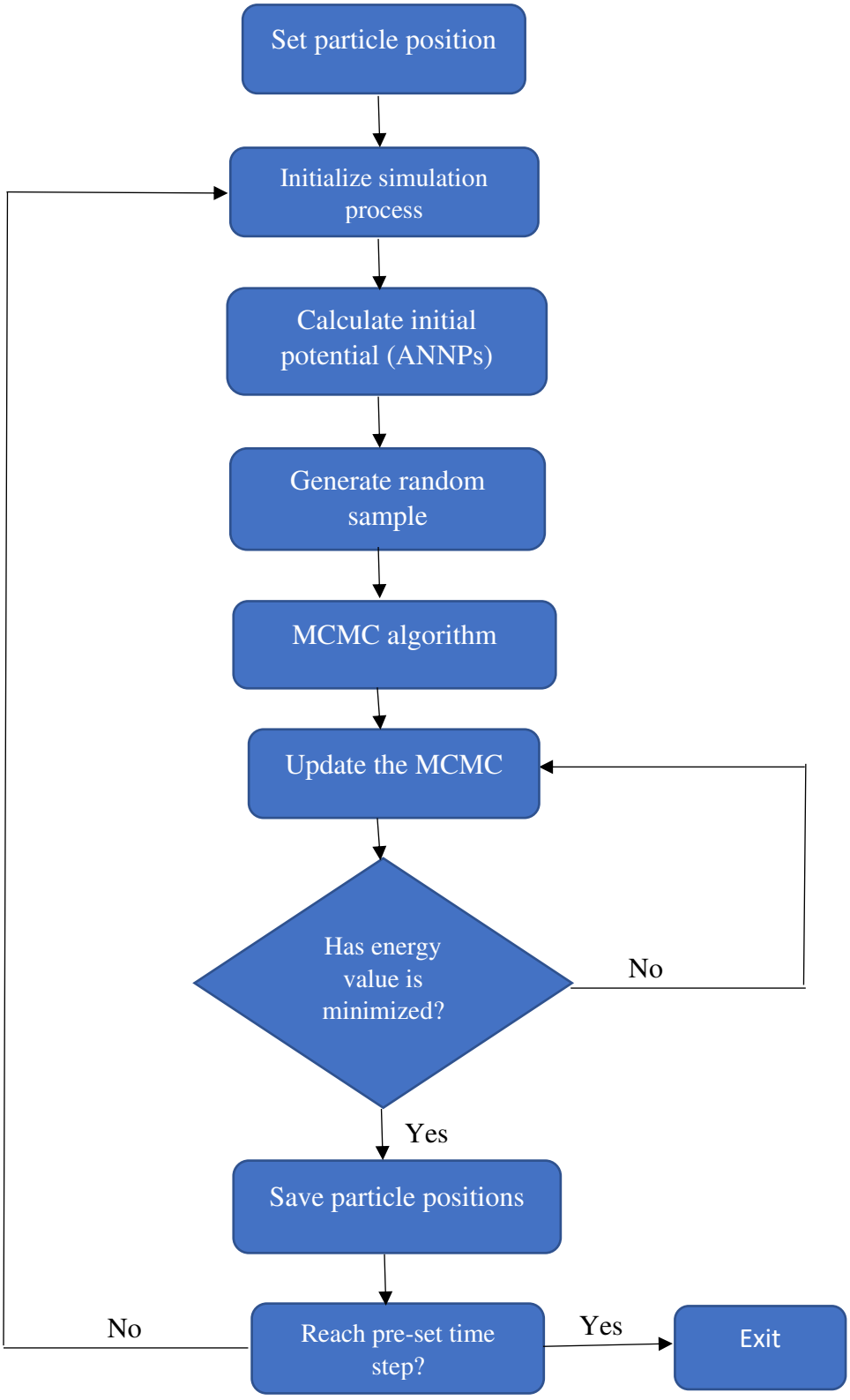


Figure 2: Flow chart of MCMC.

Chapter 3

Fortran Code

3.1 Main Code

```
      Program GMIN
      IMPLICIT NONE
      INCLUDE 'params.h'
      INCLUDE 'commons.h'
      INTEGER J1
      DOUBLE PRECISION POTEL
      COMMON /POT/ POTEL
      LOGICAL EVAP
      COMMON /EV/ EVAP
c
      PRINT*
c
      CALL KEYWORD
c
      IF (DUMPT) THEN
         OPEN (UNIT=40, FILE='QUENCH.P', STATUS='UNKNOWN')
         OPEN (UNIT=39, FILE='QUENCH.E', STATUS='UNKNOWN')
      ENDIF
c
      CALL IO1
c
      IF (CENT) CALL CENTRE(COORDS)
c
      NQ=1
      If (mcsteps(1).le.1) nsave=1
      DO J1=1, NSAVE
         QMIN(J1) = 1.0D10
         QMINB(J1) = 1.0D10
         QRSQ(J1) = 0.0d0
         QRSQB(J1) = 0.0d0
      ENDDO
c
      IF (NRUNS.GT.0) CALL MCRUNS
c
      STOP
      END
```

3.2 Code for Monte Carlo simulations

```
      SUBROUTINE MCRUNS
      IMPLICIT NONE
      INCLUDE 'params.h'
      INCLUDE 'commons.h'
      INTEGER J1
c
      DO J1=1, NRUNS
         CALL MC(MCSTEPS(J1), TFAC(J1))
      ENDDO
c
       RETURN
      END
c
      SUBROUTINE MC (NSTEPS, SCALEFAC)
      IMPLICIT NONE
      INCLUDE 'params.h'
      INCLUDE 'commons.h'
      INTEGER J1, NSUCCESS, NFAIL, NFAILT, NSUCCESST, J2,
     1        NSTEPS, ITERATIONS, NQTOT, nacc, j, jj
      DOUBLE PRECISION POTEL, SCALEFAC, X(3*MXATMS),
     1        RANDOM, EPPREV, emini, emaxi, xt, yt, zt
      LOGICAL EVAP, ATEST
      COMMON /EV/ EVAP
      COMMON /POT/ POTEL
      COMMON /TOT/ NQTOT
c
      NQTOT=0
      NSUCCESS=0
      NFAIL=0
      NSUCCESST=0
      NFAILT=0
c
      emini= 1.0d10
      emaxi=-1.0d10
c
c     Calculate the initial energy and save in EPREV
c
      WRITE (*, '(A)') 'Calculating initial energy''
      DO J1=1,3*NATOMS
         X(J1) = COORDS(J1)
      ENDDO
```

```
      CALL QUENCH (iterations)
      CALL SAVEITA (POTEL, coords)
      CALL FINALIOA
c
      NQTOT=NQTOT+1
      WRITE (*,'(A, I7, A, F20.10, A, I4, A, F15.7)') 'Quench', NQ,
     1       'energy=' POTEL, 'steps=', ITERATIONS, 'RMS force=', RMS
      EPREV=POTEL
      EPPREV=0.0D0
      WRITE (*, '(A, I10, A)') 'Starting MC run of ', NSTEPS,' steps'
      WRITE (*, '(A, F15.8, A)')
     1            'Temperature will be multiplied by ',
     2            SCALEFAC, ' at every step'
      call flush (6)
c
      if (IMCMC.eq.1) NSTEPS =NEQUI
c
      DO J1=1, NSTEPS
          NQ=NQ+1
          CALL TAKESTEP
          CALL TAKESTEPBASIC
          CALL QUENCH (iterations)
          NQTOT=NQTOT+1
            WRITE (*, '(A, I7, A, F20.10, A, I4, A, F15.7)') 'Quench', NQ,
     1       'energy=', POTEL, 'steps=', ITERATIONS, 'RMS force=', RMS
       IF (EVAP) THEN
             NFAIL=NFAIL+1
             DO J2=1,3*(NATOMS-NSEED)
                   COORDS(J2) = COORDSO(J2)
             ENDDO
             DO J2=1, NATOMS
                  VAT(J2) =VATO(J2)
             ENDDO
       ELSE
          CALL TRANSITION (POTEL, EPREV, ATEST, RANDOM)
          IF (ATEST) THEN
               NSUCCESS=NSUCCESS+1
               EPPREV=EPREV
               EPREV=POTEL
                  if (potel.lt. emini) emini=potel
                  if (potel.gt. emaxi) emaxi=potel
          ELSE
               NFAIL=NFAIL+1
```

```
            DO J2=1,3*(NATOMS-NSEED)
                 COORDS(J2) = COORDSO(J2)
            ENDDO
            DO J2=1, NATOMS
                VAT(J2) = VATO(J2)
            ENDDO
          ENDIF
        ENDIF
c
c     Check the acceptance ratio.
c
        call saveita (potel, coords)
        call finalioa
c
        IF ((MOD (J1, NACCEPT). EQ. 0). AND. (NSEED.EQ.0)) THEN
            IF (DELOAT(NSUCCESS) / DFLOAT (NSUCCESS +NFAIL.
     1           GT. ACCRAT) THEN
                IF (FIXBOTH) THEN
                ELSE IF (FIXSTEP) THEN
                    IF (. NOT. FIXTEMP) TEMP=TEMP/1.05.DO
                ELSE
                    STEP=STEP*1.05D0
                   ASTEP=ASTEP*1.05D0
                ENDIF
            ELSE
                IF (FIXBOTH) THEN
                ELSE IF (FIXSTEP) THEN
                    IF (. NOT. FIXTEMP) TEMP=TEMP*1.05D0
                ELSE
                    STEP=STEP/1.05D0
                    ASTEP=ASTEP/1.05D0
                ENDIF
            ENDIF
            IF (FIXBOTH) THEN
            ELSE IF (FIXSTEP) THEN
                IF (. NOT. FIXTEMP)
     1           WRITE (*,'(A, F12.6)') ' Temperature is now:', TEMP
            ELSE
                WRITE (*,'(A, F12.6, A, F12.6)')
     1          ' Maximum center-of-mass and angular steps are now: ', STEP,
     2           ' and ', ASTEP
            ENDIF
            WRITE (*,'(A, I4, A, F15.7)')
```

```
     1          ' Acceptance ratio for previous', NACCEPT,' steps=',
     2           DFLOAT(NSUCCESS)/DFLOAT(NSUCCESS+NFAIL)
          NSUCCESST=NSUCCESST+NSUCCESS
          NFAILT=NFAILT+NFAIL
          NSUCCESS=0
          NFAIL=0
       ENDIF
       TEMP=TEMP*SCALEFAC
       FLUSH (6)
       IF (HIT) GOTO 37
       ENDDO
c
 37    CONTINUE
        WRITE (*,10) DFLOAT(NSUCCESST) / MAX (1.0D0, DFLOAT
     1          (NSUCCESST+NFAIL)), STEP,ASTEP,TEMP
 10     FORMAT (' Acceptance ratio for run=', F12.5, ' Step='F12.5,
     1              ' Angular step factor=', F12.5,' Temperature=', F12.5)
c
       if (IMCMC.eq. 0) then
          MUPD=0
          emin = emini
          emax = emaxi
       endif
c
       call mcmc
c
       RETURN
       END
```

3.3 Multicanonical monte Carlo simulation algorithm

Code for Multi canonical monte Carlo simulation

```
       SUBROUTINE MCMC
       IMPLICIT NONE
       INCLUDE 'params.h'
       INCLUDE 'commons.h'
       INTEGER J1, L, I, IT, N, J, JJ, K, nac, nre
       DOUBLE PRECISION delE, dhi, dhii, ai, bt, potel
       DOUBLE PRECISION RANG, mini, maxi, beta0, ebinr, ebinl, hterm,
     1                  so, sn
       DOUBLE PRECISION enew, eold, elowest, acr
       DOUBLE PRECISION sumg (300,100), wei (100), ener (100)
       DOUBLE PRECISION beta (100), betap (100), alpha (100)
```

```
      DOUBLE PRECISION sold (100), snew (100)
      INTEGER ITER, H (100), M, NL, NITER, attest, nbin, idummy, iconv
       COMMON /POT/ POTEL
c
c     **************READ PARAMETERS*************
c     START OF MCMC

      write (21, *) "Start of mcmc simulation"
      write (21, *) "Temp=", TEMPA

      nl = nbins
      write (21, *) "MUPDATES:", MUPD, "NBINS=:", NL
c
      acr =1.0d0
      iconv=1
c
      beta0 = 1.0/TEMPA      ! initial temperature
c
      elowest=emin
      eold = emin
c
      write (21, *) "emin=:", emin, "emax=:", emax
c
c     number of bins is 10
      rang = emax-emin
      delE = rang/(nl-2)
c
      write (21, *) "deltaE: ", delE
c
c     ********END READ PARAMETERS**********
c
      call constH (emin, emax, dele, nl, ener)
c
      do k = 1, mupd
      do i = 1, nl
          sumg (k, i) = 0.0d0
      enddo
      enddo
c
c      ************INITIALIZE beta, alpha and wei********
c
      do i = 1, nl
          beta(i) = beta0
```

```
        alpha(i) = 0.0d0
        betap(i) = beta(i)
c     assign a boltzman weight for each bin
      if (I.lt.nl) then
      sold(i)= (beta(i)*ener(i)) – alpha(i)
      else
        sold(i) =0.00d0
      endif
      enddo
c
c     LOOP FOR MCMC UPDATE
c
      DO i = 1, nl
         h(i) = 0.0d0
      Enddo
c
      DO K= 1, mupd
c
      call quench(idummy)
      call update (emin, emax, dele, nl, h, potel, nbin)
      sold(nbin)=(beta(nbin)*potel)-alpha(nbin)
              so = sold(nbin)
c
c     INNER LOOP FOR MC STEPS
c
      nac=0
      nre=0
      do n = 1, nitera
c
c     Quench initial random xyz and update
c
      call takestep
      call finaliob
c
      if (potel.lt. elowest) then
           elowest=potel
           call saveitb (potel, coords)"
           call finaliob
      endif
c
      call getbin (emin, emax. dele, nl, potel, nbin)
c
      bt = beta(nbin) * potel
```

```
      ai = alpha(nbin)
      snew(nbin) = bt – ai
      sn = snew(nbin)
c
      attest = -1
c
      call checkw (sn, so, atest)
c
      if (attest. eq. 1) then
         nac = nac + 1
         so = sn
         eold= potel
      else
         nre = nre+1
      do j=1,3*natoms
         cords(j)=coordso(j)
      enddo
      endif
c
      call update (emin, emax, dele, nl, h, eold, nbin)
c
      if (mod (n, naccept). eq. 0) then
c
         acr = dble(nac)/dble(nac+nre)
c
         if (acr.gt. 0.5d0) then
            step = step *1.050d0
            astep =astep *1.050d0
         else
            step = step *0.952d0
            astep =astep *0.952d0
         endif
 c
         nac=0
         nre=0
      endif
c
      call saveitb (potel, coords)
      call finaliob
c
      enddo
c
      call updateAB (nl, h, sumg, ener, alpha, beta, betap, beta0, dele, k)
```

```
c
 s     do i =1, nl
         betap(i)=beta(i)
       enddo
       do i = 1, nl
         h(i) =0
       enddo
       enddo

      RETURN
      END
c
      Subroutine updateAB (1, his, sumg, ehis, a, b, bp, b0, del, mu)
      implicit none
      integer 1, i, his (100), mu, m
      double precision ehis (100), sumg (300,100), hhh (100)
      double precision a(100),b(100),bp(100),g(100),gg(100)
      double precision dhi,dhii,hterm,del,b0,dn
c
      do i = 1-2, 1, -1
c
c     calculates correction coefficient g, gg
c
           if (his(i). gt. 0.and. his(i+1) .gt. 0) then
               dhi = dble(his(i))
               dhii = dble(his(i+1))
               g(i) = dhi *dhii / (dhi + dhii)
               sumg (mu, i) = g(i)
c
c        get denominator
c
               dn = 0.0d0
               do m = 1, mu
                  dn = dn + sumg (m, i)
               enddo
               gg(i) = g(i) / dn
c        Hterm calculation
               hterm = (log(dhii) – log(dhi)) /del
               hhh(i) = hterm
               b(i) = bp(i) + (gg(i)*hhh(i))
            else
               hhh(i) =0.00d0
               gg(i) = 0.00d0
```

```
            sumg (mu, i) =0.0d0
            b(i) = bp(i)
          endif
      end do
c       now calculates alpha
      do i = 1-2, 1, -1
      a(i) = a(i+1) + ((b(i)-b(i+1)) * ehis(i))
      enddo
      end
```

Chapter 4

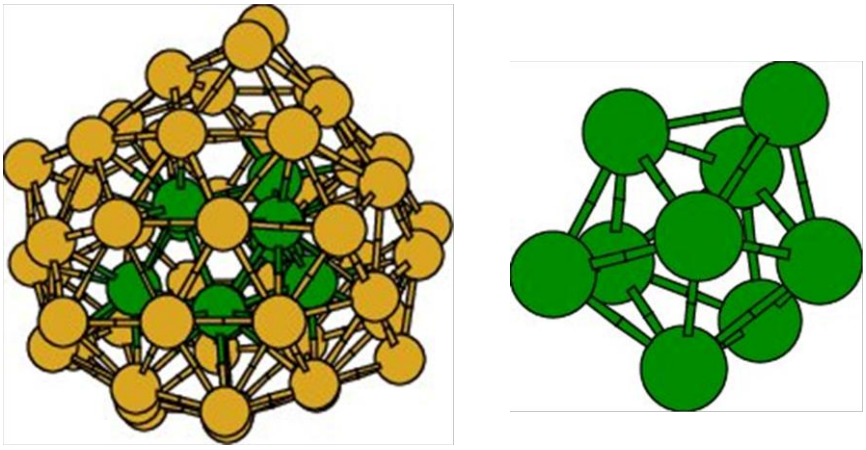
Results and Discussions

Here, we have performed a global structure search for low-energy clusters of Au_n, where n = 59, 60, 68, 70, 92, 106, 112, 138, 156, 166, 168, by using MCMC simulations combined with ANNPs and found that they exhibit core-shell type structures. For the LM search of Au_n, initial we took a random structure for MCMC simulations run at 366 K. The magic number sized clusters were selected i.e. Au_n (n = 68, 70, 92,106, 112, 138, 156, 166, 168) and the most stable assumed structure for it is an icosahedron core-shell type structure. Here, we will discuss each stable cluster in detail. Table 1. having the no of gold atoms in Au_n as its first column, no of atoms in core structure as its second column, lowest energy of the cluster (eV) as third column and no of steps that we have done to get these lowest energy structures as the fourth column of the table. The recognized structural evolution of the core-shell gold nanoclusters will be useful for future catalysis research.

Number of atoms in Au_n	Number of core structure atom	The energy of structure (eV)	Number of steps
59	9	-162.44	14,400
60	9	-165.48	15,048
68	12	-189.08	16,952
70	13	-195.48	13,350
92	20	-260.49	9,141
106	25	-302.46	12,214
112	28	-320.78	12,983
138	39	-398.28	8,416
156	51	-453.59	5,926
166	53	-484.62	4,540
168	55	-490.05	4,819

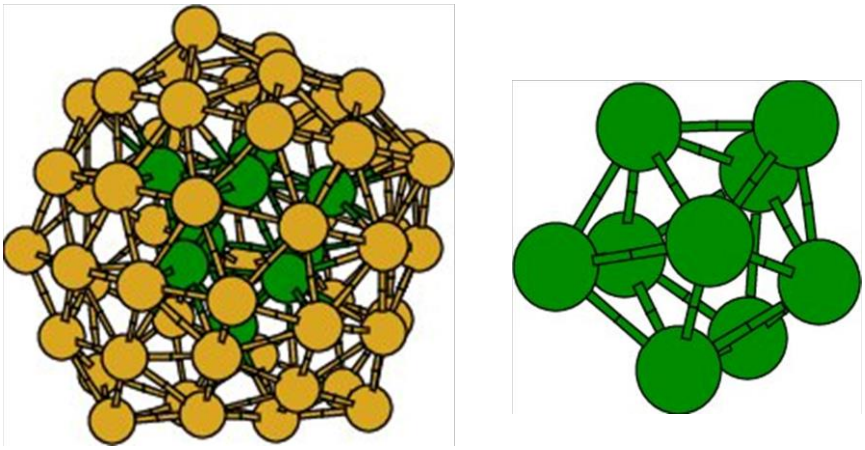
Table 1: Energy and the number of core structures atom of the structure of Au_n.

For Au_{59} and Au_{60} -: For the LM search for Au_{59}, the MCMC simulations were run at 366 K up to 14,400 steps using a random initia l structure such that many structures are obtained from these steps. Using ANNPs with simulations, the lowest energy observed for the structure having 50 atoms in the outer shell (colored yellow) and 9 atom core-shell shown in green in figure 3. For the LM search for Au_{60}, 15,048 steps of MCMC simulation were run using random initial structure at 366 K. Many structures were obtained from these steps and the structure observed with the lowest energy have 51 atoms in the outer shell (colored yellow) and 9 atom core-shell i.e. same as Au_{59} in green color as shown in figure 4.



(a) (b)

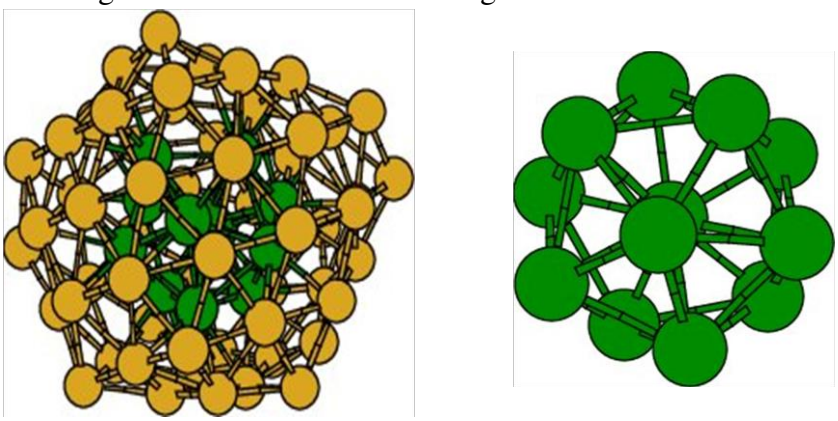
Figure 3: (a) low-energy structure of Au_{59} and (b) 9-atom core structure of Au_{59}.



(a) (b)

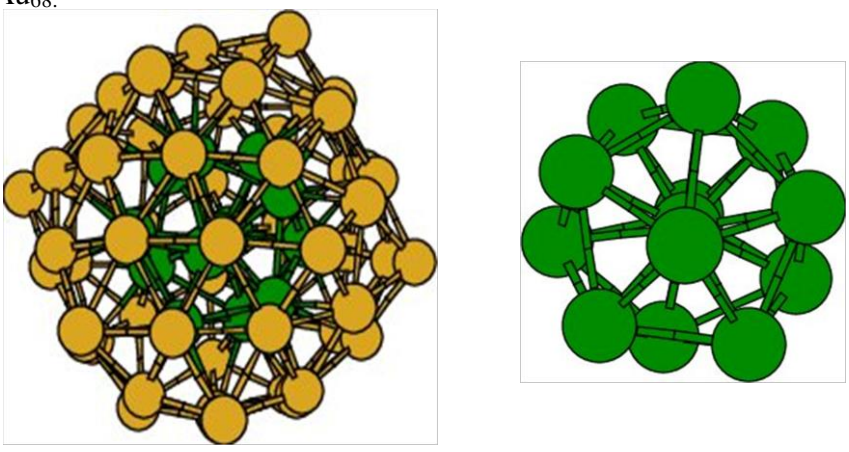
Figure 4: (a) low-energy structure of Au_{60} and (b) 9-atom core structure of Au_{60}.

For Au_{68} and Au_{70} -: For the LM search for $Au_{68,}$ 16,952 steps of MCMC simulation were run using a random initial structure at 366 K and the lowest energy structure from these observed in yellow-colored outer shell consists of 56 atoms The core-shell of the low-energy structure of Au_{68} having a 12-atoms structure, in which 11 atoms are surrounding the one central atom but one apex atom is missing here. That's why the arrangement of the core looks like an icosahedron but not the perfect icosahedron which is in green color that is shown in figure 5. For the LM search for $Au_{70,}$ 13350 steps of MCMC simulation were run using a random initial structure at 366 K and we observed the lowest energy structure of Au_{70} using ANNPs that has yellow-colored outer shell consists of 57 atoms and the core-shell consists of 13-atoms in which the outer 12 atoms are surrounding the one central atom. This arrangement of the core structure is perfect icosahedron which is in green color that is shown in figure 6.



(a) (b)

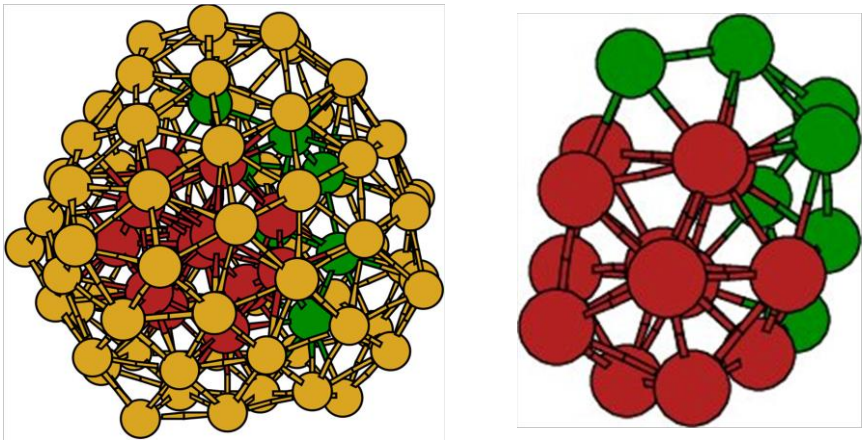
Figure 5: (a) low-energy structure of Au_{68} and (b) 12-atom core structure of $Au_{68.}$



(a) (b)

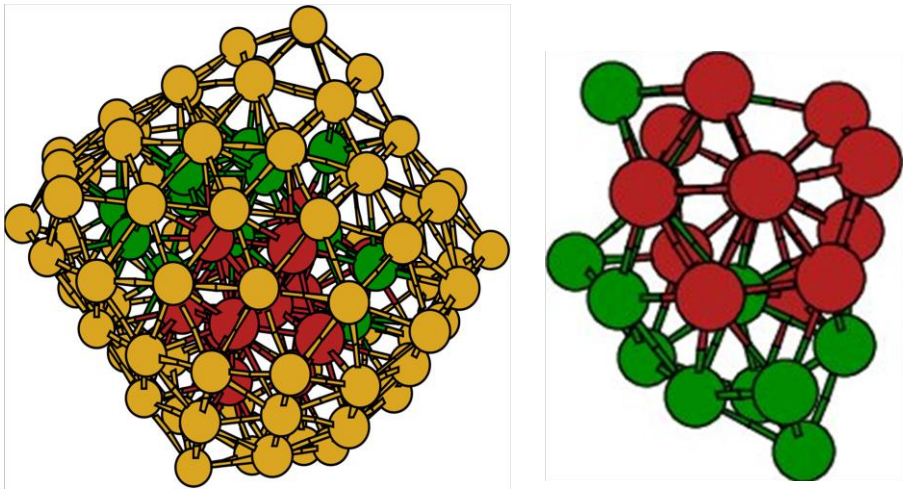
Figure 6: (a) low-energy structure of Au_{70} and (b) 13-atom core structure of Au_{70}.

For Au_{92} and Au_{106} -: For the LM search for Au_{92}, 9,141 steps of MCMC simulations were run using a random initial structure at 366 K. Many structures are obtained from these steps and the lowest energy structure consists of 72 atoms in the outer shell (colored yellow) and 20 atoms core-shell which are arranged red color of 13 atoms in an icosahedral manner with 7 bridging atoms which are in green color for better visualization. This is not the regular icosahedron structure, because the lower ring is leaning towards the bridged atom that deviates it from the perfect geometry as shown in figure 7. For the LM search of Au_{106}, 12,214 steps of MCMC simulation were run using a random initial structure at 366 K and we observed the lowest energy structure from these simulation steps which has yellow-colored outer shell consists of 81 atoms and the core-shell of Au_{106} consists of 25 atoms which are arranged red color of 13 atoms in an icosahedral manner with 12 bridging atoms which are in green color. Here also, the lower ring is more leaned towards the bridged atom that's why it is not the regular icosahedral structure that is shown in figure 8.



(a) (b)

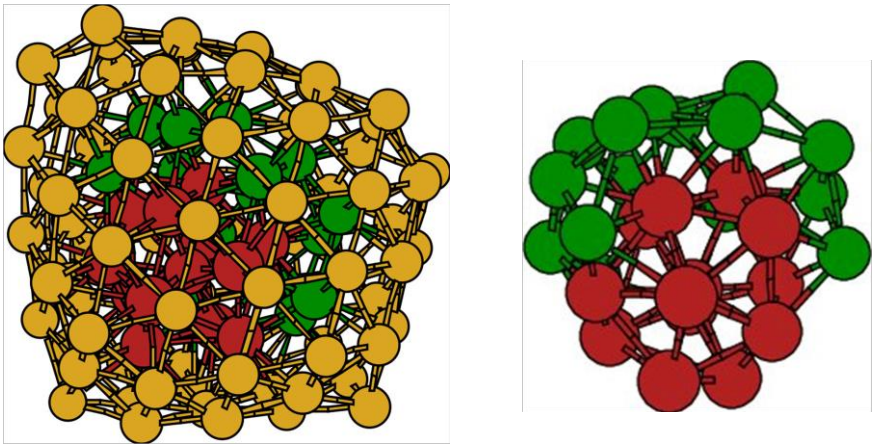
Figure 7: (a) low-energy structure of Au_{92} and (b) 20-atom core structure of Au_{92}.



(a) (b)

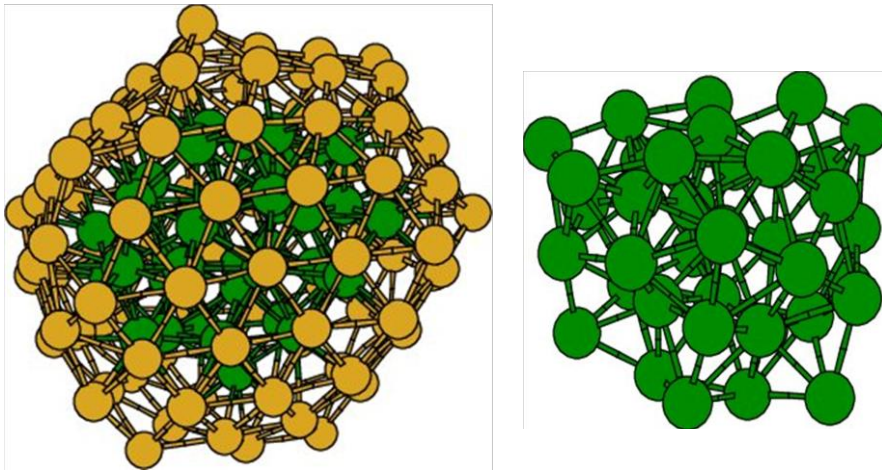
Figure 8: (a) low-energy structure of Au_{106} and (b) 25-atom core structure of Au_{106}.

For Au_{112} and Au_{138} -: For the LM search of Au_{112}, 12983 steps of MCMC simulation were ran using a random initial structure at 366 K and we observed the lowest energy structure from these simulation steps which has yellow-colored outer shell consists of 84 atoms and the core-shell of Au_{112} consists of 28 atoms which are arranged red color of 13 atoms in an icosahedral manner with 15 bridging atoms which are in green color. This is not the regular icosahedral structure, because the lower ring is leaning towards the bridged atom as shown in figure 9. For the LM search of Au_{138}, 8,416 steps MCMC simulation were run using a random initial structure at 366 K. we observed the lowest energy structure from these simulation steps which has yellow-colored outer shell consists of 99 atoms and the core-shell of Au_{106} consists of 39 atoms but their arrangement is quite different from the icosahedral symmetry which is in green color that is shown in figure 10.



(a) (b)

Figure 9: (a) low-energy structure of Au_{112} and (b) 28-atom core structure of Au_{112}.



(a) (b)

Figure 10: (a) low-energy structure of Au_{138} and (b) 39-atom core structure of Au_{138}.

For Au_{156} -: For the LM search for Au_{156}, 5,926 steps of MCMC simulations were run using a random initial structure at 366 K. Many structures are obtained from these steps and we observed the lowest energy from these simulations using ANNPs that have 105 atoms in the outer shell (colored yellow) and arrangement of the atom in the outer shell-like an icosahedral but not perfect icosahedral structure that is shown in figure 11(a). and the core-shell consists of 51 atoms which are arranged in an icosahedral manner with few missing atoms as shown in figure 11(b). While the inner core-shell consists of red color 12 atoms which are arranged in a perfect icosahedral manner as shown in figure 11(d). And the outer core consists of green color of 39 atoms which are arranged in an icosahedral manner with few missing atoms. That's why it is not a regular icosahedral structure is shown in figure 11(c).

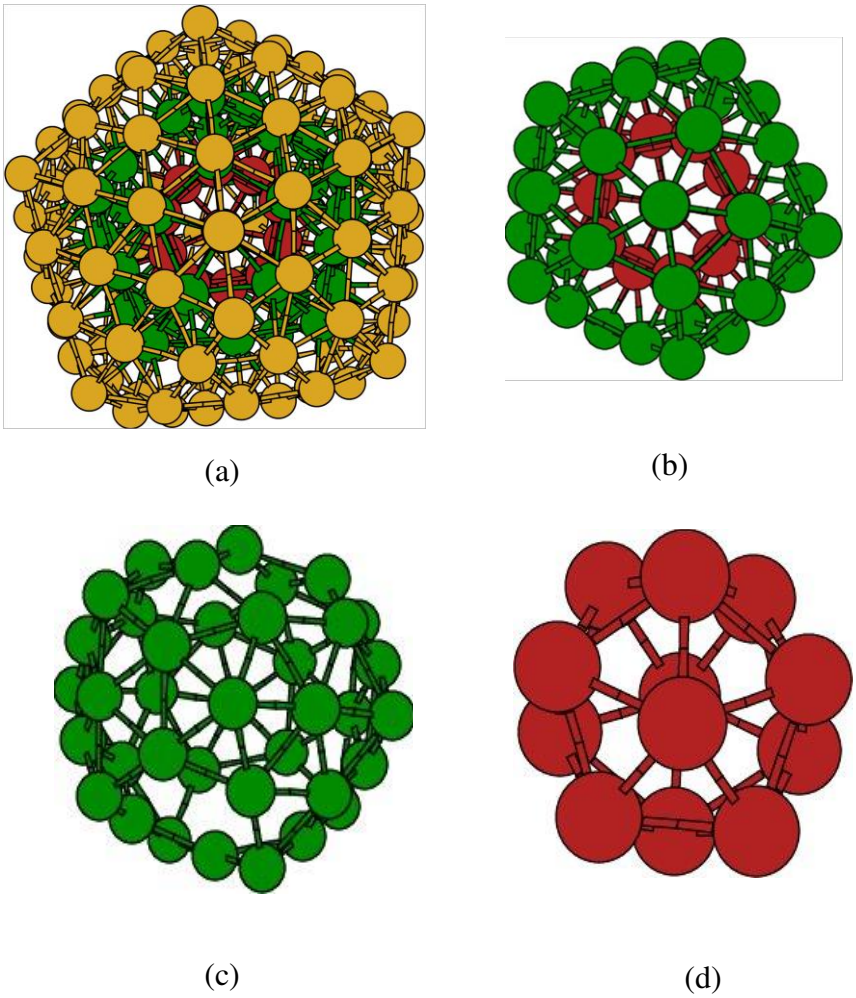
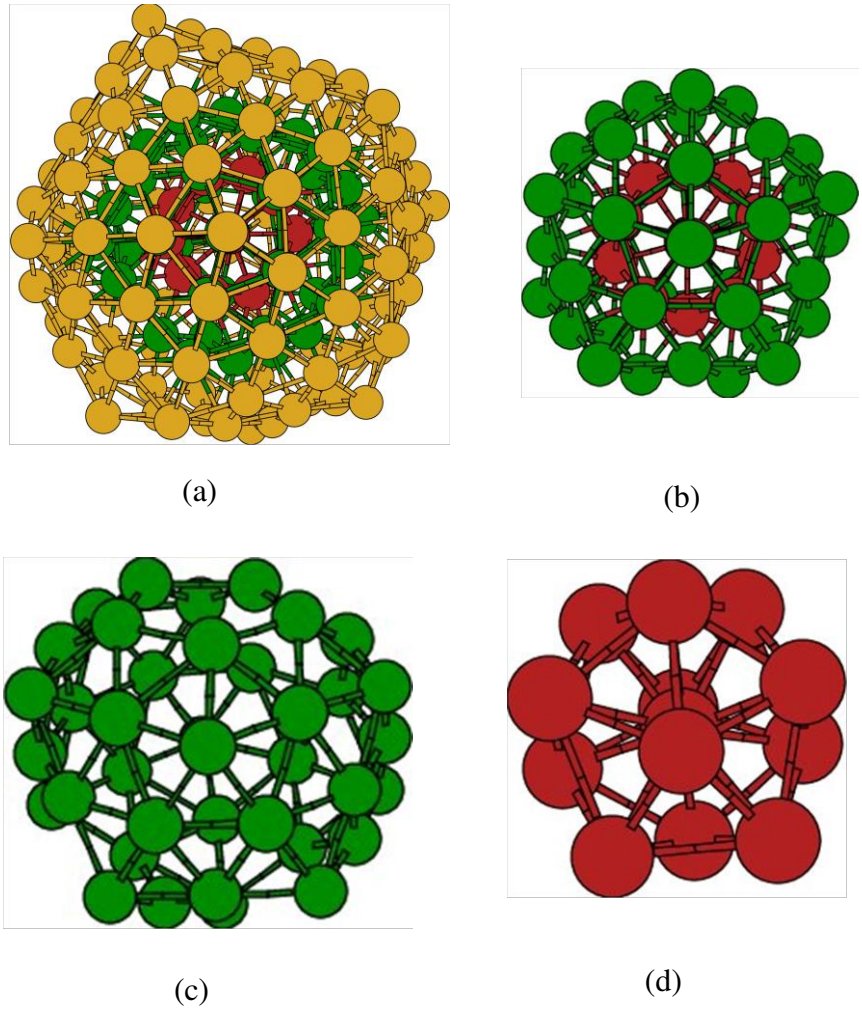


Figure 11: (a) low-energy structure of Au_{156}, (b) 51-atom core structure of Au_{156}, (c) 39-atom outer core structure of Au_{156}, and (d) 12-atom inner core structure of Au_{156}.

For Au_{166} -: For the LM search for Au_{166}, 4540 steps of MCMC simulations were run using a random initial structure at 366 K and from these simulation steps, we observed the lowest energy from these simulations using ANNPs that have 113 atoms in the outer shell (colored yellow) that is shown in figure 11(a). and the core-shell is having a 53-atom structure in which atoms are arranged in an icosahedral manner with few missing atoms as shown in figure 12(b). The inner core-shell consists of a red color of 13-atom structure that is a perfect icosahedron having a central atom as shown in figure 12(d). while the outer core-shell structure consists of green color of the 40-atom structure, having the icosahedral geometry with few missing atoms that deviates it from perfect geometry that is shown in figure 12(c).

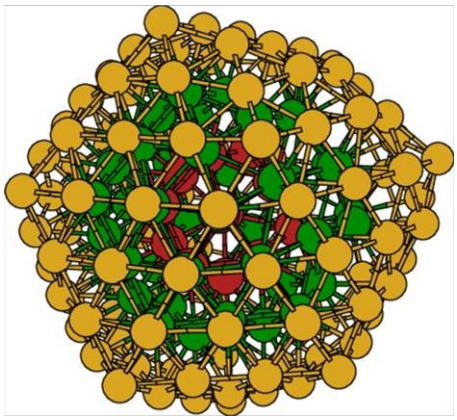


(a) (b)

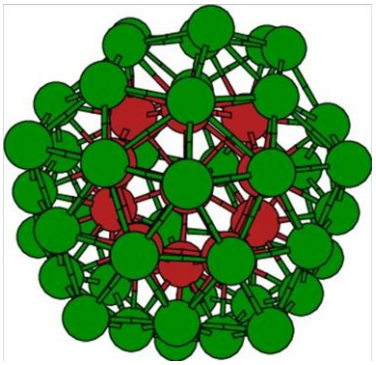
(c) (d)

Figure 12: (a) low-energy structure of Au_{166}, (b) 53-atom core structure of Au_{166}, (c) 40-atom outer core structure of Au_{166}, and (d) 13-atom inner core structure of Au_{166}.

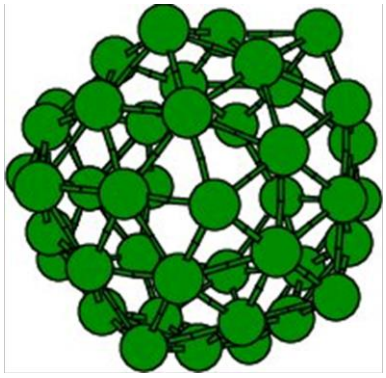
For Au_{168} -: For the LM search of Au_{168}, 4819 steps of MCMC simulations were run using a random initial structure at 366 K and we observed the lowest energy structure from these simulation steps which has yellow-colored outer shell consists of 113 atoms and arrangement of the atom in the outer shell-like an icosahedral but not perfect icosahedral structure that is shown in figure 13(a). the core-shell of Au_{168} consists of 55 atoms structure, with the irregular icosahedral geometry i.e. shown in figure 13(b). The inner core-shell consists of the red color of 12 atoms structure, having the icosahedron arrangement with the central atom but one apex atom is missing as shown in figure 13(d). And the outer core-shell consists of green color of the 43-atom structure, having icosahedral geometry but not regular as shown in figure 13(c).



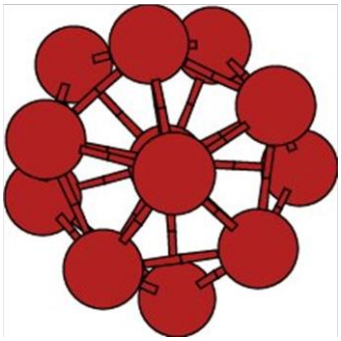
(a)



(b)



(c)



(d)

Figure 13: (a) low-energy structure of Au_{168}, (b) 55-atom core structure of Au_{168}, (c) 43-atom outer core structure of Au_{168}, and (d) 12-atom inner core structure of Au_{168}.

We performed structure evolution of low energy gold nanoclusters Au_n (n = 59, 60, 68, 70, 92, 106, 112, 138, 156, 166, 168) exhibit core-shell type structures by using MCMC simulations combined with ANNPs. From these LM structures of different sized structures, we observed that as the number of gold atom increases, the core-shell atoms increase and stability of core-shell is also increased. As we choose the size selected magic number cluster, so the most stable assumed structure for it is an icosahedron. For the above structures of Au_n (n = 59, 60, 68, 70) we found that as the number of gold atom increases, the stability of the core-shell structure also increases. In the low-energy structure of $Au_{70,}$ the arrangement of core-shell is perfect icosahedron as shown in figure 6(b) as compared to Au_n (59, 60, 68) as shown in figure 3(b), 4(b) and 5(b) respectively. Also for the structures of Au_n (n = 92, 106, 112, 138, 156, 166, 168) the stability of the core-shell structure increases as the number of gold atom increases and formed a new layer core-shell which is also going to be like an icosahedral stable structure. Therefore, we can conclude that the structure evolution of gold nanoclusters Au_n (n = 59, 60, 68, 70, 92, 106, 112, 138, 156, 166, 168) exhibit stable icosahedral core-shell type structure using MCMC simulations with ANNPs.

Chapter 5

Conclusion

In this project, we have evaluated the core-shell of gold nanoclusters Au_n, where n = 59, 60, 68, 70, 92, 106, 112, 138, 156, 166, 168. By this evaluation, we found that the most stable structure of gold nanoclusters exhibit core-shell type structures. The stable structure of Au_n (n = 59, 60) exhibits 9-atom core, Au_{68} exhibits a 12–atom icosahedral core with one missing apex atom, and Au_{70} exhibits icosahedral core structure. While the stable structure of Au_n (n = 92, 106, 112) exhibits 20-atom, 25-atom and, 28-atom respectively, which are arranged in an icosahedral manner with bridging atom that is not the regular icosahedron structure. In the stable structure of Au_{138}, the core consists of 39 atoms but their arrangement is different from the icosahedral symmetry. The core consists of 51-atom and 53-atom structure of Au_{156} and Au_{166} in which atoms are arranged in an icosahedral manner with few missing atoms. While the core consists of a 55-atom structure of Au_{168} with the irregular icosahedral geometry. Here, we conclude that as the number of gold atom increases, the core-shell atoms also increase which leads to an increase in the stability of the core-shell structure. The recognized structural evolution of the core-shell gold nanoclusters will be useful for future studies of the structure-catalytic-activity relationship.

REFERENCES

[1] Zhan L., Chen J. Z. Y., Liu W., and Lai S. K. (2005), Asynchronous multicanonical basin hopping method and its application to cobalt nanoclusters. J. Chem. Phys., 122, 244707.

[2] Allen M. P., and Tildesley D. J. (1987), Computer simulation of liquids, Oxford University Press Inc., New York, pp. 110-123.

[3] Pande S., Gong X., Wang L., and Zeng X. C. (2019), Au_{60}^-: The Smallest Gold Cluster with the High-Symmetry Icosahedral Core Au_{13}. J. Phys. Chem. Lett., 10, 1820-1827.

[4] Pande S., Huang W., Shao N., Wang L., Khetrapal N., Mei W., Jian T., Wang L. S., and Zeng X. C. (2016), Structural Evolution of Core-Shell Gold Nanoclusters: Au_n^- (n= 42-50). ACS Nano., 10, 10013-10022.

[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] Jindal S., Chiriki S., and Bulusu S. S. (2017), Spherical harmonics based descriptor for neural network potentials: Structure and dynamics of Au_{147} nanocluster. J. Chem. Phys., 146, 204301.

[7] Berg B. A. (1998), Algorithmic aspects of multicanonical simulations. Nuclear Physics B., 63A-C, 982-984.