MACHINE LEARNING APPROACH TO SOLVE SCHRODINGER EQUATION FOR A ONE DIMENSIONAL SYSTEM IN A GAUSSIAN POTENTIAL WELL AND IN A HARMONIC POTENTIAL WELL

CH-800

M.Sc Thesis

By

APARNA GANGWAR

(1903131014)

DEPARTMENT OF CHEMISTRY

INDIAN INSTITUTE OF TECHNOLOGY INDORE

June 2021

MACHINE LEARNING APPORACH TO SOLVE SCHRODINGER EQUATION FOR A ONE DIMENSIONAL SYSTEM IN A GAUSSIAN POTENTIAL WELL AND IN A HARMONIC POTENTIAL WELL

A THESIS

Submitted in partial fulfillment of the

requirement for the award of the degree

of

Master of Science

by

APARNA GANGWAR

DEPARTMENT OF CHEMISTRY

INDIAN INSTITUTE OF TECHNOLOGY INDORE

June 2021

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the Thesis entitled **Machine learning approach to solve Schrodinger Equation for a One Dimensional System in a Gaussian Potential Well and in a Harmonic Potential Well** 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 2020 to June 2021 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.

Aparna

(Aparna Gangwar)

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)

APARNA GANGWAR has successfully given his/her M.Sc. Oral Examination held on 08/06/2021.

Signature of Supervisor of MSc thesis	Convener, DPGC
Date:	Date:
Signature of PSPC Member	Signature of PSPC Member
Date: 10-06-2021	Date: 09-06-2021

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 master's project. I have been extremely lucky to have a supervisor who cared so much about my work and promptly responded to my questions and queries. I would also like to thank my PSPC members Dr. Anjan Chakraborty and Dr. Tridib Kumar Sharma.

I would like to acknowledge this assistance given to me by my lab senior, 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.

Aparna Gangwar

Abstract

It is well known that the solution of the Schrodinger equation beyond the hydrogen atom is a non-trivial problem. Our main objective in the thesis is to use machine learning techniques to solve Schrodinger equation. In the study we considered one dimensional Schrodinger wave equation in Gaussian potential well and also in harmonic potential well as a case study to use machine learning techniques. Since the 1- dimensional Schrodinger equation in Gaussian potential well has no analytical solution, we used the Numerov method to solve it. Similarly, the Numerov method is used to solve the 1- dimensional Schrodinger wave equation in harmonic potential well. Using the numerov method, we calculated wave functions, probability densities and energies for systems with single electrons, 2 electrons, 3 electrons and 4 electrons. Now our aim is to map the probability densities to energies using artificial neural networks. For this we made a dataset of about 5000 probability densities using the Numerov method and train these probability densities to the known energies obtained. The dataset is obtained by randomly changing the parameters of Gaussian potential well and the force constant values in harmonic potential well. The inputs for the artificial neural networks will be probability densities and the output will be total energies of the system. Such models can be used to calculate energies of 1-dimensional Schrodinger equation in a similar but unknown potential energy well without really solving the Schrodinger equation analytically.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS/ UNITS

J	Joule (Kgm^2s^{-2})
V	Potential
Y	Wave function
E	Energy
m	Kg
h	Planck constant (Js)
$\frac{m}{\hbar^2}$	$Kg\ (Js)^{-2}$

ACRONYMS

MD – Molecular Dynamics

PES – Potential Energy Surface

DFT – Density Functional Theory

ANN – Artificial Neural Network

NN – Neural Network

Chapter 1

Introduction

1.1General Introduction:

It is well known that the solution of the Schrodinger equation beyond hydrogen is a non – trivial problem due to the presence of an inter-electronic repulsion term, and there is no analytical way to solve it. The Born-Oppenheimer approximation [1] also justified this because the Hamiltonian or the ground state potential energy is completely defined by the atomic positions, nuclear charges, and the total charge of the system. From here, we observe that a well-defined relationship exists between the atomic structure and its potential energy. It seems like that for reliable computer simulations in chemistry, physics, and material science, the accurate description of the atomic interactions is of vital importance. Based on quantum mechanical laws [2], various electronic structure methods can calculate the potential energy and the nuclear forces for a given atomic configuration using the Born- Oppenheimer approximation.

But in the calculation of energies and forces of any system, the choice of the electronic structure [3] method plays a significant role. The choice of the electronic structure method depends on the system. It usually requires an acceptable compromise between efficiency and accuracy for the problem of interest because we want to calculate the exact solution of the Schrodinger equations. And we know that it is impossible for essentially all problems beyond the hydrogen atom. The calculation of each electronic structure then provides a particular point on the multidimensional potential-energy surface (PES), which is a real-valued function and depending upon the atomic coordinates which are used to determine the potential energy of the system. The number of electronic structure energies that can be calculated and stored is limited. Consequently, in abinitio MD [4], density functional theory (DFT) [5, 6], naturally "on the fly," is used to calculate the energies and forces. Alternatively, an analytic expression for the PES can be constructed and used in the simulations, allowing performing MD simulations more efficiently. The estimation of such types of expressions is much faster than solving the quantum mechanical problem. The Born-Oppenheimer approximation also justifies this approach. From the above discussion, we observe that a well-defined relationship exists between the atomic structure and its potential energy. So, suppose we do not have any information about the atomic structure, inter-electronic repulsion, nuclear charges, and the total charges. In that case, we cannot calculate the potential energy of any system.

Artificial neural network (ANN) [7], which is inspired by the biological neural network, is one of the most important machine learning theory methods. Just like our brain, an artificial neural network consists of a network of neurons. We can say that an artificial neural network is an attempt to make a computer model of the brain. The

working of an artificial neural network is inspired by but not identical to the biological neurons. Examples of ANN are feed-forward neural network [8], radial basis function network [9], and restricted Boltzmann machine [10]. As a universal approximator [11], an ANN can be used to represent functions. Naturally, it is possible to use ANN to describe the wave function in a quantum system [12]. Researchers have been trying to combine neural network theory and quantum mechanics [13]. For example, using neural network in the real space to solve differential equations, especially the Schrodinger equation with some specific potential [14].On using neural network, there is no need to change the input to get the desired output. In this method, input values are not changing for a particular result. Only the modification of weights connection between the neurons of a specified network is required. The NNs energy expression is unbiased, which means it does not require any type of system modifications generally applicable to all types of bonding [15]. In this project, we are using machine learning technique, artificial neural networks to map probability densities with energies. Here, we are using probability density as input for artificial neural networks, and the output will be the total energies of the system.

Our main motive in this thesis is to use machine learning techniques to solve Schrodinger equation. In this study we considered one-dimensional Schrodinger wave equation in gaussian potential well and in harmonic potential well as a case study to use machine learning techniques. For our project we selected one-dimensional Schrodinger equation in gaussian potential well. Because 1-dimensional Schrodinger equation in a gaussian potential well has no analytical solution. This project has been taken as a prototype for typical problems.

Potential formula for both systems given below:-

Harmonic Potential well, $V(x) = \frac{1}{2}kx^2$

Gaussian Potential well [16], $V(x) = -\sum_{i=1}^{3} a_i e^{[-(x-b_i)^2/2c_i^2]}$

The well-known fact that the One- Dimensional box for a free particle is easily solvable and has an analytical solution. But with potential it is not solvable analytically. So here, we examined Schrodinger equation with gaussian potential well by using Numerical method i.e. Numerov method. Here, the values of potential energy parameters a, b, and c vary from1 <a< 20, 0.2 <b< 0.8, and 0.01 <c< 0.3. [17]. Solving real problems using numerical methods is costly. So here, we are using machine learning method (Artificial neural networks) to map density with total energy. For this we made a dataset of about 5000 probability densities using the Numerov and train these probability densities to the known energies obtained. The dataset is obtained by randomly changing the parameters of gaussian potential (a, b, and c) and the force constant values in harmonic oscillator potential.

1.2 Organization of the Thesis:

Chapter 2: Theory

Chapter 3: FORTRAN Code

Chapter 4: Results and Discussion

Chapter 5: Conclusion

Chapter 2

Theory

2.1 Time Independent Schrodinger Equation:-

$$\frac{-\hbar^2}{2m}\frac{d^2Y(x)}{dx^2} + V(x)Y(x) = EY(x) \quad (1)$$

Kinetic energy operator (K.E) = $\frac{-\hbar^2}{2m}\frac{d^2}{dx^2}$

Potential energy operator (P.E) = $V(x)$

It can rewrite as:-

$$\left(\frac{-\hbar^2}{2m}\frac{d^2}{dx^2} + V(x)\right) Y(x) = EY(x) \quad (2)$$

$$(K.E + P.E)\, Y(x) = E\, Y(x) \quad (3)$$

Where, $Y(x)$ is the wave function

Or,

$$HY(x) = EY(x) \quad (4)$$

Here, H is the Hamiltonian operator which is equal to the sum of kinetic energy and potential energy. E is the eigenvalue and represents total energy of the system.

2.1.1Schrodinger Equation for Particle in box:-

In case of free particle [16] $V(x) = 0$ because free particle means that, the particle experience no potential energy.

For free particle in 1-D box, equation (1) can write as

$$\frac{d^2Y(x)}{dx^2} + \frac{2mE}{\hbar^2} Y(x) = 0 \quad (5)$$

(where x is vary from 0 to 1 ($0 \leq x \leq 1$)

The particle is restricted to the region $0 \leq x \leq 1$. The probability of finding the particle outside this reason is zero. It means $Y(x) = 0$ outside the region. To fulfill the conditions that the wave function should be continuous and restricted within the given region, the elementary point is that it should follow given conditions.:-

$$Y(0) = Y(1) = 0$$

Above equation is easily solvable and the general solution is –

$$Y(x) = A \cos kx + B \sin kx \qquad (6)$$

With,

$$k^2 = \frac{2mE}{\hbar^2}$$

On applying boundary conditions and solving, we find that

$$E_n = \frac{h^2 n^2}{8ma^2} \text{ J} \qquad \text{where n = 1, 2, 3 ……}$$

2.1.2 Schrodinger Equation for Harmonic Oscillator:-

From equation (1)-

$$\frac{-\hbar^2}{2m}\frac{d^2Y}{dx^2} + V(x)Y(x) = EY(x)$$

Here, $V(x) = \frac{1}{2}kx^2$

Above equation can rewrite as -

$$\frac{d^2Y}{dx^2} = -\frac{2m}{\hbar^2}(E - \frac{1}{2}kx^2)\, Y(x) = 0 \qquad (7)$$

For adimensional results we are introducing adimensional variables x and e.

Where,

$$x = \left(\sqrt[4]{\frac{mk}{\hbar^2}}\right) * x$$

And $e = \frac{E}{\hbar w} \qquad (w = \sqrt{\frac{k}{m}})$

k, denotes the force constant and *w* frequency of classical oscillator.

equation (2) can rewrite by using adimensional variables a and b and the formula is given below:-

$$\frac{d^2Y}{dx^2} - 2(e - \frac{x^2}{2})\, Y(x) = 0 \qquad (8)$$

On solving above we get the eigenvalues for harmonic oscillator:-

$$e = (n + \frac{1}{2}) \qquad n = 0, 1, 2……..$$

Here, n is the quantum number and the value of n is vary from 0 to ∞ . There is no unit of energy.

2.1 Necessity of Numerov's method for integrating the one-dimensional Schrodinger equation:-

Equation for 1-D box with potential is shown in equation (1), and it can rewrite as given below-

$$\frac{d^2Y}{dx^2} + k^2Y(x) = 0 \qquad (9)$$

Where,

$$k^2(x) = \frac{2m}{\hbar^2}[E - V(x)]$$

On analyzing the Schrodinger equation for particle in box with some potential it was found that, we are not able to solve the second order differential equation. There is no analytical or straightforward method to solve so, here we are using numerical method. Firstly here we are explaining about numerov's method and then gives the description and final equation which is used to calculate wave function and probability density.

2.1.1 What is numerov's method?

A numerical method is a method which is used to solve the second-order ordinary differential equation in which the first-order term does not appear. Numerov's method [17] is a suitable algorithm to determine this type of problem because numerov method is simpler and one order higher (fifth) than RK4.

From equation (9)

$$\frac{d^2Y(x)}{dx^2} + k^2(x)\,Y(x) = 0$$

The above equation is linear in Y, and there is no term involving the first derivative. So, Numerov's method is a suitable algorithm for this type of problem. So, we have used this method to calculate wave functions and densities.

2.1.2 Description of Numerov's method:-

To describe the Numerov's method [18], firstly, we will write the Talor series for Ψ(x+h).

So,

$$Y(x+h) = Y(x) + h\,Y'(x) + \frac{h^2}{2}Y''(x) + \frac{h^3}{6}Y'''(x) + \frac{h^4}{24}Y''''(x) + \ldots\ldots\ldots \qquad (10)$$

On adding Y(x+h) and Y(x-h) all the h of odd powers will be terminate

$$Y(x+h) + Y(x-h) = 2Y(x) + h\,Y''(x) + \frac{h^4}{24}\,Y''''(x) + O(h^6) \quad (11)$$

Subsequently, we can write second order Schrodinger equation as given below-

$$Y''(x) = \frac{(Y(x+h)+Y(x-h)-2Y(x))}{h^2} - \frac{h^2}{12}Y''''(x) - O(h^4) \quad (12)$$

We want to estimate the term including the 4th derivative so, for this; we will work on Eq. (1) with $1 + \frac{h^2}{12}\frac{d^2Y}{dx^2}$ which gives

$$Y''(x)+\frac{h^2}{12}Y''''(x)+k^2(x)Y(x) + \frac{h^2}{12}\frac{d^2Y}{dx^2}[k^2\,(x)\,Y(x)] = 0 \quad (13)$$

Here,

$$k^2 = \frac{[(E-V(x)2m]}{\hbar^2}$$

Substituting for $Y''(x) + \frac{h^2}{12}Y''''(x)$ from equation 13 in to 12

$$Y(x+h)+Y(x-h)-2Y(x)+h^2k^2(x)Y(x) + \frac{h^4}{12}\frac{d^2Y}{dx^2}[k^2(x)Y(x)] + O(h^6) = 0 \quad (14)$$

So, now we evaluate $\frac{d^2}{dx^2}[k^2\,(x)\,Y(x)]$ by using elementry difference formula

$$\frac{d^2}{dx^2}[k^2(x)Y(x)]\sim \frac{(k^2(x+h)Y(x+h)+k^2(x-h)Y(x-h)-2k^2(x)Y(x))}{h^2} \quad (15)$$

Now equation (12) is substituting in equation (11) and rearranging, after that on assuming $x_\circ = x_n = x_\circ + nh$ and defining $k_n = k\,(x_n)$ we get

Final equation:-

$$Y_{n+1} = \frac{2\left(1-\frac{5}{12}h^2k_n^2\right)Y_n-(1+\frac{1}{12}h^2k_{n-1}^2)Y_{n-1}}{1+\frac{1}{12}h^2k_{n+1}^2} \quad (16)$$

Equation (13) which is given above is used to determine Y_n for n= 2, 3, 4…But there is a condition two initial values Y_0, and Y_1should be given.

2.2 Working of Artificial neural network-

An artificial neural network consists of a network of artificial neurons, or we can say that artificial neural networks are parallel computing devices, which is basically an attempt to make a computer model of the brain. A simple example of a feed-forward neural network that consists of three layers of artificial neurons is given below [6]. Each neuron is

represented by a circle. Suppose we have N inputs denoted by $x_i^{(1)}$. i= 1, 2... N. These inputs are represented by N neurons in the input layer. The input layer can be fed to the hidden layer through the relation.

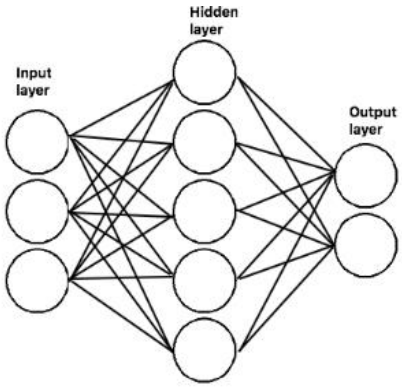

Fig1. Feed- forward three layer neural network.

$$Y_j^1 = \sum_i w_{ji}^{(1)} x_i^{(1)} + b_j^{(1)} \quad (17)$$

Here, $w_{ji}^{(1)}$ is called weight and $b_j^{(1)}$ is called bais. j=1,2,...,M is the index labeling the hidden layer and M is the number of neurons in the hidden layer. In the hidden layer, each$Y_j^{(1)}$ is transformed to the input of the next layer through the activation function

$$x_j^{(2)} = a^*(y_j^{(1)}) \quad (18)$$

It has been reported that mostly sigmoid function is used as activation function.

$$\text{Sigmoid (x)} = \sigma(x) = \frac{1}{1+e^{-x}} \quad (19)$$

After the activation function, $x_j^{(2)}$ is fed to the output layer through

$$Y_k^{(2)} = \sum_j w_{kj}^{(2)} x_j^{(2)} + b_k^{(2)} \quad (20)$$

Here, k labeling the output layer. Then $Y_k^{(2)}$ is trans-formed to the final output of the neural network through the activation function

$$Z_k = \sigma^*(Y_k^{(2)}) \quad (21)$$

The learning process can be carried out by minimizing the error function

$$E(w, b) = \frac{1}{2}\sum_{i=1}^{N_t} |Z(x_i; w, b) - Z_o(x_l)| \vee^2 \quad (22)$$

In this equation w, b are the weights and biases in the neural network. $N_{t,}$ denotes the number of elements in the training set, and Z, Z_0 is the output of the neural network and the measured value in the training set respectively. The main objective of the learning process is to find out the optimal w and b so that the error function E(w,b) become less.

Chapter 3

FORTRAN Code

3.1 Main Code

Code for 1-D system with Gaussian Potential

```
program Gaussian
implicit none
integer, parameter :: dp = selected_real_kind(14,200)
integer, parameter:: n = 3
integer i, j, kkk, icl, G
integer nodes, hnodes, ncross, kkk, n_iter
real(dp) a(n),b(n),c(n)
real(dp) xmax, dx, ddx12, norm, arg, djump, fac
real(dp) eup, elw, e, V
real(dp), allocatable :: x(:), y(:), p(:), V1(:), f(:)
character (len=80) :: fileout
integer, parameter :: k = 1 !k = m/hbar^2
c
read(15,*)G
read(15,*)
read(15,*)xmax
c
do j = 1, n
read(15,*) a(j),b(j),c(j)
end do
c
dx = xmax/G
ddx12=dx*dx/12.0_dp
c
allocate (x(0:G),V1(0:G),p(0:G),Y(0:G),F(0:G))
c
do i = 0,G
V1(i)=0.0d0
x(i) = i * dx
do j = 1,n
V = -a(j)*exp(-(x(i)-b(j))**2/(2.0*c(j)**2))
V1(i) = V1(i) + V
```

```
end do
write(3,*) x(i),V1(i)
end do

write(*,*)"('output file name > ')"
read (*,*) fileout
open (2,file=fileout, status='unknown', form='formatted')
```

Initially, like all other FORTRAN codes, we have here the declaration of variables. Here, icl denotes the classical inversion point, and ncross indicates the number of times solution changes sign. Array controlling the value of the potential at different values of x. Code first asks the value of xmax and integrates from –xmax to +xmax, but in reality, we do only from 0 to xmax and then take according to whether we want an even number of nodes or an odd number of nodes. G is the number of grid points we want to do the integration on, and then using that, we determined grid size. The division of xmax by G gives the difference between the two consecutive values of x. After that, we will allocate the array, set the potential, and also give the name of the output file to store the data.

```
search_loop: do
c
write(*,"('nodes (type -1 to stop) > ')")
read (*,*) nodes
if (nodes < 0) then
close(2)
deallocate ( f, V1 , p, y, x )
stop
end if
c
c  INITIALLY SET LOWER AND UPPER BOUNDS TO THE EIGENVALUE
eup=maxval (V1(:))
elw=minval (V1(:))
c
c  SET TRIAL ENERGY
c
write(*,"('Trial energy (0=search with bisection) > ')")
read (*,*) e
If ( e == 0.0_dp ) then
c  SEARCH EIGENVALUES WITH BISECTION METHOD (max 1000 iterations)
e = 0.5_dp * (elw + eup)
n_iter = 1000
else
c  TEST A SINGLE ENERGY VALUE (no bisection)
```

```
n_iter = 1
end if
```

Now, this is the time to set the entry point for the search of eigenvalue. So, firstly we will write the no. of nodes. If the given value of the node is less than zero, then deallocate, and the code will stop. After setting nodes, we will set eup (upper bound) and elw (lower bound) for energy. Where eup is the maximum value of potential and elw is the minimum value of potential. If we do not give any guess energy or set it zero, the code will start with the initial guess, which is the mid-point of eup and elw. It will keep on bisection until it finds the energy. If we want to evaluate results for a single value of energy, then input the energy value, and the code will give the wave function.

```
iterate: do kkk = 1, n_iter
c
f(0)=ddx12*k*(2.0_dp*(V1(0)-e))
icl=-1
do i=1,G
f(i)=ddx12*k*2.0_dp*(V1(i)-e)
if ( f(i) == 0.0_dp) f(i)=1.d-20
if ( f(i) /= sign(f(i),f(i-1)) ) icl=i
end do

if (icl >= G-2) then
deallocate ( f, V1, p, y, x )
print *, 'Error: last change of sign too far'
stop 1
else if (icl < 1) then
deallocate ( f, V1,p, y, x )
print *, 'Error: no classical turning point' stop 1
end if
c
c f(x) AS REQUIRED BY THE NUMEROV ALGORITHM
c
f = 1.0_dp – f
y = 0.0_dp
c
c BEWARE THE INTEGER DIVISION: 1/2 = 0!
c  hnodes is thus the number of nodes in the x>0 semi-axis (x=0 excepted)
c
if (2*hnodes == nodes) then
c even number of nodes: wavefunction is even
y(0) = 1.0_dp
```

```
c  assume f(-1) = f(1)
y(1) = 0.5_dp*(12.0_dp-10.0_dp*f(0))*y(0)/f(1)
else
c  odd number of nodes: wavefunction is odd
y(0) = 0.0_dp
y(1) = dx
end if
```

In this part of the program, we will set up the f- function used by the Numerov algorithm and determine the position of its last crossing i.e. change of sign. Change of sign is observed when potential energy becomes equal to the total energy. If f < 0, it means potential energy is less than total energy. This condition is known as classically allowed region, and here kinetic energy will be positive. But when f > 0, it means potential energy is greater than total energy. It is known as classically forbidden region; here, kinetic energy will be negative, so according to classical mechanics, particles cannot present in the forbidden region. Now we will integrate over the whole grid to calculate f (i). If the potential energy is greater than total energy, then f (i) will be positive either f(i) will be negative. The product of these two changes the sign and the point at which change in sign is taking place is called the classical point of inversion. It determines the value of i, and also, we will check how far it is from G. If the condition shown in the code is not satisfied the move to the Numerov algorithm. Firstly, we will check for even and then for odd no. of nodes.

OUTWARD INTEGRATION AND COUNT NUMBER OF CROSSING

```
ncross=0
do i =1,icl-1
y(i+1)=((12.0_dp-10.0_dp*f(i))*y(i)-f(i-1)*y(i-1))/f(i+1)
if ( y(i) /= sign(y(i),y(i+1)) ) ncross=ncross+1
end do
fac = y(icl)
if (2*hnodes == nodes) then
c  even number of nodes: no node in x=0
ncross = 2*ncross
else
c  odd number of nodes: node in x=0
ncross = 2*ncross +1
end if
c   check number of crossings
if ( n_iter > 1 ) then
if (ncross /= nodes) then
```

```
c  Incorrect number of crossings: adjust energy
if(kkk ==1) print'("Bisection    Energy    Nodes    Discontinuity")'
print '(i5,f25.15,i5)', kkk, e, ncross
if (ncross > nodes) then
c  CURRENT ENERGY IS TOO HIGH, LOWER THE UPPER BOUND
eup = e
else
c CURRENT ENERGY IS TOO LOWER, RAISE THE LOWER BOUND
elw = e
end if
c  NEW TRIAL VALUE :
e = 0.5_dp * (eup+elw)
c  GO TO BEGINNING OF DO LOOP, DON'T PERFORM INWARD INTEGRATION
cycle
end if
else
print *, e, ncross, nodes
end if
c
c  IF CORRECT NUMBER OF CROSSINGS: PROCEED TO INWARD
INTEGRATION
c  assuming y(G+1) = 0

y(G) = dx
y(G-1) = (12.0_dp-10.0_dp*f(G))*y(G)/f(G-1)
norm = 1.0d100
do i = G-1,icl+1,-1
y(i-1)=((12.0_dp-10.0_dp*f(i))*y(i)-f(i+1)*y(i+1))/f(i-1)
c THE FOLLOWING LINES PREVENT OVERFLOWS IF STARTING FROM TOO
c FAR
if ( abs(y(i-1)) > norm ) then
y(i-1:G) = y(i-1:G) / norm
end if
end do
```

After setting initial conditions in the above part of program, we started outward integration and examined the no. of crossings. If the number of crossings is not correct, then we will adjust the energy and not performed inward integration. If ncross is greater than nodes, the current is too high than eup, replace with guess energy e (eup = e) and bracketing the region go to the lower half. If ncross is less than nodes, elw replace with guess energy e (elw = e) and go to the upper half.

```
RESCALE FUNCTION TO MATCH AT THE CLASSICAL TURNING POINT (icl)
fac = fac/y(icl)
y(icl:) = y(icl:)*fac
c  NORMALIZE ON THE [-xmax, xmax] SEGMENT
c   the x=0 point must be counted once
norm = (2.0_dp*dot_product (y, y) - y(0)*y(0))*dx
y = y / sqrt(norm)
if ( n_iter > 1 ) then
c CALCULATE THE DISCONTINUITY IN THE FIRST DERIVATIVE
c
c  y'(i;RIGHT) - y'(i;LEFT)
djump = (y(icl+1)+y(icl-1)-(14.0_dp-12.0_dp*f(icl))*y(icl))/dx
print '(i5,f25.15,i5,f14.8)', kkk, e, nodes, djump
if (djump*y(icl) > 0.0_dp) then
c Energy is too high --> choose lower energy range
eup = e
else
c  Energy is too low --> choose upper energy range
elw = e
end if
e = 0.5_dp * (eup+elw)
c  ---- convergence test
if ( eup-elw < 1.d-10) exit iterate
end if
end do iterate
```

We rescaled the function to match at the classical inversion point and also normalized the wave function. If iteration is greater than one, we calculate the discontinuity (djump) in the first derivative. If the djump is equal to zero, then we get the correct energy. Otherwise, we will adjust energy by setting new trial energy, and this process proceeds till the difference between eup, and elw is less than or equal to 10^{-10} (eup-elw < 1.d-10).

```
CALCULATION OF THE CLASSICAL PROBABILITY DENSITY FOR ENERGY e:
c
norm = 0.0_dp
p(icl:) = 0.0_dp
do i=0,icl
arg = (e - V1(i)) if ( arg > 0.0_dp) then
p(i) = 1.0_dp/sqrt(arg)
else
```

```
p(i) = 0.0_dp
end if
norm = norm + 2.0_dp*dx*p(i)
end do
c  THE POINT AT (x=0 ) MUST BE COUNTED ONCE:
norm = norm - dx*p(0)
c  Normalize p(x) so that Int p(x)dx = 1
p(:icl-1) = p(:icl-1)/norm
c  lines starting with # ignored by gnuplot
write (2,'(" #  x   y(x)  y(x)^2    classical p(x)     V")')
c   x <0 region:
do i = G,1,-1
c   if the exponent is > 99, the format X.Y-100 is misinterpreted by gnuplot
if ( abs(y(i)) < 1.0D-50 ) y(i) = 0.0_dp
write (2,*)-x(i), (-1)**nodes*y(i), y(i)*y(i), p(i), V1(i)
end do
c   x>0 region:
c do i= 0,G
write (2,*) x(i), y(i), y(i)*y(i), p(i), V1(i)
end do
c two blank lines separating blocks of data, useful for gnuplot plotting
write (2,'(/)')
end do search_loop
end program Gaussian
```

After that, we will calculate classical probability density for energy e and write the data in output file for both positive and negative values of x. The code is the same for both the systems harmonic potential well and Gaussian potential well, and there is the only change of potential. After changing the potential in the main code, no other change will require for harmonic potential well.

3.2 Code for Harmonic Potential

```
Program harmonic

implicit none
integer, parameter :: dp = selected_real_kind(14,200)
integer :: G, i, icl
integer :: nodes, hnodes, ncross, kkk, n_iter
real(dp) :: xmax, dx, ddx12, norm, arg, djump, fac
real(dp) :: eup, elw, e
real(dp), allocatable :: x(:), y(:), p(:), vpot(:), f(:)
```

```
character (len=80) :: fileout
c  Adimensional units: x = (m*K/hbar^2)^(1/4)*X
c   e = E/(hbar*omega)
c
Write(*,*)"('Max value for x (typical value: 10)')"
read (*,*) xmax
write(*,*)"('Number of grid points (typically=100)')"
read (*,*) G
allocate(x(0:N), y(0:N), p(0:N), vpot(0:N),f(0:N) )
c
dx = xmax/G
ddx12= dx*dx/12.0_dp
c
do i = 0,G
x(i) = float(i) * dx
vpot(i) = 0.5_dp * x(i)*x(i)
end do
```

After that, the code will remain same for harmonic potential well as code for Gaussian potential well. We have to replace only the V1 of the main code with vpot for harmonic potential well. Here, V1 represents the potential for Gaussian potential well and vpot potential for harmonic potential well.

Chapter 4

Result and Discussion

Here, we have solved the Schrodinger equation for a 1- dimensional system in a Gaussian potential well and in a harmonic potential well by using the numerov method. We calculated wave functions, probability densities, and energies for systems with single electron, 2 electrons, 3 electrons, and 4 electrons. Firstly we solved probability density for one set of potential energy parameters. But as we want to map probability densities with energies using artificial neural networks. So, for this, we made a dataset of about 5000 probability densities using the Numerov method and trained these probability densities to the known energies obtained. The calculated densities will be input for artificial neural networks, and output will be the total energy of the system. Also, we evaluated fitting probability densities with analytical equations and collected coefficients. Initially, we collected coefficients for one density and repeated this process for many random densities to compare the results which is examined by using Numerov method.

We will discuss the dataset for ground state and first excited state for both systems. Our results showed the data for wave function, probability density, and energy before filling of electrons and after filling of electrons. Here, we take four types of systems single electron, two electrons, three electrons, and four electrons. And fill electrons according to the Pauli Exclusion Principle to calculate energy. We found a good correlation between original and fitted probability density shown in plots 10, 11, 12, and13. Table (1) having information about state and energy for harmonic potential well, and table (2) having information about state and energy for Gaussian potential well. Table 3 and 4 having energy data after filling electrons in the system. Table 5 to 8 consists of fitted coefficients for a single, two, three, and four electrons system. Fig.1 shows feed-forward three-layer artificial neural networks. Fig.2 to fig .9 shows the plots for wave function and probability density for ground and first excited state.

State	Energy
Ground State	0.50
First Excited State	1.50

Table 1. Represents state and energy for the harmonic potential well.

State	Energy
Ground State	-2.05
First Excited State	-0.19

Table 2. Represents state and energy for Gaussian potential well.

Ground State wave function and Probability density

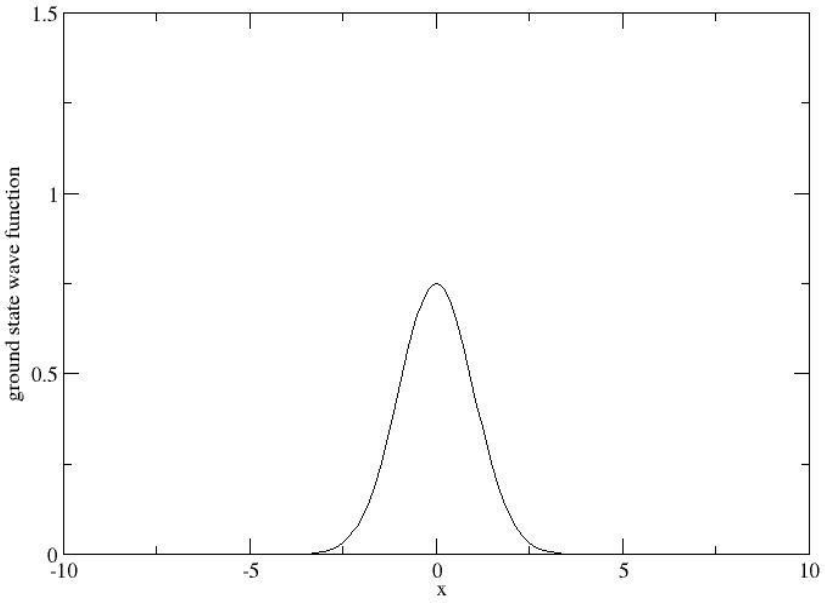

Fig 2. Wave function for harmonic Potential well.

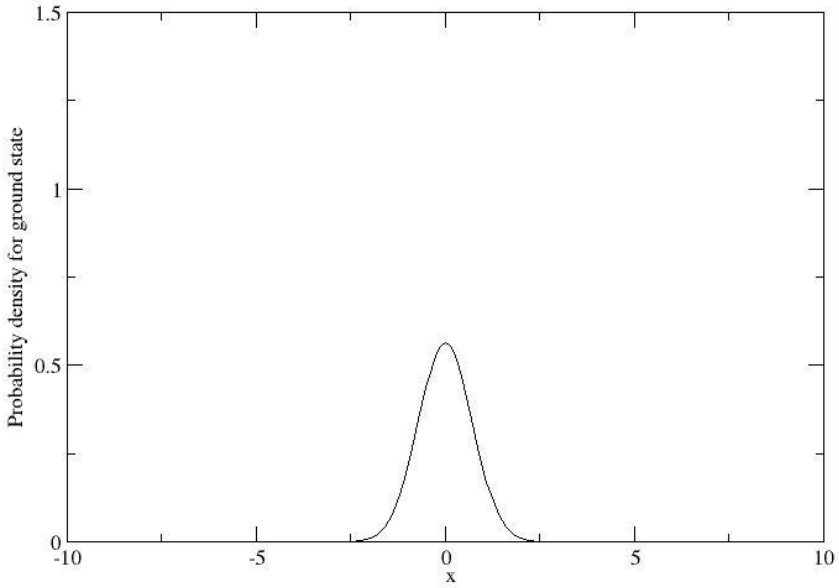

Fig3. Probability density for harmonic potential well.

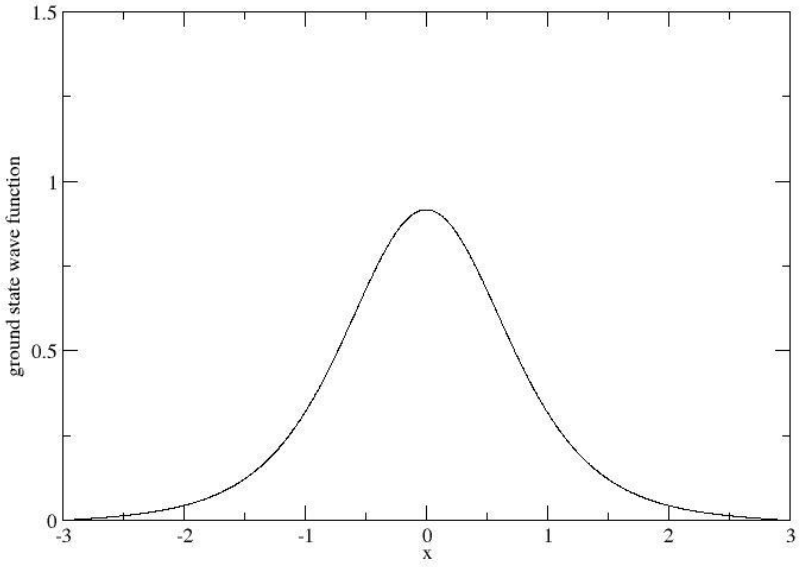

Fig4. Wave function for Gaussian potential well.

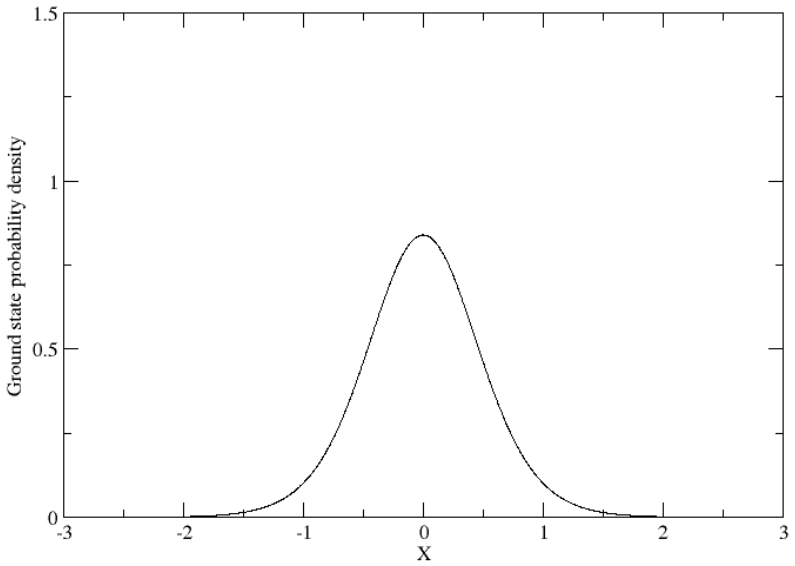

Fig5. Probability density for gaussian Potential well.

First Excited State wave function and probability density

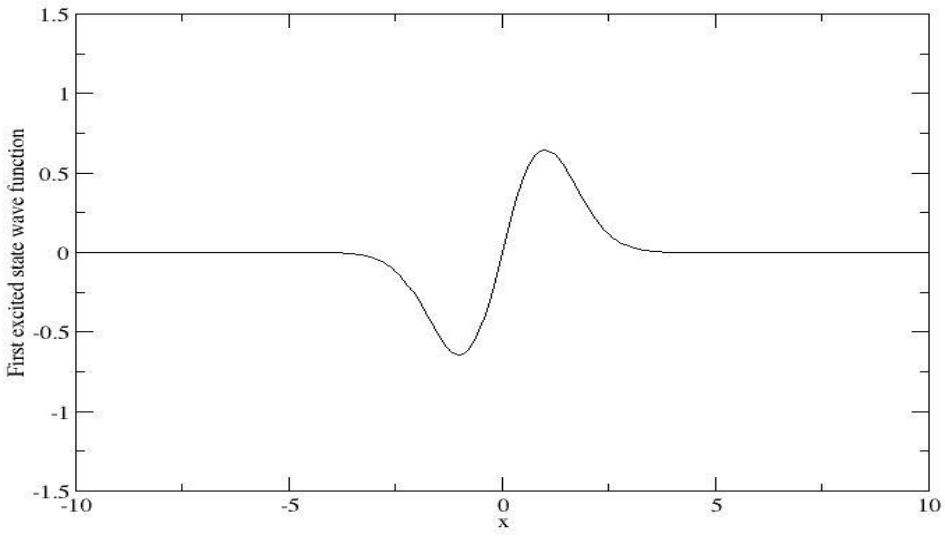

Fig6. First excited state wave function for harmonic potential well.

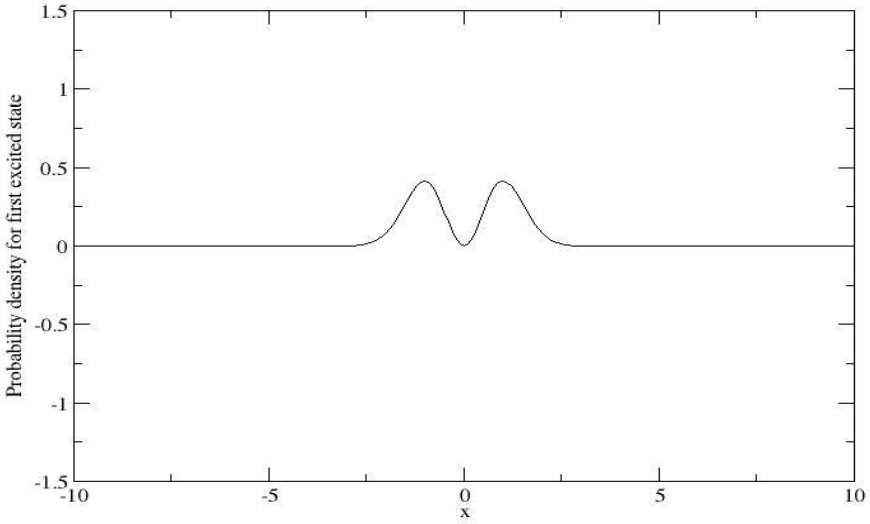

Fig7. First excited state probability density for harmonic potential well.

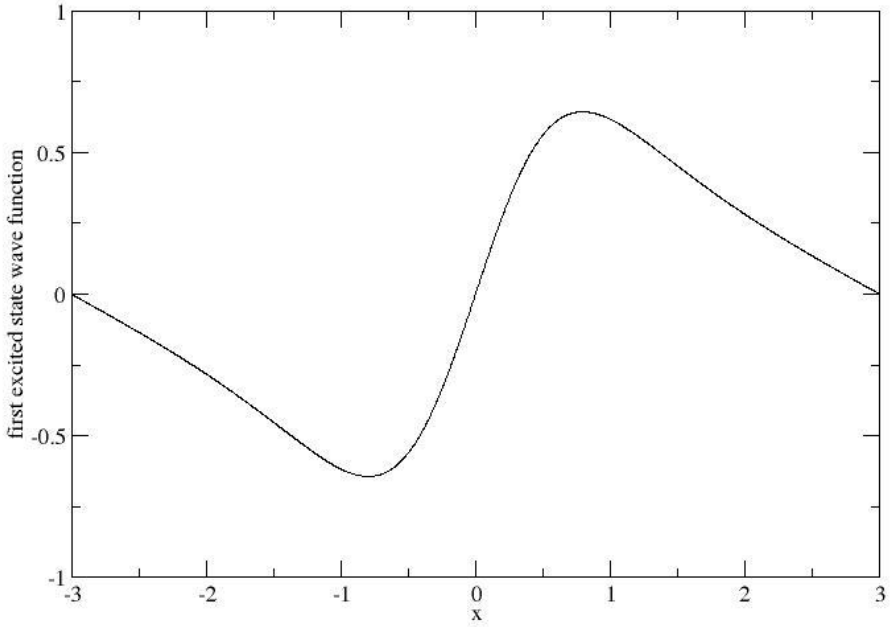

Fig8. First excited wave function for gaussian potential well.

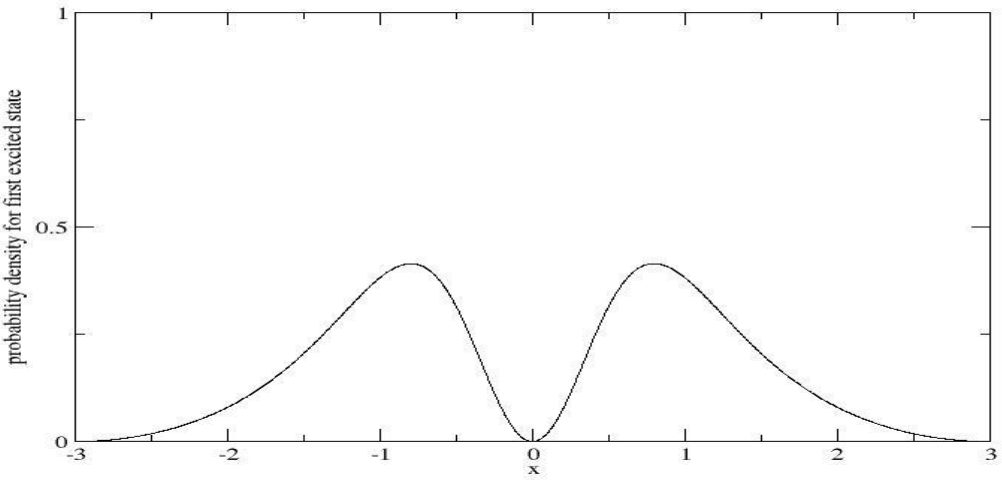

Fig9. First excited state probability density for Gaussian potential well.

System Energies On filling up electrons

Energy = no. of electrons present in particular state *Energy of the state

For Harmonic potential well:-

Energy for ground state = 0.50

Energy for first excited state = 1.50

	System	Energy
1.	One electron system electron present in ground state.	1*(0.50) = 0.50
2.	Two electron system both electrons present in ground state.	2*(0.50) = 1.00
3.	Two electrons system one electron present in ground one in first excited state.	1*(0.50)+1*(1.50) = 2.00
4.	Three electrons system two electrons present in ground and one in first excited state.	2*(0.50)+1*(1.50) = 2.50
5.	Four electrons system two electrons present in ground and two in first excited state.	2*(0.50)+2*(1.50) = 4.00

Table 3. Electronic system and energy for harmonic potential well.

For Gaussian potential well:-

Energy for ground state = -2.05

Energy for first excited state = -0.19

	System	Energy
1.	One electron system electron present in ground state.	1*(-2.05) = -2.05
2.	Two electrons system both electrons presents in first excited state	2*(-2.05) = -4.10
3.	Two electrons system one in ground state one in first excited state	1*(-2.05) +1*(-0.19) = - 2.24
4.	Three electrons system two electrons present in ground one in first excited state	2*(-2.05) +1*(-0.19)= - 4.30
5.	Four electrons system two electrons present in ground two in first excited state	2*(-2.05) + 2*(-0.19) = - 4.47

Table 4. Electronic system and energy for gaussian potential well.

Fitting Equation and plots for all the systems

Here, we have established the general form of total probability density to use the coefficients as an input for neural networks. We need to ensure that all the fitted coefficients are relatively independent .Thus we choose hermite polynomial. Since, we know Orthogonality is the important condition for independency. So, form harmonic oscillator if two hermite polynomials are orthogonal to each other this means they are independent.

The fitting equation for all the system considered in this study is of the form

$$n(x) = [a_0 * H_0(x) + a_1 * H_1(x) + a_2 H_2(x) + a_3 * H_3(x) + a_4 * H_4(x) + a_5 * H_5(x)\ ]\ e^{-(1.5*x*x)/2}$$

Where n(x) is total probability density of all electrons in the system
$a_0, a_1, a_2, a_3, a_4, a_5$ are fitted coefficients.
$H_0(x)$, H1(x), H2(x), $H_3(x)$, $H_4(x)$, $H_5(x)$ are hermite polynomials up to order 5.

Values of $a_0, a_1, a_2, a_3, a_4, a_5$ for 1 electron zero node system are

Fitted coefficient	Value of fitted coefficient
a_0	0.75
a_1	0.36
a_2	0.08
a_3	0.00
a_4	0.0008
a_5	0.0003

Table 5. Fitted coefficients for 1electron zero node system.

Total probability density equation for 1electron zero node system is -

$$n(x)= [0.75 * H_0(x) + 0.36 * H_1(x) + 0.08 * H_2(x) + 0.00 * H_3(x) + 0.0008 * H_4(x) + 0.0003 * H_5(x)]e^{\frac{-(1.5*x*x)}{2}}$$

Plot for 1 electron and zero node system

Here, fig. 10 shows the comparison between original probability density and fitted probability density for 1 electron and 0 node systems.

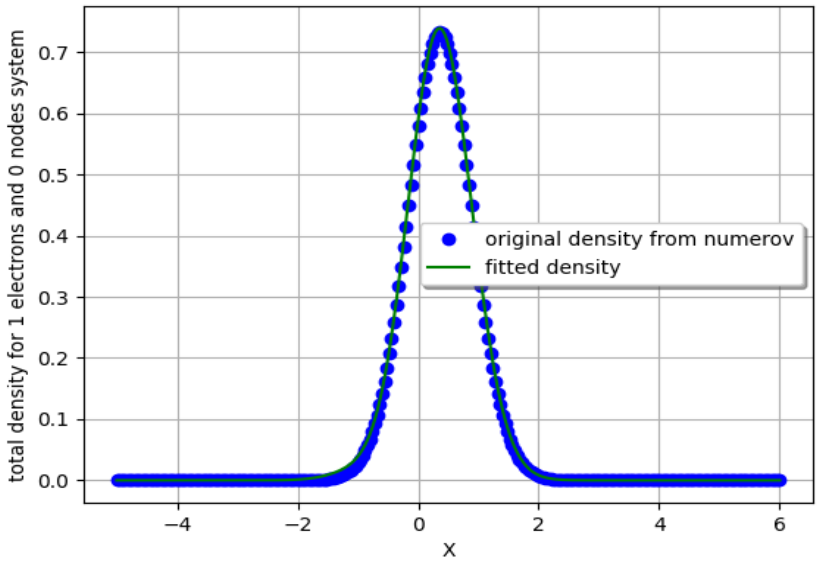

Fig10. Probability density graph for 1 electron and 0 nodes system.

Values of $a_0, a_1, a_2, a_3, a_4, a_5$ for 2electrons 1node system are

Fitted coefficient	Value of fitted coefficient
a_0	1.87
a_1	1.24
a_2	0.84
a_3	0.26
a_4	0.06
a_5	0.003

Table6. Fitted coefficient for 2electrons 1 node system.

Total probability density equation for 2electrons and one node system is -

$$n(x)= [1.87 * H_0(x) + 1.24 * H_1(x) + 0.84 * H_2(x) + 0.26 * H_3(x) + 0.06 * H_4(x) + 0.003 * H_5(x)]e^{\frac{-(1.5*x*x)}{2}}$$

Plot for 2 electrons and 1 node system

This graph represents the comparison between original probability density and fitted probability density for 2 electrons and 1 node system.

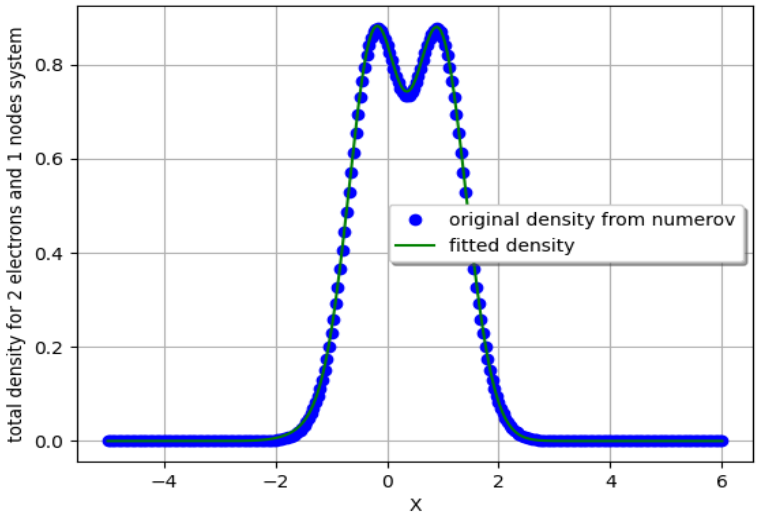

Fig11. Probability density graph for 2 electrons and 1 node system.

Values of a_0, a_1, a_2, a_3, a_4, a_5 for 3electrons 1node system are

Fitted coefficient	Value of fitted coefficient
a_0	2.62
a_1	1.60
a_2	0.93
a_3	0.26
a_4	0.06
a_5	0.003

Table7. Fitted coefficient for 3 electrons 1 node system

Total probability density equation for 3 electrons and 1 node system is -

$$n(x)= [2.62 * H_0(x) + 1.60 * H_1(x) + 0.93 * H_2(x) + 0.26 * H_3(x) + 0.06 * H_4(x) + 0.003 * H_5(x)]e^{\frac{-(1.5*x*x)}{2}}$$

Plot for three electrons and one-node system

This graph represents the comparison between original probability density and fitted probability density for 3 electrons and 1 node system.

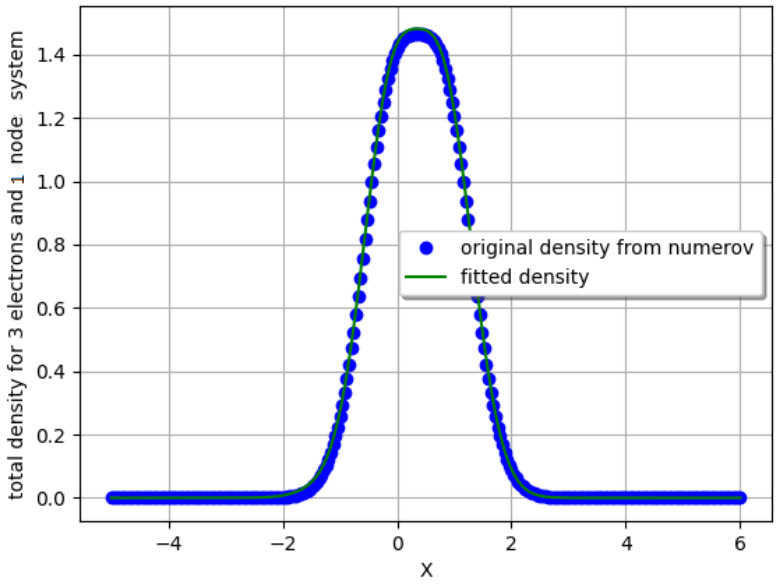

Fig.12 Probability density graph for 3 electrons and 1 node system.

Values of a_0, a_1, a_2, a_3, a_4, a_5 for 4 electrons 1 node system are-

Fitted coefficient	Value of fitted coefficient
a_0	3.74
a_1	2.65
a_2	1.71
a_3	0.50
a_4	0.12
a_5	0.00

Table8. Fitted coefficient for 4 electrons 1 node system.

Total probability density equation for 4 electrons and 1 node system is -

$$n(x)= [3.74 * H_0(x) + 2.65 * H_1(x) 1.71 * H_2(x) + 0.50 * H_3(x) + 0.12 * H_4(x) + 0.00 * H_5(x)] e^{\frac{-(1.5*x*x)}{2}}$$

Plot for 4 electrons and 1nodes system

This graph represents the comparison between original probability density and fitted probability density for four electrons and one node system.

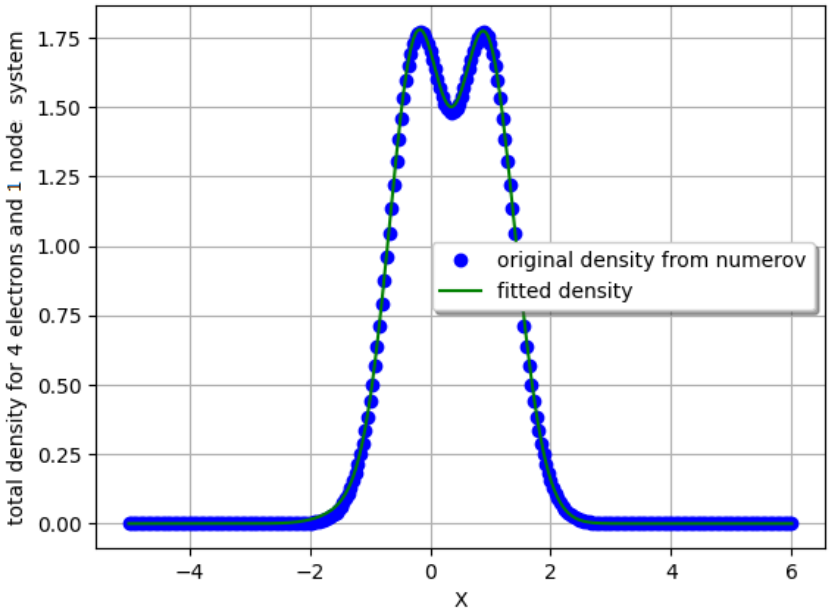

Fig.13 Probability density graph for 4 electrons and 1 node system.

Chapter 5

Conclusion

In this project, we evaluated Schrodinger Equation for one-dimensional system in a Gaussian potential well and Harmonic potential well by using the Numerov method. Since one -dimensional Schrodinger equation in Gaussian potential well has no analytical solution. We want to map probability densities to energies using artificial neural networks. So, we made a dataset of about 5000 probability densities using the numerov method for the training of neural networks. The dataset is obtained by randomly changing the parameters of Gaussian potential well and force constant values in the harmonic potential well. We examine results for total probability density calculated by the numerov method and fitting probability density calculated by using an analytical equation. After examining plots obtained using both methods, we found good resemblance. So, from the future prospective, such models can be used to calculate energies of 1- dimensional Schrodinger equation in a similar but for unknown potential energy well without really solving the Schrodinger equation analytically.

REFERENCES

1. Born, M.; Oppenheimer, R.; *Ann. Phys.*, 1927, 389, 457.
2. Car, R.; Parrinello, M.; *Phys. Rev. Lett.*, 1985, 55, 2471.
3. Behler, J.; "Constructing high dimensional neural network potentials: A tutorial review"; *International journal of Quantum Chemistry*, 2015, DOI: 10.1002/qua.24890
4. Marx, D.; Hutter, J.; "Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods"; Cambridge University Press: Cambridge, 2009.
5. Parr, R. G.; Yang, W.; "*Density Functional Theory of Atoms and Molecules"*; Oxford University Press: Oxford, 1989.
6. Koch, W.; Holthausen, M. C.; "A Chemist's Guide to Density Functional Theory "; 2nd ed.; Wiley-VCH, 2001.
7. McCulloch, W. S.; Pitts, W.; "The bulletin of mathematical biophysics"; 1943, 5,115.
8. Bishop, C. M.; "Pattern recognition and machine learning"; 1st ed., Springer-Verlag, New York, 2006.
9. Schwenker, F.; Kestler, H. A.; Palm, G.; "Neural Net-works"; 2001, 14, 439.
10. Scarselli, F.; Tsoi, A. C.; "Neural Networks"; 1998, 11, 15.
11. Park, J.; Sandberg, I. W.; "Neural Computation"; 1991, 3, 246, http://dx.doi.org/10.1162/neco.1991.3.2.246.
12. Teng, P.; "Machine- learning quantum mechanics: Solving quantum mechanics problems using radial basis function networks"; *Phys. Rev. E98,* 2018.
13. Lagaris, I.; Likas, A.; Fotiadis, D.; "Computer Physics Communications"; 1997, 104, 1.
14. Da Silva, A. J.; Ludermir, T. B.; De Oliveira, W.R.; "Neural Networks"; 2016, 76, 55.
15. Behler, J.; "Condensed Matter"; J. Phys., 2014, 26, 183001.
16. Synder, J. C.; Rupp, M.; Hansen, K.; Muller, K. R.; Burke, K.;"Finding Density Functionals with Machine Learning"; *Phys. Rev. Lett*, 2012, 108, 253002.
17. Dreizler, R. M.; Gross, E. K. U.; "Density Functional Theory: An Approach to the Quantum Many-Body Problem"; Springer, Berlin, 1990.
18. McQuarrie, D. A.; Simon, J. D.; "Physical Chemistry: a molecular approach";*J. Chem. Educ.*, 1998, 75, 5, 545 .
19. Peter, Y.;"Numerov method for integrating the one dimensional Schrodinger equation"; *Phys. 115/242*, 2009.
20. Hairer, E.; Nørsett, P.; Syvert Paul, P.; Wanner, G.; "Solving Ordinary Differential Equations I: Nonstiff Problems", Springer, New York, 1993.