Assignment 5 - Harmonic Oscillator - I

SGTB Khalsa College, University of Delhi

Preetpal Singh(2020PHY1140)(20068567043)

Unique Paper Code: 32221501

Paper Title: Quantum Mechanics and Applications

Submitted on: August 24, 2022

B.Sc(H) Physics Sem V

Submitted to: Dr. Mamta

Name ⇒ Preetpal Singh

Roll No ⇒ 2020PHY1140

Assignment ⇒ Harmonic Oscillator-I

Date ..../..../......

(1) Time Independent Schrodinger Equation:

$$\frac{d^2\psi}{dx^2} + \frac{2m}{\hbar^2}(E - V(x))\psi = 0$$

For harmonic oscillator, $V(x) = \frac{1}{2}kx^2 = \frac{1}{2}m\omega^2x^2$

$$\frac{d^2\psi}{dx^2} + \frac{2m}{\hbar^2}\left(E - \frac{1}{2}m\omega^2x^2\right)\psi = 0$$

To make D.E. dimensionless:

Take $x = \xi/L$ $\qquad \frac{dx}{d\xi} = \frac{1}{L}$

$xL = \xi$
dimensionless

$$\frac{d\xi}{dx} = L$$

$$\frac{d\psi}{dx} = \frac{d\psi}{d\xi}\times\frac{d\xi}{dx} = \frac{d\psi}{d\xi}\times L$$

$$\frac{d^2\psi}{dx^2} = L\frac{d}{dx}\left(\frac{d\psi}{d\xi}\right) = \frac{L\times d^2\psi}{dx\,d\xi} = L\frac{d}{d\xi}\left(\frac{d\psi}{dx}\right)$$

$$= L\times\frac{d}{d\xi}\left(\frac{d\psi}{dx}\times\frac{d\xi}{d\xi}\right) = L^2\times\frac{d}{d\xi}\left(\frac{d\psi}{d\xi}\right) = L^2\frac{d^2\psi}{d\xi^2}$$

$$L^2\frac{d^2\psi}{d\xi^2} + \frac{2m}{\hbar^2}\left[E - \frac{1}{2}m\omega^2\times\frac{\xi^2}{L^2}\right]\psi = 0$$

$$\frac{d^2\psi}{d\xi^2} + \frac{2m}{\hbar^2}\left[\frac{E}{L^2} - \frac{1\,m\omega^2\xi^2}{2L^4}\right]\psi = 0$$

$$\frac{d^2\psi}{d\xi^2} + \left[\frac{2mE}{\hbar^2L^2} - \frac{m^2\omega^2\xi^2}{L^4\hbar^2}\right]\psi = 0$$

Let's take $\frac{m^2\omega^2}{L^4\hbar^2} = 1$

$$L^4 = \frac{m^2\omega^2}{\hbar^2}$$

$$\boxed{L = \sqrt{\frac{m\omega}{\hbar}}}$$

Spiral

Date ..../..../.........

$$\frac{d^2\psi}{d\xi^2} + \left[\frac{2mE}{\hbar^2 \times \frac{m\omega}{\hbar}} - \xi^2\right]\psi = 0$$

$$\frac{d^2\psi}{d\xi^2} + \left[\frac{2E}{\hbar\omega} - \xi^2\right]\psi = 0$$

Take $\beta = \frac{2E}{\hbar\omega}$

$$\boxed{\frac{d^2\psi}{d\xi^2} + \left[\beta - \xi^2\right]\psi = 0}$$

Dimensionless Schrodinger Wave Equation for harmonic oscillator

(+) Dimensionless Schrodinger Equation for harmonic oscillator

$$\frac{d^2\psi}{d\xi^2} + (\beta - \xi^2)\psi = 0$$

There will be asymptotic solution at $\xi \to \pm\infty$

So, $\psi(\xi)$ must satisfy $\psi(\pm\infty) \to 0$

when $\xi \to \pm\infty$, β can be neglected

Now, the eqn becomes

$$\frac{d^2\psi}{d\xi^2} - \xi^2\psi = 0$$

$$\psi = e^{\pm \xi^2/2}$$

we'll neglect. $\psi = e^{\xi^2/2}$ gives infinite values when $\xi \to \infty$

$$\psi = e^{-\xi^2/2}$$

let's consider $\psi = e^{-\xi^2/2}$ is asymptotic solution. However exact solution will be of the form

Date ..../..../.........

$\psi(\xi) = H(\xi) e^{-\xi^2/2}$

$H(\xi)$ is unknown function

$H(\xi)$ will satisfy Hermite differential Equation

$$\frac{d^2H(\xi)}{d\xi^2} - 2\xi \frac{dH(\xi)}{d\xi} + (\beta-1) H(\xi) = 0$$

From Frobenius method iwell recurrence relation of the following format

$$a_{n+2} = \frac{2n+1-\beta}{(n+1)(n+2)} a_n$$

$\therefore$ Power series solution of $H(\xi)$ is given by

$$H(\xi) = a_0\left[1 + \frac{(1-\beta)\xi^2}{2!} + \frac{(1-\beta)(5-\beta)\xi^4}{4!} + \cdots\right] + a_1\left[\xi + \frac{(3-\beta)\xi^3}{3!} + \frac{(3-\beta)(7-\beta)\xi^5}{5!} + \cdots\right]$$

Comparing asymptotic behaviour of $H(\xi)$ & e^{ξ^2}

$$e^{\xi^2} = 1 + \frac{\xi^2}{1!} + \frac{\xi^4}{2!} + \frac{\xi^6}{3!} + \cdots$$

Ratio of successive terms of seriest

$$\frac{a_{n+2}}{a_n} = \frac{2n+1-\beta}{(n+1)(n+2)} \xrightarrow[n\to\infty]{} \frac{2}{n}$$

$$\frac{b_{n+2}}{b_n} = \frac{(2/n+2)!}{(2/n)!} \xrightarrow[n\to\infty]{} \frac{2}{n}$$

So, for large values of n, $H(\xi)$ behaves like e^{ξ^2}

Now, the solution becomes,

$$\psi(\xi) = H(\xi) e^{-\xi^2/2} = e^{\xi^2/2} \cdot e^{-\xi^2/2} = -e^{\xi^2/2}$$

But when $\xi \to \infty$ $\psi(\xi) \to \infty \not\to 0$

It's not acceptable solution.

It can be true only when

$$\beta = 2n+1 = \frac{2E}{\hbar\omega}$$

$$E_n = \left(n+\frac{1}{2}\right)\hbar\omega$$

When $n=0$

$$E_0 = \frac{1}{2}\hbar\omega$$

$$\frac{1}{2} m\omega^2 x^2 = \frac{1}{2}\hbar\omega$$

$$x^2 = \hbar/m\omega$$

$$x_{cl} = \sqrt{\frac{\hbar}{m\omega}}$$

This is classical turning point for ground state

For general case,

$$\frac{1}{2} m\omega^2 x^2 = \left(n+\frac{1}{2}\right)\hbar\omega = \left(\frac{2n+1}{2}\right)\hbar\omega$$

$$x_{cl} = \pm\sqrt{\left(\frac{2n+1}{2}\right)\frac{\hbar}{m\omega}}$$

Take $L^2 = \frac{\hbar}{m\omega}$

$$x_{cl} = \pm\sqrt{\left(\frac{2n+1}{2}\right)L^2}$$

$$\frac{x_{cl}}{L} = \pm\sqrt{(2n+1)}$$

Date ..../..../......

This is classical turning point in dimensionless form

Energy Eigen values:

The solⁿ of $\frac{d^2\psi}{d\varepsilon_1^2} + (\beta - \varepsilon_1^2)\psi = 0$ is

$$\psi_n(\varepsilon_1) = e^{-\varepsilon_1^2/2} H_n(\varepsilon_1)$$

The general solⁿ is

$$\psi_n(\varepsilon_1) = N_n e^{-\varepsilon_1^2/2} H_n(\varepsilon_1) \quad -(1)$$

Here N_n is Normalisation factor

Using Normalisation condition,

$$\int_{-\infty}^{\infty} \psi_n(\varepsilon_1)\, \psi_n^*(\varepsilon_1)\, dx = 1$$

$$\varepsilon_1 = xL$$

$$\varepsilon_1/L = x \qquad dx = \frac{d\varepsilon_1}{L}$$

$$\frac{|N|^2}{L} \int \psi_n(\varepsilon_1)\, \psi_n^*(\varepsilon_1)\, d\varepsilon_1 = 1$$

Using Orthogonality Conditions,

$$\int_{-\infty}^{\infty} e^{-\varepsilon_1^2} H_n(\varepsilon_1) H_m(\varepsilon_1)\, d\varepsilon_1 = \begin{cases} 0 & \text{if } n \neq m \\ \sqrt{\pi}\, 2^n n! & \text{if } n = m \end{cases}$$

$$\frac{|N|^2}{L} \times \sqrt{\pi}\, 2^n n! = 1$$

$$N_n = \left(\frac{1}{\sqrt{\pi}\, 2^n n!}\right)^{1/2}$$

Eqⁿ (1) becomes

$$\psi_n(\varepsilon_1) = \left(\frac{1}{\sqrt{\pi}\, 2^n n!}\right)^{1/2} e^{-\varepsilon_1^2/2} H_n(\varepsilon_1)$$

Spiral

(c) Expression of 1st five normalised Energy eigenfunctions

$$\psi_n(\xi_1) = \sqrt{\frac{1}{\sqrt{\pi}\, 2^n n!}}\; e^{-\xi_1^2/2} H_n(\xi_1)$$

For $n=0$

$$\psi_0(\xi_1) = \sqrt{\frac{1}{\sqrt{\pi}}}\; e^{-\xi_1^2/2} H_0(\xi_1)$$

For $n = 1, 2, 3, 4$

$$\psi_1(\xi_1) = \sqrt{\frac{1}{\sqrt{\pi}\times 2}}\; e^{-\xi_1^2/2} H_1(\xi_1)$$

$$\psi_2(\xi_1) = \sqrt{\frac{1}{\sqrt{\pi}\times 4\times 2}}\; e^{-\xi_1^2/2} H_2(\xi_1) = \sqrt{\frac{1}{8\sqrt{\pi}}}\; e^{-\xi_1^2/2} H_2(\xi_1)$$

$$\psi_3(\xi_1) = \sqrt{\frac{1}{\sqrt{\pi}\, 2^3 \times 3!}}\; e^{-\xi_1^2/2} H_3(\xi_1) = \sqrt{\frac{1}{\sqrt{\pi}\times 48}}\; e^{-\xi_1^2/2} H_3(\xi_1)$$

$$\psi_4(\xi_1) = \sqrt{\frac{1}{\sqrt{\pi}\, 2^4 \times 4!}}\; e^{-\xi_1^2/2} H_4(\xi_1) = \frac{1}{\sqrt{\pi\times 384}}\; e^{-\xi_1^2/2} H_4(\xi_1)$$

d) As we know, quantum mechanically

$$E_n = \left(n+\tfrac{1}{2}\right)\hbar\omega$$

At ground state, $n=0$

$$E_0 = \tfrac{1}{2}\hbar\omega$$

Quantum Mechanically, we're getting some energy at ground state

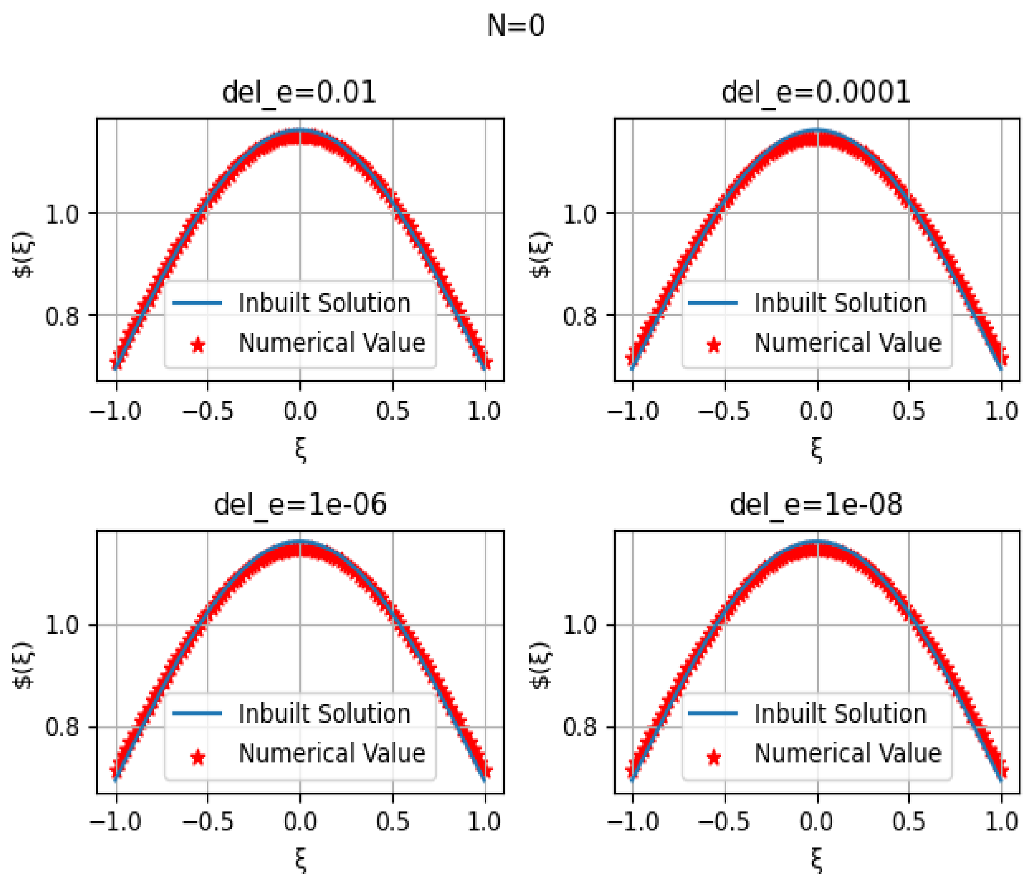
Programming

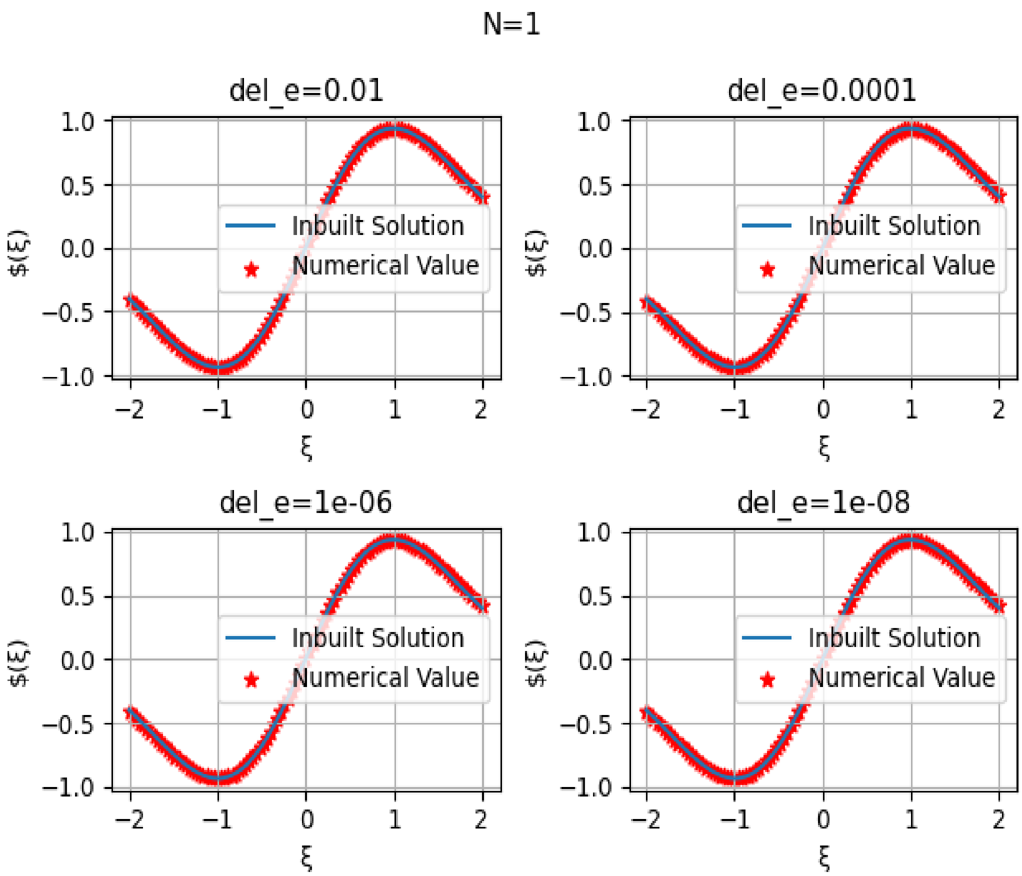
```python
import math
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
import scipy.integrate as integrate
import pandas as pd
def alpha(n,x,del_e):
    return 2*(n+(1/2)+del_e-(x**2)/2)
def func_(x,x_vec):
    ans_vec = np.zeros((2))
    ans_vec[0] = x_vec[1]
    ans_vec[1] = -2*(n+(1/2)+(1e-2)-(x**2)/2)*x_vec[0]
    return ans_vec
def sub_plot(ax,a,b,d,title,x_label,y_label):
    ax.scatter(a,b,label="Numerical Value",marker="*",color="red")
    ax.plot(a,d,label="Inbuilt Solution")
    ax.set_title(title)
    ax.set_xlabel(x_label)
    ax.set_ylabel(y_label)
    ax.legend()
    ax.grid(True)

def numerov(x_min, x_max, u_0, u_prime,n, N,del_e):
    c_i =[];u=[]
    x = np.linspace(x_min,x_max,N+1)
    Alpha = alpha(n,x,del_e)
    h = x[1]-x[0]
    u_1 = 1 + ((h**2)/math.factorial(2)) + 3*((h**4)/math.factorial(4))
    u.append(u_0);u.append(u_1)
    ddx_12 = (h**2)/12
    for i in range(0,N+1):
        c_i_ = 1 + np.multiply(ddx_12,Alpha[i])
        c_i.append(c_i_)
    for i in range(2,N+1):
        u_ = (1/c_i[i])*(((12-10*c_i[i-1])*u[i-1])-c_i[i-2]*u[i-2])
        u.append(u_)
    u_norm=u/np.sqrt(integrate.simps(np.power(u,2),x))
    sol = solve_ivp(func_, [x_min,x_max], [u_0,u_prime],dense_output=True)
    inbuilt = sol.sol(x)
    sol_ = inbuilt[0]/np.sqrt(integrate.simps(np.power(inbuilt[0],2),x))
    return x, u_norm, sol_, Alpha
'''-----------------------------------------------------N=2----------------------------------------------'''
def combine(list1, list2):
    list1_ = np.delete(list1, len(list1)-1)
    result_list = []
    result_list = list(list1_)
    for item in list2:
        result_list.append(item)
    return result_list

def parity(x_min, x_max, u_0, u_prime,n, N,del_e):
    p = numerov(x_min, x_max, u_0, u_prime,n, N,del_e)
    t = p[1][::-1]
    t_1 = p[2][::-1]
    array = [];array_1=[]
    if n%2 == 0:
        array_1 = combine(t_1, p[2])
        array = combine(t, p[1])
    elif n%2 != 0:
        array_1 = combine((np.multiply(-1,t_1)),p[2])
        array = combine((np.multiply(-1,t)),p[1])
    #print("u_norm",array)
    # plt.scatter(np.linspace(-x_max,x_max,2*N+1),array_1)
    # plt.plot(np.linspace(-x_max,x_max,2*N+1),array)
```

```python
    # plt.show()
    return array, array_1
# '''----------------------------u vs x Plotting ---------------------------------------'''
# x_min = 0; n = 0; x_max = n+1;N=50
# if n%2 ==0:
#     u_0 = 1
#     u_prime=0
# else:
#     u_0 = 0
#     u_prime = 1
# fig2, ((axx1, axx2), (axx3, axx4)) = plt.subplots(2,2)
# fig2.suptitle(f'N={n}')
# dict = {'0': axx1,'1': axx2,'2': axx3,'3': axx4}
# for i in range(2,10,2):
#     par_1, par_2 = parity(x_min, x_max, u_0, u_prime,n, N,10**-i)
#     if n ==2 or n==3:
#         sub_plot(dict[str(int((i-2)/2))],np.linspace(-x_max,x_max,2*N +1),np.multiply(-1,par_1),np.multiply(-1,par_2),f'del_e={10**-i}',"\u03BE","u(\u03BE)")
#     else:
#         sub_plot(dict[str(int((i-2)/2))],np.linspace(-x_max,x_max,2*N +1),par_1,par_2,f'del_e={10**-i}',"\u03BE","u(\u03BE)")
# plt.tight_layout()
# plt.show()

'''----------------------------Probability Density vs x Plotting ---------------------------------------'''
# x_min = 0 ;N=100;del_e=1e-6
# fig2, ((axx1, axx2), (axx3, axx4)) = plt.subplots(2,2)
# fig2.suptitle("Probability Density")
# dict = {'0': axx1,'1': axx2,'2': axx3,'3': axx4}
# for n in range(0,4,1):
#     x_max = n+1
#     if n%2 ==0:
#         u_0 = 1
#         u_prime=0
#     else:
#         u_0 = 0
#         u_prime = 1
#     par_1, par_2 = parity(x_min, x_max, u_0, u_prime,n, N,del_e)
#     if n ==2 or n==3:
#         sub_plot(dict[str(int(n))],np.linspace(-x_max,x_max,2*N +1),np.power(np.multiply(-1,par_1),2),np.power(np.multiply(-1,par_2),2),f'N={n}',"\u03BE","$u(\u03BE)^2$")
#     else:
#         sub_plot(dict[str(int(n))],np.linspace(-x_max,x_max,2*N +1),np.power(par_1,2),np.power(par_2,2),f'N={n}',"\u03BE","$u(\u03BE)^2$")
# plt.tight_layout()
# plt.show()
'''-----------------------------------------Q-C ----------------------------------------------'''
# x_min = 0; n = 0; x_max = n+1;N=10;del_e=1e-6;e = 1.6e-19; h_cut = 1.0545e-34; umega=5.5e14
# if n%2 ==0:
#     u_0 = 1
#     u_prime=0
# else:
#     u_0 = 0
#     u_prime = 1
# par_1, par_2, alpha_ = parity(x_min, x_max, u_0, u_prime,n, N,del_e)
# print("                              Energy eigenvalues in eV")
# data = {
#     "Approximated Eigen Values":np.multiply(alpha_,(h_cut*umega)/e),
#     "Analytical Eigen Values":np.multiply(alpha_-del_e,(h_cut*umega)/e)
# }
# print(pd.DataFrame(data))
```

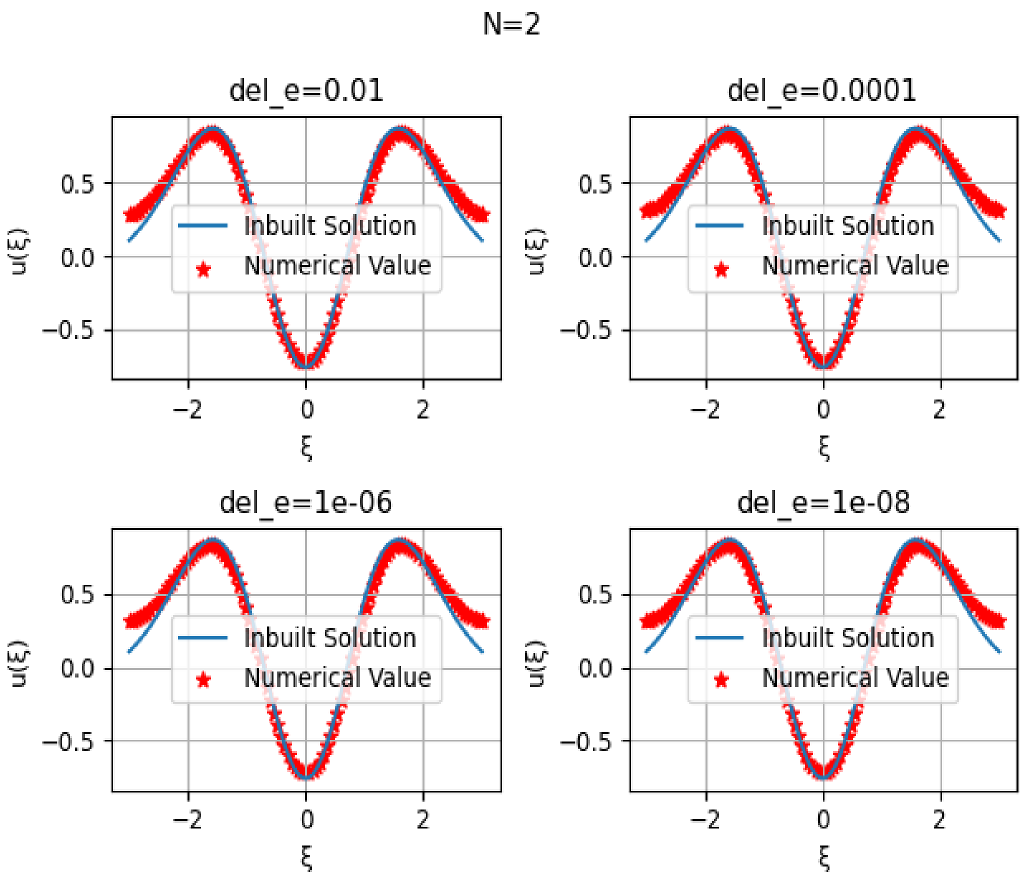
```
'''------------------------------------------------------Q_d-------------------------------------------'''
# x_min = 0; n = 0; x_max = n+4;N=100;del_e=1e-6
# if n%2 ==0:
#     u_0 = 1
#     u_prime=0
# else:
#     u_0 = 0
#     u_prime = 1
# x_test = np.linspace(x_min,x_max,N+1)
# alfa = alpha(n,x_test,del_e)
# # par_1, par_2, alpha_, x_ = parity(x_min, x_max, u_0, u_prime,n, N,del_e)
# j=0;prob_density=[]
# for i in alfa:
#     j+=1
#     if i<0:
#         print("Probability of finding electron in the classically forbidden region when it is in the ground state")
#         # print(len(np.power(par_1[j-1:],2)))
#         # print(integrate.simps(np.power(par_1[j-1:],2),x_[j-1:]))
#         print(len(alfa[j-1:]))
#         print((alfa[j-1:]))
#         p_ = numerov(x_test[j-1], x_max, u_0, u_prime,n, len(x_test[j-1:])-1, del_e)
#         print(len(x_test[j-1:]))
#         print(len(p_[1]))
#         print("probability",integrate.simps(np.power(p_[1],2),x_test[j-1:]))
#         # par_1, par_2, alpha_, x_ = parity(x_test[j-1], x_max, u_0, u_prime,n, len(x_test[j-1:]),del_e)
# #       print(np.power(par_1,2))
#         # t = p_[1][::-1]
#         # print(integrate.simps(np.power(t,2),np.linspace(-x_test[j-1],-x_max,len(x_test[j-1:])))+integrate.simps(np.power(p_[1],2),np.linspace(x_test[j-1], x_max,len(x_test[j-1:]))))

#         # print(integrate.simps(np.power(t,2),np.linspace(-x_test[j-1],-x_max,len(x_test[j-1:]))))
#         break
```

Result and Discussion

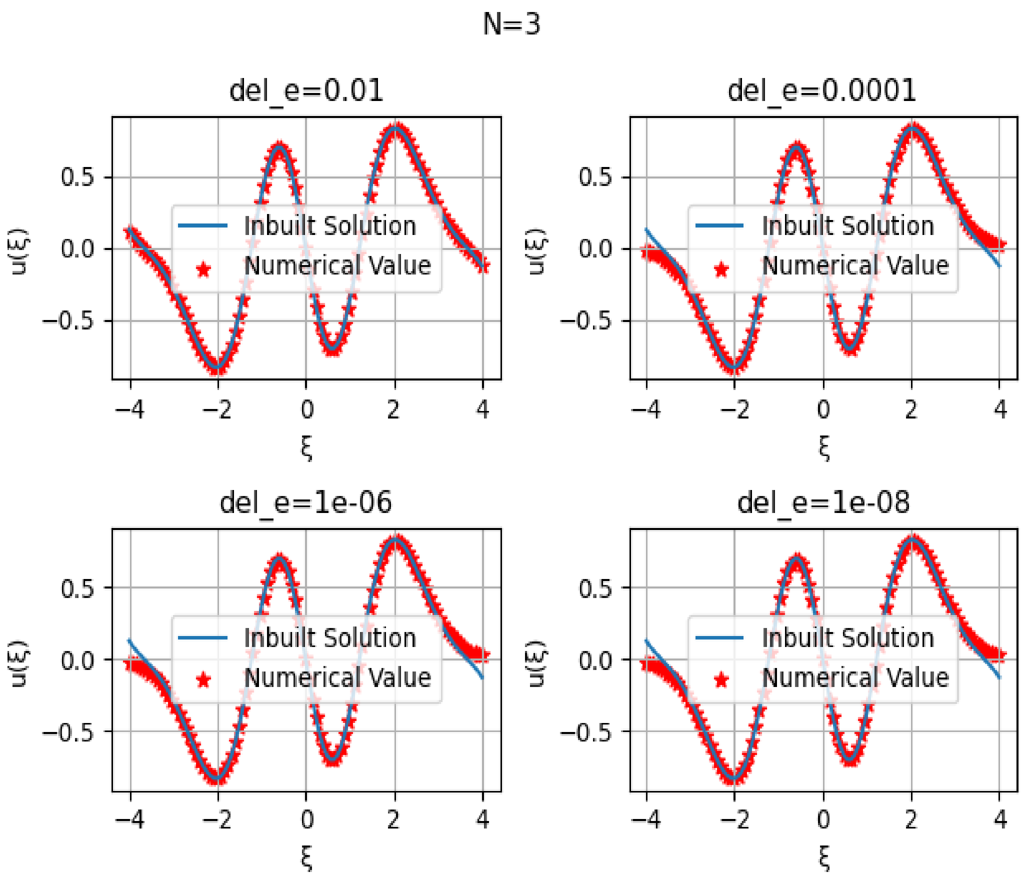


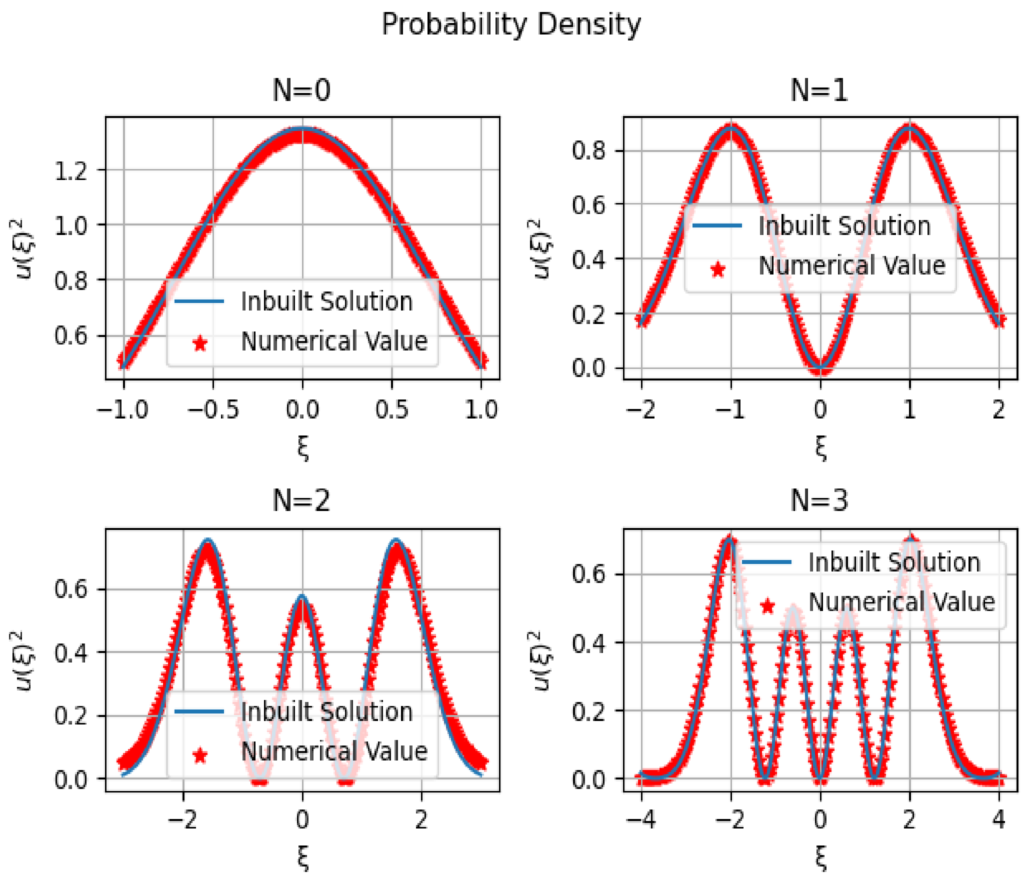


N=1
del_e=0.01
del_e=0.0001
del_e=1e-06
del_e=1e-08
Inbuilt Solution
Numerical Value
$(ξ)
ξ



N=2
del_e=0.01
del_e=0.0001
del_e=1e-06
del_e=1e-08
Inbuilt Solution
Numerical Value
u(ξ)
ξ


N=3
del_e=0.01
del_e=0.0001
del_e=1e-06
del_e=1e-08
Inbuilt Solution
Numerical Value
u(ξ)
ξ
0.5
0.0
−0.5
−4
−2
0
2
4


Probability Density
N=0
N=1
N=2
N=3
Inbuilt Solution
Numerical Value
$u(\xi)^2$
ξ
−1.0
−0.5
0.0
0.5
1.0
1.2
0.8
0.6
0.4
0.2
−2
−1
0
1
2
−4
4

```
PS C:\Users\adn19> & C:/Users/adn19/anaconda3/python.exe "d:/Sem 5/Quantum M
                         Energy in eV at N=0
   Approximated Eigen Values  Analytical Eigen Values
0              3.624851e-01           3.624847e-01
1              3.588603e-01           3.588599e-01
2              3.479857e-01           3.479854e-01
3              3.298615e-01           3.298611e-01
4              3.044876e-01           3.044872e-01
5              2.718640e-01           2.718636e-01
6              2.319907e-01           2.319904e-01
7              1.848678e-01           1.848674e-01
8              1.304951e-01           1.304947e-01
9              6.887276e-02           6.887239e-02
10             7.249688e-07           3.624844e-07
PS C:\Users\adn19>
```