Assignment 10 - H-Atom Using Shooting Method

SGTB Khalsa College, University of Delhi Preetpal Singh(2020PHY1140)(20068567043)

Unique Paper Code: 32221501

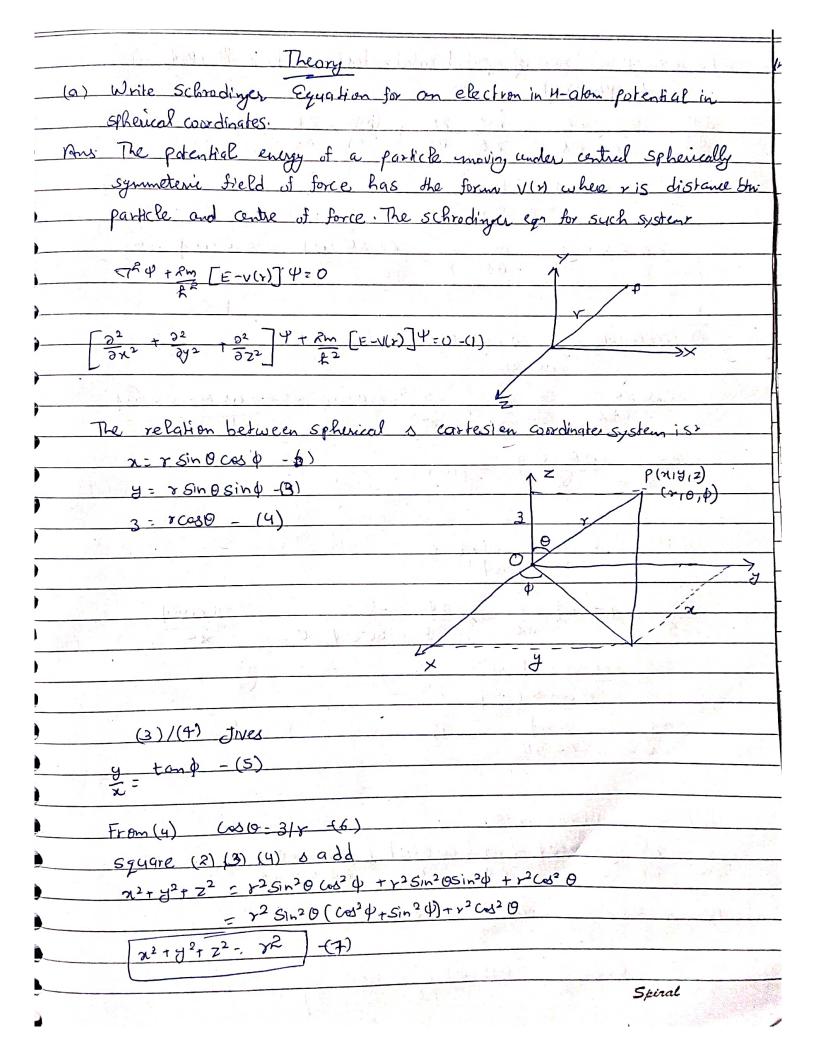
Paper Title: Quantum Mechanics and Applications

Submitted on: September 22, 2022

B.Sc(H) Physics Sem V

Submitted to: Dr. Mamta

Name => Preetpul Singh
Roll No-=> 2020PHY1140
B.Sc(Hons) Physics - Sem 5
amlab Assignment - 10
GMLab Assignment - 10 H Alom Using Shooting Method



Spiral

S.E in Poler form is given ast 1 2 (12 24) + 1 2 (51n 0 34) + 1 324 + 2m (E-V=0 Multiply both sides by 2 sin20 $\frac{\sin^2\theta}{\partial r} \frac{\partial}{\partial r} \left(\frac{r^2 \partial r}{\partial r} \right) + \frac{\partial}{\partial \theta} \left(\frac{\sin\theta}{\partial \theta} \right) + \frac{\partial^2 \psi}{\partial \theta^2} + \frac{2m(E-v)r^2 \sin^2\theta}{E^2} = 0$ Take 4 (r, 0, 0)= R(r)P(0) Cx (0) -(2) Put (2) in (1) (1) 5in20 3 (r2 2R(r)) + 1 0 (sin0 2P(0)) sin0 + 1 2 Q(0)

R(r) 30 (30) 4 Q(4) 342 2m (E-V) r2Sin20 = 0 $\frac{1}{R(x)} \frac{\sin^2 \theta \cdot \partial}{\partial r} \left(\frac{\gamma^2}{\partial x} \frac{\partial x}{\partial r} \right) + \frac{1}{P(\theta)} \frac{\partial}{\partial \theta} \left(\frac{\sin \theta}{\partial \theta} \frac{\partial P(\theta)}{\partial \theta} \right) \frac{\sin \theta}{2} + \frac{2m(E-v)}{2} \frac{\gamma^2 \sin^2 \theta}{2}$ LHS. depends on road o whereas RHS depends on 4, but IHS-RHS, so they can be equated to a constant und 1 Sin20 2 (r2 2 R(r)) + 1 2 (Sin 10 2 P(0)) Sin 0 + 2m (E-V) r2 Sim 0 = 12 (3 $\frac{1}{R(r)} \frac{1}{2r} \left(\frac{3r}{3r}\right) + \frac{1}{P(\theta)} \frac{3}{5\theta} \left(\frac{1}{5\ln\theta} \frac{3P(\theta)}{3\theta}\right) \frac{1}{5\ln\theta} + \frac{2m(E+V)}{42} = \frac{m^2}{5\ln^2\theta}$ Now again by separation of constant $\frac{1}{R(r)} \frac{\partial}{\partial r} \left(\frac{r^2}{\partial r} \frac{\partial R(r)}{\partial r} \right) = \frac{2m(E-V)r^2}{4\pi} = \frac{m^2}{\sin^2 \theta}$) 2 (Sinp 2P(0)) 1 P(0) 20 (Sinp 2P(0)) 5in 0

Equate (4) with a constant

 $\frac{1}{R(r)} \frac{2}{5r} \left(\frac{r^2 2R(r)}{5r} \right) - \frac{2m(E-V)}{R^2} = \frac{2(Rr)}{R^2}$

L Radial Part

 $\frac{m^2}{\sin^2 \Theta} = \frac{1}{8(0)} \frac{3}{30} \left(\frac{\sin \theta}{30} \frac{38(0)}{30} \right) \frac{1}{\sin \theta} - \frac{1}{8(0)}$

Angular Part

(C) Convert Radial Part into dimensionless form. Rescale & by 6. Roradius and energy with ground state energy

| 0 (r2 pR(r)) - 2m (E-v)r2 P(P+1) -(1)

Taking R=K

dr dr -(1)

	Date
From (3) Vessective ist	The Aurilla Strain
C) ESTECHIE IST	The King of the same
Veff = V(Y) + P(P+1) &2	
Veff = V(Y) + P(P+1) &2 2mx2	1 1 1 1 1 1 1 1 1
Taking v(r) from classical theory	Y-10-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
V(x)=-e2	
UNEOR	
Vess = -e2 P(P+1) R2 Un For Zmr2	
From Harlro 27 (rarto	and the state of t
2	
Vest = -e2 PLRADE2 UTTEO XYYO ZMX YIZYZ	
= exxme2 pressed	dry del .
- e xmez t l(l+1) f2 x m2 e4 UTEUXYX UTEO FZ Zm xx x 16 172 E3 f4	
=-me4 P(P+1) # m²e4	
16 M2 E02 21 K2 32 M TRE E02 KM 212	
2200	The state of the s
16 17 2 50 2 K2 81 7 7 7 12 - (5)	•
P ((1)) ((5)) (a)	and the state of t
Put (4) &(5) in (3)	and the second second
2m ro2 ori2 (one4 (-1) + P(P+1)) = EK	Harris Land

$$\frac{2\sqrt{3}}{2\sqrt{11}} = \frac{2\sqrt{3}}{2\sqrt{2}} = \frac{2\sqrt{3}}{$$

Ground State Energy of Hydrogen quan is

1E,1- cme4

282(4118.)2

$$\frac{\left[-\frac{3^{2}}{3^{2}}\left(\frac{2}{r^{1}}+\frac{P(P+1)}{2^{2}}\right)\right]}{2(4176)^{2}}\frac{|k(r)|}{2(4176)^{2}}$$

Take E - e, which is dimensionless

$$\left[\frac{-\partial^2 k}{\partial s^{1/2}} + 2\left(\frac{-1}{s} + \frac{\varrho(\varrho+1)}{\varrho(\varrho+1)}\right)\right] = k(\gamma') e$$

This is radial Equation in Dimensionless form

(d)	boundary Conditions for mumerical solutions ysing RK4
	boundary Conditions for mumerical solutions using RK4 with shooting and Numerov with shooting method.
Ang.	
the.	Boundary Conditions for Numerov Methodt
	4(2)=0
	4(47h)= h
	here 'h' is step size which is the gold size
	here 'h' is step size which is the gold size obtained by diving X-array into N+1 gold points.
	Boundary Conditions for RK4 Methods
	u(4)= 0
	u'(a)= orbitrary Number

Programming

```
import numpy as np
from scipy.linalg import eigh
3 import matplotlib.pyplot as plt
4 import scipy.integrate as integrate
5 import pandas as pd
6 from scipy.special import assoc_laguerre
7 def diag_mat(xi,xf,N,1):
      X = np.linspace(xi,xf,N+2)
      x = X[1:-1]
9
10
      h = x[1] - x[0]
      a,v=np.zeros((len(x),len(x))),np.zeros((len(x),len(x)))
11
      for i in range(len(x)):
12
13
           for j in range(len(x)):
               if i==j:
14
                   a[i][i]=2/h**2
15
16
                   v[i][i]=(-2/x[i])+((1)*(1+1))/(x[i]**2)
               elif i==j+1:
17
18
                   a[i][j]=-1/h**2
               elif i == j-1:
19
                   a[i][j] = -1/h**2
      A = (a + v)
21
22
      eig = eigh(A)
23
      return eig,x
24
25 def Analytic(x,n,1):
      return ((2*x/n)**(1)*assoc_laguerre(2*x/n,n-1-1,2*1+1))/(np.exp(x/n))*x
26
27
28 m=1.67*10**(-27)
29 def Veff(x, 1):
      Vef = (1 * (1 + 1) / (x ** 2)) - (2 / x)
30
       V = -(2 / x)
31
32
33
      return Vef, V
34
35 def plot(i, l, power):
      H, x = diag_mat(0.01, 30, 1000, 1)
36
37
      u = H[1][:, i]
      c = integrate.simps(u ** 2, x)
38
39
      N = u / np.sqrt(c)
      plt.plot(x, N ** power, label=f'l, n={l, i+1}')
40
      plt.title("Radial Probability Density for different n and 1")
41
      plt.xlabel("x")
42
      plt.ylabel("\$(u_r(\u03BE))^2\$")
43
      plt.grid()
44
45
      plt.legend()
46
47
      # plt.show()
48
49 N = 1000
xi = 10 * * -14; xf = 30
52 #A-i
53 # for i in range(0, 4):
54 #
        H, x = diag_mat(0.1, 50, N, i)
        Vef, V = Veff(x, i)
55 #
56 #
        plt.scatter(x, V,label=f'V,l={i}')
        # plt.plot(x, Vef,label=f'V_eff,l={i}')
57 #
        plt.xlabel("x")
58
         plt.ylabel("V_eff")
59 #
60 #
        plt.legend()
61 # plt.show()
62
63 # for i in range(0, 4):
H, x = diag_mat(0.1, 50, N, i)
^{65} # Vef, V = Veff(x, i)
```

```
plt.plot(x, V,label=f'V,l={i}')
66 #
        plt.xlabel("x")
67 #
68 #
        plt.ylabel("V")
69 #
        plt.legend()
70 # plt.show()
71
72 #-----#
73 for i in range(0,4,1):
      U_,x=diag_mat(xi,xf,N,0)
74
      U=U_[1][:,i]
75
76
      u_norm=U/np.sqrt(integrate.simps(np.power(U,2),x))
      U_anal = Analytic(x,i,0)
77
      u_anal_normalised=U_anal/np.sqrt(integrate.simps(np.power(U_anal,2),x))
78
79
      plt.scatter(x,np.power(u_norm,1),label=f'Numerical, n={i+1}')
      \verb|plt.plot(x,np.power(u_anal_normalised,1),label=f'Analytical, n=\{i+1\}')|
80
81
      plt.xlabel("x")
      plt.ylabel("$(u_r(\u03BE))$")
82
      plt.title("Radial Wavefunction for 1=0")
83
84 plt.legend()
85 plt.show()
87 # #-----#
88 # print("First 10 Energy Eigen Values for 1=0 and r_max=30")
89 # p=[]
90 # for i in range(1,11):
        p.append(-1/i**2)
91 #
92 # data ={
93 #
        "Numerical Eigen Values ": U_[0][:10],
94 #
        "Analytical Eigen Values ":p
95 # }
96 # print(pd.DataFrame(data))
97
98 # #-----
                          -----#
99 # for i in range(0,4,1):
100 #
        U_,x=diag_mat(xi,xf,N,1)
101 #
        U=U_[1][:,i]
102 #
       u_norm=U/np.sqrt(integrate.simps(np.power(U,2),x))
103 #
       U_anal = Analytic(x,i,1)
       u_anal_normalised=U_anal/np.sqrt(integrate.simps(np.power(U_anal,2),x))
104 #
      plt.scatter(x,np.power(u_norm,1),label=f'Numerical, n={i+1}')
105 #
        \verb|plt.plot(x,np.power(u_anal_normalised,1),label=f'Analytical, n=\{i+1\}'|)|
106 #
# plt.legend()
108 # plt.show()
109
# print("First 10 Energy Eigen Values for l=1 and r_max=30")
111 # p=[]
112 # for i in range(1,11):
# p.append(-1/(i+1)**2)
114 # data ={
       "Numerical Eigen Values ": U_[0][:10],
116 #
        "Analytical Eigen Values ":p
117 # }
# print(pd.DataFrame(data))
119
                          -------Qb_i------#
121 # for i in range(0,4,1):
        U_,x=diag_mat(xi,xf,N,2)
122 #
123 #
        U=U_[1][:,i]
124 #
        u_norm=U/np.sqrt(integrate.simps(np.power(U,2),x))
125 #
        U_anal = Analytic(x,i,2)
126 #
        u_anal_normalised=U_anal/np.sqrt(integrate.simps(np.power(U_anal,2),x))
       plt.scatter(x,np.power(u_norm,1),label=f'Numerical, n={i+1}')
        \verb|plt.plot(x,np.power(u_anal_normalised,1),label=f'Analytical, n=\{i+1\}'|
128 #
# plt.legend()
130 # plt.show()
131
# print("First 10 Energy Eigen Values for 1=2 and r_max=30")
```

```
133 # p=[]
134 # for i in range(1,11):
p.append(-1/(i+2)**2)
136 # data ={
"Numerical Eigen Values ": U_[0][:10],
"Analytical Eigen Values ":p
139 # }
# print(pd.DataFrame(data))
141
142 # C
143
144 # '''for n=1:'''
145 # plot(0, 0, 2)
146 # plt.show()
147 # '''for n=2:'''
148 # plot(1, 0, 2)
149 # plot(1, 1, 2)
150 # plt.show()
151 #n=3
152 # plot(2, 0, 2)
153 # plot(2, 1, 2)
154 # plot(2, 2, 2)
155 # plt.show()
```

Result and Discussion

state	Energy 1 = 0	Energy L = 1	Energy 1 = 2
1	-0.992816	0.000000	0.000000
2	-0.249941	-0.249989	0.000000
3	-0.111063	-0.111111	-0.111111
4	-0.062439	-0.062500	-0.062500
5	-0.039927	-0.040001	-0.040000

