Scilab Manual for CC-XI: Quantum Mechanics &Applications (32221501)

by Dr Neetu Agrawal
Physics
Daulat Ram College, University Of Delhi [1]

Solutions provided by
Dr Neetu Agrawal
Physics
Daulat Ram College, University Of Delhi

May 10, 2025

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website http://scilab.in

Contents

List of Experiments

List of Figures

Experiment: 1

Solve the s-wave Schrodinger equation for the ground state and the first excited state of the hydrogen atom.

Scilab code Solution 1.01 Hydrogen atom problem

```
1 // Submitted by Dr. Neetu Agrawal. Assistant
    Professor, Physics Dept., Daulat Ram College,
    Univ. of Delhi
2
3 // Aim: Solve the s-wave Schrodinger equation for
    the ground state and the first excited
4 //state of the hydrogen atom.
```

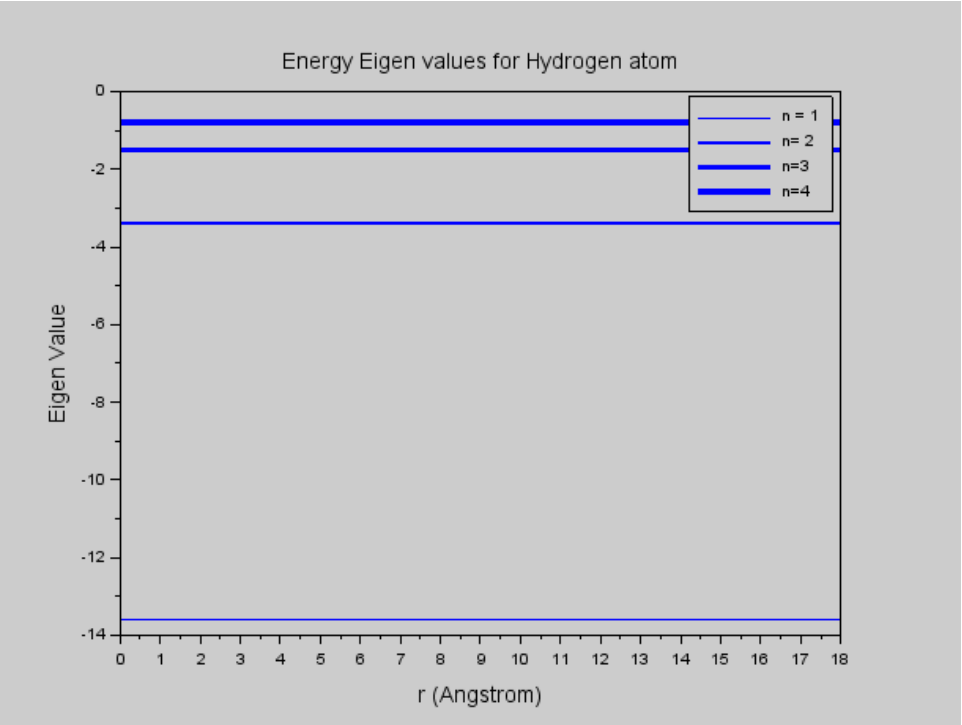

Figure 1.1: Hydrogen atom problem

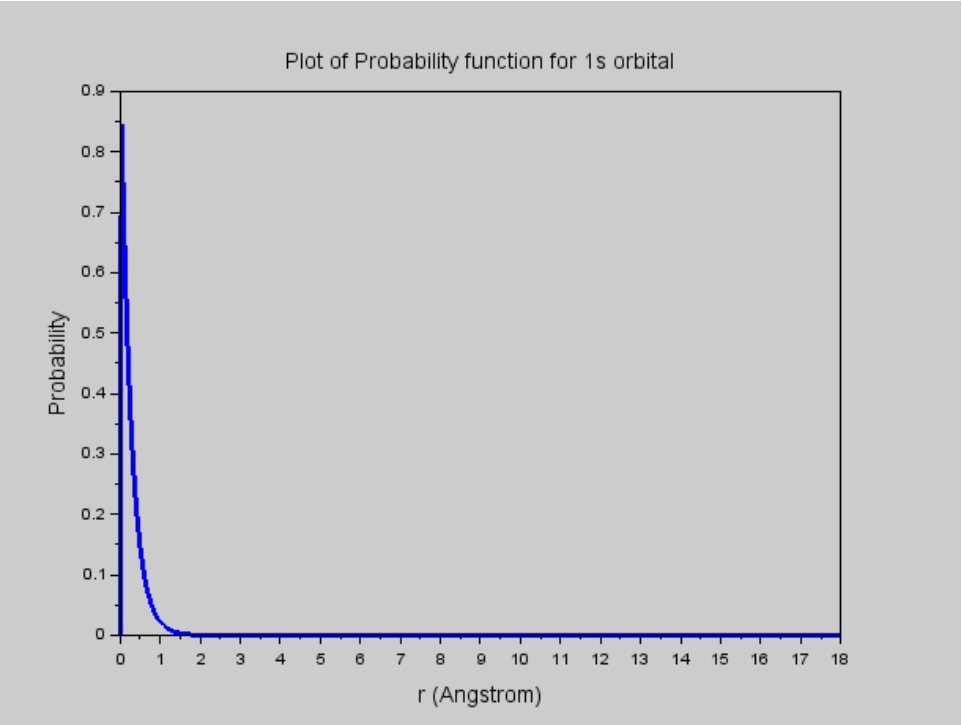

Figure 1.2: Hydrogen atom problem

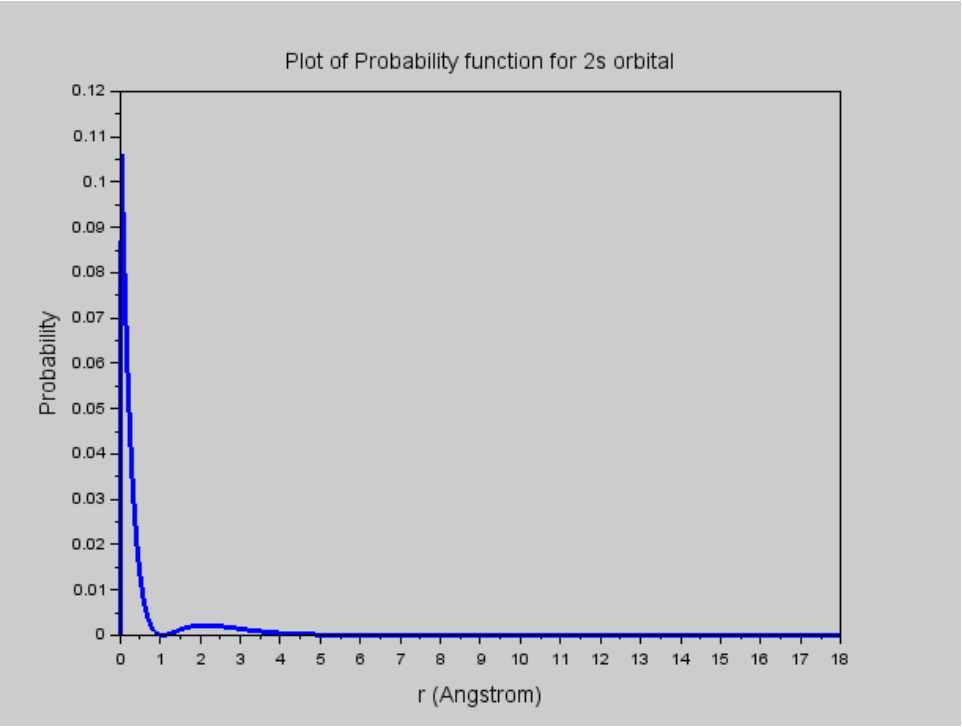

Figure 1.3: Hydrogen atom problem

```
Input the number of intervals (should be around 500 to 750 for good computation)500
 Ground state energy (1S orbital) for hydrogen atom is (in eV) :
  - 13.598212
 First excited state (2S orbital) energy  for hydrogen atom is (in eV):
  - 3.4025004
 Second excited state (3S orbital) energy  for hydrogen atom is (in eV):
  - 1.5118515
 Third excited state (4S orbital) energy  for hydrogen atom is (in eV):
  - 0.7744925
-->
```

Figure 1.4: Hydrogen atom problem

```
5  //Here, m is the reduced mass of the electron.
     Obtain the energy eigenvalues and plot
6  // the corresponding wavefunctions. Remember that
     the ground state energy of the
7  //hydrogen atom is -13.6 eV. Take e = 3.795 (eV )
     ^1/2, c = 1973 (eV ) and m = 0.511x10^6 eV/c2.
8
9  close;
10 clear;
11 clc;
12
13 // declaring constant values
14
15 hbarc=1973;//Plancks constant h divided by 2*(pi)
     called as hbar=(h/2*pi). This factor when
     multiplied by speed of light c then hbar*c in the
      units of (eV ) comes out to be 1973;
16 mcsq=0.511*10^(6);//This is mass of electron*c ^2 we
      call it mcsq in units of (eV);
17 e = 3.795; // (eV )^1/2
18
19 // We hereby input the 'r' values so as to obtain
     the potential V(r) as a function of 'r')
20 r_min=0 // in units of angstrom
21 r_max=18 // in units of angstrom
22 N =  input("Input the number of intervals (should be
      around 500 to 750 for good computation)")
23 s = (r_max-r_min)/N; //step size
24 factor1=-(hbarc^2)/(2*mcsq*s^2);  // this factor is
     (hbar^2*c^2/2m*c^2) divided by s^2 //k=(hbar_c*
     hbar_c)/(2*m)
25
26 // Kinetic energy matrix (Using central difference
     formula)
27
28 T=zeros(N-1,N-1)
29 for i=1:N-1
30     T(i,i)=-2
```

```
31      if i<N-1 then
32          T(i,i+1)=1
33          T(i+1,i)=1
34       end
35  end
36
37  T_matrix = factor1*T;     // Kinetic Energy Matrix
       final (scaling done)
38
39  //Potential energy matrix
40  V_matrix=zeros(N-1,N-1)
41  for i=1:N-1
42      r(i)=r_min+i*s
43      V_matrix(i,i)=-(e*e)/r(i);
44  end
45
46  // Hamiltonian matrix
47  H_matrix=T_matrix+V_matrix
48
49  // energy eigenvalue and eigenstates
50  [u,eigen]=spec(H_matrix);
51
52  // displaying of the ground and first excited state
       energies
53  disp("Ground state energy (1S orbital) for hydrogen
       atom is (in eV) : ")
54  disp(eigen(1,1))
55  disp("First excited state (2S orbital) energy  for
       hydrogen atom is (in eV): ")
56  disp(eigen(2,2))
57  disp("Second excited state (3S orbital) energy  for
       hydrogen atom is (in eV): ")
58  disp(eigen(3,3))
59  disp("Third excited state (4S orbital) energy  for
       hydrogen atom is (in eV): ")
60  disp(eigen(4,4))
61
62  rmatrix = [0;r;15]; //including the first and last
```

```
    point at which wavefunction is zero.
63
64 //Displaying the first four energy eigen values
65 figure(1);
66 //scf()
67 for n = 1:1:4
68 plot(rmatrix, eigen(n,n)*ones(N+1,1), 'linewidth',n)
69 hl=legend(['n = 1';'n= 2';'n=3';'n=4']);
70 title('Energy Eigen values for Hydrogen atom','
    fontsize',3)
71 xlabel('r (Angstrom)','fontsize',3)
72 ylabel('Eigen Value','fontsize',3)
73 end
74
75 //Plotting the Probability |Psi^2| as a function of
    r
76 figure(2);
77 //scf()
78 R_wave_1s=u(:,1)./r;   //Radial wavefuction
79 R_wave_1s_final=[0;R_wave_1s;0];  //including the
    first and last point at which wavefunction is
    zero.
80 // plot of probability function
81 P_wave_1s= R_wave_1s_final.*R_wave_1s_final;
82 plot(rmatrix,P_wave_1s, 'linewidth',3)
83 title('Plot of Probability function for 1s orbital',
    'fontsize',3)
84 xlabel('r (Angstrom)','fontsize',3)
85 ylabel('Probability','fontsize',3)
86
87 figure(3);
88 //scf()
89 R_wave_2s=u(:,2)./r;
90 R_wave_2s_final=[0;R_wave_2s;0];   //including the
    first and last point at which wavefunction is
    zero.
91 // plot of probability function
92 P_wave_2s= R_wave_2s_final.*R_wave_2s_final;
```

```
93 plot(rmatrix,P_wave_2s, 'linewidth',3)
94 title('Plot of Probability function for 2s orbital',
     'fontsize',3)
95 xlabel('r (Angstrom)','fontsize',3)
96 ylabel('Probability','fontsize',3)
```

Experiment: 2

Solve the s-wave radial Schrodinger equation for an atom for the screened coulomb potential.

Scilab code Solution 2.0 Screened coulomb potential

```
1 //Solve the s-wave Schrodinger equation for the
    ground state and the first excited
2 //state of the hydrogen atom.
3
4 close;
5 clear;
6 clc;
7
8 // declaring constant values
9 hbarc=1973;// Plancks constant = h , hbar*c=(h/2*pi)
    ^c.
10 mcsq=0.511*10^(6);     // (mass of electron)*c ^2 in
    units of (eV);
11 e = 3.795; // (eV )^1/2
12 //a = input("Input the vaue of a in units of
```

```
    angstrom: ") ;
13 a=3 ; // in units of Angstrom
14 // we can check it for 3, 5, 7 angstrom etc..
15
16 r_min=0 // in units of angstrom
17 r_max=5 // in units of angstrom
18 N =  input("Input the number of intervals (should be
       around 500 to 1000 for good computation): ")
19 s = (r_max-r_min)/N; //step size
20 factor1=-(hbarc^2)/(2*mcsq*s^2);
21 // this factor is (hbar^2*c^2/2m*c^2) divided by s^2
       //k=(hbar_c*hbar_c)/(2*m)
22
23 // Kinetic energy matrix (Using central difference
    formula)
24 T=zeros(N-1,N-1)
25 for i=1:N-1
26     T(i,i)=-2
27     if i<N-1 then
28         T(i,i+1)=1
29         T(i+1,i)=1
30      end
31 end
32 // Kinetic Energy Matrix final (scaling factor
    included)
33 T_matrix = factor1*T;
34
35 //Potential energy matrix
36 V_matrix=zeros(N-1,N-1)
37 for i=1:N-1
38     r(i)=r_min+i*s
39     V_vector(i,1) = -((e*e)/r(i))*exp(-r(i)/a);
40     V_matrix(i,i)=-((e*e)/r(i))*exp(-r(i)/a);
41 end
42
43 // This is to plot the potential V(r)as a function
    of r
44 figure;
```

```
45 plot(r,V_vector,'-.','linewidth',3)
46 xlabel('r (in Angstrom units)','fontsize',2)
47 ylabel('V(r)','fontsize',2)
48 legend (['Screened coloumb potential'])
49
50 // Hamiltonian matrix
51 H_matrix=T_matrix+V_matrix
52
53 // energy eigenvalue and eigenstates
54 [u,eigen]=spec(H_matrix);
55
56 // displaying of the groundstate energies
57 disp("Ground state energy in the presence of
      screened coloumb potential is : ")
58 disp(eigen(1,1))
59
60 //Radial wavefuction
61 figure();
62 Psi=u(:,1)./r;
63 plot(r,Psi, 'm', 'linewidth',3)
64 title('Eign function plot in presence of screened
      coloumb potential','fontsize',2)
65 xlabel('r (in Angstrom units)','fontsize',2)
66 ylabel('Eigen Function','fontsize',2)
67 legend (['Ground state wavefunction'])
```

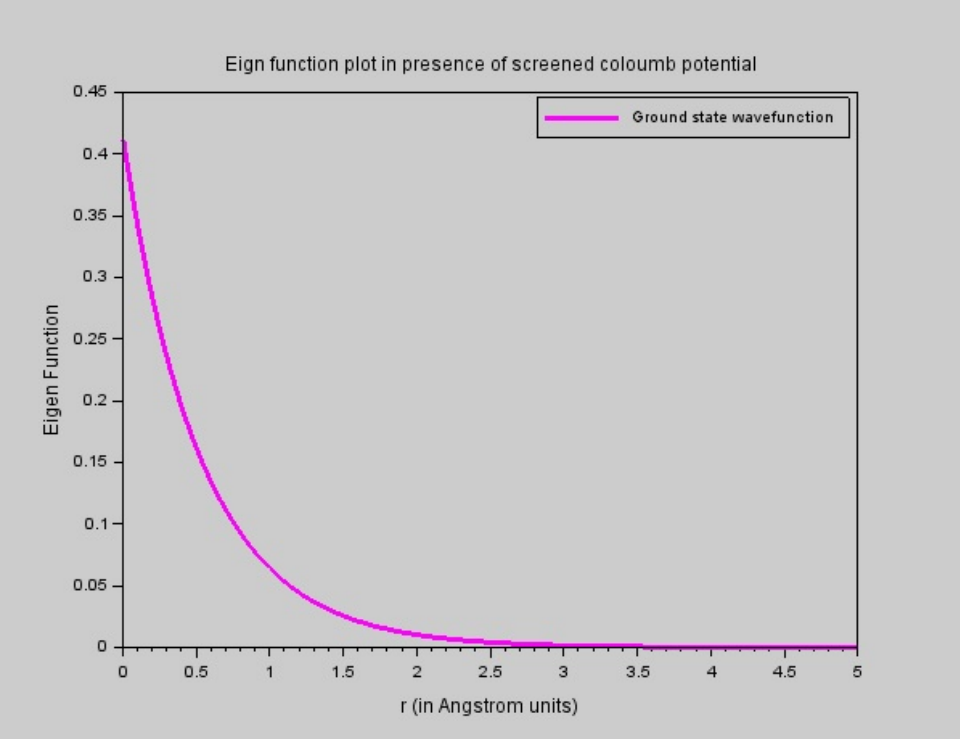

Figure 2.1: Screened coulomb potential

```
Scilab 6.1.0 Console

Input the number of intervals (should be around 500 to 1000 for good computation): 750


  "Ground state energy in the presence of screened coloumb potential is : "

  -9.3865350

-->
```

Figure 2.2: Screened coulomb potential

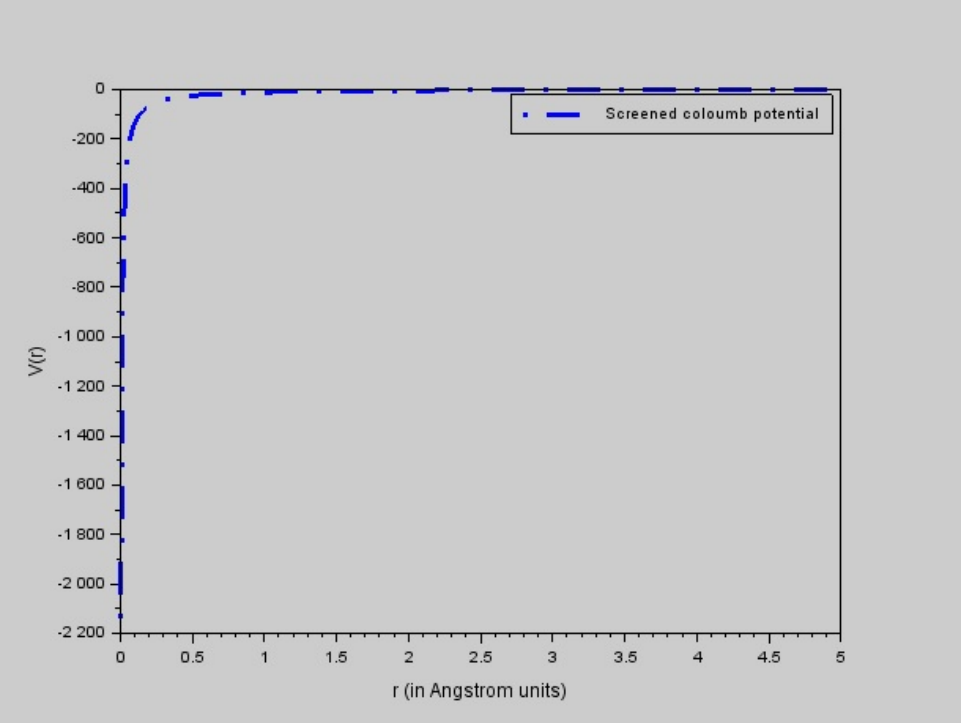

Figure 2.3: Screened coulomb potential

Experiment: 3

Solve the s-wave radial Schrodinger equation for a particle of mass m, for an harmonic oscillator potential. Plot wavefunctions.

Scilab code Solution 3.0 Harmonic Oscillator

```
1 //Solve the s-wave radial Schrodinger equation for a
      particle of mass m
2 //for an harmonic oscillator potential for ground
     state energy (in MeV).
3
4 close;
5 clear;
6 clc;
7
8 // declaring constant values
9 hbarc=197.3  //Plancks constant h divided by 2*(pi)
     called as hbarc=(h/2*pi)*c.
10 mcsq=940;  // mass of electron*c^2 = mcsq in units
```

```
    of (MeV);
k = 100;    // harmonic oscillator potential constant
    in units of MeV fm-2
b = 0;
disp("The value of b in the potential is chosen to
    be:");
disp(b)
//To obtain the plot the potential V(r) as a
    function of 'r'
r_min = 0;
r_max = 4;
N =  input("Input the number of intervals (should be
    around 500 to 1000 for good computation: )")
s = (r_max-r_min)/N;  //step size
factor1=-(hbarc^2)/(2*mcsq*s^2);

// making a row vector to input r values
for i=1:1:N
    rmat(1,i)=r_min+(i-1)*s
end

for ir = 1:1:size(rmat,2)
    r = rmat(ir);
V(1,ir) = 0.5*k*(r)^2 + 0.5*b*(r)^3;
end
figure;
plot(rmat,V,'-.','linewidth',3)

// Kinetic energy matrix (Using central difference
    formula)
T=zeros(N,N)
for i=1:1:N
    T(i,i)=-2;
    if (i<N)
        T(i,i+1)=1;
        T(i+1,i)=1
    end
end
```

```
43
44 T_matrix = factor1*T; // Kinetic Energy Matrix final
       in MeV
45
46 // Potential energy matrix in MeV
47 U_matrix = zeros(N,N)
48 for i = 1:1:N
49 U_matrix(i,i)=V(i)
50 end
51
52 // Hamiltonian matrix
53 Ham = T_matrix+U_matrix;
54 [u,eigen] = spec(Ham);
55 eigval_numeric = spec(Ham)
56
57
58 disp ('The first five eigen values obtained using
     FDM are:')
59 disp(eigval_numeric(1:5))
60
61 //Plotting the eigenvalues obtained by numerical
     computation
62
63 for n =1:1:5
64     eigvalue_num = eigval_numeric(n)
65     eigvalue_num_vector = eigvalue_num*ones(1,N);
66     plot(rmat,eigvalue_num_vector,'r')
67 end
68
69
70 // normalisation check (The value comes out to be 1)
71 normalisation = sum((u(:,1).*conj(u(:,1))))
72
73 // plotting the eigen functions.
74 //(plot of mod psi squared). Done the scaling of
     eigen function magnitude
75
76 psisq1 = (1/normalisation)*(u(:,1).*conj(u(:,1)))
```

```
77 plot(rmat(1,:),8000*psisq1'+eigval_numeric(1),'k')
78
79 psisq2 = (1/normalisation)*(u(:,2).*conj(u(:,2)))
80 plot(rmat(1,:),8000*psisq2'+eigval_numeric(2),'k')
81
82 psisq3 = (1/normalisation)*(u(:,3).*conj(u(:,3)))
83 plot(rmat(1,:),8000*psisq3'+eigval_numeric(3),'k')
84
85 psisq4 = (1/normalisation)*(u(:,4).*conj(u(:,4)))
86 plot(rmat(1,:),8000*psisq4'+eigval_numeric(4),'k')
87
88 psisq5 = (1/normalisation)*(u(:,5).*conj(u(:,5)))
89 plot(rmat(1,:),8000*psisq5'+eigval_numeric(5),'k')
90
91 title('Plot of first 5 probability density functions
      ','fontsize',2);
92 xlabel('r (in fm) -->','fontsize',2)
93 ylabel('Energy(in MeV)and scaled probability density
      functions)','fontsize',2)
94
95 legend(['Potential Plot','Eigenvalues'])
```

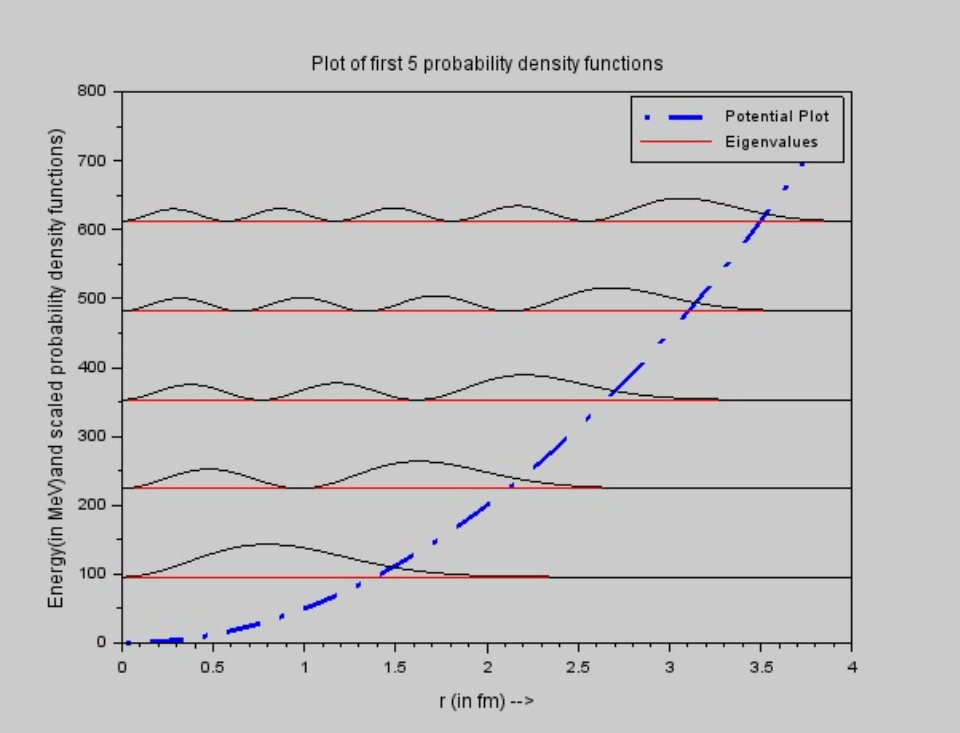

Figure 3.1: Harmonic Oscillator

```
Scilab 6.1.0 Console

  "The value of b in the potential is chosen to be:"

   0.
Input the number of intervals (should be around 500 to 1000 for good computation: )700


  "The first five eigen values obtained using FDM are:"

   96.011767
   224.45556
   352.969
   481.69927
   612.51479

-->
```

Figure 3.2: Harmonic Oscillator

Experiment: 4

Solve the s-wave radial Schrodinger equation for the vibrations of hydrogen molecule. Find the lowest vibrational energy (in MeV

Scilab code Solution 4.0 Morse Potential

```
1 //This program solves the s-wave radial Schrodinger
    equation for the vibrations of hydrogen molecule
    for the Morse potential
2 //Find the lowest vibrational energy (in MeV) of the
     molecule. Also plot the corresponding wave
    function. //Take: m = 940x106 eV/c2, D = 0.755501
     eV,    = 1.44, r0 = 0.131349    //Where    s
    the reduced mass of the two-atom system for the
    Morse potential //Find the lowest vibrational
```

```
Enter the number of intervals(should be around 500 to 1000 for good computation)750
Warning : redefining function: eval                    . Use funcprot(0) to avoid this message


 The Eigenvalues calculated using FDM are

    0.4042986

    0.7535893

    0.8403509

    0.9841226

-->
```

Figure 4.1: Morse Potential

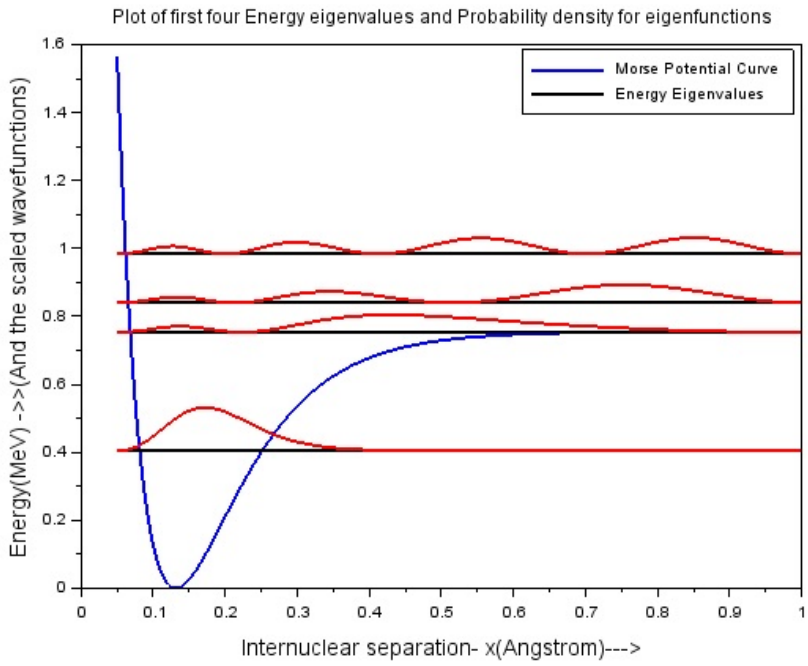

Figure 4.2: Morse Potential

```
    energy (in MeV) of the molecule to an accuracy of
     three significant digits. Also plot the  //
    corresponding wave function. //Take: m = 940x106
    eV/c2, D = 0.755501 eV,    = 1.44, ro = 0.131349

3
4  clear
5  close
6  clc
7
8  //Declaring the values of constants
9  hcutc=1973 //Planck's constant h divided by 2*pi (
       hcut=h/2*pi).This when multiplied by speed of
       light c gives hcutc(in units of eV A)
10 mcsq=940*10^6; //mass*c^2 in units of eV/c^2
11
12 x0=0.131349 //In units of Angstrom
13 alpha=1.44 //a*x0 where 'a' is a constant for
       particular molecule
14 D=0.755501 //Dissociation Energy In units of eV
15
16 //Getting values of x to plot 'V(x)' v/s 'x' plot
17 xmin=0.05; //in units of A^o
18 xmax=1; //in units of A^o
19
20 n=input("Enter the number of intervals(should be
       around 500 to 1000 for good computation)");
21 s=(xmax-xmin)/n; //step size
22
23 for i=1:1:n
24 x(1,i)=xmin+s*(i-1); //x vector of 1 row and n
       columns to input values of internuclear
       seperation
25 x_(1,i)=(x(1,i)-x0)/x0 //x_ vector= (x-x0)/x0
26 end
27
28 factor1=-(hcutc^2)/(2*mcsq*s^2) //this factor is (
       hcut^2*c^2/2*m*c^2*s^2)
```

```
29
30 //plot of potential V(x) v/s x
31 V=zeros(n,n); //Potential energy matrix
32 for i=1:1:n
33 V_matrix(i,i)=(D*(1-exp(-alpha*x_(1,i)))^2) //
      formula for potential
34 end
35
36 for j=1:1:size(V_matrix,2)
37 V_vec(1,j)=V_matrix(j,j)
38 end
39 plot(x,(V_vec),"linewidth",2) //linewidth command to
       set width of line
40 title("Plot of first four Energy eigenvalues and
      Probability density for eigenfunctions","fontsize
      ",2) //title of the plot
41 xlabel('Internuclear separation- x(Angstrom)---->',"
      fontsize",3) //fontsize command to set the font
      size of labels
42 ylabel('Energy(MeV) -->>(And the scaled wavefunctions
      )',"fontsize",3)
43
44 for i=1:1:n
45 for j=1:1:n
46 if i==j then
47 K(i,j)=-2;
48 elseif i==(j-1)|i==(j+1) then
49 K(i,j)=1;
50 else
51 K(i,j)=0;
52 end
53 end
54 end
55 T_matrix=factor1*K; //Kinetic energy matrix
56 H_matrix=V_matrix+T_matrix; //Hamiltonian matrix
57 eval=spec(H_matrix); //eval stores the eigenvalues
      of matrix H
58 [a,b]=spec(H_matrix); //a stores the eigenvectors
```

```
59 disp("The Eigenvalues calculated using FDM are")
60 for i=1:1:4
61 disp(eval(i,1))
62 end
63
64 //Plotting the eigenvalues obtained by Numerical
     computation
65 for i=1:1:4
66 eval_num=eval(i)
67 eval_num_vec=eval_num*ones(1,n)
68 plot(x,eval_num_vec,'k',"linewidth",2)
69 end
70 //normalization check
71 normalisation=sum(a(:,1).*conj(a(:,1)));
72 //Plotting the first four Eigen functions (plot of
     mod psi square).We do the scaling of Eigen
     functions magnitudes by
73 //a factor of 15 and raise to level of eigen values
     to make the conventional plot with eigen values
     and
74 //eigen functions on the same line
75 psisq1=(1/normalisation)*(a(:,1).*conj(a(:,1)))
76 psisq2=(1/normalisation)*(a(:,2).*conj(a(:,2)))
77 psisq3=(1/normalisation)*(a(:,3).*conj(a(:,3)))
78 psisq4=(1/normalisation)*(a(:,4).*conj(a(:,4)))
79 plot(x(1,:),15*psisq1'+eval(1),'r',"linewidth",1.5)
80 plot(x(1,:),15*psisq2'+eval(2),'r',"linewidth",1.5)
81 plot(x(1,:),15*psisq3'+eval(3),'r',"linewidth",1.5)
82 plot(x(1,:),15*psisq4'+eval(4),'r',"linewidth",1.5)
83 legend([' Morse Potential Curve','Energy Eigenvalues
     '])
```

Experiment: 5

Plot and analyse the wavefunctions for particle in an infinite potential well.

Scilab code Solution 5.0 1D Box potential

```
1   close
2 clear
3 clc
4
5 //This is program calculates the energy eigen vaules
      and eigen functions
6 //for a particle in an infinite potential well of
     width a
7
8
9 // declaring constant values
```

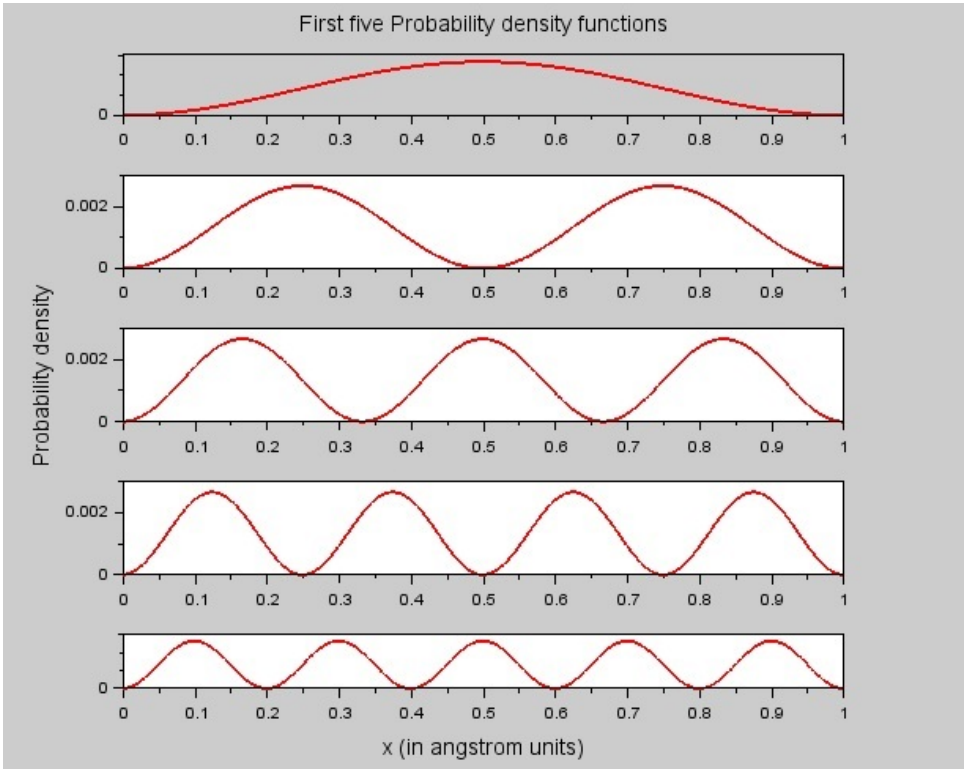

Figure 5.1: 1D Box potential

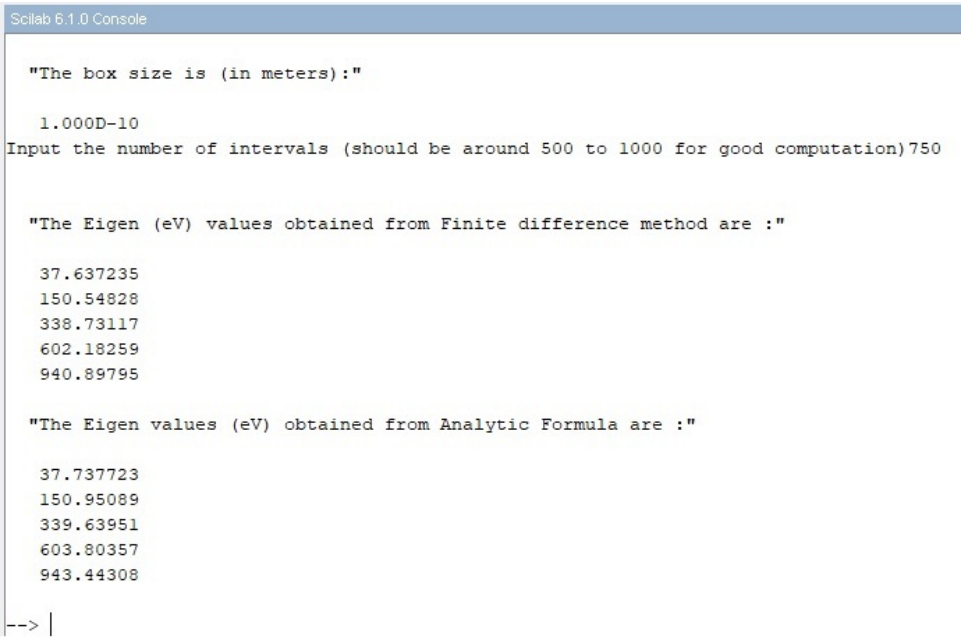

Figure 5.2: 1D Box potential

```
10 m = 9.1e-31;  // mass of electron
11 hplanck= 6.63*1e-34  // value of plancks constant
12 hbar = hplanck/(2*%pi)    // value of h/2*pi (let we
       call it hbar)
13 hbarsqbytwo_m_term = (hbar^2)/(2*m);  //(value of
      hbar square by 2*m)
14 eV = 1.6e-19; //   vale of electron volt si units
15 MeV = (1e+6)*eV; // writing Mega elctron volt
16 Angst = 1e-10;   // value of one angstrom
17
18 x_min = 0;
19 x_max = 1*Angst; //This value can be changed to
      change width of 1D box
20
21 disp ('The box size is (in meters):' )
22 disp(x_max)
23
24 N = input("Input the number of intervals (should be
```

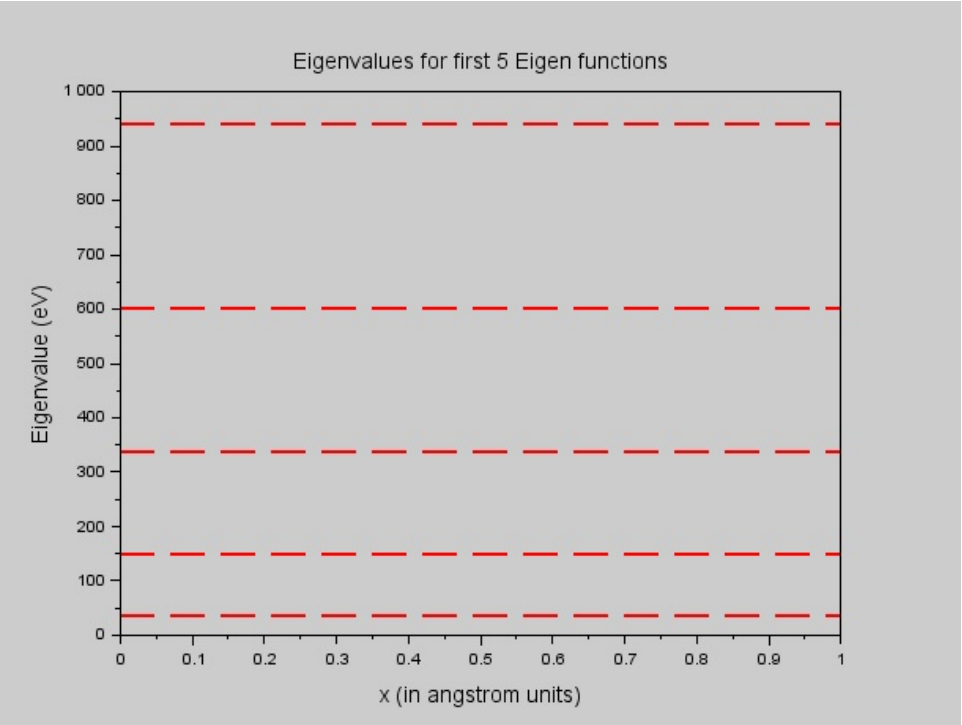

Figure 5.3: 1D Box potential

```
    around 500 to 1000 for good computation)")
a = (x_max-x_min);    // width of 1 d box
s = (x_max-x_min)/N;    // step size
fac = -hbarsqbytwo_m_term/(s^2);    // this factor
    is (hbar^2/2m) divided by h^2

// making a row vector to input x values
for i=1:1:N
    x(1,i)=x_min+(i-1)*s
end
// Kinetic energy matrix (Using central difference
    formula)
T=zeros(N,N)
for i=1:1:N
    T(i,i)=-2;
    if (i<N)
        T(i,i+1)=1;
        T(i+1,i)=1
    end
end
T_matrix = fac*T/eV; // Kinetic Energy Matrix in eV

// potential energy matrix
U_matrix = zeros(N,N)
for i = 1:1:N
U_matrix(i,i)=0;
end

// Hamiltonian matrix H=U+T
H_matrix = T_matrix+U_matrix;
[u,eigen] = spec(H_matrix);
eigval_numeric = spec(H_matrix)

// normalisation check
normalisation = sum((u(:,1).*conj(u(:,1))))

// By theoretically achieved Formulae
for n= 1:1:5
```

```
60 eigval_theory = (n^2*%pi^2*hbar^2)/(2*m*a^2)  // in
   SI units
61 eigval_th_eV(n) = eigval_theory/eV   // in eV
62 end
63
64 disp('The Eigen (eV) values obtained from Finite
   difference method are :')
65 disp(eigval_numeric(1:5))
66 disp('The Eigen values (eV) obtained from Analytic
   Formula are :')
67 disp(eigval_th_eV(1:5))
68
69 //Plotting the eigenvalues
70 figure;
71 for n =1:1:5
72     temp1 = eigval_numeric(n)
73     eigval_numeric_vector = temp1*ones(1,N);
74     plot(x/Angst,eigval_numeric_vector,'r--', '
       linewidth', 2)
75     xlabel('x (in angstrom units)','fontsize',3)
76     ylabel('Eigenvalue (eV)','fontsize',3)
77     title('Eigenvalues for first 5 Eigen functions',
       'fontsize',3)
78 end
79
80
81 // plotting the Probbility functions. (plot of mod
   psi squared)
82
83 figure;
84 for in =1:1:5
85 psisq(:,in) = (u(:,in).*conj(u(:,in)))
86 end
87
88 subplot(5,1,1)
89 title('First five Probability density functions','
   fontsize',3)
90 plot(x(1,:)/Angst,psisq(:,1)','r','linewidth',2)
```

```
91  subplot(5,1,2)
92  plot(x(1,:)/Angst,psisq(:,2)','r','linewidth',  2)
93  subplot(5,1,3)
94  plot(x(1,:)/Angst,psisq(:,3)','r','linewidth', 2)
95  ylabel("Probability density",'fontsize',3)
96  subplot(5,1,4)
97  plot(x(1,:)/Angst,psisq(:,4)','r','linewidth', 2)
98  subplot(5,1,5)
99  plot(x(1,:)/Angst,psisq(:,5)','r','linewidth', 2)
100 xlabel("x (in angstrom units)",'fontsize',3)
```