Numerov Morse Final Artigo

May 2, 2022

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
from numpy import random
from random import random

x = np.linspace(-10,10,1001)

def  V(x):
     Vx=16*(1.-np.exp(-2*x))**2      #effective potential
     return  Vx

def vec_max(dim,x):                 #left maximum
    xmax=0
    N=dim
    for j in range(int(N)):
        if j<int(N) and abs(x[j])>xmax:
           xmax=abs(x[j])
        else:
           continue
    return xmax

def nrovl(y0, y1, x0, E, h, iflag):  #left solution
    q0 = (E-V(x0))
    q1 = (E-V(x0+h))
    q2 = (E-V(x0+h+h))
    p0 = (1 + h*h*q0/12)
    p1 = 2*(1 - 5*h*h*q1/12)
    p2 = 1 + h*h*q2/12
    y2 = (p1*y1-p0*y0)/p2
    if iflag<1:
        print(" x0 = ", x0," y0 = ", y0," V = ",V(x0))
        print(" x1 = ", x0+h," y1 = ",y1," V = ",V(x0+h),
        " y2 = ",y2)
    return y2


def  nrovr(y0, y1, x0, E, h, iflag):  #right solution
```

```
    q0  =  (E-V(x0))
    q1  =  (E-V(x0-h))
    q2  =  (E-V(x0-h-h))
    p0  =  (1 + h*h*q0/12)
    p1   =   2*(1 - 5*h*h*q1/12)
    p2   =   1 + h*h*q2/12
    y2   =  (p1*y1-p0*y0)/p2
    if iflag<1:
       print(" x_100 = ", x0," y0_100 = ", y0," V = ",V(x0))
       print(" x_99 = ", x0-h," y_99 = ",y1," V = ",V(x0-h)," y_98 = ",y2)
    return  y2

def espectro(xl,xu,h,delta,eps,dim,nmax,kmax,Ein,Vmax,dE,iflag):

    xx=list(range(dim));     yy=list(range(dim));
    yl=list(range(dim));     yr=list(range(dim))
    ee=list(range(nmax));    ff=list(range(nmax));    ff2=list(range(nmax))
    yy1=list(range(dim));    yy2=list(range(dim));    yy3=list(range(dim))
    nk=list(range(nmax))
    E_old = Ein;   E = Ein + dE


    # here yy1, yy2 and yy3 represent the three wave functions that the
    #code wants to find, if you want to find more solutions you must add
    #new parameters in the code

    for m in range(nmax):
       print("  ****************   Eigenvalue  #",m+1,"  ***************** ")


       f_old=0

       for k in range(kmax): #iteration for candidate eigenvalue
           #print("  ****************   Iteration (for candidate)  =  ",k," ␣
↪****************** ")
           imatch=0

           for j in range(dim-1):  #classical right turning point
               xx[0]=xl;          xx[dim-1]=xu
               DE1 = E - V(xx[j])
               xx[j+1] = xx[j]+h
               DE2 = E - V(xx[j+1])
               D1D2=DE1*DE2
               if D1D2<=0  and DE1 > 0:  #match point
                  imatch = j+1
                  print(" imatch = ",imatch," xmatch = %2.3f" %(xx[imatch]),"␣
↪V(xmatch) = %2.5f" %(V(xx[imatch])),"  E = %2.4f" %(E))
```

```
            xmatch=xx[imatch]
            ii=range(imatch+2)
            i_lim = ii[2:imatch+2]

            xx[0]=xl;  xx[1]=xl+h   #initial values
            yy[0]=0;   yy[1] = delta
            #vv[0]=0;   vv[1] = delta

            for i in i_lim:                                          #numerov left␣
↪solution
                yy[i]=nrovl(yy[i-2],yy[i-1],xx[i-2],E,h,iflag)
                xx[i]= xx[i-1]+h

            jjj=list(range(dim+1))
            j_lim = list(jjj[imatch-1:dim+1])
            comp_j=len(j_lim)
            jj=sorted(j_lim,key=abs,reverse=True)

            for i in range(dim):
                if i<=imatch+1:
                   yl[i]=yy[i]
                if i>imatch+1:
                   yl[i]=0

            for i in jj:                                             #numerov right␣
↪solution
                if i==dim:
                    yr[dim-1]=0
                if i==(dim-2):
                    yr[dim-2]=2*delta
                if i<(dim-2):
                    yr[i]=nrovr(yr[i+2],yr[i+1],xx[i+2],E,h,iflag)
                    xx[i]= xx[i+1]-h

            for i in range(imatch-1):
                yr[i]=0

            ymatch=yy[imatch]
            yrmatch=yr[imatch]
            ylmatch=yl[imatch]
            #print("  ")
            #print(" imatch =",imatch," xmatch =",xx[imatch]," ylmatch␣
↪=",ylmatch," yrmatch =",yrmatch)

            if ymatch != 0:
                scale=yrmatch/ymatch
```

```
          else:
              continue

          for t in range(imatch+1):    #  y_left
              yy[t] = yy[t]*scale
              yl[t] = -yl[t]*scale
          yl[imatch+1]=-yl[imatch+1]*scale
          #print(" imatch =",imatch," xmatch =",xx[imatch]," ylmatch␣
↪=",yy[imatch]," yrmatch =",yrmatch)
          ymatch=yy[imatch]
          dlmatch=yy[imatch+1]*scale-yy[imatch-1]                  # dif1_left

          t_lim=list(range(dim+1))
          tt=list(t_lim[imatch+1:dim])

          drmatch=yr[imatch+1]-yr[imatch-1]                        # dif1_right

          f =(dlmatch-drmatch)/(2*h)

          for t in tt:                    #  y_right
              yy[t] = yr[t]

          #print(" # iteracoes =",nk[m],"  dl =",dlmatch,"  dr =",drmatch," ␣
↪f_old = %2.1e   f = %2.1e   f2 = %2.1e" %(f_old,f,f2))

          #eps=0.0001

          delta_E=-f*(E-E_old)/(f-f_old)
          #print(" k = %2d   E_old = %2.3f    Eingen = %2.3f   f_old = %2.3e  ␣
↪f = %2.3e  (E_next-Eingen) = %2.3e" %(k,E_old,E,f_old,f,delta_E))

          if abs(delta_E)<eps:  # determination of the root (energy) of f(E)␣
↪by the secant method
             ee[m] = E
             nk[m] = k
             ff[m] = f
             k = kmax
             break
          else:
             f_old=f;  E_old=E;   E = E + 7.2*dE

      if m==0:
                E = E + 70*dE
         for j in range(dim):
             yy1[j]=yy[j]
      if m==1:
         E = E + dE/10
```

```
            for j in range(dim):
                yy2[j]=yy[j]
        if m>=2:
            E = E_old + 50*dE
            for j in range(dim):
                yy3[j]=yy[j]
        print(" k = %2d   E_old = %2.3f    Eingen = %2.3f   f_old = %2.3e   f =␣
 ↪%2.3e   delta_E = %2.3e" %(nk[m],E_old,ee[m],f_old,f,delta_E))
        print()

    return ee, xx, yy1, yy2, yy3

#=========================================================================

a=-1.01; b= 5.01; h=0.006
xl=a; xu=b; D = xu-xl
delta = 0.01; eps = 0.00001
dim=int(D/h); kmax=100; nmax=2
n =0; iflag=0
Rydberg=13.605693122994

x0=xl; y0=0. ; y1=delta; iflag=1
nrovl(y0,y1,x0,0,h,iflag)
nrovr(y0,y1,xu,0,h,iflag)

dE = delta
#Ein = -4
#Ein = -1.6
#Ein = -5.2
Ein = 0.0
Vmax= 16.

print("␣
 ↪=========================================================================")
print("  ")
print("         Potential : V(x) = 16*(1.-exp(-2*x))**2   (Morse L=0)   ")
print("  ")
print("␣
 ↪=========================================================================")

print(" E_in = %2.4f" %(Ein)," dE = ",dE," h = ",h," dim = ",dim)
print("  ")



ee,xx,yy1,yy2,yy3=espectro(xl,xu,h,delta,eps,dim,nmax,kmax,Ein,Vmax,dE,iflag)
```

```
A=1.                        # amplitude normalization in 1 unit
ymax1=vec_max(dim,yy1);    ymax2=vec_max(dim,yy2);    ymax3=vec_max(dim,yy3)

for i in range(dim):        # amplitude normalization
    yy1[i] = yy1[i]/ymax1;  yy2[i] = yy2[i]/ymax2;    yy3[i] = yy3[i]/ymax3

colors =['b','r','g','c','m','y','k'];    #color list (blue, red, green, cyan,␣
 ↪yellow, magenta and black)

plt.figure(figsize=(8, 6), dpi=800)
plt.plot(x,V(x),color=colors[6],linewidth=2)
plt.ylim(-10,20)
plt.xlim(-1,10)
plt.xlabel('x')
plt.ylabel('Effective Potential')
plt.grid()
plt.show()


print(" ")
print(" %60s" %('Eigenfunctions (normalized amplitude)'))
plt.figure(figsize=(8, 6), dpi=800)
plt.plot(xx,yy1,color=colors[0])
plt.plot(xx,yy2,':',color=colors[1])
plt.legend([' y1',' y2',' y3'],prop={"size":10},frameon=False)
plt.ylim(-1,1.1)
plt.xlim(-1,5)
plt.grid()
plt.show()

efic=list(range(nmax));  I=list(range(nmax));   eI=list(range(nmax))

S=A*(xu-xl)                      # Born normalizatiom
l0=0; l1=0; l2=0
N=10000
for i in range(N):               # integral of y*y with the Monte Carlo method
    y = A*random()
    j=np.random.randint(dim)
    yj1=yy1[j]*yy1[j];   yj2=yy2[j]*yy2[j];   yj3=yy3[j]*yy3[j]
    if y <= yj1:
       l0 += 1
    if y <= yj2:
       l1 += 1
    if y <= yj3:
       l2 += 1

efic[0] = float(l0)/N;  efic[1] = float(l1)/N
```

```
I[0] = S*efic[0];       I[1] = S*efic[1];
eI[0] = (S/np.sqrt(N))*np.sqrt(efic[0]*(1-efic[0]))
eI[1] = (S/np.sqrt(N))*np.sqrt(efic[1]*(1-efic[1]))


print("  ")
print("  probability normalization ")
print(" efic1 = %2.3f   I1 = %2.3f   eI1 = %2.3f" %(efic[0],I[0],eI[0]))
print(" efic2 = %2.3f   I2 = %2.3f   eI2 = %2.3f" %(efic[1],I[1],eI[1]))


for i in range(dim):                          # probability normalization
    yy1[i]=yy1[i]/np.sqrt(I[0]); yy2[i]=yy2[i]/np.sqrt(I[1])


ymax1=vec_max(dim,yy1);   ymax2=vec_max(dim,yy2)

S0=ymax1*ymax1*(xu-xl);   S1=ymax2*ymax2*(xu-xl)

l0=0;  l1=0           # checking probability
N=10000
for i in range(N):
    y1 = ymax1*ymax1*random(); y2 = ymax2*ymax2*random()
    j=np.random.randint(dim)
    yj1=yy1[j]*yy1[j];   yj2=yy2[j]*yy2[j]
    if y1<= yj1:
       l0 += 1
    if y2 <= yj2:
       l1 += 1


efic[0] = float(l0)/N;  efic[1] = float(l1)/N
I[0] = S0*efic[0];      I[1] = S1*efic[1]
eI[0] = (S0/np.sqrt(N))*np.sqrt(efic[0]*(1-efic[0]))
eI[1] = (S1/np.sqrt(N))*np.sqrt(efic[1]*(1-efic[1]))


print("  ")
print("  checking probability ")
print(" efic1 = %2.3f   I1 = %2.3f   eI1 = %2.3f" %(efic[0],I[0],eI[0]))
print(" efic2 = %2.3f   I2 = %2.3f   eI2 = %2.3f" %(efic[1],I[1],eI[1]))


et=[-1,-1/4,-1/9];      sig_e=[0.005,0.005,0.005]
```

```
plt.figure(figsize=(8, 6), dpi=800)
plt.plot(xx,yy1,color=colors[0])
plt.plot(xx,yy2,':',color=colors[1])
plt.legend(['y1','y2','y3'],prop={"size":15},frameon=True)
plt.ylim(-2.,2.)
plt.xlim(-1,5)
plt.xlabel('x')
plt.ylabel('Normalized Eigenfunctions')
plt.grid(True)
plt.show()
```

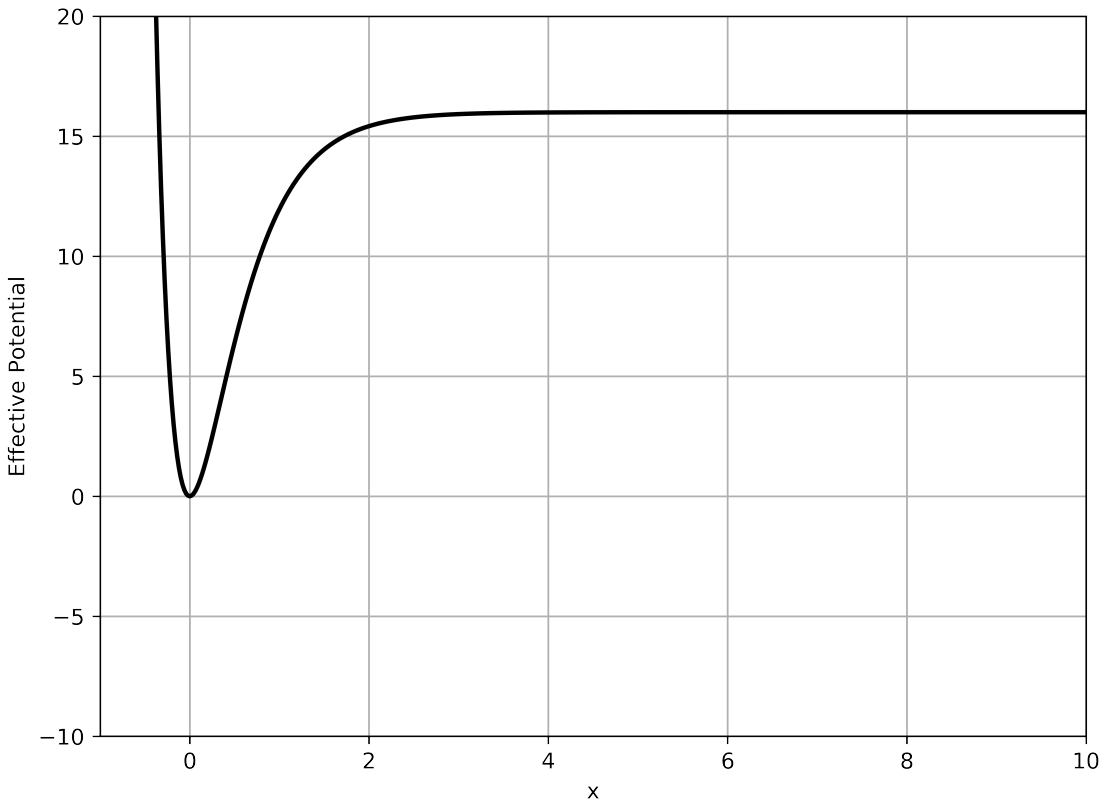
```
================================================================================

        Potential : V(x) = 16*(1.-exp(-2*x))**2   (Morse L=0)

================================================================================
E_in = 0.0000  dE =  0.01  h =  0.006  dim =  1003

 ****************   Eigenvalue  # 1   *****************
imatch =  171  xmatch = 0.016  V(xmatch) = 0.01587  E = 0.0100
imatch =  175  xmatch = 0.040  V(xmatch) = 0.09458  E = 0.0820
imatch =  177  xmatch = 0.052  V(xmatch) = 0.15610  E = 0.1540
imatch =  179  xmatch = 0.064  V(xmatch) = 0.23096  E = 0.2260
imatch =  181  xmatch = 0.076  V(xmatch) = 0.31815  E = 0.2980
imatch =  183  xmatch = 0.088  V(xmatch) = 0.41671  E = 0.3700
imatch =  184  xmatch = 0.094  V(xmatch) = 0.46997  E = 0.4420
imatch =  185  xmatch = 0.100  V(xmatch) = 0.52574  E = 0.5140
imatch =  187  xmatch = 0.112  V(xmatch) = 0.64439  E = 0.5860
imatch =  188  xmatch = 0.118  V(xmatch) = 0.70707  E = 0.6580
imatch =  189  xmatch = 0.124  V(xmatch) = 0.77187  E = 0.7300
imatch =  190  xmatch = 0.130  V(xmatch) = 0.83868  E = 0.8020
imatch =  191  xmatch = 0.136  V(xmatch) = 0.90741  E = 0.8740
imatch =  192  xmatch = 0.142  V(xmatch) = 0.97799  E = 0.9460
imatch =  193  xmatch = 0.148  V(xmatch) = 1.05032  E = 1.0180
imatch =  194  xmatch = 0.154  V(xmatch) = 1.12432  E = 1.0900
imatch =  195  xmatch = 0.160  V(xmatch) = 1.19991  E = 1.1620
imatch =  196  xmatch = 0.166  V(xmatch) = 1.27701  E = 1.2340
imatch =  197  xmatch = 0.172  V(xmatch) = 1.35556  E = 1.3060
imatch =  198  xmatch = 0.178  V(xmatch) = 1.43547  E = 1.3780
imatch =  199  xmatch = 0.184  V(xmatch) = 1.51667  E = 1.4500
imatch =  200  xmatch = 0.190  V(xmatch) = 1.59910  E = 1.5220
imatch =  200  xmatch = 0.190  V(xmatch) = 1.59910  E = 1.5940
imatch =  201  xmatch = 0.196  V(xmatch) = 1.68269  E = 1.6660
imatch =  202  xmatch = 0.202  V(xmatch) = 1.76737  E = 1.7380
imatch =  203  xmatch = 0.208  V(xmatch) = 1.85308  E = 1.8100
imatch =  204  xmatch = 0.214  V(xmatch) = 1.93976  E = 1.8820
imatch =  205  xmatch = 0.220  V(xmatch) = 2.02736  E = 1.9540
```

```
imatch =  205  xmatch = 0.220  V(xmatch) = 2.02736  E = 2.0260
imatch =  206  xmatch = 0.226  V(xmatch) = 2.11581  E = 2.0980
imatch =  207  xmatch = 0.232  V(xmatch) = 2.20506  E = 2.1700
imatch =  208  xmatch = 0.238  V(xmatch) = 2.29506  E = 2.2420
imatch =  209  xmatch = 0.244  V(xmatch) = 2.38575  E = 2.3140
imatch =  210  xmatch = 0.250  V(xmatch) = 2.47709  E = 2.3860
imatch =  210  xmatch = 0.250  V(xmatch) = 2.47709  E = 2.4580
imatch =  211  xmatch = 0.256  V(xmatch) = 2.56902  E = 2.5300
imatch =  212  xmatch = 0.262  V(xmatch) = 2.66150  E = 2.6020
imatch =  213  xmatch = 0.268  V(xmatch) = 2.75449  E = 2.6740
imatch =  213  xmatch = 0.268  V(xmatch) = 2.75449  E = 2.7460
imatch =  214  xmatch = 0.274  V(xmatch) = 2.84793  E = 2.8180
imatch =  215  xmatch = 0.280  V(xmatch) = 2.94179  E = 2.8900
imatch =  216  xmatch = 0.286  V(xmatch) = 3.03602  E = 2.9620
imatch =  216  xmatch = 0.286  V(xmatch) = 3.03602  E = 3.0340
imatch =  217  xmatch = 0.292  V(xmatch) = 3.13059  E = 3.1060
imatch =  218  xmatch = 0.298  V(xmatch) = 3.22545  E = 3.1780
imatch =  219  xmatch = 0.304  V(xmatch) = 3.32058  E = 3.2500
imatch =  220  xmatch = 0.310  V(xmatch) = 3.41593  E = 3.3220
imatch =  220  xmatch = 0.310  V(xmatch) = 3.41593  E = 3.3940
imatch =  221  xmatch = 0.316  V(xmatch) = 3.51146  E = 3.4660
imatch =  222  xmatch = 0.322  V(xmatch) = 3.60715  E = 3.5380
imatch =  223  xmatch = 0.328  V(xmatch) = 3.70296  E = 3.6100
imatch =  223  xmatch = 0.328  V(xmatch) = 3.70296  E = 3.6820
imatch =  224  xmatch = 0.334  V(xmatch) = 3.79887  E = 3.7540
imatch =  225  xmatch = 0.340  V(xmatch) = 3.89483  E = 3.8260
imatch =  226  xmatch = 0.346  V(xmatch) = 3.99082  E = 3.8980
imatch =  226  xmatch = 0.346  V(xmatch) = 3.99082  E = 3.9700
imatch =  227  xmatch = 0.352  V(xmatch) = 4.08682  E = 4.0420
imatch =  228  xmatch = 0.358  V(xmatch) = 4.18279  E = 4.1140
imatch =  229  xmatch = 0.364  V(xmatch) = 4.27871  E = 4.1860
imatch =  229  xmatch = 0.364  V(xmatch) = 4.27871  E = 4.2580
imatch =  230  xmatch = 0.370  V(xmatch) = 4.37456  E = 4.3300
imatch =  231  xmatch = 0.376  V(xmatch) = 4.47030  E = 4.4020
imatch =  232  xmatch = 0.382  V(xmatch) = 4.56592  E = 4.4740
imatch =  232  xmatch = 0.382  V(xmatch) = 4.56592  E = 4.5460
imatch =  233  xmatch = 0.388  V(xmatch) = 4.66140  E = 4.6180
imatch =  234  xmatch = 0.394  V(xmatch) = 4.75670  E = 4.6900
imatch =  235  xmatch = 0.400  V(xmatch) = 4.85182  E = 4.7620
imatch =  235  xmatch = 0.400  V(xmatch) = 4.85182  E = 4.8340
imatch =  236  xmatch = 0.406  V(xmatch) = 4.94672  E = 4.9060
imatch =  237  xmatch = 0.412  V(xmatch) = 5.04140  E = 4.9780
imatch =  238  xmatch = 0.418  V(xmatch) = 5.13583  E = 5.0500
imatch =  238  xmatch = 0.418  V(xmatch) = 5.13583  E = 5.1220
imatch =  239  xmatch = 0.424  V(xmatch) = 5.22999  E = 5.1940
imatch =  240  xmatch = 0.430  V(xmatch) = 5.32387  E = 5.2660
imatch =  241  xmatch = 0.436  V(xmatch) = 5.41745  E = 5.3380
imatch =  241  xmatch = 0.436  V(xmatch) = 5.41745  E = 5.4100
```

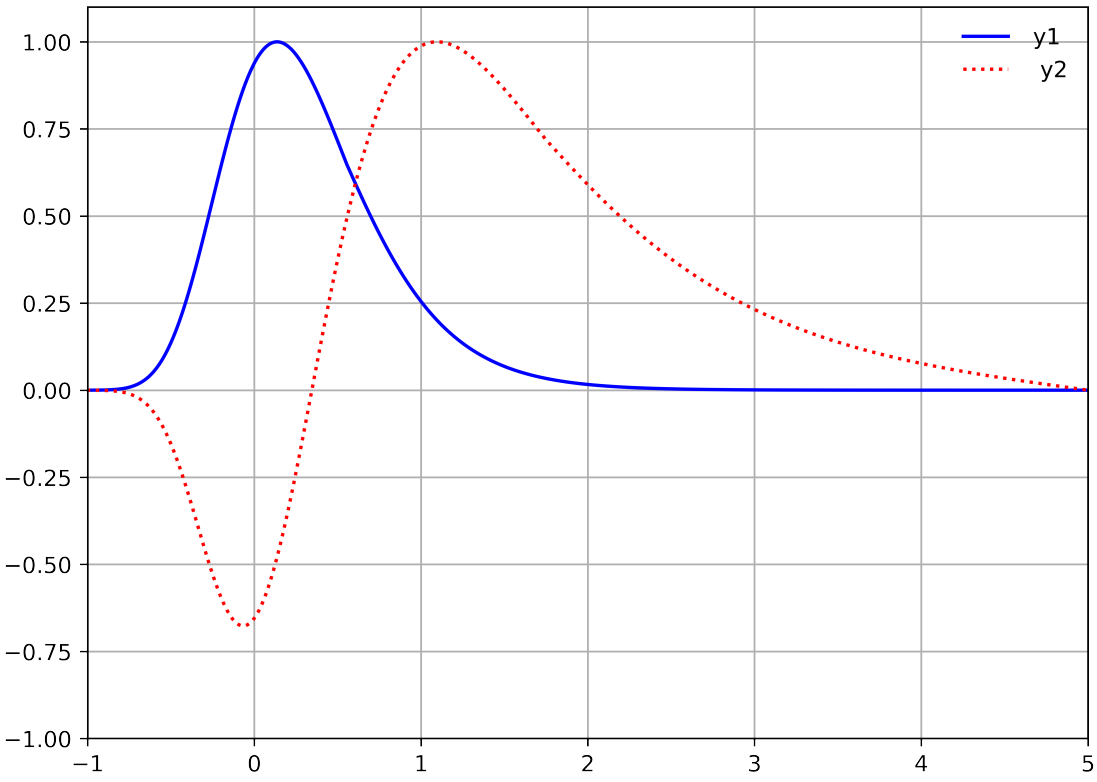
```
 imatch =  242  xmatch = 0.442  V(xmatch) = 5.51072  E = 5.4820
 imatch =  243  xmatch = 0.448  V(xmatch) = 5.60365  E = 5.5540
 imatch =  244  xmatch = 0.454  V(xmatch) = 5.69624  E = 5.6260
 imatch =  245  xmatch = 0.460  V(xmatch) = 5.78847  E = 5.6980
 imatch =  245  xmatch = 0.460  V(xmatch) = 5.78847  E = 5.7700
 imatch =  246  xmatch = 0.466  V(xmatch) = 5.88033  E = 5.8420
 imatch =  247  xmatch = 0.472  V(xmatch) = 5.97180  E = 5.9140
 imatch =  248  xmatch = 0.478  V(xmatch) = 6.06287  E = 5.9860
 imatch =  248  xmatch = 0.478  V(xmatch) = 6.06287  E = 6.0580
 imatch =  249  xmatch = 0.484  V(xmatch) = 6.15354  E = 6.1300
 imatch =  250  xmatch = 0.490  V(xmatch) = 6.24378  E = 6.2020
 imatch =  251  xmatch = 0.496  V(xmatch) = 6.33359  E = 6.2740
 imatch =  252  xmatch = 0.502  V(xmatch) = 6.42296  E = 6.3460
 imatch =  252  xmatch = 0.502  V(xmatch) = 6.42296  E = 6.4180
 imatch =  253  xmatch = 0.508  V(xmatch) = 6.51188  E = 6.4900
 imatch =  254  xmatch = 0.514  V(xmatch) = 6.60034  E = 6.5620
 imatch =  255  xmatch = 0.520  V(xmatch) = 6.68833  E = 6.6340
 imatch =  256  xmatch = 0.526  V(xmatch) = 6.77585  E = 6.7060
 imatch =  257  xmatch = 0.532  V(xmatch) = 6.86288  E = 6.7780
 imatch =  257  xmatch = 0.532  V(xmatch) = 6.86288  E = 6.8500
 imatch =  258  xmatch = 0.538  V(xmatch) = 6.94941  E = 6.9220
 imatch =  259  xmatch = 0.544  V(xmatch) = 7.03545  E = 6.9940
 imatch =  260  xmatch = 0.550  V(xmatch) = 7.12098  E = 7.0660
 imatch =  261  xmatch = 0.556  V(xmatch) = 7.20599  E = 7.1380
 k =  0   E_old = 7.138    Eingen = 0.000   f_old = -3.773e+04   f = -3.773e+04
delta_E = -1.411e-01

  ****************   Eigenvalue  # 2   *****************
 imatch =  270  xmatch = 0.610  V(xmatch) = 7.94721  E = 7.9100
 imatch =  271  xmatch = 0.616  V(xmatch) = 8.02683  E = 7.9820
 imatch =  272  xmatch = 0.622  V(xmatch) = 8.10589  E = 8.0540
 imatch =  273  xmatch = 0.628  V(xmatch) = 8.18438  E = 8.1260
 imatch =  274  xmatch = 0.634  V(xmatch) = 8.26232  E = 8.1980
 imatch =  275  xmatch = 0.640  V(xmatch) = 8.33968  E = 8.2700
 imatch =  276  xmatch = 0.646  V(xmatch) = 8.41648  E = 8.3420
 imatch =  276  xmatch = 0.646  V(xmatch) = 8.41648  E = 8.4140
 imatch =  277  xmatch = 0.652  V(xmatch) = 8.49271  E = 8.4860
 imatch =  278  xmatch = 0.658  V(xmatch) = 8.56836  E = 8.5580
 imatch =  279  xmatch = 0.664  V(xmatch) = 8.64344  E = 8.6300
 imatch =  280  xmatch = 0.670  V(xmatch) = 8.71795  E = 8.7020
 imatch =  281  xmatch = 0.676  V(xmatch) = 8.79188  E = 8.7740
 imatch =  282  xmatch = 0.682  V(xmatch) = 8.86524  E = 8.8460
 imatch =  283  xmatch = 0.688  V(xmatch) = 8.93802  E = 8.9180
 imatch =  284  xmatch = 0.694  V(xmatch) = 9.01023  E = 8.9900
 imatch =  285  xmatch = 0.700  V(xmatch) = 9.08186  E = 9.0620
 imatch =  286  xmatch = 0.706  V(xmatch) = 9.15291  E = 9.1340
 imatch =  287  xmatch = 0.712  V(xmatch) = 9.22339  E = 9.2060
 imatch =  288  xmatch = 0.718  V(xmatch) = 9.29329  E = 9.2780
```

```
imatch =  289  xmatch = 0.724  V(xmatch) = 9.36262  E = 9.3500
imatch =  290  xmatch = 0.730  V(xmatch) = 9.43138  E = 9.4220
imatch =  291  xmatch = 0.736  V(xmatch) = 9.49956  E = 9.4940
imatch =  292  xmatch = 0.742  V(xmatch) = 9.56717  E = 9.5660
imatch =  294  xmatch = 0.754  V(xmatch) = 9.70068  E = 9.6380
imatch =  295  xmatch = 0.760  V(xmatch) = 9.76658  E = 9.7100
imatch =  296  xmatch = 0.766  V(xmatch) = 9.83191  E = 9.7820
imatch =  297  xmatch = 0.772  V(xmatch) = 9.89668  E = 9.8540
imatch =  298  xmatch = 0.778  V(xmatch) = 9.96088  E = 9.9260
imatch =  299  xmatch = 0.784  V(xmatch) = 10.02453  E = 9.9980
imatch =  300  xmatch = 0.790  V(xmatch) = 10.08761  E = 10.0700
imatch =  301  xmatch = 0.796  V(xmatch) = 10.15013  E = 10.1420
imatch =  303  xmatch = 0.808  V(xmatch) = 10.27352  E = 10.2140
imatch =  304  xmatch = 0.814  V(xmatch) = 10.33438  E = 10.2860
imatch =  305  xmatch = 0.820  V(xmatch) = 10.39469  E = 10.3580
imatch =  306  xmatch = 0.826  V(xmatch) = 10.45446  E = 10.4300
imatch =  307  xmatch = 0.832  V(xmatch) = 10.51368  E = 10.5020
imatch =  309  xmatch = 0.844  V(xmatch) = 10.63050  E = 10.5740
imatch =  310  xmatch = 0.850  V(xmatch) = 10.68810  E = 10.6460
imatch =  311  xmatch = 0.856  V(xmatch) = 10.74517  E = 10.7180
imatch =  312  xmatch = 0.862  V(xmatch) = 10.80171  E = 10.7900
imatch =  314  xmatch = 0.874  V(xmatch) = 10.91320  E = 10.8620
imatch =  315  xmatch = 0.880  V(xmatch) = 10.96816  E = 10.9340
imatch =  316  xmatch = 0.886  V(xmatch) = 11.02259  E = 11.0060
imatch =  318  xmatch = 0.898  V(xmatch) = 11.12993  E = 11.0780
imatch =  319  xmatch = 0.904  V(xmatch) = 11.18282  E = 11.1500
imatch =  320  xmatch = 0.910  V(xmatch) = 11.23521  E = 11.2220
imatch =  322  xmatch = 0.922  V(xmatch) = 11.33848  E = 11.2940
imatch =  323  xmatch = 0.928  V(xmatch) = 11.38937  E = 11.3660
imatch =  324  xmatch = 0.934  V(xmatch) = 11.43976  E = 11.4380
imatch =  326  xmatch = 0.946  V(xmatch) = 11.53907  E = 11.5100
imatch =  327  xmatch = 0.952  V(xmatch) = 11.58799  E = 11.5820
imatch =  329  xmatch = 0.964  V(xmatch) = 11.68440  E = 11.6540
imatch =  330  xmatch = 0.970  V(xmatch) = 11.73189  E = 11.7260
imatch =  332  xmatch = 0.982  V(xmatch) = 11.82545  E = 11.7980
imatch =  333  xmatch = 0.988  V(xmatch) = 11.87154  E = 11.8700
imatch =  335  xmatch = 1.000  V(xmatch) = 11.96232  E = 11.9420
imatch =  337  xmatch = 1.012  V(xmatch) = 12.05129  E = 12.0140
imatch =  338  xmatch = 1.018  V(xmatch) = 12.09510  E = 12.0860
imatch =  340  xmatch = 1.030  V(xmatch) = 12.18139  E = 12.1580
imatch =  342  xmatch = 1.042  V(xmatch) = 12.26592  E = 12.2300
imatch =  343  xmatch = 1.048  V(xmatch) = 12.30754  E = 12.3020
imatch =  345  xmatch = 1.060  V(xmatch) = 12.38951  E = 12.3740
imatch =  347  xmatch = 1.072  V(xmatch) = 12.46979  E = 12.4460
imatch =  349  xmatch = 1.084  V(xmatch) = 12.54842  E = 12.5180
imatch =  351  xmatch = 1.096  V(xmatch) = 12.62543  E = 12.5900
imatch =  352  xmatch = 1.102  V(xmatch) = 12.66333  E = 12.6620
imatch =  354  xmatch = 1.114  V(xmatch) = 12.73794  E = 12.7340
```

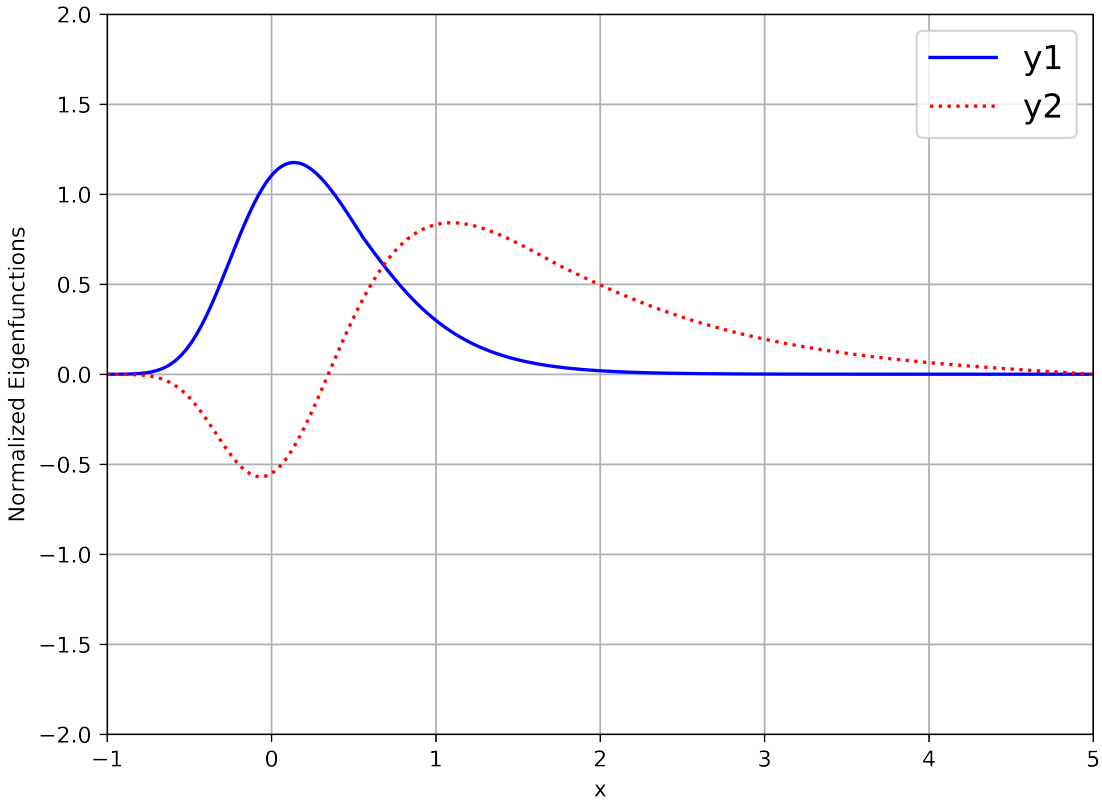
```
 imatch =  356  xmatch = 1.126  V(xmatch) = 12.81100  E = 12.8060
 imatch =  358  xmatch = 1.138  V(xmatch) = 12.88252  E = 12.8780
 imatch =  360  xmatch = 1.150  V(xmatch) = 12.95255  E = 12.9500
 imatch =  363  xmatch = 1.168  V(xmatch) = 13.05482  E = 13.0220
 imatch =  365  xmatch = 1.180  V(xmatch) = 13.12120  E = 13.0940
 imatch =  367  xmatch = 1.192  V(xmatch) = 13.18616  E = 13.1660
 imatch =  369  xmatch = 1.204  V(xmatch) = 13.24974  E = 13.2380
 imatch =  371  xmatch = 1.216  V(xmatch) = 13.31196  E = 13.3100
 imatch =  374  xmatch = 1.234  V(xmatch) = 13.40280  E = 13.3820
 imatch =  376  xmatch = 1.246  V(xmatch) = 13.46173  E = 13.4540
 imatch =  379  xmatch = 1.264  V(xmatch) = 13.54774  E = 13.5260
 imatch =  381  xmatch = 1.276  V(xmatch) = 13.60354  E = 13.5980
 imatch =  384  xmatch = 1.294  V(xmatch) = 13.68496  E = 13.6700
 imatch =  387  xmatch = 1.312  V(xmatch) = 13.76374  E = 13.7420
 imatch =  389  xmatch = 1.324  V(xmatch) = 13.81482  E = 13.8140
 imatch =  392  xmatch = 1.342  V(xmatch) = 13.88935  E = 13.8860
 imatch =  395  xmatch = 1.360  V(xmatch) = 13.96144  E = 13.9580
 imatch =  398  xmatch = 1.378  V(xmatch) = 14.03115  E = 14.0300
 imatch =  402  xmatch = 1.402  V(xmatch) = 14.12054  E = 14.1020
 imatch =  405  xmatch = 1.420  V(xmatch) = 14.18500  E = 14.1740
 imatch =  408  xmatch = 1.438  V(xmatch) = 14.24731  E = 14.2460
 imatch =  412  xmatch = 1.462  V(xmatch) = 14.32718  E = 14.3180
 imatch =  416  xmatch = 1.486  V(xmatch) = 14.40352  E = 14.3900
 imatch =  420  xmatch = 1.510  V(xmatch) = 14.47647  E = 14.4620
 imatch =  424  xmatch = 1.534  V(xmatch) = 14.54617  E = 14.5340
 imatch =  428  xmatch = 1.558  V(xmatch) = 14.61276  E = 14.6060
 imatch =  433  xmatch = 1.588  V(xmatch) = 14.69182  E = 14.6780
 imatch =  437  xmatch = 1.612  V(xmatch) = 14.75188  E = 14.7500
 imatch =  442  xmatch = 1.642  V(xmatch) = 14.82318  E = 14.8220
 imatch =  448  xmatch = 1.678  V(xmatch) = 14.90348  E = 14.8940
 imatch =  453  xmatch = 1.708  V(xmatch) = 14.96627  E = 14.9660
 imatch =  460  xmatch = 1.750  V(xmatch) = 15.04827  E = 15.0380
 k =  1   E_old = 15.038    Eingen = 1.000   f_old = -3.599e+00   f = -3.599e+00
delta_E = -3.635e-02
```



Eigenfunctions (normalized amplitude)




```
 probability normalization
efic1 = 0.120   I1 = 0.723   eI1 = 0.020
efic2 = 0.235   I2 = 1.413   eI2 = 0.026

 checking probability
efic1 = 0.121   I1 = 1.011   eI1 = 0.027
efic2 = 0.239   I2 = 1.017   eI2 = 0.018
```


y1
y2
Normalized Eigenfunctions
x
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−1
0
1
2
3
4
5

[ ]: