University of Iceland

Solid State Physics

Fall 2014

Numerical estimation of the energy spectrum and probability density of an electron in a periodic potential

Author: Róbert Karl Lárusson

Introduction and theory

Our aim is to numerically estimate the energy spectrum of a atom for a specific **one dimensional** potential $V(x)$ with a period a. We will use *Bloch waves* as the solutions ψ_{nk} to the Shrödinger equation. Bloch waves are a product of a plane wave e^{ikx} and a periodic function u_{nk} of same period a (in this case representing the *lattice constant*), meaning it has the property $u_{nk}(x) = u_{nk}(x+a)$. We will be representing u_{nk} as a linear combination of planewaves,

$$u_{nk} = \sum_{K'} C_{kK'} e^{-iK'x} \tag{1}$$

taking the product of equation 1 and e^{ikx} yields,

$$\psi(x)_{n,k} = \sum_{K'} C_{kK'} e^{i(k-K')x} \quad \text{where} \quad K' = \frac{2\pi}{a} n' \quad \text{where} \quad n' \in \mathbb{Z} \tag{2}$$

The (one-dimensional) time-independant Shrödinger equation,

$$E\psi(x) = -\frac{\hbar^2}{2m}\frac{d^2}{dx^2}\psi(x) + V(x)\psi(x) \quad \text{or} \quad E\psi(x) = \hat{H}\psi(x) \tag{3}$$

Where $\hat{H}$ is the Hamiltonian operator

$$\hat{H} = -\frac{\hbar^2}{2m}\frac{d^2}{dx^2} + V(x) = \hat{H}_0 + V(x) \tag{4}$$

Thus in dirac notation, we can write the energy of the free particle ($V(x) = 0$) as $E_{n,k}$

$$\hat{H}_0 \left|n,k\right\rangle = E_{n,k} \left|n,k\right\rangle \quad \text{where} \quad E_{n,k} = \frac{\hbar^2}{2m^*}k^2 \tag{5}$$

and let us represent equation 3 as the following eigenvalue problem,

$$\hat{H}|n,k) = \varepsilon_{n,k}|n,k) \tag{6}$$

To get $\hat{H}$'s energy spectrum, we take the inner product of equation 6 with $|n,k-K\rangle$ and get,

$$\langle n,k-K|\,\hat{H}|n,k) = \mathbf{H}\mathbf{C}_k = \sum_{K'} C_{kK'}\left(E_{n,k-K'}\delta_{KK'} + \left\langle n,k-K \,\middle|\, V \,\middle|\, n,k-K'\right\rangle\right) = \varepsilon_k \mathbf{C}_k \tag{7}$$

where $|n,k)$ and $|n,k\rangle$ are related by, $|n,k) = \sum_{K'} C_{kK'}\left|n,k-K'\right\rangle$. From equation 7 we start with,

$$E_{n,k-K'}\delta_{KK'} = \frac{\hbar^2}{2m^*}(k-K')^2 = \frac{\hbar^2}{2m^*}\left(k - \frac{2\pi}{a}n'\right)^2 \delta_{KK'} \tag{8}$$

We will assume our potential is $V(x) = \pm\sin^2\left(\frac{\pi x}{a}\right)$ so the latter part of equation 7 is,

$$\left\langle n,k-K \,\middle|\, V \,\middle|\, n,k-K'\right\rangle = \int_0^a e^{-i(k-K')x} V(x) e^{i(k-K')x} dx = \pm\int_0^a V_0 \sin^2\left(\frac{\pi x}{a}\right) e^{i(K-K')x} dx \tag{9}$$

We need dimensionless variables, looking at equations 8 and 9, we redefine the following,

$$\frac{\hbar^2}{2m^*} \to 1 \qquad a \to 1 \qquad k^* \to \frac{2\pi}{a}k \qquad x^* \to \frac{x}{a} \tag{10}$$

So, equation 8 then becomes,

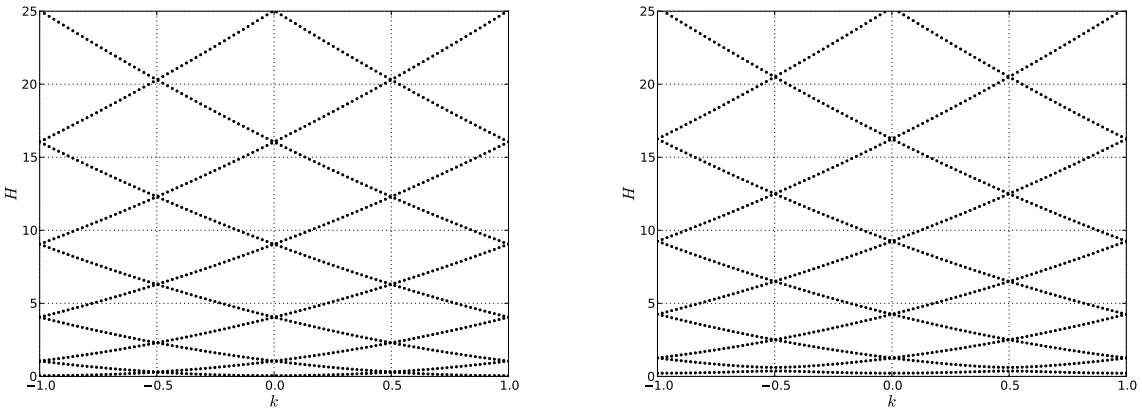
$$E_{n,k-K'}\delta_{KK'} = \left(k^* - n'\right)^2 \delta_{KK'} \tag{11}$$

And equation 9 becomes,

$$\left\langle n, k-K \,\middle|\, V \,\middle|\, n, k-K' \right\rangle = \pm V_0 \int_0^1 \sin^2(\pi x^*) e^{i(K-K')x^*} dx \tag{12}$$

It's clear that the kinetic part of the energy is only non-zero at the diagonal elements of the matrix (where $K = K'$). We write a Python script to numerically evaluate the integral in equation 12 and then we construct the Hamiltonian matrix, find its eigen- values & vectors. We use the eigenvalues to obtain a energy spectrum and the eigenvectors to calculate the eigenfunctions (equation 1).

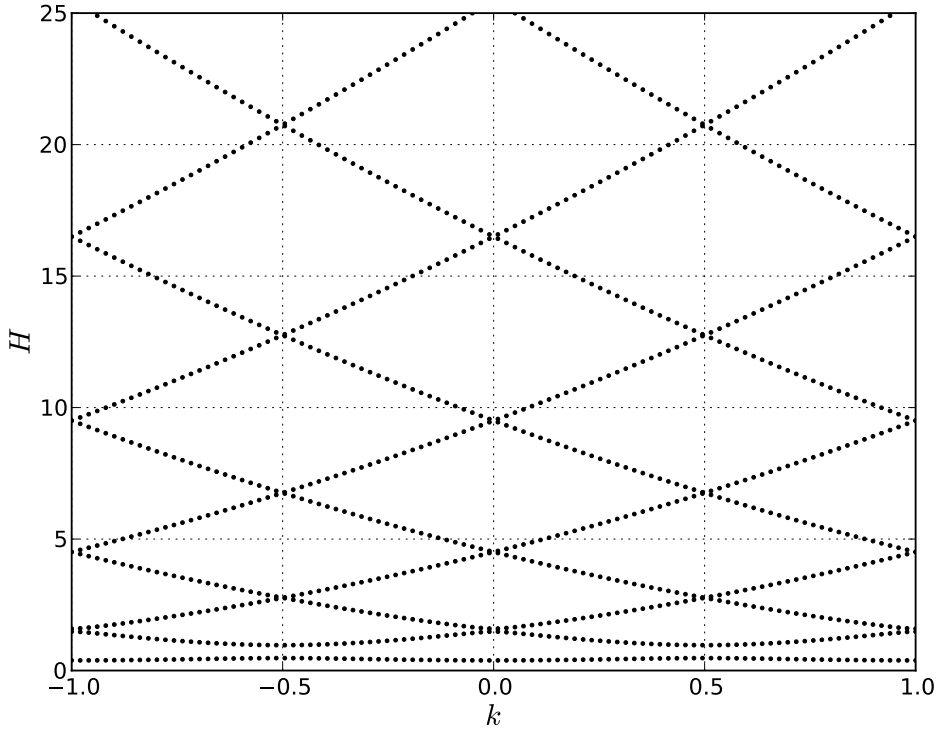
Results



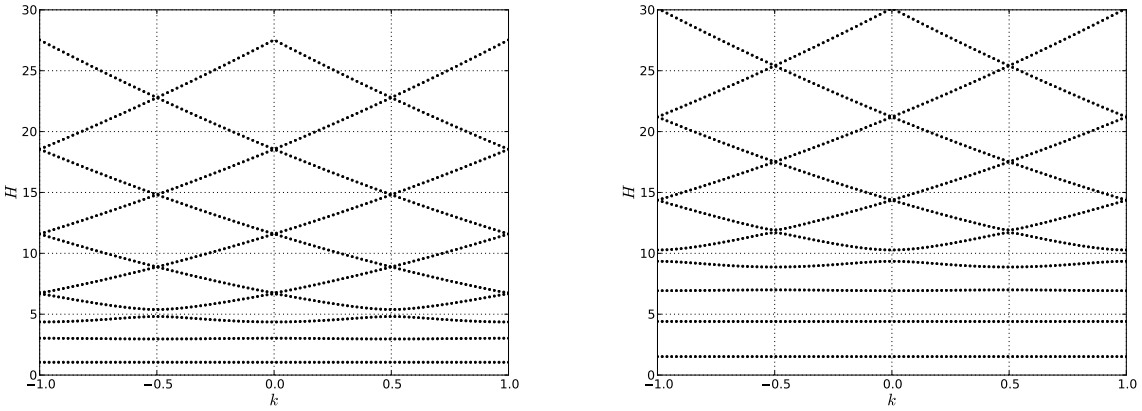
(a) The Energy spectrum for $V_0/E = 0.01$ (b) $V_0/E = 0.5$

Mynd 1: The energy spectrum for an electron in this periodic potential for two values of $V_0 < 1$

From comparing figures 1 and 2 that the energy gap between levels increases as the potential V_0 increases.



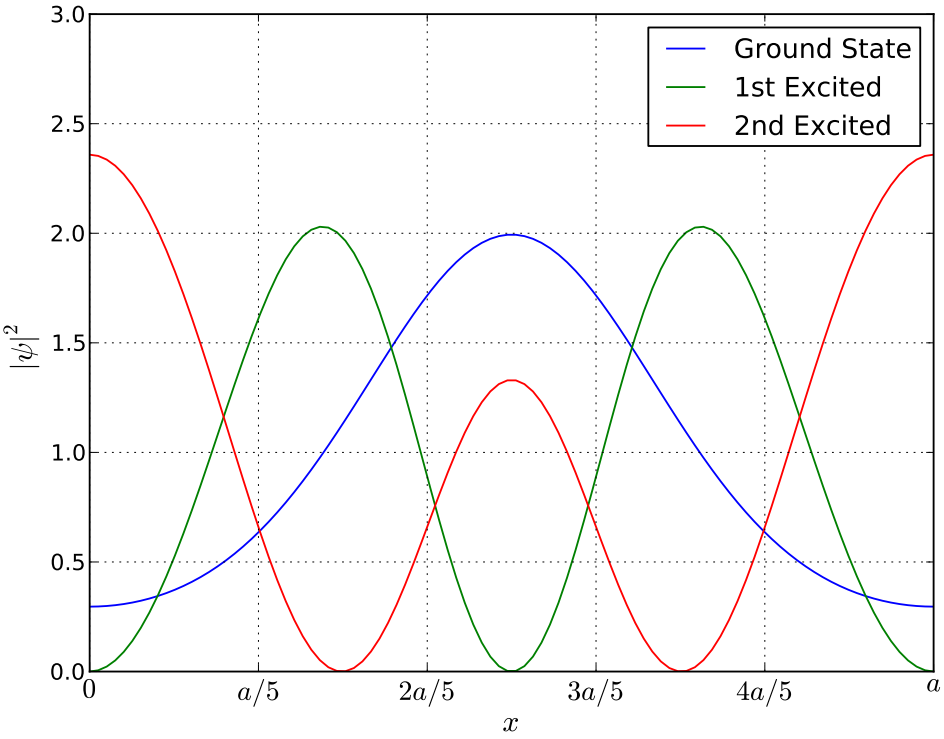
Mynd 2: Here we have the spectrum for $V_0/E = 1$



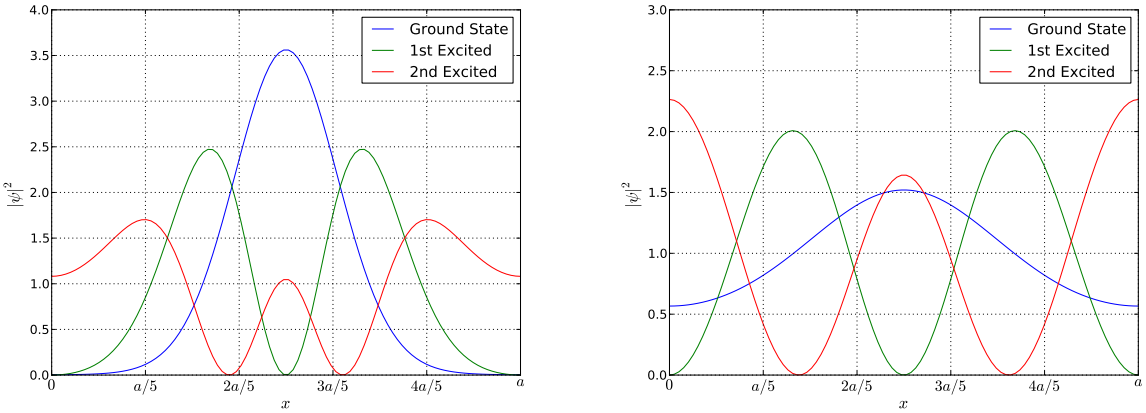
(a) The Energy spectrum for $V_0/E = 5.0$

(b) $V_0/E = 10.0$

Mynd 3



Mynd 4: The absolute value of the probability density squared for $V_0/E = 1.0$



(a) The Energy spectrum for $V_0/E = 5.0$ (b) $V_0/E = 0.5$

Mynd 5: Here we see the probabilty density for different values of V_0

Source code

```
import numpy as np
from scipy.integrate import quad
from matplotlib import pyplot as plt

#DEFINE PARAMETERS
N = input('Pick the dimension of your basis, N = ')
V0 = input('Pick the amplitude of the periodic potential, V_0 = ')
a = 1.0
kmaxDa=1.0
vidd = range(1,int(N+1)) #To map elements to array/matrix
stepsize=100

# FUNCTIONS THAT CALCULATES AN INTEGRAL AND STUFF
def integrFunc(x,k1,k2):  # integrFunc evaluates a function that we numerically evaluate.
    return (V0*(np.sin(np.pi*x)**2)*np.exp(complex(0,x*(k1-k2))).real)

def getEkin(k): # Returns expected kinetic energy (for a given value k (N values on a matrix diagonal))
    Ekin=[]
    for k2 in vidd:
        Ekin.append((2.0*np.pi*(k-(k2-N/2.0))**2))
    return np.diag(Ekin)

def getEpot(): # Returns the expected potential energy for a N by N basis
    Epot=[]
    for k2 in vidd:
        EpotCol=[]
        for k1 in vidd:
            EpotCol.append(-1*quad(integrFunc,0,a,args=(k1,k2))[0]) #Calculate the potential Energy
        Epot.append(EpotCol)
    return Epot

def getHmat(k): # Returns a matrix representing the Hamiltonian (= getEkin + getEpot)
    Hmat=[]
    Kinetic = getEkin(k)
    for v in vidd:
        Hmat.append(map(sum, zip(Potential[v-1],Kinetic[v-1])))
    return Hmat

def getEigs(matrix): # Takes in a square matrix and returns its eigenvalues & eigenvectors.
    return np.linalg.eigh(matrix)

def getWavefunction(x,k,state): # Returns absolut value of the probabilty density squared for a given energystate, position x and wave vector k.
    probDens = 0.0
    for n in vidd:
        probDens = probDens + EigVec[n-1,state]*np.exp(complex(0,2.0*np.pi*(k-N/2.0-float(n))*x))
    return abs(probDens)**2

def makeMatrix(N): #Defines an array that contains N empty arrays.
```

```
49      Fylki = []
50      for incr in range(0,int(N)):
51          Fylki.append([])
52      return Fylki
53
54  Potential = getEpot()
55  kHnit = np.linspace(-kmaxDa,kmaxDa,stepsize)
56  yHnit = makeMatrix(N)
57  for k in kHnit:
58      EigVal,EigVec = getEigs(getHmat(k))
59      for g in vidd:
60          yHnit[g-1].append(EigVal[g-1])
61
62  xHnit = np.linspace(0,1,stepsize)
63  states=range(0,3)
64  for state in states:
65      psi=[]
66      for b in range(0,len(xHnit)):
67          psi.append(getWavefunction(xHnit[b],100.0,state))
68      plt.plot(xHnit,psi)
69
70  #CREATE & DEFINE THE PARAMETERS OF THE PLOT
71  for incr in range(0,5):
72      plt.plot(kHnit,yHnit[incr],linestyle='', markersize=3,marker='o',color='
        black')                                 #Create the plot
73  plt.ylabel('Energy $H_{nk}$',fontsize=16)                                    #
    Axis labels created
74  plt.xlabel('Wave number $k$',fontsize=16)
75  plt.ylim(0,35.0)
76  plt.xlim(-kmaxDa,kmaxDa)
77  plt.show()
```