
Free_electrons_in_Na

Unknown Author

February 25, 2014

The following document is available for consultation/download/modification at

http://nbviewer.ipynb.org/github/dombro/CMI/blob/master/notebooks/Free_electrons_in_Na.ipynb?create=1

1 *Using the shape of the computed band structure of Na, please show what the empty BCC lattice band structure may look like (assuming isotropic effective mass m_0)*

The empty BCC lattice is obtained when setting $V \equiv 0$. The effect of the weak finite potential V is to lift the accidental band degeneracies at $k = 0$ and at the zone boundaries (high symmetry points) in the Brillouin zone. Such splittings vary for the various bands and at different high symmetry points. In order to obtain the empty lattice structure from the real structure, one needs to merge the bands at the places where the effect of the finite value of V is to split them. The final result remains close to the computed band structure, because the nearly-free electron approximation is quite justified in the case of Na.

1.1 General expression of energy

For a 3D crystal, the free electron energies and wave functions can be expressed in the following way:

Decompose the plane wave vector as a sum of Bloch wave vector and reciprocal lattice vector, where the Bloch wave vector k is restricted to the first Brillouin zone. The reciprocal lattice vectors are given by $K_l = l_1 b_1 + l_2 b_2 + l_3 b_3$ where (l_1, l_2, l_3) are integers and b_i are primitive translations of the reciprocal lattice. The plane wave function is thus given by $\Psi_l(k, r) = e^{ik \cdot r} e^{iK_l \cdot r}$, where the second factor has the periodicity of the lattice since $e^{iK_l \cdot R} = 1$ for any vector R of the direct lattice, i.e. this form respects Bloch's theorem.

The energy of such plane wave function is then given by $E_l(k) = \frac{\hbar^2}{2m}(k + K_l)^2$. It follows that each band is labelled by $l \equiv (l_1, l_2, l_3)$, with the corresponding wave function and energy expression above.

1.2 The Body Centered Cubic lattice

(for a graphic representation of the reciprocal bcc lattice, check **Dresselhaus, Fig. C.5.b, p 511**).

The primitive translations of the reciprocal lattice are: $b_1 = \frac{2\pi}{a}(x + z)$, $b_2 = \frac{2\pi}{a}(-x + y)$, $b_3 = \frac{2\pi}{a}(-y + z)$.

A general reciprocal lattice vector is thus given by $G_{h_1 h_2 h_3} = \frac{2\pi}{a}[(h_1 - h_2)x + (h_2 - h_3)y + (h_1 + h_3)z]$.

The 12 shortest reciprocal lattice vectors are $\pm \frac{2\pi}{a}(x \pm y)$, $\pm \frac{2\pi}{a}(y \pm z)$, $\pm \frac{2\pi}{a}(x \pm z)$

The empty lattice energy bands can be written as $E_{l_1 l_2 l_3} = \frac{2\pi^2 \hbar^2}{ma^2} [(l_1 + \xi)^2 + (l_2 + \eta)^2 + (l_3 + \zeta)^2]$ with $l_1 = h_1 - h_2$, $l_2 = h_2 - h_3$, $l_3 = h_1 + h_3$ and h_i an integer, and $k = \frac{2\pi}{a}(\xi, \eta, \zeta)$ in the first Brillouin zone.

The special points have the following coordinates:

$$H = \frac{2\pi}{a}(1, 0, 0), P = \frac{2\pi}{a}(1/2, 1/2, 1/2), N = \frac{2\pi}{a}(1/2, 1/2, 0)$$

With this, the energy levels at the special points may be written as:

$$E_{h_1 h_2 h_3}(\Gamma) = (h_1 - h_2)^2 + (h_2 - h_3)^2 + (h_1 + h_3)^2$$

$$E_{h_1 h_2 h_3}(H) = (h_1 - h_2 + 1)^2 + (h_2 - h_3)^2 + (h_1 + h_3)^2$$

$$E_{h_1 h_2 h_3}(P) = (h_1 - h_2 + \frac{1}{2})^2 + (h_2 - h_3 + \frac{1}{2})^2 + (h_1 + h_3 + \frac{1}{2})^2$$

$$E_{h_1 h_2 h_3}(N) = (h_1 - h_2 + \frac{1}{2})^2 + (h_2 - h_3 + \frac{1}{2})^2 + (h_1 + h_3)^2$$

These expressions allow us to determine the authorized values of the energy levels at all special points, as well as determine their degeneracy. Such points should then be joined by parabolic branches in order to build the empty lattice band structure. For a detailed look at the result, please check **Dresselhaus, Fig.12.6.a., p293**.

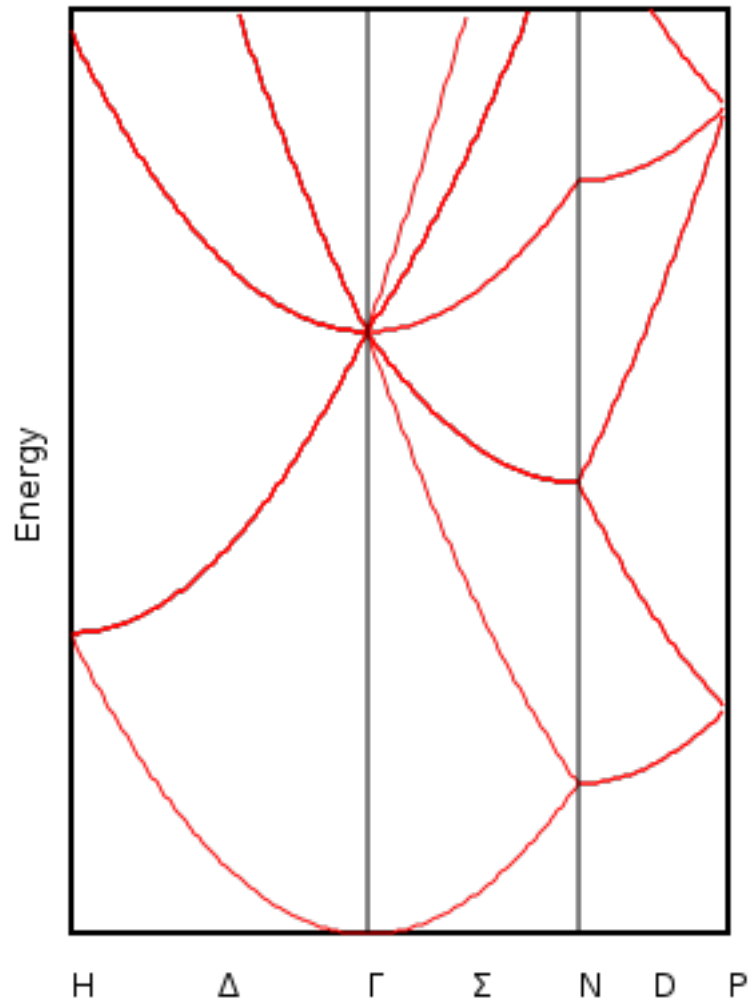
Refs:

Dresselhaus, Chapter 12

Pengcheng Dai, Lecture notes, Physics 671, University of Tennessee, Knoxville

```
In [3]: from IPython.display import Image
        i = Image(filename='empty_BCC_bands.png')
        i
```

Out [3]:



1.3 Group of a vector k

The group of symmetry operations which leave the lattice invariant also leaves the reciprocal lattice invariant. Suppose we know some wave function Ψ_k . A rotation or reflection operation of the point group acting on Ψ_k will give the same result as the rotation or reflection of k itself, that is $R\Psi_k(x) = \Psi_{Rk}(x) = \Psi_k(R^{-1}x)$. Here we used the fact that applying the same orthogonal transformation to both vectors in a scalar product does not change the value of the product, for example $k \cdot R^{-1}r = Rk \cdot RR^{-1}r = Rk \cdot r$. By applying every symmetry of the group to a wave vector k , we generate the star of k . For the BCC lattice, all operations leave Γ invariant so Γ is its own star. For a general point k , there will be g (where g is the order of the group of the crystal) points in the star of k .

The group (or subgroup of the original point group) of rotations and reflections that transform k into itself or into a new k vector separated from the original k point by a reciprocal lattice vector, are said to belong to the group of the wave vector k .

When the empty lattice bands are calculated, there are often a number of degenerate bands at points of high symmetry like the Γ point, or the X point. Because the operations of the group of the wave vector Γ (or X) leave this point invariant, one can construct linear combinations of these degenerate states that belong to representations of the group of the wave vector Γ (or X etc.). Once these linear combinations are built and the corresponding representation identified, one can infer what degeneracies will be lifted by the introduction of a finite crystal potential.

1.4 Lifting degeneracies: finite potential

Th. on matrix elements: Matrix elements of any operator which is invariant under all the operations of a group are zero between functions belonging to different IR's of the group. Matrix elements are also zero between functions belonging to different rows of the same representation.

As a consequence, if the degenerate states are classified according to the IR's of the group of the wave vector, many of the degenerate states will belong to different IR's and therefore the off-diagonal matrix elements of $V(r)$ between them will vanish. The matrix of $V(r)$ will acquire a block form, with the dimension of each block giving the new degeneracy level of the split energy levels. This lets one give a particularly simple form to the secular equation, which allows an easier computation of the energy levels.

The IR to which a direction of the crystal belongs also determines which lines can or cannot cross in the band structure profile: If they belong to the same IR, they are coupled by the potential, and therefore cannot cross. If they belong to different IRs, they may cross. Note such an interesting occurrence in **Fig. 12.6. of Dresselhaus, 293**, where we clearly see the two $F1$ lines repelling each other, while F_1 and F_3 have no problem crossing each other.

Following the previous calculation, it is in this way possible to identify to which IR each wave function belongs, and thus to name each point and branch, as is done in the complete computed band structure by Ching and Callaway.

2 In the above mentioned model, how high above the bottom of the conduction band does the Fermi energy lie?

For the alkali metals, the nearly-free electron approximation is valid. This leads to a nearly spherical Fermi surface, and small band gaps. The crystal structure for the alkali metals is body centered cubic (BCC).

In the figure representing the computed band structure of Na, the lowest band is the 3s conduction band. The filled valence bands lie much lower in energy and are not represented. For the case of Na, the 3s conduction band is very nearly free electron-like and the dispersion relations are closely isotropic. The isotropic effective mass model with parameter m_0 is therefore a very good approximation.

Since Na has a the $[\text{Ne}]3s^1$ electronic structure, the Fermi level is determined so that the 3s band is exactly half-occupied. Therefore the radius of the Fermi surface k_F satisfies the relation

$$\frac{4\pi k_F^3}{3} = \frac{1}{2} * 2 * \left(\frac{2\pi}{a}\right)^3 \Rightarrow k_F = \left(\frac{3}{4\pi}\right)^{1/3} \left(\frac{2\pi}{a}\right) \simeq 0.62 \left(\frac{2\pi}{a}\right)$$

The shortest distance from Γ to a Brillouin zone face is $\Gamma N = \left(\frac{2\pi}{a}\right) \sqrt{(1/2)^2 + (1/2)^2 + 0^2} \simeq 0.71 \left(\frac{2\pi}{a}\right)$

This indicates that the Fermi sphere remains a safe distance away from the zone boundary, where only narrow band gaps distort the otherwise quasi free-electron like dispersion. It is therefore quite a safe approximation to consider a quadratic dispersion with effective mass m_0 for all electrons of Na, and a spherical Fermi surface. This is also visible in the position of the E_F line in the computed band structure, which lies safely below the lowest lying band gap at the N point.

The energy E_F thus verifies $E_F \simeq \frac{\hbar^2 k_F^2}{2m_0} = \frac{\hbar^2 (3/4\pi)^{2/3}}{2m_0} \left(\frac{2\pi}{a}\right)^2$

Using:

1. $a = 0.423 \times 10^{-9} \text{m}$
2. $m_0 = 0.97 \cdot m_e = 0.97 \cdot 9.1 \times 10^{-31} \text{kg}$
3. $\hbar = 1.05 \times 10^{-34} \text{m}^2 \cdot \text{kg/s}$

we get $E_F \simeq 5.2 \times 10^{-19} \text{J} = 3.2 \text{eV} = 0.24 \text{Ry}$

Ref:

1. Ashcroft Mermin, Ch. 15
2. Pastori-Paravicini, p. 223 for effective mass.

3 Justify why Na is called “the simplest of simple metals”

As illustrated above, the various effects arising from the periodic potential of the lattice materialize in the most simple manner in Na:

1. The effective mass is isotropic
2. The effective mass is very close to the rest mass of a free electron
3. The Fermi sphere does not get close to the band gaps.
4. The width of the band gaps is small compared to bandwidth
5. For all practical purposes, the dispersion relation of the electrons is therefore identical to the free electron dispersion relation

In other terms, it is not possible to find a real material which deviates less from the free electron gas case, than does Na. As a consequence, it is the most simple to model.

4 Result of analytical computation of bands:

$$E_{h_1 h_2 h_3}(\Gamma) = (h_1 - h_2)^2 + (h_2 - h_3)^2 + (h_1 + h_3)^2$$

$$E_{h_1 h_2 h_3}(H) = (h_1 - h_2 + 1)^2 + (h_2 - h_3)^2 + (h_1 + h_3)^2$$

$$E_{h_1 h_2 h_3}(P) = (h_1 - h_2 + \frac{1}{2})^2 + (h_2 - h_3 + \frac{1}{2})^2 + (h_1 + h_3 + \frac{1}{2})^2$$

$$E_{h_1 h_2 h_3}(N) = (h_1 - h_2 + \frac{1}{2})^2 + (h_2 - h_3 + \frac{1}{2})^2 + (h_1 + h_3)^2$$

```
In [47]: %matplotlib inline
import matplotlib.pyplot as plt
#from pylab import figure, xticks, yticks
#import pylab
from IPython.display import display, Math, Latex

special_points = ["G", "H", "P", "N"]

def get_energy(point, coords):
    if point == "G":
        return (coords[0]-coords[1])**2+(coords[1]-coords[2])**2+(coords[0]+coords[2])**2
    elif point == "H":
        return (coords[0]-coords[1]+1.0)**2+(coords[1]-coords[2])**2+(coords[0]+coords[2])**2
    elif point == "P":
        return (coords[0]-coords[1]+0.5)**2+(coords[1]-coords[2]+0.5)**2+(coords[0]+coords[2])**2
    elif point == "N":
        return (coords[0]-coords[1]+0.5)**2+(coords[1]-coords[2]+0.5)**2+(coords[0]+coords[2])**2
    else:
        print "ERROR - invalid point requested: ", point

#recursive backtracking in order to get the valid reciprocal lattice points
def get_next(cand):
    out = [elem for elem in cand]
    if out[2]==3:
        if out[1]==3:
```

```

        if out[0]==3:
            return None
        else:
            out[0]+=1
            out[1]=-3
            out[2]=-3
    else:
        out[1]+=1
        out[2]=-3
else:
    out[2]+=1
return out

h_list=[[-3,-3,-3]]
cand = get_next(h_list[-1])
while cand != None:
    h_list.append([elem for elem in cand])
    cand = get_next(h_list[-1])

all_energies = {}
all_occurences = {}

#build the dictionnaries containing possible energy levels
#and their degeneracy
for point in special_points:
    all_energies[point]=[]
    for elem in h_list:
        tt = get_energy(point, elem)
        all_energies[point].append(tt)
        if (point, tt) in all_occurences:
            all_occurences[(point, tt)].append(elem)
        else:
            all_occurences[(point, tt)] = [elem]

#display(Math(r'F(k) = \int_{-\infty}^{\infty} f(x) e^{2\pi i k} dx'))
print "\n"
print "\n"
display(Latex(r'Energy levels per special point, in units of  $(\hbar^2/2m)(2\pi/a)^2$ '))
for point in special_points:
    print "point: ", point
    valid_energies = list(set(all_energies[point]))
    valid_energies.sort()
    for e in valid_energies:
        if e<=9:
            print "E = ", e, "; degeneracy: ", all_energies[point].count(e)

print "\n"

print "A few examples of degeneracy (h1, h2, h3 triplets associated with a special poi"
print "point N, E=1.5: ", all_occurences[("N", 1.5)]
print "point G, E=2: ", all_occurences[("G", 2)]
print "\n"

```

Energy levels per special point, in units of $(\hbar^2/2m)(2\pi/a)^2$

```

point:  G
E =  0 ; degeneracy:  1
E =  2 ; degeneracy:  12
E =  4 ; degeneracy:  6
E =  6 ; degeneracy:  24
E =  8 ; degeneracy:  12
point:  H
E =  1.0 ; degeneracy:  6
E =  3.0 ; degeneracy:  8

```

```

E = 5.0 ; degeneracy: 24
E = 9.0 ; degeneracy: 30
point: P
E = 0.75 ; degeneracy: 4
E = 2.75 ; degeneracy: 12
E = 4.75 ; degeneracy: 12
E = 6.75 ; degeneracy: 16
E = 8.75 ; degeneracy: 24
point: N
E = 0.5 ; degeneracy: 2
E = 1.5 ; degeneracy: 4
E = 2.5 ; degeneracy: 4
E = 3.5 ; degeneracy: 8
E = 4.5 ; degeneracy: 6
E = 5.5 ; degeneracy: 4
E = 6.5 ; degeneracy: 12
E = 7.5 ; degeneracy: 8
E = 8.5 ; degeneracy: 8

```

A few examples of degeneracy (h1, h2, h3 triplets associated with a special point/energy couple)

```

point N, E=1.5: [[-1, -1, 0], [-1, 0, 0], [0, 0, 1], [0, 1, 1]]
point G, E=2:  [[-1, -1, 0], [-1, 0, 0], [-1, 0, 1], [0, -1, -1], [0,
-1, 0], [0, 0, -1], [0, 0, 1], [0, 1, 0], [0, 1, 1], [1, 0, -1], [1,
0, 0], [1, 1, 0]]

```

In [56]:

```

import numpy as np

fig = plt.figure(figsize=(6*3,10))
ymin = 0
ymax = 3
xmin = 0
xmax = 1
x_values = np.linspace(0, 1, num=100)
titles = [r'H - $\Gamma$', r'$\Gamma$ - N', r'N-P']

starts=[[ (0,1)], [(0,1), (0,3), (0,5)], [(0,1)]], [(0,0)], [(0, 2)], [(0,4)], [(0,
ends=[[ (1, 0)], [(1,2)], [(1,4)],[(1,0.5)], [(1, 0.5), (1,1.5), (1, 2.5), (1, 3.5),

for plot_idx in range(3):
    axes = fig.add_subplot(1,3,1+plot_idx)
    for idx, a in enumerate(starts[plot_idx]):
        for b_elem in ends[plot_idx][idx]:
            for a_elem in a:
                axes.plot(x_values, [a_elem[1]+(b_elem[1]-a_elem[1])*zz for zz in x_val
#axes.set_xlabel('$k$ (Angstrom)', fontsize=18)
#axes.set_ylabel('$\omega$ (meV)', fontsize=18)
axes.set_title(titles[plot_idx], fontsize=20)
#axes.legend(loc=2) # upper left corner
axes.set_yticks(np.linspace(ymin,ymax,1+ymax-ymin,endpoint=True))
axes.set_xticks([0,1])
axes.set_ylim([ymin,ymax])
axes.set_xlim([xmin,xmax])
axes.grid(True)

```

