DOS_Si_Ge_Sn

April 22, 2014

0.0.1 *From the computed densities of states of Si, Ge, and Sn, indicate the density of states of the empty diamond lattice (with isotropic effective mass m_0 and lowest energy $E_0 = -13eV$). What is the Fermi energy in this model?*

The outer electronic structures of Si, Ge, and Sn are similar: ns^2np^2, with n=3, 4, 5 respectively, which explains the similarity in the shape of the DOS for those three elements.

In the case of free electrons in a 3D lattice, within the Sommerfeld theory of metals, the DOS is given by $g(\epsilon) = \theta(\epsilon) \cdot \frac{m}{\hbar^3\pi^2}\sqrt{2m_0\epsilon} = \frac{3}{2}\theta(\epsilon) \cdot \frac{n}{\epsilon_F}\sqrt{\frac{\epsilon}{\epsilon_F}}$. Such a square root profile for the DOS is only met in reality if the system is very close to the free electron approximation.

For semi conductors, such expression may be generalised to the bottom of the conduction band and the top of the valence band as follows:

$g_c(E)dE = \frac{4\pi}{\hbar^3}(2m_n)^{3/2}\sqrt{E - E_c}$ and $g_v(E)dE = \frac{4\pi}{\hbar^3}(2m_p)^{3/2}\sqrt{E_v - E}$ where m_n and m_p are the DOS effective masses, computed in a way such such as to take properly into account the shape and (an)isotropy of the hole/electron pockets at the extremum of the considered band.

The presence of a finite potential lifts accidental degeneracies and splits peaks in the DOS, which would in the free case occur in areas where the slope of the band structure is small, at the zone boundaries, and at the Γ point. In this more general case, the DOS is directly related to the band structure via $g_n(\epsilon) = \int_{S_n(\epsilon)} \frac{dS}{4\pi^3}\frac{1}{|\nabla\epsilon_n(k)|}$. Note that at k such that $\nabla\epsilon_n(k) = 0$, we will observe Van Hove singularities. The empty diamond lattice structure (fcc) is therefore expected to exhibit a structure similar to the figures, considering that some double peaks, split by the effect of the finite potential, should be merged.

The Fermi energy may be deduced from the difference between the experimental and the theoretical curves: The theoretical curves display both the valence and conduction bands, while the experimental data only show the occupied bands, i.e. the valence bands s^2p^2. This fact is more clearly expressed by noting that the real DOS is $g(\epsilon) \cdot f(\epsilon)$ where $f(\epsilon) = \frac{1}{e^{(\epsilon-\epsilon_F)/k_BT} + 1}$. The factor $f(\epsilon)$ goes rapidly from 1 to 0 when ϵ reaches ϵ_F, causing the conduction band to be empty, and not visible in the experimental data. In the provided figures, $\epsilon_F \simeq 0$eV.

ref: Ashcroft, Mermin, p44-Eq2.61, pp 143ff-Eq. 8.63

The following computations use the GPAW code available at https://wiki.fysik.dtu.dk/gpaw/index.html . You can also get an online version of this document at

http://nbviewer.ipython.org/urls/raw.githubusercontent.com/dombrno/CMII/master/notebooks/DOS_Si_Ge_Sn.ipynb?c and copy/paste/modify the code for your own calculations.

```
In [28]: %matplotlib inline
```

1 Si

Si (14) structure: [Ne]$3s^23p^2$

```
In [29]: import pickle
         import numpy as np
```

```
from ase.lattice import bulk
from gpaw import GPAW, PW, FermiDirac
from ase.dft.kpoints import ibz_points, get_bandpath

# Perform standard ground state calculation (with plane wave basis)
si = bulk('Si', 'diamond', 5.43)
calc = GPAW(mode=PW(200),
            xc='PBE',
            kpts=(8, 8, 8),
            occupations=FermiDirac(0.01),
            txt='Si_gs.txt')
si.set_calculator(calc)
si.get_potential_energy()
calc.write('Si_gs.gpw')
```

In [30]:
```
# Restart from ground state and obtain Fermi level
calc = GPAW('Si_gs.gpw',
            txt=None,
            fixdensity=True,
            eigensolver='cg',
            usesymm=None,
            convergence={'bands': 7})
ef = calc.get_fermi_level()

# Use ase.dft module for obtaining k-points along high symmetry directions
points = ibz_points['fcc']
G = points['Gamma']
X = points['X']
W = points['W']
K = points['K']
L = points['L']
kpts, x, X = get_bandpath([W, L, G, X, W, K], calc.atoms.cell)
calc.set(kpts=kpts)
calc.get_potential_energy()
e_kn = np.array([calc.get_eigenvalues(k) for k in range(len(kpts))])
e_kn -= ef
```

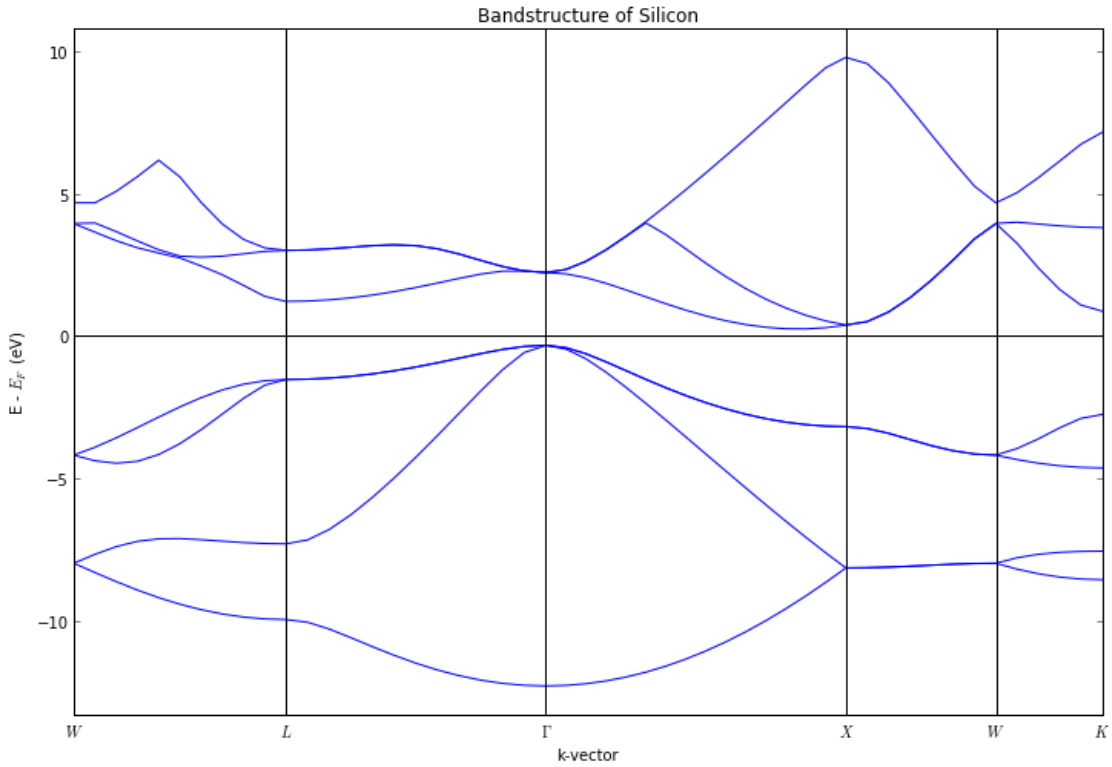
In [31]:
```
# Plot the band structure
#import matplotlib
#matplotlib.use('Agg')
#matplotlib.rc('text', usetex=True)
import matplotlib.pyplot as plt

point_names = ['W', 'L', '\Gamma', 'X', 'W', 'K']

emin = e_kn.min() - 1
emax = e_kn[:,7].max() + 1

plt.figure(figsize=(12, 8))
for n in range(7):
    plt.plot(x, e_kn[:, n], 'b')
for p in X:
    plt.plot([p, p], [emin, emax], 'k-')
plt.plot([0, X[-1]], [0, 0], 'k-')
plt.xticks(X, [r'$%s$' % n for n in point_names])
```

```
plt.axis(xmin=0, xmax=X[-1], ymin=emin, ymax=emax)
plt.xlabel('k-vector')
plt.ylabel(r'E - $E_F$ (eV)')
plt.title('Bandstructure of Silicon')
plt.show()
plt.savefig('bandstructure.pdf')
```




```
<matplotlib.figure.Figure at 0x5d19e10>
```

2 Ge

Ge (32) structure: $[\mathrm{Ar}]3d^{10}4s^2 4p^2$

```
In [32]: # Perform standard ground state calculation (with plane wave basis)
         si = bulk('Ge', 'diamond', 5.64)
         calc = GPAW(mode=PW(200),
                     xc='PBE',
                     kpts=(8, 8, 8),
                     occupations=FermiDirac(0.01),
                     txt='Ge_gs.txt')
         si.set_calculator(calc)
         si.get_potential_energy()
         calc.write('Ge_gs.gpw')
```

```
In [33]: # Restart from ground state and obtain Fermi level
         calc = GPAW('Ge_gs.gpw',
```

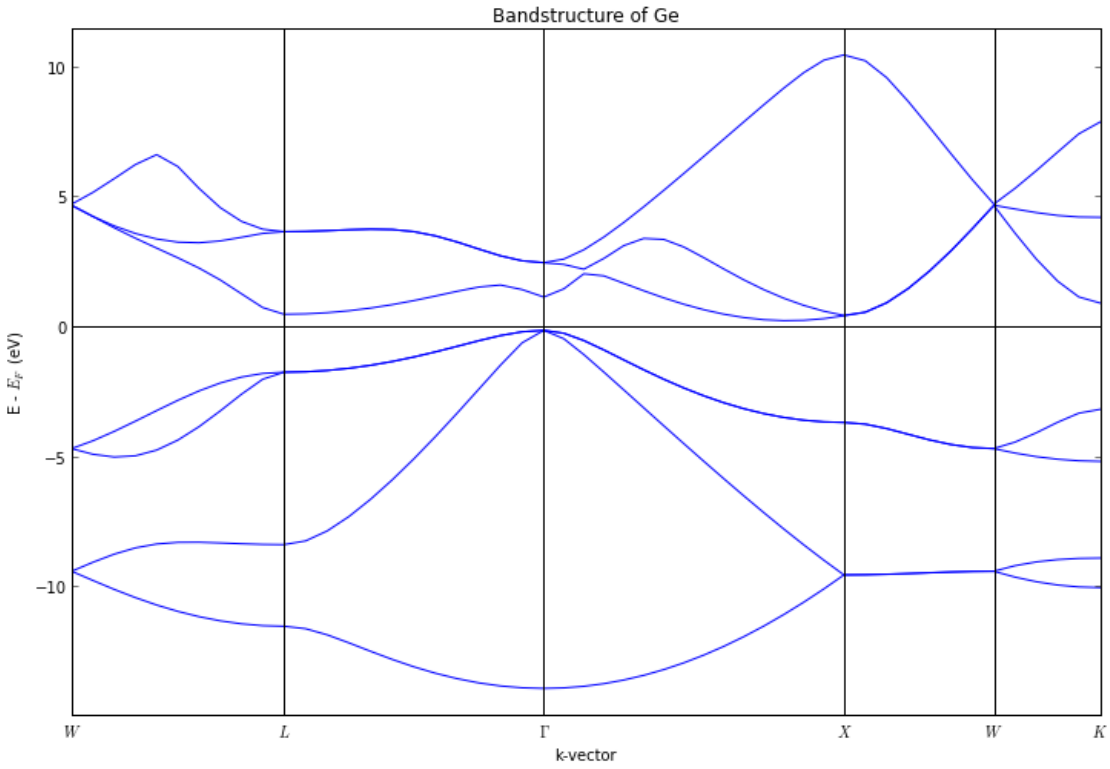
```
                txt=None,
                fixdensity=True,
                eigensolver='cg',
                usesymm=None,
                convergence={'bands': 'occupied'})
        ef = calc.get_fermi_level()

        # Use ase.dft module for obtaining k-points along high symmetry directions
        points = ibz_points['fcc']
        G = points['Gamma']
        X = points['X']
        W = points['W']
        K = points['K']
        L = points['L']
        kpts, x, X = get_bandpath([W, L, G, X, W, K], calc.atoms.cell)
        calc.set(kpts=kpts)
        calc.get_potential_energy()
        e_kn = np.array([calc.get_eigenvalues(k) for k in range(len(kpts))])
        e_kn -= ef
```

In [34]:
```
        # Plot the band structure
        point_names = ['W', 'L', '\Gamma', 'X', 'W', 'K']

        emin = e_kn.min() - 1
        emax = e_kn[:,7].max() + 1

        plt.figure(figsize=(12, 8))
        for n in range(7):
            plt.plot(x, e_kn[:, n], 'b')
        for p in X:
            plt.plot([p, p], [emin, emax], 'k-')
        plt.plot([0, X[-1]], [0, 0], 'k-')
        plt.xticks(X, [r'$%s$' % n for n in point_names])
        plt.axis(xmin=0, xmax=X[-1], ymin=emin, ymax=emax)
        plt.xlabel('k-vector')
        plt.ylabel(r'E - $E_F$ (eV)')
        plt.title('Bandstructure of Ge')
        plt.savefig('Ge_bandstructure.pdf')
```



3 Sn

Sn (50) structure: $[\mathrm{Kr}]4d^{10}5s^25p^2$

```
In [75]: # Perform standard ground state calculation (with plane wave basis)
         si = bulk('Sn', 'diamond', 6.48)
         calc = GPAW(mode=PW(200),
                     xc='PBE',
                     kpts=(8, 8, 8),
                     occupations=FermiDirac(0.01),
                     txt='Sn_gs.txt')
         si.set_calculator(calc)
         si.get_potential_energy()
         calc.write('Sn_gs.gpw')

In [76]: # Restart from ground state and obtain Fermi level
         calc = GPAW('Sn_gs.gpw',
                     txt=None,
                     fixdensity=True,
                     eigensolver='cg',
                     usesymm=None,
                     convergence={'bands': 'occupied'})
         ef = calc.get_fermi_level()

         # Use ase.dft module for obtaining k-points along high symmetry directions
         points = ibz_points['fcc']
```

```
G = points['Gamma']
X = points['X']
W = points['W']
K = points['K']
L = points['L']
kpts, x, X = get_bandpath([W, L, G, X, W, K], calc.atoms.cell)
calc.set(kpts=kpts)
calc.get_potential_energy()
e_kn = np.array([calc.get_eigenvalues(k) for k in range(len(kpts))])
e_kn -= ef
```

In [78]:
```
# Plot the band structure
point_names = ['W', 'L', '\Gamma', 'X', 'W', 'K']
taille = e_kn.shape[1]
emin = e_kn.min() - 1
emin = -12
emax = e_kn[:,:].max() + 1

plt.figure(figsize=(12, 8))
for n in range(taille):
    plt.plot(x, e_kn[:, n], 'b')
for p in X:
    plt.plot([p, p], [emin, emax], 'k-')
plt.plot([0, X[-1]], [0, 0], 'k-')
plt.xticks(X, [r'$%s$' % n for n in point_names])
plt.axis(xmin=0, xmax=X[-1], ymin=emin, ymax=emax)
plt.xlabel('k-vector')
plt.ylabel(r'E - $E_F$ (eV)')
plt.title('Bandstructure of Sn')
plt.savefig('Sn_bandstructure.pdf')
```