DOS_Si_Ge_Sn

D. Geffroy

February 23, 2014

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/github/dombrno/CMII/blob/master/notebooks/DOS_Si_Ge_Sn.ipynb?create=1

and copy/paste/modify the code for your own calculations.

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

Part I

Si

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']
```

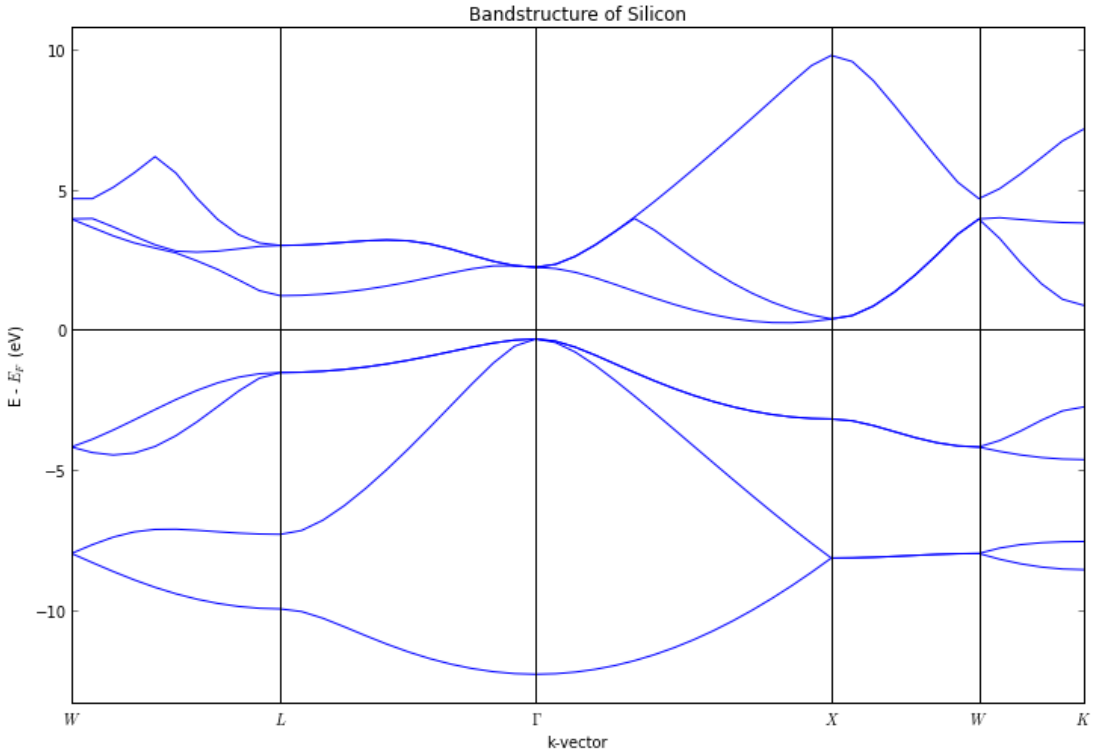
```
K = points['K']
L = points['L']
kpts, x, X = get_bandpath([W, L, G, X, W, K], calc.atoms.cell)
calc.set(kpts=kpts)
calc.get_potential_energy()
e_kn = np.array([calc.get_eigenvalues(k) for k in range(len(kpts))])
e_kn -= ef
```

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>
```

Part II

Ge

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')
```

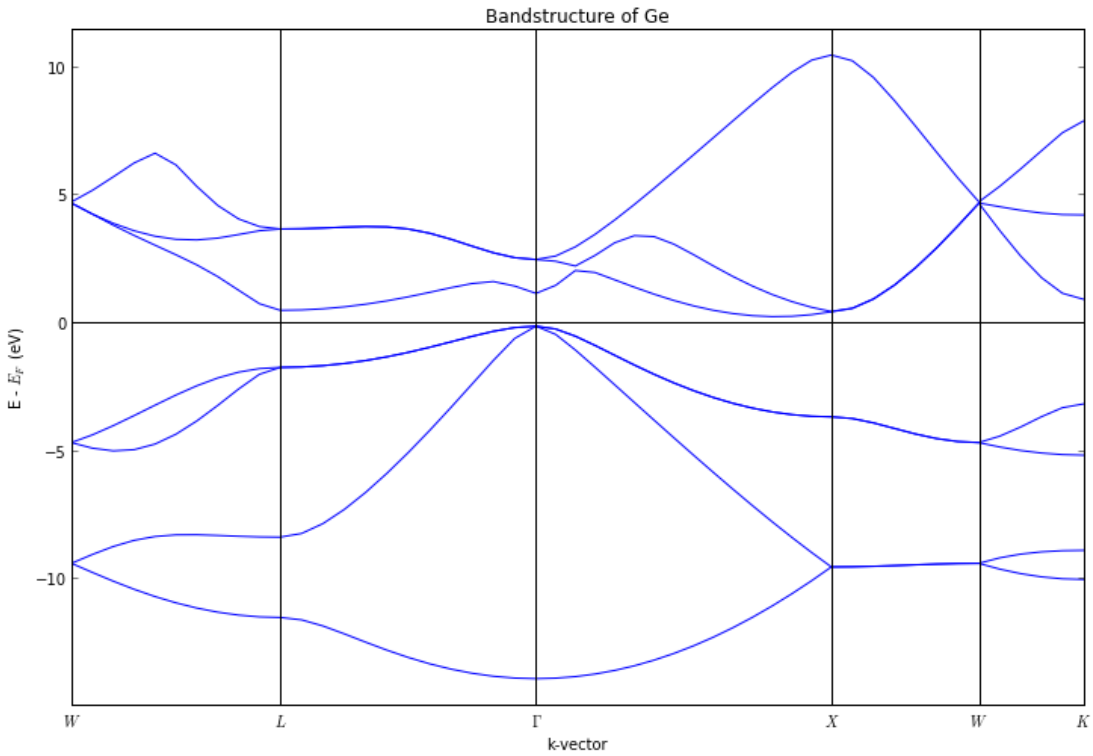
In [33]:
```
# Restart from ground state and obtain Fermi level
calc = GPAW('Ge_gs.gpw',
            txt=None,
            fixdensity=True,
            eigensolver='cg',
            usesymm=None,
            convergence={'bands': 'occupied'})
ef = calc.get_fermi_level()

# 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')
```



Part III

Sn

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')
```

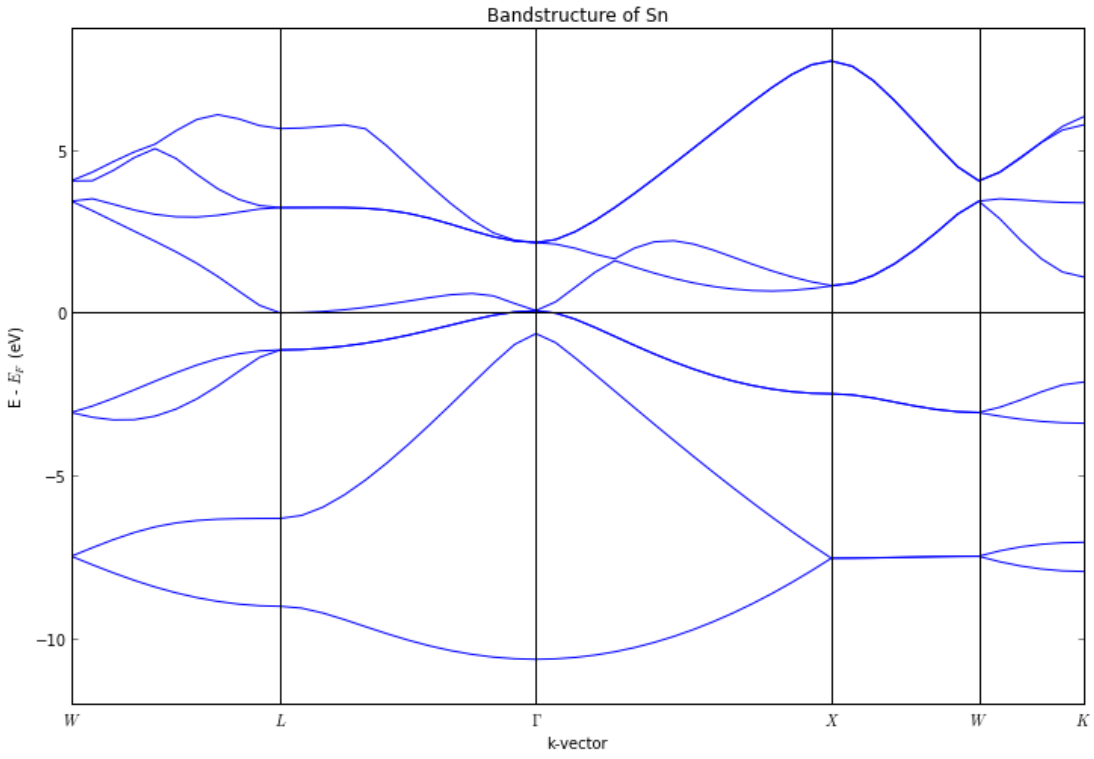
In [76]:
```
# Restart from ground state and obtain Fermi level
calc = GPAW('Sn_gs.gpw',
            txt=None,
            fixdensity=True,
            eigensolver='cg',
            usesymm=None,
            convergence={'bands': 'occupied'})
ef = calc.get_fermi_level()

# 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')
```



In []: