1 Density of states

We already talked about the density of states (DOS) of the homogeneous electron gas and its definition in general (see exercise sheet 2). Now we want to calculate it *ab inito* for different materials like insulators, semi-conductors and metals.

1. Perform ground state calculations for silicon, aluminum and sodium chloride (optional: gallium arsenide, iron). Plot the density of states for each material using

   ```
   en, dos = si.calc.get_dos(spin=0, npts=201, width=0.1)
   ```

   and study their differences. What can you tell about the DOS at the Fermi level $D(E_\mathrm{F})$?

   Note: Especially for metals, you need a rather dense $\boldsymbol{k}$-grid. Once your code works without errors, you should use a BZ sampling of $12 \times 12 \times 12$ points and a plane wave cut-off of $800\,\mathrm{eV}$.

2. What can you conclude for the bandstructure just from looking at the DOS plots?

3. What is the result of integrating the DOS of non-metals a) over the entire computed range and b) up to the Fermi energy? What changes if you vary the `nbands` parameter?

   Hint: For finding the index of the energy nearest to E_F, slicing the DOS-array at E_F and plotting a vertical line at E_F, you can use this snippet:

   ```
   idx = np.abs(en - efermi).argmin()
   dosUpToEf, enUpToEf= dos[:(idx+1)], en[:(idx+1)]
   plt.axvline(x=efermi, ls="--", alpha=0.5)
   ```

Solution:

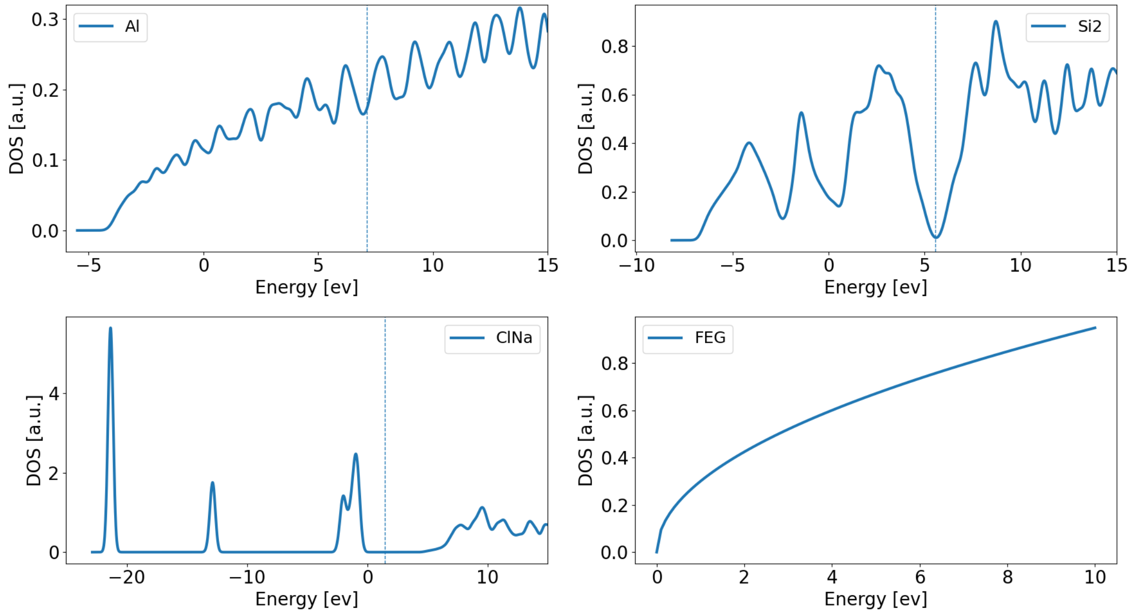
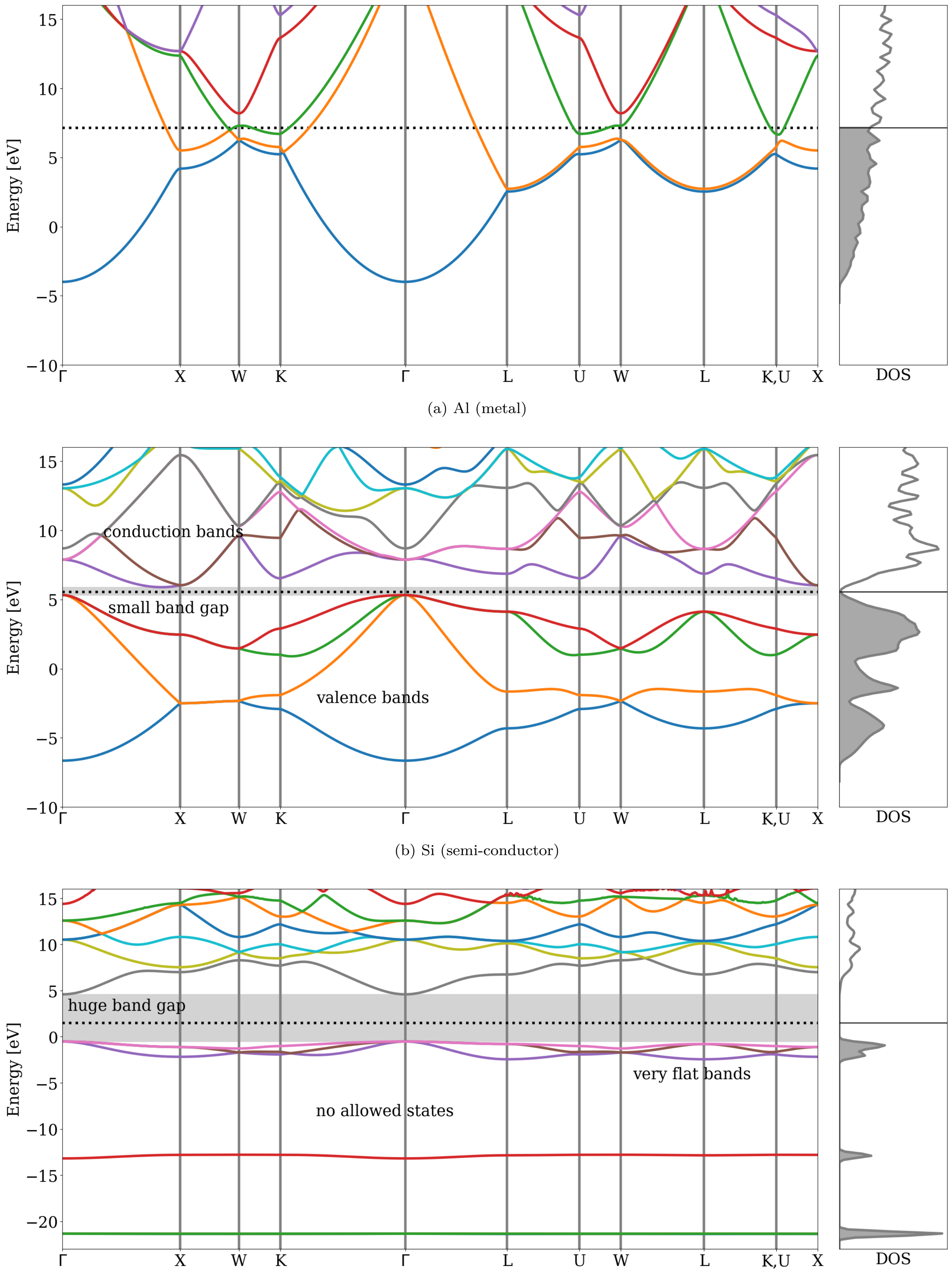


Figure 1: Ab initio calculated density of states (DOS) of Al, Si, NaCl and the free electron gas for comparison. The vertical line shows the Fermi level E_F. For metals there is no band gap, for semi-conductors the Fermi level lies exactly in the gap. Insulators have a rather wide gap as well as very small bandwidths. In their band structure this corresponds to very flat bands and regions with no allowed states at all. Keep in mind that absolute energy values on the x-axis (for the bandstructure: y-axis) have no physical meaning. Some electronic structure codes prefer to return energies relative to E_F, however, GPAW does not by default.

DOS calculations:

```python
from ase.build import bulk
from gpaw import GPAW, PW
import matplotlib.pyplot as plt
import numpy as np

# create some semi-conductor, insulator and metal structures
al = bulk("Al", "fcc", a=4.05)
fe = bulk("Fe", "bcc", a=2.87)
si = bulk("Si", "diamond", a=5.43)
gaas = bulk("GaAs", "zincblende", a=5.65)
nacl = bulk("NaCl", "rocksalt", a=5.64)

# create figure with multiple subplots (=axes)
fig, axarray = plt.subplots(2, 2)
# flatten (2x2) array of axes to (4x1) for comfortable iteration later
axes = axarray.flatten()

# loop over cells
for idx, cell in enumerate([al, si, nacl]):
    # perform ground state calculations and save results for later
    cell.calc = GPAW(xc="PBE", mode=PW(800), kpts=(12, 12, 12))
    etot = cell.get_potential_energy()
    efermi = cell.calc.get_fermi_level()
    formula = cell.get_chemical_formula()
    cell.calc.write(f"{formula}.gpw")
    # get density of states and plot it; width --> Gaussian smearing
    en, dos = cell.calc.get_dos(width=0.3, npts=500)
    axes[idx].plot(en, dos, label=f"{formula}")
    # highlight Fermi level
    axes[idx].axvline(x=efermi, ls="--", lw=1)
    # set labels and legend
    axes[idx].legend(loc="best")
    axes[idx].set_xlabel("Energy [ev]")
    axes[idx].set_ylabel("DOS [a.u.]")

# create plot of sqrt-function of free electron gas
feg_en = np.linspace(0, 10, 100)
# adapt scaling for better comparison with silicon DOS
feg_dos = 0.3 * np.sqrt(feg_en)
# array[-1] returns the last element of the array: [1, 2, 3, 9][-1] = 9
axes[-1].plot(feg_en, feg_dos, label="FEG")
axes[-1].legend(loc="best")
axes[-1].set_xlabel("Energy [ev]")
axes[-1].set_ylabel("DOS [a.u.]")

# draw plots to screen
plt.show()
```



(a) Al (metal)

(b) Si (semi-conductor)

(c) NaCl (insulator)

Figure 2: Bandstructures with attached DOS as calculated using GPAW with $E_{\mathrm{cut}} = 800\,\mathrm{eV}$, 16 bands and a Brillouin zone sampling of 500 points along the special high-symmetry $\boldsymbol{k}$-point path for fcc.

DOS integration:

```python
from ase.build import bulk
from gpaw import GPAW, PW
from scipy.integrate import trapezoid

# create bulk structure and perform ground state calculation
cell = bulk("GaAs", "zincblende", a=5.65)
cell.calc = GPAW(xc="PBE", mode=PW(500), kpts=(8, 8, 8), nbands=12)
etot = cell.get_potential_energy()
ef = cell.calc.get_fermi_level()
en, dos = cell.calc.get_dos(width=0.2, npts=1001)
print("DOS range: {:.3f}  -->  {:.3f} eV".format(en.min(), en.max()))

# find array index nearest to Fermi energy
idx = abs(en - ef).argmin()
# for insulators and semi-conductors, DOS(EF) should be zero
print("DOS(EF) =  {:.3f}".format(dos[idx]))

# integrate entire DOS
totInt = trapezoid(dos, en)
print("total integral = {:.3f}".format(totInt))
# integrate only up to Fermi level
efInt = trapezoid(dos[:idx+1], en[:idx+1])
print("integral up to EF = {:.3f}".format(efInt))
print("nbands = ", cell.calc.get_number_of_bands())
```

Integrating the entire range of the computed DOS function returns the number of bands (up to some numerical errors). Note, that this is not equal to the total number of states in the system, since the latter depends on the chosen $\boldsymbol{k}$-grid via the number of unit cells, which in turn determines the total number of electrons and consequently the number of Bloch states $\psi_{n\boldsymbol{k}}(\boldsymbol{x})$ (see theory sheet 3).

For this reason, the DOS as calculated by electronic structure codes is usually returned as a "weighted" or normalized function. For the simplest case this typically means division by the total number of $\boldsymbol{k}$-points, which guarantees a $\boldsymbol{k}$-grid-independent result. Integrating this weighted function up to E_F then yields the number of occupied states per unit cell.

For a spin-unpolarized DOS, i. e. a DOS containing both "spin channels", the so obtained number of occupied states is equal to the number of electrons in the unit cell divided by two. Note, that the integration result of course depends on the data range of the DOS function. GPAW by default returns an energy list that covers at least all *valence states* but not the so-called "frozen core", hence the integration yields half the number of valence electrons.

$$\text{Ga: [Ar]} + \frac{\upharpoonleft\downharpoonright}{4\mathrm{s}^2} \quad \frac{\upharpoonleft\;\_\_\;\_\_}{4\mathrm{p}^1} \qquad \text{As: [Ar]} + \frac{\upharpoonleft\downharpoonright}{4\mathrm{s}^2} \quad \frac{\upharpoonleft\;\upharpoonleft\;\upharpoonleft}{4\mathrm{p}^3} \qquad \text{Si: [Ne]} + \frac{\upharpoonleft\downharpoonright}{3\mathrm{s}^2} \quad \frac{\upharpoonleft\;\upharpoonleft\;\_\_}{3\mathrm{p}^2}$$

For instance, similar to silicon, each gallium arsenide pair shares 8 total valence electrons from their outermost shells which can occupy $8/2 = 4$ states per unit cell or, equivalently, per $\boldsymbol{k}$-point.

$$\text{Na: [Ne]} + \frac{\upharpoonleft}{3\mathrm{s}^1} \qquad \text{Cl: [Ne]} + \frac{\upharpoonleft\downharpoonright}{3\mathrm{s}^2} \quad \frac{\upharpoonleft\downharpoonright\;\upharpoonleft\downharpoonright\;\upharpoonleft}{3\mathrm{p}^5}$$

In contrast to the covalent bonds in Si and GaAs, NaCl forms ionic bonds where sodium donates its 3s electron to chlorine to completely fill their 3p subshell. These p-orbitals form the 3 bands right below the band gap, while the conduction band is mainly derived from the sodium ion's empty 3s and 3p orbitals. After increasing the plot range to show even lower states, it looks like there are two more bands roughly located at -12 and -21 eV, but you can tell from the plot colors that the lowest band is in fact a 3-fold degenerated overlap (default color order is: blue, orange, green, red).[1] These 4 lower bands can be attributed to the Na $2p^6$ and Cl $3s^2$ orbitals as illustrated in Fig. 3c.[2]

2 Projected density of states

As briefly mentioned in last lab's sheet, there is a method to project the DOS onto a set of atomic orbitals in order to calculate the so-called *projected* density of states. It is sometimes used by researchers for bonding analysis and can be motivated as follows:

We start with the definition of the DOS and add a "one" of the form $1 = \langle\psi_n|\psi_n\rangle$, where $\{|\psi_n\rangle\}$ is the set of (orthonormalized) Kohn-Sham eigenstates (= auxiliary states used in DFT),

$$D(E) = \sum_n \langle\psi_n|\psi_n\rangle\, \delta(E - E_n)\,. \tag{1}$$

Now we can insert identity again using the completeness property of bases,

$$\mathbb{1} = \sum_k |k\rangle\langle k| = \int \mathrm{d}^3\boldsymbol{x}\; |\boldsymbol{x}\rangle\langle\boldsymbol{x}| = \sum_i |\phi_i^a\rangle\langle\phi_i^a|\;, \tag{2}$$

where $\{|\phi_i^a\rangle\}$ is again a set of orthonormal, but *atomic* eigenfunctions corresponding to atom a. This transforms the DOS into

$$D(E) = \sum_{n,i} |\langle\phi_i^a|\psi_n\rangle|^2\, \delta(E - E_n) = \sum_i D_i(E)\,, \tag{3}$$

where the individual PDOS functions are now simply defined as

$$D_i(E) = \sum_n |\langle\phi_i^a|\psi_n\rangle|^2\, \delta(E - E_n)\,. \tag{4}$$

The total density of states "TDOS" is then reobtained by summing over all PDOS,

$$D(E) = \sum_i D_i(E)\,. \tag{5}$$

A nice summary of this topic can be found in the GPAW documentation:
https://wiki.fysik.dtu.dk/gpaw/tutorialsexercises/electronic/pdos/pdos.html

You can use the function

```
calc.get_orbital_ldos(a=atomID, spin=0, angular=orbital, width=0.3, npts=2001)
```

to calculate the DOS projected on wave functions of atom with index a (e. g. 0 and 1 for As and Ga in GaAs or Si #1 and Si #2 for Si in diamond structure) and orbital `'s'`, `'p'`, `'d'` or `'f'`. The parameter `width` is again a smearing factor which is used to smooth out the resulting plot.

[1] Eigenvalue degeneration be can alternatively checked via `calc.band_structure().energies[0,kpt,band]`.

[2] Technically, the Na $2p^6$ electrons are not part of the valence shell, but GPAW includes them for higher accuracy, see https://wiki.fysik.dtu.dk/gpaw/setups/Na.html. Also check GPAW output for species setup during initialization. This is why the DOS integrates to $4 + 6/2 = 7$ occupied states instead of only 4 for NaCl.

Note, that although in theory all PDOS should sum up to the total DOS, there are practical issues preventing this in general.[3] Therefore, you should never rely on the PDOS concept for quantitative studies but only reference it for qualitative trends. This effect is also clearly visible in the following plots and becomes worse for energy ranges way above the Fermi level.

Partial DOS plot:

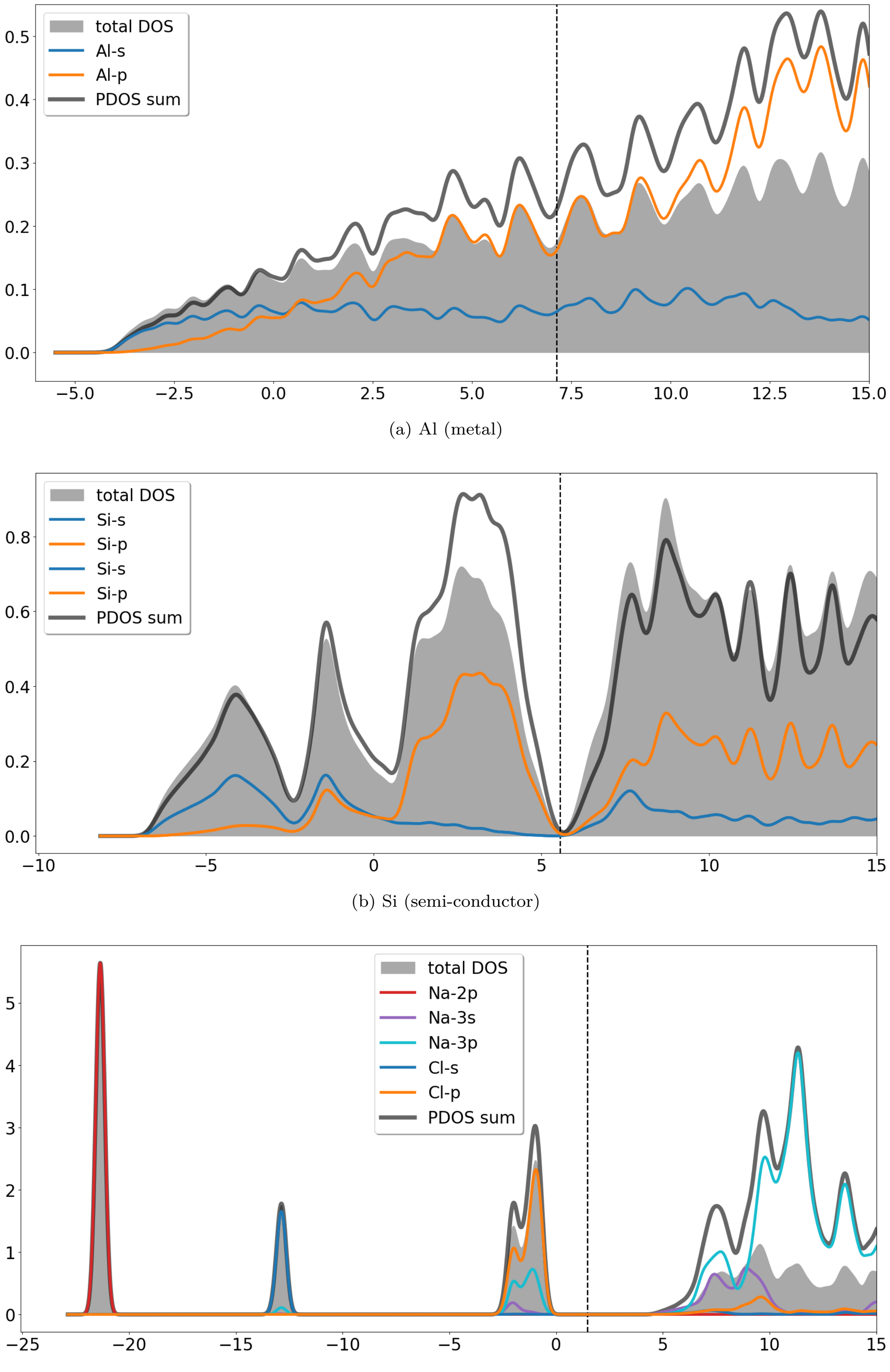
```python
import numpy as np
from gpaw import GPAW
import matplotlib.pyplot as plt

# restart from save file
calc = GPAW("Al.gpw", txt=None)
ef = calc.get_fermi_level()
symbols = calc.get_atoms().get_chemical_symbols()
width = 0.3

# calculate total DOS for comparison
en0, dos = calc.get_dos(width=width, npts=2001)
plt.fill(en0, dos, label="total DOS", c="darkgrey")
plt.axvline(x=ef, ls="--", lw=2, c="black")

# for each atom loop through 's' and 'p' orbitals and plot PDOS
all_pdos = []
for a, atom in enumerate(symbols):
    for orb in "sp":
        en, pdos = calc.get_orbital_ldos(
            a=a, spin=0, angular=orb, width=width, npts=2001
        )
        plt.plot(en, pdos, label=f"{atom}-{orb}", lw=2)
        all_pdos.append(pdos)

# plot sum of all PDOS
pdossum = np.sum(all_pdos, axis=0)
plt.plot(en0, pdossum, lw=6, c="k", alpha=0.6, zorder=1, label="PDOS sum")

# draw plots to screen
plt.legend(fancybox=True, shadow=True)
plt.show()
```

[3] A very good description of these issues is given in this stackexchange thread:
https://mattermodeling.stackexchange.com/questions/10530/why-do-we-not-get-dos-by-summing-over-all-the-pdos

> "The fundamental problem is that the basis set used for the partial density-of-states (PDOS) does not span the plane-wave basis set space, so it is not usually possible to represent the Kohn-Sham states exactly in the atom-centred basis used for PDOS. There are two main issues: (i) Finite radial extent and (ii) Limited spherical harmonics."

This issue is also remarked in the GPAW documentation, where it states:
https://wiki.fysik.dtu.dk/gpaw/tutorialsexercises/electronic/pdos/pdos.html#atomic-orbital-pdos

> "Note that the set of atomic partial waves do not form an orthonormal basis, thus the properties of the introduction [i. e. inserting a complete orthonormal set in form of identity] are not fulfilled. This PDOS can however be used as a qualitative measure of the local character of the DOS."



(a) Al (metal)

(b) Si (semi-conductor)

(c) NaCl (insulator)

Figure 3: Partial and total density of states as calculated using GPAW. The sum of all partial DOS lines does not match the total DOS as it should in theory. This is not a smearing effect but rather an inherent issue with the practical implementation of band projectors onto atomic orbitals. For conduction bands, which DFT cannot reproduce very well anyway, this effect is even more visible. For NaCl for example, the sum is completely off compared to the TDOS even when using well converged parameters.