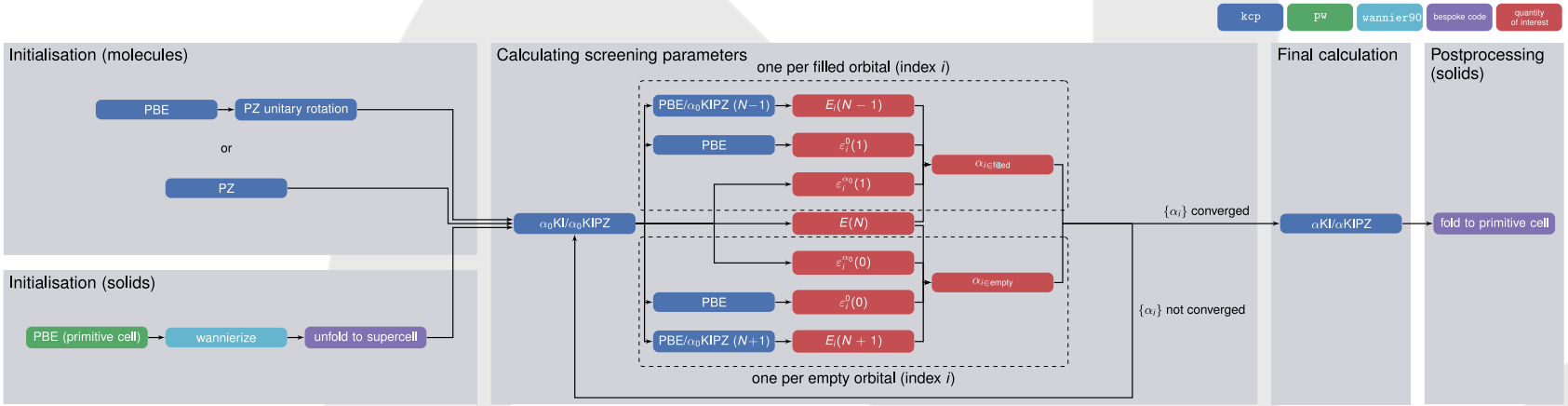
Advanced Quantum ESPRESSO tutorial: Hubbard and Koopmans functionals from linear response

Edward Linscott, Riccardo De Gennaro, and Nicola Colonna,

DAY 3 – Hands-on

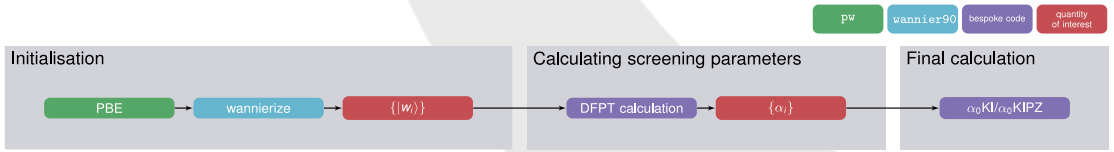
What we will do today



Koopmans calculations using finite differences to calculate screening parameters

Tutorial 1: ozone (via `koopmans`)

Tutorial 2: silicon (via `koopmans`)



Koopmans calculations using DFPT to calculate screening parameters

Tutorial 3: ozone (with `QE` directly)

Tutorial 4: silicon (with `QE` directly)

Tutorial 5: zinc oxide (with `QE` directly and via `koopmans`)

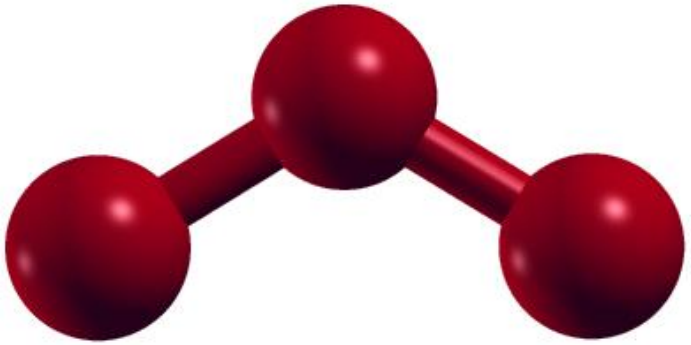
Tutorial #1

Ozone molecule with KCP

Goal of the tutorial:
compute the **charged excitations** of the **ozone** molecule.

Method:
- KI functional
- Screening parameters computed from finite-energy differences (KCP code)

Main steps:
1. DFT inizialization
2. Screening parameters
3. Final KI calculation



WORKDIR = `Day3/exercise1`

How to run:
1. Activate the `koopmans` virtual environment
   `conda activate koopmans`
2. Run the calculation
   `koopmans ozone.json`

```
Koopmans spectral functional calculations with Quantum ESPRESSO

version 1.0.0b4

Written by Edward Linscott, Riccardo De Gennaro, and Nicola Colonna

Please cite the papers listed in ozone.bib in work involving this calculation
```

JSON input file

```
{
  "workflow": {
    "functional": "ki",
    "method": "dscf",
    "init_orbitals": "kohn-sham",
    "from_scratch": true,
    "n_max_sc_steps": 1,
    "pseudo_library": "sg15"
  },
  "atoms": {
    "cell_parameters": {
      "vectors": [[8.1738, 0.0, 0.0],
                  [0.0, 6.0, 0.0],
                  [0.0, 0.0, 6.66]],
      "units": "angstrom",
      "periodic": false
    },
    "atomic_positions": {
      "units": "angstrom",
      "positions": [
        ["O", 4.0869, 3.0, 2.89],
        ["O", 5.1738, 3.0, 3.55],
        ["O", 3.0, 3.0, 3.55]
      ]
    }
  },
  "calculator_parameters": {
    "ecutwfc": 50.0,
    "ecutrho": 200.0,
    "nbnd": 10
  }
}
```

WORKDIR = `Day3/exercise1`

orbital-density-dependent/density functional to use (`ki, kipz, pkipz, dft, all`)

the method to calculate the screening parameters: either `dscf` or `dfpt`

which orbitals to use as an initial guess for the variational orbitals (`pz, kohn-sham, mlwfs, projwfs`)

max number of self-consistency steps for calculating α_i

the pseudopotential library to use (`sg15, sg15_relativistic, pseudo_dojo_standard, pseudo_dojo_stringent`)

For more info:
https://koopmans-functionals.org/en/latest/input_file.html

Orbitals initialization

In this tutorial we chose to initialize the variational orbitals to Kohn-Sham.

WORKDIR = `Day3/exercise1/init`

```
Initialization of density and variational orbitals
==================================================
 Running init/dft_init_nspin1... done
 Running init/dft_init_nspin2_dummy... done
 Running init/dft_init_nspin2... done
 Overwriting the variational orbitals with Kohn-Sham orbitals
 Copying the spin-up variational orbitals over to the spin-down channel
```

- KCP implementation computes α_i from finite-differences.
  It requires spin-polarized calculations (`nspin=2`), even for spin-unpolarized systems.

- To avoid spin contamination we perform first a spin-unpolarized calc (`dft_init_nspin1`, where `nspin=1`).

- Then we copy the spin-up KS orbitals into the spin-down channel and restart a spin-polarized calc (`dft_init_nspin2`). The `dummy` calc serves just to create the restart directory.

Orbitals initialization

Let's change directory and go to `init`.

WORKDIR = `Day3/exercise1/init`

Let's open `dft_init_spin2.cpo` and check the energy minimization and the final eigenvalues (to be compared later with the final KI eigenvalues).

```
PERFORMING CONJUGATE GRADIENT MINIMIZATION OF EL. STATES

     CP           :     0.24s CPU time,    0.36s wall time

     iteration =   1  eff iteration =   1   Etot (Ha) =    -47.52963244943611
     CP           :     0.27s CPU time,    0.39s wall time

     iteration =   2  eff iteration =   2   Etot (Ha) =    -47.52963245625526
     CP           :     0.30s CPU time,    0.43s wall time

     iteration =   3  eff iteration =   3   Etot (Ha) =    -47.52963245948904 delta_E=  0.68191496893633E-08
     CP           :     0.33s CPU time,    0.46s wall time

     iteration =   4  eff iteration =   4   Etot (Ha) =    -47.52963246118958 delta_E=  0.32337865718546E-08
     CP           :     0.36s CPU time,    0.49s wall time

     iteration =   5  eff iteration =   5   Etot (Ha) =    -47.52963246206975 delta_E=  0.17005348240673E-08

   Empty-states: WFCs read from file
   Empty-states: Going to re-orthogonalize to occ manifold

   Empty states minimization starting
   nfi         dekinc         ekinc
```

CONVERGED!

```
 HOMO Eigenvalue (eV)

 -7.9229

 LUMO Eigenvalue (eV)

 -6.1058

 Electronic Gap (eV) =     1.8170


 Eigenvalues (eV), kp =   1 , spin =  1

-34.7197  -27.7828  -19.5738  -14.9924  -14.8150  -14.5220
-9.1648   -8.0885   -7.9229

 Empty States Eigenvalues (eV), kp =   1 , spin =  1

 -6.1058

 Eigenvalues (eV), kp =   1 , spin =  2

-34.7197  -27.7828  -19.5738  -14.9924  -14.8150  -14.5220
-9.1648   -8.0885   -7.9229

 Empty States Eigenvalues (eV), kp =   1 , spin =  2

 -6.1058
```

Screening parameters α_i

The α_i are initialized to $\alpha_i^{(0)} = 0.6$.

WORKDIR = `Day3/exercise1/calc_alpha`

```
Calculating screening parameters
================================
 Running calc_alpha/ki... done

Orbital 1
---------
 Running calc_alpha/orbital_1/dft_n-1... done

Orbital 2
---------
 Running calc_alpha/orbital_2/dft_n-1... done

Orbital 3
---------
 Running calc_alpha/orbital_3/dft_n-1... done

Orbital 4
---------
 Running calc_alpha/orbital_4/dft_n-1... done

Orbital 5
---------
 Running calc_alpha/orbital_5/dft_n-1... done

Orbital 6
---------
 Running calc_alpha/orbital_6/dft_n-1... done

Orbital 7
---------
 Running calc_alpha/orbital_7/dft_n-1... done

Orbital 8
---------
 Running calc_alpha/orbital_8/dft_n-1... done

Orbital 9
---------
 Running calc_alpha/orbital_9/dft_n-1... done

Orbital 10
----------
 Running calc_alpha/orbital_10/pz_print... done
 Running calc_alpha/orbital_10/dft_n+1_dummy... done
 Running calc_alpha/orbital_10/dft_n+1... done

 alpha
          1         2         3         4         5         6         7         8         9        10
 0  0.600000  0.600000  0.600000  0.600000  0.600000  0.600000  0.600000  0.600000  0.600000  0.600000
 1  0.656974  0.729771  0.785787  0.668565  0.775551  0.730287  0.735302  0.746247  0.784111  0.725779
```

$$\alpha_i = \alpha_i^{(0)} \frac{\Delta E_i - \langle \phi_i | \hat{h}^{DFT} | \phi_i \rangle}{\langle \phi_i | \hat{h}_i^{KI} | \phi_i \rangle - \langle \phi_i | \hat{h}^{DFT} | \phi_i \rangle}$$

E^N, $\langle \phi_i | \hat{h}_i^{KI} | \phi_i \rangle$ and $\langle \phi_i | \hat{h}^{DFT} | \phi_i \rangle$ are obtained from the trial KI calc

$E_i^{N\pm1}$ are computed orbital by orbital (for empty states, a couple of dummy calcs are required before performing to prepare the $(N+1)$-electron system)

Increase `n_max_sc_steps` for more iterations (unnecessary for KI)

Final KI calculation

Diagonalize $\hat{h}^{DFT} + \alpha_i \hat{v}_i^{KI}$

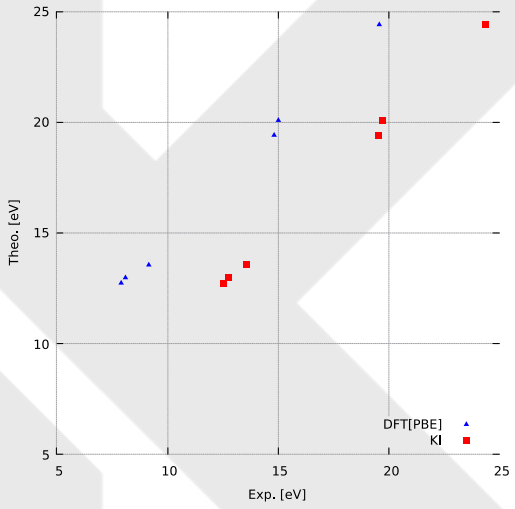
Go to `final` and open `ki_final.cpo`

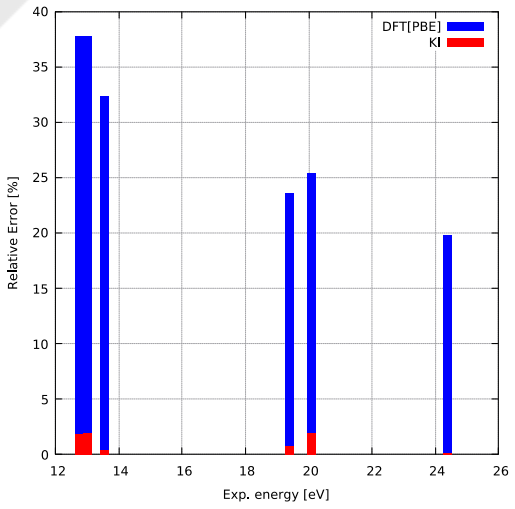
```
 HOMO Eigenvalue (eV)
-12.4945
 LUMO Eigenvalue (eV)
 -1.7184
 Electronic Gap (eV) =    10.7761

 Eigenvalues (eV), kp =   1 , spin =  1
-40.3490  -33.0412  -24.3772  -19.7139  -19.5385  -19.2977  -13.5960  -12.7467  -12.4945
 Empty States Eigenvalues (eV), kp =   1 , spin =  1
 -1.7184
 Eigenvalues (eV), kp =   1 , spin =  2
-40.3490  -33.0412  -24.3772  -19.7139  -19.5385  -19.2977  -13.5960  -12.7467  -12.4945
 Empty States Eigenvalues (eV), kp =   1 , spin =  2
 -1.7184
```

WORKDIR = `Day3/exercise1/final`

For a more direct comparison KI-vs-DFT you can run the two scripts:

```
> sh get_BE.sh
> gnuplot BE.gnu
```





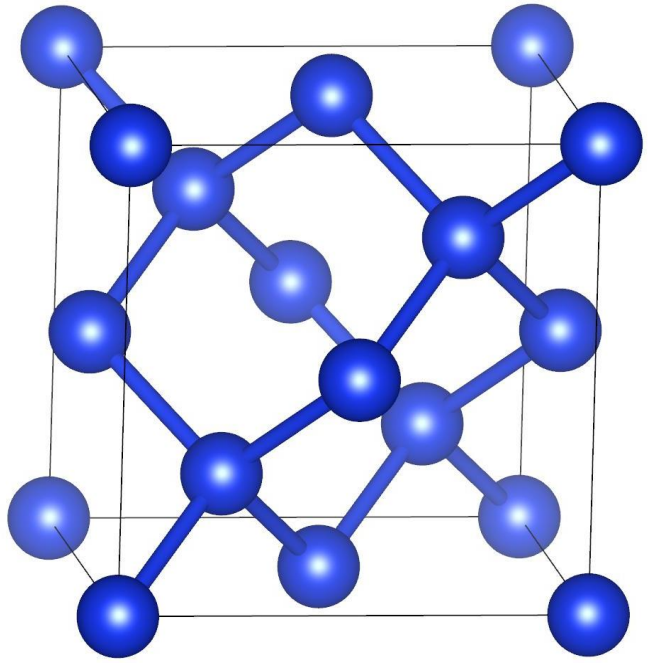
Tutorial #2

Bulk silicon with KCP

Goal of the tutorial:
compute the **band structure** of silicon.

Method:
- KI and pKIPZ
- Screening parameters computed from finite-energy differences (KCP code)

Main steps:
1. DFT initialization
2. Maximally localized Wannier functions
3. Screening parameters
4. Final KI calculation
5. Band structure unfolding & interpolation



WORKDIR = `Day3/exercise2`

JSON input file

```
{
    "workflow": {
        "functional": "ki",
        "base_functional": "lda",
        "method": "dscf",
        "mp_correction": false,
        "init_orbitals": "mlwfs",
        "alpha_guess": 0.077,
        "orbital_groups": [0, 0, 0, 0, 1, 1, 1, 1],
        "pseudo_library": "pseudo_dojo_standard",
        "from_scratch": true
    },
    "atoms": {
        "cell_parameters": {
            "periodic": true,
            "ibrav": 2,
            "celldms": {"1": 10.2622}
        },
        "atomic_positions": {
            "units": "crystal",
            "positions": [["Si", 0.00, 0.00, 0.00],
                          ["Si", 0.25, 0.25, 0.25]]
        }
    },
    "kpoints": {
        "grid": [2, 2, 2],
        "offset": [0, 0, 0],
        "path": "LGXKG"
    },
```

WORKDIR = `Day3/exercise2`

Activating/deactivating Makov-Payne charge corrections for the N±1 calculations

Initializing orbitals to maximally localized Wannier functions (MLWFs)

Initial guess for α_i

<u>Orbitals grouping</u>: tells the workflow how to group the orbitals during the calculation of α_i. Usually this requires to know already the form of the variational orbitals, i.e. the MLWFs. In this specific case we have 8 states (4 occ, 4 emp): the 4 occupied MLWFs have a bonding sp3-like form (group 0), while the 4 MLWFs have an antibonding sp3-like form (group 1).

Alternatively, one can define a threshold (`orbital_groups_self_hartree_tol`) which differentiates between orbitals that have a different self-Hartree energy.

For more info:
https://koopmans-functionals.org/en/latest/input_file.html

JSON input file

WORKDIR = `Day3/exercise2`

```
    "calculator_parameters": {
        "ecutwfc": 20.0,
        "pw": {
            "system": {
                "nbnd": 20
            }
        },
        "w90": {
            "bands_plot": true,
            "projections": [{"fsite": [ 0.25, 0.25, 0.25 ], "ang_mtm": "sp3"}],
            "emp": {
                "dis_froz_max": 10.6,
                "dis_win_max": 16.9
            }
        },
        "ui": {
            "smooth_int_factor": 4
        }
    },
    "plotting": {
        "Emin": -15,
        "Emax": 10,
        "degauss": 0.2
    }
}
```

Num of bands in the initial PW DFT calc, preceding the calculation of MLWFs.

Wannier90 block:
projections, disentanglement windows, etc.

Smooth interpolation method (discussed later on)

Plotting block:
parameters relevant for the DOS

For more info:
https://koopmans-functionals.org/en/latest/input_file.html

Orbitals initialization

In this tutorial we chose to initialize the variational orbitals to maximally localized Wannier functions (MLWFs).

WORKDIR = `Day3/exercise2/init`

```
 Initialization of density and variational orbitals
 ==================================================

  Wannierization
  ==============
   Running wannier/scf... done
   Running wannier/nscf... done
   Running wannier/occ/wann_preproc... done
   Running wannier/occ/pw2wan... done
   Running wannier/occ/wann... done
   Running wannier/emp/wann_preproc... done
   Running wannier/emp/pw2wan... done
   Running wannier/emp/wann... done

  Folding to supercell
  --------------------
   Running occ/w2kcp... done
   Running emp/w2kcp... done
  Running init/dft_dummy...UserWarning: Small box parameters "nrb" not provided
  Estimated real mesh dimension (nr1, nr2, nr3) = 45 45 45
  Small box mesh dimension (nr1b, nr2b, nr3b) = 18 18 18

done
  Running init/dft_init... done
```

PW+W90

PC to SC extension of the Wannier functions

Final KI calculation

Extract results from the `.kwf` file.

WORKDIR = `Day3/exercise2/final`

Use `python` in the interactive mode and run:

```
from koopmans.io import read

wf = read('si.kwf')
```

`wf` contains information (including the results of each calculation) of the entire workflow

E.g. `wf.calculations` contains the list of all the calcs that were run by the workflow.

```
calcs = [c for c in wf.calculations if c.prefix in ['dft_init', 'ki_final']]

for c in calcs:
    ho = c.results['homo_energy']
    lu = c.results['lumo_energy']
    gap = lu - ho
    print(f'  {c.prefix} :\t {ho:.2f} eV\t {lu:.2f} eV\t {gap:.2f} eV')
```

	ε_{HO}	ε_{LU}	E_{gap}
dft_init :	3.81 eV	4.24 eV	0.42 eV
ki_final :	3.07 eV	4.36 eV	1.29 eV

Yet, only the energies corresponding to the points in the original $\boldsymbol{k}$-grid are included.
→ Bands unfolding&interpolation

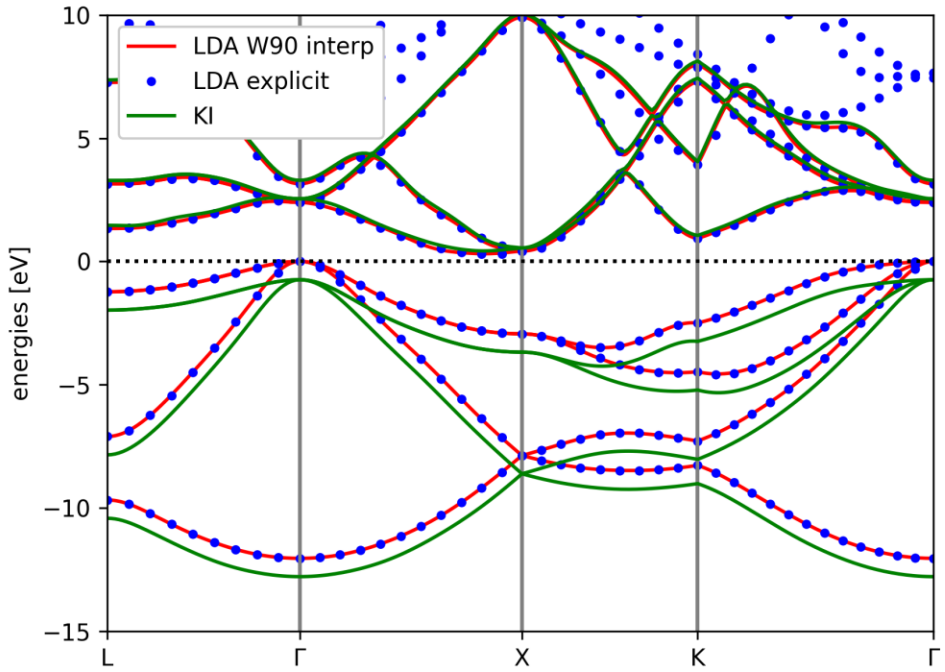
Postproc: band structure

In order to reconstruct the $\boldsymbol{k}$-dispersion of the electronic energies, we need to unfold the supercell band structure to the primitive cell.

WORKDIR = `Day3/exercise2/postproc`

```
Postprocessing
===============

  Wannierization
  ==============
   Running wannier/scf... done
   Running wannier/nscf... done
   Running wannier/occ/wann_preproc... done
   Running wannier/occ/pw2wan... done
   Running wannier/occ/wann... done
   Running wannier/emp/wann_preproc... done
   Running wannier/emp/pw2wan... done
   Running wannier/emp/wann... done
   Running wannier/bands... done
   Running pdos/projwfc... done
  Running occ/ki... done
  Running emp/ki... done

Workflow complete
```

```
python plot_bs.py
```



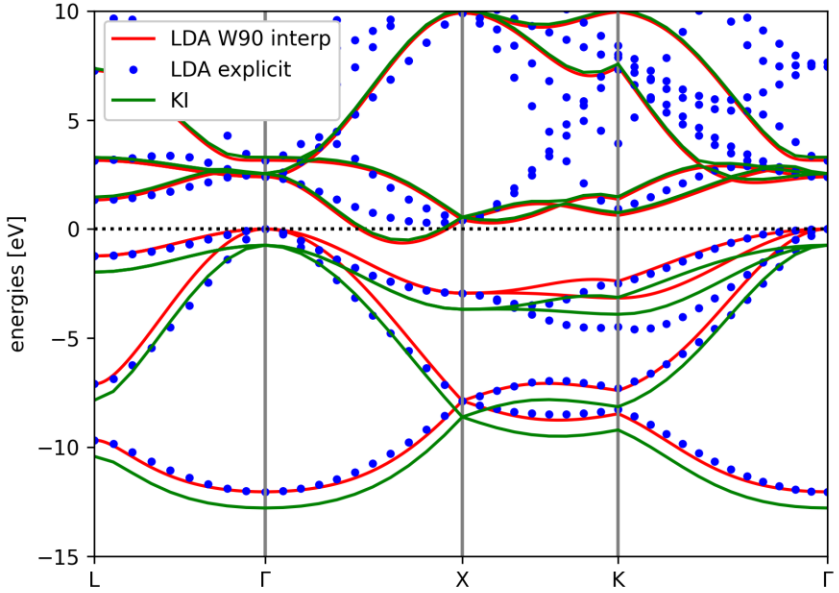
Postproc: smooth interpolation

$$h_{mn}^{KI}(\boldsymbol{k}) = \sum_{\boldsymbol{R}} e^{i\boldsymbol{k}\cdot\boldsymbol{R}} \langle w_{m\boldsymbol{0}} | \hat{h}^{DFT} | w_{n\boldsymbol{R}} \rangle + \sum_{\boldsymbol{R}} e^{i\boldsymbol{k}\cdot\boldsymbol{R}} \langle w_{m\boldsymbol{0}} | \hat{v}^{KI} | w_{n\boldsymbol{R}} \rangle$$

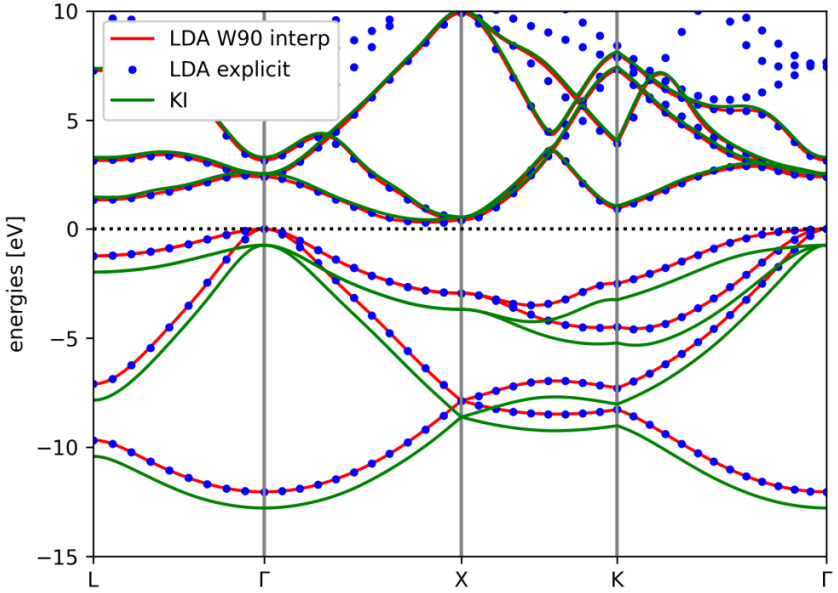
Interpolated on superfine $\boldsymbol{k}$-grid

Computed on the original supercell

without smooth interpolation



with smooth interpolation



Bonus: pKIPZ calculation

With a few changes you can carry out a *perturbative* KIPZ calculation:

- KIPZ Hamiltonian
- KI screening parameters
- No further orbitals optimization

Go back to the main dir of the exercise and change in the input JSON file (`si.json`):

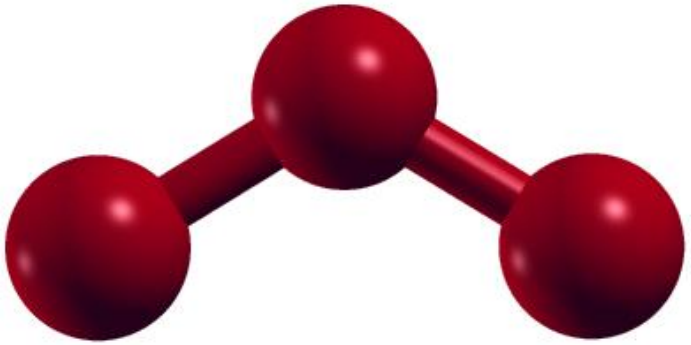
1. `functional` from `ki` to `pkipz`
2. `from_scratch` from `true` to `false`
3. you might want to copy your `postproc` folder and `si.kwf` file somewhere else (or just change name) to avoid them to be overwritten

Run `koopmans si.json` and enjoy your pKIPZ results!

Tutorial #3

Ozone molecule with KCW

Goal of the tutorial:
compute the **charged excitations** of the **ozone** molecule.



Method:

- KI functional (to 2nd order)
- Screening parameters computed from linear response (KCW code)

Main steps:

1. DFT initialization
2. Interface to `kcw`
3. Screening parameters
4. Final KI calculation

WORKDIR = `Day3/exercise3`

1. DFT initialization

```
&CONTROL
   calculation      = 'scf'
   verbosity        = 'high'
   outdir           = '../out'
   prefix           = 'kc'
   pseudo_dir       = '../../../files/pseudo/'
/
&SYSTEM
   ibrav            = 0
   nbnd             = 10
   tot_charge       = 0
   tot_magnetization = 0
   ecutwfc          = 50.0
   nspin            = 2
   assume_isolated  = 'm-t'
   starting_magnetization(1) = 0.0
   ntyp             = 1
   nat              = 3
/
&ELECTRONS
   conv_thr         = 3.6000000000000005e-08
/
ATOMIC_SPECIES
O  15.999 O_ONCV_PBE-1.2.upf

K_POINTS automatic
1 1 1  0 0 0

CELL_PARAMETERS angstrom
8.17380000000000 0.00000000000000 0.00000000000000
0.00000000000000 6.00000000000000 0.00000000000000
0.00000000000000 0.00000000000000 6.66000000000000

ATOMIC_POSITIONS angstrom
O  4.0869000000 3.0000000000 2.8900000000
O  5.1738000000 3.0000000000 3.5500000000
O  3.0000000000 3.0000000000 3.5500000000
```

WORKDIR = `Day3/exercise3/1_init`

executable = `pw.x`

KCW always requires a calculation in which the spin channel are treated separately (even for spin-unpolarized systems). **Always set `nspin = 2` for a meaningful KCW calculation**

Use the Martyna-Tuckerman scheme to deal with spurious periodic replica interactions

KCW does not support Gamma-point trick
Always use complex wavefunctions.

2. Interface to KCW

```
&CONTROL
   prefix            = 'kc'
   outdir            = '../out'
   kcw_iverbosity    = 1
   kcw_at_ks         = .true.
   calculation       = 'wann2kcw'
   mp1               = 1
   mp2               = 1
   mp3               = 1
   assume_isolated   = 'm-t'
   spin_component    = 1
/
```

WORKDIR = `Day3/exercise3/1_init`

executable = `kcw.x`

Specify where to find output files from the DFT initialization

Specify we are going to use KS canonical orbitals as the minimizing one
Reasonable choice for finite systems

Specify the kind of calculation:
`wann2kcw`: interface between PW and W90, and KCW

The Monkhost-Pack grid used in the DFT initialiaztion

Use the Martyna-Tuckerman scheme to deal with spurious periodic replica interactions

Specify which spin channel

https://www.quantum-espresso.org/Doc/INPUT_kcw.html

3. Screening parameters α_i

```
&CONTROL
   prefix            = 'kc'
   outdir            = '../out'
   kcw_iverbosity    = 1
   kcw_at_ks         = .true.
   calculation       = 'screen'
   lrpa              = .false.
   mp1               = 1
   mp2               = 1
   mp3               = 1
   assume_isolated   = 'm-t'
   spin_component    = 1
/
&SCREEN
   tr2               = 1e-14
   nmix              = 4
   niter             = 33
   check_spread      = .true.
/
```

WORKDIR = `Day3/exercise3/2_screening`
executable = `kcw.x`

Specify the kind of calculation:
`screen`: compute the screening coefficients

Set it to `.true.` to neglect xc-effects in the response

Parameters controlling the SCF convergence

Decide whether two orbitals are "equivalent" comparing their spread (self-hartree)
Used to reduce the number of LR calculations

https://www.quantum-espresso.org/Doc/INPUT_kcw.html

3. Screening parameters α_i

Inspect the `kc.kso` output file

Screening coefficients

```
iwann  =    1   relaxed =    0.82001572   unrelaxed =    1.27258121   alpha =  0.64437202   self Hartree =  0.68584472
iwann  =    2   relaxed =    0.76619522   unrelaxed =    1.07137799   alpha =  0.71514930   self Hartree =  0.57687992
iwann  =    3   relaxed =    0.69500288   unrelaxed =    0.90455472   alpha =  0.76833701   self Hartree =  0.48559248
iwann  =    4   relaxed =    0.71813816   unrelaxed =    1.07741830   alpha =  0.66653607   self Hartree =  0.59688898
iwann  =    5   relaxed =    0.69963243   unrelaxed =    0.91616468   alpha =  0.76365358   self Hartree =  0.50592569
iwann  =    6   relaxed =    0.70686887   unrelaxed =    0.98286136   alpha =  0.71919489   self Hartree =  0.53807085
iwann  =    7   relaxed =    0.67502334   unrelaxed =    0.91908165   alpha =  0.73445415   self Hartree =  0.51530769
iwann  =    8   relaxed =    0.68685282   unrelaxed =    0.94106332   alpha =  0.72986886   self Hartree =  0.52320093
iwann  =    9   relaxed =    0.66876849   unrelaxed =    0.87416135   alpha =  0.76504010   self Hartree =  0.47896016
iwann  =   10   relaxed =    0.67685466   unrelaxed =    0.89486563   alpha =  0.75637576   self Hartree =  0.49368862
```

4. Final KI calculation

Diagonalize $\hat{h}^{DFT} + \alpha_i \hat{v}_i^{KI}$

```
&CONTROL
   prefix           = 'kc'
   outdir           = '../out'
   kcw_iverbosity   = 1
   kcw_at_ks        = .true.
   calculation      = 'ham'
   lrpa             = .false.
   mp1              = 1
   mp2              = 1
   mp3              = 1
   assume_isolated  = 'm-t'
   spin_component   = 1
/
&HAM
   do_bands         = .false.
   write_hr         = .true.
/
```

WORKDIR = `Day3/exercise3/3_hamiltonian`
executable = `kcw.x`

All the `&CONTROL` parameters need to be consistent all along the different kcw calculations (`wann2kcw`, `screen`, and `ham`)

Specify the kind of calculation:
`ham`: compute and diagonalize the KI hamiltonian

Write the KI hamiltonian H(**R**) on a formatted file similar to that produced by Wannier90

https://www.quantum-espresso.org/Doc/INPUT_kcw.html

4. Final KI calculation

Inspect the output file: `kc.kho`

```
READING SCREENING PARAMETERS

INFO: alphas read from:../out/kcw/kc.alpha.dat

iwann =     1   alpha =      0.64437202
iwann =     2   alpha =      0.71514930
iwann =     3   alpha =      0.76833701
iwann =     4   alpha =      0.66653607
iwann =     5   alpha =      0.76365358
iwann =     6   alpha =      0.71919489
iwann =     7   alpha =      0.73445415
iwann =     8   alpha =      0.72986886
iwann =     9   alpha =      0.76504010
iwann =    10   alpha =      0.75637576
```

WORKDIR = `Day3/exercise3/3_hamiltonian`
executable = `kcw.x`

File produced by the previous `screen` calculation

4. Final KI calculation

Inspect the output file: `kc.kho`

```
READING SCREENING PARAMETERS

INFO: alphas read from:../out/kcw/kc.alpha.dat

iwann =     1   alpha =      0.64437202
iwann =     2   alpha =      0.71514930
iwann =     3   alpha =      0.76833701
iwann =     4   alpha =      0.66653607
iwann =     5   alpha =      0.76365358
iwann =     6   alpha =      0.71919489
iwann =     7   alpha =      0.73445415
iwann =     8   alpha =      0.72986886
iwann =     9   alpha =      0.76504010
iwann =    10   alpha =      0.75637576
```

WORKDIR = `Day3/exercise3/3_hamiltonian`
executable = `kcw.x`

File produced by the previous `screen` calculation

Alternatively one can specify the alpha parameters in a file named `file_alpharef.txt` to be placed in the working directory (this has the priority)

```
10
 1    0.644372016025    0.685844719722
 2    0.715149297272    0.576879918554
 3    0.768337013151    0.485592479713
 4    0.666536067994    0.596888975285
 5    0.763653579131    0.505925686597
 6    0.719194888715    0.538070848354
 7    0.734454152397    0.515307690616
 8    0.729868860603    0.523200928600
 9    0.765040102518    0.478960160670
10    0.756375756994    0.493688617998
```

Total number of orbitals

Screening Coefficients

Self-Hartree

Final KI calculation

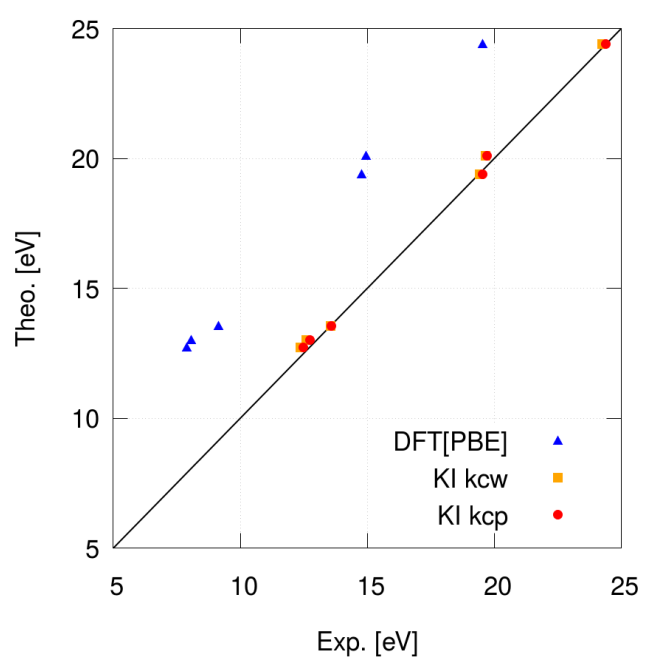
Inspect the output file: `kc.kho`

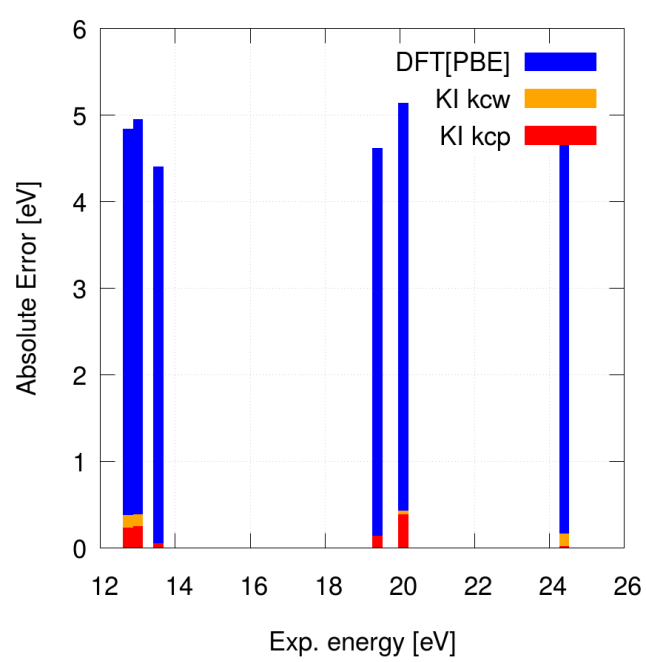
```
KI[Full]    -40.2080   -32.9030   -24.2399   -19.6693   -19.4346   -19.1930   -13.5610   -12.6129
KI[Full]    -12.3542    -1.4999
```

```
KI[full] highest occupied, lowest unoccupied level (ev):   -12.3542   -1.4999
```

For a more direct comparison KI-vs-DFT you can run the two scripts:

```
> sh get_BE.sh
> gnuplot BE.gnu
```





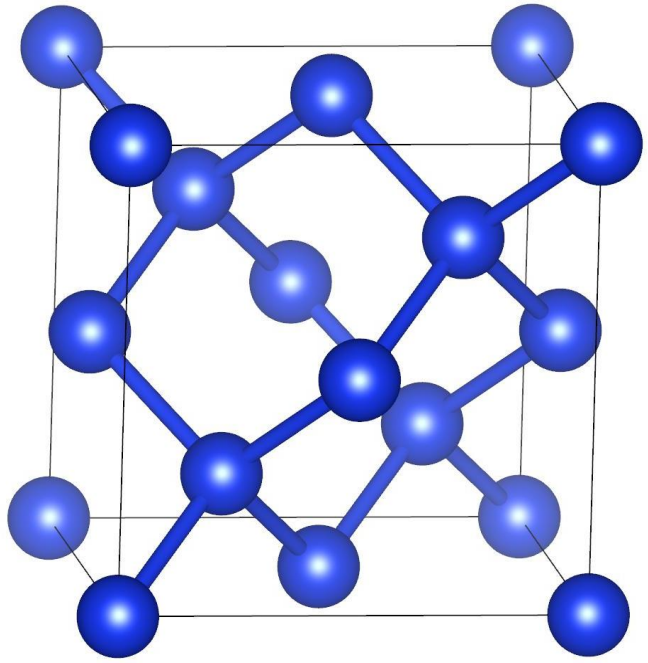
Tutorial #4

Bulk silicon with KCW

Goal of the tutorial:
compute the **band structure** of silicon.

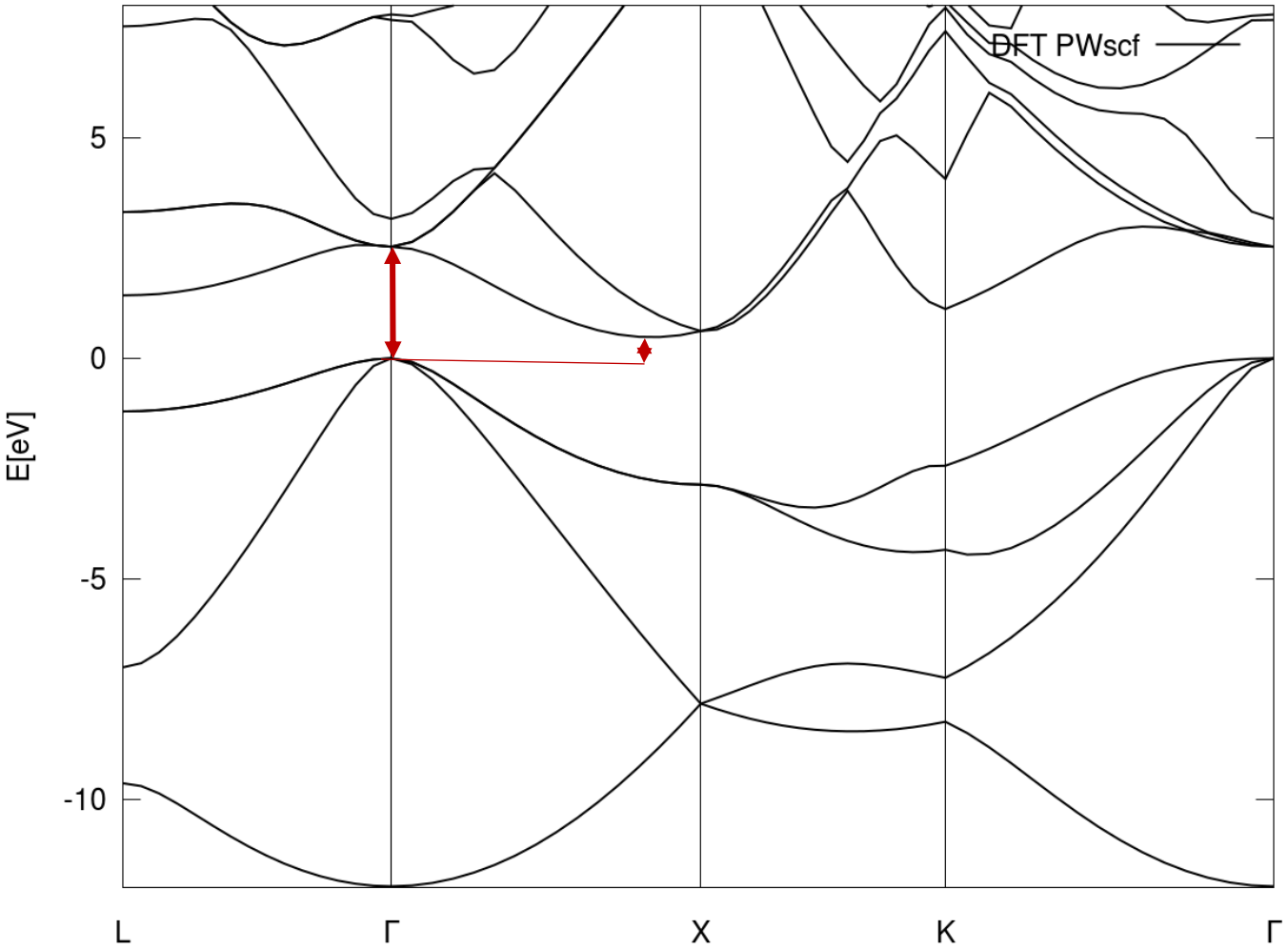
Method:
- KI
- Screening parameters computed from linear response (KCW code)

Main steps:
1. DFT calculation
2. Maximally localized Wannier functions
3. Screening parameters
4. Final KI calculation
5. Band structure interpolation with Wannier90



WORKDIR = `Day3/exercise4`

0. LDA band structure



WORKDIR = `Day3/exercise4/0_dft`
executables = `pw.x` and `bands.x`

	indirect	direct
E_g LDA.	0.48	2.53
E_g Exp.	1.17	3.35

1. DFT initialization

SCF calculation

```
&CONTROL
   calculation      = 'scf'
   verbosity        = 'high'
   outdir           = '../out'
   pseudo_dir       = '../../../files/pseudo/'
   prefix           = 'si'
/
&SYSTEM
   ibrav            = 2
   celldm(1)        = 10.262200042863787
   tot_magnetization = 0
   ecutwfc          = 20.0
   nspin            = 2
   starting_magnetization(1) = 0.0
   ntyp             = 1
   nat              = 2
/
&ELECTRONS
   conv_thr         = 1.6e-08
/

ATOMIC_SPECIES
Si  28.085 Si.upf

K_POINTS automatic
6 6 6  0 0 0

ATOMIC_POSITIONS crystal
Si  -0.0000000000 0.0000000000 -0.0000000000
Si  0.2500000000 0.2500000000 0.2500000000
```

WORKDIR = `Day3/exercise4/1_init`

executable = `pw.x`

Always set `nspin = 2` for a meaningful KCW calculation

NSCF calculation

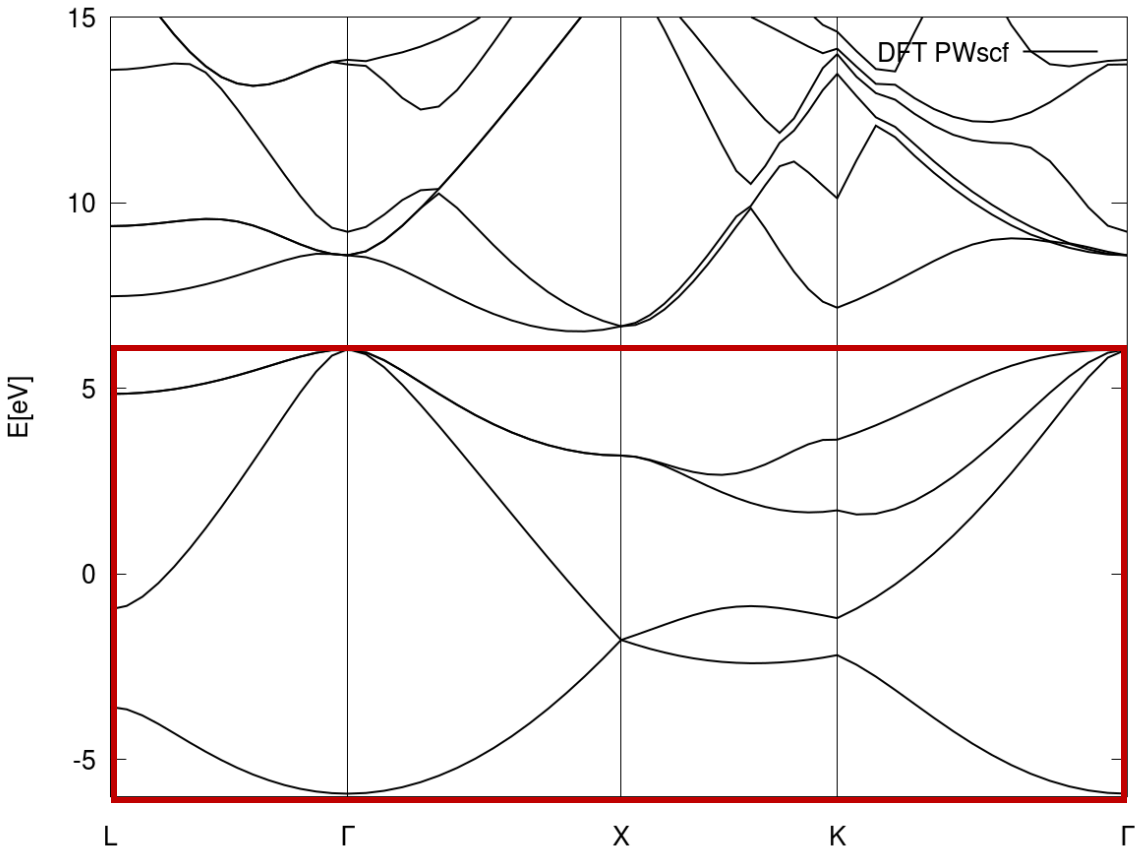
```
&CONTROL
   calculation      = 'nscf'
/
&SYSTEM
  ...
/
&ELECTRONS
  ...
/

ATOMIC_SPECIES
Si  28.085 Si.upf

ATOMIC_POSITIONS crystal
Si  -0.0000000000 0.0000000000 -0.0000000000
Si  0.2500000000 0.2500000000 0.2500000000

K_POINTS crystal
216
  0.00000000  0.00000000  0.00000000  4.629630e-03
  0.00000000  0.00000000  0.16666667  4.629630e-03
  0.00000000  0.00000000  0.33333333  4.629630e-03
  0.00000000  0.00000000  0.50000000  4.629630e-03
  ...
```

NSCF on a regular 6x6x6 k-point grid (you can use the `kmesh.pl` utility of W90)

2. Wannierization

WORKDIR = `Day3/exercise4/1_init/occ`
executable = `wannier90.x, pw2wannier90.x`

Occupied Manifold

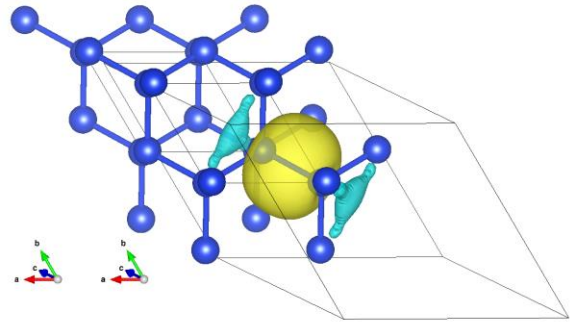



```
begin projections
  f=0.25,0.25,0.25:sp3
end projections
num_wann = 4
num_bands = 4
exclude_bands = 5-20
mp_grid = 6 6 6
```

4 MLWFs out of 4 bands

Exclude all the empty states

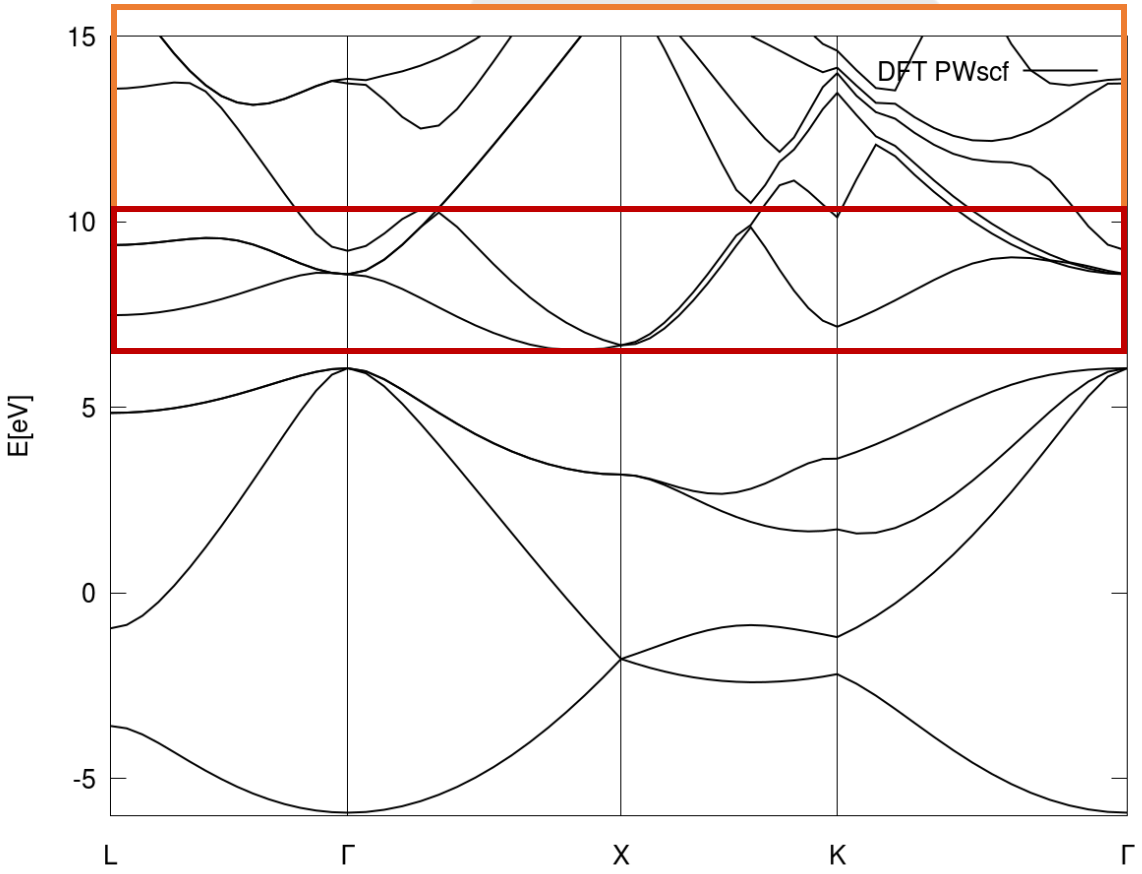
Use 4 sp^3 atomic orbital centered at (0.25, 0.25, 0.25) as initial guess



2. Wannierization

WORKDIR = Day3/exercise4/1_init/occ
executable = wannier90.x, pw2wannier90.x

Empty Manifold

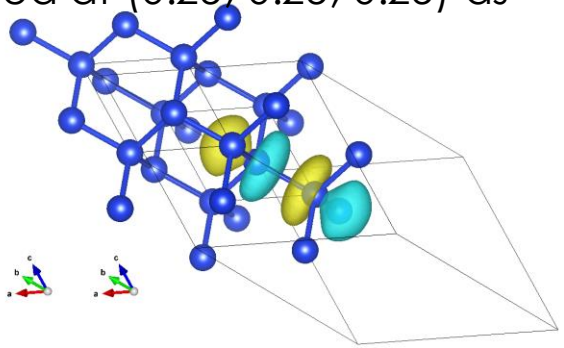



```
begin projections
  f=0.25,0.25,0.25:sp3
end projections
dis_froz_max = 10.6
dis_win_max = 16.9
num_wann = 4
num_bands = 16
exclude_bands = 1-4
mp_grid = 6 6 6
```

4 MLWFs out of 16 bands (disentanglement needed!)

Exclude all the occupied states

Use 4 sp^3 atomic orbital centered at (0.25, 0.25, 0.25) as initial guess



2. Interface to KCW

```
&CONTROL
   prefix            = 'si'
   outdir            = '../out'
   kcw_iverbosity    = 1
   kcw_at_ks         = .false.
   calculation       = 'wann2kcw'
   mp1               = 6
   mp2               = 6
   mp3               = 6
   read_unitary_matrix = .true.
   l_vcut            = .true.
   spin_component    = 1
/
&WANNIER
   seedname          = 'wann'
   check_ks          = .true.
   num_wann_occ      = 4
   num_wann_emp      = 4
   have_empty        = .true.
   has_disentangle   = .true.
/
```

WORKDIR = `Day3/exercise4/1_init`

executable = `kcw.x`

Specify where to find output files from the DFT initialization

Specify we are going to use something different from KS orbitals as the minimizing ones.
This will be MLWF: reasonable choice for extended systems

Use the Gygi-Baldereschi scheme to deal with the long-range coulomb interactions

Specify where to find output files from W90 (essentially the unitary matrices)

Number of occ and empty MLWFs

https://www.quantum-espresso.org/Doc/INPUT_kcw.html

3. Screening parameters α_i

```
&CONTROL
   prefix           = 'si'
   outdir           = '../out'
   kcw_iverbosity   = 1
   kcw_at_ks        = .false.
   calculation      = 'screen'
   lrpa             = .false.
   mp1              = 2
   mp2              = 2
   mp3              = 2
   read_unitary_matrix = .true.
   l_vcut           = .true.
   spin_component   = 1
/
&WANNIER
   seedname         = 'wann'
   check_ks         = .true.
   num_wann_occ     = 4
   num_wann_emp     = 4
   have_empty       = .true.
   has_disentangle  = .true.
/
&SCREEN
   tr2              = 1e-14
   nmix             = 4
   niter            = 33
   check_spread     = .true.
/
```

WORKDIR = `Day3/exercise4/2_screening`

executable = `kcw.x`

Screening coefficients on a reduced k-mesh (requires to re-do the Wannierization on the reduced mesh)

Parameters controlling the SCF convergence

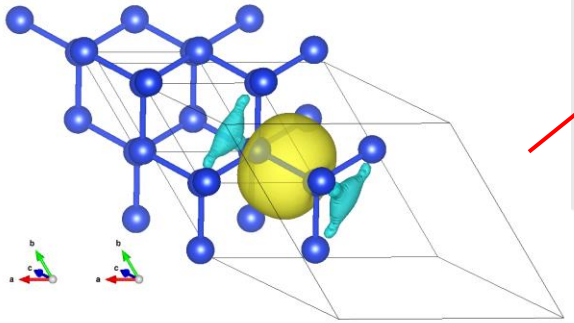
https://www.quantum-espresso.org/Doc/INPUT_kcw.html

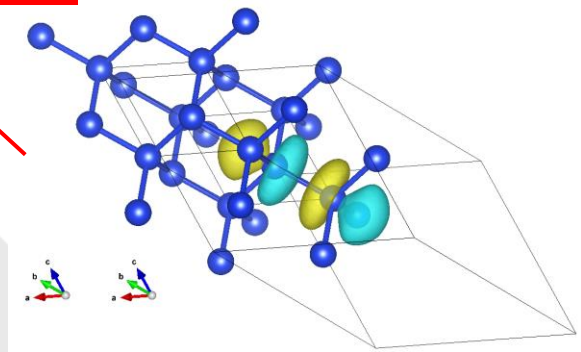
3. Screening parameters α_i

Inspect the `kc.kso` output file

Screening coefficients

```
iwann  =    1   relaxed =    0.13192255   unrelaxed =    0.86937149   alpha =  0.15174474   self Hartree =  0.50016280
iwann* =    2   relaxed =    0.13192255   unrelaxed =    0.86937149   alpha =  0.15174474   self Hartree =  0.50016280
iwann* =    3   relaxed =    0.13192255   unrelaxed =    0.86937149   alpha =  0.15174474   self Hartree =  0.50016280
iwann* =    4   relaxed =    0.13192255   unrelaxed =    0.86937149   alpha =  0.15174474   self Hartree =  0.50016280
iwann  =    5   relaxed =    0.01934112   unrelaxed =    0.53054776   alpha =  0.03645499   self Hartree =  0.28835881
iwann* =    6   relaxed =    0.01934112   unrelaxed =    0.53054776   alpha =  0.03645499   self Hartree =  0.28835881
iwann* =    7   relaxed =    0.01934112   unrelaxed =    0.53054776   alpha =  0.03645499   self Hartree =  0.28835881
iwann* =    8   relaxed =    0.01934112   unrelaxed =    0.53054776   alpha =  0.03645499   self Hartree =  0.28835881
```





4. Final KI calculation

Diagonalize $\hat{h}^{DFT} + \alpha_i \hat{v}_i^{KI}$

```
&CONTROL
   ...
/
&WANNIER
   ...
/
&HAM
   do_bands          = .true.
   use_ws_distance   = .true.
   write_hr          = .true.
/
K_POINTS crystal_b
5
 0.50000   0.50000   0.50000 15 ! L
 0.00000   0.00000   0.00000 15 ! G
 0.50000   0.00000   0.50000 15 ! X
 0.37500   0.37500   0.75000 15 ! K
 0.00000   0.00000   0.00000  1 !G
```

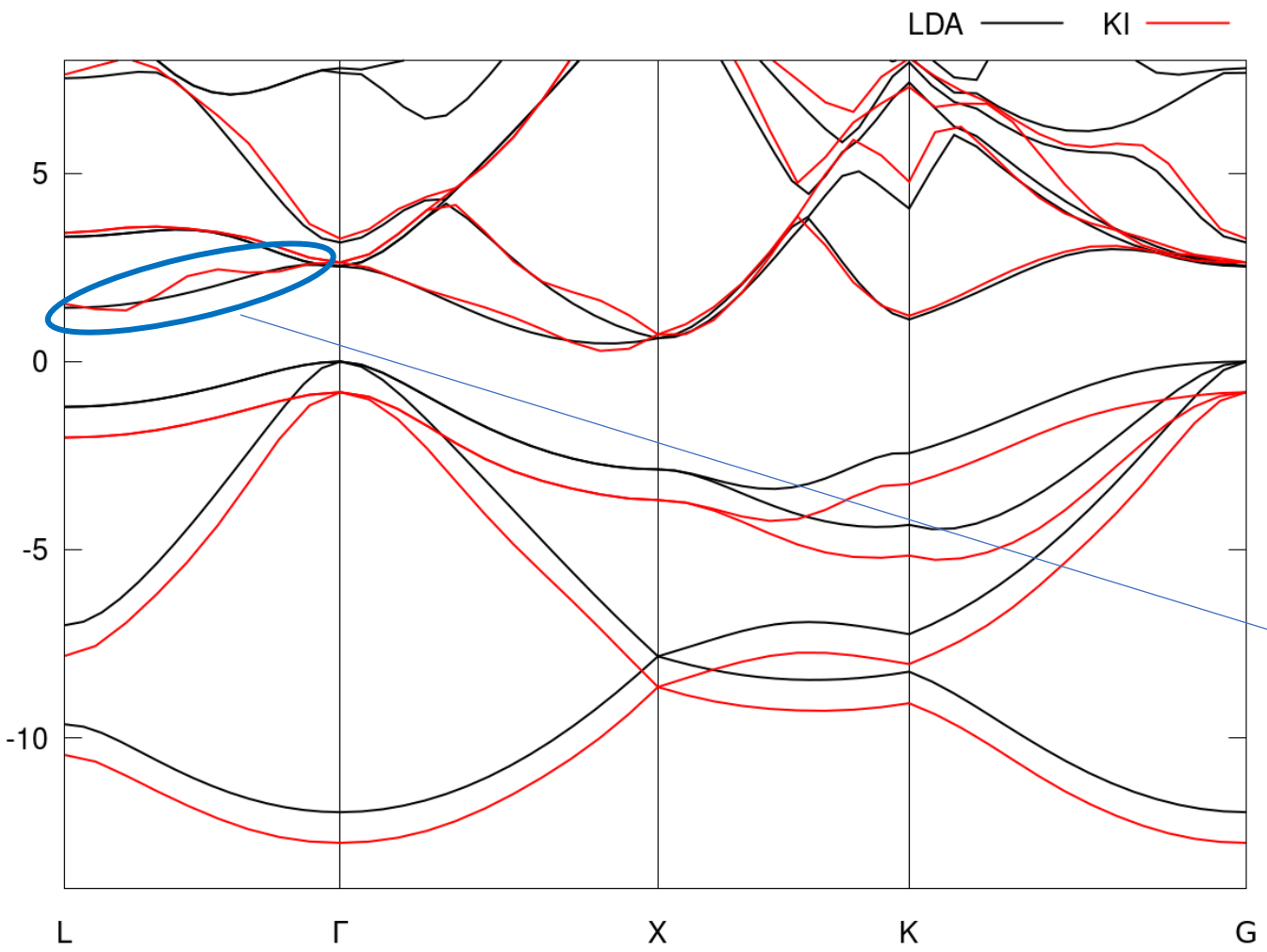
WORKDIR = `Day3/exercise4/3_hamiltonian`
executable = `kcw.x`

All the `&CONTROL` parameters need to be consistent all along the different kcw calculations (`wann2kcw`, `screen`, and `ham`)

Use the MLWFs to interpolate the band structure on an arbitrary path

https://www.quantum-espresso.org/Doc/INPUT_kcw.html

4. Final KI calculation



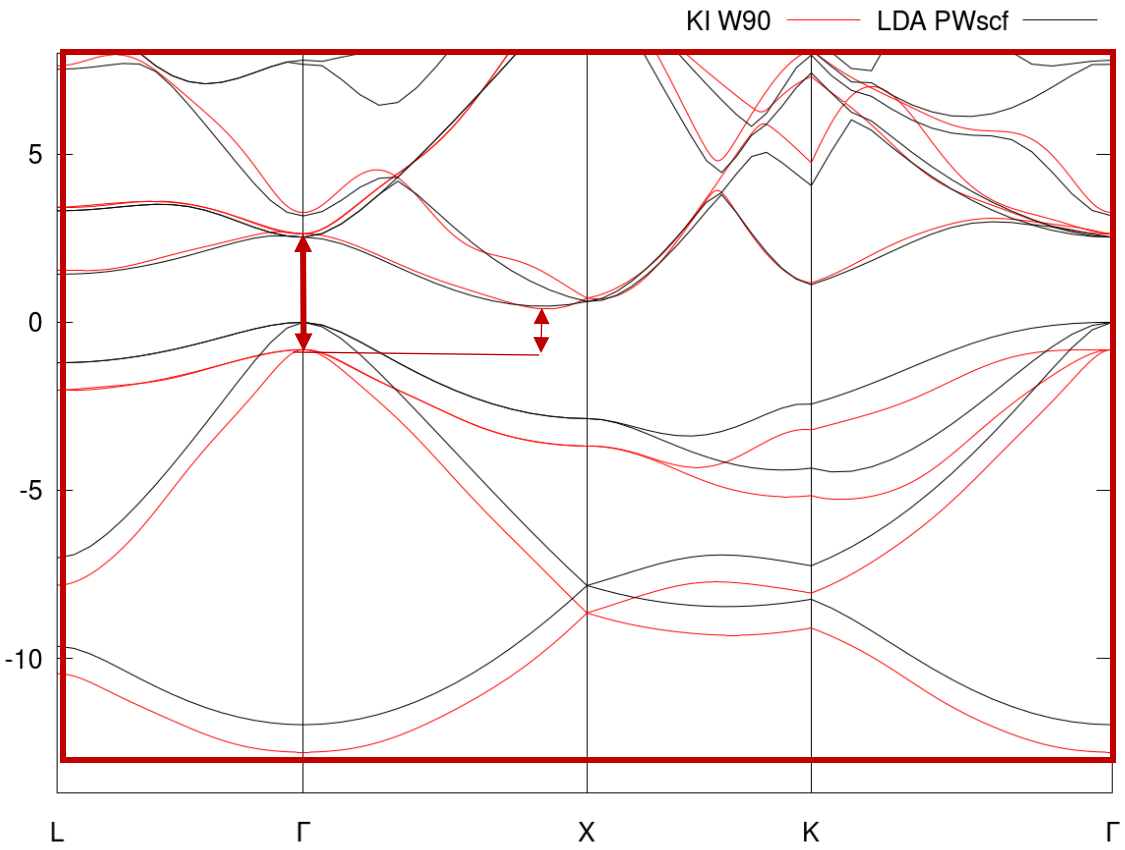
WORKDIR = `Day3/exercise4/3_hamiltonian`
executable = `kcw.x`

	indirect	direct
E_g LDA.	0.48	2.53
E_g KI.	??	3.47
E_g Exp.	1.17	3.35

Better interpolation with an extra unique (occ+empty) wannierization

5. Wannierization (post-proc)

WORKDIR = `Day3/exercise4/3_hamiltonian/wannier_post`
executable = `wannier90.x, pw2wannier90.x`




```
begin projections
  random
end projections
num_wann = 8
num_bands = 8
mp_grid = 6 6 6
```

8MLWFs out of 8 bands
(NO disentanglement needed!)

Use randomly-centred s-type Gaussian function as initial guess

	indirect	direct
E_g LDA.	0.48	2.53
E_g KI.	1.30	3.45
E_g Exp.	1.17	3.35

Even better interpolation possible using the smooth-interpolation technique (see tutorial #2)

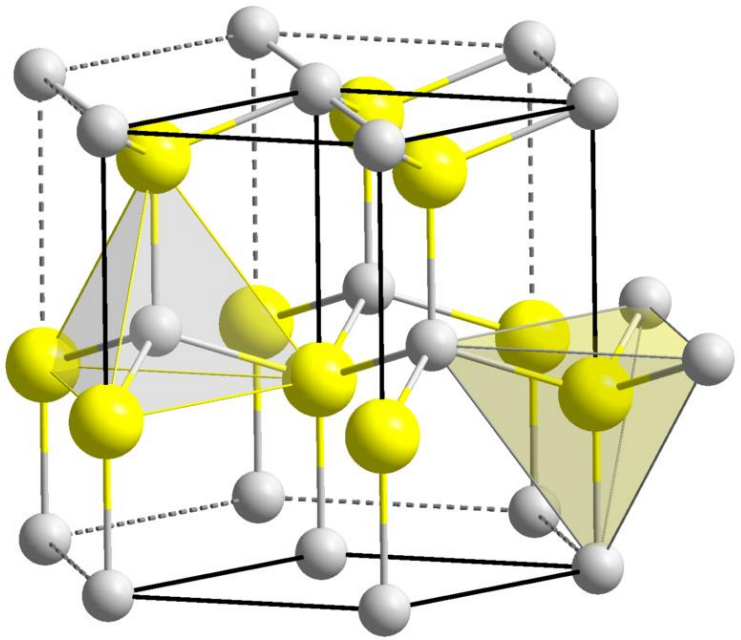
Tutorial #5

Wurtzite ZnO with KCW

Goal of the tutorial:
compute the **band structure** of ZnO

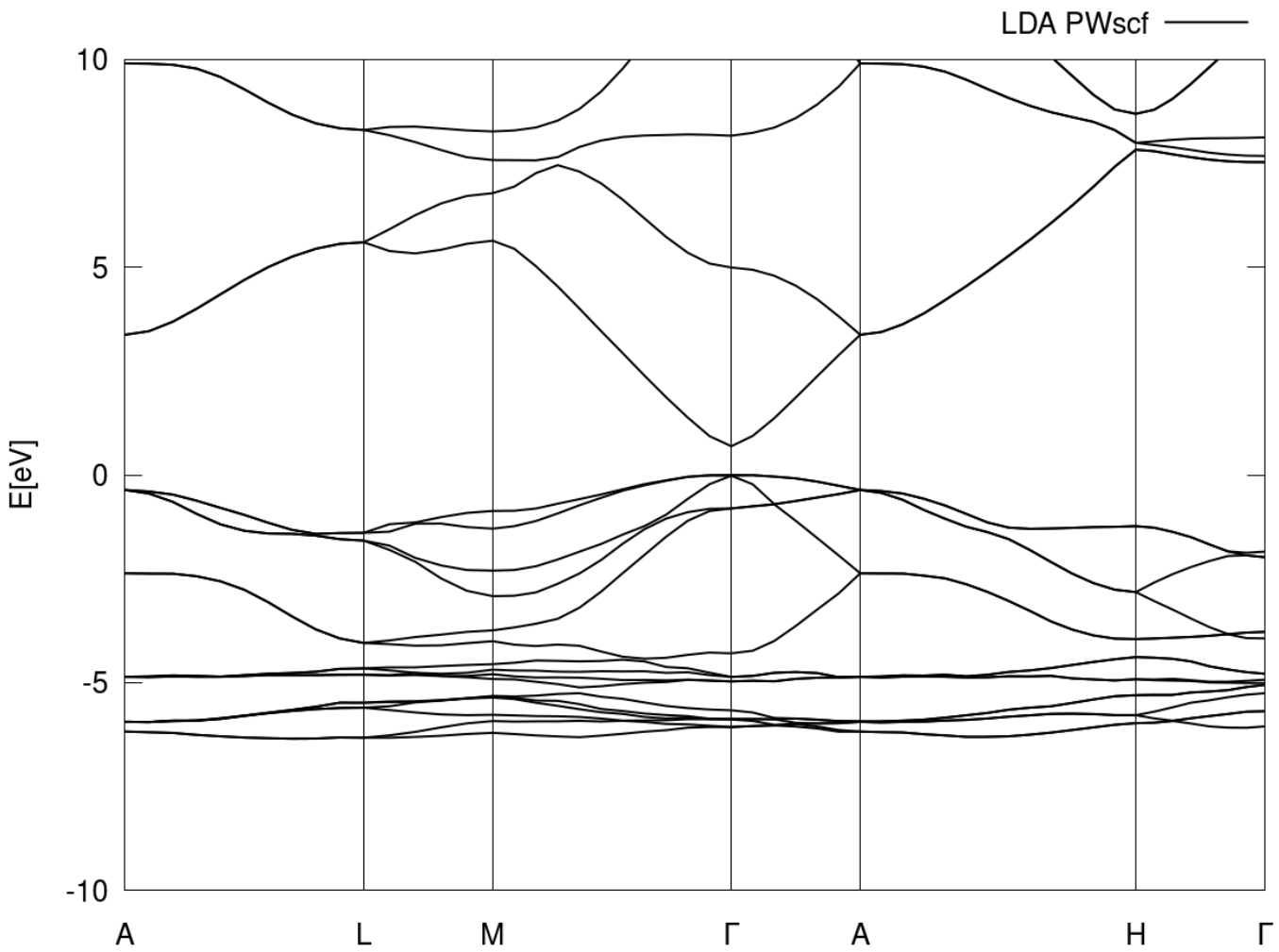
Method:
- KI
- Screening parameters computed from linear response (KCW code)

Main steps:
1. DFT calculation
2. Projected Wannier functions
3. Screening parameters
4. Final KI calculation



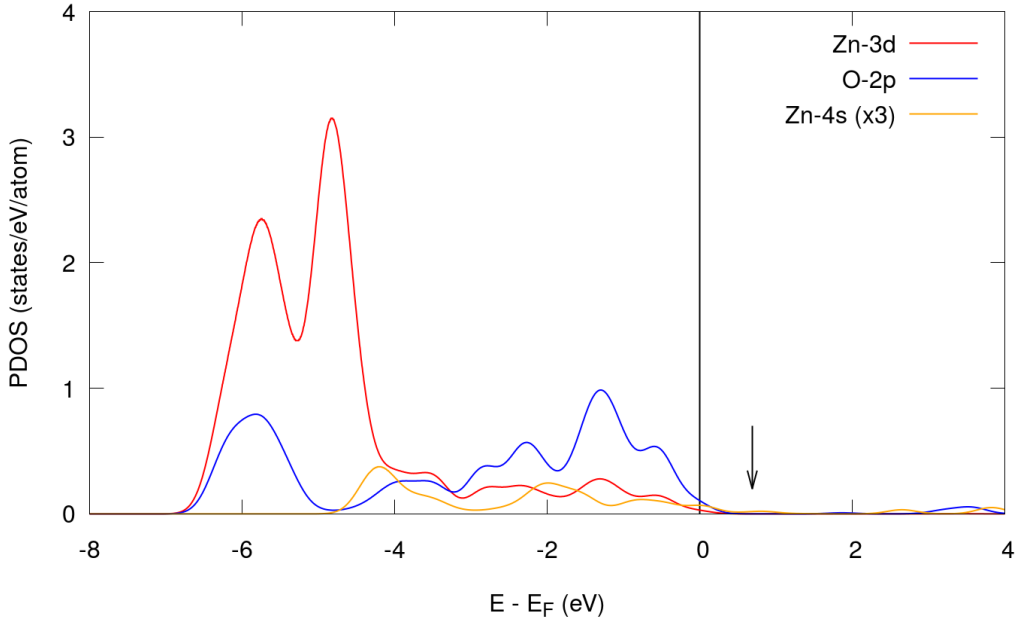
WORKDIR = `Day3/exercise5`

0. LDA band structure



WORKDIR = `Day3/exercise5/0_dft`
executables = `pw.x` and `bands.x`

	Gap	d-states
LDA.	0.70	-5.1
Exp.	3.60	-7.5 / -8.0



1. DFT initialization

WORKDIR = Day3/exercise5/1_init
executable = pw.x

SCF calculation

```
&CONTROL
   calculation      = 'scf'
   verbosity        = 'high'
   prefix           = 'kc'
   outdir           = '../out'
   pseudo_dir       = '../../../files/pseudo'
/
&SYSTEM
   ecutwfc          = 50.0
   nspin            = 2
   tot_magnetization = 0
   starting_magnetization(1) = 0.0
   starting_magnetization(2) = 0.0
   ntyp             = 2
   nat              = 4
   ibrav            = 0
/
&ELECTRONS
   conv_thr         = 1.04e-07
/

ATOMIC_SPECIES
Zn  65.38 Zn.upf
O  15.999 O.upf

K_POINTS automatic
4 4 4  0 0 0

CELL_PARAMETERS angstrom
-1.62472500000000 -2.81410624830000 0.00000000000000
-1.62472500000000 2.81410624830000 0.00000000000000
0.00000000000000 0.00000000000000 -5.20574000000000

ATOMIC_POSITIONS crystal
Zn  0.3333000000 0.6667000000 0.5000000000
Zn  0.6667000000 0.3333000000 -0.0000000000
O  0.3333000000 0.6667000000 0.1172500000
O  0.6667000000 0.3333000000 0.6172500000
```

Always set `nspin = 2` for a meaningful KCW calculation

NSCF calculation

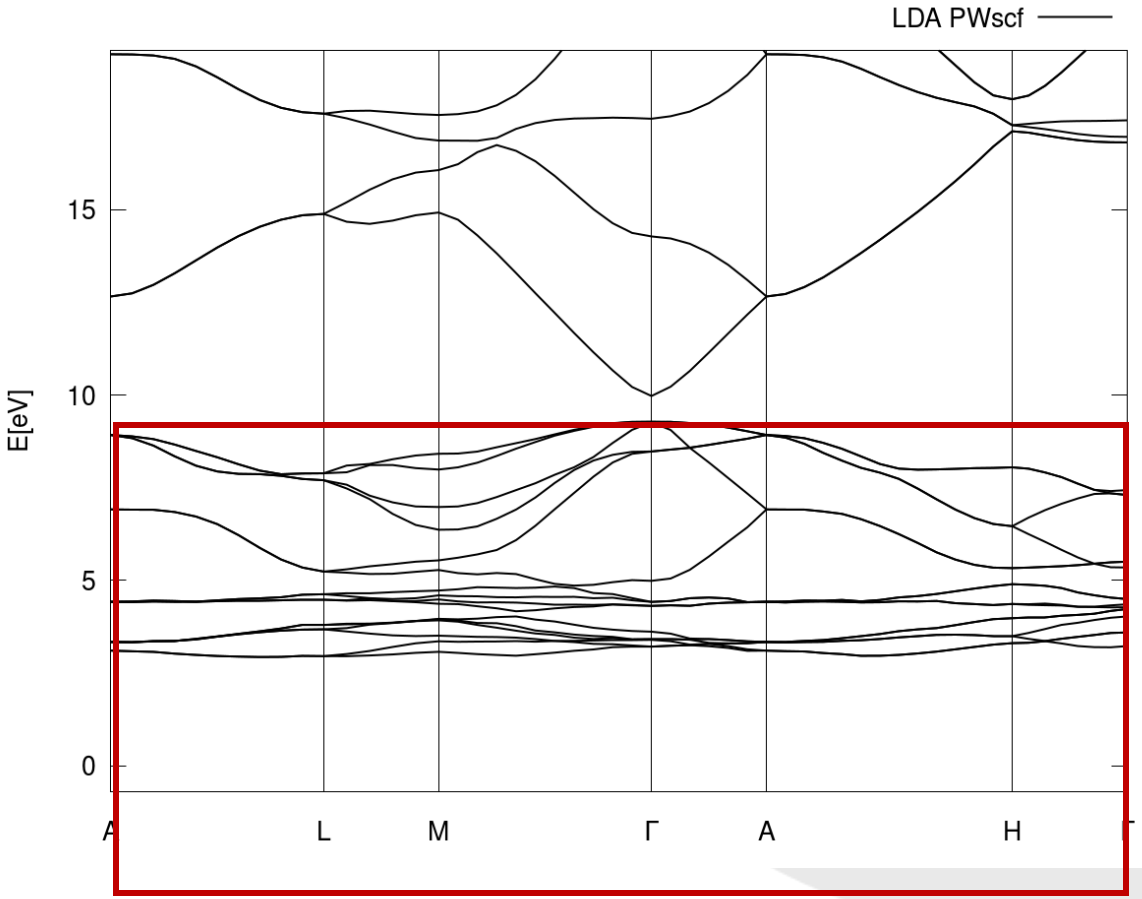
```
&CONTROL
   calculation      = 'nscf'
/
&SYSTEM
   nbnd             = 52
   nosym            = .true.
   noinv            = .true.
/
&ELECTRONS
   ...
/

ATOMIC_SPECIES
Zn  65.38 Zn.upf
O  15.999 O.upf

K_POINTS automatic
4 4 4  0 0 0

CELL_PARAMETERS angstrom
 ...

ATOMIC_POSITIONS crystal
 ...
```

NSCF on a regular 4x4x4 k-point grid (disabling all the symmetries)

2. Wannierization

WORKDIR = Day3/exercise5/1_init/occ
executable = wannier90.x, pw2wannier90.x

Occupied Manifold




```
num_iter = 0
begin projections
  f=0.3333,0.6667,0.50000 :l=0
  f=0.3333,0.6667,0.50000 :l=1
  f=0.3333,0.6667,0.50000 :l=2
  f=0.6667,0.3333,0.00000 :l=0
  f=0.6667,0.3333,0.00000 :l=1
  f=0.6667,0.3333,0.00000 :l=2
  f=0.3333,0.6667,0.11725 :l=0
  f=0.3333,0.6667,0.11725 :l=1
  f=0.6667,0.3333,0.61725 :l=0
  f=0.6667,0.3333,0.61725 :l=1
end projections

num_wann = 26
num_bands = 26
exclude_bands = 27-52
mp_grid = 4 4 4
```

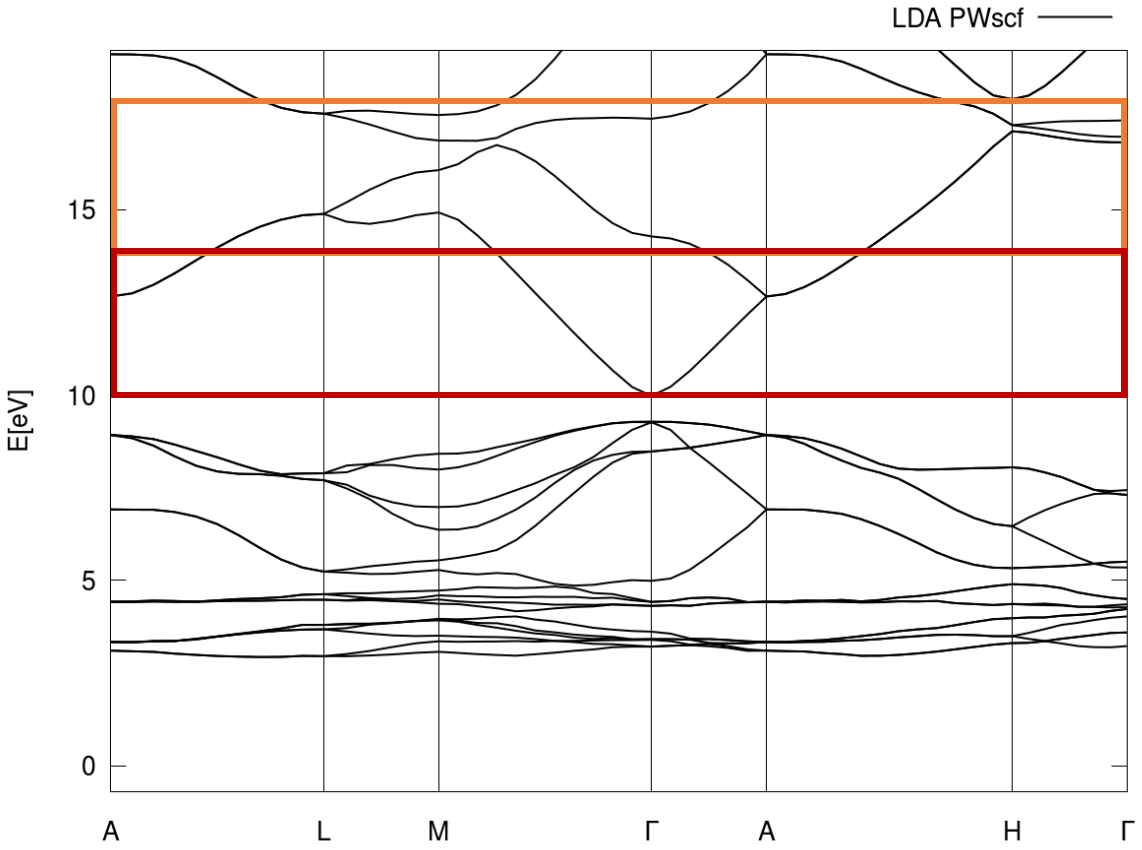
No minimization to avoid orbital mixing

Exclude all the empty states

2. Wannierization

WORKDIR = Day3/exercise4/1_init/occ
executable = wannier90.x, pw2wannier90.x

Empty Manifold




```
begin projections
  f = 0.3333, 0.6667, 0.5000 :l=0
  f = 0.6667, 0.3333, 0.0000 :l=0
end projections
dis_froz_max = 14.5
dis_win_max = 17.0
num_wann = 2
num_bands = 26

exclude_bands = 1-26
mp_grid = 4 4 4
```

2 Projected WFs out of 26 bands (disentanglement needed!)

Exclude all the occupied states

Use 2 s atomic orbital centered on the Zn atoms as initial guess

2. Interface to KCW

```
&CONTROL
   prefix            = 'kc'
   outdir            = '../out'
   kcw_iverbosity    = 1
   kcw_at_ks         = .false.
   calculation       = 'wann2kcw'
   lrpa              = .false.
   homo_only         = .false.
   read_unitary_matrix = .true.
   l_vcut            = .true.
/
&WANNIER
   seedname          = 'wann'
   check_ks          = .true.
   num_wann_occ      = 26
   num_wann_emp      = 2
   have_empty        = .true.
   has_disentangle   = .true.
/
```

WORKDIR = `Day3/exercise5/1_init`

executable = `kcw.x`

Specify where to find output files from the DFT initialization

Specify we are going to use something different from KS orbitals as the minimizing ones.
This will be MLWF: reasonable choice for extended systems

Use the Gygi-Baldereschi scheme to deal with the long-range coulomb interactions

Specify where to find output files from W90 (essentially the unitary matrices)

Number of occ and empty MLWFs

https://www.quantum-espresso.org/Doc/INPUT_kcw.html

3. Screening parameters α_i

```
&CONTROL
   prefix            = 'kc'
   outdir            = '../out'
   kcw_iverbosity    = 1
   kcw_at_ks         = .false.
   calculation       = 'screen'
   lrpa              = .false.
   mp1               = 4
   mp2               = 4
   mp3               = 4
   read_unitary_matrix = .true.
   l_vcut            = .true.
   spin_component    = 1
/
&WANNIER
   seedname          = 'wann'
   check_ks          = .true.
   num_wann_occ      = 26
   num_wann_emp      = 2
   have_empty        = .true.
   has_disentangle   = .true.
/
&SCREEN
   tr2               = 1e-18
   nmix              = 4
   niter             = 33
   eps_inf           = 5.28
   check_spread      = .true.
/
```

WORKDIR = `Day3/exercise5/2_screening`

executable = `kcw.x`

Screening coefficients on a reduced k-mesh (requires to re-do the Wannierization on the reduced mesh)

Parameters controlling the SCF convergence

Use the value of the macroscopic dielectric function to improve the convergence wrt k/q-point sampling `eps_inf` computed with `ph.x`

https://www.quantum-espresso.org/Doc/INPUT_kcw.html

3. Screening parameters α_i

Inspect the `kc.kso` output file

Screening coefficients

```
iwann  =    1   relaxed =     0.87843125   unrelaxed =     2.45386354   alpha =  0.35797885   self Hartree =  1.28839450
iwann  =    2   relaxed =     0.81529427   unrelaxed =     2.23902891   alpha =  0.36412851   self Hartree =  1.20757411
iwann  =    3   relaxed =     0.81605442   unrelaxed =     2.24199291   alpha =  0.36398617   self Hartree =  1.20911418
iwann* =    4   relaxed =     0.81529427   unrelaxed =     2.23902891   alpha =  0.36412851   self Hartree =  1.20757411
iwann  =    5   relaxed =     0.59809671   unrelaxed =     1.67495743   alpha =  0.35708174   self Hartree =  0.91532024
iwann  =    6   relaxed =     0.60247901   unrelaxed =     1.68417736   alpha =  0.35772895   self Hartree =  0.92078535
iwann  =    7   relaxed =     0.60238148   unrelaxed =     1.68391723   alpha =  0.35772630   self Hartree =  0.92063443
iwann  =    8   relaxed =     0.60048766   unrelaxed =     1.68075860   alpha =  0.35727180   self Hartree =  0.91868906
...
```

4. Final KI calculation

Diagonalize $\hat{h}^{DFT} + \alpha_i \hat{v}_i^{KI}$

```
&CONTROL
   ...
/
&WANNIER
   ...
/
&HAM
   do_bands          = .true.
   use_ws_distance   = .true.
   write_hr          = .true.
/
K_POINTS crystal_b
 7
 0.00000   0.00000   0.50000 10 ! A
 0.50000   0.00000   0.50000 10 ! L
 0.50000   0.00000   0.00000 10 ! M
 0.00000   0.00000   0.00000 10 ! G
 0.00000   0.00000   0.50000 10 ! A
 0.33333   0.33333   0.50000 10 ! H
 0.33333   0.33333   0.00000  1 ! K
```

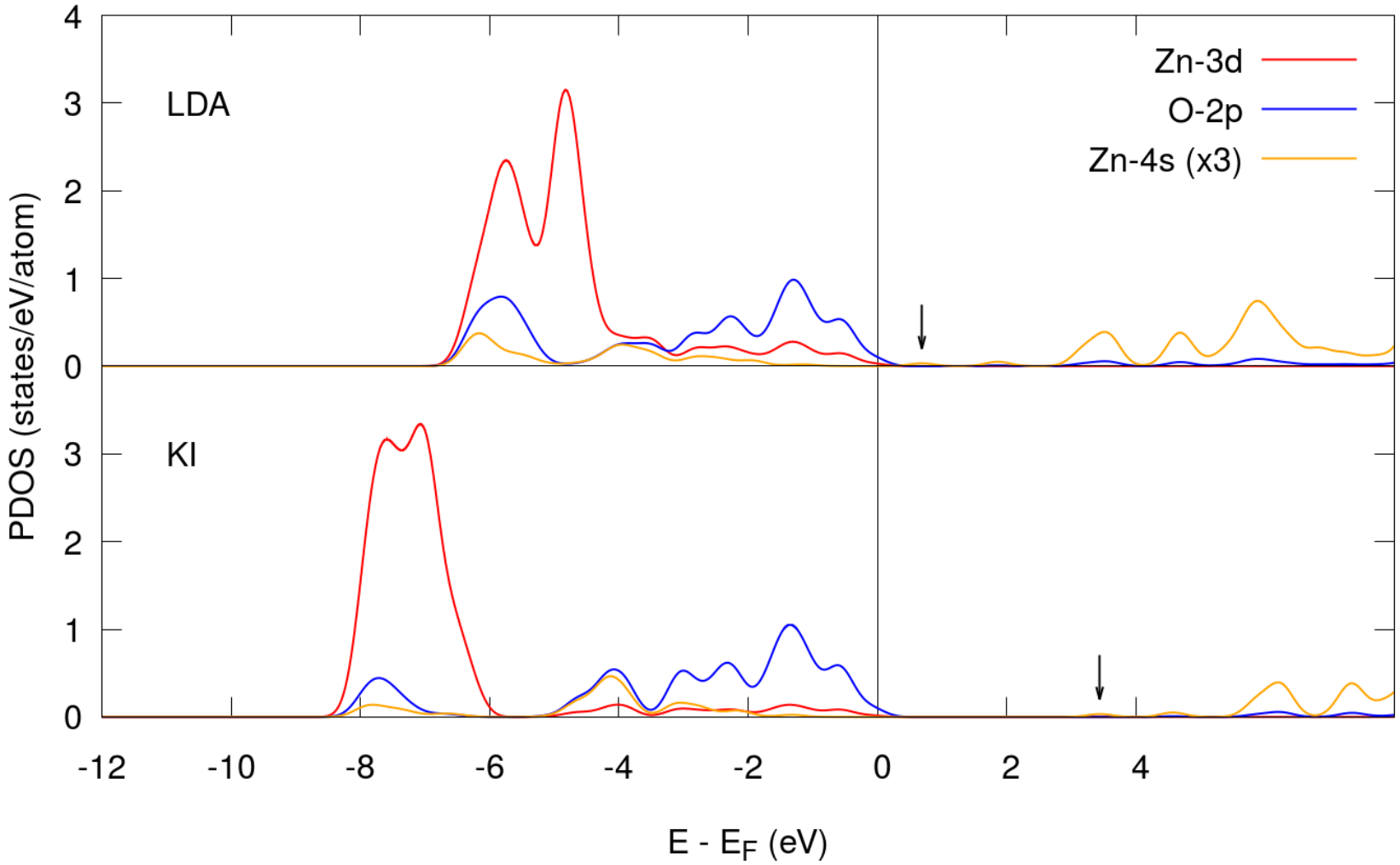
WORKDIR = `Day3/exercise5/3_hamiltonian`
executable = `kcw.x`

All the `&CONTROL` parameters need to be consistent all along the different kcw calculations (`wann2kcw`, `screen`, and `ham`)

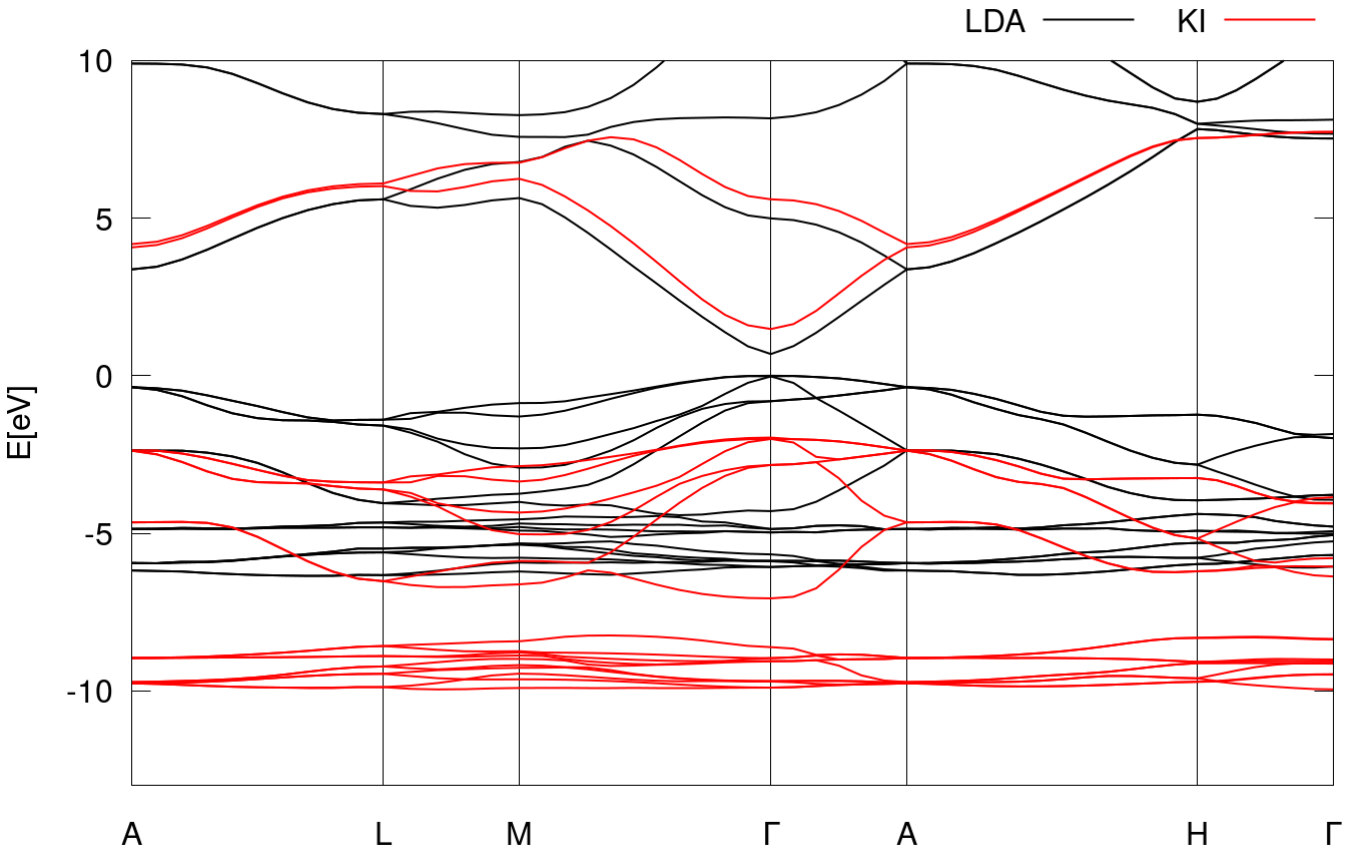
Use the MLWFs to interpolate the band structure on an arbitrary path

https://www.quantum-espresso.org/Doc/INPUT_kcw.html

4. Final KI calculation



4. Final KI calculation



WORKDIR = `Day3/exercise5/3_hamiltonian`
executable = `kcw.x`

	Gap	d-states
LDA.	0.70	-5.1
KI	3.4	-7.5
Exp.	3.60	-7.5 / -8.0

5. Using koopmans

Now, instead of running everything by hand, let's use `koopmans`

In the directory `Day3/exercise5/automated` you will find the input JSON file

```
{
    "workflow": {
        "task": "singlepoint",
        "functional": "ki",
        "base_functional": "lda",
        "method": "dfpt",
        "init_orbitals": "mlwfs",
        "calculate_alpha" : false,
        "alpha_guess": [[0.3580, 0.3641, 0.3640, 0.3641, 0.3571,
0.3577, 0.3577, 0.3573, 0.3573, 0.3580, 0.3641, 0.3640, 0.3641,
0.3571, 0.3577, 0.3577, 0.3573, 0.3573, 0.2158, 0.2323, 0.2344,
0.2343, 0.2158, 0.2323, 0.2344, 0.2343, 0.2231, 0.2231]],
        "pseudo_library": "pseudo_dojo_standard",
        "gb_correction" : true,
        "eps_inf": 5.3,
        "from_scratch": true,
        "npool": 1
    },
```

Now use DFPT compared to the finite difference approach

For the sake of time we'll skip the calculation of the screening parameters and instead provide some pre-computed values here

5. Using koopmans

As before, run `koopmans zno.json` to run the entire workflow
You will see some familiar steps...

```
 Wannierization
 ==============
  Running wannier/scf... done
  Running wannier/nscf... done
  Running wannier/occ_block1/wann_preproc... done
  Running wannier/occ_block1/pw2wan... done
  Running wannier/occ_block1/wann... done
  Running wannier/occ_block2/wann_preproc... done
  Running wannier/occ_block2/pw2wan... done
  Running wannier/occ_block2/wann... done
  Running wannier/occ_block3/wann_preproc... done
  Running wannier/occ_block3/pw2wan... done
  Running wannier/occ_block3/wann... done
  Running wannier/occ_block4/wann_preproc... done
  Running wannier/occ_block4/pw2wan... done
  Running wannier/occ_block4/wann... done
  Running wannier/emp/wann_preproc... done
  Running wannier/emp/pw2wan... done
  Running wannier/emp/wann... done
  Running wannier/bands... done
  Running pdos/projwfc... done

Conversion to Koopmans format
-----------------------------
 Running wannier/kc... done

Construction of the Hamiltonian
===============================
 Running hamiltonian/kc... done

Workflow complete
```

DFT initialisation

Wannierisation (now block-by-block MLWFs to avoid mixing)

(generate a PDOS plot)

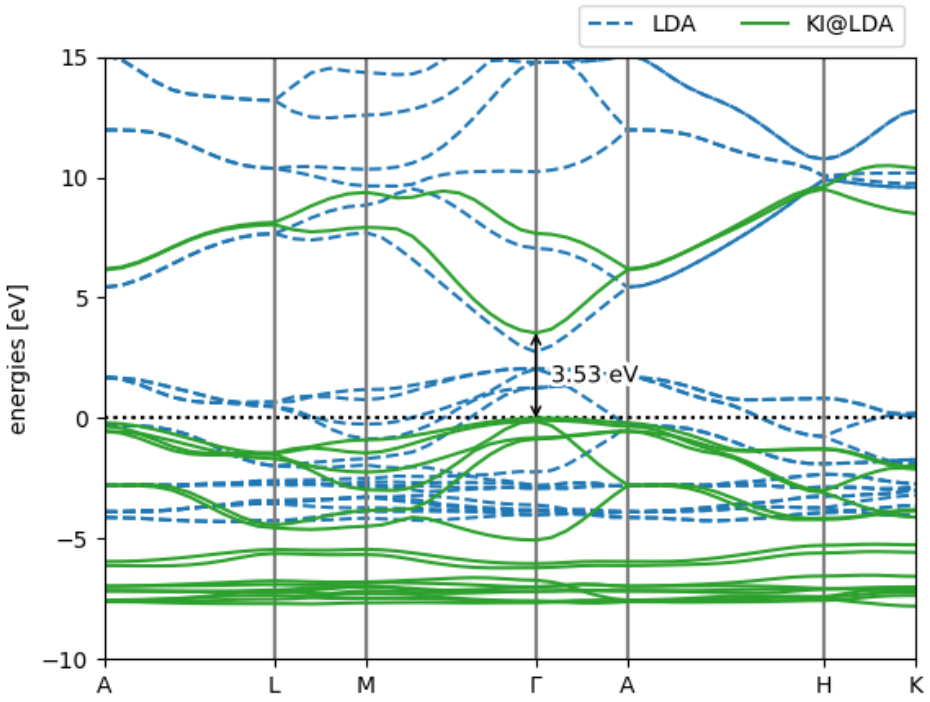
convert to KCW format

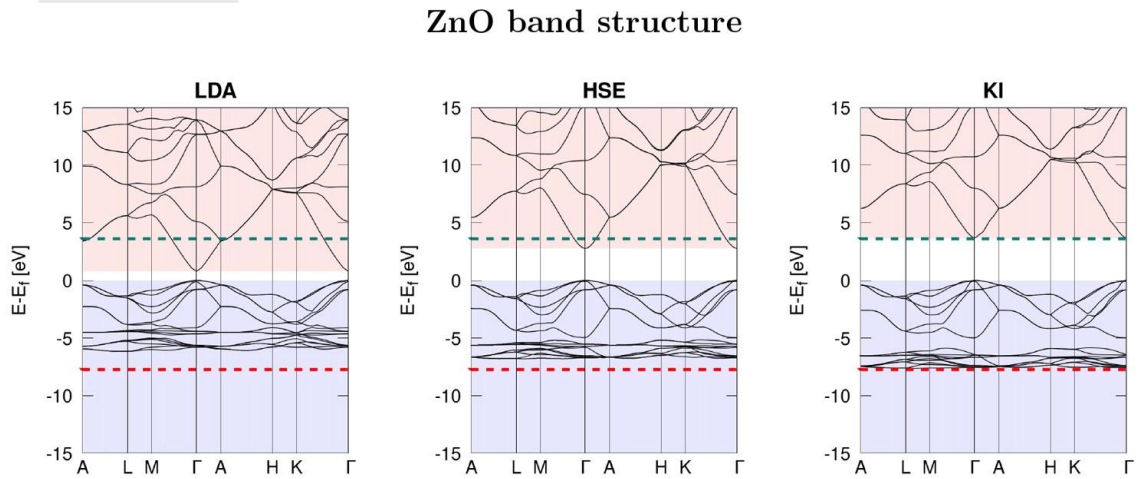
The screening parameters would appear here if we weren't skipping them

Final KI calculation

5. Using koopmans

The workflow will generate a basic band structure plot
We can use the script `plot_bandstructure.py` to generate a nicer figure





	LDA	HSE	GW_0	$scG\tilde{W}$	KI	Exp.
E_{gap}(eV)	0.79	2.79	3.0	3.2	3.62	$3.60^{(*)}$
$\langle\varepsilon_d\rangle$(eV)	-5.1	-6.1	-6.4	-6.7	-6.9	-7.5/-8.0

Colonna et al. 2022 (JCTC)

Want to find out more?

Website (for documentation, papers to read, and more)
koopmans-functionals.org

Google group (for asking questions)
groups.google.com/g/koopmans-users

Github repository (for reporting bugs or contributing new features)
github.com/epfl-theos/koopmans