DFT+*U* hands-on

Iurii Timrov

Theory and Simulation of Materials (THEOS), and National Centre for Computational Design and Discovery of Novel Materials (MARVEL),
École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

Eastern Africa School on Electronic Structure Methods and Applications

Addis Ababa, Ethiopia
2 July 2019

Outline

Exercise 1: Calculation of U_{eff} and DFT+U study of LiCoO2

Exercise 2: Calculation of U_{eff} and DFT+U study of FeO

Exercise 3: Calculation of U_{eff} and DFT+U study of NiO

Outline

Exercise 1: Calculation of U_{eff} and DFT+U study of LiCoO$_2$

Exercise 2: Calculation of U_{eff} and DFT+U study of FeO

Exercise 3: Calculation of U_{eff} and DFT+U study of NiO

DFT+*U*

The DFT+*U* total energy:

$$E_{\text{DFT}+U} = E_{\text{DFT}} + E_U$$

The Hubbard correction energy:

$$E_U = \frac{1}{2} \sum_{I,m,m',\sigma} \underbrace{(U^I - J^I)}_{U^I_{\text{eff}}} \left(\delta_{mm'} - n^{I\sigma}_{mm'}\right) n^{I\sigma}_{m'm}$$

effective Hubbard parameter

The occupation matrix:

$$n^{I\sigma}_{mm'} = \sum_{v,\mathbf{k}} f^{\sigma}_{v\mathbf{k}} \langle \psi^{\sigma}_{v\mathbf{k}} | \varphi^I_{m'} \rangle \langle \varphi^I_m | \psi^{\sigma}_{v\mathbf{k}} \rangle$$

The total occupation of localized states (*d* or *f*) at site *I* :

$$n^I = \sum_{m,\sigma} n^{I\sigma}_{mm}$$

The Kohn-Sham equation:

$$\left[-\frac{1}{2}\nabla^2 + \hat{V}^{\sigma}_{\text{KS}} + \hat{V}^{\sigma}_U \right] \psi^{\sigma}_{v\mathbf{k}}(\mathbf{r}) = \varepsilon^{\sigma}_{v\mathbf{k}} \psi^{\sigma}_{v\mathbf{k}}(\mathbf{r})$$

Exersice 1: LiCoO$_2$

Go to the directory with the input files:

cd   Day2_DFT+U/Exercise1

In this directory you will find:

- README   – File describing how to do the exercise

- LiCoO2.pw.in   – Input file for the DFT ground-state calculation

- LiCoO2.hp.in   – Input file for the DFPT calculation of U_{eff}

- LiCoO2.vcrelax.in   – Input file for the structural optimization

- pseudopotentials   – Directory with the pseudopotentials

- reference   – Directory with the reference results

- tmp   – Directory for temporary files

Exersice 1: LiCoO2

Task 1

Perform a DFT ground-state calculation

Perform a linear-response (DFPT) calculation of the effective Hubbard parameter
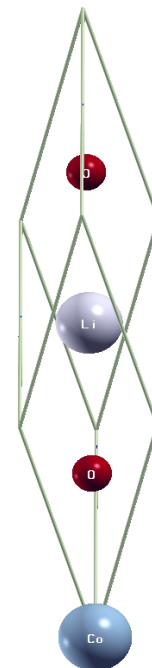
Analyze the input and output files

Input file LiCoO2.pw.in

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='LiCoO2'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
    verbosity='high'
 /
 &system
    ibrav = 5,
    celldm(1) = 9.3705,
    celldm(4) = 0.83874,
    nat = 4,
    ntyp = 3,
    ecutwfc = 30.0
    ecutrho = 240.0
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 1.d-8
 /
 &electrons
    conv_thr =  1.d-12
    mixing_beta = 0.7
 /
ATOMIC_SPECIES
Co  59.0    co_pbesol_v1.2.uspp.F.UPF
O   16.0    O.pbesol-n-kjpaw_psl.0.1.UPF
Li   7.0    li_pbesol_v1.4.uspp.F.UPF
ATOMIC_POSITIONS {crystal}
 Co  0.0000000000    0.0000000000    0.0000000000
 O   0.2604885000    0.2604885000    0.2604885000
 O   0.7395115000    0.7395115000    0.7395115000
 Li  0.5000000000    0.5000000000    0.5000000000
K_POINTS {automatic}
 2 2 2 0 0 0
```

DFT self-consistent (scf) ground-state calculation

Rhombohedral cell and its lattice parameters
(experimental lattice parameters)

Atomic positions in crystal units
(experimental atomic positions)

Input file LiCoO2.pw.in

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='LiCoO2'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
    verbosity='high'
 /
 &system
    ibrav = 5,
    celldm(1) = 9.3705,
    celldm(4) = 0.83874,
    nat = 4,
    ntyp = 3,
    ecutwfc = 30.0
    ecutrho = 240.0
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 1.d-8
 /
 &electrons
    conv_thr =  1.d-12
    mixing_beta = 0.7
 /
ATOMIC_SPECIES
Co  59.0    co_pbesol_v1.2.uspp.F.UPF
O   16.0    O.pbesol-n-kjpaw_psl.0.1.UPF
Li   7.0    li_pbesol_v1.4.uspp.F.UPF
ATOMIC_POSITIONS {crystal}
 Co  0.0000000000    0.0000000000    0.0000000000
 O   0.2604885000    0.2604885000    0.2604885000
 O   0.7395115000    0.7395115000    0.7395115000
 Li  0.5000000000    0.5000000000    0.5000000000
K_POINTS {automatic}
 2 2 2 0 0 0
```

Kinetic-energy cutoffs:

- for wavefunctions (ecutwfc)

- for density and potentials (ecutrho)

Ultrasoft (uspp) and PAW pseudopotentials
PBEsol functional

Selecting pseudopotentials from the SSSP library

Legend:
- pslibrary.1.0.0 US (high acc.)
- pslibrary.1.0.0 PAW (high acc.)
- pslibrary.0.3.1 US (high acc.)
- pslibrary.0.3.1 PAW (high acc.)
- GBRV-1.2 (US)
- GBRV-1.4 (US)
- GBRV-1.5 (US)
- SG15 (NC)
- SG15-1.1 (NC)
- Pseudo Dojo (NC)
- Goedecker
- THEOS
- RE Wentzcovitch

$E_{\Delta(4)}$

E: element
Δ: suggested wave function cutoff in Ry for SSSP efficiency *
(): suggested dual (if default is not used)
*the background color corresponds to the pseudopotential library chosen for SSSP efficiency

SSSP Efficiency (version 1.1)

$\Delta_{eff} = 0.44$ meV

⬇ Cutoffs table   ⬇ Pseudos   Switch to SSSP Precision

Updates v1.1 beta

Input file LiCoO2.pw.in

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='LiCoO2'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
    verbosity='high'
/
&system
    ibrav = 5,
    celldm(1) = 9.3705,
    celldm(4) = 0.83874,
    nat = 4,
    ntyp = 3,
    ecutwfc = 30.0
    ecutrho = 240.0
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 1.d-8
/
&electrons
    conv_thr =  1.d-12
    mixing_beta = 0.7
/
ATOMIC_SPECIES
Co  59.0    co_pbesol_v1.2.uspp.F.UPF
O   16.0    O.pbesol-n-kjpaw_psl.0.1.UPF
Li   7.0    li_pbesol_v1.4.uspp.F.UPF
ATOMIC_POSITIONS {crystal}
 Co  0.0000000000    0.0000000000    0.0000000000
 O   0.2604885000    0.2604885000    0.2604885000
 O   0.7395115000    0.7395115000    0.7395115000
 Li  0.5000000000    0.5000000000    0.5000000000
K_POINTS {automatic}
 2 2 2 0 0 0
```


k points sampling of the Brillouin zone

Input file LiCoO2.pw.in

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='LiCoO2'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
    verbosity='high'
 /
 &system
    ibrav = 5,
    celldm(1) = 9.3705,
    celldm(4) = 0.83874,
    nat = 4,
    ntyp = 3,
    ecutwfc = 30.0
    ecutrho = 240.0
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 1.d-8
 /
 &electrons
    conv_thr =  1.d-12
    mixing_beta = 0.7
 /
ATOMIC_SPECIES
Co   59.0    co_pbesol_v1.2.uspp.F.UPF
O    16.0    O.pbesol-n-kjpaw_psl.0.1.UPF
Li   7.0     li_pbesol_v1.4.uspp.F.UPF
ATOMIC_POSITIONS {crystal}
 Co  0.0000000000    0.0000000000    0.0000000000
 O   0.2604885000    0.2604885000    0.2604885000
 O   0.7395115000    0.7395115000    0.7395115000
 Li  0.5000000000    0.5000000000    0.5000000000
K_POINTS {automatic}
 2 2 2 0 0 0
```

DFT+U specific variables

`lda_plus_u` logical keyword

It means that we want to perform a DFT+U calculation

Note: it is not necessarily LDA+U, it refers also to GGA+U or any other functional +U.

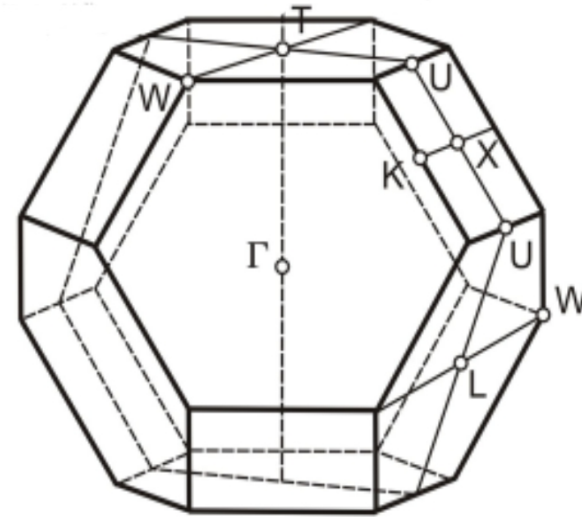This name is for historical reasons.

Input file LiCoO2.pw.in

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='LiCoO2'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
    verbosity='high'
 /
 &system
    ibrav = 5,
    celldm(1) = 9.3705,
    celldm(4) = 0.83874,
    nat = 4,
    ntyp = 3,
    ecutwfc = 30.0
    ecutrho = 240.0
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 1.d-8
 /
 &electrons
    conv_thr =  1.d-12
    mixing_beta = 0.7
 /
ATOMIC_SPECIES
Co  59.0    co_pbesol_v1.2.uspp.F.UPF
O   16.0    O.pbesol-n-kjpaw_psl.0.1.UPF
Li   7.0    li_pbesol_v1.4.uspp.F.UPF
ATOMIC_POSITIONS {crystal}
 Co  0.0000000000    0.0000000000    0.0000000000
 O   0.2604885000    0.2604885000    0.2604885000
 O   0.7395115000    0.7395115000    0.7395115000
 Li  0.5000000000    0.5000000000    0.5000000000
K_POINTS {automatic}
 2 2 2 0 0 0
```

DFT+U specific variables

`lda_plus_u` logical keyword

It means that we want to perform a DFT+U calculation

Note: it is not necessarily LDA+U, it refers also to GGA+U or any other functional +U.

This name is for historical reasons.

`lda_plus_u_kind=0` means a simplified rotationally-invariant formulation of DFT+U according to [1].

[1] *S.L. Dudarev et al., Phys. Rev. B **57**, 1505 (1998)*

Input file LiCoO2.pw.in

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='LiCoO2'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
    verbosity='high'
/
&system
    ibrav = 5,
    celldm(1) = 9.3705,
    celldm(4) = 0.83874,
    nat = 4,
    ntyp = 3,
    ecutwfc = 30.0
    ecutrho = 240.0
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 1.d-8
/
&electrons
    conv_thr =  1.d-12
    mixing_beta = 0.7
/
ATOMIC_SPECIES
Co  59.0    co_pbesol_v1.2.uspp.F.UPF
O   16.0    O.pbesol-n-kjpaw_psl.0.1.UPF
Li  7.0     li_pbesol_v1.4.uspp.F.UPF
ATOMIC_POSITIONS {crystal}
 Co  0.0000000000    0.0000000000    0.0000000000
 O   0.2604885000    0.2604885000    0.2604885000
 O   0.7395115000    0.7395115000    0.7395115000
 Li  0.5000000000    0.5000000000    0.5000000000
K_POINTS {automatic}
 2 2 2 0 0 0
```

DFT+*U* specific variables

`lda_plus_u` logical keyword

It means that we want to perform a DFT+*U* calculation

Note: it is not necessarily LDA+*U*, it refers also to GGA+*U* or any other functional +*U*.

This name is for historical reasons.

`lda_plus_u_kind=0` means a simplified rotationally-invariant formulation of DFT+*U* according to [1].

`U_projection_type` selects the type of localized orbitals which we want to use in projectors.

$$n_{mm'}^{I\sigma} = \sum_{v,\mathbf{k}} f_{v\mathbf{k}}^{\sigma} \langle \psi_{v\mathbf{k}}^{\sigma} | \varphi_{m'}^{I} \rangle \langle \varphi_{m}^{I} | \psi_{v\mathbf{k}}^{\sigma} \rangle$$

[1] *S.L. Dudarev et al., Phys. Rev. B **57**, 1505 (1998)*

Input file LiCoO2.pw.in

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='LiCoO2'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
    verbosity='high'
/
&system
    ibrav = 5,
    celldm(1) = 9.3705,
    celldm(4) = 0.83874,
    nat = 4,
    ntyp = 3,
    ecutwfc = 30.0
    ecutrho = 240.0
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 1.d-8
/
&electrons
    conv_thr =  1.d-12
    mixing_beta = 0.7
/
ATOMIC_SPECIES
Co   59.0    co_pbesol_v1.2.uspp.F.UPF
O    16.0    O.pbesol-n-kjpaw_psl.0.1.UPF
Li    7.0    li_pbesol_v1.4.uspp.F.UPF
ATOMIC_POSITIONS {crystal}
 Co  0.0000000000    0.0000000000    0.0000000000
 O   0.2604885000    0.2604885000    0.2604885000
 O   0.7395115000    0.7395115000    0.7395115000
 Li  0.5000000000    0.5000000000    0.5000000000
K_POINTS {automatic}
 2 2 2 0 0 0
```

DFT+*U* specific variables

`lda_plus_u` logical keyword

It means that we want to perform a DFT+*U* calculation

Note: it is not necessarily LDA+*U*, it refers also to GGA+*U* or any other functional +*U*.

This name is for historical reasons.

`lda_plus_u_kind=0` means a simplified rotationally-invariant formulation of DFT+*U* according to [1].

`U_projection_type` selects the type of localized orbitals which we want to use in projectors.

$$n_{mm'}^{I\sigma} = \sum_{v,\mathbf{k}} f_{v\mathbf{k}}^{\sigma} \langle \psi_{v\mathbf{k}}^{\sigma} | \varphi_{m'}^{I} \rangle \langle \varphi_{m}^{I} | \psi_{v\mathbf{k}}^{\sigma} \rangle$$

Value of the Hubbard *U*eff parameter.

Here we have essentially 0. We put this very small value in order just to let the code know which element we want to consider as a Hubbard atom.

[1] *S.L. Dudarev et al., Phys. Rev. B* **57**, 1505 (1998)

Input file LiCoO2.pw.in

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='LiCoO2'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
    verb
/
&system
    ibrav
    cello
    cello
    nat =
    ntyp
    ecutw
    ecutr
    lda_
    lda_
    U_pro
    Hubba
/
&electro
    conv_
    mixin
/
ATOMIC_SP
Co   59.0
O    16.0    O.pbesol-n-kjpaw_psl.0.1.UPF
Li    7.0    li_pbesol_v1.4.uspp.F.UPF
ATOMIC_POSITIONS {crystal}
 Co  0.0000000000    0.0000000000    0.0000000000
 O   0.2604885000    0.2604885000    0.2604885000
 O   0.7395115000    0.7395115000    0.7395115000
 Li  0.5000000000    0.5000000000    0.5000000000
K_POINTS {automatic}
 2 2 2 0 0 0
```

DFT+U specific variables

`lda_plus_u` logical keyword

It means that we want to perform a DFT+U calculation

... lso to GGA+U

... sons.

... nally-invariant
... cording to [1].

... of localized

$|\psi_{v\mathbf{k}}^{\sigma}\rangle$

Detailed documentation can be found here

quantum-espresso/PW/Doc/INPUT_PW.txt

or here

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

Value of the Hubbard U_{eff} parameter.

Here we have essentially 0. We put this very small value in order just to let the code know which element we want to consider as a Hubbard atom.

[1] *S.L. Dudarev et al., Phys. Rev. B* **57**, *1505 (1998)*

Localized-states manifold

How do we know to what states of a Hubbard atom the U_{eff} correction will be applied?

Currently, this is hard-coded (in Quantum ESPRESSO 6.4.1)

Check two subroutines in **quantum-espresso/Modules** :

set_hubbard_n.f90

principal quantum number "n"

For Co : $n = 3$

set_hubbard_l.f90

orbital quantum number "l"

For Co : $l = 2$

Apply the Hubbard U_{eff} correction to the *3d* electrons of Co

DFPT for U_{eff}

Solve (self-consistently, iteratively) the linear-response Kohn-Sham equations for every $\mathbf{q}$:

$$\left[-\frac{1}{2}[\nabla + i(\mathbf{k}+\mathbf{q})]^2 + \hat{V}_{\text{KS},\mathbf{k}+\mathbf{q}}^{\sigma\circ} + \hat{V}_{U,\mathbf{k}+\mathbf{q}}^{\sigma\circ} - \varepsilon_{v\mathbf{k}}^{\sigma\circ}\right] \Delta_{\mathbf{q}}^s u_{v\mathbf{k}}^{\sigma}(\mathbf{r}) = -\hat{\mathcal{P}}_{\mathbf{k}+\mathbf{q}}^{\sigma}\left[\Delta_{\mathbf{q}}^s V_{\text{Hxc}}^{\sigma} + \sum_m |\phi_{m\mathbf{k}+\mathbf{q}}^s\rangle\langle\phi_{m\mathbf{k}}^s|\right] u_{v\mathbf{k}}^{\sigma\circ}(\mathbf{r})$$

Compute linear-response occupation matrices:

$$\Delta_{\mathbf{q}}^{s'} n_{mm'}^{s\sigma} = \frac{1}{N}\sum_{v,\mathbf{k}}\left[\langle u_{v\mathbf{k}}^{\sigma\circ}|\phi_{m'\mathbf{k}}^s\rangle\langle\phi_{m\mathbf{k}+\mathbf{q}}^s|\Delta_{\mathbf{q}}^{s'}u_{v\mathbf{k}}^{\sigma}\rangle + \langle u_{v\mathbf{k}}^{\sigma\circ}|\phi_{m\mathbf{k}}^s\rangle\langle\phi_{m'\mathbf{k}+\mathbf{q}}^s|\Delta_{\mathbf{q}}^{s'}u_{v\mathbf{k}}^{\sigma}\rangle\right]$$

Sum up over $\mathbf{q}$ and compute the response matrices:

$$\chi_{IJ} = \sum_{m,\sigma}\frac{dn_{mm}^{I\sigma}}{d\alpha_J}, \qquad \chi_{IJ}^0 = \sum_{m,\sigma}\frac{dn_{mm}^{0\,I\sigma}}{d\alpha_J}$$

Compute Hubbard U_{eff} :
$$U_{\text{eff}}^I = \left((\chi^0)^{-1} - \chi^{-1}\right)_{II}$$

Input file LiCoO2.hp.in

```
&inputhp
   prefix = 'LiCoO2',
   outdir = './tmp/',
   nq1 = 2, nq2 = 2, nq3 = 2,
   conv_thr_chi = 1.0d-5,
   iverbosity = 2
/
```

Prefix and **outdir** must be exactly the same as in the input file LiCoO2.pw.in

Size of the **q** point mesh: 2x2x2 in this case

Convergence threshold for the calculation of the response matrices χ_{IJ}

Controls how many details of the calculation is printed in the output file

Detailed documentation can be found here

quantum-espresso/HP/Doc/INPUT_HP.txt

or here

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

Exersice 1: LiCoO2

Perform a DFT ground-state calculation using the **pw.x** code:

mpirun -np 2 $PATH/pw.x < LiCoO2.pw.in > LiCoO2.pw.out

$PATH=/home/max/codes/q-e-qe-6.4.1/bin

Perform a DFPT calculation using the **hp.x** code (**H**ubbard **P**arameters → **HP**):

mpirun -np 2 $PATH/hp.x < LiCoO2.hp.in > LiCoO2.hp.out

$PATH=/home/max/codes/q-e-qe-6.4.1/bin

Analyze the output files LiCoO2.pw.out, LiCoO2.hp.out, and LiCoO2.Hubbard_parameters.dat
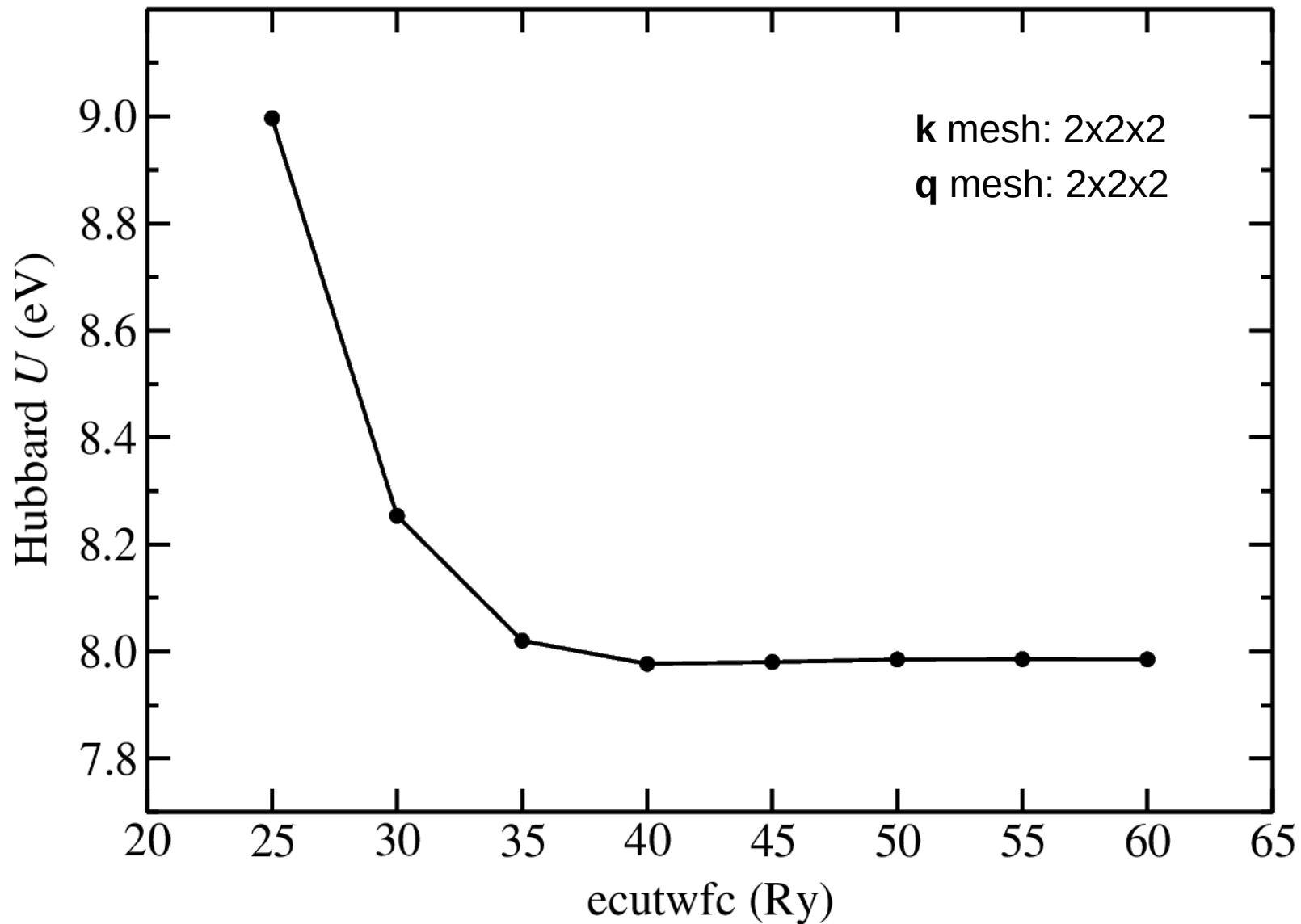
Exersice 1: LiCoO$_2$

Task 2

Perform convergence tests of U_{eff} with respect to the kinetic-energy cutoff.

Hint: try these values of ecutwfc = 25, 30, 35, 40, 45, 50, 55, 60 (Ry).

Keep the ratio ecutrho / ecutwfc = 8 in this exercise.

Use **k** point mesh 2x2x2 and **q** point mesh 2x2x2.

Convergence of U_{eff} with respect to cutoff

k mesh: 2x2x2
q mesh: 2x2x2

ecutrho = 8 * ecutwfc

Convergence of U_{eff} with respect to cutoff
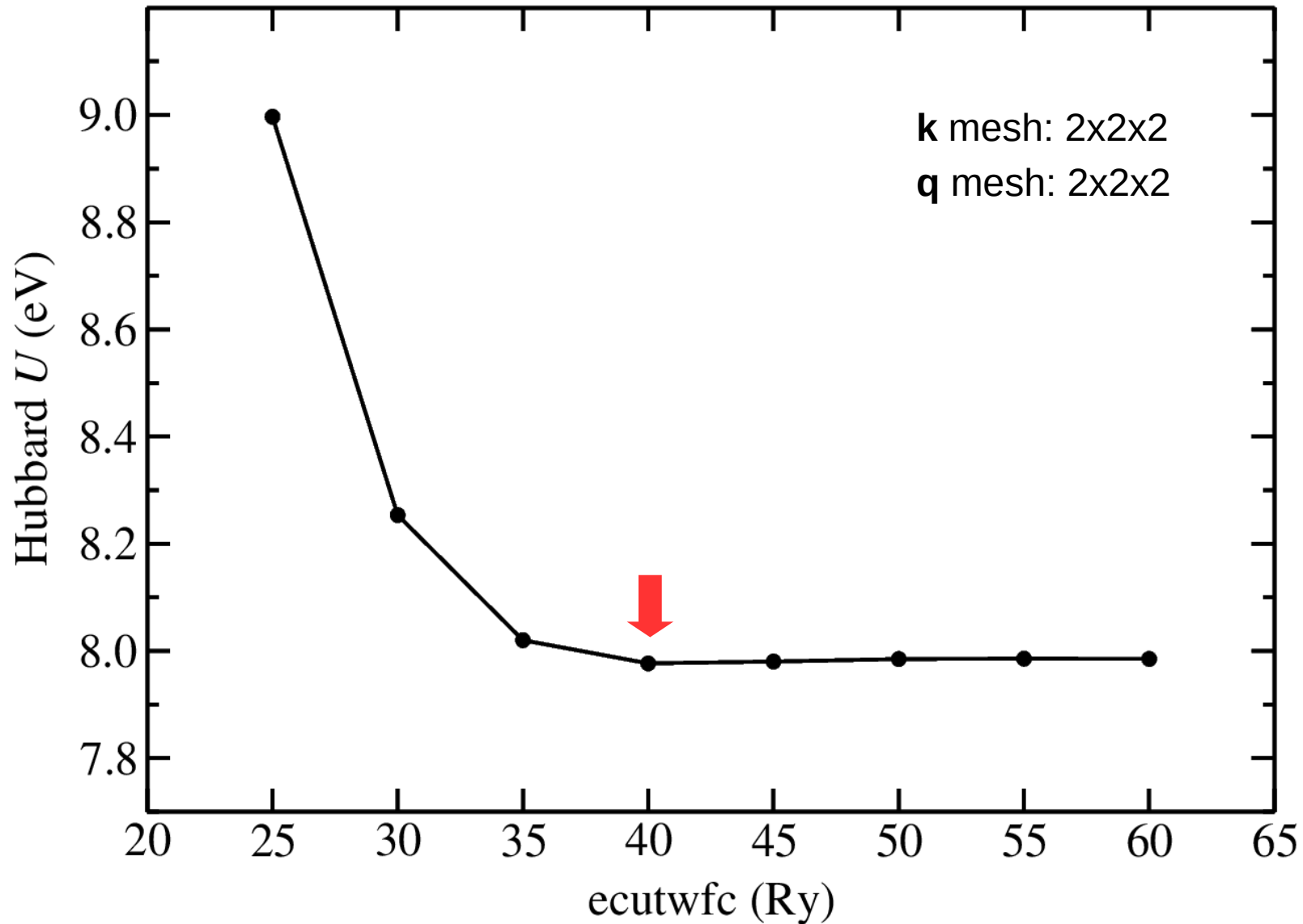
Let us use the cutoff ecutwfc = 40 (Ry) : U_{eff} is converged with the accuracy of ~0.01 (eV).

Convergence of U_{eff} with respect to cutoff

In the SSSP library it is recommended to use such cutoffs:

Li : ecutwfc = 40 (Ry), ecutrho = 320 (Ry)

Co : ecutwfc = 45 (Ry), ecutrho = 360 (Ry)

O : ecutwfc = 50 (Ry), ecutrho = 400 (Ry)

The rule is that when we mix these elements, we need to select the highest cutoff (i.e. ecutwfc = 50 (Ry), ecutrho = 400 (Ry)) and use it for all elements in the compound.

However, from our convergence tests of U_{eff} for Co-3d states in LiCoO2 we found that already at ecutwfc = 40 (Ry) and ecutrho = 320 (Ry) the value of U_{eff} is converged very good (with the accuracy of 0.01 (Ry)).

This is not a universal finding! One has to check carefully the convergence of U_{eff} with respect to the cutoff for every compound.

Exersice 1: LiCoO$_2$

Task 3

Perform convergence tests of U_{eff} with respect to the **k** mesh and **q** mesh.

Hint: use the converged value of the cutoffs obtained in the previous step.

Try these combinations:

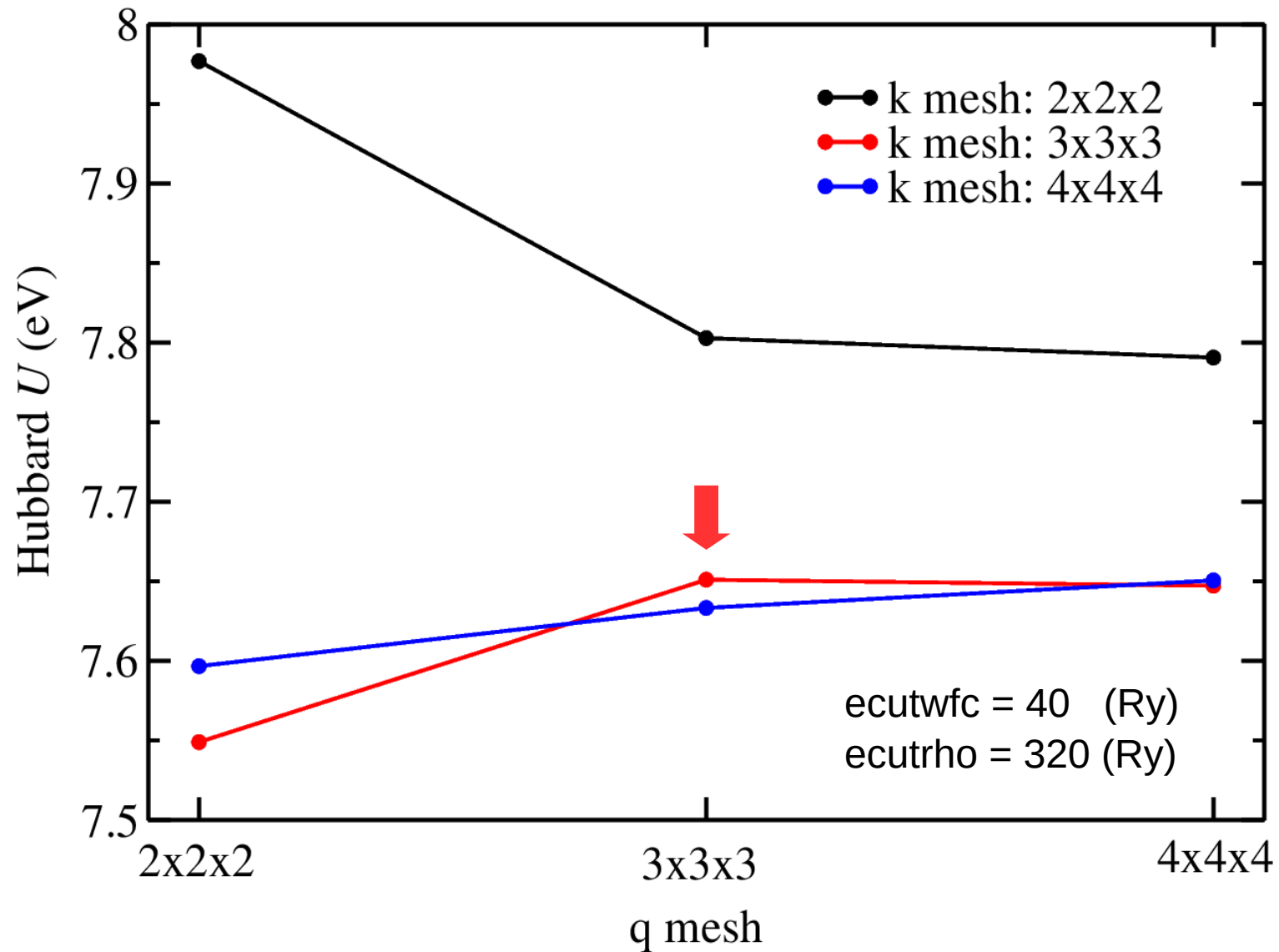| **k** mesh	**q** mesh
2x2x2	2x2x2
2x2x2	3x3x3
2x2x2	4x4x4
3x3x3	2x2x2
3x3x3	3x3x3
3x3x3	4x4x4
4x4x4	2x2x2
4x4x4	3x3x3
4x4x4	4x4x4

Convergence of U_{eff} with respect to k and q

Let us use the **k** mesh 3x3x3 and **q** mesh 3x3x3 : U_{eff} is converged with the accuracy of ~0.01 (eV).

Exersice 1: LiCoO2

Task 4

Perform a self-consistent calculation of U_{eff} (including a structural optimization)

Hint: use the convergence parameters obtained in previous steps.

Self-consistent calculation of Hubbard U_{eff}

So far we have performed a "one-shot" calculation of U_{eff}
using the experimental geometry.

After convergence tests we obtained the value of U_{eff}
for Co-3d which is 7.63 (eV) with the accuracy of 0.01 (eV).

This value was obtained using:

ecutwfc = 40   (Ry)
ecutrho = 320 (Ry)

k mesh: 3x3x3
q mesh: 3x3x3

Self-consistent calculation of Hubbard U_{eff}

So far we have performed a "one-shot" calculation of U_{eff} using the experimental geometry.

After convergence tests we obtained the value of U_{eff} for Co-3d which is 7.63 (eV) with the accuracy of 0.01 (eV).
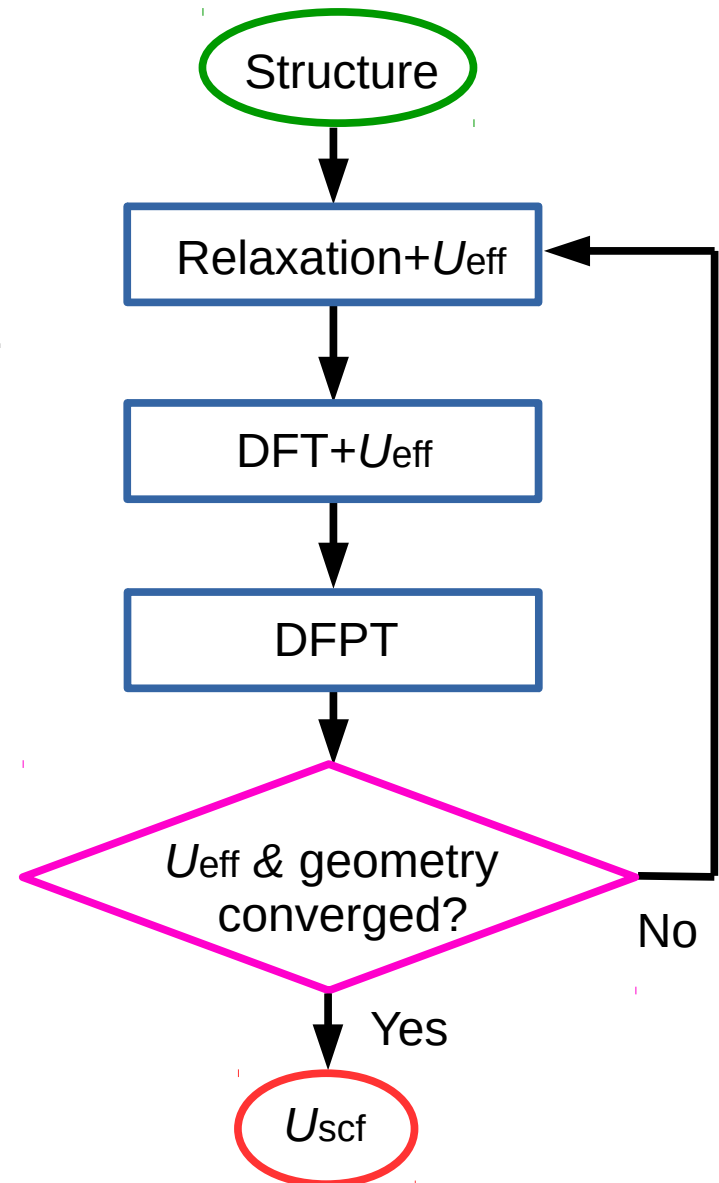
This value was obtained using:
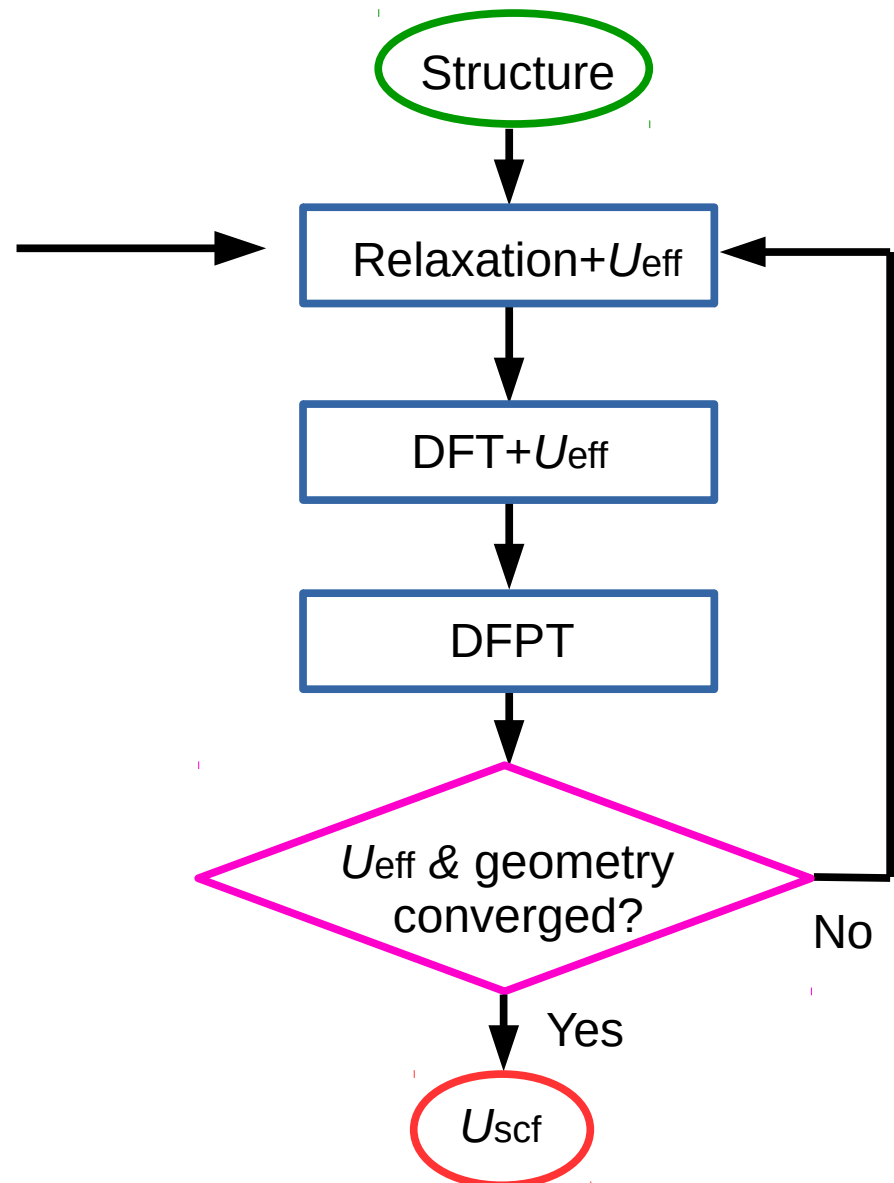
ecutwfc = 40   (Ry)

ecutrho = 320 (Ry)

k mesh: 3x3x3

q mesh: 3x3x3

Now let us compute U_{eff} in a self-consistent way as shown on the schema on the right-hand side.

```
        Structure
            ↓
      Relaxation+U_eff  ←─────┐
            ↓                 │
        DFT+U_eff             │
            ↓                 │
         DFPT                 │
            ↓                 │
    U_eff & geometry   ── No ─┘
       converged?
            ↓ Yes
          U_scf
```

Self-consistent calculation of Hubbard U_{eff}

Our next step is to perform a variable-cell relaxation of the structure using the Hubbard correction with U_{eff} = 7.63 (eV).

Now let us compute U_{eff} in a self-consistent way as shown on the schema on the right-hand side.

Input file LiCoO2.vcrelax.in

```
&control
    calculation='vc-relax'
    restart_mode='from_scratch',
    prefix='LiCoO2'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
    verbosity='high'
    etot_conv_thr = 1.d-5
    forc_conv_thr = 1.d-4
/
&system
    ibrav = 5,
    celldm(1) = 9.3705,
    celldm(4) = 0.83874,
    nat = 4,
    ntyp = 3,
    ecutwfc = 40.0
    ecutrho = 320.0
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 7.63
/
&electrons
    conv_thr =  1.d-12
    mixing_beta = 0.7
/
&ions
    ion_dynamics = 'bfgs'
/
&cell
    cell_dynamics = 'bfgs'
/
ATOMIC_SPECIES
Co  59.0    co_pbesol_v1.2.uspp.F.UPF
O   16.0    O.pbesol-n-kjpaw_psl.0.1.UPF
Li   7.0    li_pbesol_v1.4.uspp.F.UPF
ATOMIC_POSITIONS {crystal}
 Co  0.0000000000    0.0000000000    0.0000000000
 O   0.2604885000    0.2604885000    0.2604885000
 O   0.7395115000    0.7395115000    0.7395115000
 Li  0.5000000000    0.5000000000    0.5000000000
K_POINTS {automatic}
 3 3 3 0 0 0
```

Structural optimization (variable-cell relaxation)

Convergence thresholds on the total energy (a.u.) and on forces (a.u.) for ionic minimization

DFT+U specific parameters

Ionic dynamics section

Cell dynamics section

'**bfgs**' is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton iterative algorithm for solving nonlinear optimization problems.

Structural optimization

Input

```
ibrav = 5,
celldm(1) = 9.3705,
celldm(4) = 0.83874,
```

```
ATOMIC_POSITIONS {crystal}
 Co   0.0000000000   0.0000000000   0.0000000000
 O    0.2604885000   0.2604885000   0.2604885000
 O    0.7395115000   0.7395115000   0.7395115000
 Li   0.5000000000   0.5000000000   0.5000000000
```

Output

```
CELL_PARAMETERS (alat=  9.37050000)
    0.281280268   -0.162397239    0.930675561
    0.000000000    0.324794477    0.930675561
   -0.281280268   -0.162397239    0.930675561
```

```
ATOMIC_POSITIONS (crystal)
Co        0.000000000   -0.000000000   -0.000000000
O         0.259303480    0.259303480    0.259303480
O         0.740696520    0.740696520    0.740696520
Li        0.500000000    0.500000000    0.500000000
```

It is possible to convert the data above for cell parameters to ibrav, celldm(1), and celldm(4) using a simple program **cell2ibrav.x** located in quantum-espresso/PW/tools.

Conversion of structural parameters

Output

```
CELL_PARAMETERS (alat=  9.37050000)
   0.281280268  -0.162397239   0.930675561
   0.000000000   0.324794477   0.930675561
  -0.281280268  -0.162397239   0.930675561
```

By using **cell2ibrav.x** we obtain the following:

```
Enter the lattice vectors (one per line)
0.281280268, -0.162397239, 0.930675561
0.000000000, 0.324794477, 0.930675561
-0.281280268, -0.162397239, 0.930675561
Enter alat (or 'x' to skip if axis are in bohr)
9.3705
ibrav =        5
 celldm(1) =        9.23671081
 celldm(4) =        0.83714565
```

Note: This conversion is optional. You can still set up the input for pw.x using cell parameters as will be shown on next slides.

Structural optimization

Input

```
ibrav = 5,
celldm(1) = 9.3705,
celldm(4) = 0.83874,
```

```
ATOMIC_POSITIONS {crystal}
 Co  0.0000000000    0.0000000000    0.0000000000
 O   0.2604885000    0.2604885000    0.2604885000
 O   0.7395115000    0.7395115000    0.7395115000
 Li  0.5000000000    0.5000000000    0.5000000000
```

Output

```
ibrav = 5,
celldm(1) = 9.2367,
celldm(4) = 0.83715,
```

```
ATOMIC_POSITIONS (crystal)
Co        0.000000000  -0.000000000  -0.000000000
O         0.259303480   0.259303480   0.259303480
O         0.740696520   0.740696520   0.740696520
Li        0.500000000   0.500000000   0.500000000
```

After a structural optimization at the DFT+U level the atomic positions and the lattice parametershave changed. Now we need to recompute U_{eff} using this new geometry.

Two ways of setting up a new input file LiCoO2.pw.in
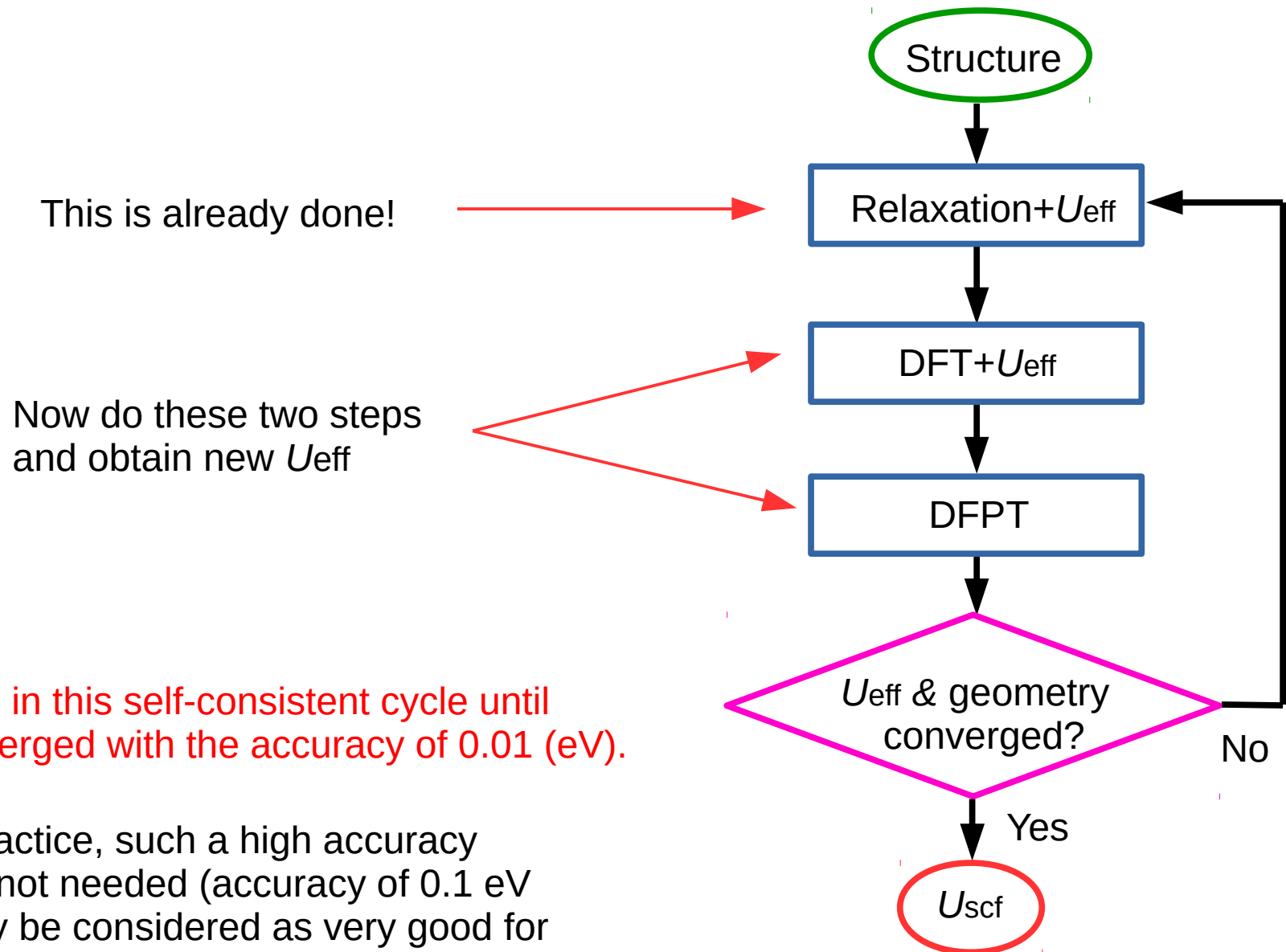
Option 1 :

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='LiCoO2'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
    verbosity='high'
/
&system
    ibrav = 5,
    celldm(1) = 9.2367,          <----- new
    celldm(4) = 0.83715,
    nat = 4,
    ntyp = 3,
    ecutwfc = 40.0
    ecutrho = 320.0
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 7.63
/
&electrons
    conv_thr =  1.d-12
    mixing_beta = 0.7
/
ATOMIC_SPECIES
Co  59.0    co_pbesol_v1.2.uspp.F.UPF
O   16.0    O.pbesol-n-kjpaw_psl.0.1.UPF
Li   7.0    li_pbesol_v1.4.uspp.F.UPF
ATOMIC_POSITIONS {crystal}
 Co      0.000000000    0.000000000    0.000000000
 O       0.259303480    0.259303480    0.259303480
 O       0.740696520    0.740696520    0.740696520
 Li      0.500000000    0.500000000    0.500000000
K_POINTS {automatic}
 3 3 3 0 0 0
```

Option 2 :

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='LiCoO2'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
    verbosity='high'
/
&system
    ibrav = 0,
    celldm(1) = 9.3705,          <----- old
    nat = 4,
    ntyp = 3,
    ecutwfc = 40.0
    ecutrho = 320.0
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 7.63
/
&electrons
    conv_thr =  1.d-12
    mixing_beta = 0.7
/
ATOMIC_SPECIES
Co  59.0    co_pbesol_v1.2.uspp.F.UPF
O   16.0    O.pbesol-n-kjpaw_psl.0.1.UPF
Li   7.0    li_pbesol_v1.4.uspp.F.UPF
CELL_PARAMETERS {alat}
   0.281280268   -0.162397239    0.930675561
   0.000000000    0.324794477    0.930675561
  -0.281280268   -0.162397239    0.930675561
ATOMIC_POSITIONS {crystal}
 Co      0.000000000    0.000000000    0.000000000
 O       0.259303480    0.259303480    0.259303480
 O       0.740696520    0.740696520    0.740696520
 Li      0.500000000    0.500000000    0.500000000
K_POINTS {automatic}
 3 3 3 0 0 0
```

Self-consistent calculation of Hubbard U_{eff}

Structure

This is already done! → Relaxation+U_{eff}

Now do these two steps
and obtain new U_{eff}

DFT+U_{eff}

DFPT

U_{eff} & geometry
converged?

No

Yes

U_{scf}

Keep going in this self-consistent cycle until
U_{eff} is converged with the accuracy of 0.01 (eV).

Note: in practice, such a high accuracy
is typically not needed (accuracy of 0.1 eV
can already be considered as very good for
most studies).

Self-consistent calculation of Hubbard U_{eff}
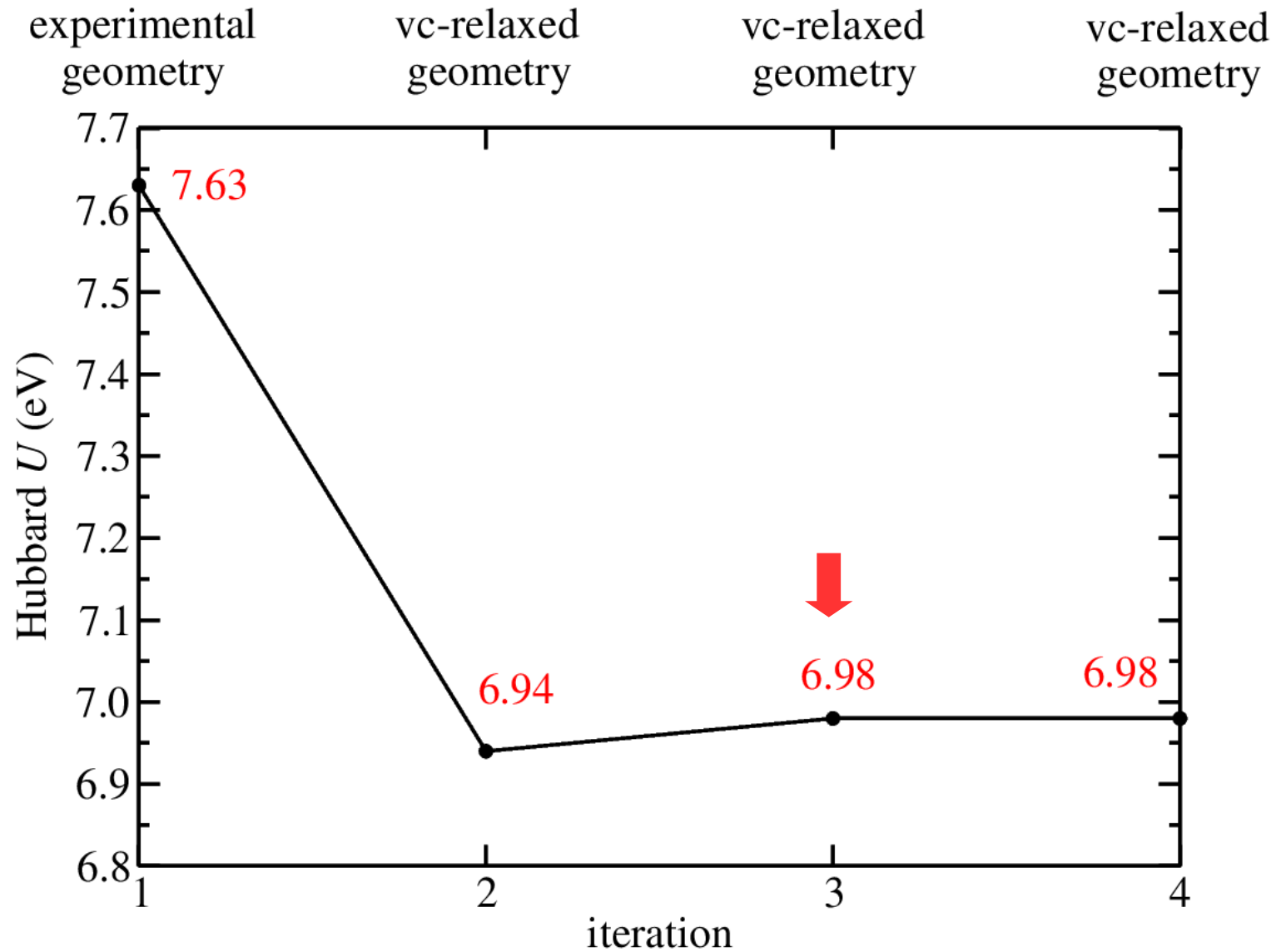
Only 3 iterations is enough to converge U_{eff} with the accuracy of 0.01 (eV) for $LiCoO_2$.

Exersice 1: LiCoO2

Task 5

Perform calculations of the band structure and of the projected density of states (PDOS) using DFT+U with the value of the self-consistent U_{eff} which you obtained in previous steps.

What is the value of a band gap?
How does it compare with the experimental gap: 2.7 ± 0.3 (eV) ?

What is the difference in PDOS obtained at the DFT and DFT+U levels of theory?

In particular, plot Co-3d and O-2p PDOS and check their relative location.

Which conclusions can you make?

Comparison of DFT and DFT+*U* results

DFT
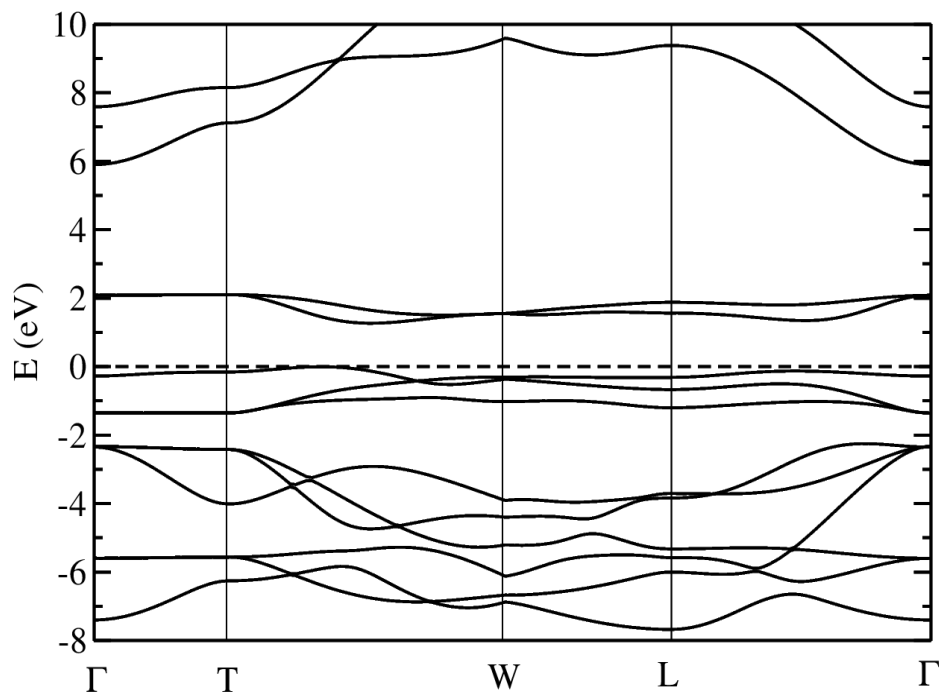
Band gap = 1.14 (eV)

Experimental band gap = 2.7 ± 0.3 (eV)

DFT with the PBEsol functional underestimates the band gap.

Comparison of DFT and DFT+*U* results

DFT
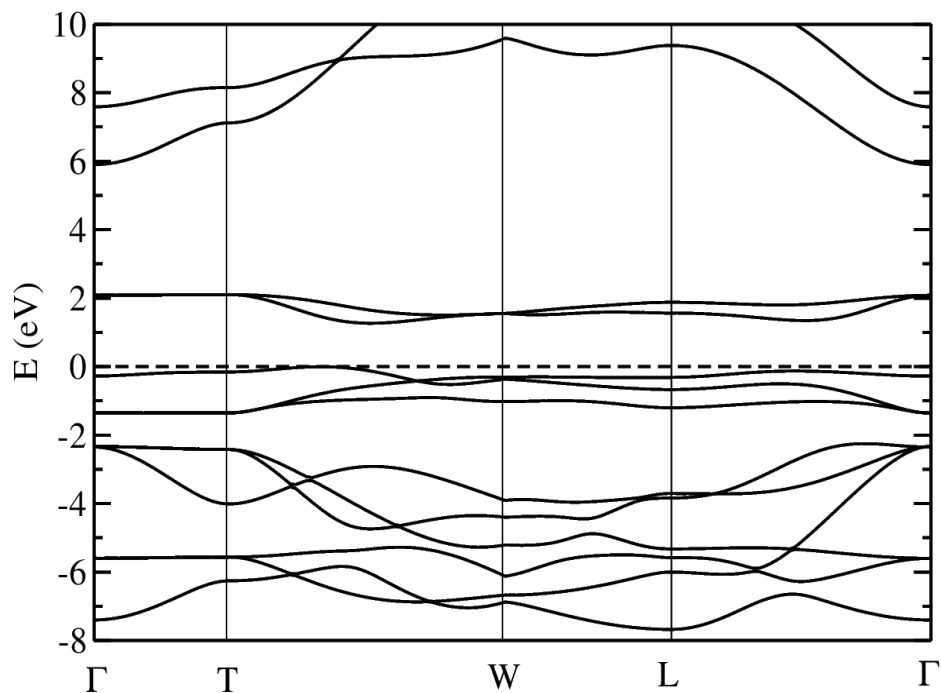
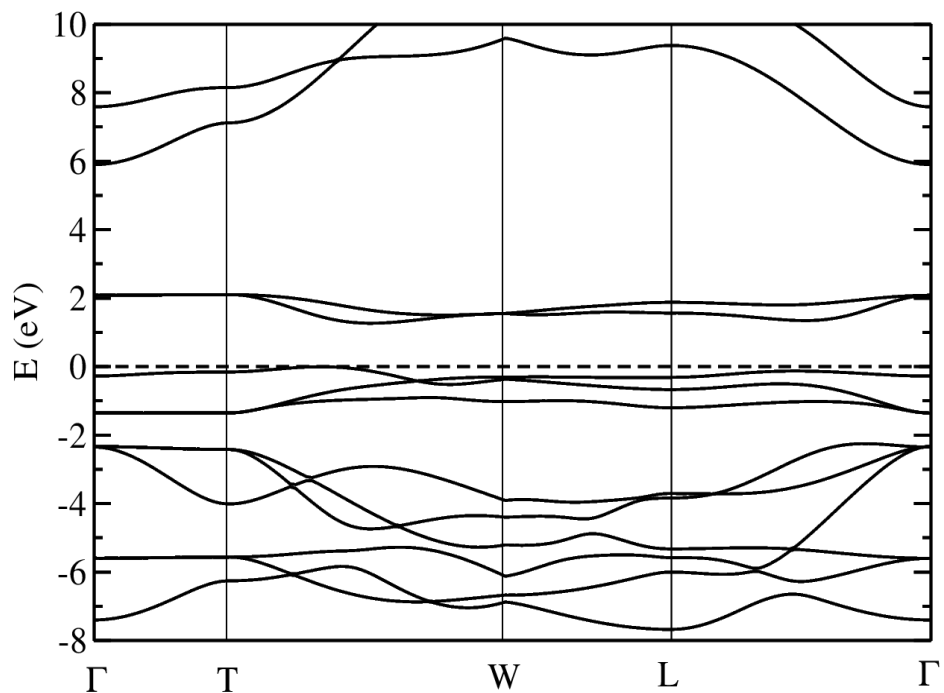## DFT+*U*

Band gap = 1.14 (eV)
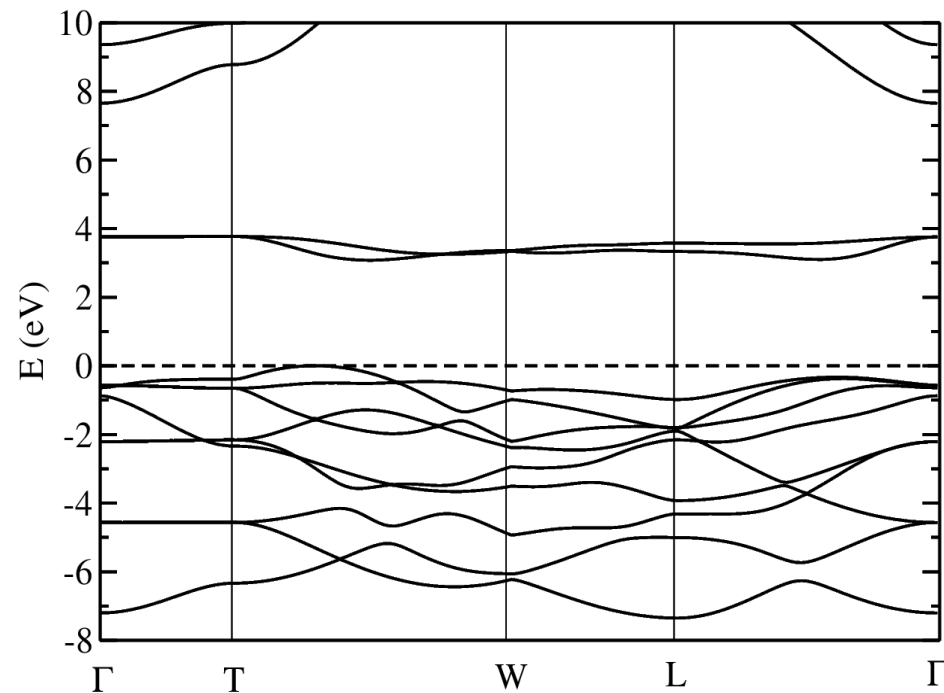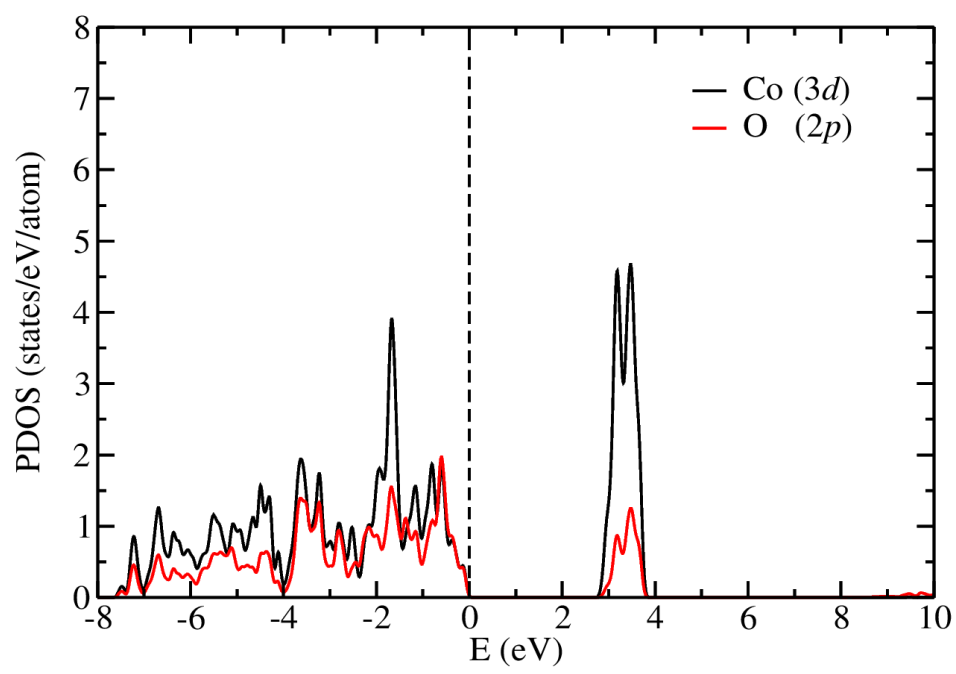
Band gap = 2.94 (eV)

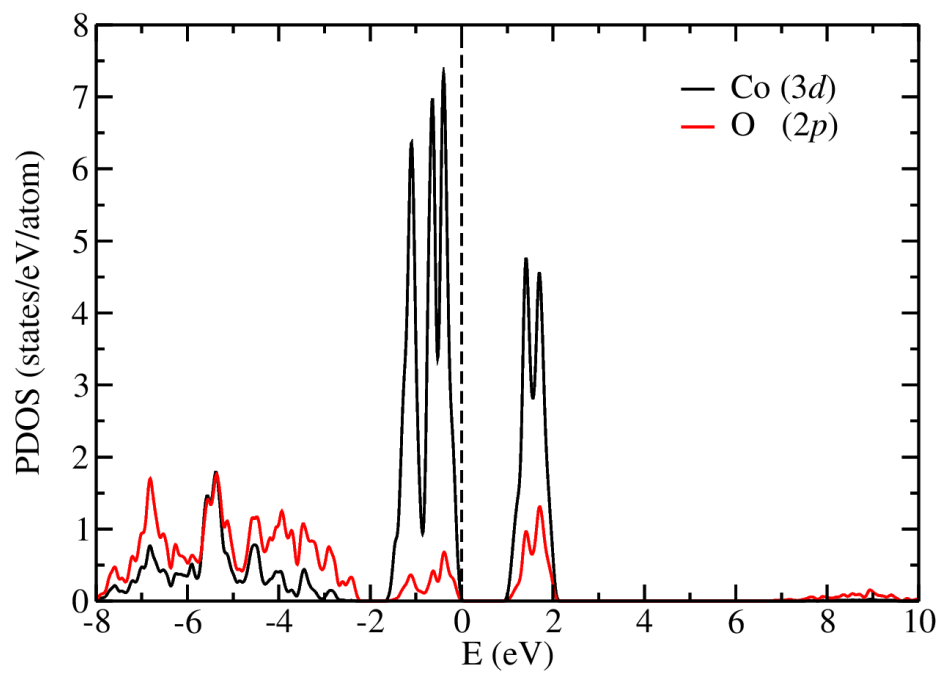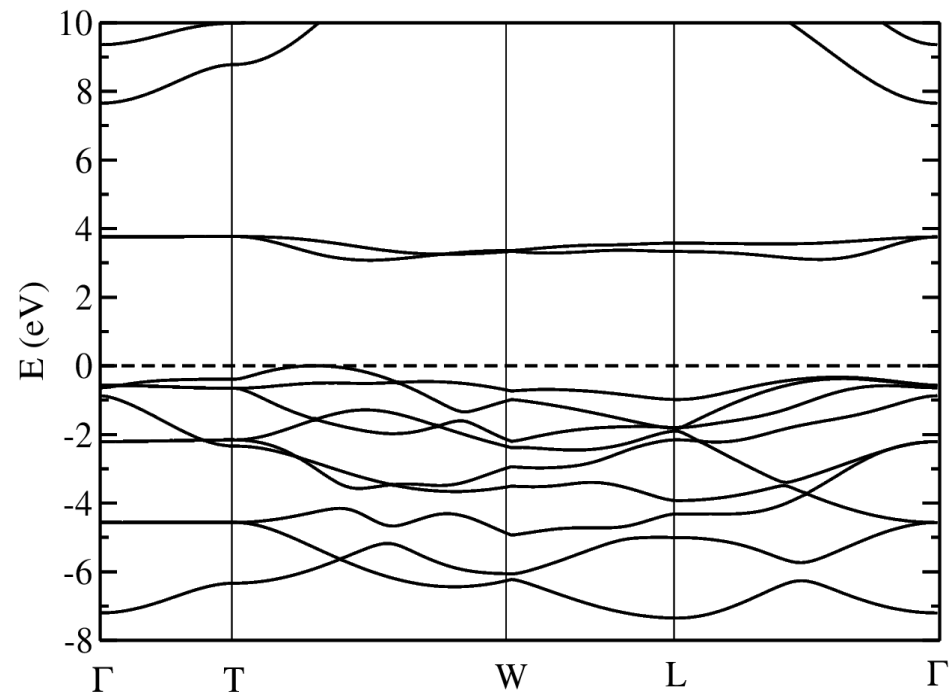Experimental band gap = 2.7 ± 0.3 (eV)

Application of the Hubbard correction to Co-3*d* significantly improves the band gap.

Comparison of DFT and DFT+*U* results

DFT

DFT+*U*

Outline

Exercise 1: Calculation of U_{eff} and DFT+U study of LiCoO$_2$

Exercise 2: Calculation of U_{eff} and DFT+U study of FeO

Exercise 3: Calculation of U_{eff} and DFT+U study of NiO

Exersice 2: FeO

Go to the directory with the input files:

cd   Day2_DFT+U/Exercise2

In this directory you will find:

- README — File describing how to do the exercise
- FeO.pw.in — Input file for the DFT ground-state calculation
- FeO.hp.in — Input file for the DFPT calculation of U_{eff}
- FeO.pw_bands.in — Input file for the band structure calculation
- FeO.pw_nscf.in — Input file for the non-SCF calculation
- FeO.pdos.in — Input file for the PDOS calculation
- FeO.plotband.in — Input file for the band structure plot
- pseudopotentials — Directory with the pseudopotentials
- reference — Directory with the reference results
- tmp — Directory for temporary files

Input file FeO.pw.in

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='FeO'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
/
&system
    ibrav = 0,
    celldm(1) = 8.19,
    nat = 4,
    ntyp = 3,
    ecutwfc = 30.0,
    ecutrho = 240.0,
    occupations = 'smearing',
    smearing = 'mv',
    degauss = 0.02,
    nspin = 2,
    starting_magnetization(1) =  0.5,
    starting_magnetization(2) = -0.5,
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 1.d-8
    Hubbard_U(2) = 1.d-8
/
&electrons
    conv_thr =  1.d-9
    mixing_beta = 0.3
/
ATOMIC_SPECIES
Fe1  55.845  Fe.pbesol-spn-kjpaw_psl.0.2.1.UPF
Fe2  55.845  Fe.pbesol-spn-kjpaw_psl.0.2.1.UPF
O     16.0   O.pbesol-n-kjpaw_psl.0.1.UPF
CELL_PARAMETERS {alat}
 0.50  0.50  1.00
 0.50  1.00  0.50
 1.00  0.50  0.50
ATOMIC_POSITIONS {crystal}
 Fe1 0.00  0.00  0.00
 Fe2 0.50  0.50  0.50
 O   0.25  0.25  0.25
 O   0.75  0.75  0.75
K_POINTS {automatic}
 3 3 3 0 0 0
```
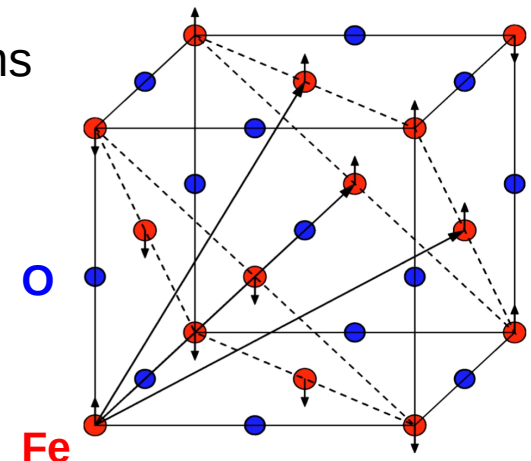
Experimental lattice parameter ← celldm(1) = 8.19,

'MV' smearing for metals (FeO is metallic @PBEsol) ← smearing = 'mv', degauss = 0.02,

Spin-polarized (collinear) calculation (antiferromagnetic ordering) ← starting_magnetization

Initialize Hubbard parameters for two types of Fe ← Hubbard_U(1), Hubbard_U(2)

Two sublattices of Fe atoms ← Fe1 / Fe2

O

Fe

Exersice 2: FeO
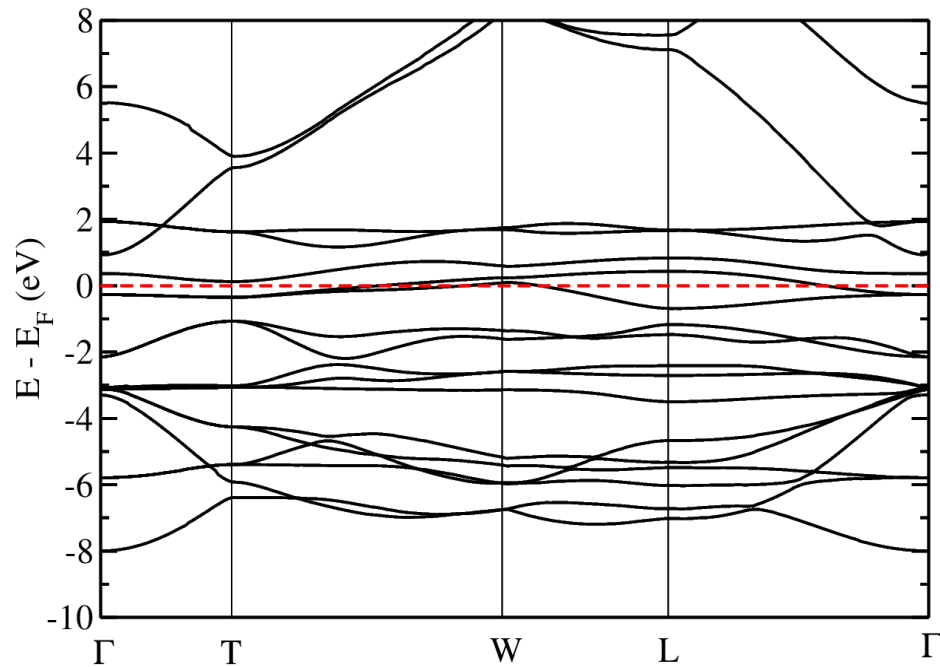
Task 1

Perform a ground-state calculation

Perform a "one-shot" calculation of the effective Hubbard parameter U_{eff}.

Task 2

Perform a calculation of the band structure and PDOS using DFT+U and U_{eff} which you obtained in the previous step.
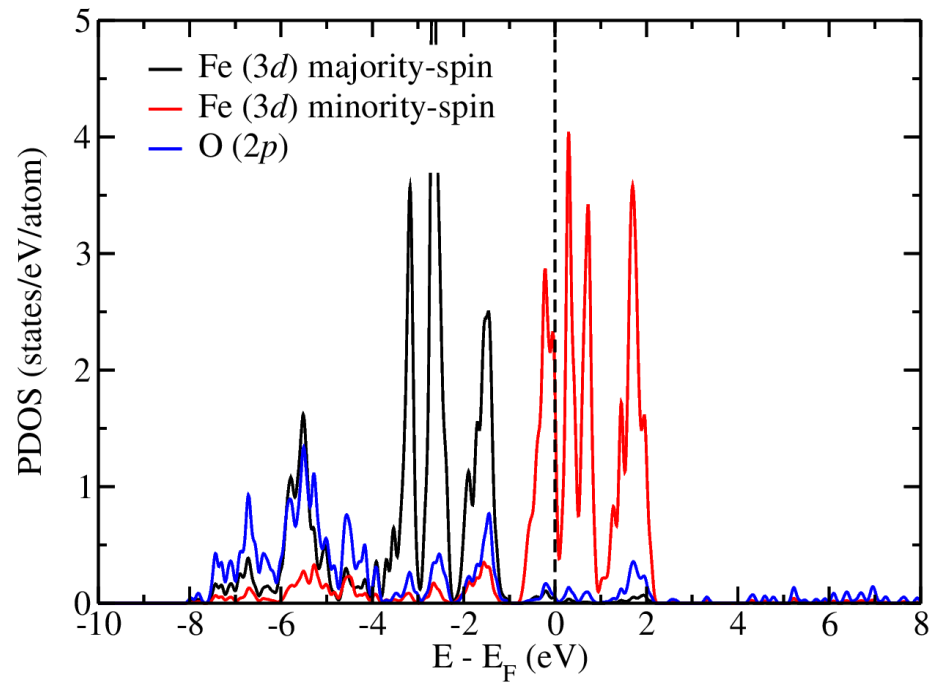
Comparison of DFT and DFT+*U* results

DFT

DFT-PBEsol predicts FeO to be metallic
(but this is wrong)

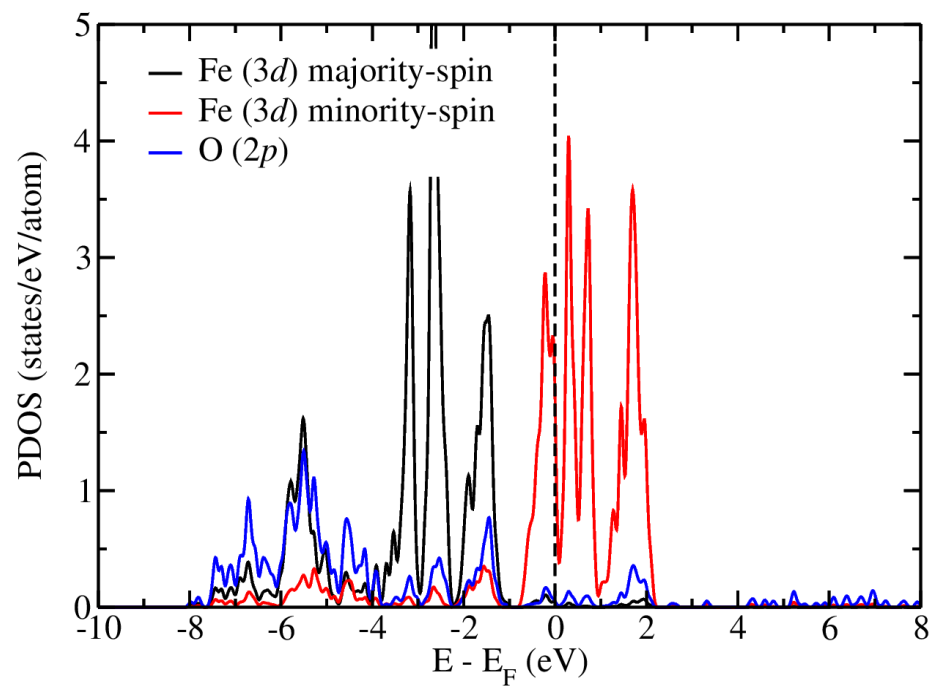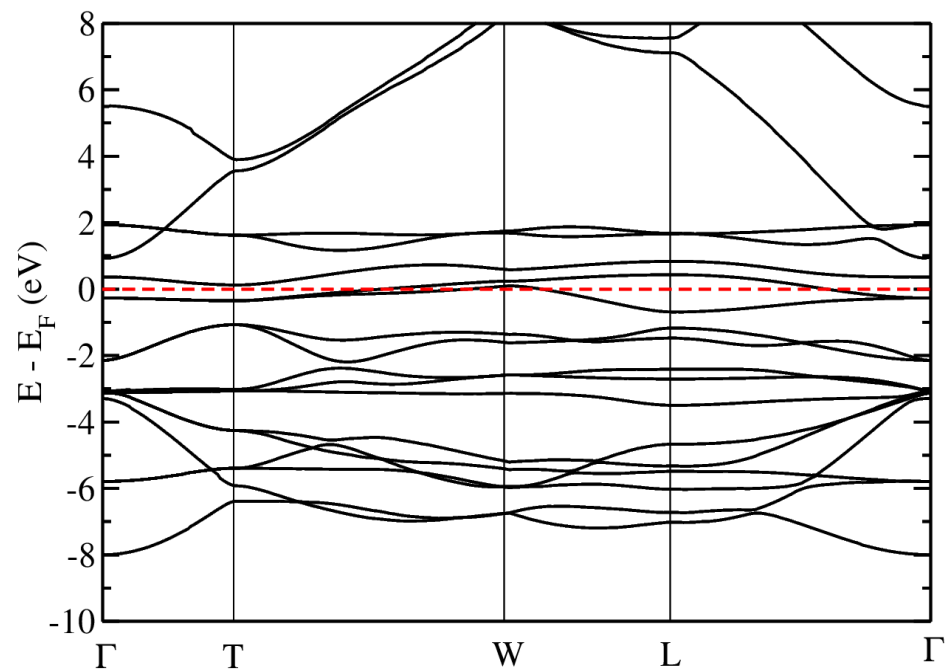Experimentally FeO is know to be insulating

First weak peak in the absorption spectrum
is reported between 0.5 and 2 eV and
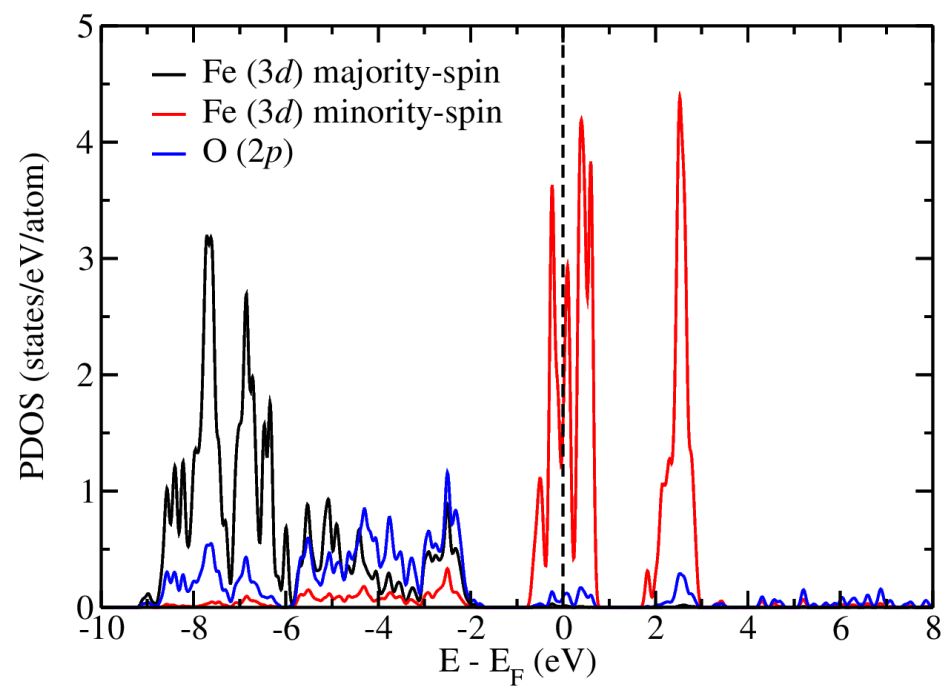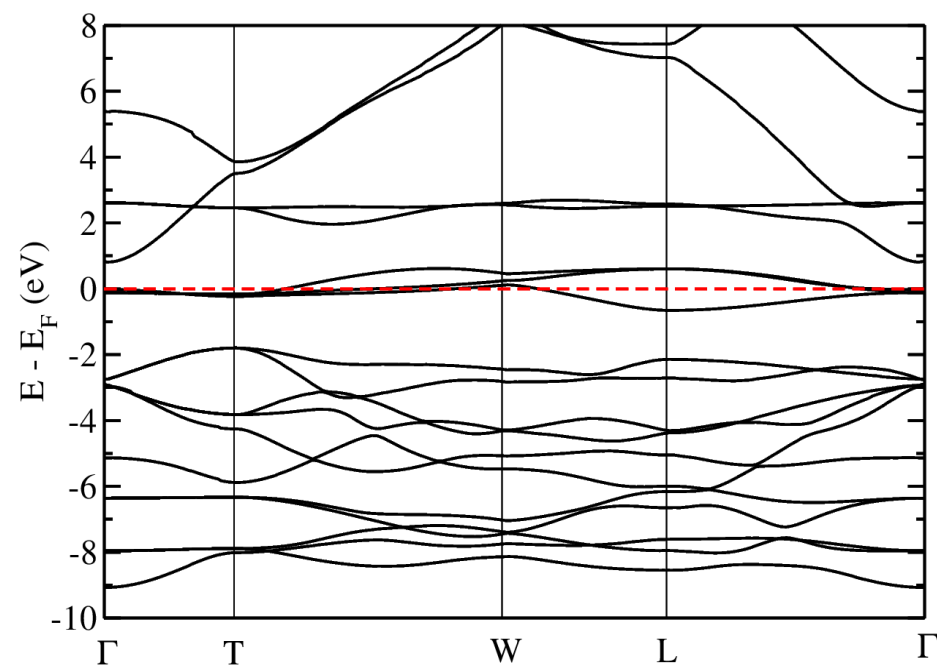a stronger absorption peak occurs at 2.4 eV

M. Cococcioni and S. de Gironcoli, Phys. Rev. B 71, 035105 (2005)

Comparison of DFT and DFT+*U* results

DFT

DFT+*U*

Comparison of DFT and DFT+*U* results

DFT

DFT+*U*

The solution with DFT+*U* is still metallic!

The *d* states of Fe a fully occupied for the majority spin, while for the minority spin they are still partially occupied.

The reason of this failure of DFT+*U* is that it gets stuck in a local minimum (which corresponds to a metallic state) and it needs a "hint" to reach the correct (insulating) ground state.
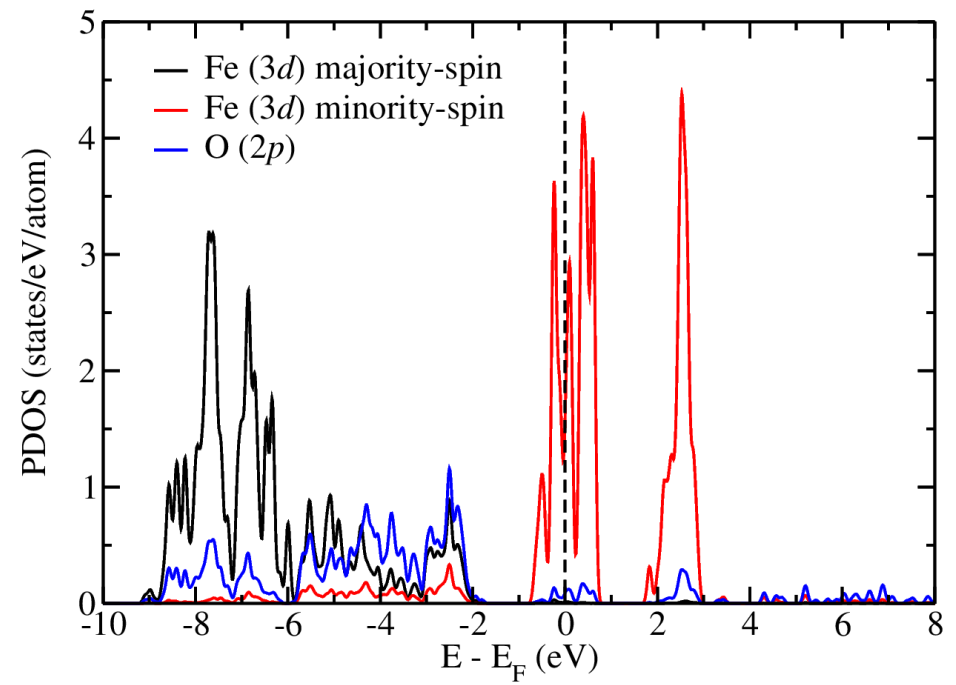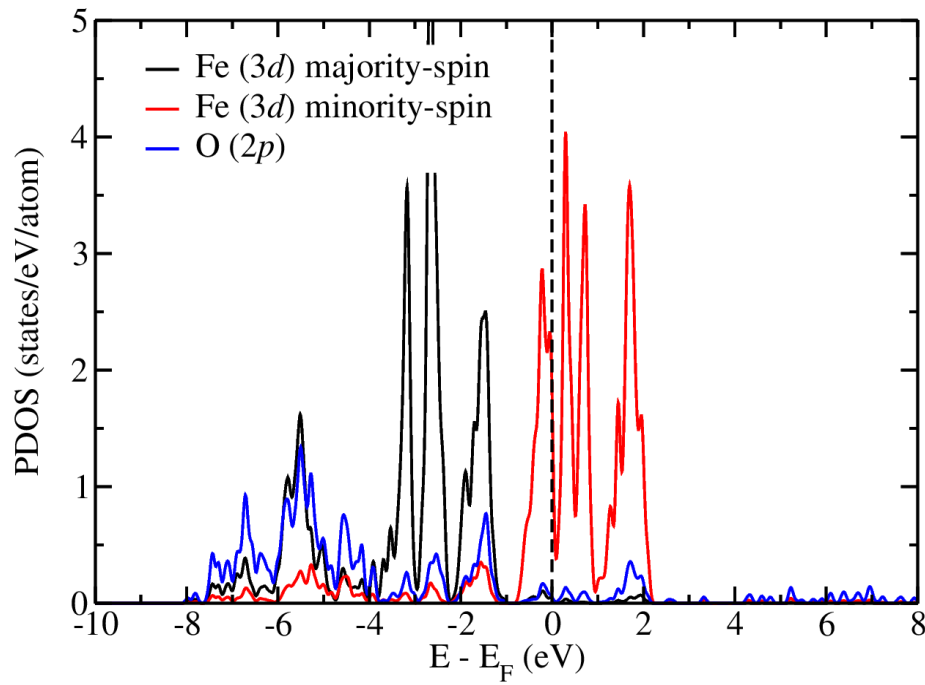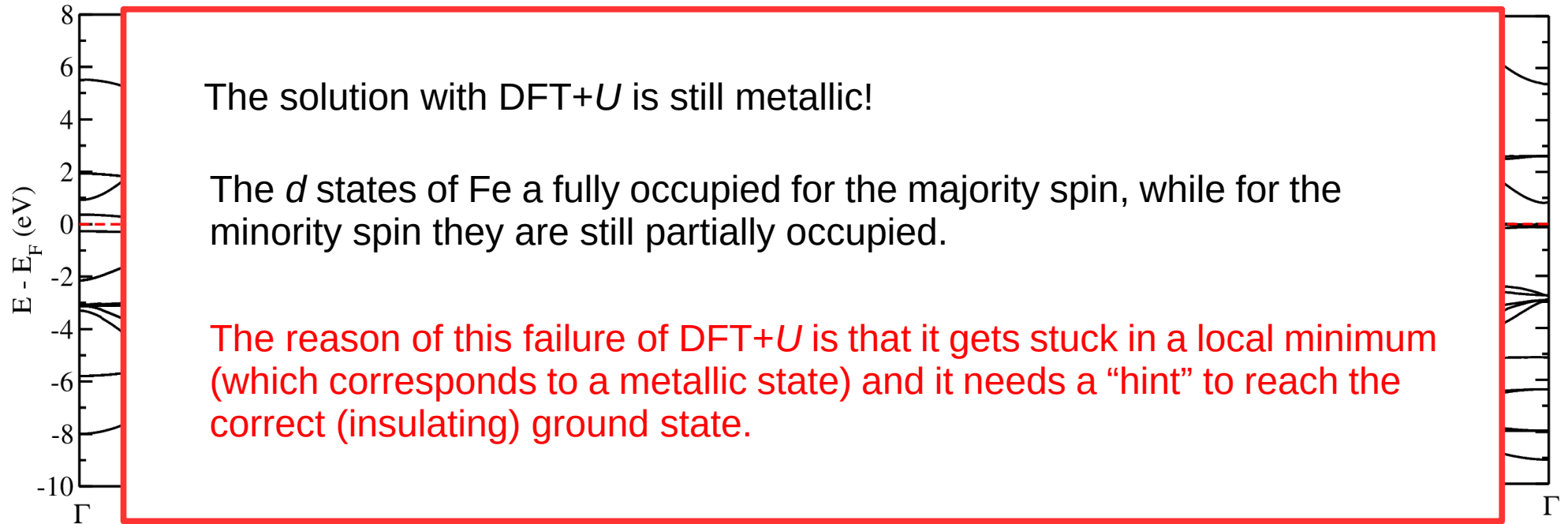
Exersice 2: FeO

Task 3

Perform a calculation of the band structure and PDOS using DFT+U and U_{eff} which you obtained in the previous step, but this time use the input parameter **starting_ns_eigenvalue**.

A "hint" for DFT+*U* where to look for a correct solution

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='FeO'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
 /
 &system
    ibrav = 0,
    celldm(1) = 8.19,
    nat = 4,
    ntyp = 3,
    ecutwfc = 30.0,
    ecutrho = 240.0,
    occupations = 'smearing',
    smearing = 'mv',
    degauss = 0.02,
    nspin = 2,
    starting_magnetization(1) =  0.5,
    starting_magnetization(2) = -0.5
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 5.2
    Hubbard_U(2) = 5.2
    starting_ns_eigenvalue(5,2,1) = 1.0
    starting_ns_eigenvalue(5,1,2) = 1.0
 /
 &electrons
    conv_thr =  1.d-9
    mixing_beta = 0.3
 /
ATOMIC_SPECIES
Fe1  55.845  Fe.pbesol-spn-kjpaw_psl.0.2.1.UPF
Fe2  55.845  Fe.pbesol-spn-kjpaw_psl.0.2.1.UPF
O     16.0   O.pbesol-n-kjpaw_psl.0.1.UPF
CELL_PARAMETERS {alat}
 0.50  0.50  1.00
 0.50  1.00  0.50
 1.00  0.50  0.50
ATOMIC_POSITIONS {crystal}
 Fe1  0.00  0.00  0.00
 Fe2  0.50  0.50  0.50
 O     0.25  0.25  0.25
 O     0.75  0.75  0.75
K_POINTS {automatic}
 3 3 3 0 0 0
```

starting_ns_eigenvalue(m,ispin,ityp)

In the first iteration of a DFT+*U* run it overwrites the **m**-th eigenvalue of the **ns** occupation matrix for the **ispin** spin-component of atomic species of type **ityp**.

Why? This is useful to suggest the desired orbital occupations when the default choice takes another path.

A "hint" for DFT+*U* where to look for a correct solution

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='FeO'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
/
&system
    ibrav = 0,
    celldm(1) = 8.19,
    nat = 4,
    ntyp = 3,
    ecutwfc = 30.0,
    ecutrho = 240.0,
    occupations = 'smearing',
    smearing = 'mv',
    degauss = 0.02,
    nspin = 2,
    starting_magnetization(1) =  0.5,
    starting_magnetization(2) = -0.5
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 5.2
    Hubbard_U(2) = 5.2
    starting_ns_eigenvalue(5,2,1) = 1.0
    starting_ns_eigenvalue(5,1,2) = 1.0
/
&electrons
    conv_thr =  1.d-9
    mixing_beta = 0.3
/
ATOMIC_SPECIES
Fe1  55.845  Fe.pbesol-spn-kjpaw_psl.0.2.1.UPF
Fe2  55.845  Fe.pbesol-spn-kjpaw_psl.0.2.1.UPF
O      16.0    O.pbesol-n-kjpaw_psl.0.1.UPF
CELL_PARAMETERS {alat}
 0.50  0.50  1.00
 0.50  1.00  0.50
 1.00  0.50  0.50
ATOMIC_POSITIONS {crystal}
 Fe1  0.00  0.00  0.00
 Fe2  0.50  0.50  0.50
 O    0.25  0.25  0.25
 O    0.75  0.75  0.75
K_POINTS {automatic}
 3 3 3 0 0 0
```

starting_ns_eigenvalue(m,ispin,ityp)

In the first iteration of a DFT+*U* run it overwrites the **m**-th eigenvalue of the **ns** occupation matrix for the **ispin** spin-component of atomic species of type **ityp**.

Why? This is useful to suggest the desired orbital occupations when the default choice takes another path.

5-th eigenvalue   spin 2 (down)   type 1: Fe1   fully occupied

starting_ns_eigenvalue(5,2,1)= 1.0

starting_ns_eigenvalue(5,1,2)= 1.0

5-th eigenvalue   spin 1 (up)   type 2: Fe2   fully occupied

Understanding the setup of starting_ns_eigenvalue

Check the output file FeO.pw.out from the DFT+U calculation (without starting_ns_eigenvalue):

After the 1-st iteration we have:

Fe1
```
atom     1  Tr[ns(na)] (up, down, total) =    5.00634  1.09448  6.10082
   spin  1
     eigenvalues:
   1.000  1.000  1.002  1.002  1.002

   spin  2
     eigenvalues:
   0.129  0.129  0.270  0.270  0.296
```

Fe2
```
atom     2  Tr[ns(na)] (up, down, total) =    1.09429  5.00635  6.10063
   spin  1
     eigenvalues:
   0.129  0.129  0.270  0.270  0.296

   spin  2
     eigenvalues:
   1.000  1.000  1.002  1.002  1.002
```

Understanding the setup of starting_ns_eigenvalue

Check the output file FeO.pw.out from the DFT+U calculation (without starting_ns_eigenvalue):

After the 1-st iteration we have:

Fe1
```
atom    1  Tr[ns(na)] (up, down, total) =   5.00634  1.09448  6.10082
   spin  1
     eigenvalues:
   1.000  1.000  1.002  1.002  1.002
```
```
      spin  2
        eigenvalues:
      0.129  0.129  0.270  0.270  0.296
```
starting_ns_eigenvalue(5,2,1)= 1.0

Fe2
```
atom    2  Tr[ns(na)] (up, down, total) =   1.09429  5.00635  6.10063
   spin  1
     eigenvalues:
   0.129  0.129  0.270  0.270  0.296
```
starting_ns_eigenvalue(5,1,2)= 1.0
```
      spin  2
        eigenvalues:
      1.000  1.000  1.002  1.002  1.002
```

We force/suggest the occupancy of the 5-th state to be 1.0 instead of 0.296 !

Why 5-th state? Because it is the one which is non-degenerate and if occupied fully could lead to an insulating result.

Understanding the setup of starting_ns_eigenvalue

Check the output file FeO.pw.out from the DFT+U calculation (with starting_ns_eigenvalue):

After the 1-st iteration (i.e. when we forced the 1.0 occupancy of the 5-th state) we have:

```
Modify starting ns matrices according to input values
```

Fe1
```
atom     1   Tr[ns(na)] (up, down, total) =    5.00634  1.79838  6.80472
   spin   1
     eigenvalues:
   1.000  1.000  1.002  1.002  1.002

   spin   2
     eigenvalues:
   0.129  0.129  0.270  0.270  1.000
```

```
atom     2   Tr[ns(na)] (up, down, total) =    1.79825  5.00635  6.80460
   spin   1
     eigenvalues:
   0.129  0.129  0.270  0.270  1.000

   spin   2
     eigenvalues:
   1.000  1.000  1.002  1.002  1.002
```
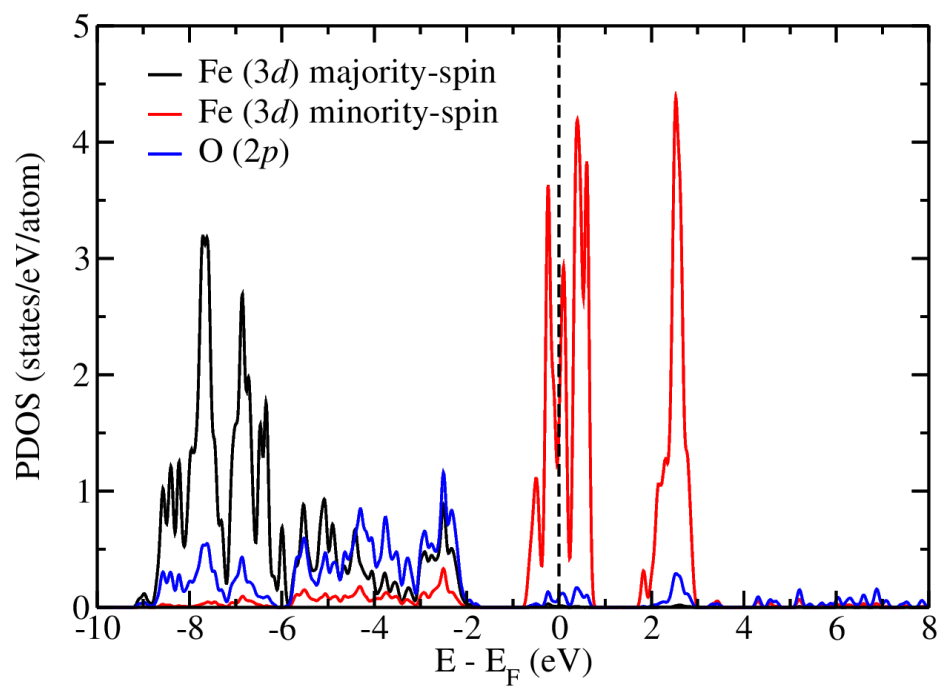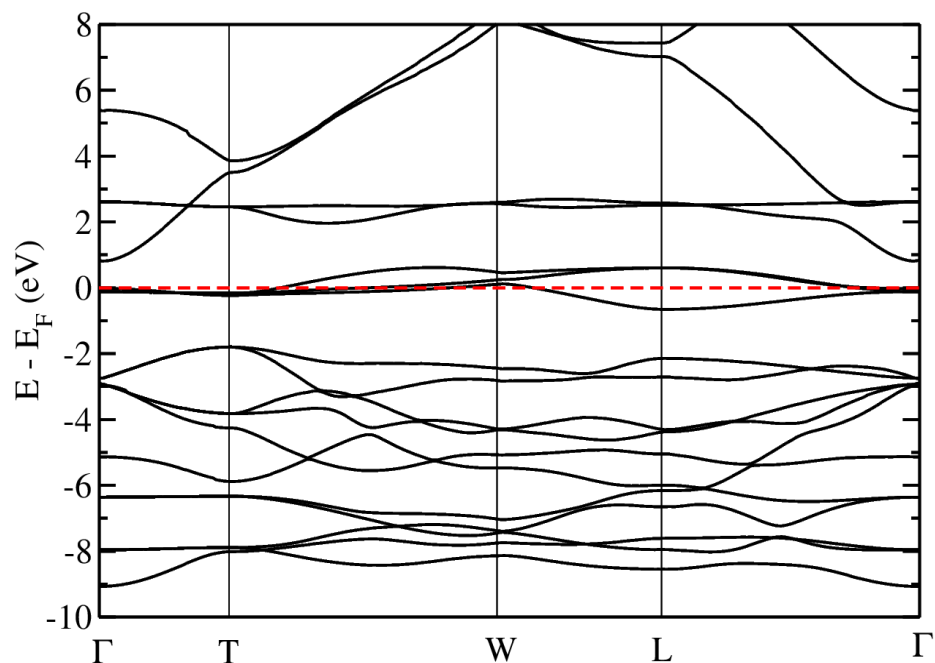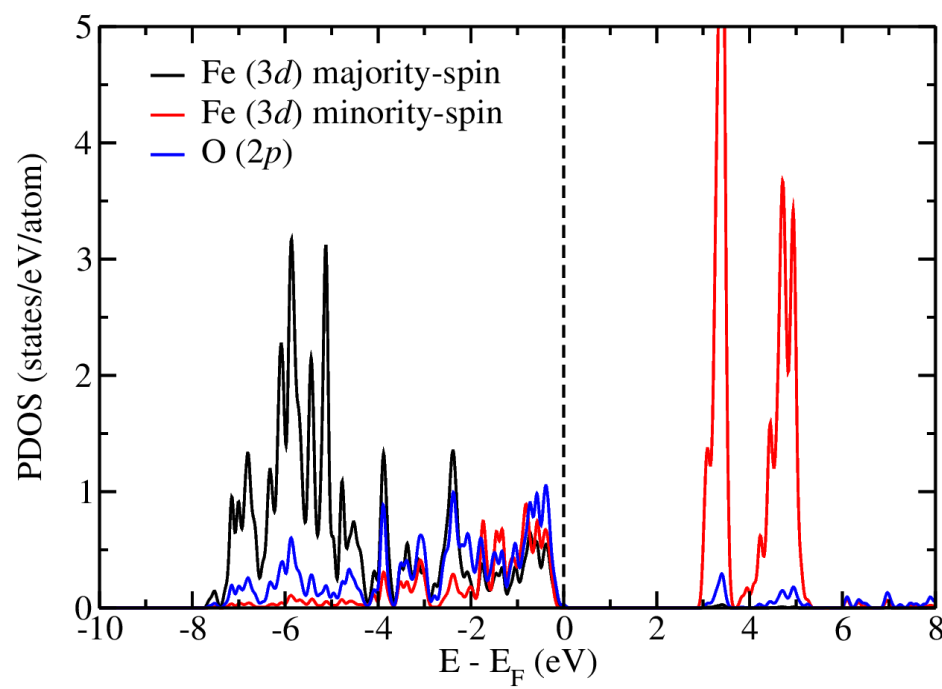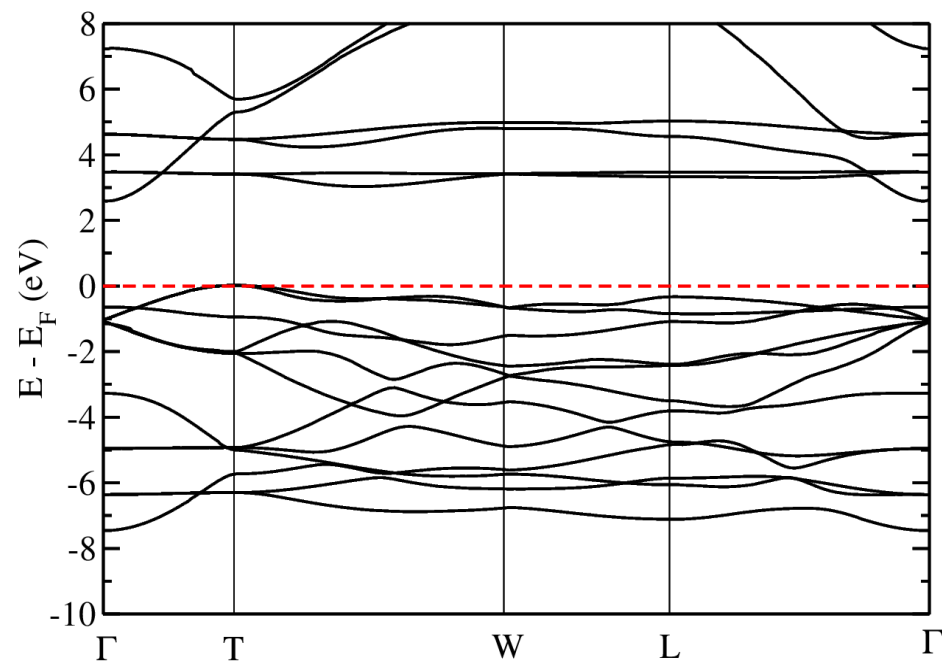
After this the DFT+U calculation converges to an insulating ground state with lower energy than the previous one.

Comparison of DFT+*U* results w/o and w/ starting_ns_eigenvalue

Outline

Exercise 1: Calculation of U_{eff} and DFT+U study of LiCoO$_2$

Exercise 2: Calculation of U_{eff} and DFT+U study of FeO

Exercise 3: Calculation of U_{eff} and DFT+U study of NiO

Exersice 3: NiO

Go to the directory with the input files:

cd   Day2_DFT+U/Exercise3

In this directory you will find:

- README — File describ ing how to do the exercise

- NiO.pw.in — Input file for the DFT ground-state calculation

- NiO.hp.in — Input file for the DFPT calculation of U_{eff}

- pseudopotentials — Directory with the pseudopotentials

- reference — Directory with the reference results

- tmp — Directory for temporary files

Input file NiO.pw1.in

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='NiO'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
 /
 &system
    ibrav = 0,
    celldm(1) = 7.88,
    nat = 4,
    ntyp = 3,
    ecutwfc = 30.0,
    ecutrho = 240.0,
    occupations = 'smearing',
    smearing = 'mv',
    degauss = 0.005,
    nspin = 2,
    starting_magnetization(1) =  0.5,
    starting_magnetization(2) = -0.5
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 1.d-8
    Hubbard_U(2) = 1.d-8
 /
 &electrons
    conv_thr =  1.d-9
    mixing_beta = 0.3
 /
ATOMIC_SPECIES
Ni1  58.693   ni_pbesol_v1.4.uspp.F.UPF
Ni2  58.693   ni_pbesol_v1.4.uspp.F.UPF
O      16.0    O.pbesol-n-kjpaw_psl.0.1.UPF
CELL_PARAMETERS {alat}
 0.50  0.50  1.00
 0.50  1.00  0.50
 1.00  0.50  0.50
ATOMIC_POSITIONS {crystal}
 Ni1  0.00  0.00  0.00
 Ni2  0.50  0.50  0.50
 O     0.25  0.25  0.25
 O     0.75  0.75  0.75
K_POINTS {automatic}
 3 3 3 0 0 0
```
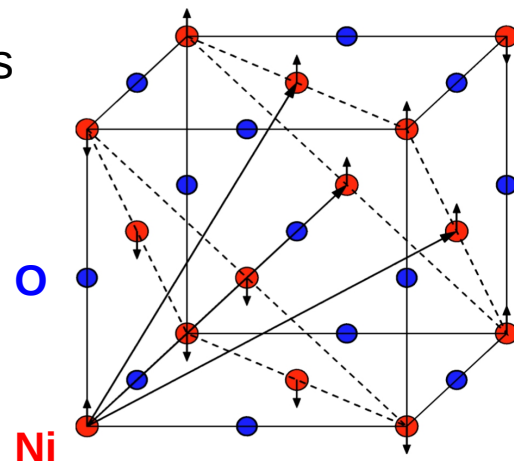
← Experimental lattice parameter

← 'MV' smearing (NiO is insulating @PBEsol)
(needed because we want to use starting_magnetization -
 this is a technical issue)

← Spin-polarized (collinear) calculation
   (antiferromagnetic ordering)

← Initialize Hubbard parameters for two types of Ni

← Two sublattices of Ni atoms

O

Ni

Input file NiO.pw2.in

```
&control
    calculation='scf'
    restart_mode='from_scratch',
    prefix='NiO'
    pseudo_dir = './pseudopotentials/'
    outdir='./tmp/'
/
&system
    ibrav = 0,
    celldm(1) = 7.88,
    nat = 4,
    ntyp = 3,
    ecutwfc = 30.0,
    ecutrho = 240.0,
    occupations = 'fixed',
    nspin = 2,
    tot_magnetization = 0.0
    lda_plus_u = .true.,
    lda_plus_u_kind = 0,
    U_projection_type = 'atomic',
    Hubbard_U(1) = 1.d-8
    Hubbard_U(2) = 1.d-8
    nbnd = 29
/
&electrons
    conv_thr =  1.d-9
    mixing_beta = 0.3
    startingpot = 'file'
    startingwfc = 'file'
/
ATOMIC_SPECIES
Ni1  58.693  ni_pbesol_v1.4.uspp.F.UPF
Ni2  58.693  ni_pbesol_v1.4.uspp.F.UPF
O     16.0   O.pbesol-n-kjpaw_psl.0.1.UPF
CELL_PARAMETERS {alat}
 0.50  0.50  1.00
 0.50  1.00  0.50
 1.00  0.50  0.50
ATOMIC_POSITIONS {crystal}
 Ni1  0.00  0.00  0.00
 Ni2  0.50  0.50  0.50
 O     0.25  0.25  0.25
 O     0.75  0.75  0.75
K_POINTS {automatic}
 3 3 3 0 0 0
```

Now we treat the system as a real insulator (i.e. no "fake" smearing).

Now we use the value of the total magnetization, which we need to extract from the output of the previous calculation. In this case it is zero, because it is antiferromagnetic. But for other systems we may not know what is the value of the total magnetization, so the first step was needed.

Keep the number of bands (states) as in the previous calculation.

Read the density/potential and wavefunctions from the previous converged calculation. This would be a very good start for this iterative calculation.

DFPT for magnetic insulators

The two-step procedure in the ground-state calculation for magnetic insulators is needed to avoid singularity problems in DFPT.

More precisely: in DFPT for metals there is a devision by the density of states at the Fermi level, and the latter quantity is extremely small if we put a fake smearing in an insulator – so this fake smearing must be eliminated once the magnetization of the system is determined.

Task 1

Perform a "one-shot" calculation of U_{eff} by following this procedure.

Task 2

Perform convergence tests for U_{eff} and compute it self-consistenly.