

# DFT in practice: from simple to advanced functionals

Nicola Colonna<sup>1</sup> & Iurii Timrov<sup>2</sup>

<sup>1</sup>Paul Scherrer Institute and NCCR-Marvel

<sup>2</sup>EPFL and NCCR-Marvel



MARVEL



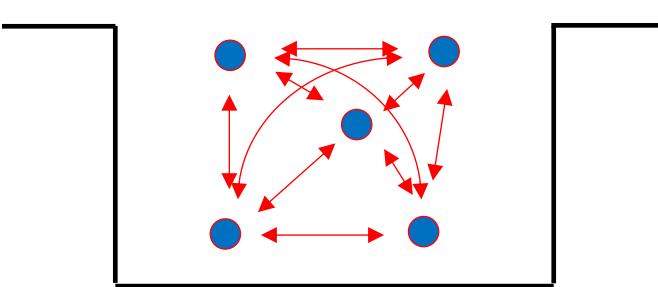
NATIONAL CENTRE OF COMPETENCE IN RESEARCH



# Outline

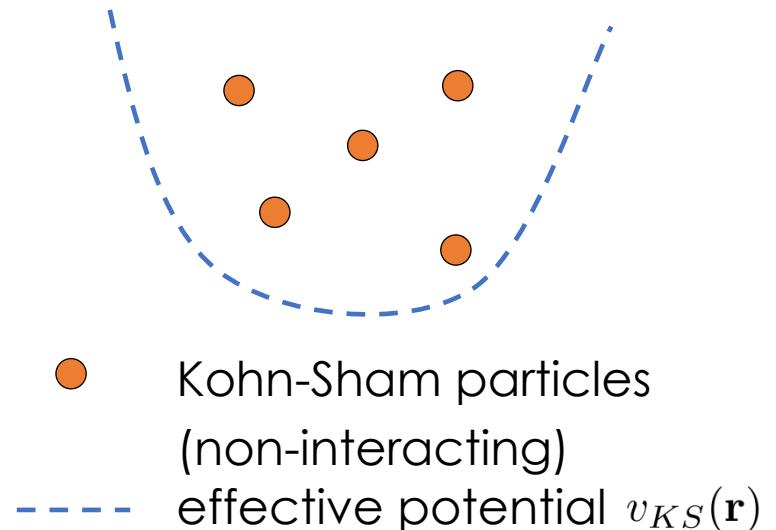
- **Brief recap on KS-DFT.** KS equations, common approximations to the xc energy functional, KS in practice
- **Exercise 0.** Standard DFT: PBE band structure of FCC Silicon
- **Exercise 1-2.** Hybrid DFT: electronic structure of FCC Silicon
- **Exercise 3-4.** Meta-GGA: electronic structure of FCC Silicon and BCC Iron using SCAN

# Kohn-Sham DFT



KS MAPPING

- electrons
- ↔ interaction
- external potential  $v_{ext}(\mathbf{r})$



- Kohn-Sham particles  
(non-interacting)
- - - effective potential  $v_{KS}(\mathbf{r})$

$$E[n] = T_s[n] + E_H[n] + E_{xc}[n] + \int d\mathbf{r} V_{ext}(\mathbf{r})n(\mathbf{r})$$

$$\left[ -\frac{\hbar^2}{2m_e} \nabla^2 + v_{KS}[n](\mathbf{r}) \right] \phi_i(\mathbf{r}) = \epsilon_i \phi_i(\mathbf{r})$$

$$n(\mathbf{r}) = \sum_{i=1}^N |\phi_i(\mathbf{r})|^2$$

$$v_{KS}[n](\mathbf{r}) = V_{ext}(\mathbf{r}) + e^2 \int d\mathbf{r}' \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + \frac{\delta E_{xc}[n]}{\delta n(\mathbf{r})}$$

# The exchange and correlation energy functional

- **LDA:** The xc energy density of the inhomogeneous system in  $\mathbf{r}$  is locally approximated with that of a homogeneous electron gas with density  $n(\mathbf{r})$

$$E_{\text{xc}}^{\text{LDA}} = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\text{xc}}^{\text{HEG}}(n(\mathbf{r}))$$

# The exchange and correlation energy functional

- **LDA:** The xc energy density of the inhomogeneous system in  $\mathbf{r}$  is locally approximated with that of a homogeneous electron gas with density  $n(\mathbf{r})$

$$E_{\text{xc}}^{\text{LDA}} = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\text{xc}}^{\text{HEG}}(n(\mathbf{r}))$$

- **GGA:** include the dependence on the gradient of the density

$$E_{\text{xc}}^{\text{GGA}} = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\text{xc}}(n(\mathbf{r}), \nabla n(\mathbf{r})) = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\text{xc}}^{\text{HEG}}(n(\mathbf{r})) F_{\text{xc}}(n(\mathbf{r}), \nabla n(\mathbf{r}))$$

# The exchange and correlation energy functional

- **LDA:** The xc energy density of the inhomogeneous system in  $\mathbf{r}$  is locally approximated with that of a homogeneous electron gas with density  $n(\mathbf{r})$

$$E_{\text{xc}}^{\text{LDA}} = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\text{xc}}^{\text{HEG}}(n(\mathbf{r}))$$

- **GGA:** include the dependence on the gradient of the density

$$E_{\text{xc}}^{\text{GGA}} = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\text{xc}}(n(\mathbf{r}), \nabla n(\mathbf{r})) = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\text{xc}}^{\text{HEG}}(n(\mathbf{r})) F_{\text{xc}}(n(\mathbf{r}), \nabla n(\mathbf{r}))$$

- **Meta-GGA:** include the dependency on the laplacian of the density and kinetic energy density

$$E_{\text{xc}}^{\text{mGGA}} = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\text{xc}}(n(\mathbf{r}), \nabla n(\mathbf{r}), \nabla^2 n(\mathbf{r}), \tau) \quad \tau = \frac{1}{2} \sum_i |\nabla \psi_i(\mathbf{r})|^2$$

# The exchange and correlation energy functional

- **LDA:** The xc energy density of the inhomogeneous system in  $\mathbf{r}$  is locally approximated with that of a homogeneous electron gas with density  $n(\mathbf{r})$

$$E_{\text{xc}}^{\text{LDA}} = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\text{xc}}^{\text{HEG}}(n(\mathbf{r}))$$

- **GGA:** include the dependence on the gradient of the density

$$E_{\text{xc}}^{\text{GGA}} = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\text{xc}}(n(\mathbf{r}), \nabla n(\mathbf{r})) = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\text{xc}}^{\text{HEG}}(n(\mathbf{r})) F_{\text{xc}}(n(\mathbf{r}), \nabla n(\mathbf{r}))$$

- **Meta-GGA:** include the dependency on the laplacian of the density and kinetic energy density

$$E_{\text{xc}}^{\text{mGGA}} = \int d\mathbf{r} n(\mathbf{r}) \varepsilon_{\text{xc}}(n(\mathbf{r}), \nabla n(\mathbf{r}), \nabla^2 n(\mathbf{r}), \tau) \quad \tau = \frac{1}{2} \sum_i |\nabla \psi_i(\mathbf{r})|^2$$

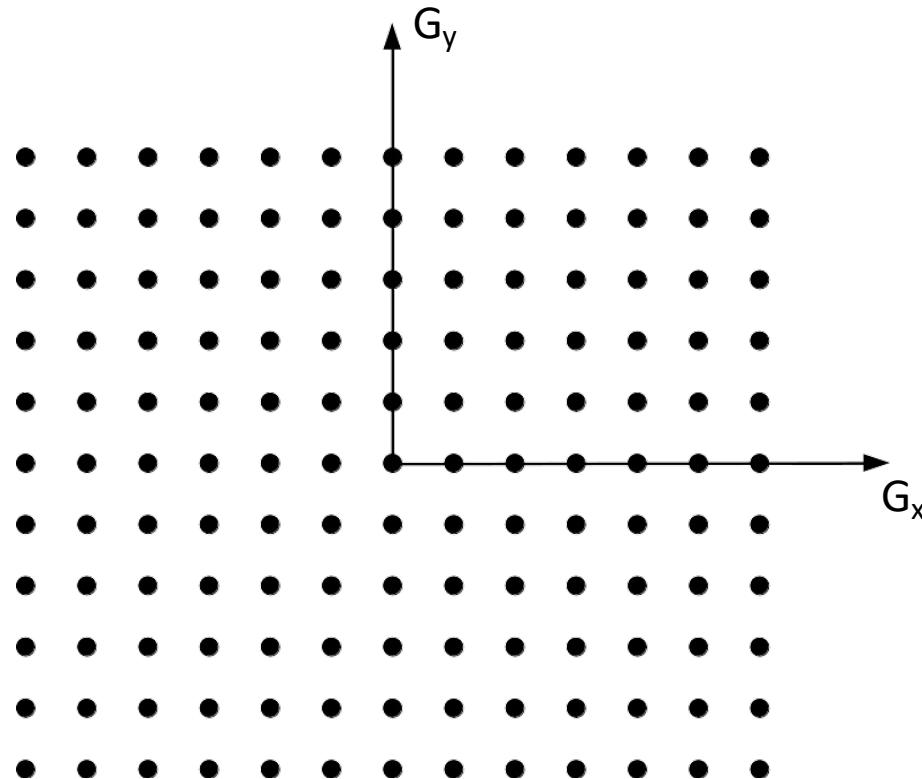
- **Hybrid functionals:** include a fraction of Fock exchange

$$E_{\text{xc}}^{\text{hyb}} = (1 - a_0) E_{\text{x}}^{\text{DFT}} + a_0 E_{\text{x}}^{\text{HF}} + E_{\text{c}}^{\text{DFT}}$$

# KS-DFT in practice: PW and PP

Wavefunctions are represented in Plane Waves (PW)

$$\phi_i(\mathbf{r}) = \sum_{\mathbf{G}} c(\mathbf{G}) \exp(i\mathbf{G} \cdot \mathbf{r})$$

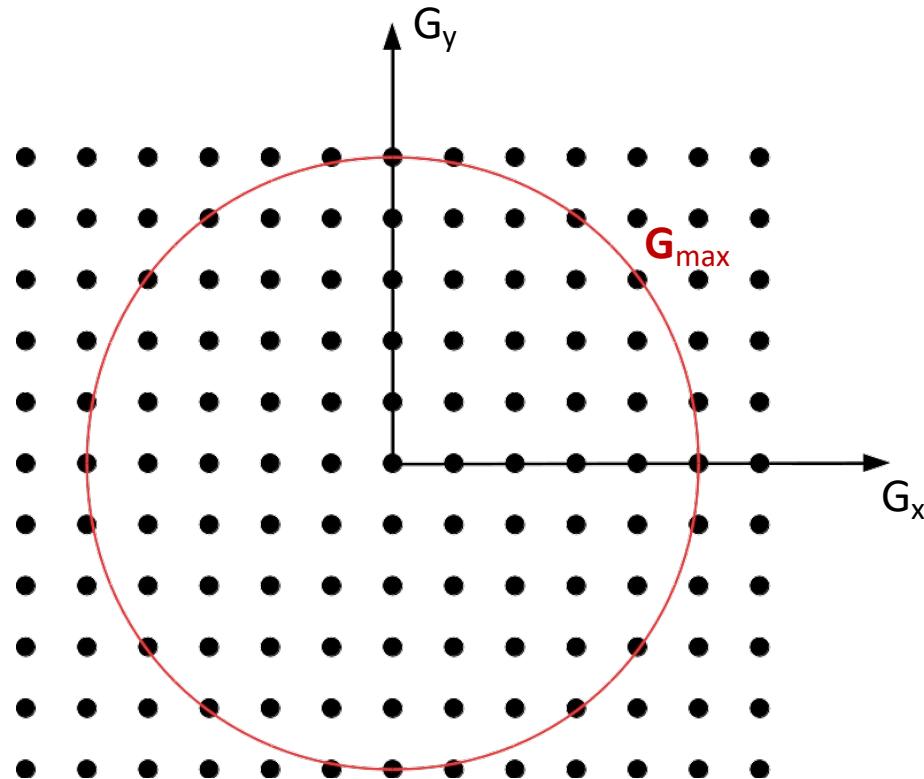


- Only Fourier components compatible with the periodicity of the lattice  $\mathbf{G}=n_1\mathbf{b}_1+n_2\mathbf{b}_2$

# KS-DFT in practice: PW and PP

Wavefunctions are represented in Plane Waves (PW)

$$\phi_i(\mathbf{r}) = \sum_{\mathbf{G}} c(\mathbf{G}) \exp(i\mathbf{G} \cdot \mathbf{r})$$

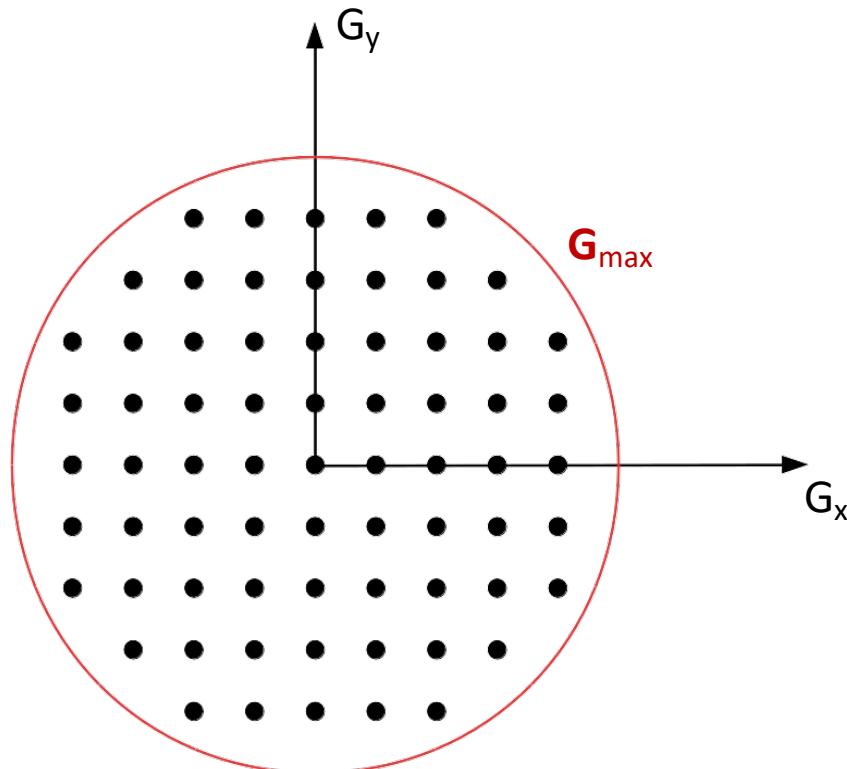


- Only Fourier components compatible with the periodicity of the lattice  $\mathbf{G}=n_1\mathbf{b}_1+n_2\mathbf{b}_2$
- All  $\mathbf{G}$  vectors with norm smaller than  $|\mathbf{G}|_{\max}$
- $|\mathbf{G}|_{\max}$  defined by setting a cutoff on the kinetic energy  $E_{\text{cut}} = \hbar^2/2\pi |\mathbf{G}|_{\max}^2$

# KS-DFT in practice: PW and PP

Wavefunctions are represented in Plane Waves (PW)

$$\phi_i(\mathbf{r}) = \sum_{\mathbf{G}} c(\mathbf{G}) \exp(i\mathbf{G} \cdot \mathbf{r})$$

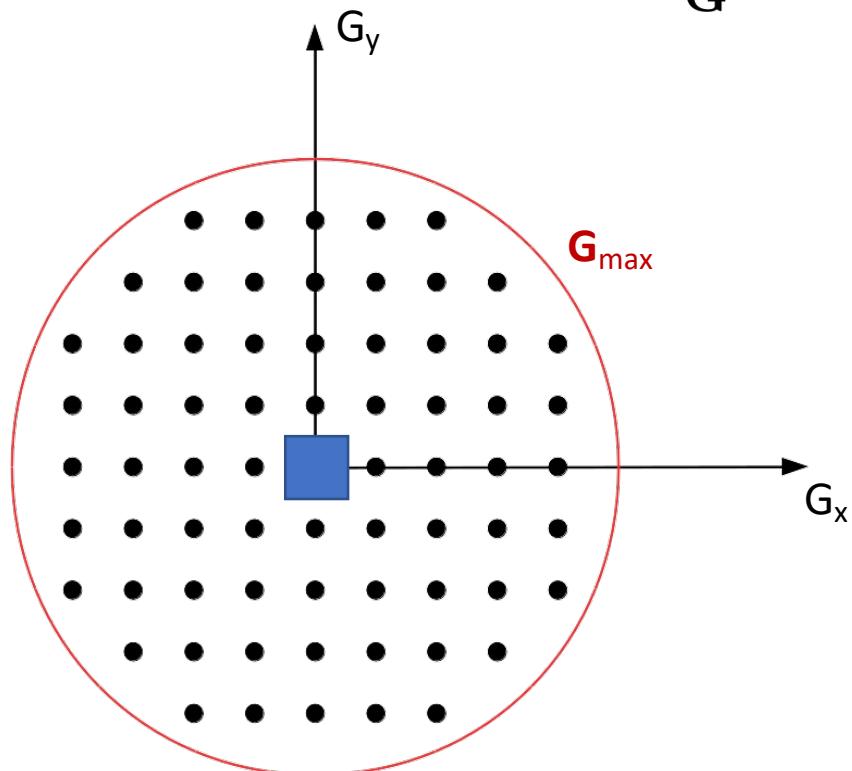


- Only Fourier components compatible with the periodicity of the lattice  $\mathbf{G}=n_1\mathbf{b}_1+n_2\mathbf{b}_2$
- All  $\mathbf{G}$  vectors with norm smaller than  $|\mathbf{G}|_{\max}$
- $|\mathbf{G}|_{\max}$  defined by setting a cutoff on the kinetic energy  $E_{\text{cut}} = \hbar^2/2\pi |\mathbf{G}|_{\max}^2$
- `ecutwfc`: QE Input parameter for  $E_{\text{cut}}$

# KS-DFT in practice: PW and PP

In a **periodic system** KS states are **Bloch functions**

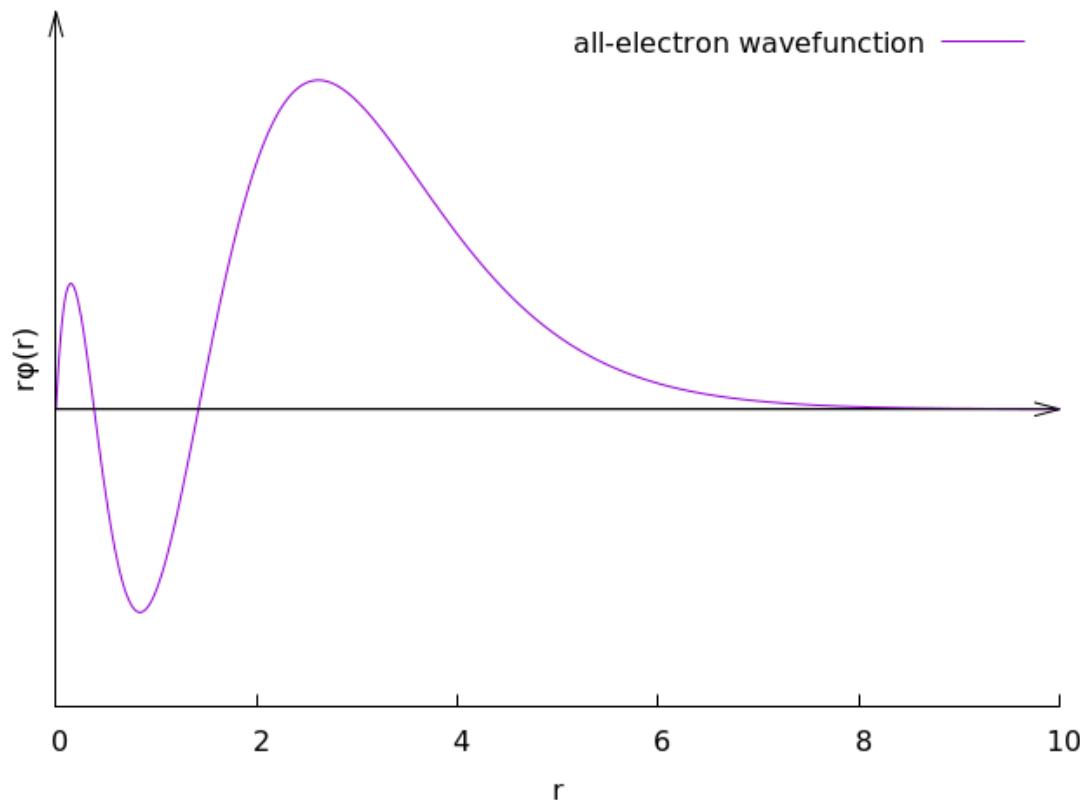
$$\psi_{\mathbf{k}n}(\mathbf{r}) = \exp[i\mathbf{k} \cdot \mathbf{r}] u_{\mathbf{k}n}(\mathbf{r}) = \sum_{\mathbf{G}} c_{\mathbf{k}n}(\mathbf{G}) \exp [i(\mathbf{G} + \mathbf{k}) \cdot \mathbf{r}]$$



- Only Fourier components compatible with the periodicity of the primitive cell  $\mathbf{G} = n_1 \mathbf{b}_1 + n_2 \mathbf{b}_2$
- $\mathbf{k}$  belong to the Brillouin zone (
- All  $\mathbf{G}$  vectors such that  $\hbar^2/2\pi | \mathbf{k} + \mathbf{G} |^2 < E_{\text{cut}}$
- Many properties requires integration over the  $\mathbf{k}$ -points -> BZ sampling (`K_POINTS` card)

# KS-DFT in practice: PW and PP

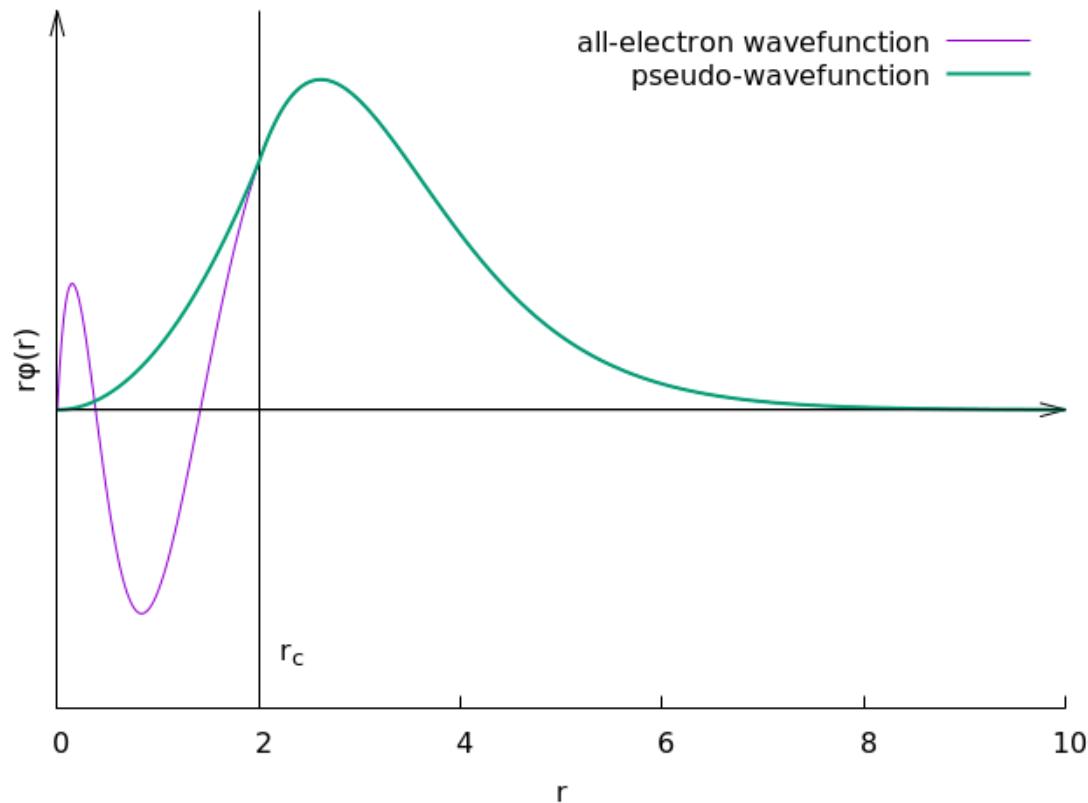
The use of pseudo-potentials allows to ignore chemically inert core electrons and make PW calculation feasible



- Strong oscillations close to the nuclei: large number of PW needed

# KS-DFT in practice: PW and PP

The use of pseudo-potentials allows to ignore chemically inert core electrons and make PW calculation feasible



- Strong oscillations close to the nuclei: large number of PW needed
- Replace the coulombic potential with a pseudo-potential that produces smooth pseudo-wavefunctions
- Different recipes: Norm-conserving, Ultra-soft, Projector-augmented-wave (PAW)

# Where to find PPs

Standard solid-state pseudopotentials (SSSP):

<https://www.materialscloud.org/discover/sssp/table/efficiency>

SSSP Efficiency (version 1.1.2)

$\Delta_{\text{eff}} = 0.44 \text{ meV}$

[Download Cutoffs table](#)

[Download Pseudos](#)

[Switch to SSSP Precision](#)

Updates v1.2.0 beta



\*

La 40<sub>(320)</sub>

Ce 40<sub>(320)</sub>

Pr 40<sub>(320)</sub>

Nd 40<sub>(320)</sub>

Pm 40<sub>(320)</sub>

Sm 40<sub>(320)</sub>

Eu 40<sub>(320)</sub>

Gd 40<sub>(320)</sub>

Tb 40<sub>(320)</sub>

Dy 40<sub>(320)</sub>

Ho 40<sub>(320)</sub>

Er 40<sub>(320)</sub>

Tm 40<sub>(320)</sub>

Yb 40<sub>(320)</sub>

Lu 45<sub>(360)</sub>

# Where to find PPs

Standard solid-state pseudopotentials (SSSP):

<https://www.materialscloud.org/discover/sssp/table/efficiency>

Quantum Espresso Website:

<https://www.quantum-espresso.org/pseudopotentials/>

Optimized Norm-Conserving Vanderbilt (ONCV): SG15

[http://www.quantum-simulation.org/potentials/sg15\\_oncv/](http://www.quantum-simulation.org/potentials/sg15_oncv/)

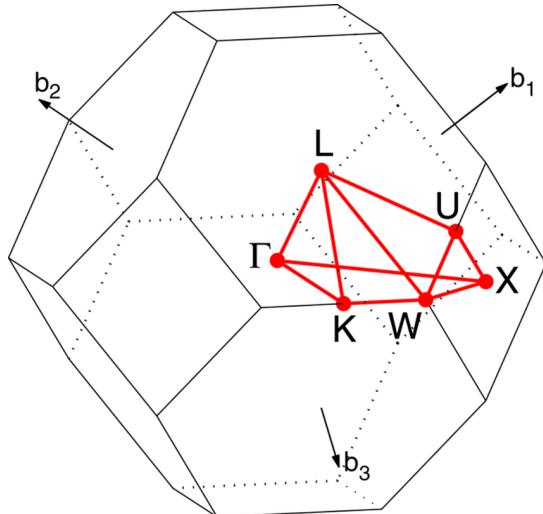
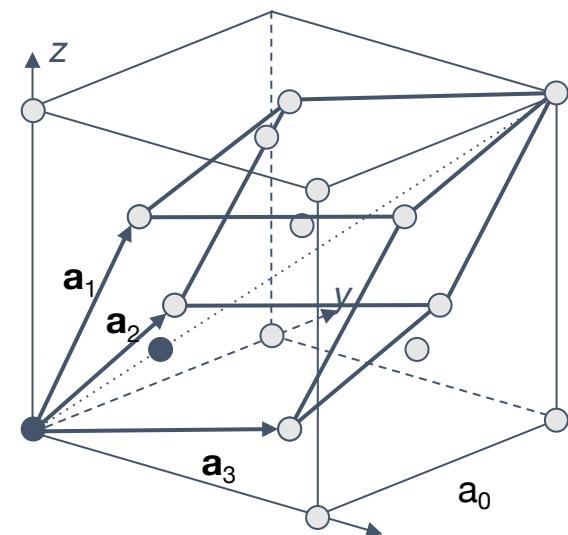
Optimized Norm-Conserving Vanderbilt (ONCV): pseudo-dojo

<http://www.pseudo-dojo.org/>

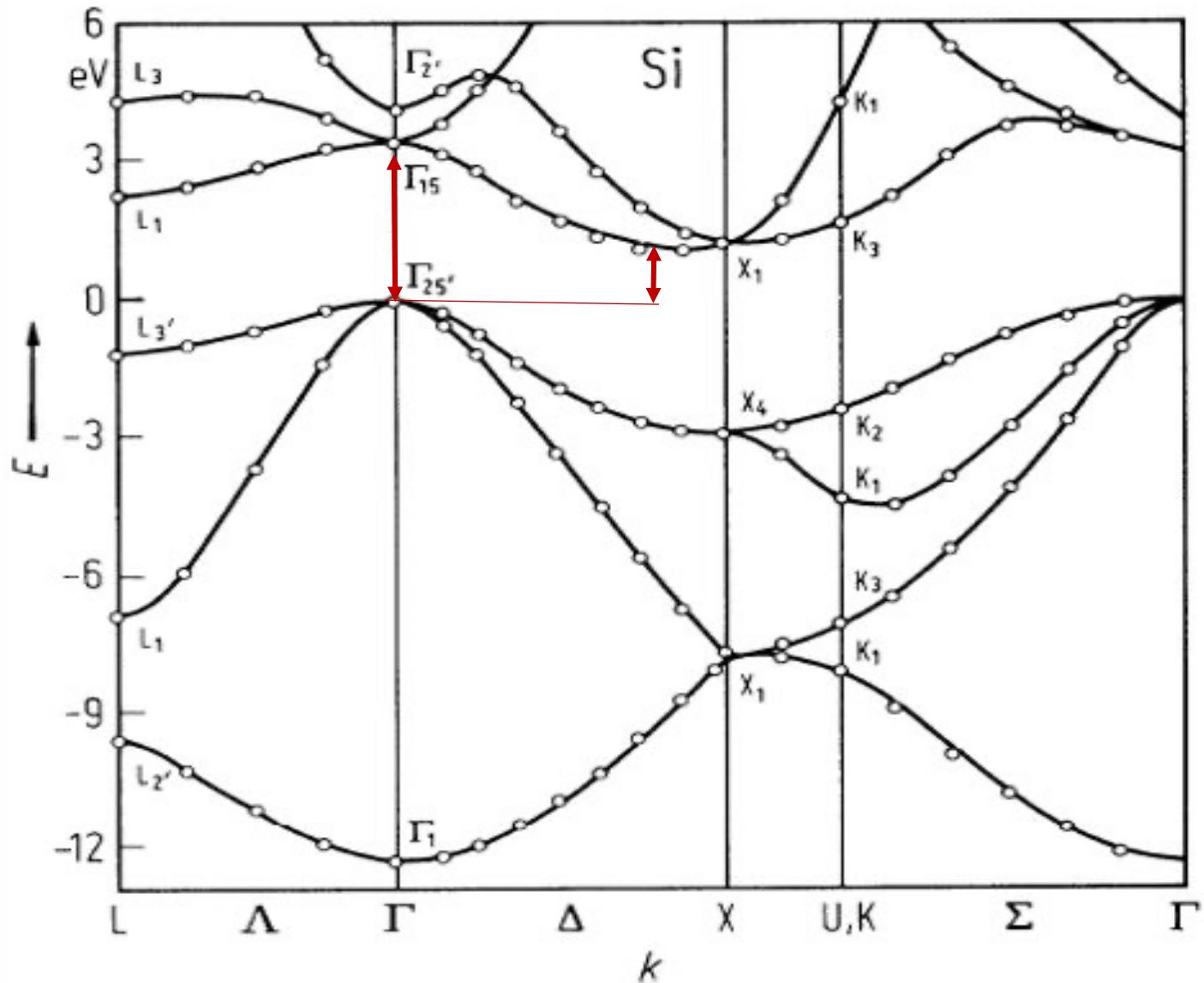
# Outline

- **Brief recap on KS-DFT.** KS equations ad common approximations to the xc energy functional, KC in practice
- **Exercise 0.** Standard DFT: PBE band structure of FCC Silicon
- **Exercise 1-2.** Hybrid DFT: electronic structure of FCC Silicon
- **Exercise 3-4.** Meta-GGA: electronic structure of FCC Silicon and BCC Iron using SCAN

# FCC Silicon

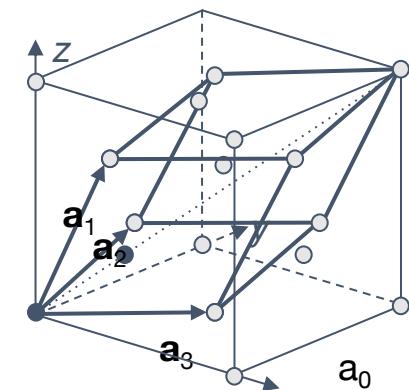


Goal of exercise0



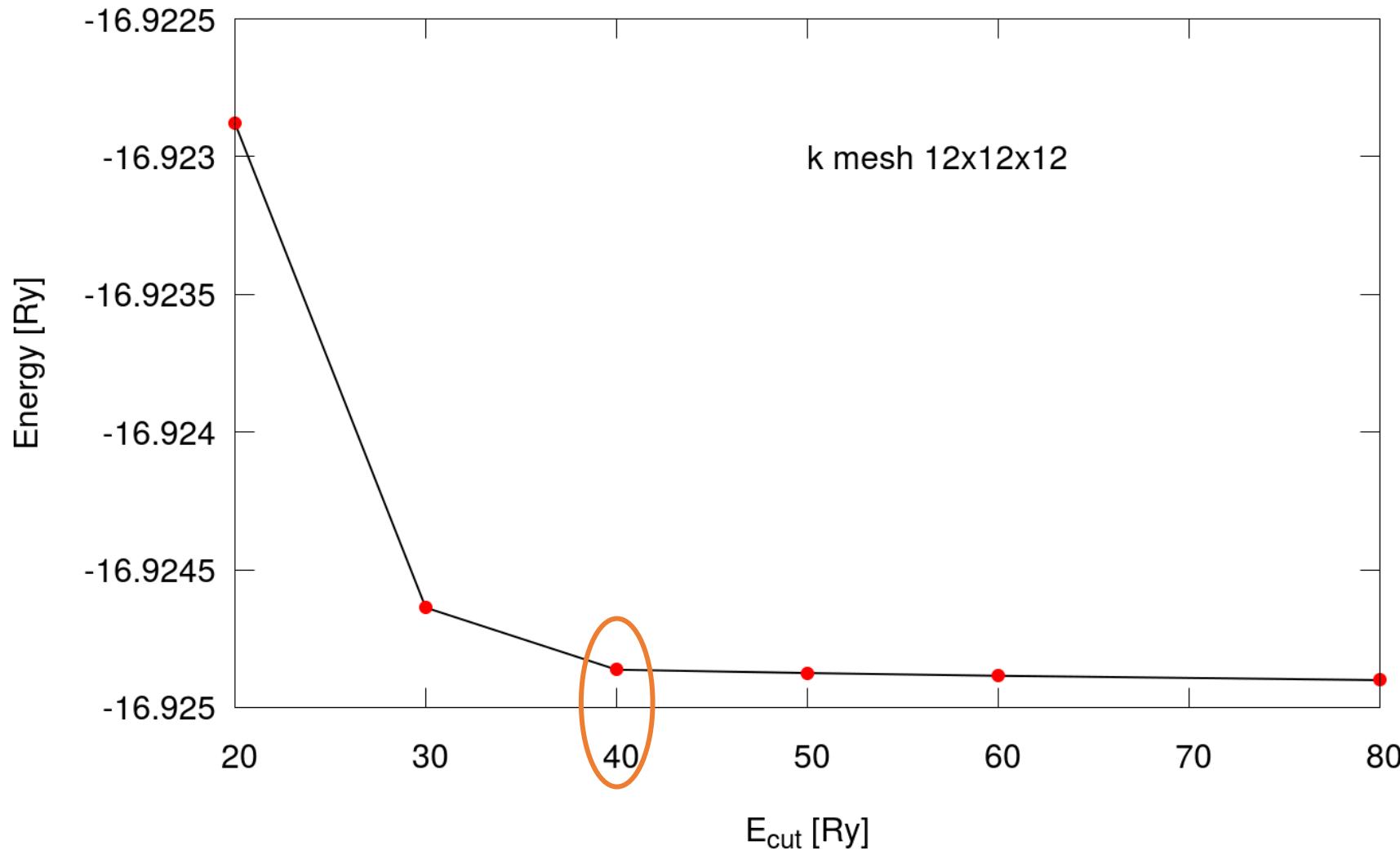
# Silicon PBE electronic structure: SCF calculation

```
&control
  calculation='scf'           ← Self-consistent field calculation
  prefix='Si'
  pseudo_dir = '../files/pseudo' ← Directory containing the PPs
 outdir='./out'               ← Temporary files re stored here
/
&system
  ibrav = 2, celldm(1) = 10.262,          ← Specify here the crystal structure:
  nat = 2, ntyp = 1,                      ← FCC, alat=10.20 Bohr, 2 Si atoms
  ecutwfc = xxx,                         ← Cutoff for the Wave-Functions
/
&electrons
  conv_thr = 1.d-8                     ← Convergence threshold for the SCF problem
/
ATOMIC_SPECIES
Si 28.086 Si.pbe_PseudoDojo.UPF ← Pseudopotential file
ATOMIC_POSITIONS {alat}
Si 0.00 0.00 0.00                   ← Atomic positions
Si 0.25 0.25 0.25
K_POINTS {automatic}
XXX YYY ZZZ 0 0 0                 ← BZ sampling
```

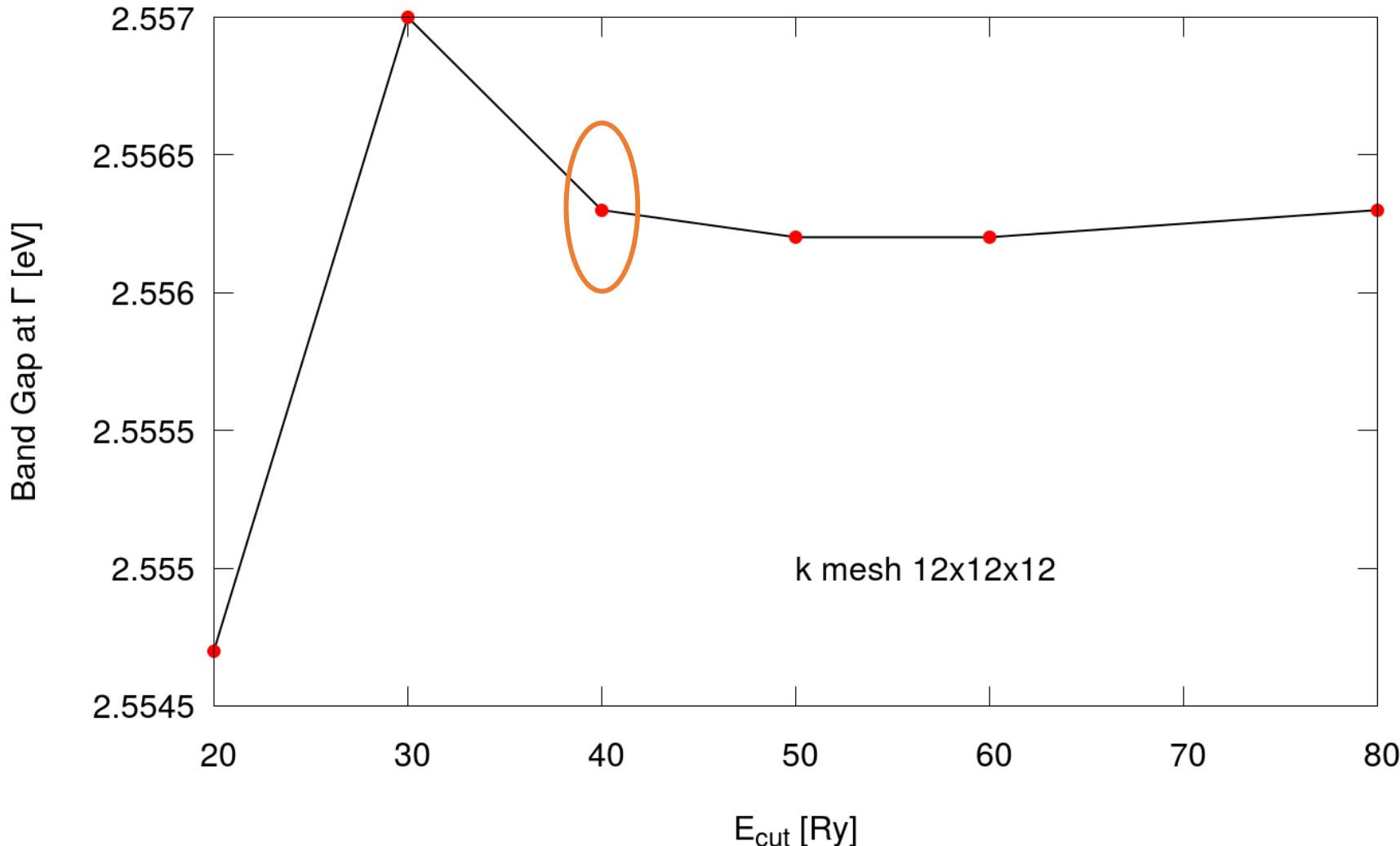


[https://www.quantum-espresso.org/Doc/INPUT\\_PW.html](https://www.quantum-espresso.org/Doc/INPUT_PW.html)

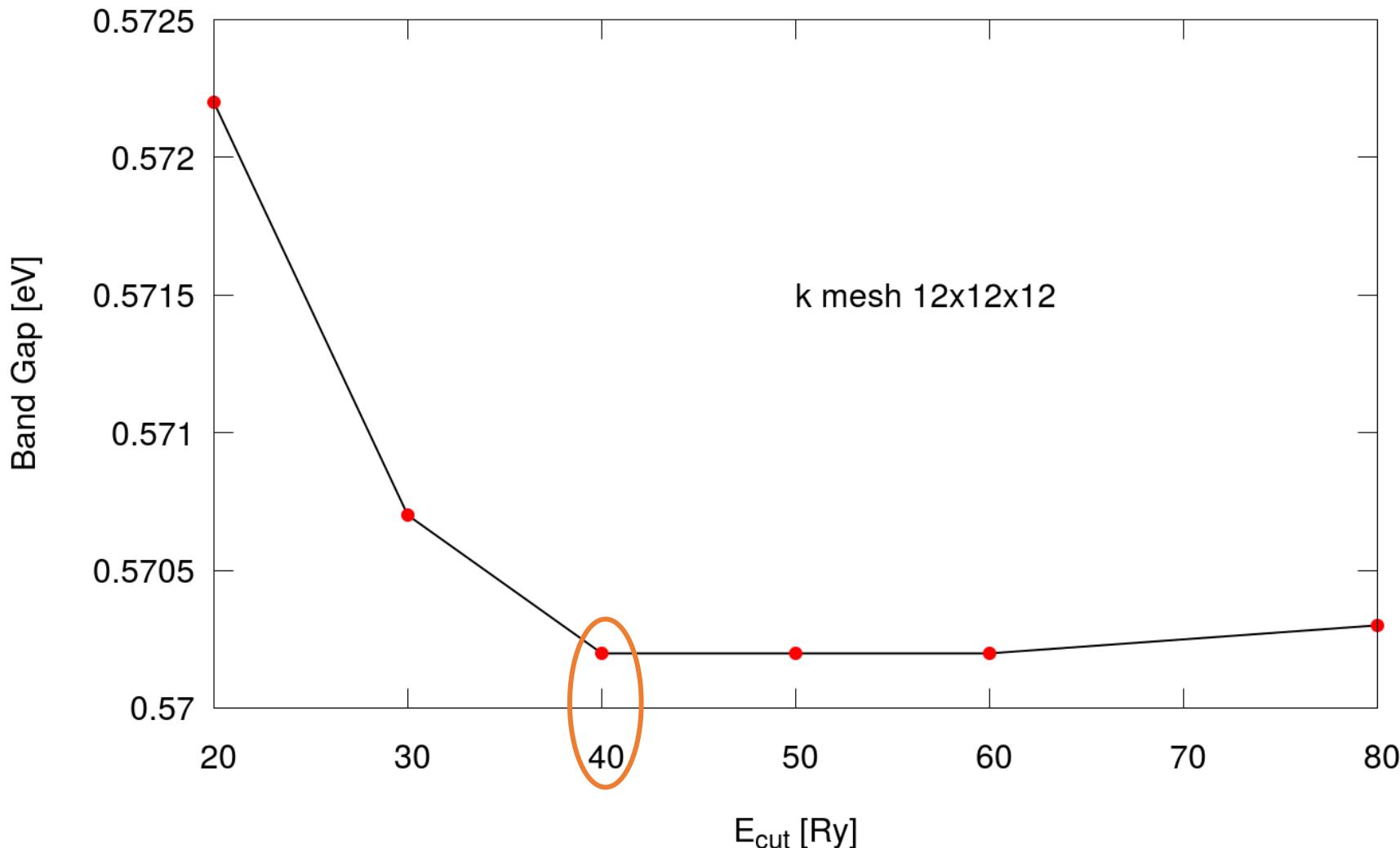
# Convergence wrt ecut: total energy



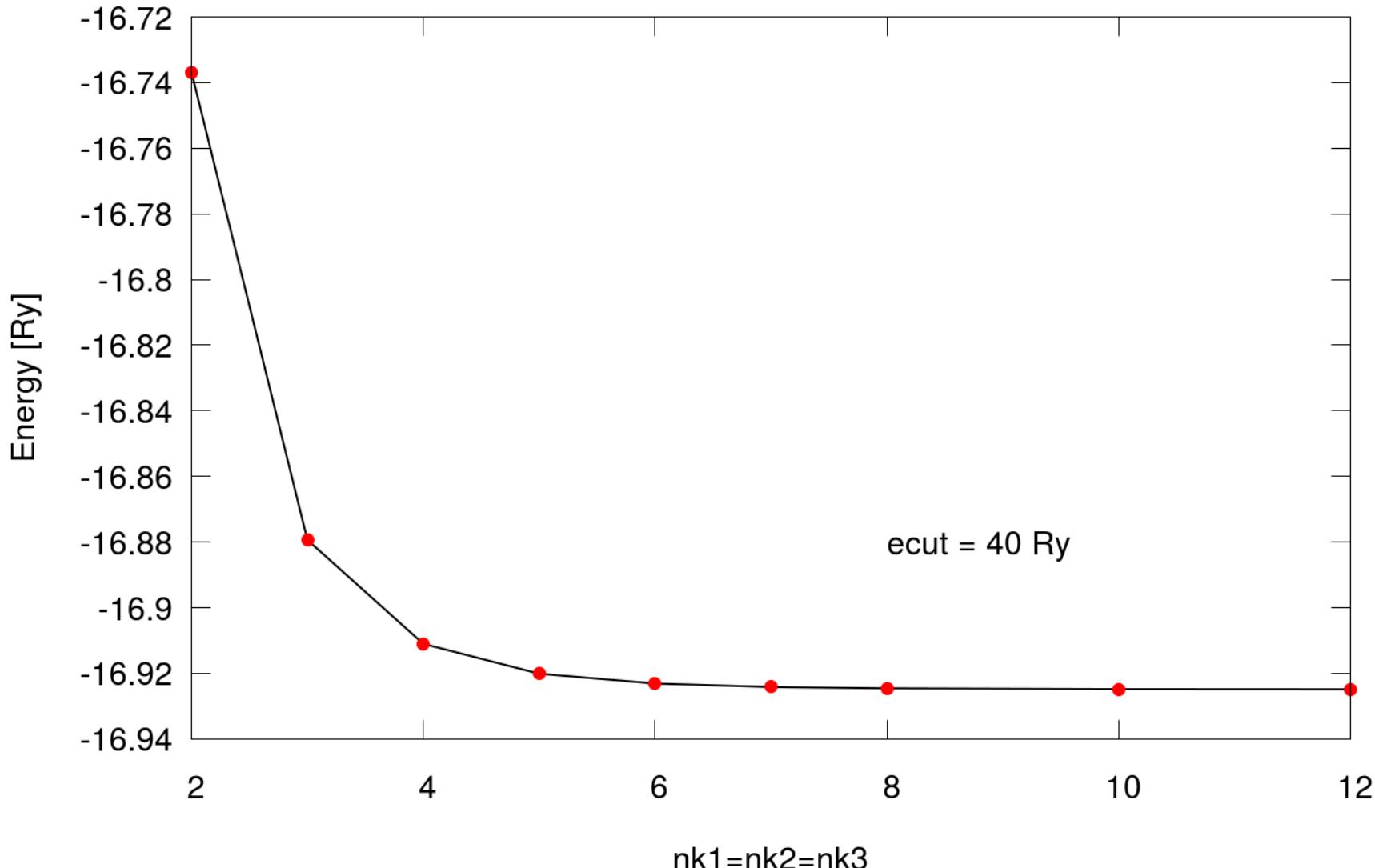
# Convergence wrt $E_{\text{cut}}$ : direct band gap



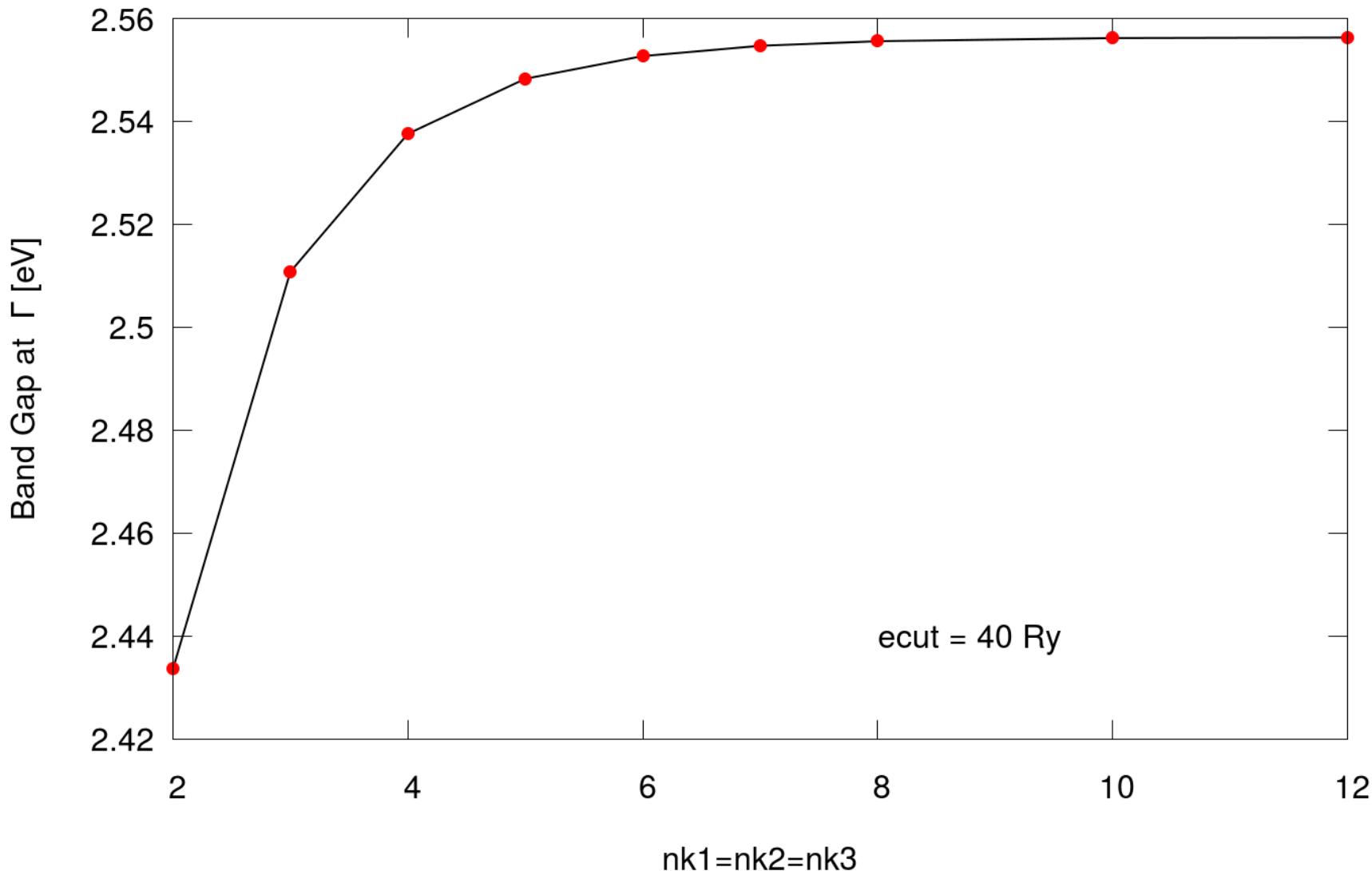
# Convergence wrt $E_{\text{cut}}$ : indirect band gap



# Convergence wrt K\_POINTS: total energy



# Convergence wrt K\_POINTS: direct band gap



# Check-out the git repo

- The hands-on materials are available at github
- Clone the repository (or Download it):

```
git clone https://github.com/materialscloud-org/hubbard-koopmans-2022
```

- Go to the exercise0 folder

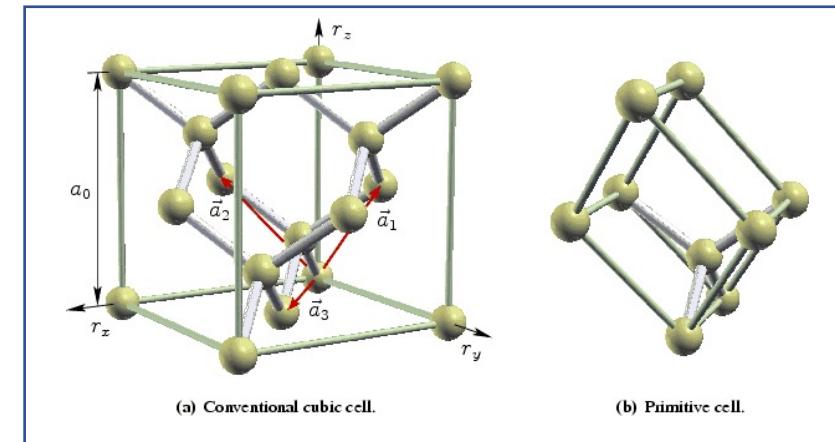
```
cd hubbard-koopmans-2022/Day1/exercise0
```

# Silicon PBE electronic structure: SCF calculation

```
&control
  calculation='scf'
  prefix='Si'
 outdir='./out'
/
&system
  ibrav = 2,  celldm(1) = 10.20,
  nat = 2,  ntyp = 1,
  ecutwfc = 40, ← Converged value for the Wave-Functions cutoff
/
&electrons
  conv_thr = 1.d-8
/
ATOMIC_SPECIES
Si 28.086 Si.pbe_PseudoDojo.UPF

ATOMIC_POSITIONS {alat}
Si 0.00 0.00 0.00
Si 0.25 0.25 0.25

K_POINTS {automatic}
12 12 12 0 0 0 ← Converged value for the
BZ sampling
```



[https://www.quantum-espresso.org/Doc/INPUT\\_PW.html](https://www.quantum-espresso.org/Doc/INPUT_PW.html)

# Silicon PBE electronic structure: BANDS calculation

```
&control
  calculation='bands' ← "Bands" calculation
```

```
  ...
```

```
/
```

```
&system
  ...
```

```
/
```

```
&electrons
  ...
```

```
/
```

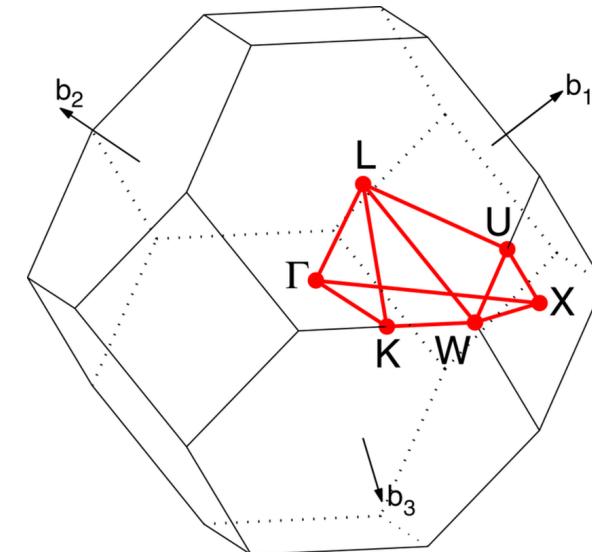
```
ATOMIC_SPECIES
  ...
```

```
ATOMIC_POSITIONS {alat}
  ...
```

```
K_POINTS {crystal_b}
```

```
5 ← Number of special k-points provided
```

|         |         |         |    |   |   |
|---------|---------|---------|----|---|---|
| 0.50000 | 0.50000 | 0.50000 | 20 | ! | I |
| 0.00000 | 0.00000 | 0.00000 | 20 | ! | G |
| 0.50000 | 0.00000 | 0.50000 | 20 | ! | X |
| 0.37500 | 0.37500 | 0.75000 | 20 | ! | K |
| 0.00000 | 0.00000 | 0.00000 | 0  | ! | G |

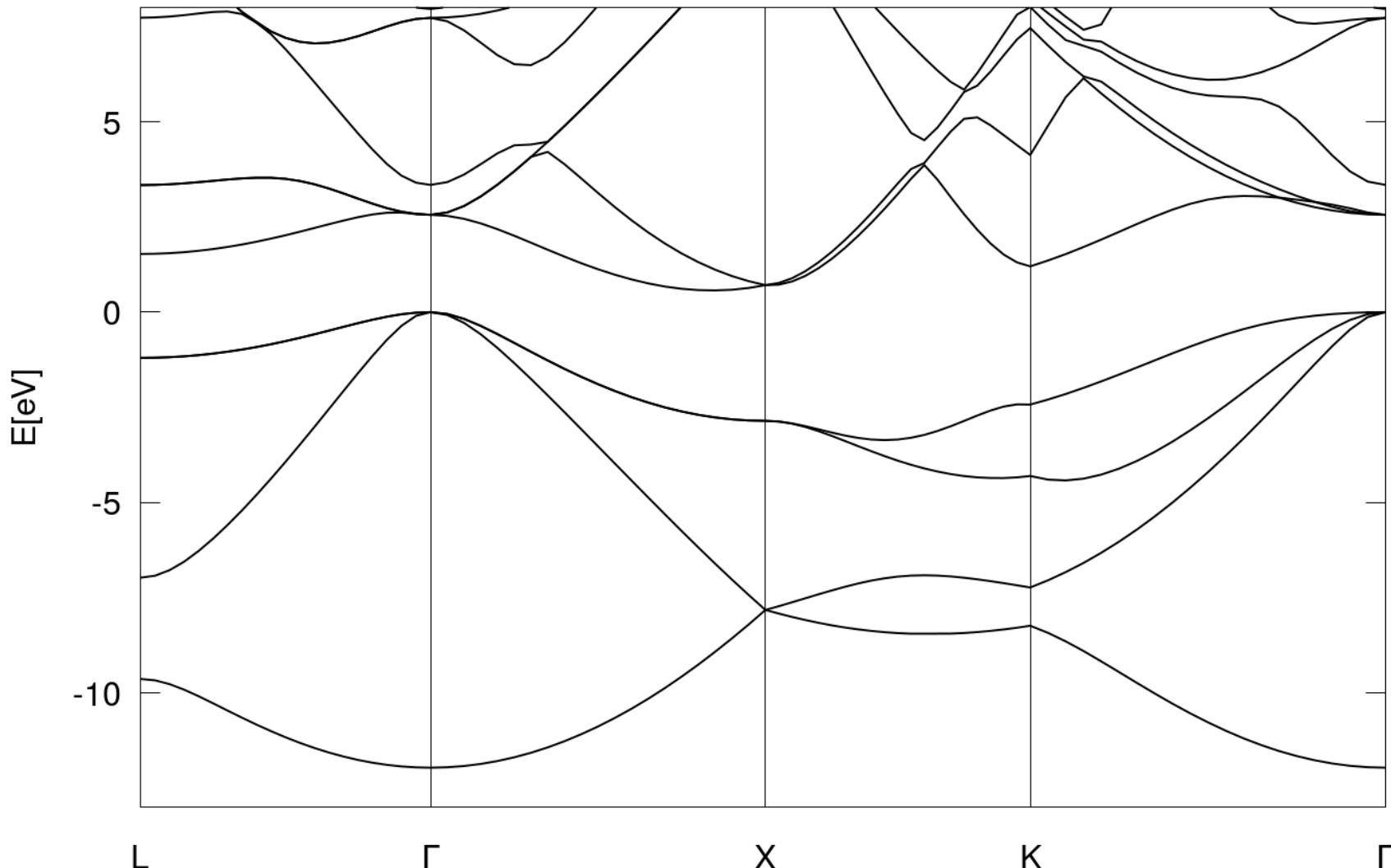


Number of special **k**-points provided

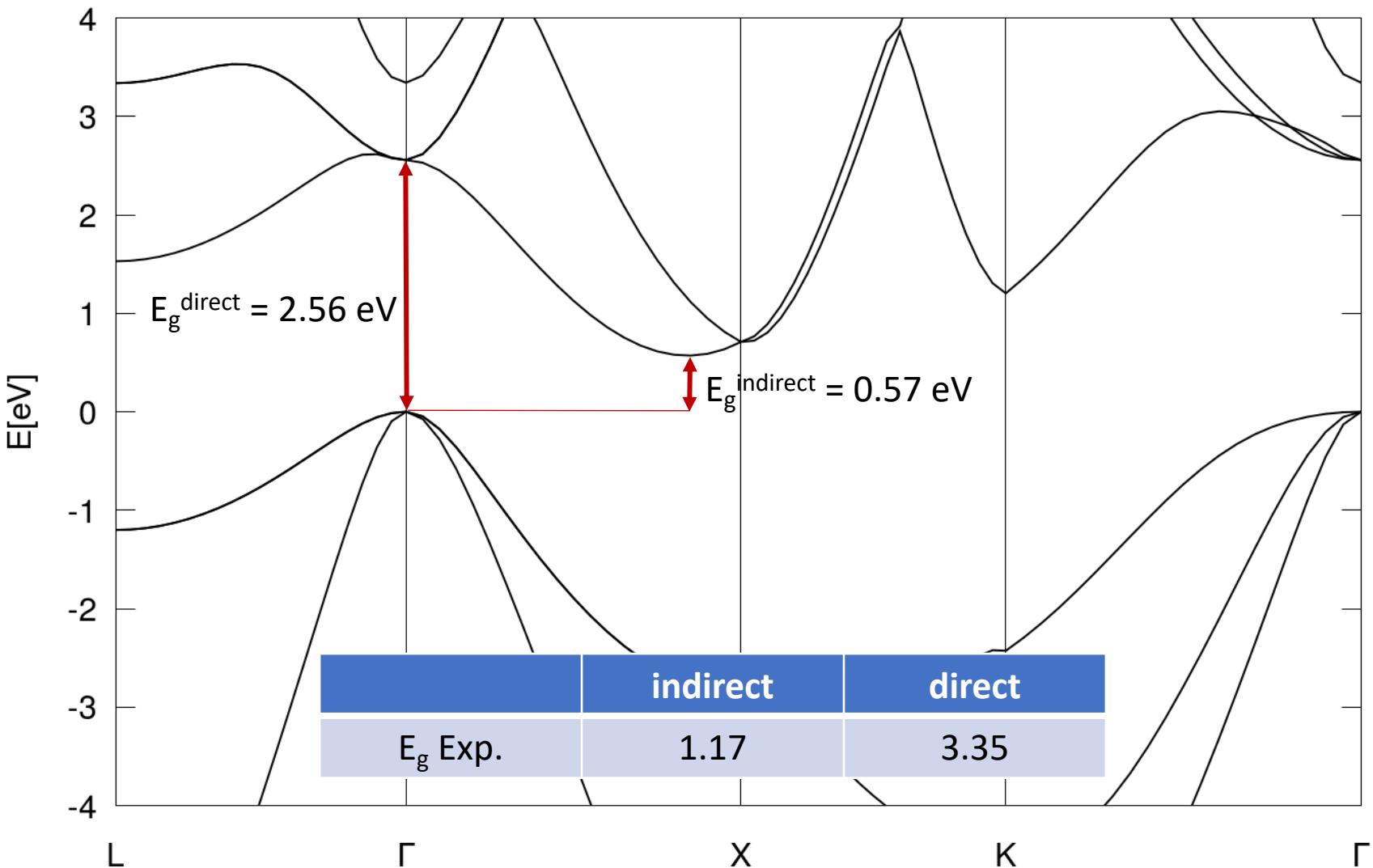
Coordinates of the special **k**-points and number of additional points QE generates between two adjacent special k-points.

[https://www.quantum-espresso.org/Doc/INPUT\\_PW.html](https://www.quantum-espresso.org/Doc/INPUT_PW.html)

# PBE Silicon band structure



# PBE Silicon band structure



# Outline

- **Brief recap on KS-DFT.** KS equations, common approximations to the xc energy functional, KS in practice
- **Exercise 0.** Standard DFT: PBE electronic structure of FCC Silicon
- **Exercise 1-2.** Hybrid DFT: electronic structure of FCC Silicon
- **Exercise 3-4.** Meta-GGA: electronic structure of FCC Silicon and BCC Iron using SCAN

# Hybrid functionals

Mix a fraction of exact-exchange from HF theory with semi-local exchange

- **3-parameters:** combination of x-HF, x-B88(xGGA), c-LYP(cGGA) and xc-LDA

$$E_{\text{xc}}^{\text{B3LYP}} = (1 - a_0)E_{\text{x}}^{\text{LDA}} + a_0 E_{\text{x}}^{\text{HF}} + a_x \Delta E_{\text{x}}^{\text{B88}} + a_c E_{\text{c}}^{\text{LYP}} + (1 - a_c)E_{\text{c}}^{\text{LDA}}$$

Fitted on thermochemical data:  $a_0 = 0.20$ ,  $a_x = 0.72$ ,  $a_c = 0.81$

[Becke et al. JCP **98**, 5648 (1993), Stephens et al. J. Phys. Chem. **98**, 11623 (1994)]

# Hybrid functionals

Mix a fraction of exact-exchange from HF theory with semi-local exchange

- **3-parameters:** combination of x-HF, x-B88(xGGA), c-LYP(cGGA) and xc-LDA

$$E_{\text{xc}}^{\text{B3LYP}} = (1 - a_0)E_{\text{x}}^{\text{LDA}} + a_0 E_{\text{x}}^{\text{HF}} + a_x \Delta E_{\text{x}}^{\text{B88}} + a_c E_{\text{c}}^{\text{LYP}} + (1 - a_c)E_{\text{c}}^{\text{LDA}}$$

Fitted on thermochemical data:  $a_0 = 0.20$ ,  $a_x = 0.72$ ,  $a_c = 0.81$

[Becke et al. JCP **98**, 5648 (1993), Stephens et al. J. Phys. Chem. **98**, 11623 (1994)]

- **Simplification:**  $a_x = (1 - a_0)$ ,  $a_c = 1$ . Only 1 mixing parameter [Becke et al. JCP **104**, 1040 (1996)]

$$E_{\text{xc}}^{\text{hyb}} = (1 - a_0)E_{\text{x}}^{\text{DFT}} + a_0 E_{\text{x}}^{\text{HF}} + E_{\text{c}}^{\text{DFT}}$$

Comparison with 4° order perturbation theory:  $a_0 = 1/4$ : **PBE0** [Perdew et al. JCP 105, 9982 (1996)]

$$E_{\text{xc}}^{\text{PBE0}} = \frac{3}{4}E_{\text{x}}^{\text{DFT}} + \frac{1}{4}E_{\text{x}}^{\text{HF}} + E_{\text{c}}^{\text{PBE}}$$

# Range-separated Hybrid functionals

Basic Idea: split the electron-electron interaction in short and long range

$$\frac{1}{r_{12}} = \underbrace{\frac{a + b \operatorname{erf}(\mu r_{12})}{r_{12}}}_{\text{Fock exchange}} + \underbrace{\frac{1 - [a + b \operatorname{erf}(\mu r_{12})]}{r_{12}}}_{\text{DFT exchange}}$$

Range-separation parameter

Long-Range

Range-separation parameter

Short-Range

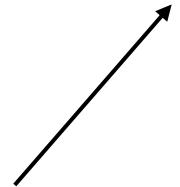
- $b = 0 \rightarrow$  Conventional hybrid functional. E.g. **b = 0** and **a = 0.25**  $\rightarrow$  PBE0
- **a = -b = 0.25** and  **$\mu = 0.106 \text{ bohr}^{-1}$**   $\rightarrow$  HSE
- **a+b=1**  $\rightarrow$  correct asymptotic ( $1/r$ ) in **finite systems**
- **a+b=1/ε**  $\rightarrow$  correct asymptotic ( $1/\varepsilon r$ ) in **extended systems**
- $\mu$  can be tuned to satisfy the IP theorem (Optimal tuning) [Lecture by Prof. Kronik]

# Exact-exchange energy

$$E^{\text{HF}} = -\frac{1}{2} \sum_{\mathbf{k}, v} \sum_{\mathbf{q}, v'} \int d\mathbf{r} d\mathbf{r}' \frac{\psi_{\mathbf{k}, v}^*(\mathbf{r}) \psi_{\mathbf{k}-\mathbf{q}, v'}(\mathbf{r}) \psi_{\mathbf{k}, v}(\mathbf{r}') \psi_{\mathbf{k}-\mathbf{q}, v'}^*(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}$$

In Reciprocal space:

$$E^{\text{HF}} = -\frac{4\pi}{2\Omega} \frac{\Omega}{(2\pi)^3} \int_{BZ} d\mathbf{q} \sum_G \frac{A(\mathbf{q} + \mathbf{G})}{|\mathbf{q} + \mathbf{G}|^2}$$



**Integration over the BZ:**

`nqx1, nqx2, nqx3`

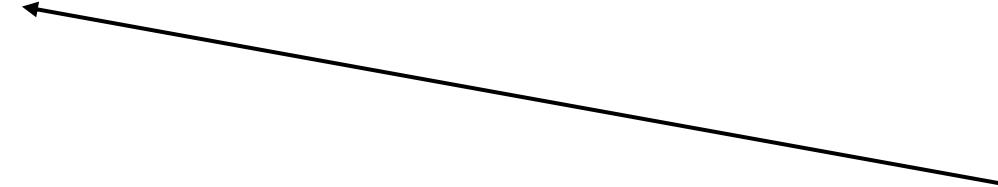
Need to be compatible  
with the k mesh

**PW expansion**

`ecutfock`

Equal to `ecutrho` by default.  
Can be reduced (but not  
below `ecutwfc`)

$$A(\mathbf{q} + \mathbf{G}) = \frac{\Omega}{(2\pi)^3} \int_{BZ} d\mathbf{k} |\rho_{\mathbf{k}-\mathbf{q}, v'}^*(\mathbf{q} + \mathbf{G})|^2$$



**Integrable divergence** for  $\mathbf{q} + \mathbf{G} > 0$

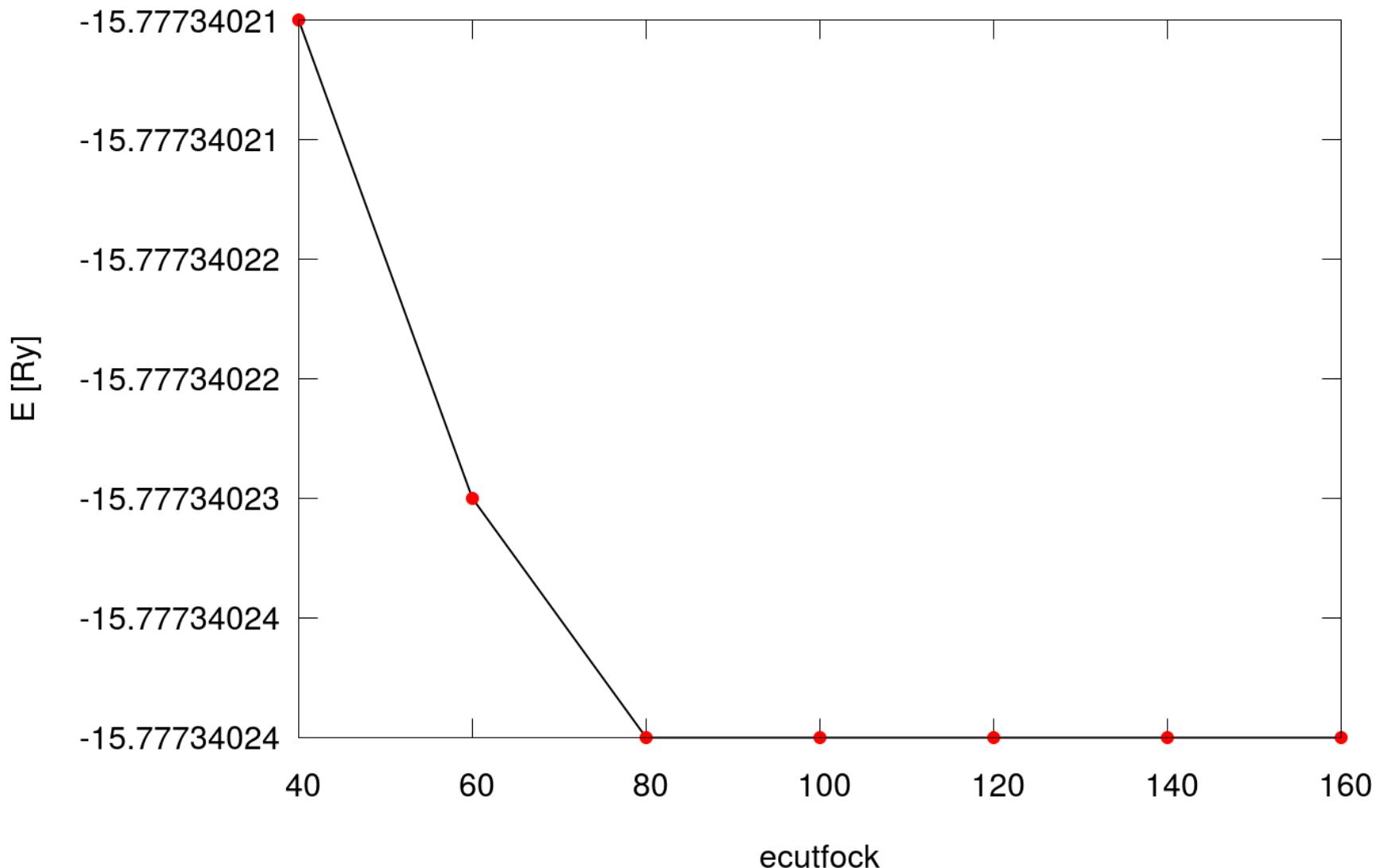
`exxdiv_treatment='gygi-baldereschi'`  
`x_gamma_extrapolation = .true.`

# Silicon PBE0 electronic structure: SCF calculation

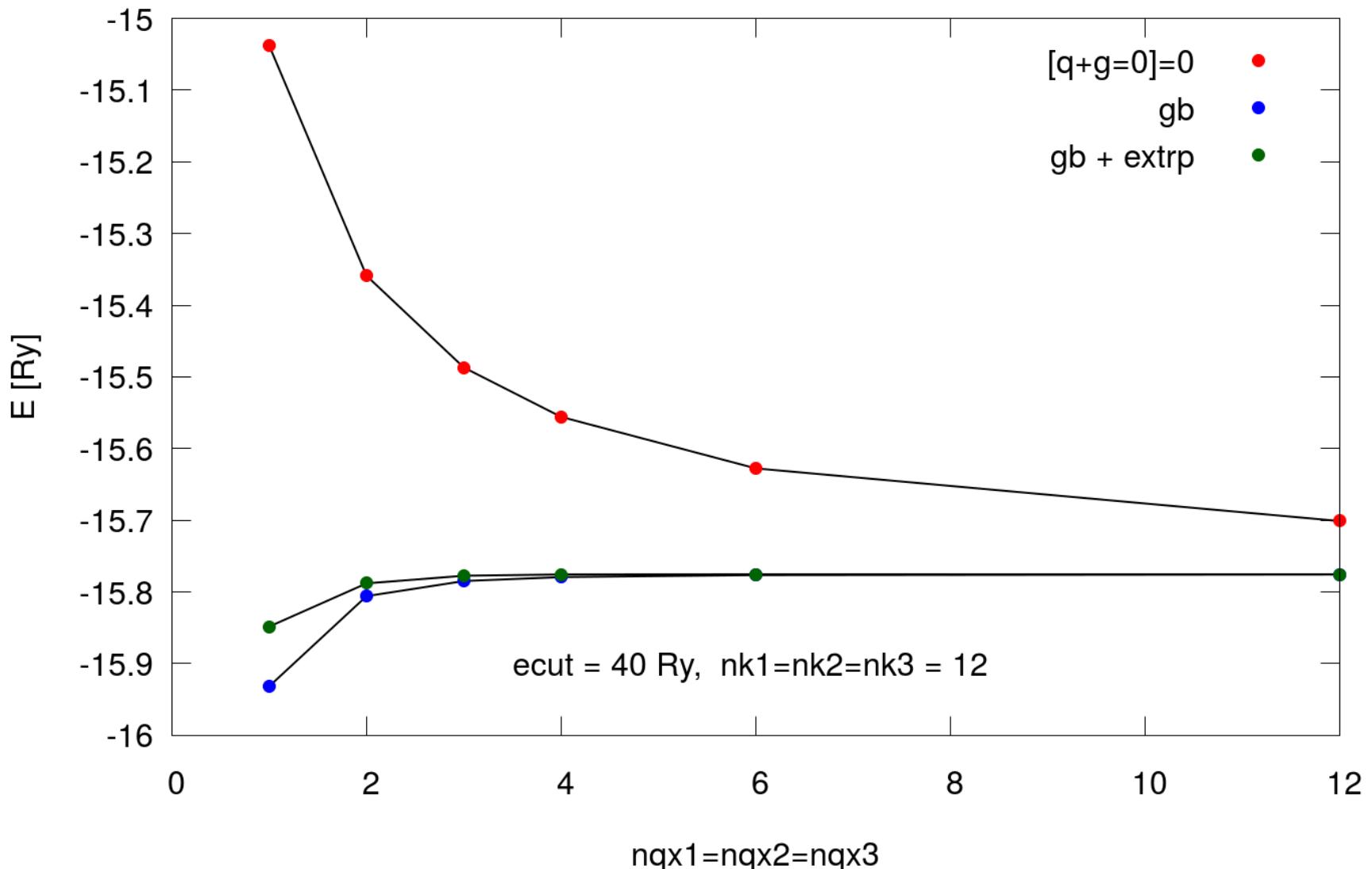
```
&control
  ...
/
&system
  ...
  input_dft      = "pbe0"           ← Specify which hybrid functional: PBE0/B3LYP/HSE
  nqx1 = XXX, nqx2 = XXX, nqx3 = XXX
  exxdiv_treatment = "gb"
  x_gamma_extrapolation = .true.
  ecutfock        = YYY
  exx_fraction    = ZZZ
  screening_parameter = WWW           ← You can change the fraction of EXX and/or the
                                         range separation parameter.
/
&electrons
  ...
/
ATOMIC_SPECIES
...
ATOMIC_POSITIONS {alat}
...
K_POINTS {automatic}
12 12 12 0 0 0
```

[https://www.quantum-espresso.org/Doc/INPUT\\_PW.html](https://www.quantum-espresso.org/Doc/INPUT_PW.html)

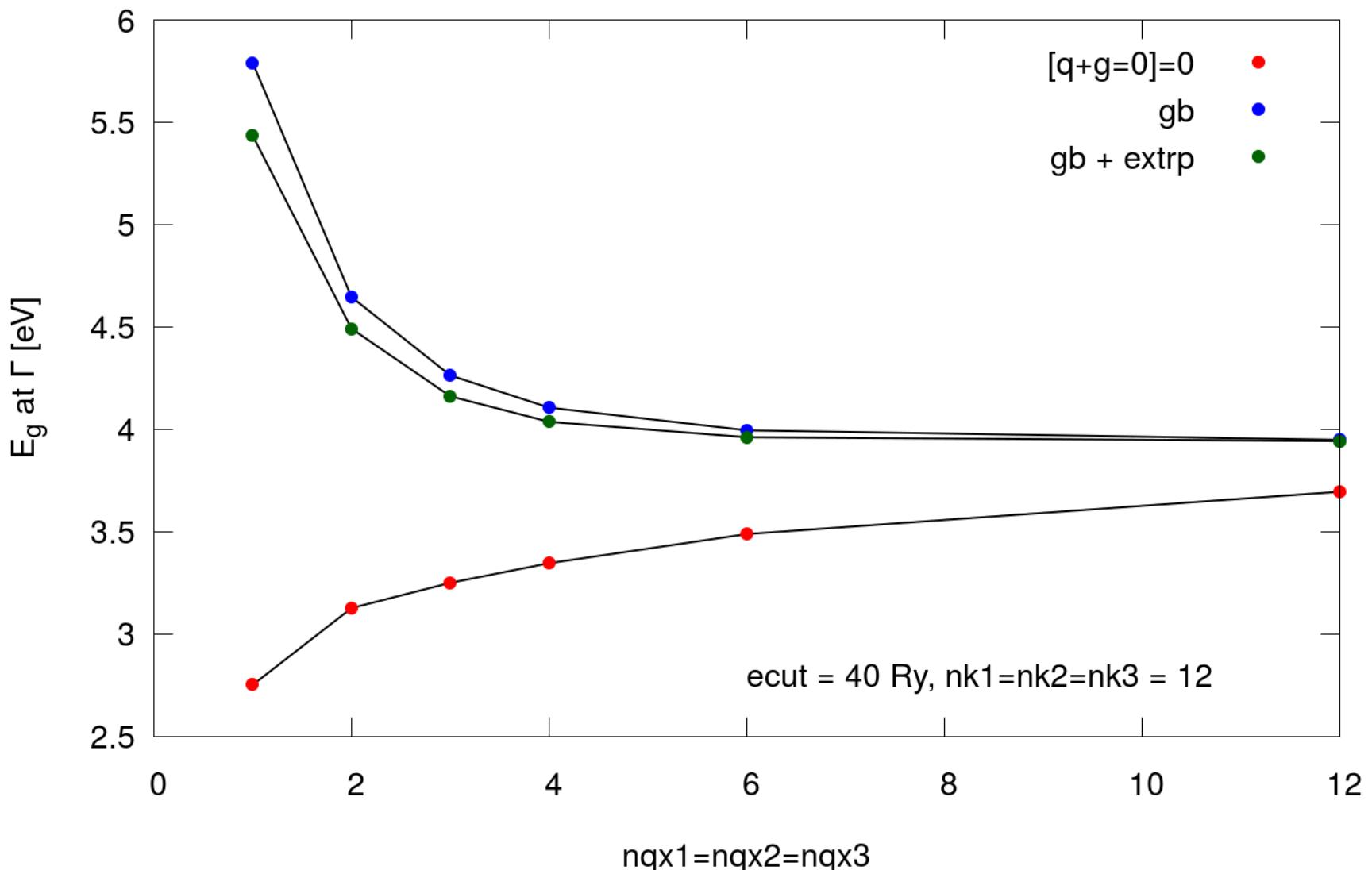
# PBE0 energy convergence wrt ecutfock



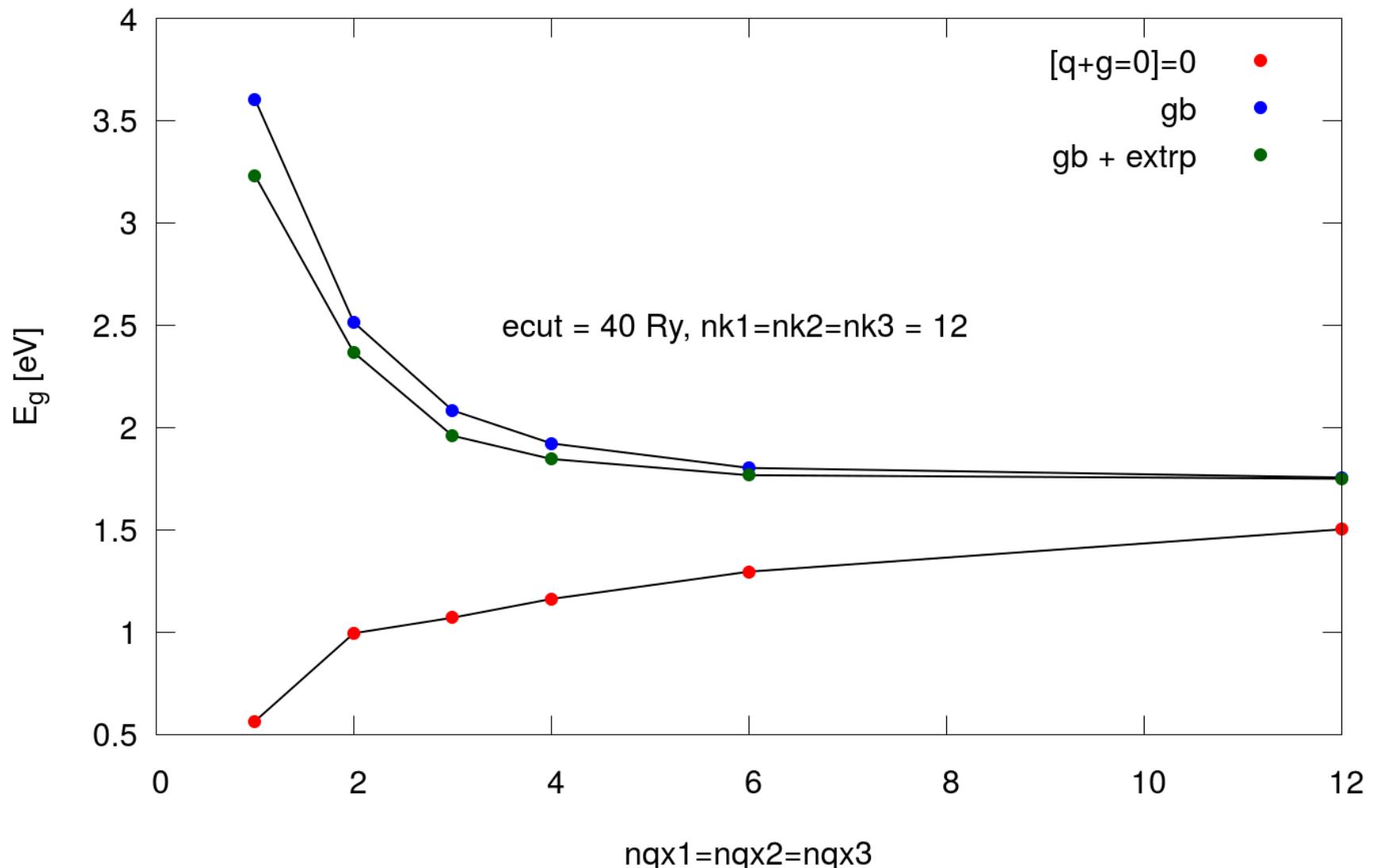
# PBE0 energy convergence wrt nqx



# PBEO direct band Gap convergence wrt nqx



# PBE0 indirect band Gap convergence wrt nqx



# Outline

- **Brief recap on KS-DFT.** KS equations, common approximations to the xc energy functional, KS in practice
- **Exercise 0.** Standard DFT: PBE electronic structure of FCC Silicon
- **Exercise 1-2.** Hybrid DFT: electronic structure of FCC Silicon
- **Exercise 3-4.** Meta-GGA: electronic structure of FCC Silicon and BCC Iron using SCAN

# Check-out the git repo

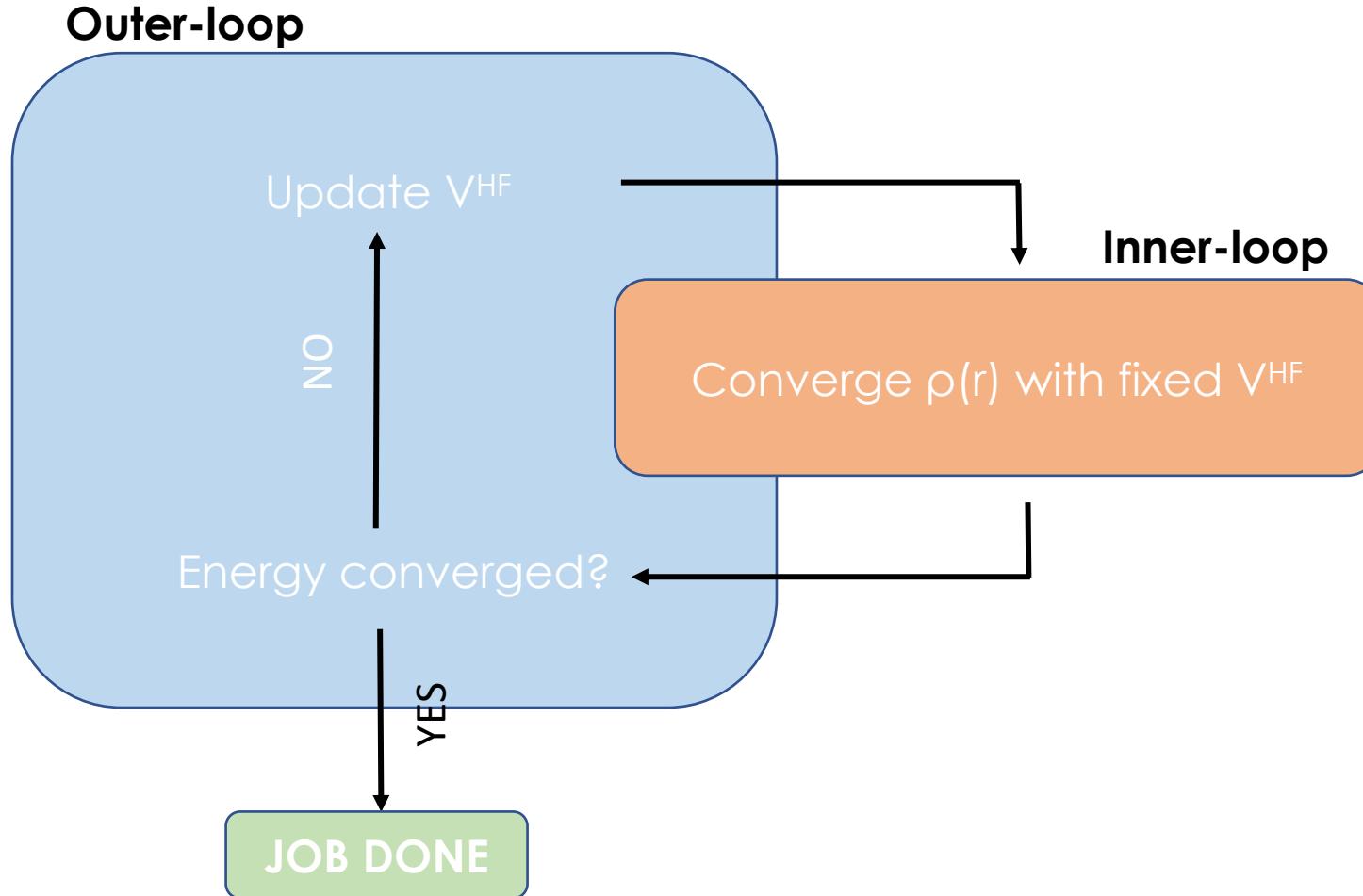
- Go to the exercise1 folder

```
cd hubbard-koopmans-2022/Day1/exercise1
```

# Silicon PBE0 electronic structure: SCF calculation

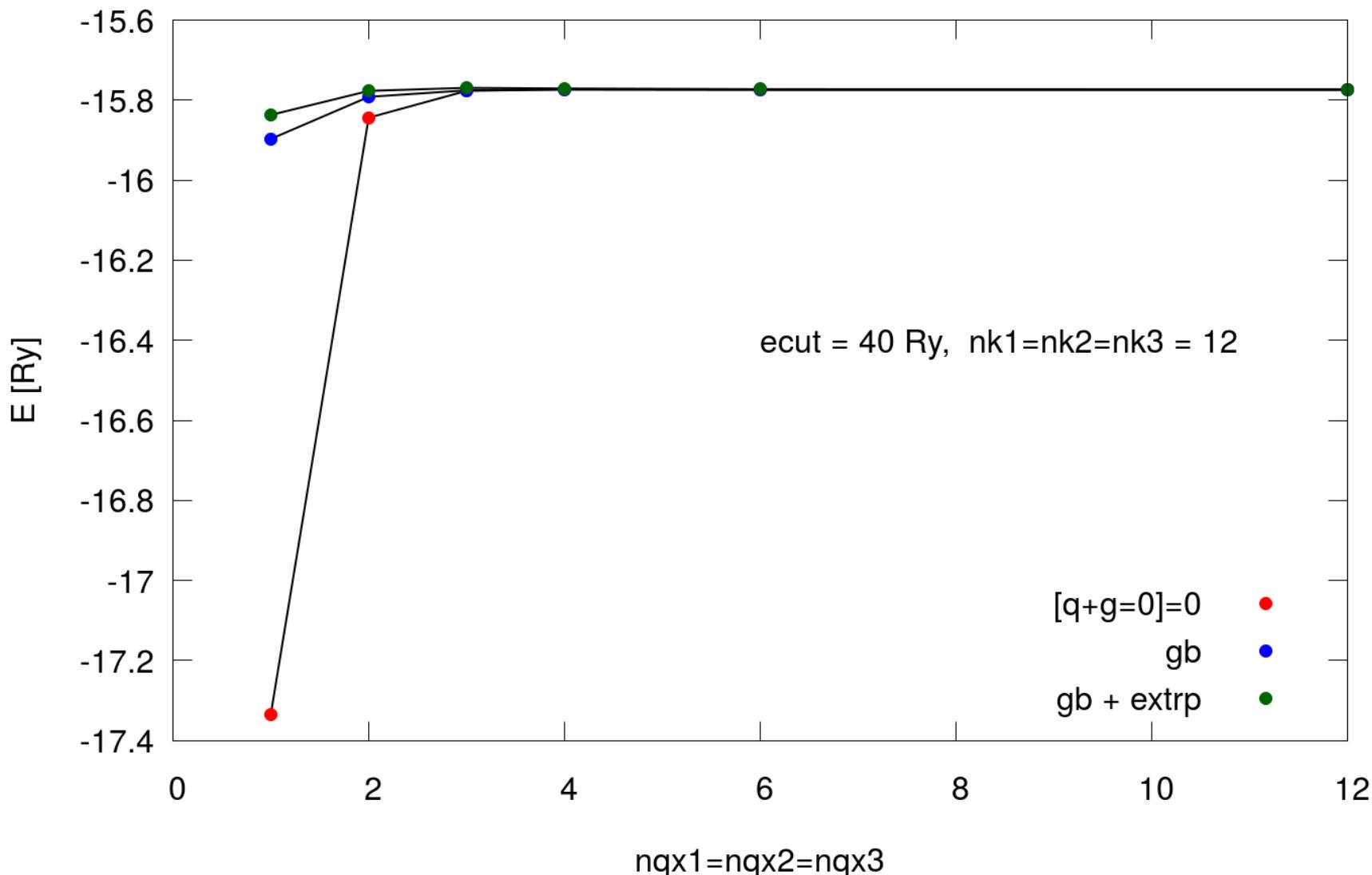
```
&control
  ...
/
&system
  ...
  input_dft      = "pbe0"
nqx1 = 4, nqx2 = 4, nqx3 = 4           ← Converged parameters for nqx
  exxdiv_treatment = "gb"
  x_gamma_extrapolation = .true.
ecutfock        = 40                   ← Converged value for ecutfock
/
&electrons
  ...
/
ATOMIC_SPECIES
...
ATOMIC_POSITIONS {alat}
...
K_POINTS {automatic}
12 12 12 0 0 0
```

# Nested self-consistent loop

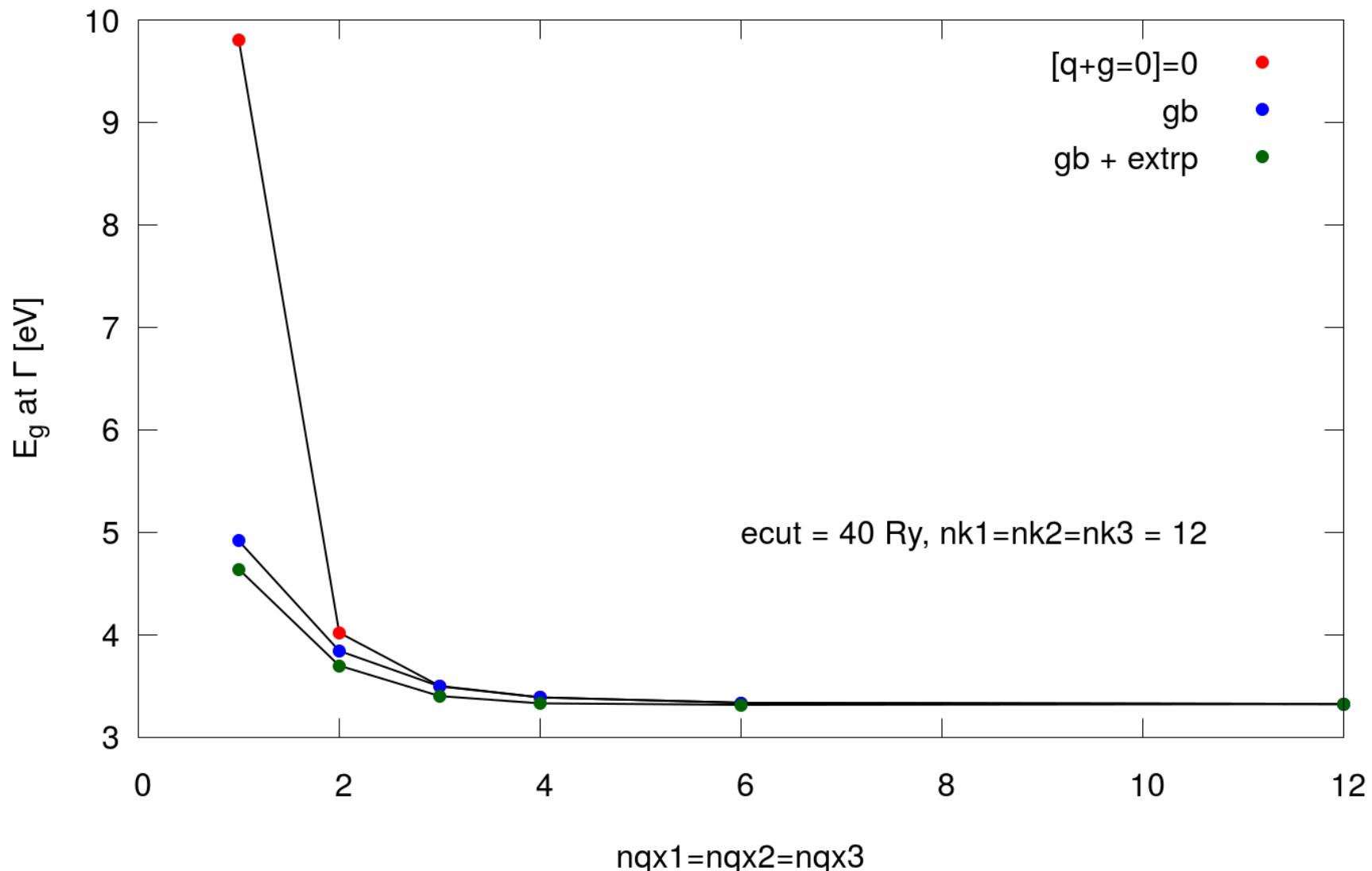


1. Normal scf (LDA or similar) is performed
2. Hybrid functional is switched.  $V^{HF}$  is calculated using the current best wfcs
3. A new scf is performed keeping  $V^{HF}$  fixed
4. Exchange error ( $d_{exx}$ ) is computed
5. If  $d_{exx} > \text{thr}$  the procedure is repeated from 1.

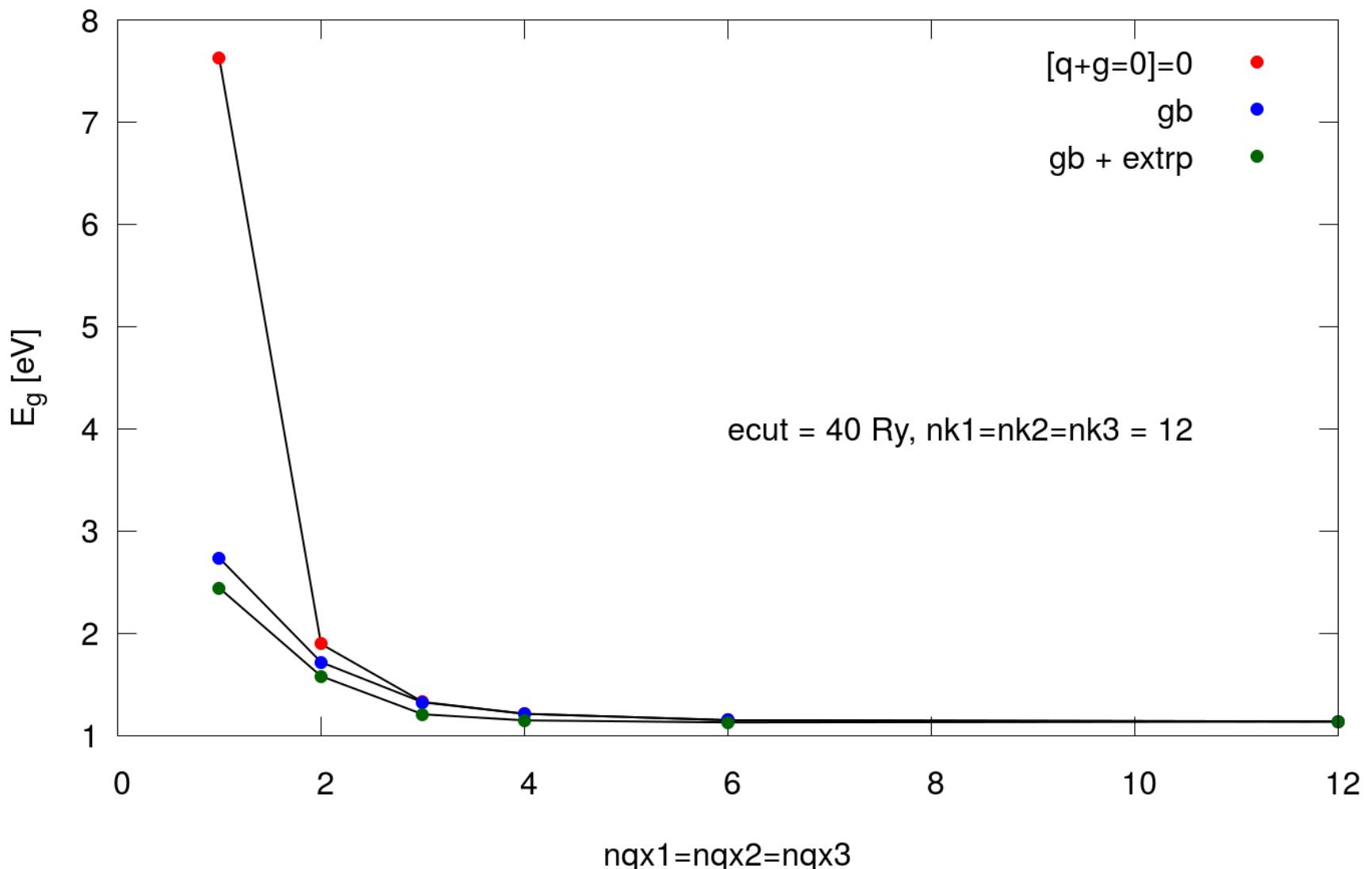
# HSE energy convergence wrt nqx



# HSE direct band Gap convergence wrt nqx



# HSE indirect band Gap convergence wrt nqx



# Silicon HSE electronic structure: SCF calculation

```
&control
  ...
/
&system
  ...
  input_dft      = "hse"
nqx1 = 4, nqx2 = 4, nqx3 = 4           ← Converged parameters for nqx
  exxdiv_treatment = "gb"
  x_gamma_extrapolation = .true.
ecutfock        = 40                   ← Converged value for ecutfock
/
&electrons
  ...
/
ATOMIC_SPECIES
...
ATOMIC_POSITIONS {alat}
...
K_POINTS {automatic}
12 12 12 0 0 0
```

# Silicon band Gap: summary

ecut = 40 Ry, k mesh: 12x12x12  
ecutfock = 40 Ry, q mesh: 4x4x4

| Functional  | indirect Gap | Direct Gap  | Band width  |
|-------------|--------------|-------------|-------------|
| PBE         | 0.57         | 2.56        | 11.96       |
| PBE0        | 1.85         | 4.04        | 13.32       |
| HSE         | 1.15         | 3.33        | 13.26       |
| <b>Exp.</b> | <b>1.17</b>  | <b>3.35</b> | <b>12.5</b> |

# Hybrids: how do they perform

## Atomization energies

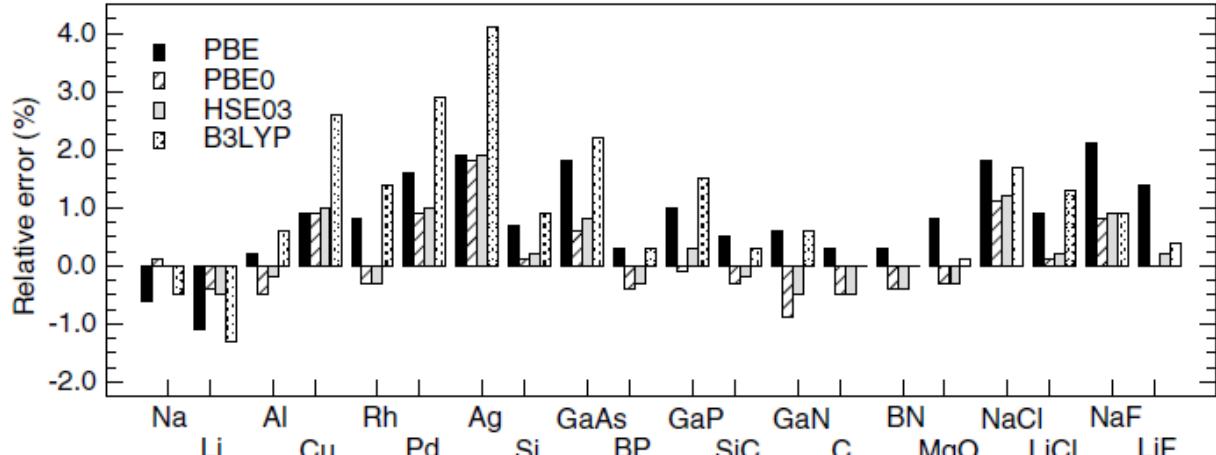
| Functional   | MAE(G2)    | MAE(G2-1)  | Max AE(G2) | Max AE(G2-1) |
|--------------|------------|------------|------------|--------------|
| LDA          | 83.7       | 36.4       | 216        | 84           |
| PBE          | 17.1       | 8.6        | 52         | 26           |
| BLYP         | 7.1        | 4.7        | 28         | 15           |
| <b>B3LYP</b> | <b>3.1</b> | <b>2.4</b> | <b>20</b>  | <b>10</b>    |
| <b>PBE0</b>  | <b>4.8</b> | <b>3.5</b> | <b>24</b>  | <b>10</b>    |

Mean Absolute Error (MAE) in Kcal/mol for the G2 (148 molecules) and G2-1( (55 molecules) sets.

Curtiss et al. JCP 106, 1063 (1997); Scuseria et al. JCP 110, 5029 (1999)

# Hybrids: how do they perform

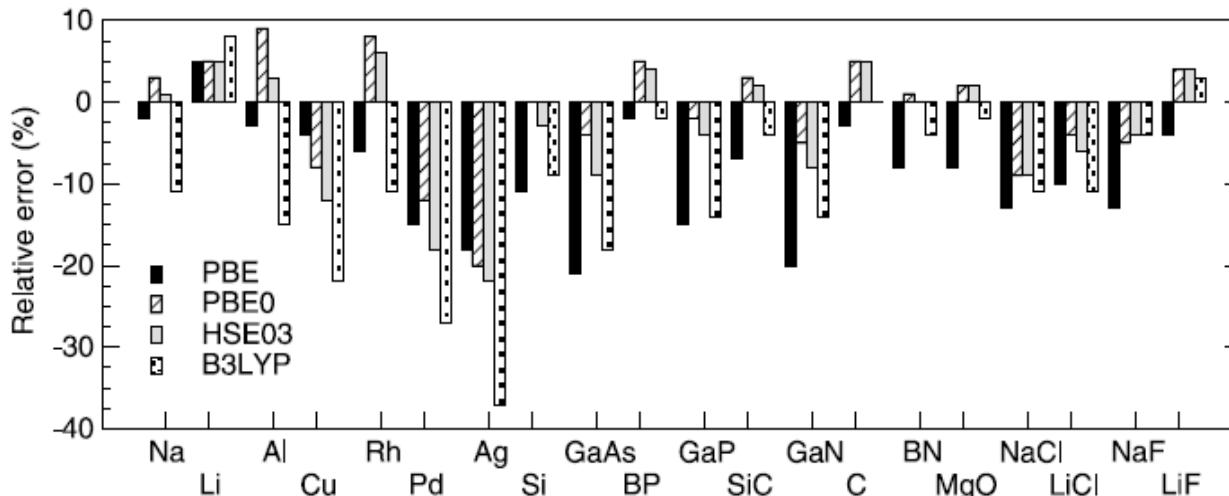
Lattice constants of solids



|           | PBE | PBE0 | HSE | B3LYP |
|-----------|-----|------|-----|-------|
| MRE       | 0.8 | 0.1  | 0.2 | 1.0   |
| MARE      | 1.0 | 0.5  | 0.5 | 1.2   |
| No-metals |     |      |     |       |
| MRE       | 1.0 | 0.0  | 0.1 | 0.8   |
| MARE      | 1.0 | 0.4  | 0.5 | 0.8   |

Bulk modulus of solids

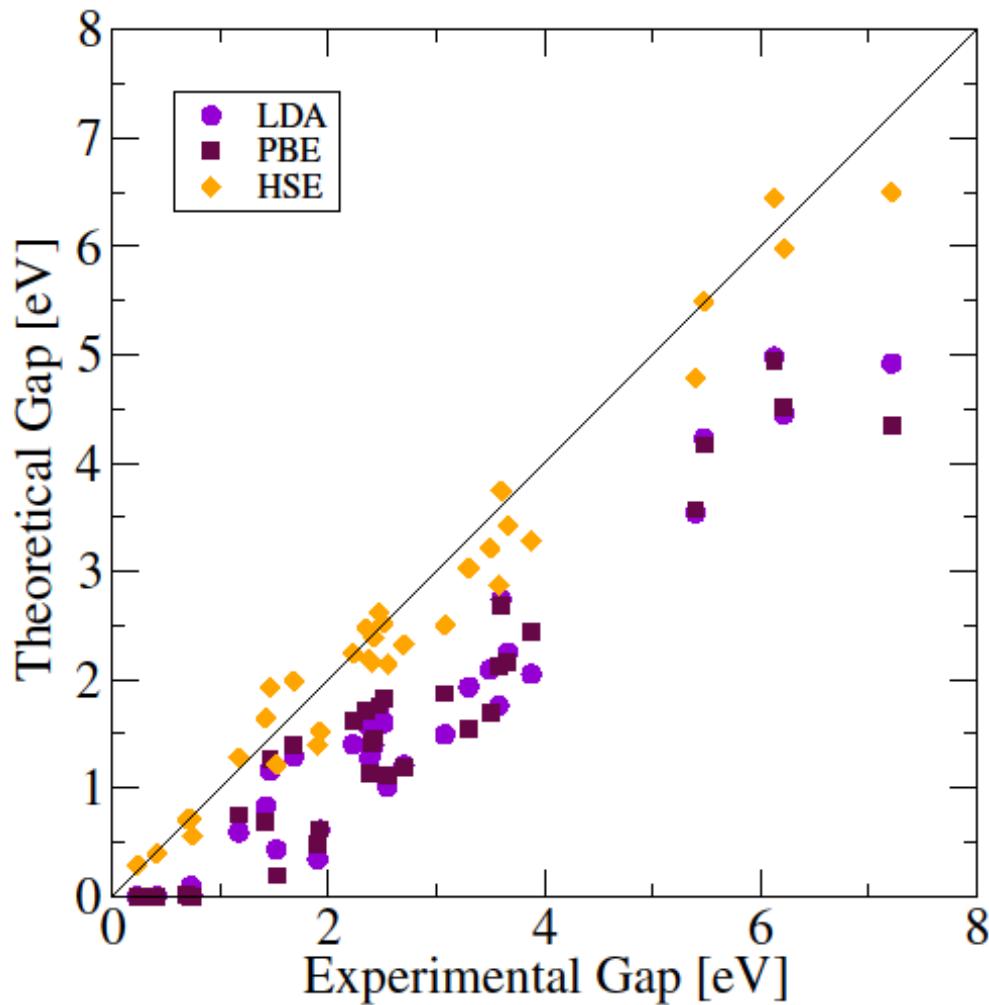
|           | PBE   | PBE0 | HSE  | B3LYP |
|-----------|-------|------|------|-------|
| MRE       | -9.8  | -1.2 | -3.1 | -10.2 |
| MARE      | 9.4   | 5.7  | 6.4  | 11.4  |
| No-metals |       |      |      |       |
| MRE       | -10.4 | -0.7 | -1.8 | -6.8  |
| MARE      | 10.4  | 3.8  | 4.6  | 7.4   |



Marsman et al. J. Phys: Condens. Matter 20, 064201 (2008)

# Hybrids: how do they perform

## Band gaps



Improved band gaps for semiconducting system

HSE correctly predict semiconducting behavior in systems where LDA/GGA predicts a metal

Error (in eV) for 40 simple and binary semiconductors and insulators

|     | LDA   | PBE   | HSE   |
|-----|-------|-------|-------|
| ME  | -1.14 | -1.13 | -0.17 |
| MAE | 1.14  | 1.13  | 0.26  |

# Caveat

1. More expensive than standard DFT (time- and memory-wise)
2. No hybrid PP. Use always a PP generated with the same local functional as the one used in the hybrid functional definition (e.g. use PBE PPs for PBE0 calculations)
3. NLCC not well defined. Avoid PPs with NLCC
4. Might need to explicitly include semi-core states electrons in the valence
5. Calculations with US and PAW PPs typically slower than those with NC PPs
6. Not all the features are implemented for US and PAW PPs
7. Wannier interpolation to get the entire band structure (see hands-on later today)

# Outline

- **Brief recap on KS-DFT.** KS equations, common approximations to the xc energy functional, KS in practice
- **Exercise 0.** Standard DFT: PBE electronic structure of FCC Silicon
- **Exercise 1-2.** Hybrid DFT: electronic structure of FCC Silicon
- **Exercise 3-4.** Meta-GGA: electronic structure of FCC Silicon and BCC Iron using SCAN

# Meta-GGA

$$E_{\text{xc}}[n_{\uparrow}, n_{\downarrow}] = \int d^3r n(\mathbf{r}) \varepsilon_{\text{xc}}(n_{\uparrow}(\mathbf{r}), n_{\downarrow}(\mathbf{r}), \nabla n_{\uparrow}(\mathbf{r}), \nabla n_{\downarrow}(\mathbf{r}), \nabla^2 n_{\uparrow}(\mathbf{r}), \nabla^2 n_{\downarrow}(\mathbf{r}), \tau_{\uparrow}(\mathbf{r}), \tau_{\downarrow}(\mathbf{r}))$$

$$n(\mathbf{r}) = \sum_{\sigma} n_{\sigma}(\mathbf{r}) \quad \text{total charge density}$$

$$n_{\sigma}(\mathbf{r}) = \sum_{v,\mathbf{k}} |\psi_{v,\mathbf{k}}^{\sigma}(\mathbf{r})|^2 \quad \text{spin-charge density} \quad \sigma = \{\uparrow, \downarrow\}$$

$$\tau_{\sigma}(\mathbf{r}) = \frac{1}{2} \sum_{v,\mathbf{k}} |\nabla \psi_{v,\mathbf{k}}^{\sigma}(\mathbf{r})|^2 \quad \text{kinetic energy density}$$

# Plethora of meta-GGA functionals

**BR89**

**r<sup>2</sup>SCAN**

**MVS**

**TPSS**

**rSCAN**

**MGGA\_MS2**

**SCAN**

**TASK**

**revTPSS**

**HLE17**

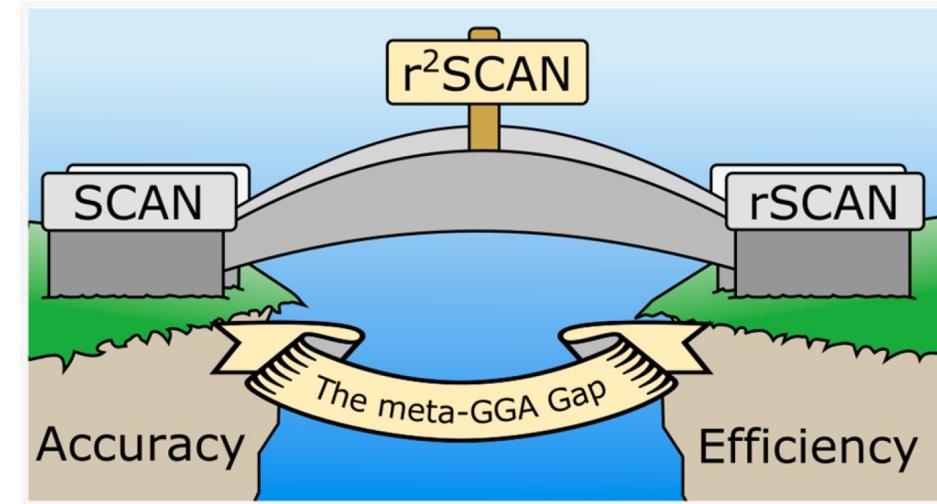
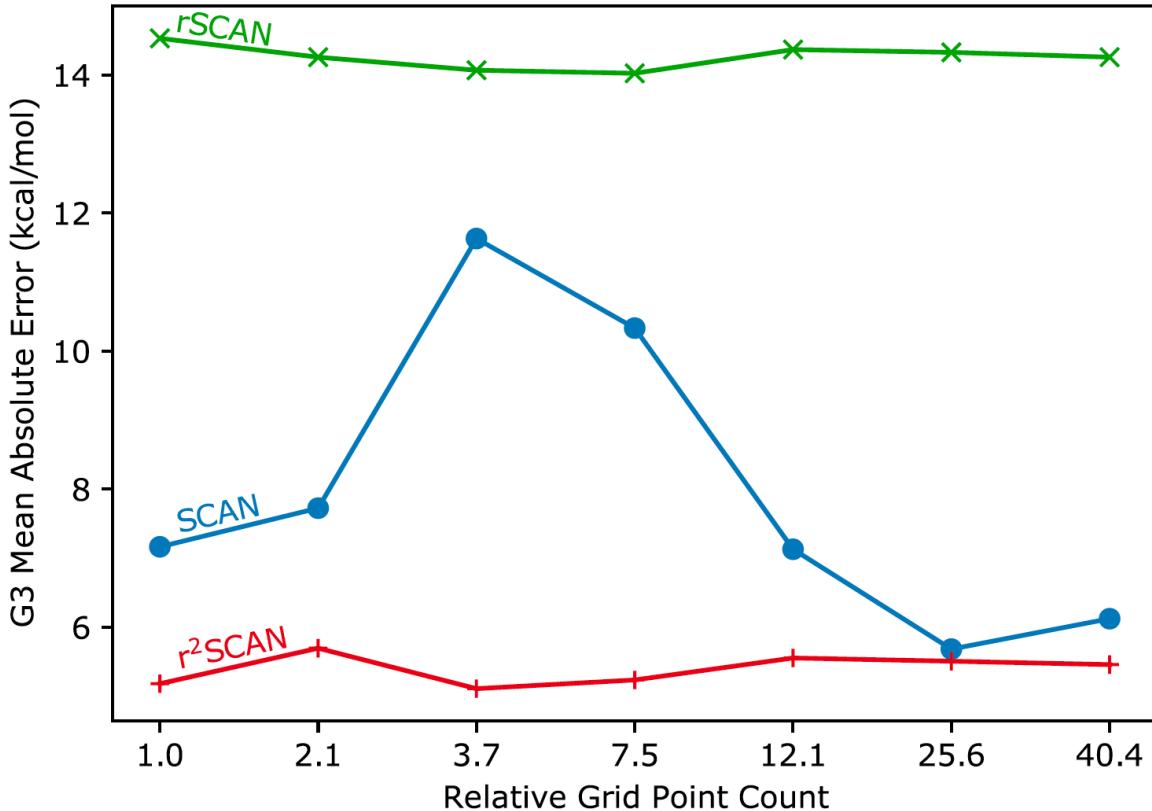
**TM**

**SCAN-L**

# SCAN, rSCAN, r<sup>2</sup>SCAN

$$E_{\text{xc}}[n_{\uparrow}, n_{\downarrow}] = \int d^3r n(\mathbf{r}) \varepsilon_{\text{xc}}(n_{\uparrow}(\mathbf{r}), n_{\downarrow}(\mathbf{r}), \nabla n_{\uparrow}(\mathbf{r}), \nabla n_{\downarrow}(\mathbf{r}), \nabla^2 n_{\uparrow}(\mathbf{r}), \nabla^2 n_{\downarrow}(\mathbf{r}), \tau_{\uparrow}(\mathbf{r}), \tau_{\downarrow}(\mathbf{r}))$$

~~$\nabla^2 n_{\uparrow}(\mathbf{r})$ ,  $\nabla^2 n_{\downarrow}(\mathbf{r})$ ,  $\tau_{\uparrow}(\mathbf{r})$ ,  $\tau_{\downarrow}(\mathbf{r})$~~



SCAN: J. Sun et al., *Phys. Rev. Lett.* **115**, 036402 (2015).

rSCAN: A. Bartok et al., *J. Chem. Phys.* **150**, 161101 (2019).

r<sup>2</sup>SCAN: J. Furness et al., *J. Phys. Chem. Lett.* **11**, 8208 (2020).

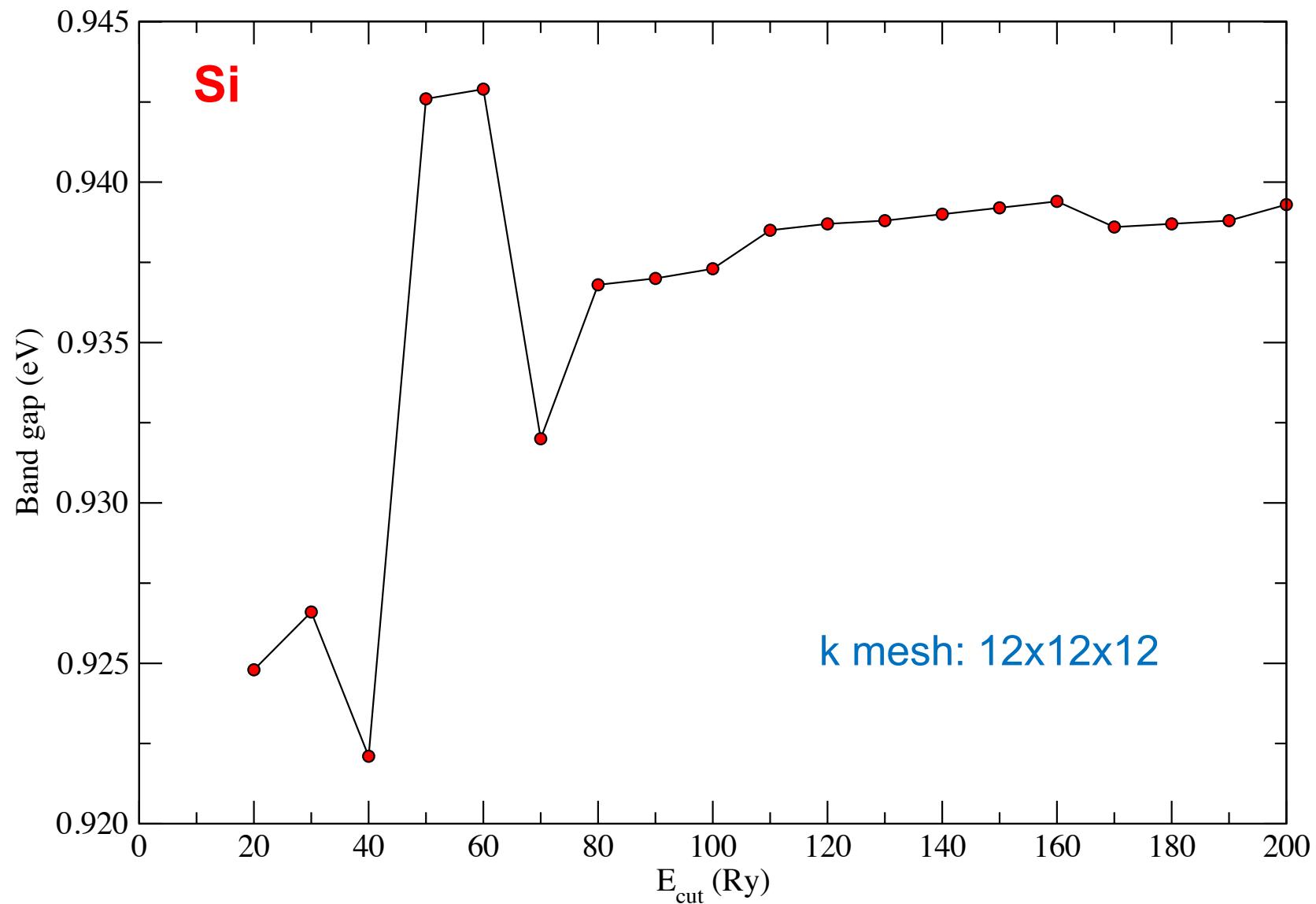
# Outline

- **Brief recap on KS-DFT.** KS equations, common approximations to the xc energy functional, KS in practice
- **Exercise 0.** Standard DFT: PBE electronic structure of FCC Silicon
- **Exercise 1-2.** Hybrid DFT: electronic structure of FCC Silicon
- **Exercise 3-4.** Meta-GGA: electronic structure of FCC Silicon and BCC Iron using SCAN

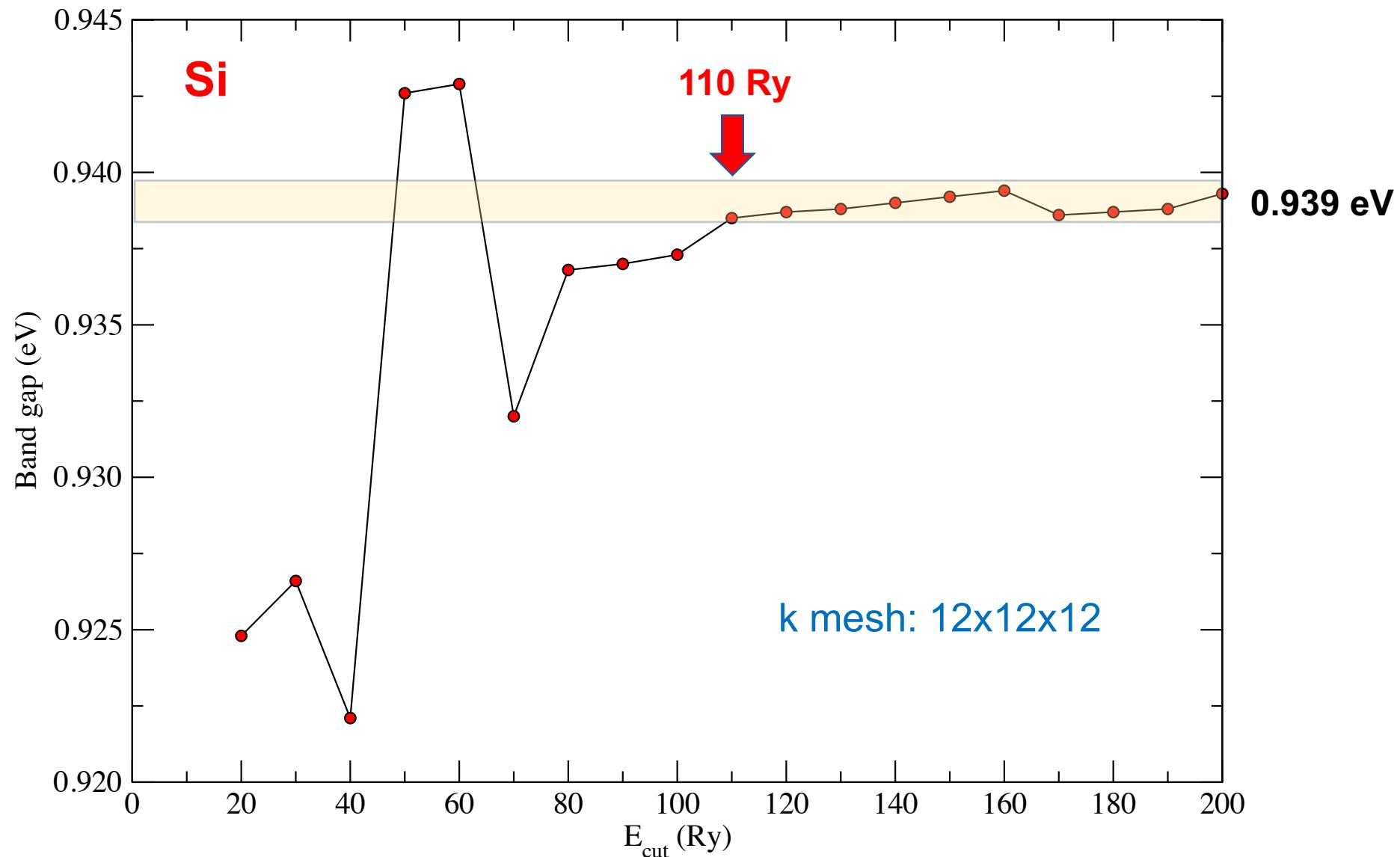
# Input file

```
&control
  calculation='scf'
  restart_mode='from_scratch',
  prefix='Si'
  pseudo_dir = '../..../files/pseudo'
 outdir='./tmp'
  verbosity='high'
/
&system
  ibrav = 2,
  celldm(1) = 10.262, ← Experimental lattice parameter
  nat = 2,
  ntyp = 1,
  ecutwfc = 40.0, ← It is necessary to perform convergence tests w.r.t ecutwfc
  input_dft = 'SCAN' ← Set up the SCAN functional from the input
  nbnd = 5
/
&electrons
  conv_thr = 1.d-10
/
ATOMIC_SPECIES
Si 28.086 Si.SCAN.UPF ← Pseudopotential generated using the SCAN functional
ATOMIC_POSITIONS {alat}
  Si 0.00 0.00 0.00
  Si 0.25 0.25 0.25
K_POINTS {automatic}
  12 12 12 0 0 0 ← It is necessary to perform convergence tests w.r.t k points mesh
```

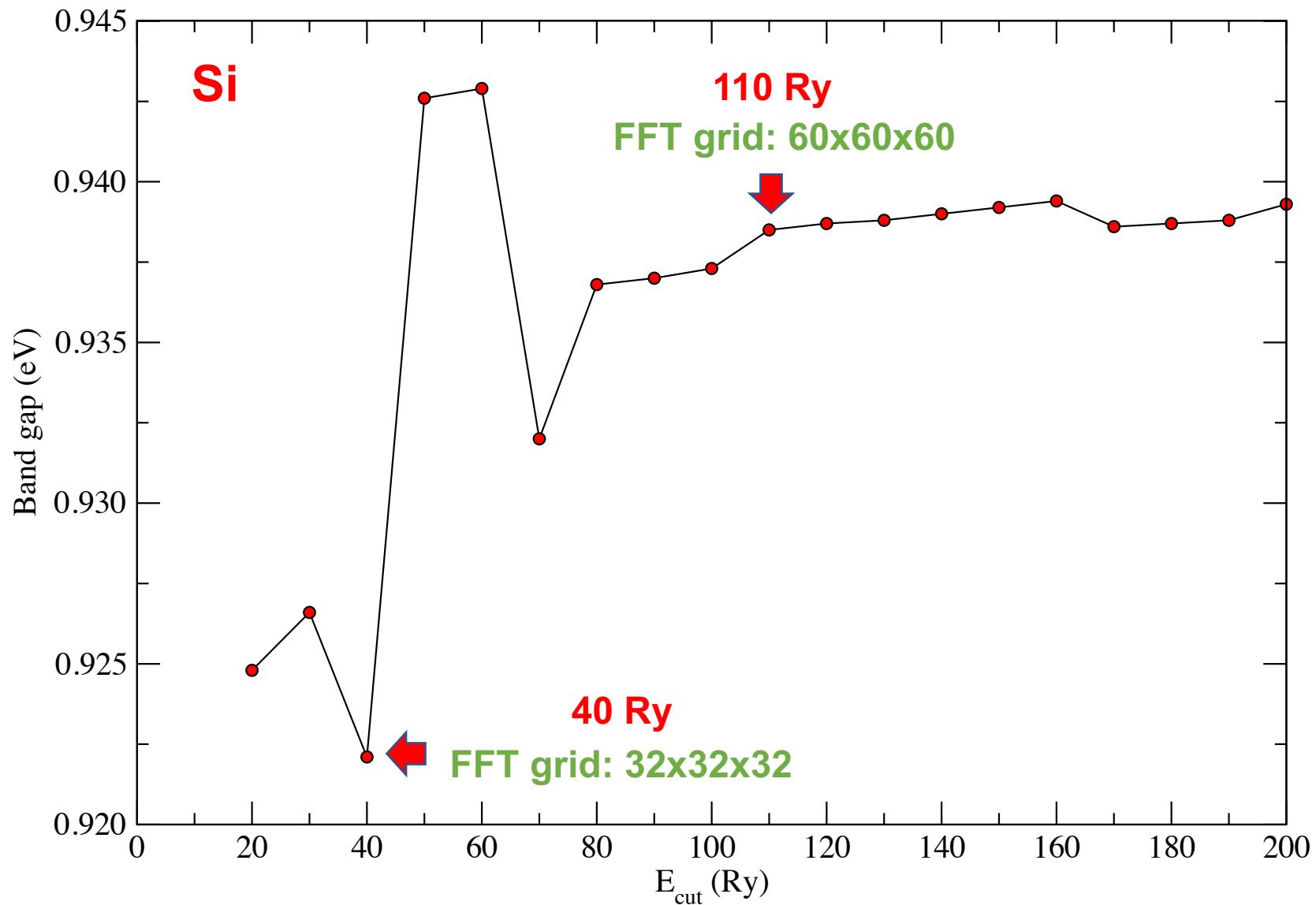
# SCAN functional & SCAN pseudopotential



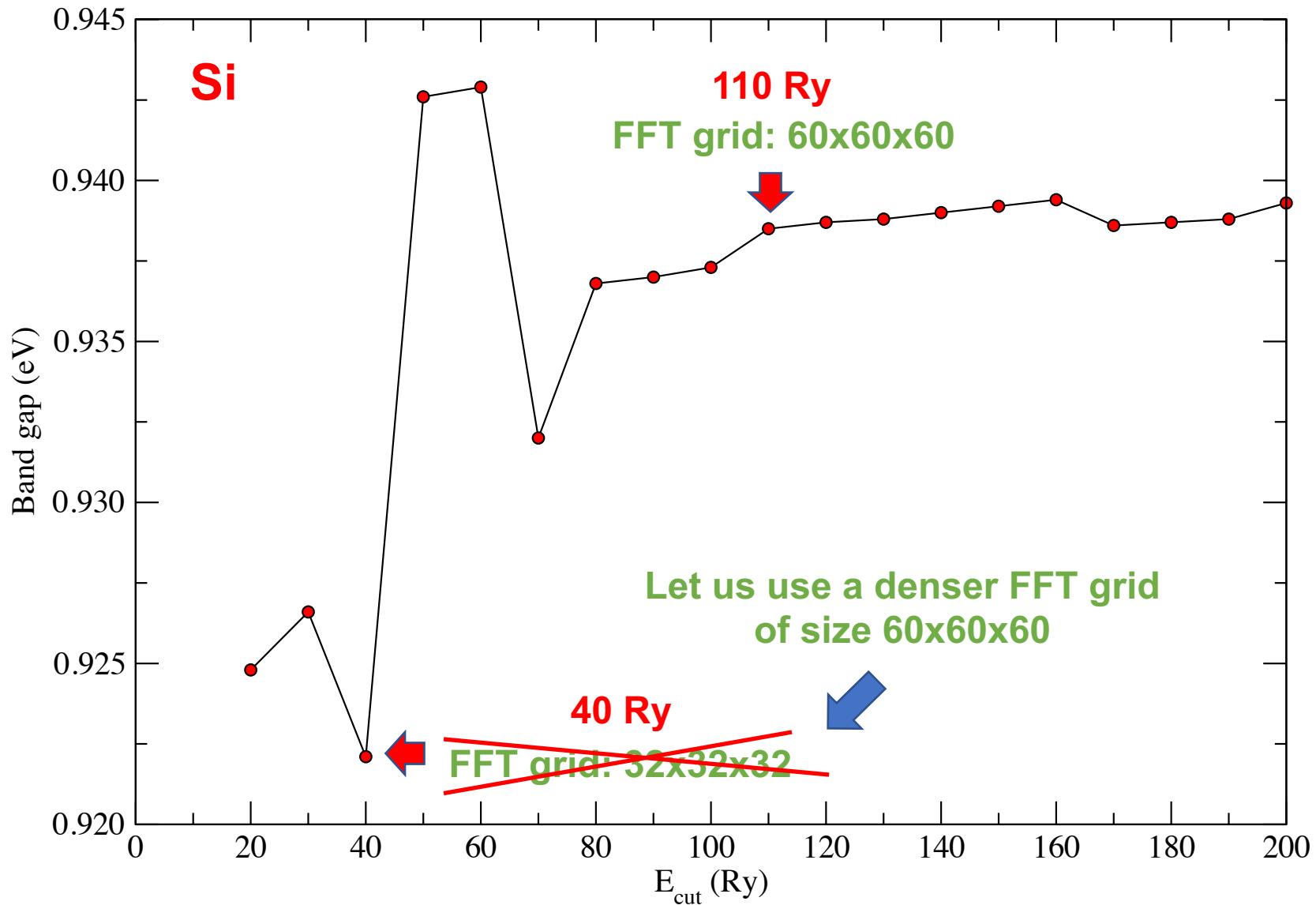
# SCAN functional & SCAN pseudopotential



# SCAN functional & SCAN pseudopotential



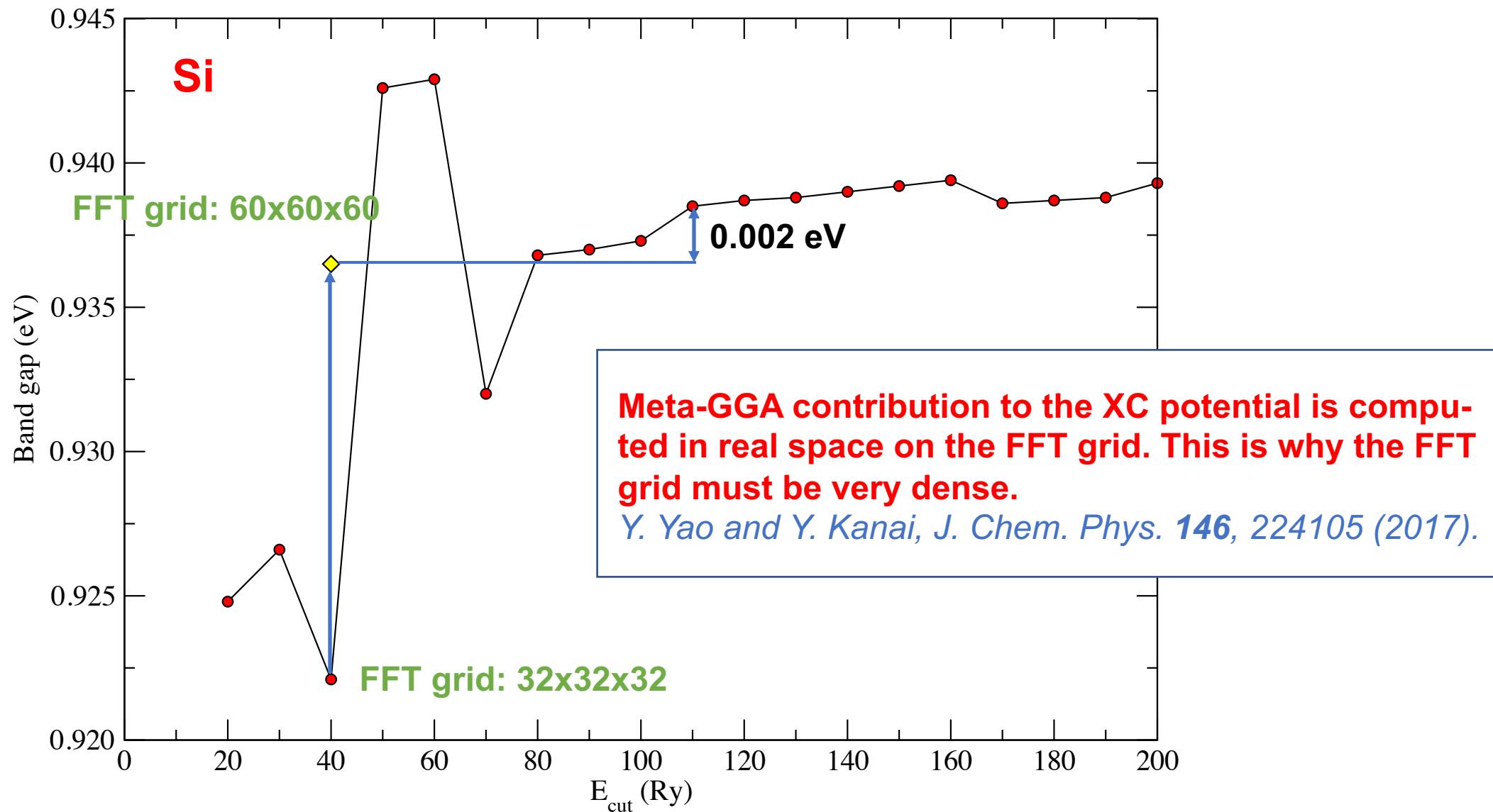
# SCAN functional & SCAN pseudopotential



```
&control
  calculation='scf'
  restart_mode='from_scratch',
  prefix='Si'
  pseudo_dir = '../files/pseudo'
 outdir='./tmp'
  verbosity='high'
/
&system
  ibrav = 2,
  celldm(1) = 10.262,
  nat = 2,
  ntyp = 1,
  ecutwfc = 40.0,
  input_dft = 'SCAN'
  nbnd = 5
  nr1 = 60
  nr2 = 60
  nr3 = 60
/
&electrons
  conv_thr = 1.d-10
/
ATOMIC_SPECIES
Si 28.086 Si.SCAN.UPF
ATOMIC_POSITIONS {alat}
  Si 0.00 0.00 0.00
  Si 0.25 0.25 0.25
K_POINTS {automatic}
  12 12 12 0 0 0
```

FFT grid specified from the input

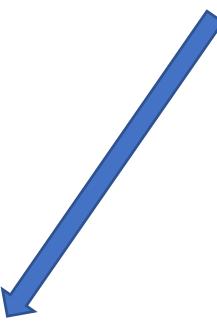
# SCAN functional & SCAN pseudopotential



# SCAN functional & PBE pseudopotential

```
&control
  calculation='scf'
  restart_mode='from_scratch',
  prefix='Si'
  pseudo_dir = '../files/pseudo'
 outdir='./tmp'
  verbosity='high'
/
&system
  ibrav = 2,
  celldm(1) = 10.262,
  nat = 2,
  ntyp = 1,
  ecutwfc = 110.0,
  input_dft = 'SCAN'
  nbnd = 5
/
&electrons
  conv_thr = 1.d-10
/
ATOMIC_SPECIES
Si 28.086 Si.pbe_PseudoDojo.UPF
ATOMIC_POSITIONS {alat}
Si 0.00 0.00 0.00
Si 0.25 0.25 0.25
K_POINTS {automatic}
12 12 12 0 0 0
```

Pseudo Dojo library  
<http://www.pseudo-dojo.org>



```
&control
  calculation='scf'
  restart_mode='from_scratch',
  prefix='Si'
  pseudo_dir = '../files/pseudo'
 outdir='./tmp'
  verbosity='high'
/
&system
  ibrav = 2,
  celldm(1) = 10.262,
  nat = 2,
  ntyp = 1,
  ecutwfc = 110.0,
  input_dft = 'SCAN'
  nbnd = 5
/
&electrons
  conv_thr = 1.d-10
/
ATOMIC_SPECIES
Si 28.086 Si_ONCV_PBE-1.2.upf
ATOMIC_POSITIONS {alat}
Si 0.00 0.00 0.00
Si 0.25 0.25 0.25
K_POINTS {automatic}
12 12 12 0 0 0
```

SG15 library

[http://www.quantum-simulation.org/potentials/sg15\\_oncv/](http://www.quantum-simulation.org/potentials/sg15_oncv/)

# SCAN functional with different pseudopotentials

Ecut = 110 (Ry), k mesh: 12x12x12

| Functional | Pseudopotential | Library          | Gap (eV) |
|------------|-----------------|------------------|----------|
| SCAN       | SCAN            | Yi Yao's library | 0.94     |
| SCAN       | PBE             | SG15 ONCV        | 0.83     |
| SCAN       | PBE             | Pseudo Dojo      | 0.64     |

Gap<sub>expt</sub> = 1.17 (eV)

# SCAN functional with different pseudopotentials

Ecut = 110 (Ry), k mesh: 12x12x12

| Functional | Pseudopotential | Library          | Gap (eV) |              |
|------------|-----------------|------------------|----------|--------------|
| SCAN       | SCAN            | Yi Yao's library | 0.94     | NLCC=.false. |
| SCAN       | PBE             | SG15 ONCV        | 0.83     | NLCC=.false. |
| SCAN       | PBE             | Pseudo Dojo      | 0.64     | NLCC=.true.  |

WARNING →

$$\text{Gap}_{\text{expt}} = 1.17 \text{ (eV)}$$

Currently, nonlinear core correction (NLCC) is not implemented for meta-GGA in Quantum ESPRESSO!

Pseudopotentials that have NLCC=.true. introduce some inconsistency for meta-GGA calculations.

# SCAN functional with different pseudopotentials

Ecut = 110 (Ry), k mesh: 12x12x12

| Functional | Pseudopotential | Library          | Gap (eV) |
|------------|-----------------|------------------|----------|
| SCAN       | SCAN            | Yi Yao's library | 0.94     |
| SCAN       | PBE             | SG15 ONCV        | 0.83     |
| SCAN       | PBE             | Pseudo Dojo      | 0.64     |
| PBE        | PBE             | SG15 ONCV        | 0.56     |
| PBE        | PBE             | Pseudo Dojo      | 0.57     |

Gap<sub>expt</sub> = 1.17 (eV)

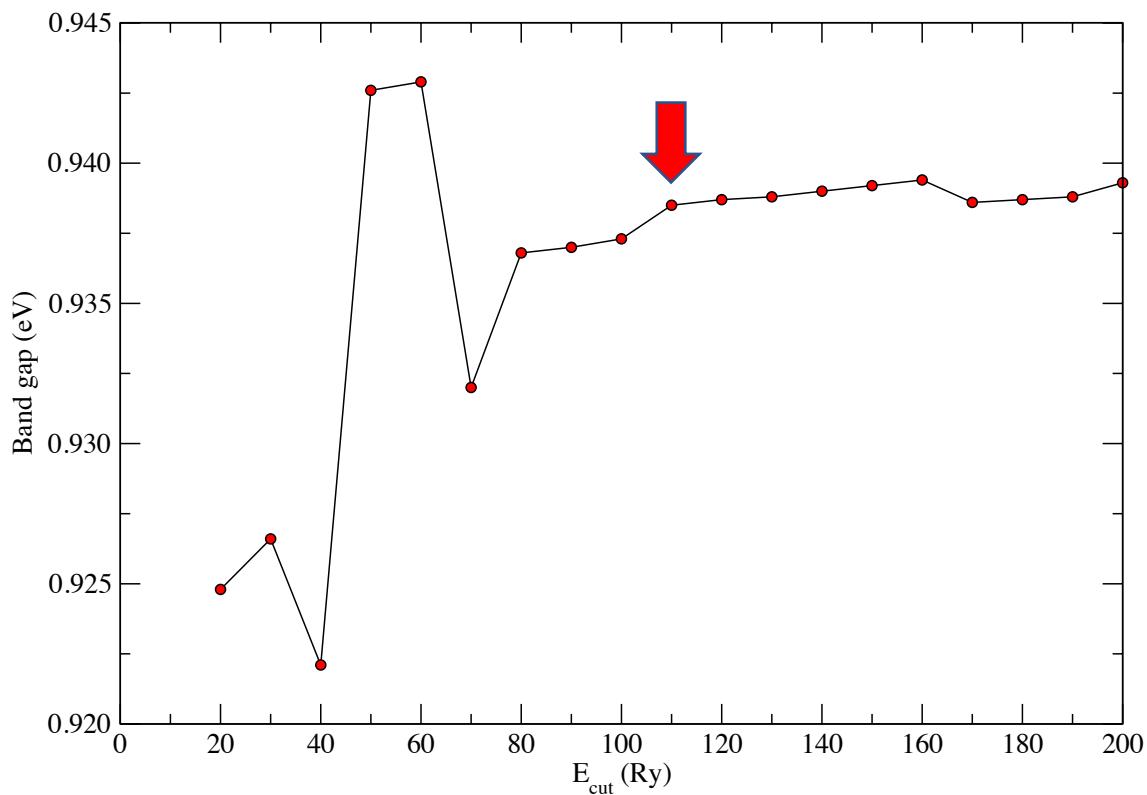
SCAN calculation with the SCAN pseudopotential gives the most accurate results

*Y. Yao and Y. Kanai, J. Chem. Phys. 146, 224105 (2017).*

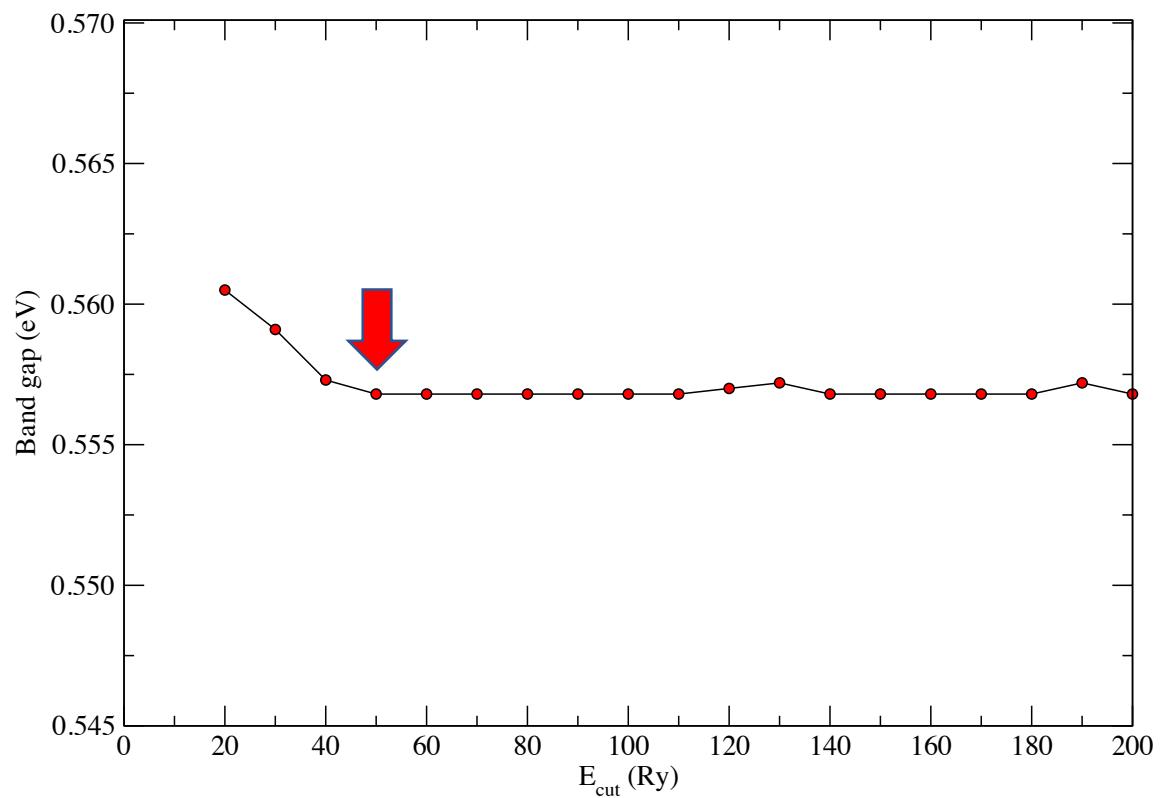
# Convergence w.r.t. Ecut

k mesh: 12x12x12

SCAN with SCAN pseudopotential



PBE with PBE pseudopotential (SG15)



Convergence is achieved much faster with PBE than with SCAN

# Setting up the XC functional

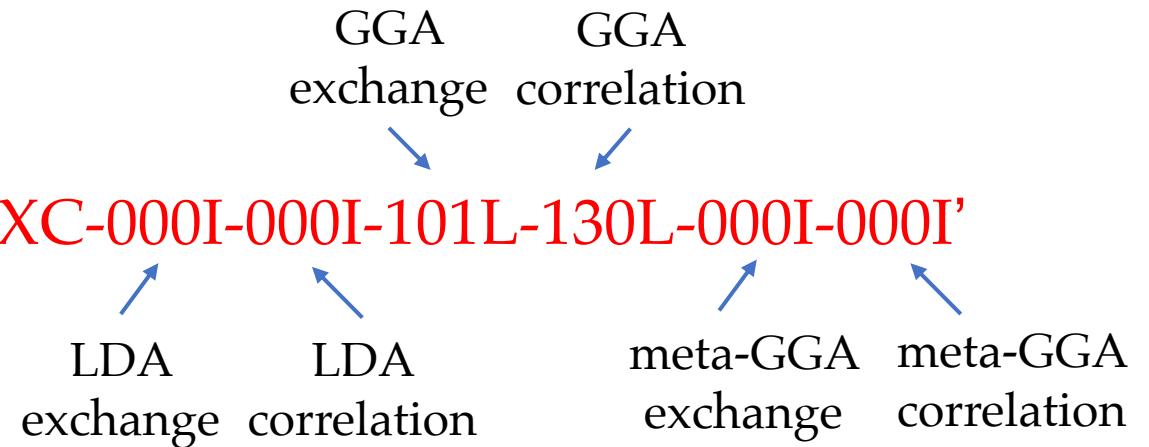
`input_dft = 'PBE' or input_dft='XC-000I-000I-101L-130L-000I-000I'`

Read more about this here: [https://www.quantum-espresso.org/Doc/user\\_guide/node13.html](https://www.quantum-espresso.org/Doc/user_guide/node13.html)

Libxc with IDs: <https://tddft.org/programs/libxc/functionals/libxc-6.0.0/>

# Setting up the XC functional

`input_dft = 'PBE' or input_dft='XC-000I-000I-101L-130L-000I-000I'`

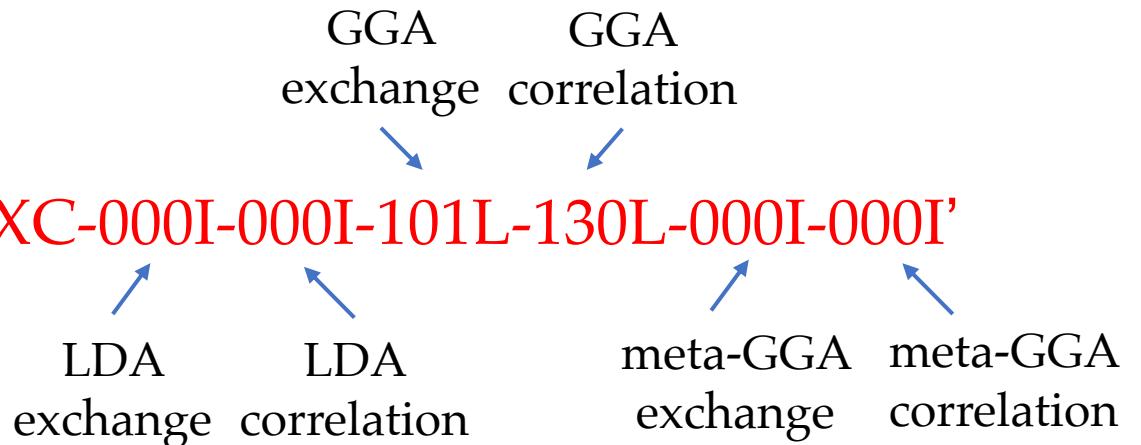


Read more about this here: [https://www.quantum-espresso.org/Doc/user\\_guide/node13.html](https://www.quantum-espresso.org/Doc/user_guide/node13.html)

Libxc with IDs: <https://tddft.org/programs/libxc/functionals/libxc-6.0.0/>

# Setting up the XC functional

`input_dft = 'PBE'` or `input_dft='XC-000I-000I-101L-130L-000I-000I'`



`input_dft = 'SCAN'` or `input_dft='XC-000I-000I-000I-000I-263L-267L'`

no short  
name  
in QE

~~`input_dft = 'rSCAN'`~~ or `input_dft='XC-000I-000I-000I-000I-493L-494L'`

`input_dft = 'r2SCAN'` or `input_dft='XC-000I-000I-000I-000I-497L-498L'`

Read more about this here: [https://www.quantum-espresso.org/Doc/user\\_guide/node13.html](https://www.quantum-espresso.org/Doc/user_guide/node13.html)

Libxc with IDs: <https://tddft.org/programs/libxc/functionals/libxc-6.0.0/>

# Different flavors of SCAN

| Functional          | Pseudopotential | Library   | Gap (eV) |
|---------------------|-----------------|-----------|----------|
| SCAN                | PBE             | SG15 ONCV | 0.83     |
| rSCAN               | PBE             | SG15 ONCV | 0.71     |
| r <sup>2</sup> SCAN | PBE             | SG15 ONCV | 0.70     |

$$\text{Gap}_{\text{expt}} = 1.17 \text{ (eV)}$$

SCAN is more accurate than rSCAN and r<sup>2</sup>SCAN for predicting the band gap of bulk Si  
(at least when using the PBE pseudopotential)

# Outline

- **Brief recap on KS-DFT.** KS equations, common approximations to the xc energy functional, KS in practice
- **Exercise 0.** Standard DFT: PBE electronic structure of FCC Silicon
- **Exercise 1-2.** Hybrid DFT: electronic structure of FCC Silicon
- **Exercise 3-4.** Meta-GGA: electronic structure of FCC Silicon and BCC Iron using SCAN

# Input

```
&control
  calculation='scf'
  restart_mode='from_scratch',
  prefix='Fe'
  pseudo_dir = ' ../../files/pseudo'
 outdir='./tmp'
  verbosity='high'
/
&system
  ibrav = 3,
  celldm(1) = 5.418,                                ← Experimental lattice parameter
  nat = 1,
  ntyp = 1,
  ecutwfc = 50.0,                                     ← It is necessary to perform convergence tests w.r.t ecutwfc
  occupations = 'smearing',
  smearing ='mv',
  degauss = 0.02,
  nspin = 2,
  starting_magnetization(1) = 0.5,
  input_dft = 'SCAN'                                    ← Set up the SCAN functional from the input
/
&electrons
  conv_thr = 1.d-10
/
ATOMIC_SPECIES
Fe 55.845 Fe_ONCV_PBE-1.2.upf                  ← Pseudopotential generated using the PBE functional
ATOMIC_POSITIONS {crystal}
Fe 0.000000 0.000000 0.000000
K_POINTS {automatic}
20 20 20 0 0 0
```

Experimental lattice parameter

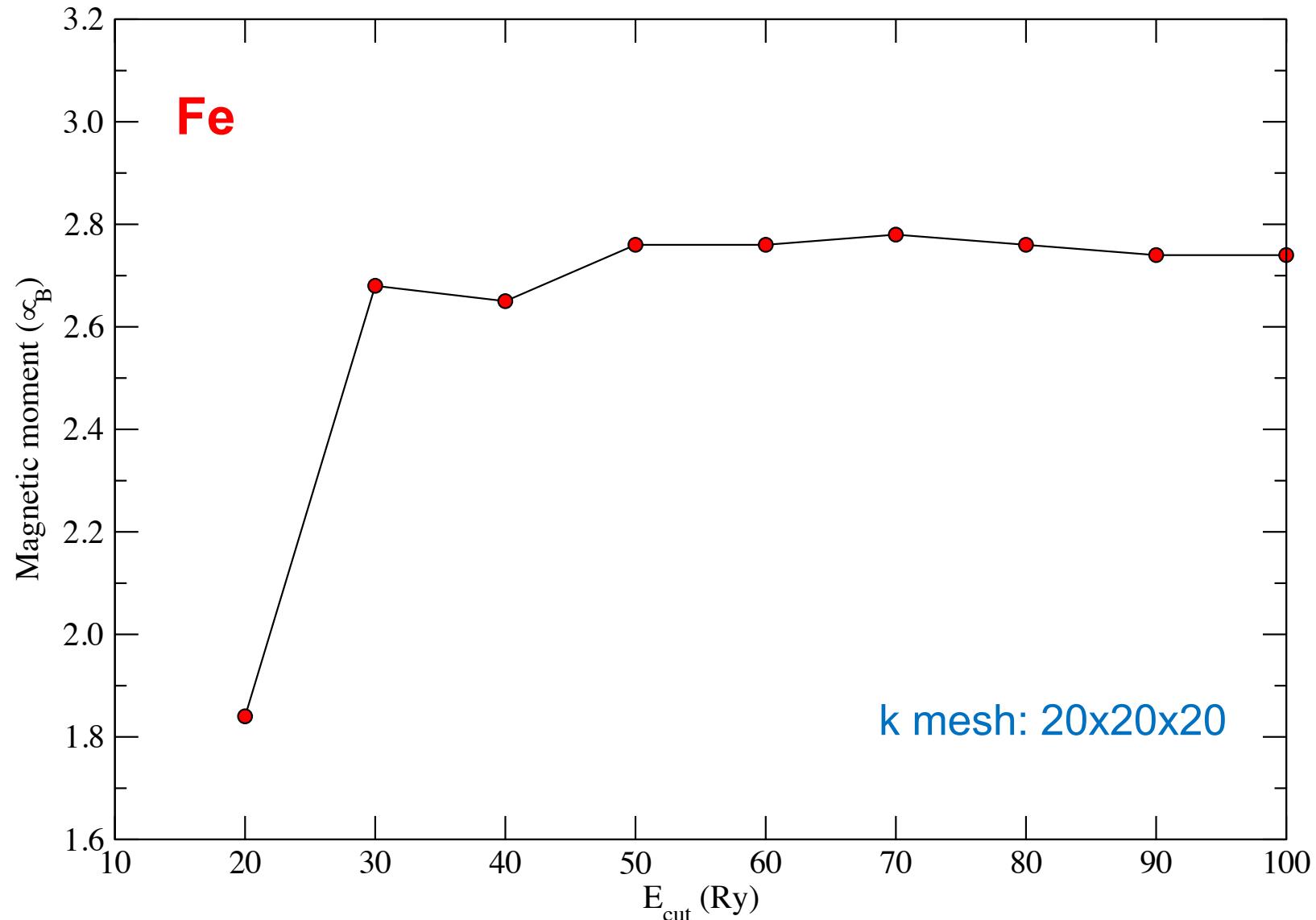
It is necessary to perform convergence tests w.r.t ecutwfc

Set up the SCAN functional from the input  
Quantum ESPRESSO must be compiled with Libxc

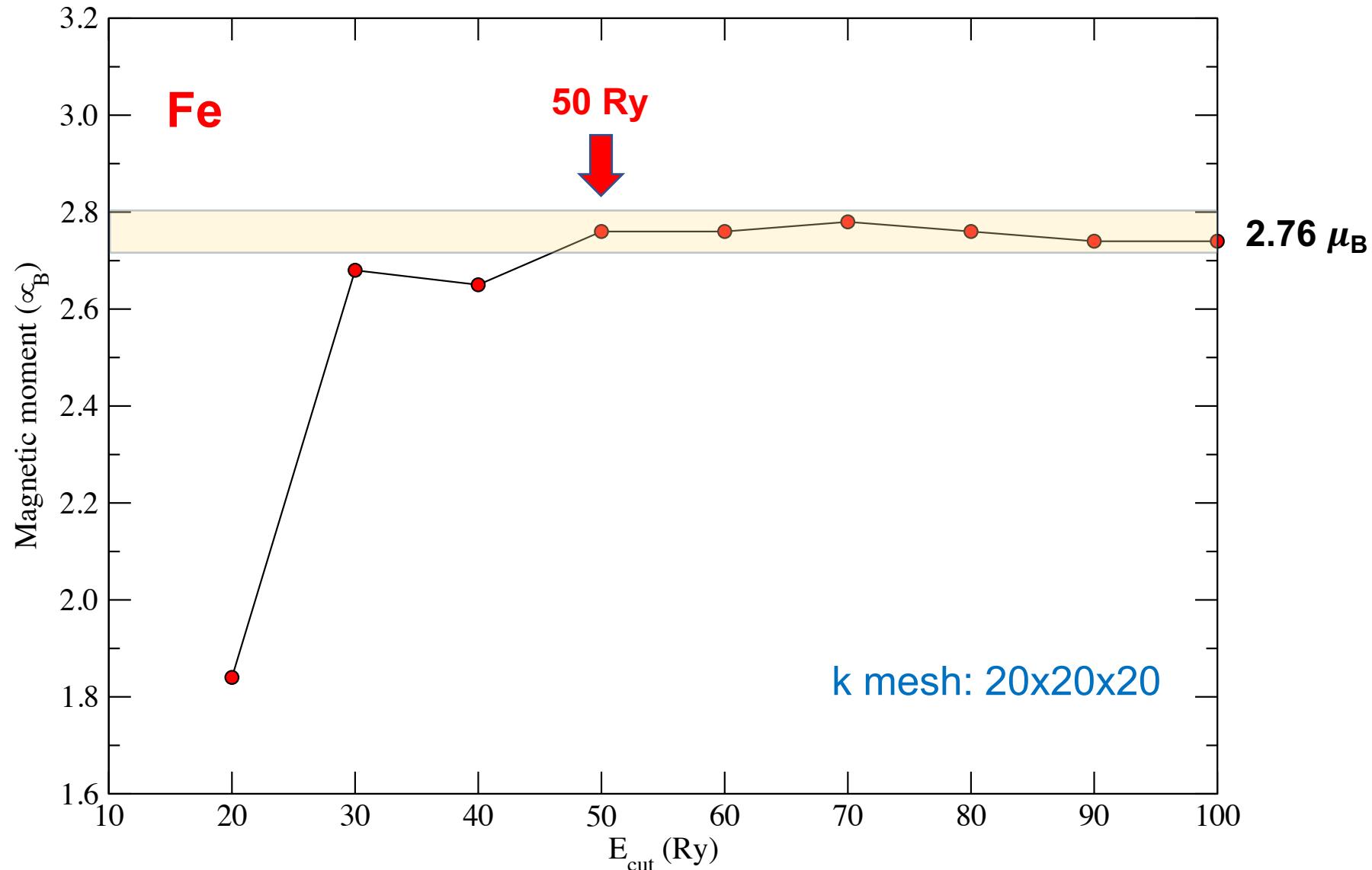
Pseudopotential generated using the PBE functional  
SG15 ONCV library (there is no SCAN pseudo for Fe)

It is necessary to perform convergence tests w.r.t k points mesh

# SCAN functional & PBE pseudopotential (SG15)



# SCAN functional & PBE pseudopotential (SG15)



# SCAN functional with different pseudopotentials

Ecut = 50 (Ry), k mesh: 20x20x20

WARNING →

| Functional | Pseudopotential | Library     | m ( $\mu_B$ ) |              |
|------------|-----------------|-------------|---------------|--------------|
| SCAN       | PBE             | SG15 ONCV   | 2.76          | NLCC=.false. |
| SCAN       | PBE             | Pseudo Dojo | 2.67          | NLCC=.true.  |
| PBE        | PBE             | SG15 ONCV   | 2.28          |              |
| PBE        | PBE             | Pseudo Dojo | 2.26          |              |

$$m_{\text{expt}} = 1.98 - 2.13 (\mu_B)$$

SCAN overestimates magnetic moments in itinerant ferromagnets

PBE is in closer agreement with experiments than SCAN for magnetic moments of itinerant ferromagnets

# Different flavors of SCAN

| Functional | Pseudopotential | Library   | $m (\mu_B)$ |
|------------|-----------------|-----------|-------------|
| SCAN       | PBE             | SG15 ONCV | 2.76        |
| rSCAN      | PBE             | SG15 ONCV | 2.73        |
| $r^2$ SCAN | PBE             | SG15 ONCV | 2.73        |

$$m_{\text{expt}} = 1.98 - 2.13 (\mu_B)$$

Different flavors of SCAN give very similar magnetic moments for bulk Fe



UNIVERSITÀ  
DEGLI STUDI  
DI TRIESTE

EPFL

# Introduction to the hands-on tutorial: **Wannier interpolation of band structures**

**Antimo Marrazzo<sup>1</sup>,**  
**Giovanni Pizzi<sup>2</sup> and Junfeng Qiao<sup>2</sup>**

**<sup>1</sup>Physics Department, University of Trieste, Italy**

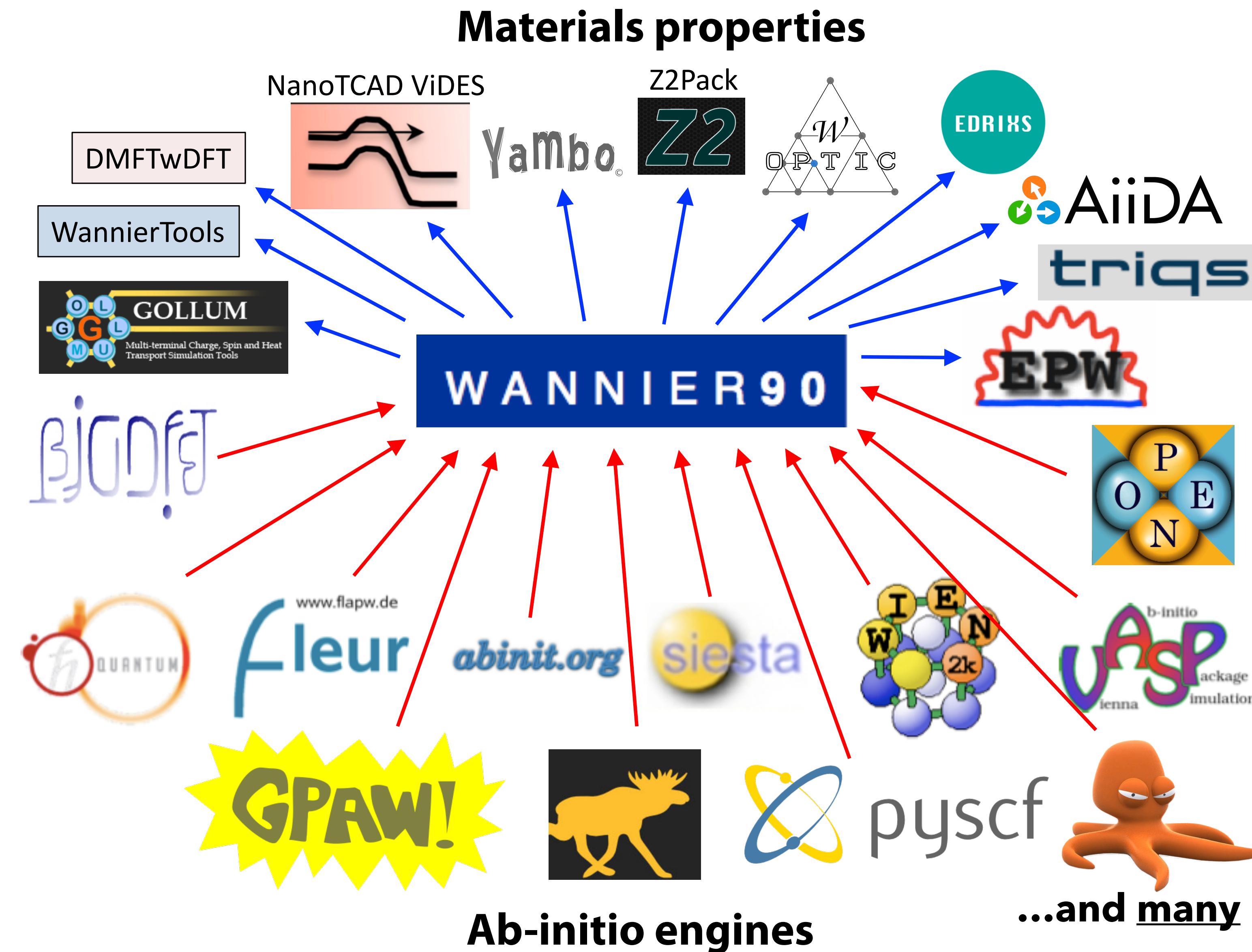
**<sup>2</sup>Theory and Simulation of Materials (THEOS), EPFL, Switzerland**

National Centre for Computational Design and Discovery of Novel Materials (MARVEL), Switzerland

9 November 2022

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

# Wannier90: The center of a software ecosystem

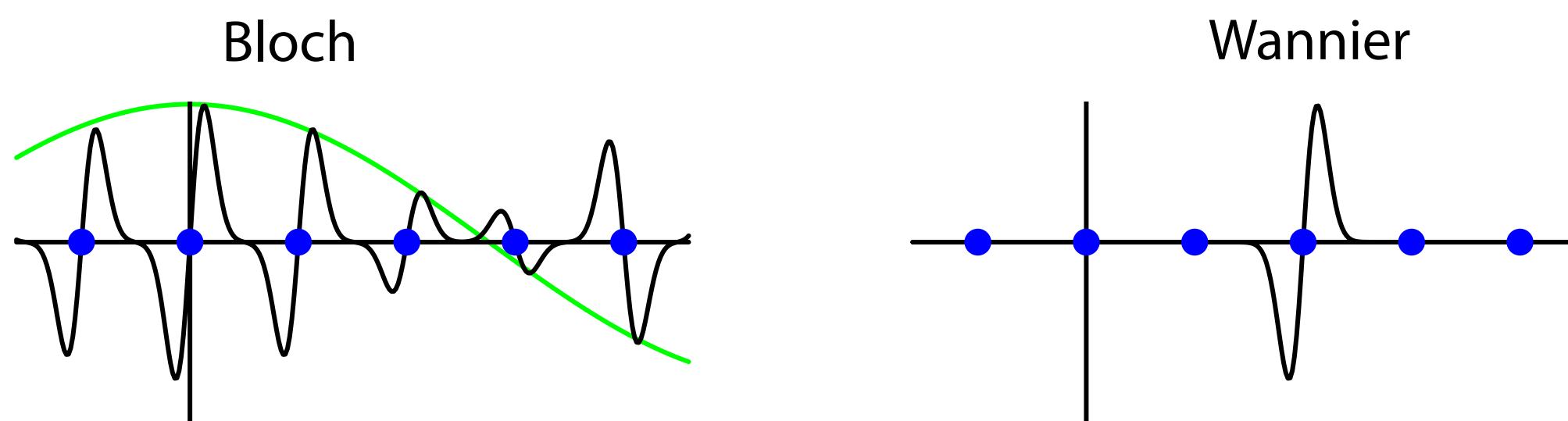


# Maximally-localised Wannier functions (MLWF) in a nutshell

Electronic ground states  
from first-principles simulations → Subspace-selection & unitary  
(e.g. Quantum ESPRESSO) transformation of Bloch wave  
functions → MLWF in real space

$$|w_{n\mathbf{R}}\rangle = V \int_{BZ} \frac{d\mathbf{k}}{(2\pi)^3} e^{-i\mathbf{k}\cdot\mathbf{R}} \sum_{m=1}^J |\psi_{m\mathbf{k}}\rangle U_{mn\mathbf{k}},$$

Choose  $U_{mn\mathbf{k}}$  such that minimise  $\Omega = \sum_{n=1}^J \left[ \langle w_{n\mathbf{0}} | \mathbf{r} \cdot \mathbf{r} | w_{n\mathbf{0}} \rangle - |\langle w_{n\mathbf{0}} | \mathbf{r} | w_{n\mathbf{0}} \rangle|^2 \right].$



# Maximally-localised Wannier functions (MLWF) in a nutshell

Gauge-invariant part of the spread functional

$$\Omega_I = \frac{1}{N} \sum_{\mathbf{k}, \mathbf{b}} w_b \left( J - \sum_{mn} |M_{mn}^{(\mathbf{k}, \mathbf{b})}|^2 \right)$$

Gauge-dependent part of the spread functional

$$\begin{aligned} \tilde{\Omega} = & \frac{1}{N} \sum_{\mathbf{k}, \mathbf{b}} w_b \sum_{m \neq n} |M_{mn}^{(\mathbf{k}, \mathbf{b})}|^2 \\ & + \frac{1}{N} \sum_{\mathbf{k} \mathbf{b}} w_b \sum_n (-\text{Im} \ln M_{nn}^{(\mathbf{k}, \mathbf{b})} - \mathbf{b} \cdot \bar{\mathbf{r}}_n)^2. \end{aligned}$$

Follow the gradient and update the unitary  
matrices  $\mathbf{U}_{\mathbf{k}}$

minimise!

$$M_{mn}^{(0)(\mathbf{k}, \mathbf{b})} = \langle u_{m\mathbf{k}}^{(0)} | u_{n, \mathbf{k}+\mathbf{b}}^{(0)} \rangle$$

$$M^{(\mathbf{k}, \mathbf{b})} = U^{(\mathbf{k})\dagger} M^{(0)(\mathbf{k}, \mathbf{b})} U^{(\mathbf{k}+\mathbf{b})}$$

In principle the overlap matrices  $M_{mn\mathbf{k}}$  are sufficient, in practice one needs (critical for disentanglement) to specify an **initial subspace** and an **initial gauge**.

$$|\phi_{n\mathbf{k}}\rangle = \sum_{m=1}^{J \text{ or } \mathcal{J}_{\mathbf{k}}} |\psi_{m\mathbf{k}}\rangle \langle \psi_{m\mathbf{k}}| g_n \rangle$$

↓

$$A_{mn\mathbf{k}} = \langle \psi_{m\mathbf{k}} | g_n \rangle$$

localised projection functions  
(e.g. s, p, d, sp3, sp2, sp3d2)

$A_{mn\mathbf{k}}$  directly from SCDM  
(not today)

$$S_{mn\mathbf{k}} = \langle \phi_{m\mathbf{k}} | \phi_{n\mathbf{k}} \rangle = (A_{\mathbf{k}}^\dagger A_{\mathbf{k}})_{mn},$$

$$|\tilde{\psi}_{n\mathbf{k}}\rangle = \sum_{m=1}^J |\phi_{m\mathbf{k}}\rangle S_{mn\mathbf{k}}^{-\frac{1}{2}}$$

$$= \sum_{m=1}^{J \text{ or } \mathcal{J}_{\mathbf{k}}} |\psi_{m\mathbf{k}}\rangle (A_{\mathbf{k}} S_{\mathbf{k}}^{-\frac{1}{2}})_{mn},$$

WANNIER90

Features   Download   Support   Papers   News   Events   People   History

Welcome! This is the home of maximally-localised Wannier functions (MLWFs) and Wannier90, the computer program that calculates them.

[FIND OUT MORE](#)

LATEST NEWS

**Wannier 2022 Summer School (and Developers Meeting)**  
January 28, 2022  
The “Wannier 2022 Summer School” will be held at ICTP (Trieste, Italy) from 16 to 20 May 2022 (and “Wannier 2022 ... [Continue reading](#)

**Videos of the “Virtual Edition” Wannier90 School now online**  
April 1, 2020  
The video recordings of the first “Virtual Edition” of the Wannier90 School are now all available in the Learn section ... [Continue reading](#)

**School on Wannier90 v3.0: new features and applications, 25-27th March 2020 – Virtual Edition**  
March 13, 2020

PLEASE CITE

*Wannier90 as a community code: new features and applications*, G. Pizzi et al., J. Phys. Cond. Matt. 32, 165902 (2020) [[ONLINE JOURNAL](#), [OPEN ACCESS](#)] [[bibTeX](#)]  
in all publications resulting from your use of Wannier90.

If you are using v2.x, please cite instead:  
*An updated version of wannier90: A tool for obtaining maximally-localised Wannier functions*, AA Mostofi, JR Yates, G Pizzi, YS Lee, I Souza, D Vanderbilt, N Marzari, Comput. Phys. Commun. 185, 2309 (2014) [[ONLINE JOURNAL](#)] [[bibTeX](#)]

If you are using v1.x, please cite instead:  
*wannier90: A tool for obtaining maximally-localised Wannier functions*, AA Mostofi, JR

WANNIER90

Features Download Support

Papers News Events People History

## Download

Latest stable release (5 March 2020): Wannier90 (v3.1.0) [[gzipped-tar](#)]

Please note that:

- Wannier90 is released under the [GNU General Public License \(v2\)](#)
- A summary of improvements may be found in [CHANGE.log](#)
- Installation instructions may be found in [README.install](#)
- The latest User Guide and Tutorial may be found [here](#). They may also be found in the current distribution.

<https://github.com/wannier-developers/wannier90>

## For developers (GitHub)

The development of Wannier90 is managed on the [Wannier developers GitHub site](#) where on-going developments, and [how to contribute to Wannier90](#).

School on Wannier90 v3.0: new features and applications, 25-26 March 2020 – Virtual Edition

March 13, 2020  
Due to ongoing concerns related to the coronavirus pandemic, we have taken the decision to cancel the physical meeting of ... [Continue reading](#)

Wannier90 (v3.1.0) released

March 5, 2020  
Wannier90 (v3.1.0) has been released today!

ALL NEWS

The screenshot shows the GitHub repository page for 'wannier-developers / wannier90'. The repository has 3 branches and 9 tags. The 'Code' tab is selected, showing a list of commits from various authors. The 'About' section indicates it's the official repository of the Wannier90 code, linking to www.wannier.org. The 'Releases' section shows 9 tags, with a link to create a new release. The 'Contributors' section lists 27 contributors with their profile pictures.

The screenshot shows the Wannier90 website with a red circle highlighting the "Support" menu item in the top navigation bar. The "Support" page contains sections for User Guide, Tutorial, and Source Code Documentation, along with links to PDF files and video recordings of a virtual edition.

**Support**

## User Guide, Tutorial and Source Code Documentation

The Wannier90 user guide and tutorial are both available in the 'doc' directory of the [current distribution](#). They are also available for direct download here:

- User guide v3.1.0: [\[PDF\]](#)
- Tutorial v3.1.0: [\[PDF\]](#)
- Tutorial solutions v3.1.0: [\[PDF\]](#)

**Videos of the “Virtual Edition” Wannier90 School**  
April 1, 2020  
The video recordings of the first “Virtual Edition” of the Wannier90 School are now available in the Learn section ... [Continue reading](#)

**School on Wannier90 v3.0: new features and applications**  
March 2020 – Virtual Edition  
March 13, 2020  
Due to ongoing concerns related to the coronavirus pandemic, the organizers have decided to cancel the physical meeting of the school ... [Continue reading](#)

**Wannier90 (v3.1.0) released**  
March 5, 2020  
Wannier90 (v3.1.0) has been released today!

**ALL NEWS**

If `bands_plot = true`, then the code will calculate the band structure, through Wannier interpolation, along the path in k-space defined by `bands_kpath` using `bands_num_points` along the first section of the path and write out an output file in a format specified by `bands_plot_format`. The default value is `false`.

### 2.9.11 kpoint\_path

Defines the path in k-space along which to calculate the bandstructure. Each line gives the start and end point (with labels) for a section of the path. Values are in fractional coordinates with respect to the primitive reciprocal lattice vectors.

```
begin kpoint_path
    G 0.0 0.0 0.0 L 0.0 0.0 1.0
    L 0.0 0.0 1.0 N 0.0 1.0 1.0
    :
end kpoint_path
```

There is no default

### 2.9.12 integer :: bands\_num\_points

wannier90: A tool for obtaining maximally-localised Wannier functions, AA Mostofi, JR Yates, YS Lee, I Souza, D Vanderbilt, N Marzari, *Comput. Phys. Commun.* **178**, 685 (2008) [[ONLINE JOURNAL](#)] [[OPEN ACCESS](#)] [[bibTeX](#)]

The Wannier90 user guide and tutorial are both available in the 'doc' directory of the [current distribution](#). They are also available for direct download here:

- User guide v3.1.0: [\[PDF\]](#)
- Tutorial v3.1.0: [\[PDF\]](#)
- Tutorial solutions v3.1.0: [\[PDF\]](#)

**8: Iron — Spin-polarized WFs, DOS, projected WFs versus MLWFs**

- Outline: Generate both maximally-localized and projected Wannier functions for ferromagnetic bcc Fe. Calculate the total and orbital-projected density of states by Wannier interpolation.

**Figure 21: Unit cell of Iron crystal plotted with the XCRYSDEN program.**

| Final State                       |             |                                     |            |
|-----------------------------------|-------------|-------------------------------------|------------|
| WF centre and spread              | 1           | ( 0.709852, 0.000108, 0.000131 )    | 1.08935224 |
| WF centre and spread              | 2           | ( 0.000131, 0.000053, -0.709852 )   | 1.08935218 |
| WF centre and spread              | 3           | ( -0.709852, -0.000108, -0.000131 ) | 1.08935221 |
| WF centre and spread              | 4           | ( 0.000108, -0.709852, -0.000053 )  | 1.08935218 |
| WF centre and spread              | 5           | ( -0.000131, -0.000053, 0.709852 )  | 1.08935226 |
| WF centre and spread              | 6           | ( 0.000000, 0.000000, 0.000000 )    | 0.43234428 |
| WF centre and spread              | 7           | ( -0.000000, 0.000000, 0.000000 )   | 0.43234429 |
| WF centre and spread              | 8           | ( -0.000108, 0.709852, 0.000053 )   | 1.08935225 |
| WF centre and spread              | 9           | ( 0.000000, 0.000000, -0.000000 )   | 0.43234428 |
| Sum of centres and spreads        |             | ( 0.000000, -0.000000, -0.000000 )  | 7.83314616 |
| Spreads (Ang <sup>-2</sup> )      | Omega I     | = 5.948424630                       |            |
| =====                             | Omega D     | = 0.017027691                       |            |
|                                   | Omega OD    | = 1.867693841                       |            |
| Final Spread (Ang <sup>-2</sup> ) | Omega Total | = 7.833146162                       |            |

If you are using v1.x, please refer to the [wannier90: A tool for obtaining maximally-localized Wannier functions](#) paper by Yates, YS Lee, I Souza, D Vanderbilt, and J. P. Perdew (2008) [[ONLINE JOURNAL](#)] [[PDF](#)].

and for spin-down MLWFs is

The screenshot shows the official website for Wannier90. At the top, there is a dark blue header bar with the text "WANNIER90". Below this is a light gray navigation bar with several links: "Features", "Download", "Support" (which is circled in red), "Papers", "News", "Events", "People", and "History". The main content area has a white background. It features a section titled "Community Email Forum". Inside this section, there is text about a mailing list, with a link to subscribe highlighted by a red oval. Another red oval highlights the email address for posting. A note below specifies that registration is required to post. Further down, it mentions access to mailing list archives. At the bottom of the page, there is a news summary about a decision to cancel a meeting, a link to the full article in a journal, and a news item about the release of version 3.1.0 of Wannier90.

WANNIER90

Features Download Support Papers News Events People History

## Community Email Forum

A fully archived Wannier90 user mailing list is hosted by Quantum ESPRESSO: [SUBSCRIBE HERE](#), but at the same time please send an email to [nicola.marzari@epfl.ch](mailto:nicola.marzari@epfl.ch) to confirm your interest (sometimes people are enrolled unwittingly by spambots).

Once subscribed, post to the list by sending your email to [wannier@lists.quantum-espresso.org](mailto:wannier@lists.quantum-espresso.org). We kindly request that you include your name and affiliation in all posts to the mailing list.

**Note that you must register in order to post emails to this list. Emails from non-registered users will be deleted automatically.**

The archives of the Wannier90 mailing list may be accessed [here](#); alternatively, a searchable version of the mailing list archive can be accessed [here](#).

decision to cancel the physical meeting or ... [Continue reading](#)

*Comput. Phys. Commun.* **185**, 2309 (2014) [[ONLINE JOURNAL](#)] [[bibTeX](#)]

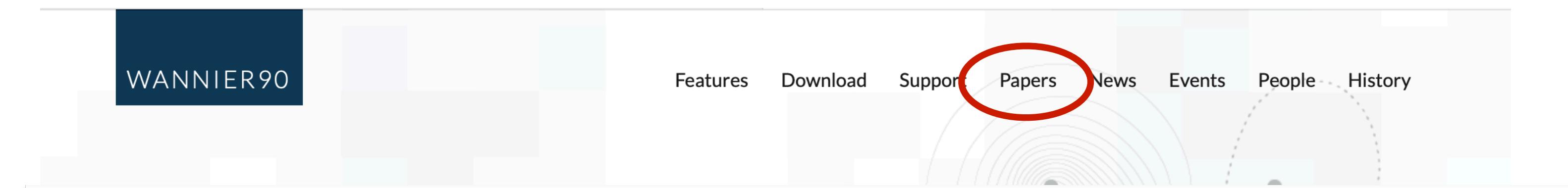
**Wannier90 (v3.1.0) released**

March 5, 2020

Wannier90 (v3.1.0) has been released today!

If you are using v1.x, please cite instead:  
wannier90: A tool for obtaining maximally-localised Wannier functions, AA Mostofi, JR Yates, YS Lee, I Souza, D Vanderbilt, N Marzari, *Comput. Phys. Commun.* **178**, 685 (2008) [[ONLINE JOURNAL](#)] [[OPEN ACCESS](#)] [[bibTeX](#)]

[ALL NEWS](#)



## In publications arising from the use of Wannier90 please cite

### ***Wannier90 as a community code: new features and applications***

G. Pizzi, V. Vitale, R. Arita, S. Blügel, F. Freimuth, G. Géranton, M. Gibertini, D. Gresch, C. Johnson, T. Koretsune, J. Ibañez-Azpiroz, H. Lee, J. M. Lihm, D. Marchand, A. Marrazzo, Y. Mokrousov, J. I. Mustafa, Y. Nohara, Y. Nomura, L. Paulatto, S. Poncé, T. Ponweiser, J. Qiao, F. Thöle, S. S. Tsirkin, M. Wierzbowska, N. Marzari, D. Vanderbilt, I. Souza, A. A. Mostofi, and J. R. Yates, *J. Phys. Cond. Matt.* **32**, 165902 (2020) [[ONLINE JOURNAL](#), [OPEN ACCESS](#)]

Please cite this new paper if you use a recent  
version of Wannier90  
(bibtex available on the homepage)

#### MARCH 2020 – VIRTUAL EDITION

March 13, 2020

Due to ongoing concerns related to the coronavirus pandemic, we have taken the decision to cancel the physical meeting of ... [Continue reading](#)

#### **Wannier90 (v3.1.0) released**

March 5, 2020

Wannier90 (v3.1.0) has been released today!

[ALL NEWS](#)

An updated version of wannier90: A tool for obtaining maximally-localised Wannier functions, AA Mostofi, JR Yates, G Pizzi, YS Lee, I Souza, D Vanderbilt, N Marzari, *Comput. Phys. Commun.* **185**, 2309 (2014) [[ONLINE JOURNAL](#)] [[bibTeX](#)]

If you are using v1.x, please cite instead:

wannier90: A tool for obtaining maximally-localised Wannier functions, AA Mostofi, JR Yates, YS Lee, I Souza, D Vanderbilt, N Marzari, *Comput. Phys. Commun.* **178**, 685 (2008) [[ONLINE JOURNAL](#)] [[OPEN ACCESS](#)] [[bibTeX](#)]

# A community effort!

## People

### Wannier Developers Group



**Arash Mostofi**  
Professor of Theory and  
Simulation of Materials –  
Imperial College London



**Jonathan Yates**  
Associate Professor of  
Materials Modelling –  
University of Oxford



**Giovanni Pizzi**  
Senior Scientist – EPFL



**Valerio Vitale**  
Postdoctoral Research  
Associate – University of  
Cambridge and Imperial  
College London



**Nicola Marzari**  
Chair of Theory and  
Simulation of Materials –  
EPFL



**Ivo Souza**  
Research Professor –  
University of the Basque  
Country



**David Vanderbilt**  
Professor of Condensed Matter  
Theory – Rutgers University

Wannier90 is a community development effort and would not be possible without the involvement and effort of a large number of contributors. A [full list of authors and contributors](#) is maintained on our [GitHub site](#).

Many more people involved  
in the past years:  
**Wannier90 transitioned  
to a community code**

*By citing the new paper, you acknowledge  
the important work of all these coauthors!*

**OPEN ACCESS**  
**IOP Publishing**

Journal of Physics: Condensed Matter

J. Phys.: Condens. Matter **32** (2020) 165902 (25pp)

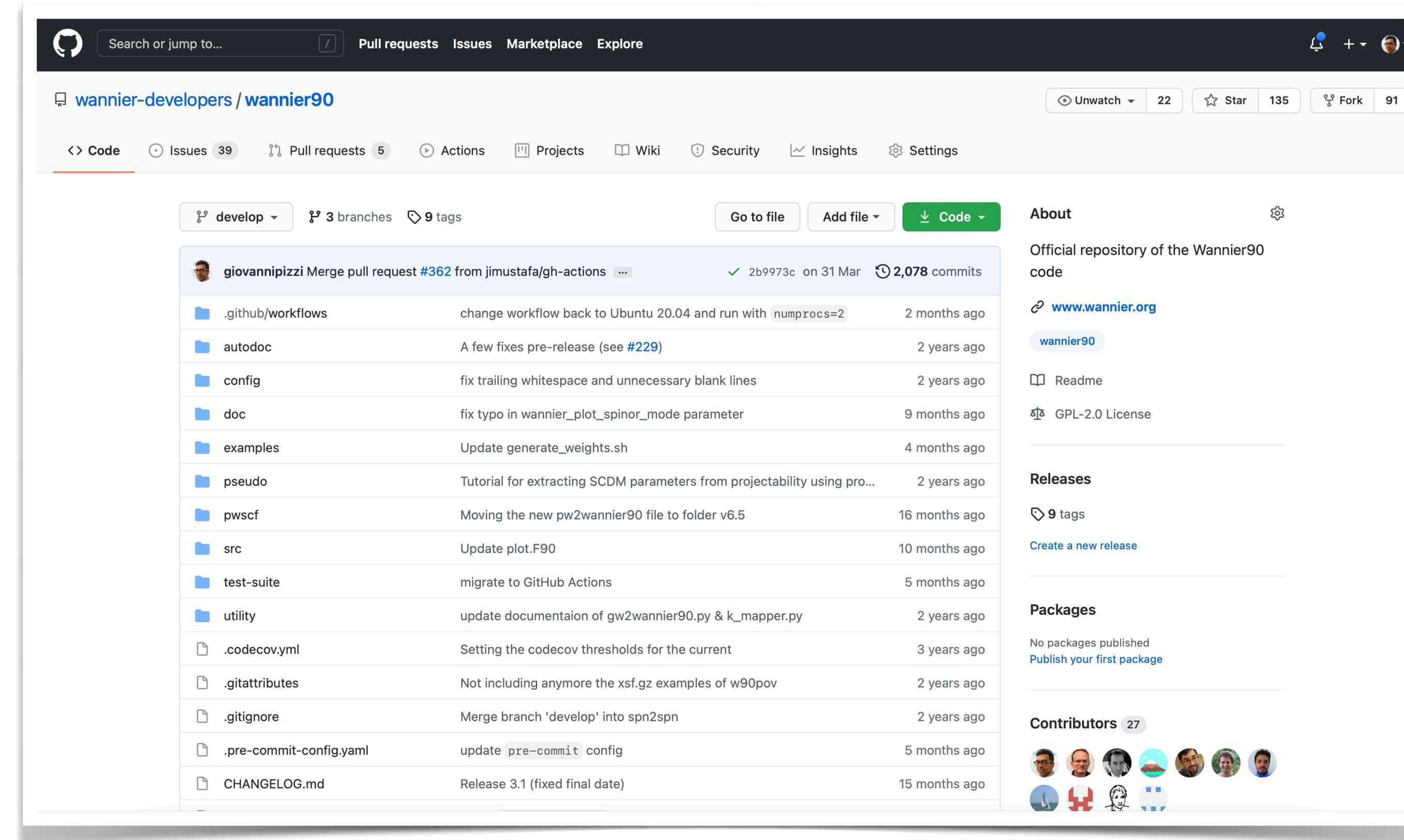
<https://doi.org/10.1088/1361-648X/ab51ff>

## Wannier90 as a community code: new features and applications

Giovanni Pizzi<sup>1,29,30</sup>, Valerio Vitale<sup>2,3,29</sup>, Ryotaro Arita<sup>4,5</sup>, Stefan Blügel<sup>6</sup>, Frank Freimuth<sup>6</sup>, Guillaume Géranton<sup>6</sup>, Marco Gibertini<sup>1,7</sup>, Dominik Gresch<sup>8</sup>, Charles Johnson<sup>9</sup>, Takashi Koretsune<sup>10,11</sup>, Julen Ibañez-Azpiroz<sup>12</sup>, Hyungjun Lee<sup>13,14</sup>, Jae-Mo Lihm<sup>15</sup>, Daniel Marchand<sup>16</sup>, Antimo Marrazzo<sup>1</sup>, Yuriy Mokrousov<sup>6,17</sup>, Jamal I Mustafa<sup>18</sup>, Yoshiro Nohara<sup>19</sup>, Yusuke Nomura<sup>4</sup>, Lorenzo Paulatto<sup>20</sup>, Samuel Poncé<sup>21</sup>, Thomas Ponweiser<sup>22</sup>, Junfeng Qiao<sup>23</sup>, Florian Thöle<sup>24</sup>, Stepan S Tsirkin<sup>12,25</sup>, Małgorzata Wierzbowska<sup>26</sup>, Nicola Marzari<sup>1,29</sup>, David Vanderbilt<sup>27,29</sup>, Ivo Souza<sup>12,28,29</sup>, Arash A Mostofi<sup>3,29</sup>, and Jonathan R Yates<sup>21,29</sup>.

# W90 as a community-driven code

<https://github.com/wannier-developers/wannier90>



You can be a contributor too!

**Create pull requests with documentation improvement,  
bug fixes, and new features: they are very welcome!**

# Code (old and new) features

---

## Calculation of Maximally-Localised Wannier Functions

- Wannier localisation scheme of Marzari and Vanderbilt [[REF](#)]
- Disentanglement scheme of Souza, Marzari and Vanderbilt [[REF](#)] for entangled bands (e.g. metals, conduction states)
- Optimised algorithm for Gamma-point calculations [[REF](#)]
- Symmetry-adapted Wannier functions [[REF](#)]
- Wannier functions without the need to define initial projections (via the SCDM method) [[REF1](#), [REF2](#)]
- Projection-only Wannier functions (without disentanglement and/or Wannierisation)
- Hamiltonian and position operators represented in the real-space Wannier function basis (eg, for use in tight-binding calculations)
- Spinor Wannier functions
- Export of Wannier functions for plotting as xsf (XCrySDen), cube format, and ray-tracing using POV-Ray
- Calculation of van der Waals energies [[REF1](#)], [[REF2](#)]
- Disentanglement within selected regions of k-space

Wannier90 exploits the real-space localisation of WFs to obtain many spectral and Fermi-surface properties at high-resolution in the Brillouin zone (so-called Wannier interpolation). Many of these properties can take advantage of multicore processors and compute clusters using MPI.

# Code (old and new) features

---

## Density of States

- Band structures
- Density of states (using fixed or adaptive smearing [[REF](#)])
- Wannier projected DOS and bandstructure
- Total spin moment
- Fermi surfaces (via bxsf file)
- GW bands interpolation (via an interface to the Yambo code)

## Berry phase properties including:

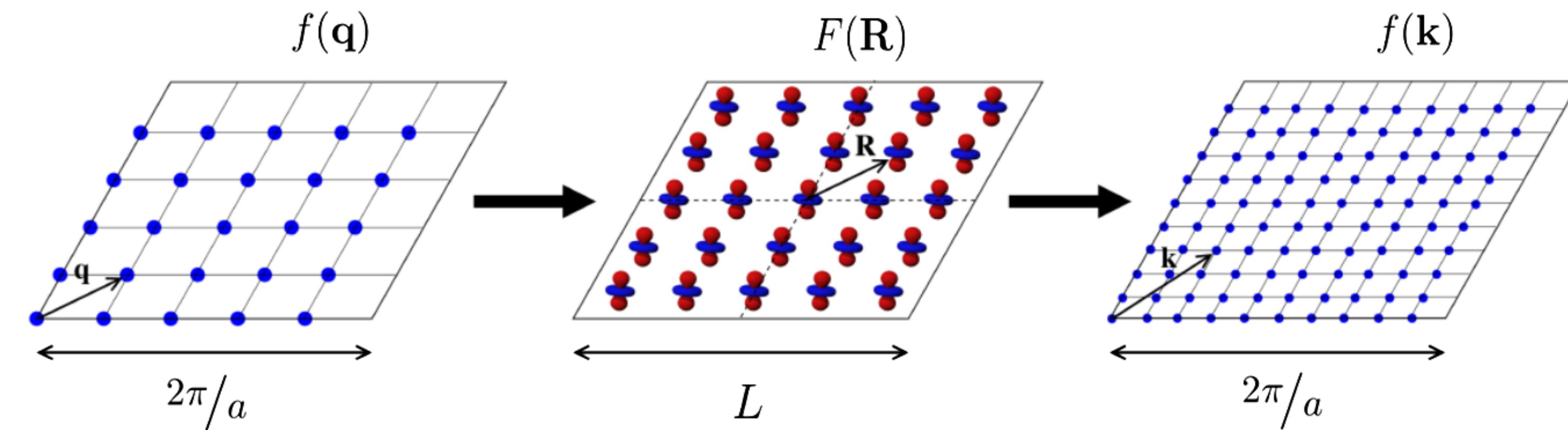
- Berry curvature [[REF](#)]
- Anomalous Hall conductivity [[REF](#)]
- Orbital magnetisation [[REF](#)]
- Shift currents [[REF](#)]
- Gyrotropic effects [[REF](#)]

## Transport

- Ballistic (Landauer-Buttiker) transport [[REF1](#)], [[REF2](#)], [[REF3](#)])
- Boltzmann transport (BoltzWann) [[REF](#)]
  - Boltzmann transport equation in the relaxation time approximation
  - Electrical conductivity
  - Seebeck coefficients
  - Electronic contribution to the thermal conductivity
  - Spin Hall conductivity [[REF](#)]

# Improved Wannier interpolation by minimal-distance replica selection

**Wannier interpolation:** similar to Fourier interpolation, which uses discrete Fourier transforms to reconstruct faithfully continuous signals from a discrete sampling, provided that the signal has a finite bandwidth and that the sampling rate is at least twice the bandwidth (Nyquist–Shannon condition).



$$\tilde{H}_{mn\mathbf{R}} = \langle w_{m\mathbf{0}} | H | w_{n\mathbf{R}} \rangle = \frac{1}{N} \sum_{j=1}^N e^{-i\mathbf{k}_j \cdot \mathbf{R}} H_{mn\mathbf{k}_j}$$

$$H_{mn\mathbf{k}} = \sum_{\mathbf{R}'} e^{i\mathbf{k} \cdot \mathbf{R}'} \tilde{H}_{mn\mathbf{R}'}$$

exponential decay w.r.t.  $|\mathbf{R}|$

The interpolated band structure at an arbitrary  $\mathbf{k}$ -point can be obtained by diagonalising  $H_{mn\mathbf{k}}$ , and many other quantities can be interpolated using the same approach!

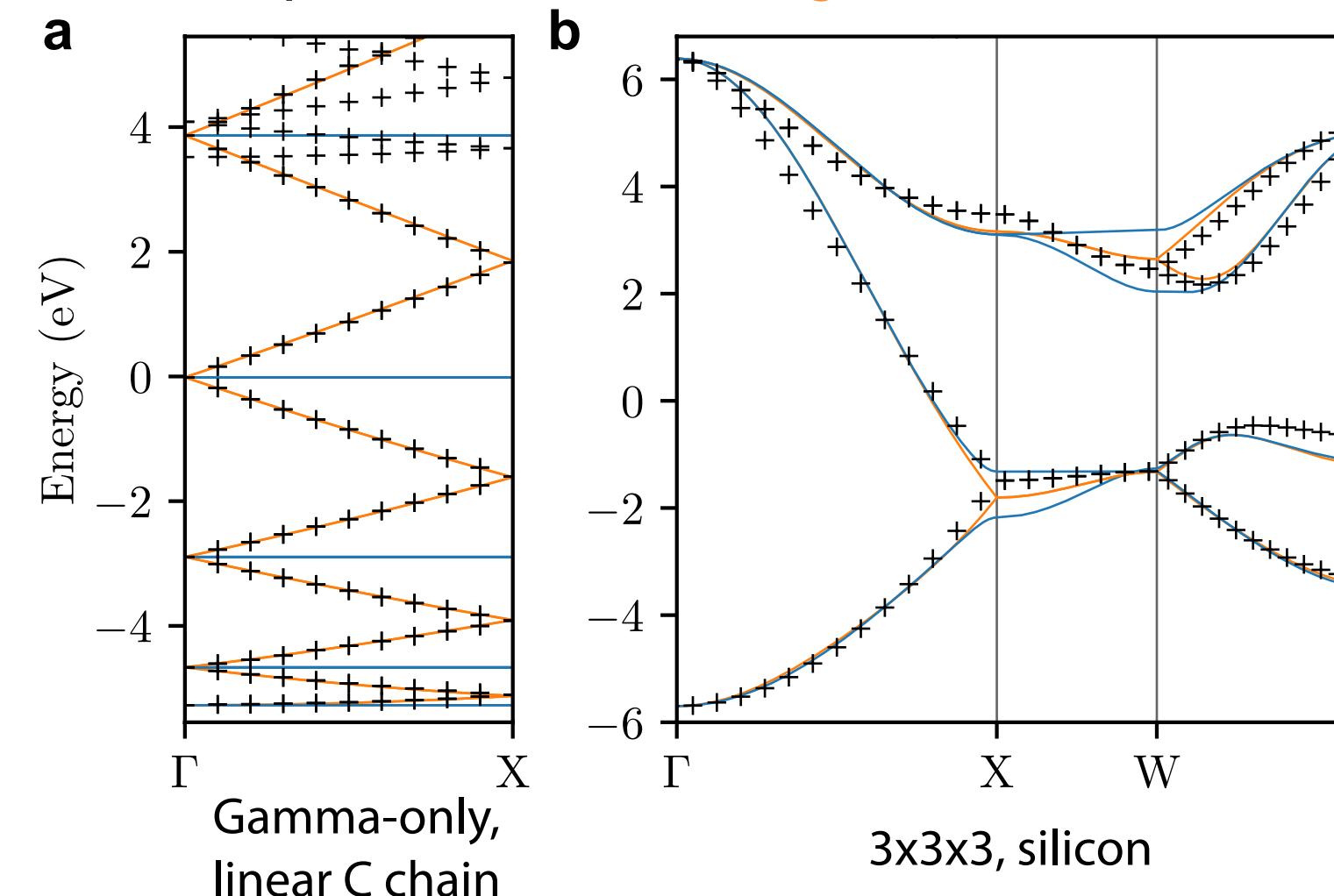
# Improved Wannier interpolation by minimal-distance replica selection

The diagram illustrates the Wigner-Seitz supercell (orange boundary) containing a green parallelogram representing the Brillouin zone. A point  $\mathbf{r}_m$  is shown within the zone. Three Wannier functions are depicted:  $w_{n\mathbf{R}}(\mathbf{r})$  (blue),  $w_{m\mathbf{0}}(\mathbf{r})$  (orange), and  $w_{n\mathbf{R}+\mathbf{T}}(\mathbf{r})$  (blue). Vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$  define the parallelogram. The equation shows the transition from the old implementation to the improved one:

$$H_{mn\mathbf{k}} = \sum_{\mathbf{R}'} e^{i\mathbf{k}\cdot\mathbf{R}'} \tilde{H}_{mn\mathbf{R}'} \quad \longrightarrow \quad H_{mn\mathbf{k}} = \sum_{\mathbf{R}} \frac{1}{N_{mn\mathbf{R}}} \sum_{j=1}^{N_{mn\mathbf{R}}} e^{i\mathbf{k}\cdot(\mathbf{R} + \mathbf{T}_{mn\mathbf{R}}^{(j)})} \tilde{H}_{mn\mathbf{R}}$$

$\{\mathbf{T}_{mn\mathbf{R}}^{(j)}\}$  vectors  $\mathbf{T}$  that minimise the distance  $|\mathbf{r}_m - (\mathbf{r}_n + \mathbf{R} + \mathbf{T})|$

Blue: old implementation, orange: current default in Wannier90 (v3.x)



Clear example:  
gamma-only

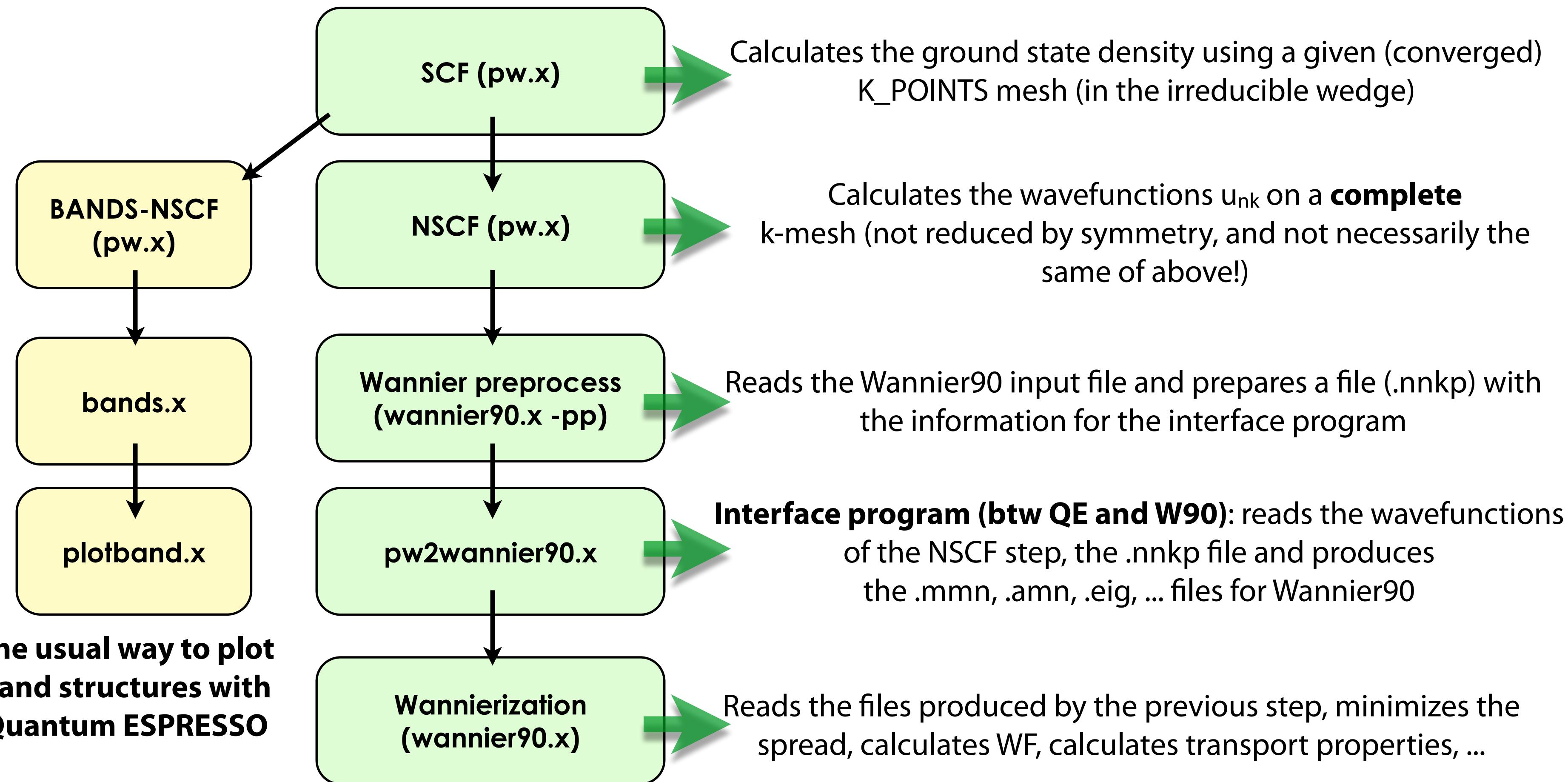
$$H_{mn\mathbf{k}} = \frac{\tilde{H}_{mn\mathbf{0}}}{N_{mn\mathbf{0}}} \sum_{j=1}^{N_{mn\mathbf{0}}} e^{i\mathbf{k}\cdot\mathbf{T}_{mn\mathbf{0}}^{(j)}}$$

**Better interpolation and degeneracies,  
especially for very coarse grids**

# Wannier90 “input data”

- W90 needs the overlap matrices  $M_{mn}^{(k,k+b)}$  between neighboring  $k$  points, and the  $A_{mn}(k)$  projection matrices
- Other possible (and common) inputs:
  - the **list of Hamiltonian eigenvalues**  $E_n(k)$  at each  $k$ -point (for interpolation)
  - the  $u_{nk}(r)$  in real space (for plotting the WFs)
  - Other (optional) matrices: spin components,  $uHu$  and  $ulu$ , and more.
- This input can be obtained from various programs; there exists interfaces for a set of ab-initio codes
  - We will use **Quantum ESPRESSO (QE)**
  - **Reminder:**  $pw.x$  documentation in  
[https://www.quantum-espresso.org/Doc/INPUT\\_PW.html](https://www.quantum-espresso.org/Doc/INPUT_PW.html)  
(you can find the link in the PDF with the exercises)

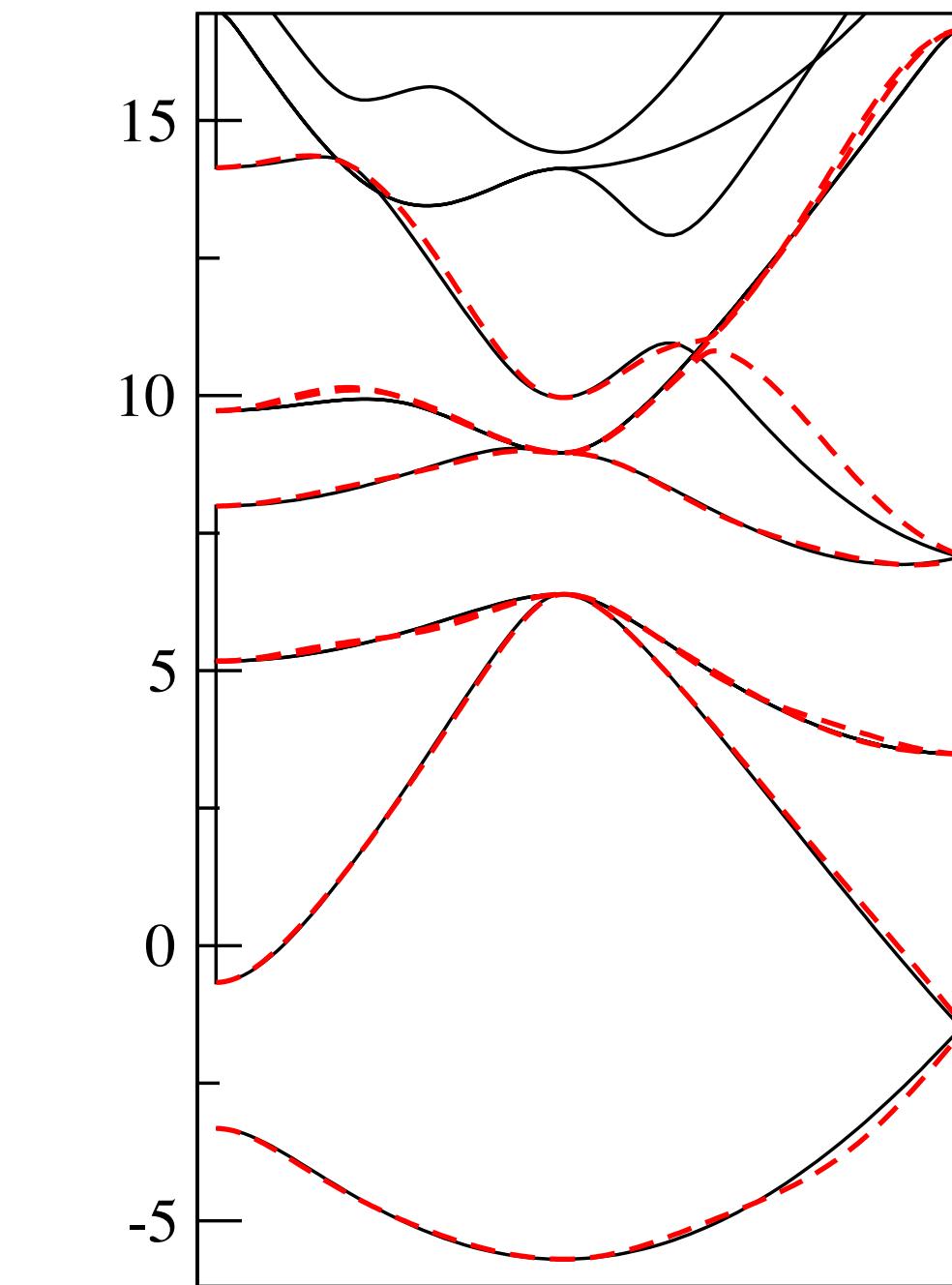
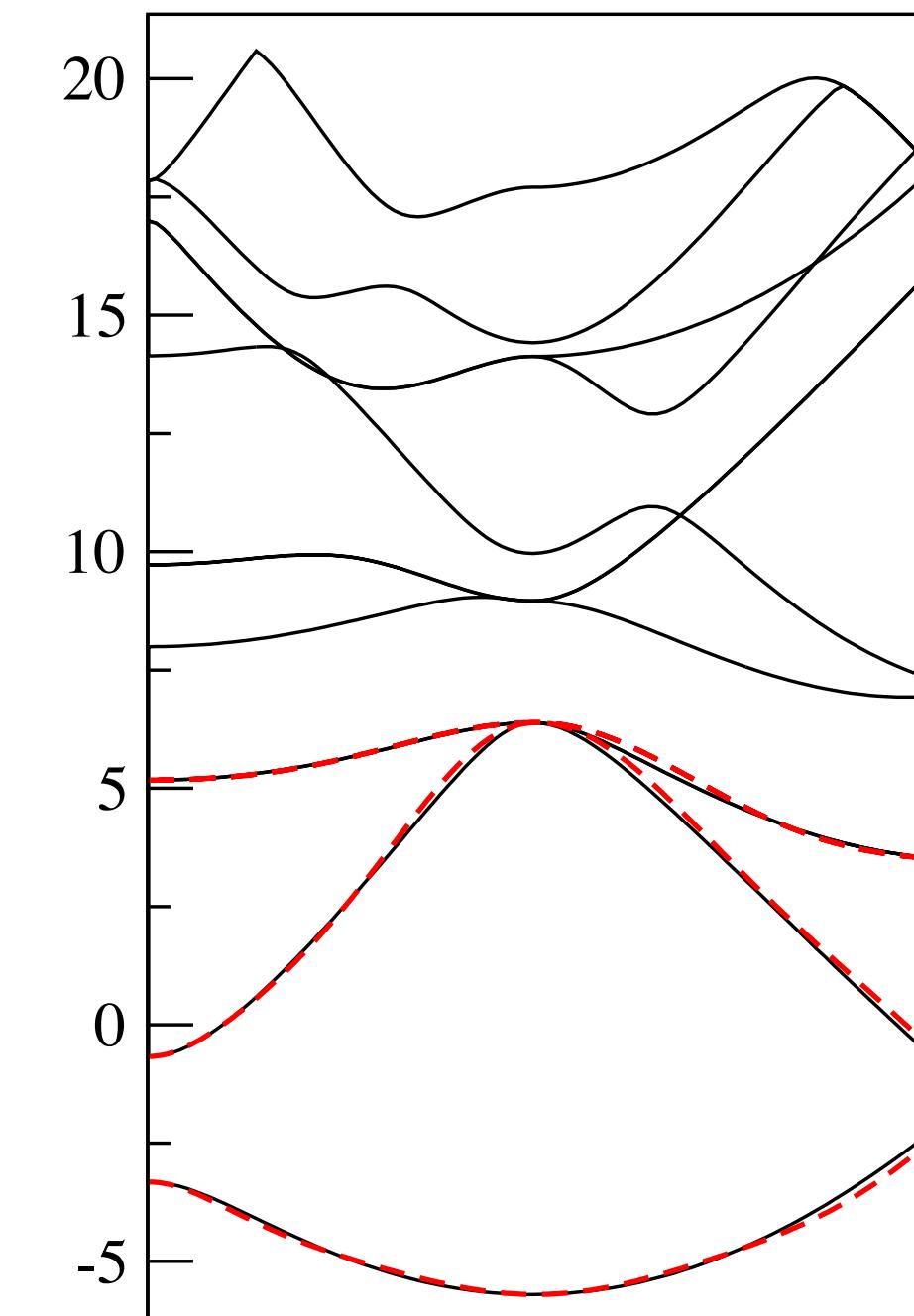
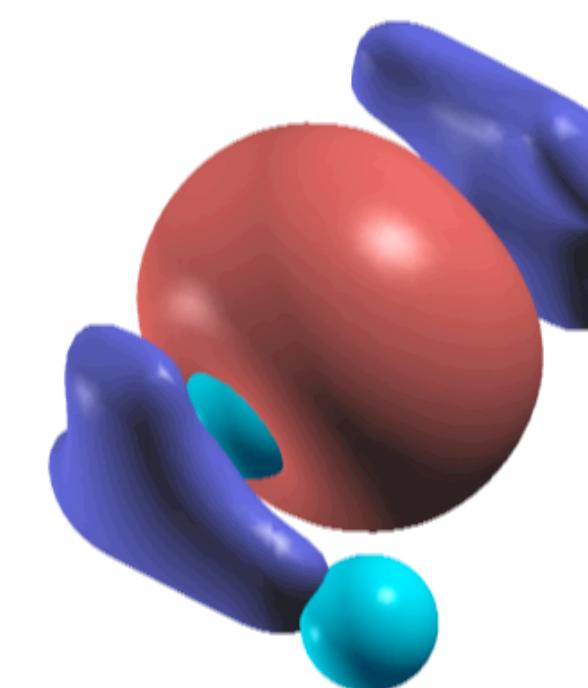
# How to run a Wannier90 calculation



**Note: DON'T MIX the yellow and green path!** Otherwise the content of the 'output' folder of Quantum ESPRESSO is overwritten and you will get some error. First follow one path (e.g. the yellow one), and when you get to the bottom box, start again from NSCF

# Exercises 1 and 2: Silicon

- Calculate Wannier functions for Silicon: valence band only (Ex. 1), and valence band+conduction band
- Check the results
- Plot the real-space WFs (using **XCrysDen** or **VESTA**)
  - You will need to run these codes on your computers
- Plot the ab-initio and the interpolated band structure (using **xmGrace** or **gnuplot**)



# The Quantum ESPRESSO input file

**NAMELISTS**

```
&control
  calculation = 'scf'
  restart_mode = 'from_scratch'
  prefix       = 'si'
  pseudo_dir   = 'pseudo/'
  outdir       = 'out/'

/>
&system
  ibrav      = 0
  nat        = 2
  ntyp       = 1
  ecutwfc    = 25.0
  ecutrho    = 200.0

/>
&electrons
  conv_thr   = 1.0d-10

/>
ATOMIC_SPECIES
Si 28. Si.pbe-n-van.UPF
ATOMIC_POSITIONS crystal
Si -0.25 0.75 -0.25
Si 0.00 0.00 0.00
K_POINTS automatic
10 10 10 0 0 0
CELL_PARAMETERS bohr
-5.1 0.0 5.1
0.0 5.1 5.1
-5.1 5.1 0.0
```

→ Type of calculation, location of pseudopotentials and of output files, ...

→ System description (number of atoms and of species, energy cutoffs, ...)

→ Thresholds for charge-density calculations

→ Definition of species ("atom types"), with mass and pseudopotential files

→ Definition of atomic positions

→ Definition of k-points grid (10x10x10 here, no shift = including Gamma)

→ Definition of the cell (FCC here)

# The Quantum ESPRESSO input generator

<https://www.materialscloud.org/work/tools/qeinputgenerator>

Quantum ESPRESSO input generator and structure visualizer

- ▶ [About the Quantum ESPRESSO input generator and structure visualizer](#)
- ▶ [Instructions](#)
- ▶ [Acknowledgements](#)

**Upload your structure**

Upload a crystal structure:  no file selected

Select here the file format:

Select here the pseudopotential library:

Select here the magnetism/smearing:

Select here the k-points distance (1/Å)

(and smearing (eV) in case of fractional occupations):

Refine cell (using spglib):

By continuing, you agree with the [terms of use](#) of this service.

# The Quantum ESPRESSO input generator

<https://www.materialscloud.org/work/tools/qeinputgenerator>

If you use the results of this tool in a publication, please cite the following works:

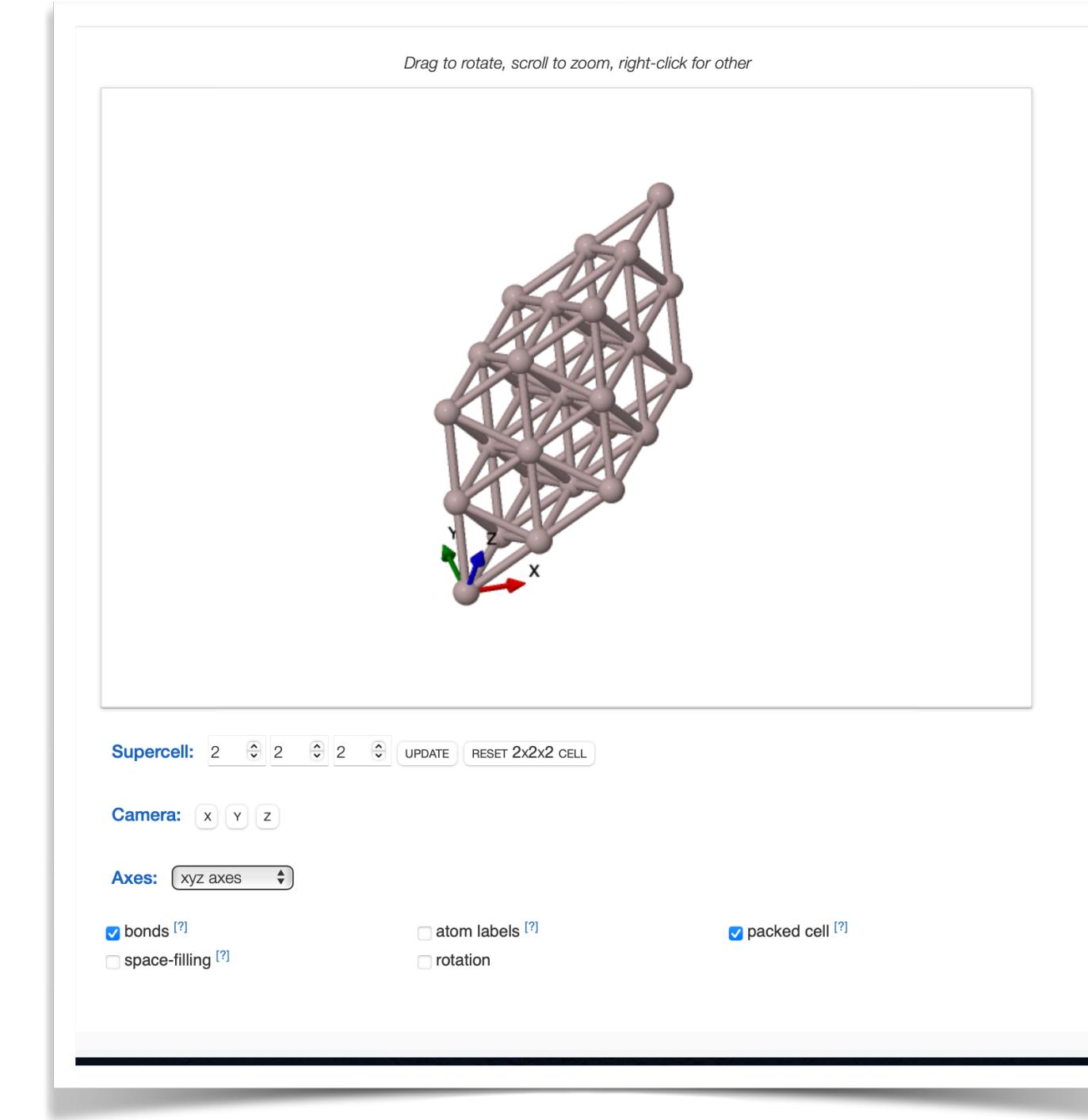
- SSSP (for the pseudopotential library)  
G. Prandini, A. Marrazzo, I. E. Castelli, N. Mounet and N. Marzari, *npj Computational Materials* **4**, 72 (2018).  
WEB: <http://materialscloud.org/sssp>.
- Pseudopotentials:
  - Al.pbesol-n-kjpaw\_psl.1.0.0.UPF,  
from [Pslibrary 1.0.0](#): A. Dal Corso, *Comput. Mater. Sci.* **95**, 337 (2014).  
DOI: [10.1016/j.commatsci.2014.07.043](https://doi.org/10.1016/j.commatsci.2014.07.043), WEB: <http://www.quantum-espresso.org/pseudopotentials>, LICENSE: GNU General Public License (version 2 or later).

[Download zip of input file and pseudopotentials](#) [Change parameters](#) [Choose a different structure](#)

Quantum ESPRESSO PWscf input

[Copy to clipboard](#)

```
&CONTROL
  calculation = 'scf'
  etot_conv_thr =  1.000000000d-05
  forc_conv_thr =  1.000000000d-04
 outdir = './out/'
  prefix = 'aiida'
  pseudo_dir = './pseudo/'
  tprnfor = .true.
  tstress = .true.
  verbosity = 'high'
/
&SYSTEM
  degauss =  1.4699723600d-02
  ecutrho =  2.400000000d+02
  ecutwfc =  3.000000000d+01
  ibrav = 0
  nat = 1
  nosym = .false.
  ntyp = 1
  occupations = 'smearing'
  smearing = 'cold'
/
&ELECTRONS
  conv_thr =  2.000000000d-10
  electron_maxstep = 80
  mixing_beta =  4.000000000d-01
/
ATOMIC_SPECIES
Al    26.981538 Al.pbesol-n-kjpaw_psl.1.0.0.UPF
ATOMIC_POSITIONS crystal
Al      0.000000000     0.000000000     0.000000000
K_POINTS automatic
14 14 14 0 0 0
CELL_PARAMETERS angstrom
  2.020000000     2.020000000     0.000000000
  2.020000000     0.000000000     2.020000000
  0.000000000     2.020000000     2.020000000
```



- Return optimal parameters and pseudopotentials from SSSP [1,2]
- Copy-paste and download options
- Also works as a structure visualiser!

[1] <https://www.materialscloud.org/sssp>

[2] G. Prandini\*, A. Marrazzo\* *et al.*, *npj Comp. Mat.* **4**, 72 (2018)

# How to run and input file

- The Wannier90 input file must have a .win extension (e.g.: **ex1.win**)
- To run the code, **pass the *basename*** (i.e., the name without the .win extension) **as a command line parameter** to wannier90.x:

**wannier90.x -pp ex1 (for the pre-process step)**

**wannier90.x ex1 (for the Wannierization step)**

- Input file format: very simple, there are *no* namelists but only:
  - **Variables** (order is not important; not case sensitive)  
**num\_wann = 4**  
**mp\_grid : 6 6 6**
  - **Blocks**  
**begin atoms\_frac**  
**Si -0.25 0.75 -0.25**  
**Si 0.00 0.00 0.00**  
**end atoms\_frac**
- **Default units:** lengths are **angstrom** (bohr are also accepted), energies are **eV**

# Example of input file (ex1)

```
num_bands      = XXX
num_wann       = XXX
num_iter       = 100

! restart       = plot
wannier_plot    = true
wannier_plot_supercell = 3

bands_plot     = true
begin kpoint_path
L 0.5 0.5 0.5 G 0.0 0.0 0.0
G 0.0 0.0 0.0 X 0.5 0.0 0.5
end kpoint_path

begin projections
f=-0.125,-0.125,0.375:s
f= 0.375,-0.125,-0.125:s
f=-0.125, 0.375,-0.125:s
f=-0.125,-0.125,-0.125:s
end projections
```

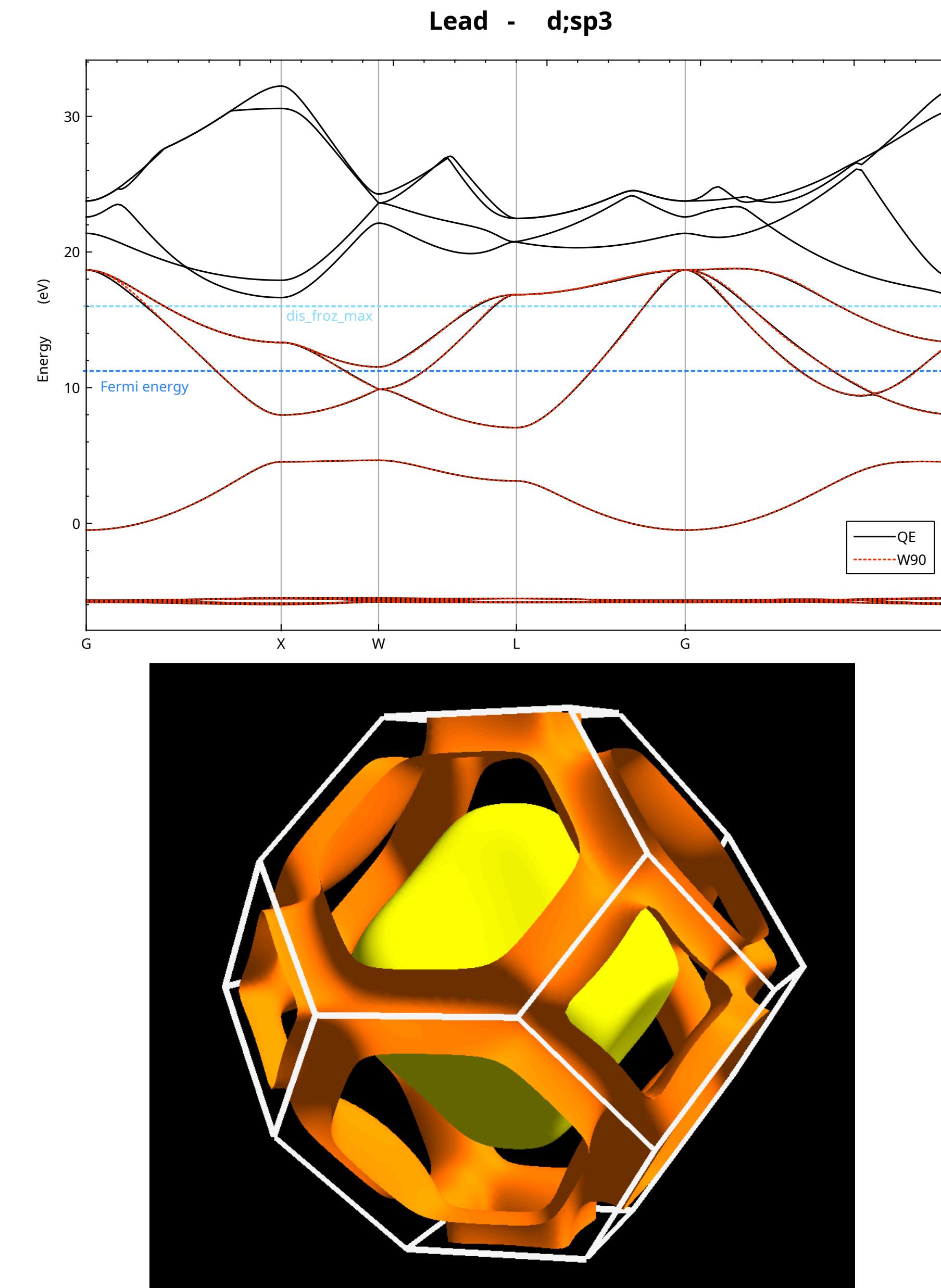
```
mp_grid = XXX XXX XXX
begin kpoints
XXX
XXX
XXX
end kpoints

begin atoms_frac
Si -0.25 0.75 -0.25
Si  0.00 0.00 0.00
end atoms_frac

begin unit_cell_cart
bohr
-5.10 0.00 5.10
  0.00 5.10 5.10
-5.10 5.10 0.00
end unit_cell_cart
```

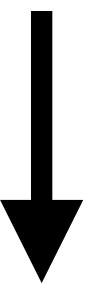
# Exercise 3: band structure and Fermi surface of lead

- Interpolate the band structure of lead
- Show the Fermi surface of lead
  - Requires a very dense grid of points in the BZ!  
 $\sim 50^3 = 125000$
  - Wannier interpolation essential to compute it efficiently



# Wannier interpolation for beyond-DFT methods

Wannier interpolation is particularly useful to calculate **band structure with beyond-DFT methods**, such as **hybrid functionals** (HSE), **Koopmans-compliant functional** and **many-body perturbation theory** (MBPT GW)



- Only in **plain DFT** the potential is actually a functional of the sole charge density, so **the Hamiltonian can be easily diagonalised at any arbitrary k-point**: regular band structure calculation with LDA or PBE.
- **Otherwise, the k-point needs to be part of the grid** used to compute the total energy (hybrids) or the self energy (MBPT). Hence, **dense band structures** can be computed in practice **only through interpolation methods**, such as Wannier interpolation.

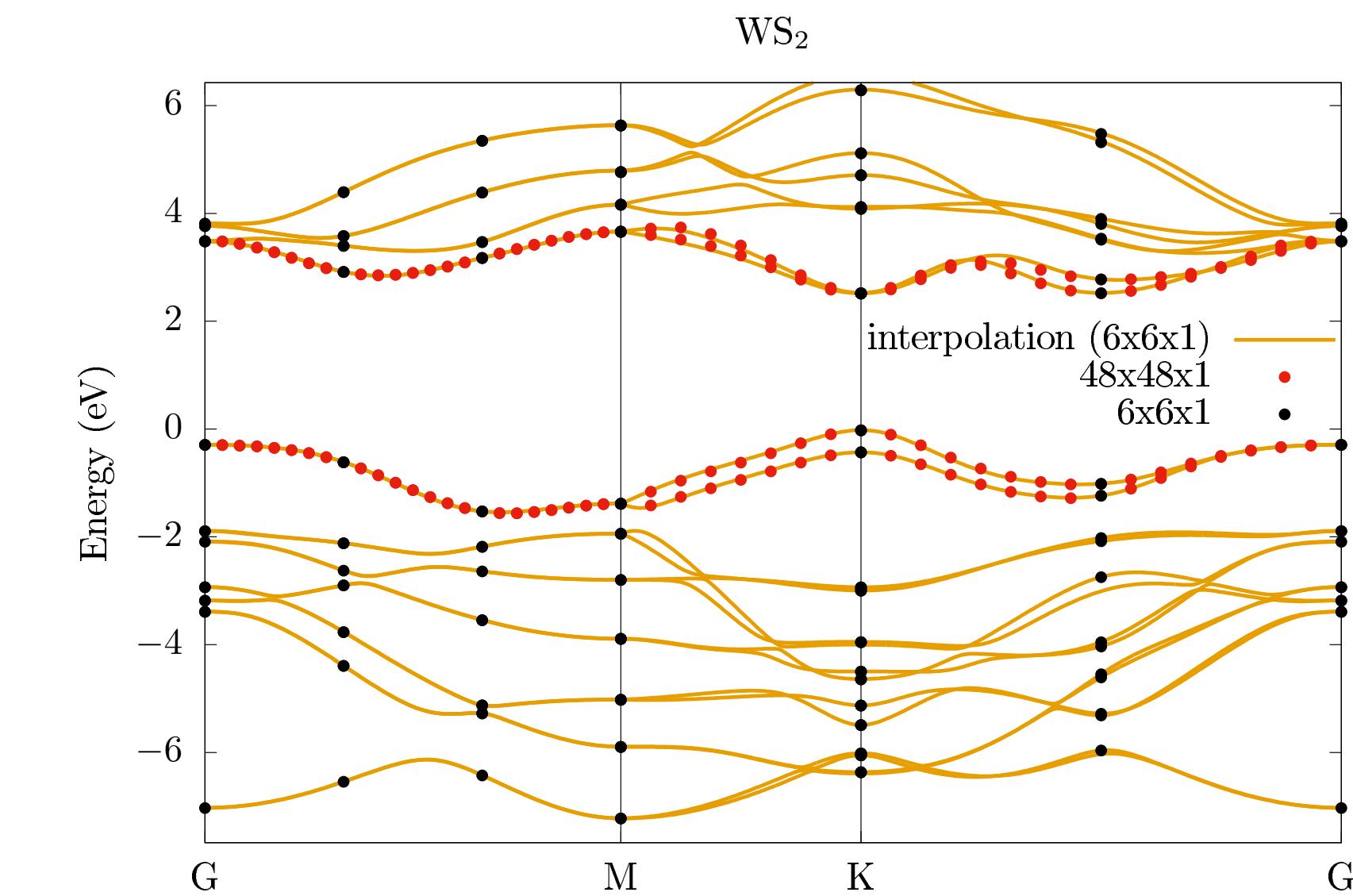
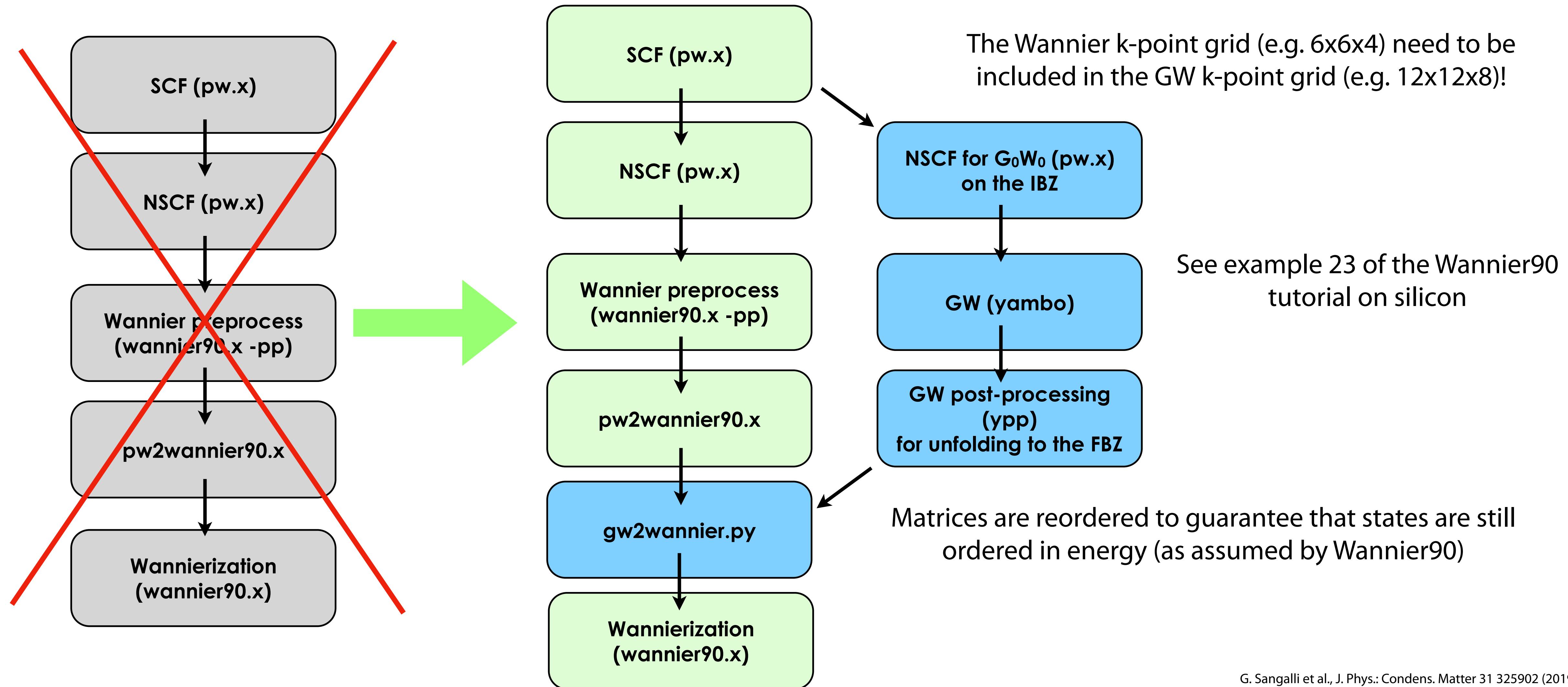


FIG. 6: (Color online).  $G_0W_0$  band structure of monolayer  $WS_2$  including spin-orbit coupling and using  $48 \times 48 \times 1$   $k$ -points grid for the self-energy. The orange lines represent Wannier-interpolated bands obtained from 7 QP energies corresponding to a  $6 \times 6 \times 1$  grid (black dots), while the red dots shows the QP energies of the full  $48 \times 48 \times 1$  grid.

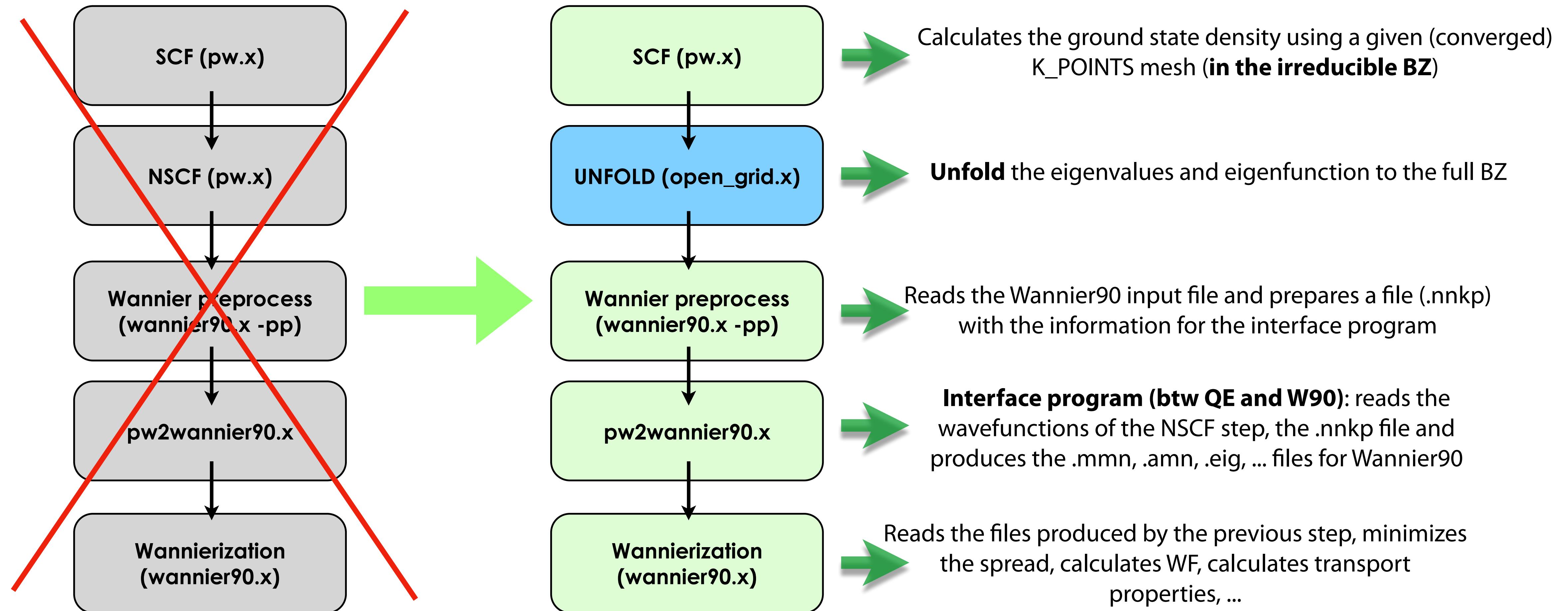
# Wannier interpolation for $G_0W_0$ with QE & YAMBO

$G_0W_0$  is done on the IBZ, on a (possibly denser) k-point grid, quasi-particle corrections on the (possibly less dense) Wannier grid and unfolded to the full BZ.



# Wannier interpolation for hybrid-functionals with QE

No NSCF with hybrids, but unfolding from the IBZ to the FBZ with open\_grid.x



Example at [https://gitlab.com/QEF/q-e/-/tree/master/PP/examples/W90\\_open\\_grid\\_example](https://gitlab.com/QEF/q-e/-/tree/master/PP/examples/W90_open_grid_example)

# Getting help

- **Today:** ask me and Junfeng :-)
- **From tomorrow on:** [www.wannier.org](http://www.wannier.org)
  - User guide, tutorials (with solutions)
  - Register to the Wannier90 mailing list
  - Actually read the source code!



Advanced Quantum ESPRESSO tutorial:  
Hubbard and Koopmans functionals from linear response  
Virtual tutorial from 9 to 11 November 2022

Hands-On Tutorial  
Wannierization and Wannier interpolation of band structure and Fermi surface  
J. Qiao, A. Marrazzo, G. Pizzi

## Foreword

This document contains 2 exercises to help you learn `wannier90` (W90) - the computer program that calculates maximally-localised Wannier functions (MLWFs).

## Required executables and tutorial files

- QE: `pw.x`, `bands.x`, `plotband.x`, `pw2wannier90.x`
- Wannier90: `wannier90.x`, `kmesh.pl`

The tutorial files are hosted on GitHub repo:

<https://github.com/materialscloud-org/hubbard-koopmans-2022/tree/main/Day1>

You can download the files through `git clone`:

```
git clone https://github.com/materialscloud-org/hubbard-koopmans-2022
```

## Software for visualization

- To plot the band structure:
  - gnuplot <http://www.gnuplot.info/index.html>
  - xmgrace <https://plasma-gate.weizmann.ac.il/Grace/>
- To plot the Wannier functions, Fermi surface:
  - xcrysden <http://www.xcrysden.org/>

These are already pre-installed in Quantum Mobile.

## 1 Silicon valence and conduction bands

In this exercise you will learn how to obtain maximally-localised Wannier functions (MLWFs) for the valence and conduction bands of silicon.

- Go to the `exercise5/` folder and inspect the input file `01_scf.in`. The first step is to perform a ground-state calculation for a silicon crystal (FCC) with two atoms per unit cell. Check if you understand all parameters (you can use the web page [https://www.quantum-espresso.org/Doc/INPUT\\_PW.html](https://www.quantum-espresso.org/Doc/INPUT_PW.html) for keywords that you do not know or ask us). To visualize the crystal structure, there are two options, you can choose either one as you prefer:
  - Quantum ESPRESSO input generator and structure visualizer: open the following link in a browser,  
<https://www.materialscloud.org/work/tools/qeinputgenerator>, click **Choose File** button, select the `01_scf.in` file, and click “Generate the PWscf input file” button. In the new webpage you find a 3D visualization of the structure.
  - xcrysden: either opening the program and selecting from the menu  
**File**→**Open PWscf...**→**Open PWscf Input File**  
 and selecting the input file, or directly from the command line with the command

```
xcrysden --pwi 01_scf.in
```

```
&control
  calculation      = 'scf'
  restart_mode     = 'from_scratch'
  prefix           = 'si'
  pseudo_dir       = '../files/pseudo/'
  outdir          = 'out/'
/
&system
  ibrav            = 0
  nat              = 2
  ntyp             = 1
  ecutwfc          = 25.0
  ecutrho          = 200.0
/
&electrons
  conv_thr         = 1.0d-10
/
ATOMIC_SPECIES
  Si 28. Si.pbe-n-van.UPF
ATOMIC_POSITIONS crystal
  Si -0.25  0.75  -0.25
  Si  0.00  0.00   0.00
K_POINTS automatic
  10 10 10 0 0 0
CELL_PARAMETERS bohr
  -5.1    0.0    5.1
  0.0     5.1    5.1
```

|      |     |     |
|------|-----|-----|
| -5.1 | 5.1 | 0.0 |
|------|-----|-----|

- The Si pseudopotential that we will use for the calculation has  $Z_{val} = 4$  (this information can be obtained reading the first lines of the pseudopotential file, for instance with the command `less .../.../files/pseudo/Si.pbe-n-van.UPF`). Using the information given above, and knowing that FCC Si is a semiconductor, how many occupied valence bands do you expect (and why)?

- 
- Run the ground state calculation using the `pw.x` code of the Quantum ESPRESSO suite. The syntax for codes in the Quantum ESPRESSO suite is: `command < inputfile > outputfile` (i.e. `pw.x < 01_scf.in > scf.out`). You may want to use the parallelization to run the simulation faster, e.g. for 2 processors, use `mpirun`:

```
mpirun -np 2 pw.x < 01_scf.in > scf.out
```

- After the calculation finishes, inspect the output file `scf.out` to check if there are any errors/warnings. Compare your answer to the previous point (number of electrons and occupied valence bands) with the information provided in the output file.
- Now we want to plot the band structure of silicon. Copy the file `01_scf.in` to a new file `02_bands.in`. Do the following modifications to the file `02_bands.in` (use the INPUT\_PW documentation from the link above for an explanation of the meaning of the flags, if needed):
  - In the `CONTROL` namelist, change the `calculation` keyword from '`scf`' to '`bands`' to perform a band structure calculation starting from the ground state density obtained from the `scf` run.
  - Ask the code to print 12 bands (flag `nbnd=12` in the `SYSTEM` namelist).
  - Set `diago_full_acc = .true.` in the `ELECTRONS` namelist (see documentation for the meaning of this flag).
  - Change the k-point list to plot the band structure along the following path (coordinates are given in crystal units), using 50 points per segment:
    - \*  $L(0.5, 0.5, 0.5) \rightarrow \Gamma(0, 0, 0)$
    - \*  $\Gamma(0, 0, 0) \rightarrow X(0.5, 0, 0.5)$

You can do this simply using the following `K_POINTS` card:

```
K_POINTS crystal_b
3
0.5 0.5 0.5 50
0. 0. 0. 50
0.5 0. 0.5 50
```

- Run the calculation using the `pw.x` code, as explained before.
- When the calculation has finished, run the file `03_bandsx.in` through the `bands.x` executable (make sure to read and understand the input file):

```
mpirun -np 2 bands.x < 03_bandsx.in > bandsx.out
```

```
&bands
  prefix = 'si'
  outdir = 'out/'
  filband = 'bands.dat'
  lsym = .false.
/
```

This will produce the `bands.dat` file.

- Finally, execute the `plotband.x` code (interactively) and answer to its questions. In particular, the input file is the `bands.dat` file created in the previous step; call the xmgrace file `qebands.agr`. When asked, call the ps file `qebands.ps`. You will be asked to provide the value of the Fermi level, which in this case can be put equal to the highest occupied energy level (see the output file `scf.out`). When asked for the deltaE and reference E for the energy axis, type 2 and Fermi level (use space to separate the 2 numbers), you can also tweak these 2 numbers to adjust the visual output of the figure. At the end, open the xmgrace file (or directly the postscript PS file) and inspect the band structure, identifying the valence and conduction bands.
- Now we are ready to calculate the wavefunctions on a complete grid of k-points. Copy the `02_bands.in` file that you created before to `05_nscf.in`, and modify the following:
  - Change the `calculation` type from `'bands'` to `'nscf'`.
  - Keep the number of bands as 12.
  - Change the k-point list to a full  $4 \times 4 \times 4$  Monkhorst-Pack mesh, that will be used to calculate the overlap matrices needed to obtain Wannier functions. To obtain the list of k-points, use the `kmesh.pl` utility which is bundled in the `Wannier90` code, that you can find it here:

```
/home/max/koopmans/quantum_espresso/q-e/wannier90-3.1.0/utility/kmesh.pl
```

To allow easier usage, let's add it to the `$PATH` variable:

```
export PATH=/home/max/koopmans/quantum_espresso/q-e/wannier90-3.1.0/utility:$PATH
```

After this you can directly call the `kmesh.pl` utility from the command line.

Then generate the kpoint coordinates using the following command for a  $4 \times 4 \times 4$  mesh:

```
kmesh.pl 4 4 4
```

(use the command without parameters to get an explanation of its usage).

The output of the command will be a list of k-points, which you can copy and paste in the `K_POINTS` card of the `05_nscf.in` file.

- Run the nscf calculation using the `pw.x` code:

```
mpirun -np 2 pw.x < 05_nscf.in > nscf.out
```

- Now we have to prepare the input file for Wannier90. Open the file **si2.win**, which is a template of the Wannier90 input file (note that Wannier90 input file must have the **.win** extension). Change the values marked with **XXX** inserting the correct values. In particular:
  - Insert the **num\_bands** value (this must be equal to the **nbnd** value set in the nscf calculation if the **exclude\_bands** parameter is not specified).
  - Insert the **num\_wann** value (this is the number of requested Wannier functions: in this case for valence and conduction bands, this is equal to 8, why? You may find some hints from the **projections** keyword as will be shown in later paragraph).
  - Set the **mp\_grid** value to **4 4 4** (since we are using a  $4 \times 4 \times 4$  k-mesh).
  - Insert, between the **begin kpoints** and **end kpoints** lines, the list of the 64 kpoints, one per line. Note that while **pw.x** requires four numbers per line (the three coordinates of each kpoint, and the weight), Wannier90 needs only three numbers (the three coordinates). To obtain these lines, use again the **kmesh.pl** utility, but this time specifying a fourth parameter to get the list in the Wannier90 format:

```
kmesh.pl 4 4 4 wan
```

**Note** Using the **kmesh.pl** utility, we are sure that we provide enough significant digits, and that the list of k-points given to **pw.x** and to **Wannier90** is the same.

- Set the maximum energy for the frozen window (flag **dis\_froz\_max**) inside the energy gap (use the band plot obtained in previous step to get a value for this flag).
- Set the maximum energy for the disentanglement (flag **dis\_win\_max**) to an energy large enough so as to contain enough bands for each k point; 17.0 eV should be a reasonable value (check where this value lies in the band plot).
- Inspect the remaining part of the input file, using the Wannier90 user guide (that can be found on the [https://github.com/wannier-developers/wannier90/raw/v3.1.0/doc/compiled\\_docs/user\\_guide.pdf](https://github.com/wannier-developers/wannier90/raw/v3.1.0/doc/compiled_docs/user_guide.pdf) page) for the input flags that you do not understand. Try to understand, in particular, the **projections** section (project to 4  $sp^3$  orbitals for each Si atom in the unit cell).

```
use_ws_distance = .true.

num_bands      = XXX
num_wann       = XXX
num_iter       = 100

dis_win_max    = XXX
dis_froz_max   = XXX
dis_num_iter   = 100

!! To plot the WFs
! restart          = plot
wannier_plot    = true
```

```
wannier_plot_supercell = 3

!! To plot the WF interpolated band structure
bands_plot      = true
begin kpoint_path
L 0.50000  0.50000  0.5000 G 0.00000  0.00000  0.0000
G 0.00000  0.00000  0.0000 X 0.50000  0.00000  0.5000
end kpoint_path

begin projections
Si: sp3
end projections

begin atoms_frac
Si -0.25  0.75  -0.25
Si  0.00  0.00   0.00
end atoms_frac

begin unit_cell_cart
bohr
-5.10  0.00  5.10
 0.00  5.10  5.10
-5.10  5.10  0.00
end unit_cell_cart

mp_grid = XXX XXX XXX
begin kpoints
XXX
end kpoints
```

- Finally, we are ready to perform a Wannier90 calculation. This is done in three steps:

- We first run a preprocessing step using the command

```
wannier90.x -pp si2
```

which produce a **si2.wout** file and **si2.nnkp** file, that contains the relevant information from the Wannier90 input file in a format to be used in the next step.

- Then we run the **pw2wannier90.x** code (of the Quantum ESPRESSO distribution). The input file for **pw2wannier90.x** is provided (file **06\_pw2wan.in**). We are asking the code to calculate the overlap matrices  $M_{mn}$  (that will be written in the **si2.mmn** file) and the  $A_{mn}$  matrices (file **si2.amn**). Since we want to plot the Wannier functions in real space, we need also the  $u_{nk}(r)$  wavefunctions on a real-space grid. We thus also set the **write\_unk** flag in **06\_pw2wan.in**, that will produce a set of files with names **UNK00001.1**, **UNK00002.1**, ... Finally, the code will also produce a **si2.eig** file, with the eigenvalues on the initial  $4 \times 4 \times 4$  k-grid. Note that the **pw2wannier90.x** expects to find the **si2.nnkp** file produced in the previous step. Run the code using

```
mpirun -np 2 pw2wannier90.x < 06_pw2wan.in > pw2wan.out
```

```
&inputpp
 outdir      = 'out/'
  prefix      = 'si'
  seedname    = 'si2'
  write_amn   = .true.
  write_mmn   = .true.
  write_unk   = .true.
/

```

3. Finally we can run Wannier90 to obtain MLWFs. Execute

```
wannier90.x si2
```

and, when it finishes, inspect the output file, called **si2.wout**.

- Before the start of the maximal localisation iterations, there is a section (containing the string `<--DIS`) with the iterations of the disentanglement procedure. It is important that at the end of this section the convergence is achieved (with a string `<< Disentanglement convergence criteria satisfied >>`).
- Check lines containing `<-- DLTA` to check for the convergence of the spread during the maximal localisation iterations.
- Check the lines after the string `Final state`: you find the centers and spreads of the maximally-localised Wannier functions.
- To check if the obtained MLWFs are correct, it is typically needed to:
  - \* *Compare the Wannier-interpolated band structure with the ab-initio one*: the provided Wannier90 input file computes the interpolated band plot; you can try to compare the ab-initio bandplot obtained in the steps before with the interpolated band structure (files `si2_band.dat`, and `si2_band.gnu`)
    - To plot it with `gnuplot`: run `gnuplot` in terminal, and in `gnuplot`, type

```
set xtics nomirr
set x2tics
set x2range [0:0.21721815E+01]
set xrange [0:1.3195]
plot 'bands.dat.gnu' w p pt 7, 'si2_band.dat' axes x2y1 w l
```

Note you can reuse the script in following exercises when comparing band structures, by replacing the file names `qebands.agr` and `si2_band.dat`.

- Or plot it with `xmgrace`: in terminal, type:

```
xmgrace qebands.agr si2_band.dat
```

Note that you may need to rescale the x axis.

The Wannier90 code also outputs in the `si2_band.kpt` file a list of the kpoints used for the interpolation, that could be used to plot the band structure on the same grid.

- \* *Plot the real-space Wannier functions and check if they are real:* if you ask **Wannier90** to plot the Wannier functions, it will print also the ratio of the imaginary and real part of each of them at the end of the **si2.wout** file: check that the value is small.
- \* A practical note: Especially when using disentanglement, it is possible that the disentanglement convergence is not achieved, and/or that the obtained Wannier functions are not real, and/or that the interpolated band structure differs significantly from the ab-initio one within the frozen window. Then, you need to change/tune the number of Wannier functions, the projections you chose and/or the energy values for the frozen and disentanglement windows, until you get “good” Wannier functions.
- Plot one of the Wannier functions, which are output in files **si2\_00001.xsf**, .... To visualize the Wannier functions, you need to install **xcrysden** or **VESTA** in your computer, and download the xsf files:
  - \* using **xcrysden**: open the **xsf** file, then choose **Tools**→**Data Grid**→**OK**, and then choose a reasonable **isovalue**, activate the **Render +/- isovalue** flag, and press **Submit**.
  - \* using **VESTA**: open the **xsf** file, **VESTA** can automatically find a isovalue.
- **Optional (Do it only if you have enough time):** Do the symmetry and the centers of the Wannier functions agree with your intuition? (We would like 4 sp<sup>3</sup>-like orbitals centered on each Si atom, with similar spreads). Try to rerun everything with a 6 × 6 × 6 kgrid for the **nscf** and **Wannier90** step to check if the results improve, and how the spreads change with respect to the grid density? Also try to tune the **dis\_froz\_max** and see the difference of MLWF spreads and band interpolation?

## 2 Lead Fermi surface and band structure

In this exercise we will see how to interpolate the band structure of lead, in particular around the Fermi energy. The first goal is to obtain the Fermi surface from Wannier interpolation. This will clearly show some of the advantages of using Wannier-interpolation schemes with respect to full k-points direct calculations. In order to build a MLWFs model that describes the band structure around the Fermi energy, we need to have an idea of the orbital character of the bands we are interested in. The crystal structure of pure lead has one atom per primitive cell, and since we also want to describe some states above the Fermi Energy, we include five  $d$  orbitals ( $d_{xy}, d_{xz}, d_{yx}, d_{z^2}$  and  $d_{x^2-y^2}$ ), and four  $sp^3$  orbitals. Hence our guess for the projections is

- 5  $d$  orbitals centered on the lead atom
- 4  $sp^3$  orbitals centred on the lead atom

Now we are ready to obtain MLWFs and describe the states of lead around the Fermi-level.

- Directory: `exercise6`
- Input Files
  - `01_scf.in` *The PWSCF input file for the ground state calculation*

```
&control
  calculation='scf'
  restart_mode='from_scratch',
  pseudo_dir = '../files/pseudo/',
  outdir='./out'
  prefix='pb'
/
&system
  ibrav = 2, celldm(1) = 9.3555, nat= 1, ntyp= 1,
  ecutwfc = 47.0, ecutrho = 189,
  occupations='smearing', smearing='cold', degauss=0.02
/
&electrons
  conv_thr = 1.0e-9
  mixing_beta = 0.7
/
ATOMIC_SPECIES
  Pb 207.2 Pb.pbe-dn-kjpaw_psl.0.2.2.UPF
ATOMIC_POSITIONS (crystal)
  Pb 0.0 0.0 0.0
K_POINTS (automatic)
  8 8 8 0 0 0
```

- `04_nscf.in` *The PWSCF input file to obtain Bloch states on a uniform grid*

```
&control
  calculation='nscf'
  pseudo_dir = '../files/pseudo/',
```

```

outdir='./out'
prefix='pb'
/
&system
ibrav = 2, celldm(1) = 9.3555, nat= 1, ntyp= 1,
ecutwfc = 47.0, ecutrho = 189,
occupations='smearing', smearing='cold', degauss=0.02
nosym=.true.,nbnd=13
/
&electrons
conv_thr = 1.0e-9
/
ATOMIC_SPECIES
Pb 207.2 Pb.pbe-dn-kjpaw_psl.0.2.2.UPF
ATOMIC_POSITIONS
Pb 0.0 0.0 0.0
K_POINTS crystal
512
0.00000000 0.00000000 0.00000000 1.953125e-03
0.00000000 0.00000000 0.12500000 1.953125e-03
...

```

- 05\_pw2wan.in *Input file for pw2wannier90*

```

&inputpp
outdir = './out'
prefix = 'pb'
seedname = 'lead'
write_amn = .true.
write_mmn = .true.
/

```

- lead.win *The wannier90 input file*

```

use_ws_distance = .true.

num_bands      = 13
num_wann       = 9
num_iter       = 200

dis_win_max    = 38.0
dis_froz_max   = 16.0
dis_num_iter   = 50
dis_mix_ratio  = 1.0

Begin Kpoint_Path
G 0.00 0.00 0.00 X 0.50 0.50 0.00
X 0.50 0.50 0.00 W 0.50 0.75 0.25
W 0.50 0.75 0.25 L 0.00 0.50 0.00
L 0.00 0.50 0.00 G 0.00 0.00 0.00
G 0.00 0.00 0.00 K 0.00 0.50 -0.50

```

```

End Kpoint_Path

! SYSTEM

begin unit_cell_cart
bohr
-4.67775 0.00000 4.67775
0.00000 4.67775 4.67775
-4.67775 4.67775 0.00000
end unit_cell_cart

begin atoms_frac
Pb 0.00 0.00 0.00
end atoms_frac

begin projections
Pb:d;sp3
end projections

! KPOINTS

mp_grid : 8 8 8

begin kpoints
0.00000000 0.00000000 0.00000000
0.00000000 0.00000000 0.12500000
...

```

1. Run PWSCF to obtain the ground state of lead  
`pw.x < 01_scf.in > scf.out`
2. Run PWSCF to obtain the Bloch states on a uniform k-point grid  
`pw.x < 04_nscf.in > nscf.out`
3. Run wannier90 to generate a list of the required overlaps (written into the `lead.nnkp` file).  
`wannier90.x -pp lead`
4. Run pw2wannier90 to compute the overlap between Bloch states and the projections for the starting guess (written in the `lead.mmn` and `lead.amn` files).  
`pw2wannier90.x < 05_pw2wan.in > pw2wan.out`
5. Run wannier90 to compute the MLWFs.  
`wannier90.x lead`

Inspect the output file `lead.wout`.

1. Use Wannier interpolation to obtain the Fermi surface of lead. Rather than re-running the whole calculation we can use the unitary transformations obtained in the first calculation and restart from the plotting routine. Add the following keywords to the `lead.win` file:

```

restart = plot
fermi_energy = [insert your value here]
fermi_surface_plot = true

```

and re-run `wannier90`. The value of the Fermi energy can be obtained from the output of the initial first principles calculation. `wannier90` calculates the band energies, through wannier interpolation, on a dense mesh of k-points in the Brillouin zone. The density of this grid is controlled by the keyword `fermi_surface_num_points`. The default value is 50 (i.e.,  $50^3$  points). The Fermi surface file `lead.bxsf` can be viewed using `XCrySDen`, e.g.,

```
xcrysden --bxsf lead.bxsf
```

2. Plot the interpolated band structure. A suitable path in k-space is

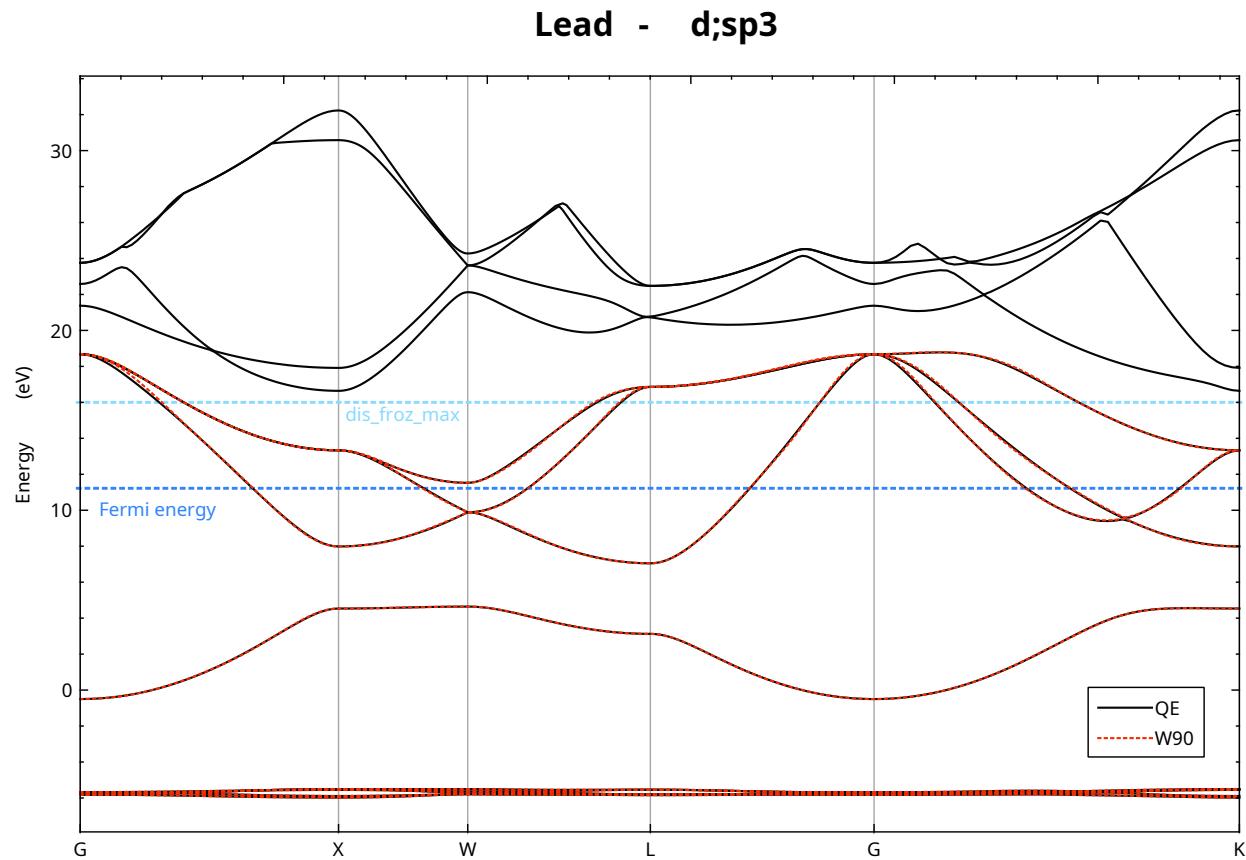
```

begin kpoint_path
G 0.00 0.00 0.00 X 0.50 0.50 0.00
X 0.50 0.50 0.00 W 0.50 0.75 0.25
W 0.50 0.75 0.25 L 0.00 0.50 0.00
L 0.00 0.50 0.00 G 0.00 0.00 0.00
G 0.00 0.00 0.00 K 0.00 0.50 -0.50
end kpoint_path

```

### Further ideas (if you have time)

- Compare the Wannier interpolated band structure with the full PWSCF band structure. Obtain MLWFs using a denser k-point grid. To plot the band structure you can use the PWSCF tool `bands.x`.
- Investigate the effects of the outer and inner energy windows on the interpolated bands.
- Instead of extracting a subspace of  $d+sp^3$  states, we could extract a different nine dimensional space (i.e., with  $s$ ,  $p$  and  $d$  character). Examine this case and compare the interpolated band structures.
- Remove the low-energy  $d$  states from the wannierization (hint: use the `exclude_bands` option in the Wannier90 input file) and compare both the spread and band structure you obtain.



Band Structure of lead showing the position of the Fermi energy and inner energy windows.