ABINIT Hands-On

for *Advanced Experimental Physics*

Matteo Cagnoni (matteo.cagnoni@polito.it)

What is ABINIT

- Software to compute properties of materials
- Density-Functional-Theory (DFT) + perturbation theories + …
- Website: https://www.abinit.org/

How to install ABINIT

- **Quantum Mobile**: https://quantum-mobile.readthedocs.io/en/latest/index.html
  - pre-configured virtual machine (please, use the last version!) for VirtualBox
  - highly recommended and only option supported by the teachers
- **Conda**: https://anaconda.org/
  - package manager for macOS, Windows, Linux
  - ABINIT support for Linux only (and probably for Windows through Windows Linux Subsystem - WLS)
  - Use Conda-Forge version:
    - Add Conda-Forge channel to Conda: conda config --add channels conda-forge
    - Create and activate new Conda environment: conda create –n cf.abinit && conda activate cf.abinit
    - Install ABINIT: conda install -c conda-forge abinit
    - Remember to activate environment (conda activate cf.abinit) when you open a new terminal
- **APT**: sudo apt install abinit
  - built-in package manager of Debian-based Linux distributions
  - available on Windows through Windows Linux Subsystem (WLS)
- **Compilation**: https://docs.abinit.org/tutorial/abinit_build/
  - recommended only for experienced users
  - NOT supported by the teachers

How to run ABINIT

1. Write input text file something.abi where values assigned to (needed) input variables (https://docs.abinit.org/variables/)
2. From terminal, execute abinit something.abi 1> something.log 2> something.err
   - abinit: command to be executed
   - something.abi: path to the input file
   - 1> something.log: save terminal standard output to something.log file (debugging)
   - 2> something.err: save terminal error output to something.err file (debugging)
3. Several output files generated:
   - something.abo: main output file summarizing the data
   - *_GSR.nc, *_EIG.nc, ...: output data files
     see https://docs.abinit.org/guide/abinit/#32-file-names-in-abinit for more info

- To speed up computation, you can run ABINIT with parallel execution. For example, if you want to use 4 cores: mpirun –n 4 abinit something.abi 1> something.log 2> something.err

Useful tools

- Certified pseudo-potentials database: http://www.pseudo-dojo.org/
- Grace: http://plasma-gate.weizmann.ac.il/Grace/
  - band structure and density-of-states plots with the commands xmgrace *.agr and xmgrace *DOS
  - already included in Quantum Mobile, can be installed with sudo apt install grace
- XCrysDen: http://www.xcrysden.org/
  - crystal structure visualization
  - Fermi surface plots with the command xcrysden --bxsf *BSXF
  - already included in Quantum Mobile
- AFLOW: http://www.aflowlib.org/
  - database of material properties (you can download crystal structures from here)
- AbiPy: https://abinit.github.io/abipy/
  - Python library with a lot of post-processing options for ABINIT output data
  - not included in Quantum Mobile, but can be installed in Quantum Mobile with Conda
  - very extensive, takes some time to learn and it is not necessary for the course

Useful resources (1/2)

- These slides are based on the following ABINIT tutorials, contaning much more information, which you can refer to if you are struggling:
  - ground-state calculations, structural relaxation, electronic properties, SOC:
    - https://docs.abinit.org/tutorial/base1/
    - https://docs.abinit.org/tutorial/base2/
    - https://docs.abinit.org/tutorial/base3/
    - https://docs.abinit.org/tutorial/base4/
    - https://docs.abinit.org/tutorial/spin/
  - phonon properties:
    - https://docs.abinit.org/tutorial/rf1/
    - https://docs.abinit.org/tutorial/rf2/
  - electron-phonon coupling:
    - https://docs.abinit.org/tutorial/eph_legacy/

Useful resources (2/2)

- Introductory guide to ABINIT:
  - https://docs.abinit.org/
- Slightly more advanced guide to ABINIT, including explanation of input and output files, and multi-dataset mode (not straightforward, hence reading is recommended):
  - https://docs.abinit.org/guide/abinit/#34-the-multi-dataset-mode
- Explanation of ABINIT and ANADDB variables:
  - https://docs.abinit.org/variables/
- ABINIT course (very advanced):
  - https://school2019.abinit.org/course-material

What are we going to do with ABINIT

- Calculation of the properties of crystalline solids:
  - ground-state calculation
  - structural relaxation
  - electronic properties: band structure, density-of-states (DOS), Fermi surface
  - phonon properties: linear response, inter-atomic force constants, dispersion relation, DOS
  - electron-phonon coupling: matrix elements, superconducting properties

Ground-state calculation

Ground-state (GS) energy

- For a given crystal structure (atomic arrangement):

1. Solve Kohn-Sham eigenvalue problem: $\widehat{H}_{\mathrm{KS}}\ \phi_{n\mathbf{k}}(\mathbf{r}) = \varepsilon_{n\mathbf{k}}\phi_{n\mathbf{k}}(\mathbf{r})$
   - $\widehat{H}_{\mathrm{KS}} = \left[-\frac{\hbar^2\nabla^2}{2m_e} + U_{\mathrm{ext}}(\mathbf{r}) + U_{\mathrm{H}}(\mathbf{r}) + U_{\mathrm{xc}}(\mathbf{r})\right]$
     - $U_{\mathrm{ext}}(\mathbf{r})$: potential energy due to electron-nuclei interaction
     - $U_{\mathrm{H}}(\mathbf{r})$: potential energy due to electron-electron interaction (Hartree potential)
     - $U_{\mathrm{xc}}(\mathbf{r})$: exchange-correlation potential energy
     - $n\mathbf{k}$: band index and electron wave vector (Bloch theorem)

2. Ground-state energy: $E_0 = \sum_{n\mathbf{k}}^{\mathrm{occ}} \varepsilon_{n\mathbf{k}} - \frac{e^2}{2}\int \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}' \frac{n(\mathbf{r})n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} + E_{\mathrm{xc}}[n] - \int \mathrm{d}\mathbf{r}\ n(\mathbf{r})\ U_{\mathrm{xc}}(\mathbf{r})$
   - $n(\mathbf{r}) = \sum_{n\mathbf{k}}^{\mathrm{occ}} |\phi_n(\mathbf{r})|^2$: electron density
   - $E_{\mathrm{xc}}[n]$: exchange-correlation functional ($U_{\mathrm{xc}} = \delta E_{\mathrm{xc}}/\delta n$)

Electron-nuclei potential energy

- Kohn-Sham eigenvalues problem:
$$\left[-\frac{\hbar^2\nabla^2}{2m_e} + U_{\text{ext}}(\mathbf{r}) + U_{\text{H}}(\mathbf{r}) + U_{\text{xc}}(\mathbf{r})\right]\phi_{n\mathbf{k}}(\mathbf{r}) = \varepsilon_{n\mathbf{k}}\phi_{n\mathbf{k}}(\mathbf{r})$$
- $U_{\text{ext}}(\mathbf{r}) = \sum_{\mathbf{R}\alpha} U_\alpha(\mathbf{r}-\mathbf{r}_\alpha-\mathbf{R})$
  - $\mathbf{R} = n_1\mathbf{R}_1 + n_2\mathbf{R}_2 + n_3\mathbf{R}_3$: Bravais lattice translation vector
  - α: label/index of the atom within unit cell
  - $\mathbf{r}_\alpha = x_{\alpha,1}\mathbf{R}_1 + x_{\alpha,2}\mathbf{R}_2 + x_{\alpha,3}\mathbf{R}_3$: atom position within unit cell (fractional coordinates)
  - U_α: atomic (pseudo-)potential

Electron-nuclei potential energy in ABINIT

- Inside ABINIT input file something.abi:

```
################################################################################
# PREAMBLE
################################################################################
outdata_prefix "./scf/scf"                                    # path prefix for output data files

################################################################################
# UNIT CELL
################################################################################
acell 10.18 10.18 10.18                                       # [Bohr] length scales by which dimensionless primitive translation vectors 'rprim' are multiplied
rprim  0.0  0.5  0.5                                          # dimensionless primitive translation vectors
       0.5  0.0  0.5
       0.5  0.5  0.0

################################################################################
# ATOM TYPES
################################################################################
ntypat 1                                                      # number of atom types
znucl 14                                                      # atomic number of the atom types (space-separated)
pp_dirpath "../../../Pseudos/PBE-SR"                          # path to directory containing pseudopotentials (within quotation marks)
pseudos "Si.psp8"                                             # pseudopotential file names (within quotation marks, comma-separated)

################################################################################
# ATOMS
################################################################################
natom 2                                                       # number of atoms in the unit cell
typat 1 1                                                     # type of the atoms within the unit cell (space separated, must be consistent with 'znucl' and 'pseudos' order)
xred  0   0   0                                               # atom fractional coordinates within unit cell
      1/4 1/4 1/4
```

L_i $\mathbf{R}_i/L_i$ $\{\mathbf{R}_1,\mathbf{R}_2,\mathbf{R}_3\}$

Z_i $i=1,\ldots,I$ $I=\text{ntypat}$

U_i $i=1,\ldots,I$ $I=\text{ntypat}$

i_α $\alpha=1,\ldots,\text{A}$ $\text{A}=\text{natom}$ x_α

Here you tell ABINIT that there are 2 atoms in the unit cell.
The first atom is of type 1 ($i_1 = 1$), the second atom is of type 1 ($i_2 = 1$).
Therefore, the first znucl Z_1 and pseudos U_1 are associated to both atoms.

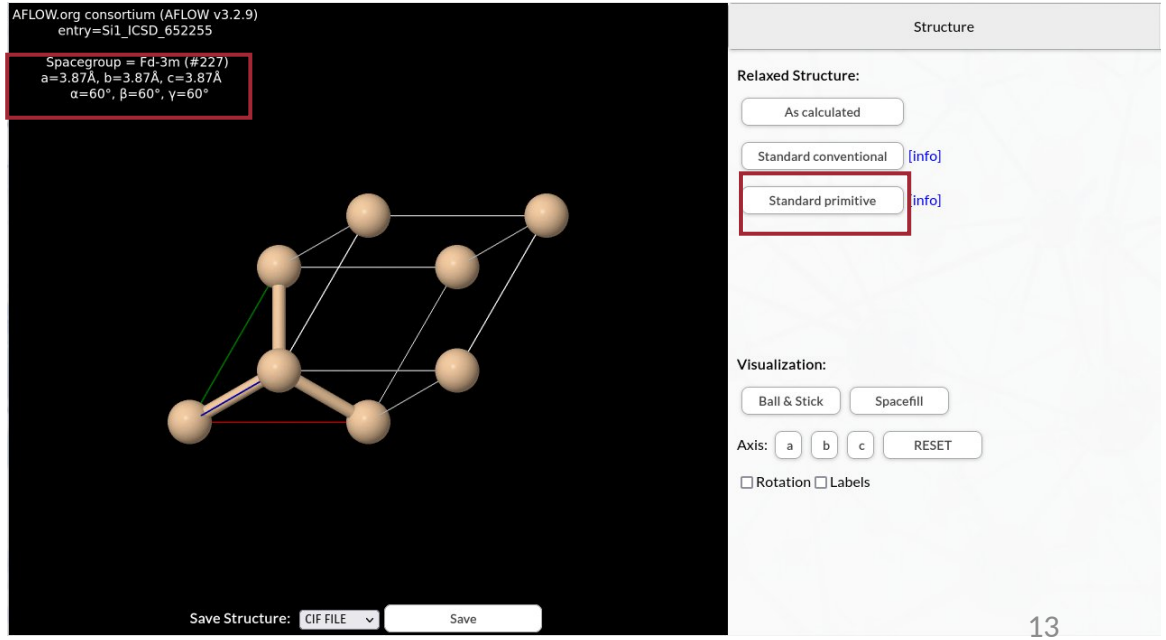
Bravais lattice specification

- Using acell and rprim is not the only way to specify the Bravais lattice underlying your crystal
- An alternative workflow that you might find easier is:
  - Find you material at http://www.aflowlib.org/search/
    (!) Be careful to choose the correct structure (space group)!

- Select “Standard primitive” unit cell
- Specify unit vector lengths and angles in degrees with the variables acell and angdeg

```
acell 7.31 7.31 7.31
angdeg 60 60 60
```

(!) Remember, acell should be specify in Bohr units



Plane-wave expansion (in ABINIT)

- Bloch theorem: $\phi_{n\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{G}} c_{n\mathbf{k}}(\mathbf{G}) \ \exp i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}$
  - $\mathbf{G}$: reciprocal lattice translation vectors
- Plane-wave representation of Kohn-Sham eigenvalue equation to describe problem in matrix form (numerical solution by calculator):

$$\hat{H}_{\mathrm{KS}}\phi_{n\mathbf{k}} = \varepsilon_{n\mathbf{k}}\phi_{n\mathbf{k}}(\mathbf{r}) \leftrightarrow \mathbf{H}_{\mathrm{KS}}(\mathbf{k})\,\mathbf{c}_{n\mathbf{k}} = \varepsilon_{n\mathbf{k}}\mathbf{c}_{n\mathbf{k}}$$

- Finite number of plane waves ($\Rightarrow n = 1,\ldots,n_{\max}$) defined by cut-off energy E_{cut}:

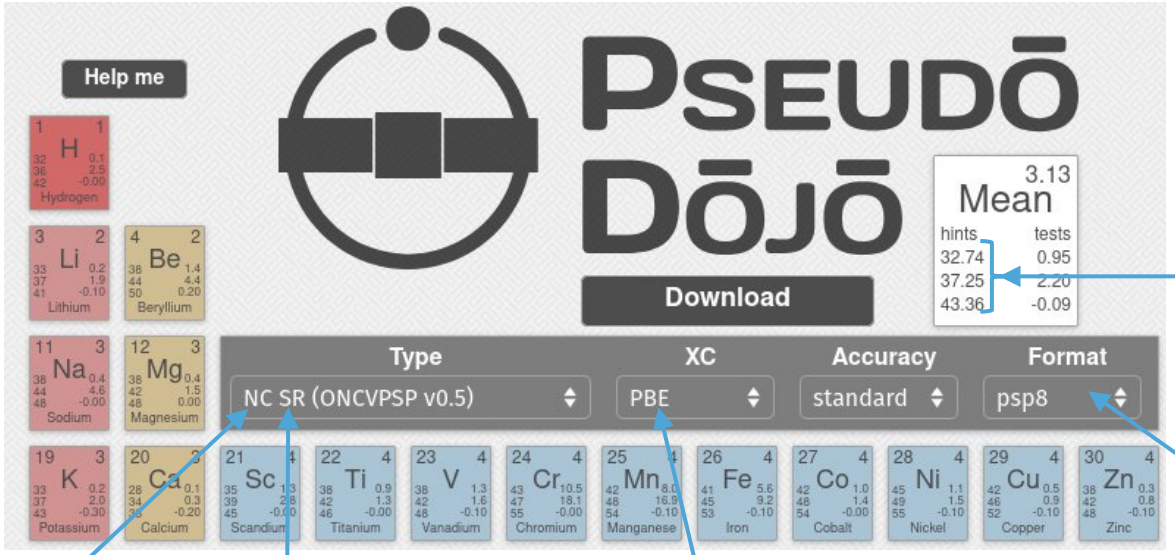
$$T = \frac{\hbar^2}{2m_e}|\mathbf{k}+\mathbf{G}|^2 < E_{\mathrm{cut}}$$

- Inside ABINIT input file something.abi:

```
############################################################
# PLANE-WAVE BASIS SET
############################################################
ecut 12.0                                                  # [Ha] plane-wave cut-off energy
                                                           # (recommended values at http://www.pseudo-dojo.org/)
```

Pseudo-potentials and plane-wave cut-off

- ABINIT “certified” repository of pseudo-potentials: http://www.pseudo-dojo.org/
- Careful to download the appropriate pseudo-potential!



Recommended plane-wave cut-off energy in Hartree, for:

- quick calculation/starting point
- good guess for high-throughput
- high-accuracy

Choose the PSP8 format

Norm-Conserving (do not use PAW!)

Scalar-Relativistic (Full-Relativistic – FR – for Spin-Orbit-Coupling – SOC)

Exchange-Correlation functional used to generate pseudo-potential. Please use PBE (Perdew-Burke-Ernzerhof). https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.77.3865

Exchange-correlation (in ABINIT)

- Kohn-Sham eigenvalues problem:
$$\left[-\frac{\hbar^2\nabla^2}{2m_e} + U_{\text{ext}}(\mathbf{r}) + U_{\text{H}}(\mathbf{r}) + U_{\text{xc}}(\mathbf{r})\right]\phi_{n\mathbf{k}}(\mathbf{r}) = \varepsilon_{n\mathbf{k}}\phi_{n\mathbf{k}}(\mathbf{r})$$
  - $U_{\text{xc}} = \delta E_{\text{xc}}/\delta n$, with $E_{\text{xc}}[n]$: exchange-correlation functional
- ABINIT supports many types of exchange-correlation functionals (input variable ixc)
  - Local-Density-Approximation (LDA): $E_{\text{xc}}[n(\mathbf{r})] = \int \mathrm{d}\mathbf{r}\ \varepsilon_{\text{xc}}(n(\mathbf{r}))\ n(\mathbf{r})$
  - Generalized-Gradient-Approximation (GGA): $E_{\text{xc}}[n(\mathbf{r})] = \int \mathrm{d}\mathbf{r}\ \varepsilon_{\text{xc}}(n(\mathbf{r}), \nabla_{\mathbf{r}} n(\mathbf{r}))\ n(\mathbf{r})$
    * PBE is a type of GGA
- Choice must be consistent with pseudopotentials used!
- ABINIT selects ixc automatically, depending on given pseudopotentials
  → no need to specify variable in input file

k-points grid

- Eigenvalue problem $\mathbf{H}_{\mathrm{KS}}(\mathbf{k})\,\mathbf{c}_{n\mathbf{k}} = \varepsilon_{n\mathbf{k}}\mathbf{c}_{n\mathbf{k}}$ can be solved only of a finite number of **k**-points
- Properties of interest (total energy, charge density, …) from average:

$$\overline{F} = \frac{1}{\mathrm{card}\,(\mathscr{K})}\sum_{k\in\mathscr{K}} F(\mathbf{k})$$

  where $\mathscr{K} \subset \mathrm{BZ}$ and $\mathrm{card}\,(\mathscr{K}) < \infty$.
- Regular grid within Irreducible Brillouin Zone (IBZ):
  1. Monkhorst-Pack grid: https://journals.aps.org/prb/abstract/10.1103/PhysRevB.13.5188
     $\mathbf{k}_{prs} = u_p\mathbf{G}_1 + u_r\mathbf{G}_2 + u_s\mathbf{G}_3$, where $u_i = (2i - q_i - 1)/(2q_i)$, with $\mathrm{i} = 1,\ldots,q_i$ $(i = p,r,s)$
  2. Sets of **k**-points with equal F-value because of symmetry replaced by one $\mathbf{k}$-point weighted according to degeneracy

k-points grid in ABINIT

- Inside ABINIT input file something.abi:

```
################################################################################
# k-POINT GRID (see https://journals.aps.org/prb/abstract/10.1103/PhysRevB.13.5188)
################################################################################
kptopt 1                                                  # method to set-up k-points;
                                                          # 1: symmetry accounted for,
                                                          # so that only k-points
                                                          # within irreducible BZ generated,
                                                          # with appropriate symmetry-defined weights
                                                          # (degeneracy); prevents redundancy
ngkpt 2 2 2                                               # number of points per grid
nshiftk 4                                                 # number of shifted grids to be used concurrently
shiftk 0.5  0.5  0.5                                      # k-grid shift vectors; this combination is very
       0.5  0.0  0.0                                      # efficient for FCC lattice;
       0.0  0.5  0.0                                      # other choices might be more suitable for other
       0.0  0.0  0.5                                      # lattices (see
                                                          # https://docs.abinit.org/variables/basic/#shiftk)
```

algorithm to set-up k points
- > 0: grid + symmetry (average quantities)
- < 0: segments (band structure)

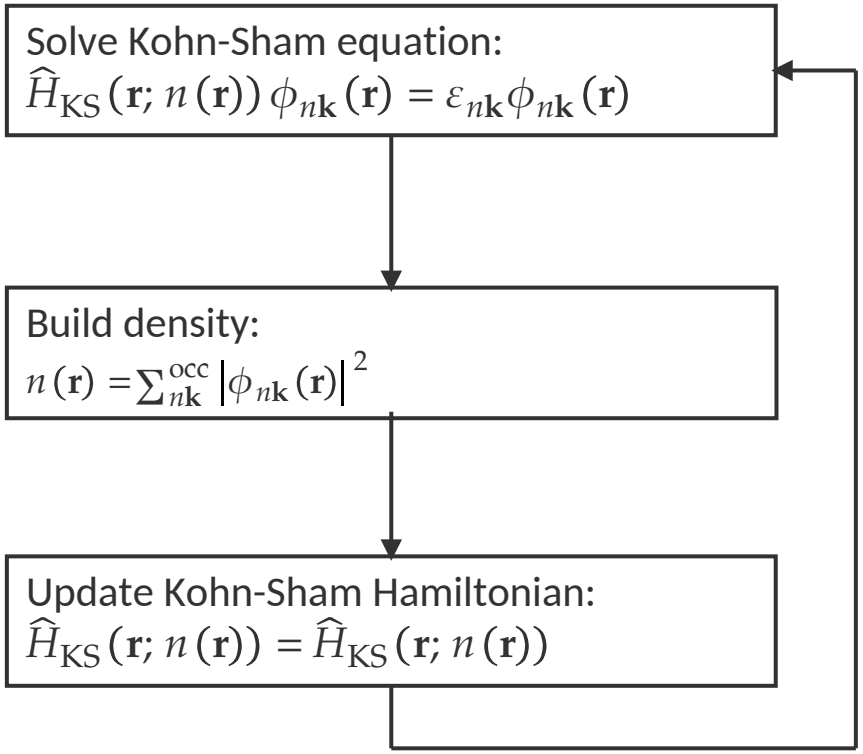
q_p, q_r, q_s

Default:
nshiftk 1
shiftk 0.5 0.5 0.5

To include origin:
nshiftk 1
shiftk 0.0 0.0 0.0

Self-consistency

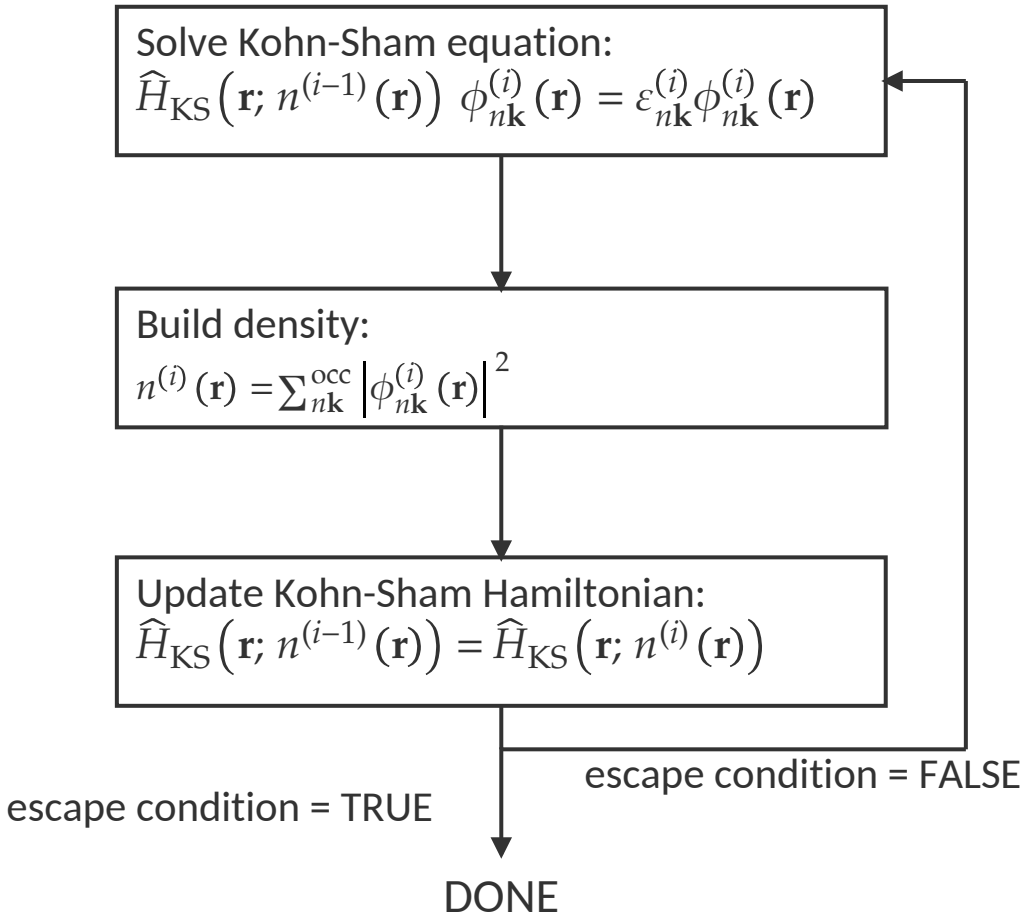
- Kohn-Sham eigenvalue problem:
$$\left[-\frac{\hbar^2\nabla^2}{2m_e} + U_{\text{ext}}(\mathbf{r}) + U_{\text{H}}(\mathbf{r}) + U_{\text{xc}}(\mathbf{r})\right]\phi_{n\mathbf{k}}(\mathbf{r}) = \varepsilon_{n\mathbf{k}}\phi_{n\mathbf{k}}(\mathbf{r})$$
  * specified * something missing…
- $U_{\text{H}}(\mathbf{r}) = U_{\text{H}}(\mathbf{r};n(\mathbf{r})) = \frac{e^2}{2}\int \mathrm{d}\mathbf{r}'\ \frac{n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}$
  $U_{\text{xc}}(\mathbf{r}) = U_{\text{xc}}(\mathbf{r};n(\mathbf{r}))$
  → dependence on solution through
  $$n(\mathbf{r}) = \sum_{n\mathbf{k}}^{\text{occ}} |\phi_{n\mathbf{k}}(\mathbf{r})|^2$$
- Solution must be <u>self-consistent</u>!



Self-consistency in ABINIT (1/2)

- Starting from initial guess $n^{(0)}(\mathbf{r})$, perform self-consistent cycle until escape condition fulfilled:
  - i > max. number of cycles
  - metric < tolerance
- Metric = change of selected quantity from previous step. One among:
  - toldfe: total energy difference
  - toldff: atomic forces difference
  - tolwfr: wave function squared residual
  - tolvrs: potential energy residual

  * see https://docs.abinit.org/topics/SCFControl/



Self-consistency in ABINIT (2/2)

- Inside ABINIT input file something.abi:

```
#################################################
# SCF PROCEDURE
#################################################
nstep 10                                          # max. number of SCF cycles
toldfe 1.0d-6                                     # convergence criterion: absolute energy difference
                                                  # between two consecutive steps
```

Recap: input for GS calculation of c-Si

```
################################################################################
# PREAMBLE
################################################################################
outdata_prefix "./scf/scf"                                          # path prefix for output data files

################################################################################
# UNIT CELL
################################################################################
acell 10.18 10.18 10.18                                             # [Bohr] length scales by which dimensionless primitive translation vectors 'rprim' are multiplied
rprim  0.0  0.5  0.5                                                # dimensionless primitive translation vectors
       0.5  0.0  0.5
       0.5  0.5  0.0

################################################################################
# ATOM TYPES
################################################################################
ntypat 1                                                            # number of atom types
znucl 14                                                            # atomic number of the atom types (space-separated)
pp_dirpath "../../../Pseudos/PBE-SR"                                # path to directory containing pseudopotentials (within quotation marks)
pseudos "Si.psp8"                                                   # pseudopotential file names (within quotation marks, comma-separated)

################################################################################
# ATOMS
################################################################################
natom 2                                                             # number of atoms in the unit cell
typat 1 1                                                           # type of the atoms within the unit cell (space separated, must be consistent with 'znucl' and 'pseudos' order)
xred  0   0   0                                                     # atom fractional coordinates within unit cell
      1/4 1/4 1/4

################################################################################
# PLANE-WAVE BASIS SET
################################################################################
ecut 12.0                                                           # [Ha] plane-wave cut-off energy (recommended values at http://www.pseudo-dojo.org/)

################################################################################
# k-POINT GRID (see https://journals.aps.org/prb/abstract/10.1103/PhysRevB.13.5188)
################################################################################
kptopt 1                                                            # method to set-up k-points;
                                                                    # 1: symmetry accounted for,
                                                                    # so that only k-points within irreducible BZ generated,
                                                                    # with appropriate symmetry-defined weights (degeneracy);
                                                                    # prevents redundancy
ngkpt 2 2 2                                                         # number of points per grid
nshiftk 4                                                           # number of shifted grids to be used concurrently
shiftk 0.5  0.5  0.5                                                # k-grid shift vectors; this combination is very efficient for FCC lattice;
       0.5  0.0  0.0                                                # other choices might be more suitable for other lattices
       0.0  0.5  0.0                                                # (see https://docs.abinit.org/variables/basic/#shiftk)
       0.0  0.0  0.5

################################################################################
# SCF PROCEDURE
################################################################################
nstep 10                                                            # max. number of SCF cycles
toldfe 1.0d-6                                                       # convergence criterion: absolute energy difference between two consecutive steps
```

Ground-state (GS) energy in ABINIT

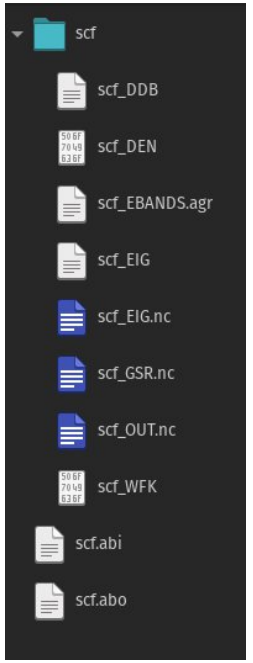
- Text file just discussed instructs ABINIT to execute self-consistent GS calculation
- Open a terminal, go to the directory containing the file, and run:
  abinit something.abi 1> something.log 2> something.err
- Summary output file something.abo with important info saved in same directory:

```
== END DATASET(S) ==============================================================
================================================================================
 -outvars: echo values of variables after computation  --------
           acell      1.0180000000E+01  1.0180000000E+01  1.0180000000E+01 Bohr
             amu      2.80855000E+01
          diemac      1.20000000E+01
            ecut      1.20000000E+01 Hartree
          etotal     -8.4553030645E+00
           fcart      1.0580138401E-27 -6.3480830408E-28 -1.0580138401E-27
                     -1.0580138401E-27  6.3480830408E-28  1.0580138401E-27
```

total energy (Ha)

forces on unit cell atoms (Ha/Bohr)
each row = force vector on one atom

- Output data files stored in the same directory by default. It is “cleaner” to specify outdata_prefix in something.abi. For example, if ./scf/scf given:



- Unless specified otherwise, ABINIT uses Hartree (Ha) for energy and Bohr for length

Matters of convergence: part 1

- Numerical computation requires finite matrices:
  - plane-wave basis set cut-off energy ecut specified by the user
  - number of $\mathbf{k}$-points ngkpt specified by the user
- Issue: results dependent on choice of ecut and ngkpt!
- Solution:
  1. Decide tolerance for total energy change as a function of ecut and ngkpt
  2. Calculate total energy as a function of ecut and ngkpt
  3. Use smallest (fastest) value that satisfies tolerance for subsequent calculations
- Always verify convergence, even if recommended values for ecut provided at PseudoDojo!

Datasets in ABINIT (1/2)

- Multi-dataset mode: https://docs.abinit.org/guide/abinit/#34-the-multi-dataset-mode
  - redo calculations for different values of some parameter, letting all other things equal
  - chains of calculations
- Inside ABINIT input file something.abi:

convergence w.r.t. cut-off energy

```
##########################################
#PREAMBLE
##########################################
outdata_prefix "./etot-vs-ecut/etot-vs-ecut"
ndtset 5

##########################################
# PLANE-WAVE BASIS SET
##########################################
ecut1 8.0
ecut2 10.0
ecut3 12.0
ecut4 14.0
ecut5 16.0
```

number of datasets

ecut values considered
appended integer identifies dataset

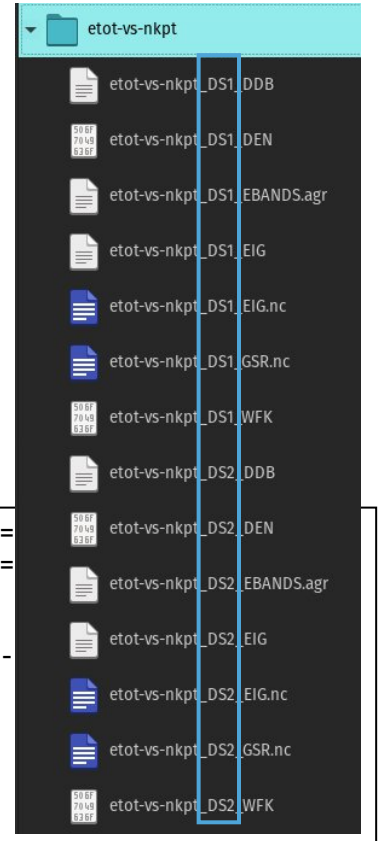
convergence w.r.t. number of **k**-points

```
##########################################
#PREAMBLE
##########################################
outdata_prefix "./etot-vs-nkpt/etot-vs-nkpt“
ndtset 4

##########################################
# k-POINT GRID
##########################################
kptopt 1
ngkpt1 2 2 2
ngkpt2 4 4 4
ngkpt3 6 6 6
ngkpt4 8 8 8
nshiftk 4
shiftk 0.5 0.5 0.5
       0.5 0.0 0.0
       0.0 0.5 0.0
       0.0 0.0 0.5
```

ngkpt values considered
appended integer identifies dataset

Datasets in ABINIT (2/2)

- Dataset-dependent results in something.abo + output files for each dataset:



convergence w.r.t. cut-off energy

```
== END DATASET(S) ==============================================
================================================================


 -outvars: echo values of variables after computation  --------
            acell      1.0180000000E+01  1.0180000000E+01  1.01
              amu      2.80855000E+01
           diemac      1.20000000E+01
             ecut1     8.00000000E+00 Hartree
             ecut2     1.00000000E+01 Hartree
             ecut3     1.20000000E+01 Hartree
             ecut4     1.40000000E+01 Hartree
             ecut5     1.60000000E+01 Hartree
           etotal1    -8.4519427804E+00
           etotal2    -8.4546405786E+00
           etotal3    -8.4553030645E+00
           etotal4    -8.4554489181E+00
           etotal5    -8.4555749123E+00
            fcart1    -1.6928221442E-27 -5.0784664327E-27  5.0784664327E-27
                       1.6928221442E-27  5.0784664327E-27 -5.0784664327E-27
            fcart2     5.2900692007E-29 -5.2900692007E-29 -0.0000000000E+00
                      -5.2900692007E-29  5.2900692007E-29 -0.0000000000E+00
            fcart3     1.0580138401E-27 -6.3480830408E-28 -1.0580138401E-27
                      -1.0580138401E-27  6.3480830408E-28  1.0580138401E-27
            fcart4     5.2900692007E-28 -3.1740415204E-28 -1.0580138401E-28
                      -5.2900692007E-28  3.1740415204E-28  1.0580138401E-28
            fcart5    -2.1160276803E-28  0.0000000000E+00  2.1160276803E-28
                       2.1160276803E-28  0.0000000000E+00 -2.1160276803E-28
```

convergence w.r.t. number of **k**-points

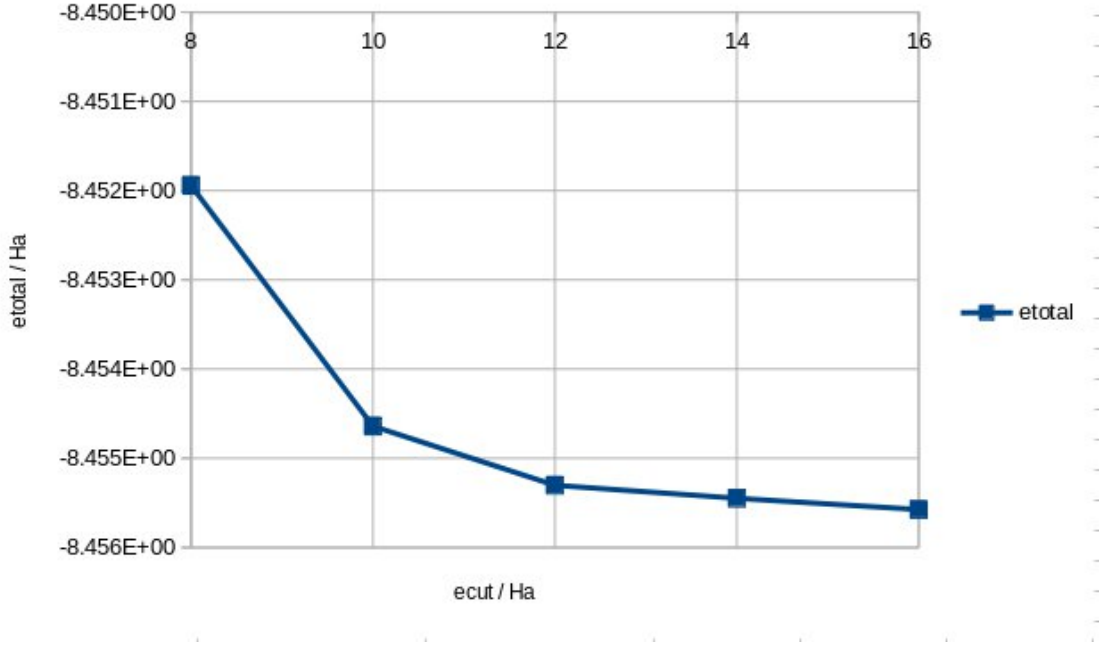
```
== END DATASET(S) ==============================================
================================================================


-outvars: echo values of variables after computation  -----
            acell      1.0180000000E+01  1.0180000000E+01
              amu      2.80855000E+01
           diemac      1.20000000E+01
             ecut      1.20000000E+01 Hartree
           etotal1    -8.4553030645E+00
           etotal2    -8.4612905266E+00
           etotal3    -8.4613855371E+00
           etotal4    -8.4613885783E+00
            fcart1     1.0580138401E-27 -6.3480830408E-28 -1.0580138401E-27
                      -1.0580138401E-27  6.3480830408E-28  1.0580138401E-27
            fcart2    -4.2320553606E-28  0.0000000000E+00  4.2320553606E-28
                       4.2320553606E-28  0.0000000000E+00 -4.2320553606E-28
            fcart3    -0.0000000000E+00 -0.0000000000E+00 -0.0000000000E+00
                      -0.0000000000E+00 -0.0000000000E+00 -0.0000000000E+00
            fcart4     0.0000000000E+00 -2.6450346003E-29  1.0580138401E-28
                       0.0000000000E+00  2.6450346003E-29 -1.0580138401E-28
              kpt1    -2.50000000E-01  5.00000000E-01  0.00000000E+00
                      -2.50000000E-01  0.00000000E+00  0.00000000E+00
              kpt2    -1.25000000E-01 -2.50000000E-01  0.00000000E+00
                      -1.25000000E-01  5.00000000E-01  0.00000000E+00
                      -2.50000000E-01 -3.75000000E-01  0.00000000E+00
                      -1.25000000E-01 -3.75000000E-01  1.25000000E-01
                      -1.25000000E-01  2.50000000E-01  0.00000000E+00
```

Matters of convergence: part 2

- Example. If I decide that convergence achieved when total energy change < 0.001 Ha

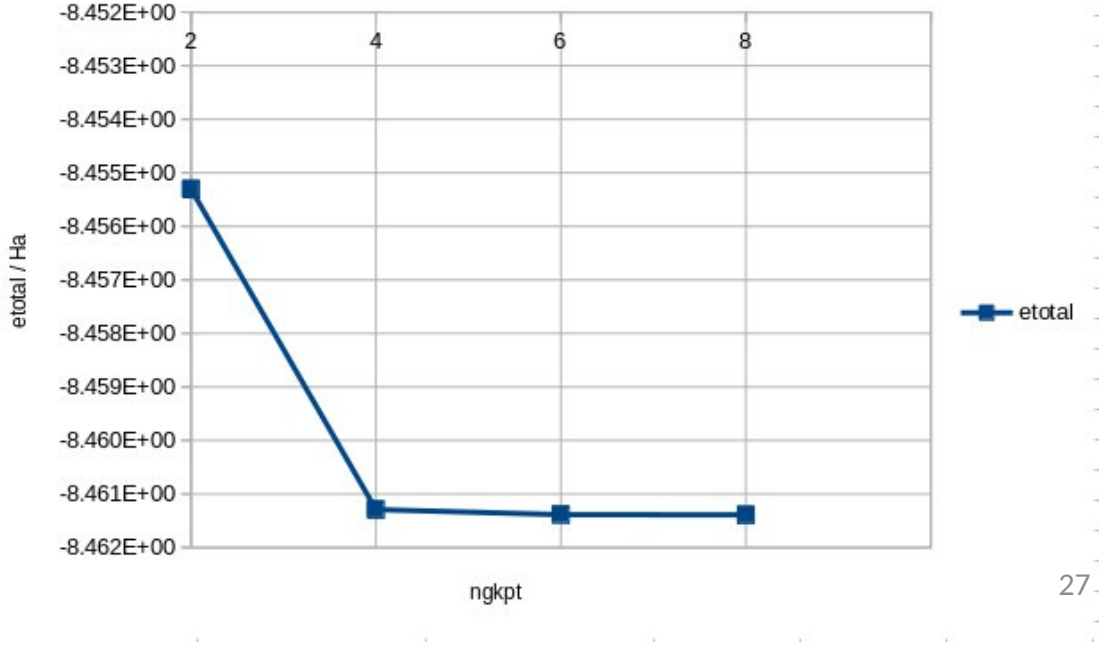
convergence w.r.t. cut-off energy

ecut	etotal	Δetotal
8	-8.452E+00	
10	-8.455E+00	-0.003
12	-8.455E+00	-0.001
14	-8.455E+00	0.000
16	-8.456E+00	0.000



convergence w.r.t. number of **k**-points

ngkpt	etotal	Δetotal
2	-8.455E+00	
4	-8.461E+00	-0.006
6	-8.461E+00	0.000
8	-8.461E+00	0.000



How to deal with metals

- Issue: finite BZ sampling makes convergence difficult for metals.
  Ex. For $\mathbf{k}$-points close to Fermi surface, highest occupied bands can enter or exit averaging sums from one iteration step to the next of the self-consistent cycle → numerical instability.
- Solution: smear Fermi surface by introducing distribution of occupation numbers.
- How:
  1. Broaden Kohn-Sham energy levels by smearing function:
     $\delta_\sigma(\varepsilon) = \frac{1}{\sigma}\,\tilde{\delta}\left(\frac{\varepsilon}{\sigma}\right)$, where $\tilde{\delta}$: smearing function, with σ: smearing parameter
  2. Local density of states becomes: $n(\mathbf{r},\varepsilon) = \sum_{n\mathbf{k}}^{\text{occ}} \frac{1}{\sigma}\tilde{\delta}\left(\frac{\varepsilon-\varepsilon_{n\mathbf{k}}}{\sigma}\right)\left|\phi_{n\mathbf{k}}(\mathbf{r})\right|^2$
  3. Electron density becomes:
     $n(\mathbf{r}) = \sum_{n\mathbf{k}}^{\text{occ}} \tilde{\theta}\left(\frac{\varepsilon_{\text{F}}-\varepsilon_{n\mathbf{k}}}{\sigma}\right)\left|\phi_{n\mathbf{k}}(\mathbf{r})\right|^2$, with $\tilde{\theta}(x) = \int_{-\infty}^{x} \mathrm{d}y\;\tilde{\delta}(y)$ and #electrons $= \sum_{n\mathbf{k}}^{\text{occ}} \tilde{\theta}\left(\frac{\varepsilon_{\text{F}}-\varepsilon_{n\mathbf{k}}}{\sigma}\right)$
- Advantage: local density-of-states can now be computed accurately on finite grid if energy levels separation small with respect to σ

How to deal with metals in ABINIT

- Inside ABINIT input file something.abi:

```
################################################################################
# OCCUPATION
################################################################################
occopt 4                                # controls how occupation of electronic states is handled
                                        # (see https://docs.abinit.org/variables/basic/#occopt)
                                        # 4: metallic occupation with "cold smearing" of Marzari as smearing function
                                        # smearing improves convergence w.r.t. to BZ sampling in metals,
                                        # where state occupation drops suddenly to zero as Fermi level crossed
                                        # otherwise, a very fine needed for accurate integration around the discontinuity
tsmear 0.05                             # [Ha] "temperature" of smearing (sigma)
                                        # convergence should be checked also w.r.t. this parameter
                                        # see http://theossrv1.epfl.ch/Main/ElectronicTemperature
```

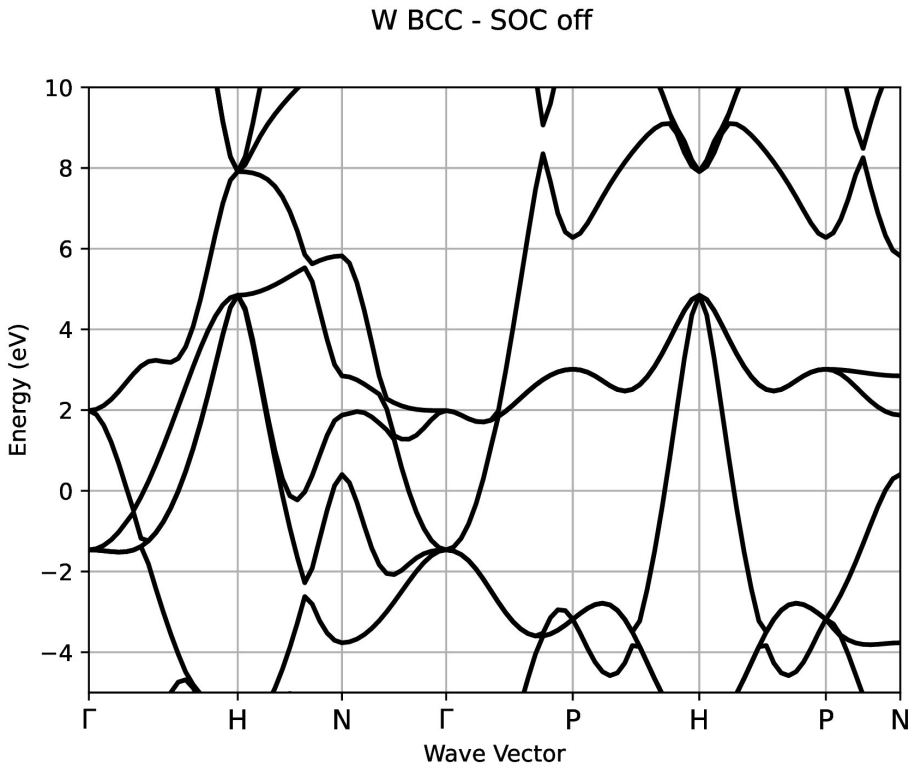
type of $\tilde{\delta}$

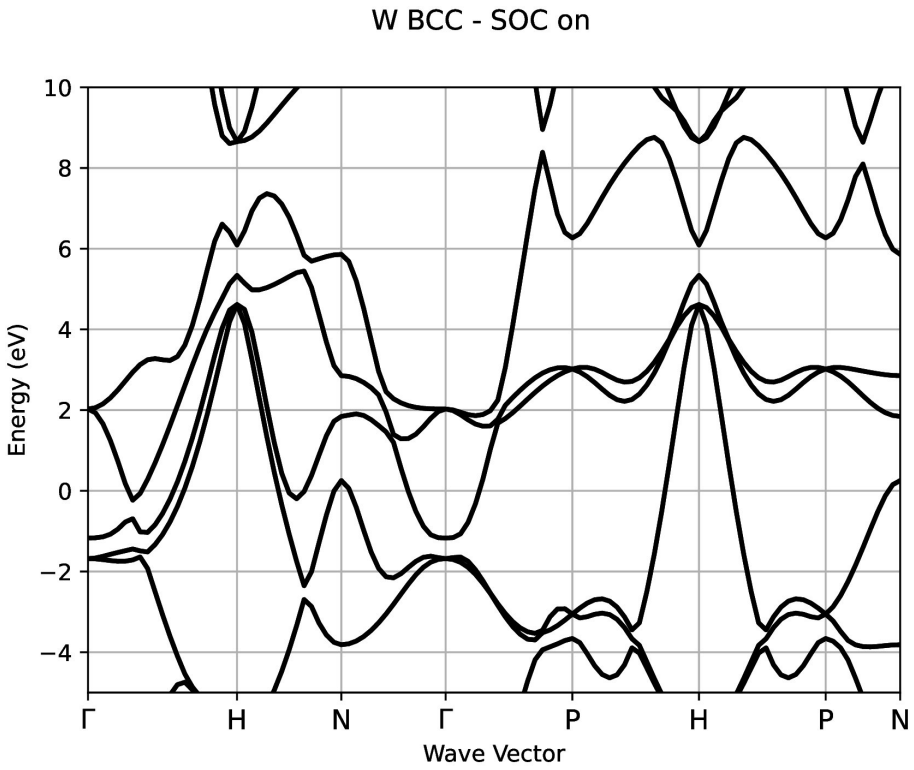
σ value

- Smearing is important only for metals (e.g., aluminum) and doped semiconductors.
- You should check convergence of ground-state energy also with respect to tsmear, to set the appropriate value, as we have seen before for ecut and ngkpt!
- **Remark.** Do not use a too large value, otherwise the results will be converged but wrong. Suggestion: start from small (e.g., 0.001) value and slightly increase up to convergence.
- **Remark.** Convergence with respect to ecut, ngkpt and tsmear is not independent. If you want to be rigorous, you should check several combinations of these parameters.

Spin-Orbit-Coupling (SOC) in ABINIT

- Inside ABINIT input file something.abi:
  - set nspinor = 2 (number of wavefunctions spinorial components – see https://docs.abinit.org/variables/gstate/#nspinor)
  - use Full Relativistic (FR) pseudo-potentials from PseudoDojo
  - double the value of nband to calculate the same number of states as for the case without SOC





Structural relaxation

Crystal structure relaxation

- Issue: crystal structure provided by the user is never the equilibrium one, i.e. atomic forces $\mathbf{F}_I = -\partial_{\mathbf{R}_I} E(\mathbf{R}) \neq 0$
- Solution: vary
  - primitive translation vectors (unit cell) $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3$
  - atomic coordinates within unit cell (basis vectors) $\{\mathbf{x}_\alpha\}_\alpha$

  until $\mathbf{F}_I = -\partial_{\mathbf{R}_I} E(\mathbf{R}) = 0$

Crystal structure relaxation in ABINIT (1/2)

- Approach:
  1. Self-consistent cycle on user-provided structure
  2. Update structure according to specified algorithms
  3. Self-consistent cycle on updated structure
  4. Repeat 2-3 until stopping criterion fulfilled
- Inside something.abi:

Stopping criteria:

- ntime: maximum number of iterations
- tolmxf: maximal change of atomic forces

  Default: 5e-5 Ha/Bohr. Specify in input file only if you want a different value.

```
#########################################
# UNIT CELL RELAXATION PROCEDURE
#########################################
optcell 1                                 # optimization type;
                                          # 1: optimize only volume, 2: full optimization of cell geometry
                                          # (see https://docs.abinit.org/variables/rlx/#optcell)
ionmov  2                                 # algorithm to change displacement of atoms;
                                          # 0: fixed, 1: molecular dynamics with viscous damping,
                                          # 2: Broyden-Fletcher-Goldfarb-Shanno minimization (BFGS)
                                          # (see https://docs.abinit.org/variables/rlx/#ionmov)
ntime  10                                 # maximum number of relaxation steps
dilatmx 1.05                              # maximum allowed dilatation of unit cell
                                          # needed to book additional memory:
                                          # increasing lattice size decreases reciprocal primitive vectors lengths,
                                          # which lead to more plane waves at constant energy cut-off
ecutsm  0.5                               # [Ha] smearing needed to remove "numerical" discontinuities
                                          # introduced by change of number of plane waves when unit cell size changes
                                          # (see https://journals.aps.org/prb/abstract/10.1103/PhysRevB.93.205147)
```

optimization algorithm for cell vectors

optimization algorithm for basis vectors

```
#################
# SCF PROCEDURE
#################
nstep 10
toldfe 1.0d-6
getwfk -1
```

Initial guess from previous relaxation step to speed up computation

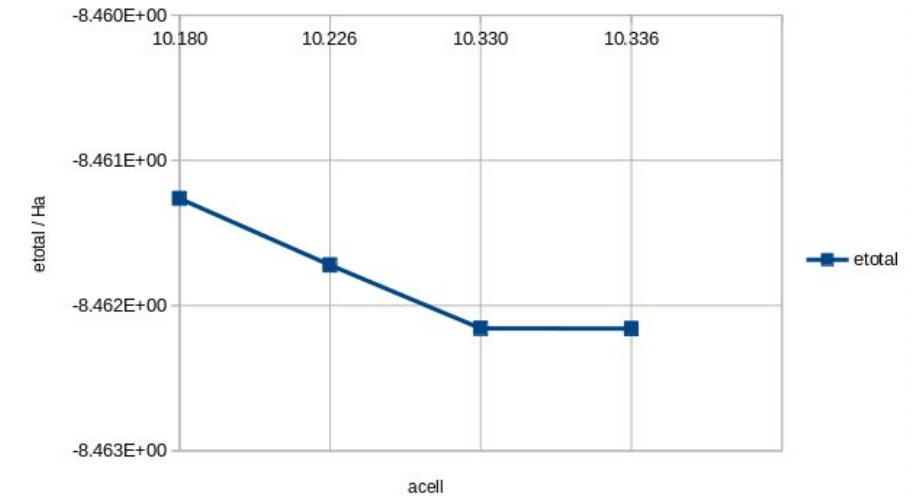
Crystal structure relaxation in ABINIT (2/2)

- Step-dependent results in something.abo:

```
--- Iteration: ( 3/10) Internal Cycle: (1/1)        <-- relaxation step
-------------------------------------------------------------------
---SELF-CONSISTENT-FIELD CONVERGENCE-------------------------------
--- !BeginCycle
iteration_state: {dtset: 1, itime: 3, icycle: 1, }
solver: {iscf: 7, nstep: 10, nline: 4, wfoptalg: 0, }
tolerances: {toldfe: 1.00E-06, }
...
     iter   Etot(hartree)      deltaE(h)  residm     vres2
 ETOT  1  -8.4620285139361    -8.462E+00 1.717E-06 2.899E-01
 ETOT  2  -8.4621521640577    -1.237E-04 3.473E-07 1.766E-02
 ETOT  3  -8.4621572711599    -5.107E-06 3.417E-07 5.050E-04
 ETOT  4  -8.4621573579434    -8.678E-08 7.893E-08 1.988E-06
 ETOT  5  -8.4621573582186    -2.752E-10 8.103E-09 1.330E-08

 At SCF step    5, etot is converged :
  for the second time, diff in etot=  2.752E-10 < toldfe=  1.000E-06
```

self-consistent cycle within relaxation step

acell / Bohr	etotal	Δetotal
10.180	-8.461E+00	
10.226	-8.462E+00	-4.59E-04
10.330	-8.462E+00	-4.36E-04
10.336	-8.462E+00	-1.49E-06



- Equilibrium structure at bottom in something.abo:

```
== END DATASET(S) ==============================================================
================================================================================
 -outvars: echo values of variables after computation  --------
           acell      1.0335899940E+01  1.0335899940E+01  1.0335899940E+01 Bohr
                                                                  L_i
              amu      2.80855000E+01
           diemac      1.20000000E+01
          dilatmx      1.05000000E+00
             ecut      1.20000000E+01 Hartree
           ecutsm      5.00000000E-01 Hartree
           etotal     -8.4621588530E+00
                                                                  F_I
           fcart      2.0841109057E-28 -0.0000000000E+00 -2.0841109057E-28
                     -2.0841109057E-28 -0.0000000000E+00  2.0841109057E-28
```

$\mathbf{R}_i/L_i$

```
 rprim      0.0000000000E+00  5.0000000000E-01  5.0000000000E-01
            5.0000000000E-01  0.0000000000E+00  5.0000000000E-01
            5.0000000000E-01  5.0000000000E-01  0.0000000000E+00
```

x_α

```
 xred     -6.7047150131E-28  3.1200116404E-28  3.3657634175E-28
           2.5000000000E-01  2.5000000000E-01  2.5000000000E-01
```

Electronic properties

Band structure (Kohn-Sham)

- Goal: obtain ε vs $n\mathbf{k}$ (including empty states) along selected high-symmetry lines of the BZ by solving eigenvalue equation $\widehat{H}_{\mathrm{KS}}\ \phi_{n\mathbf{k}}(\mathbf{r}) = \varepsilon_{n\mathbf{k}}\phi_{n\mathbf{k}}(\mathbf{r})$
- Catch: ground-state calculation of electron density and total energy only need occupied states and a “coarse” grid of k-points to converge
- Approach:
  1. Ground-state calculation without (redundant) empty states that would only slow down the self-consistent cycle and “coarse” grid of k-points
  2. Non-self-consistent calculation including empty states on a fine $\mathbf{k}$-sampling of the high-symmetry lines with electron density from previous self-consistent step

Band structure (Kohn-Sham) in ABINIT (1/4)

- Inside ABINIT input file something.abi:

```
#############################################################################
# PREAMBLE
#############################################################################
outdata_prefix "./estruct/estruct"                    # path prefix for output data files
ndtset 2                                              # number of datasets
                                                      # 1: SCF procedure to get GS electron density
                                                      # 2: one for band structure calculation
```

remember: you can use multi-dataset mode to chain calculations

Dataset 1: self-consistent GS calculation of electron density

```
#############################################################################
# k-POINT GRID (see https://journals.aps.org/prb/abstract/10.1103/PhysRevB.13.5188) for SCF procedure (dataset 1)
#############################################################################
kptopt1 1                                             # method to set-up k-points; 1: symmetry accounted for,
                                                      # so that only k-points within irreducible BZ generated,
                                                      # with appropriate symmetry-defined weights (degeneracy);
                                                      # prevents redundancy
ngkpt1 4 4 4                                          # number of grid points
nshiftk1 4                                            # number of shifted grids to be used concurrently
shiftk1 0.5 0.5 0.5                                   # k-grid shift vectors;
        0.5 0.0 0.0                                   # this combination is very efficient for FCC lattice
        0.0 0.5 0.0                                   # (see https://docs.abinit.org/variables/basic/#shiftk)
        0.0 0.0 0.5

#############################################################################
# SCF PROCEDURE
#############################################################################
nstep1 20                                             # max. number of SCF cycles
toldfe1 1.0d-6                                        # convergence criterion:
                                                      # absolute energy difference between two consecutive steps
prtden1 1                                             # save density from SCF procedure (dataset1),
                                                      # so that it can be used in the non-self-consistent
                                                      # calculation of band structure (dataset 2)
```

k-grid for dataset 1 (GS calculation)

save electron-density from dataset 1 so that can be used for dataset 2

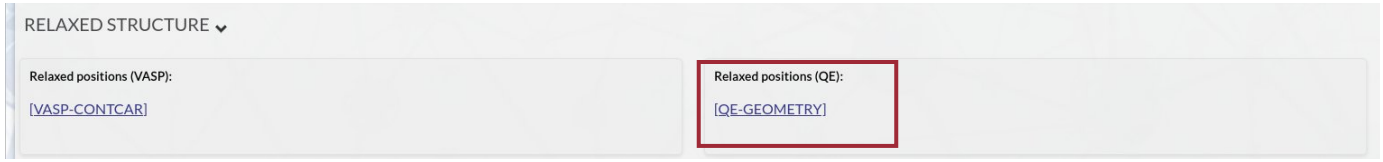
Band structure (Kohn-Sham) in ABINIT (2/4)

Dataset 2: non-self-consistent calculation of KS band structure

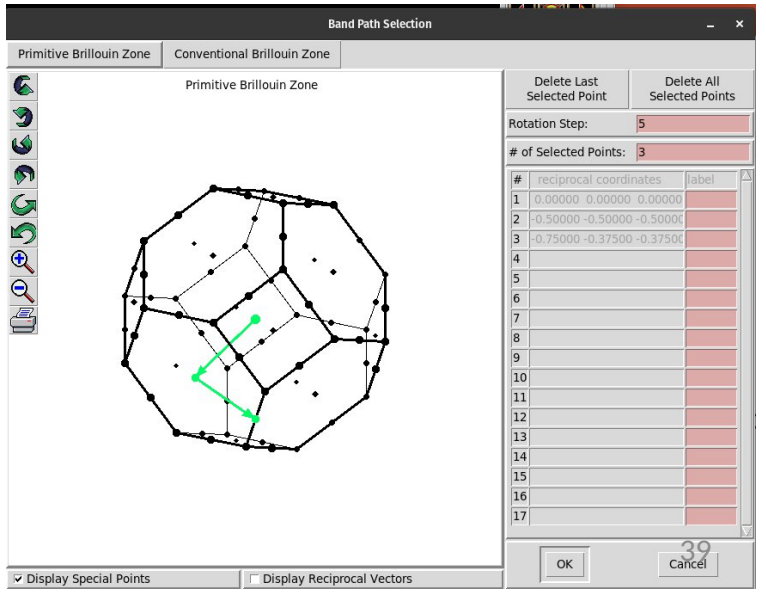
```
#########################################################################
# BAND STRUCTURE
#########################################################################
iscf2 -2                                # self-consistency of the calculation;
                                        # 0: SCF cycle (default),
                                        # -2: non-SCF calculation
getden2 -1                              # get density from previous step
kptopt2 -6                              # when negative, k-points set up
                                        # from bounds and number of sub-divisions
                                        # of segments in BZ;
                                        # absolute value equal to the number of segments
nband2 8                                # number of bands
                                        # (if unsure, look at 'nband' in output file of SCF cycle
                                        # and add some more to include empty states)
ndivsm2 10                              # number of sub-divisions of each segment
kptbounds2  0    0    0                 # bounds of band structure segments. Gamma point
            1/2  0    1/2               # X point
            1/2  1/4  3/4               # W point
            1/2  1/2  1/2               # L point
            0    0    0                 # Gamma point
            3/8  3/8  3/4               # K point
            1/2  0    1/2               # X point
tolwfr2  1.0d-12                        # convergence criterion: tolerance on wavefunction squared residual
enunit2  1                              # output energies in eV instead of Ha
```

High-symmetry points specification

- A list of the high-symmetry points associated to each type of Bravais lattice can be found in the article: https://doi.org/10.1016/j.commatsci.2010.05.010
- Alternatively, you can:
  - Download the crystal structure QE file from AFLOW:



  - Open the file with XCrysDen: xcrysden --pwi CONTCAR.relax.qe
  - Go to Tools > k-Path selection
  - Select your k-path by clicking on the available high-symmetry points
  - Copy the reciprocal coordinates into your ABINIT input file



Recap: input for GS calculation followed by band-structure of c-Si

```
#####################################
# PREAMBLE
#####################################
outdata_prefix "./estruct/estruct"
ndtset 2


#####################################
# UNIT CELL
#####################################
acell 10.336 10.336 10.336
rprim 0.0 0.5 0.5
      0.5 0.0 0.5
      0.5 0.5 0.0
#####################################
# ATOM TYPE
#####################################
ntypat 1
znucl 14
pp_dirpath "../../../Pseudos/PBE-SR"
pseudos "Si.psp8“


#####################################
# ATOMS
#####################################
natom 2
typat 1 1 1
xred 0   0   0
     1/4 1/4 1/4

#####################################
# PLANE-WAVE BASIS SET
#####################################
ecut 12.0
```

```
#####################################
# k-POINT GRID
#####################################
kptopt1 1









ngkpt1 4 4 4
nshiftk1 4
shiftk1 0.5 0.5 0.5
        0.5 0.0 0.0
        0.0 0.5 0.0
        0.0 0.0 0.5
#####################################
# SCF PROCEDURE
#####################################
nstep1 20
toldfe1 1.0d-6


prtden1 1
```

```
#####################################
# BAND STRUCTURE
#####################################
iscf2 -2



getden2 -1
kptopt2 -6







nband2 8



ndivsm2 10
kptbounds2  0    0    0
            1/2  0    1/2
            1/2  1/4  3/4
            1/2  1/2  1/2
            0    0    0
            3/8  3/8  3/4
            1/2  0    1/2
tolwfr2  1.0d-12
enunit2  1
```

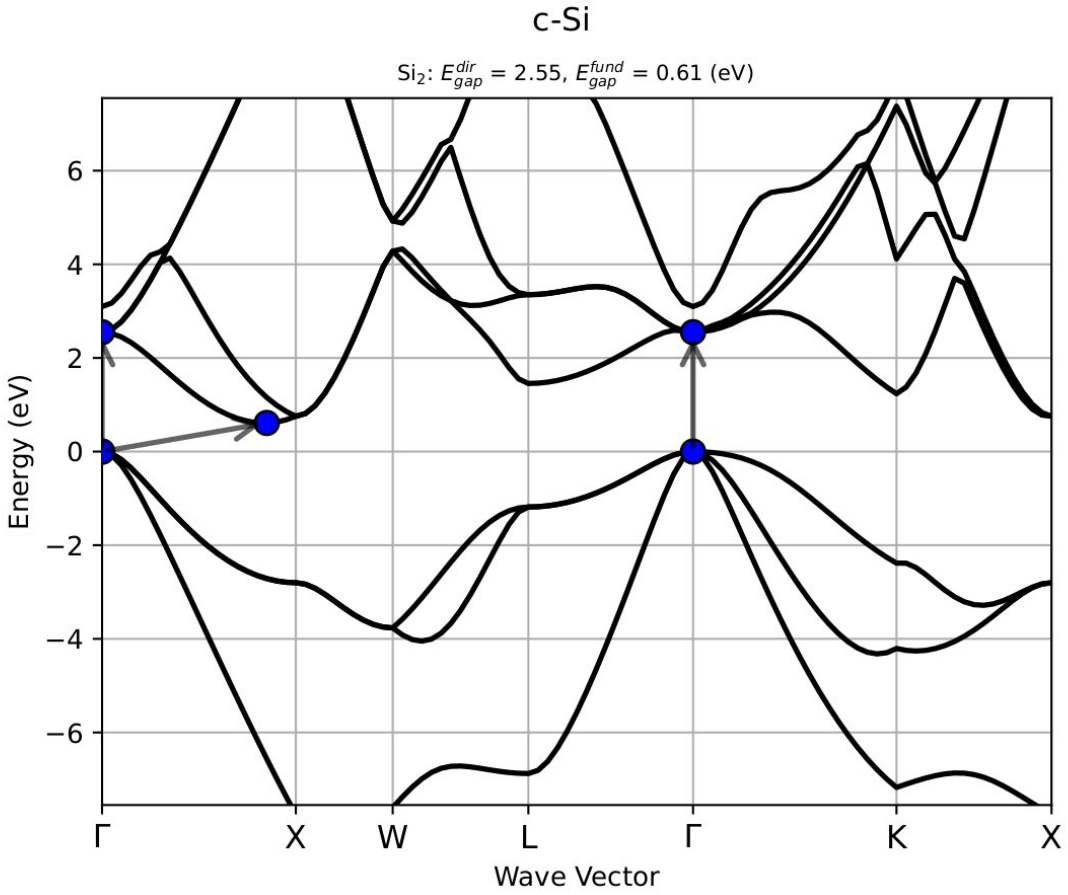
Band structure (Kohn-Sham) in ABINIT (3/4)

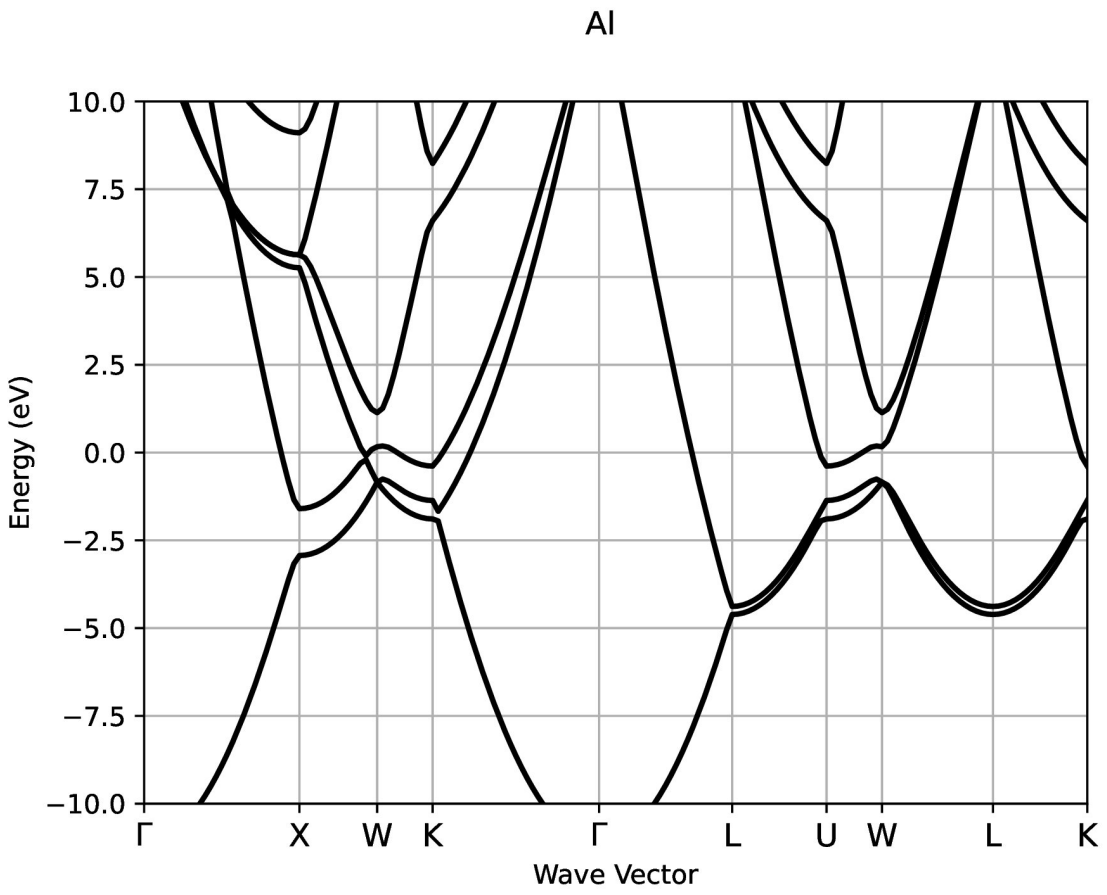
- Electronic states can be found inside ABINIT output file something.abo, under under "Eigenvalues (eV) for nkpt=[...]":

```
Eigenvalues (   eV  ) for nkpt=  99  k points:
 kpt#   1, nband=  8, wtk=  1.00000, kpt=  0.0000  0.0000  0.0000 (reduced coord)
  -7.65220    4.16904    4.16904    4.16904    6.71902    6.71902    6.71902    7.26628
 kpt#   2, nband=  8, wtk=  1.00000, kpt=  0.0250  0.0000  0.0250 (reduced coord)
  -7.64146    4.09112    4.12327    4.12327    6.69248    6.77808    6.77808    7.34253
 kpt#   3, nband=  8, wtk=  1.00000, kpt=  0.0500  0.0000  0.0500 (reduced coord)
  -7.60924    3.87793    3.99753    3.99753    6.61599    6.94365    6.94365    7.54778
 kpt#   4, nband=  8, wtk=  1.00000, kpt=  0.0750  0.0000  0.0750 (reduced coord)
  -7.55560    3.57129    3.81680    3.81680    6.49775    7.19015    7.19015    7.82965
 kpt#   5, nband=  8, wtk=  1.00000, kpt=  0.1000  0.0000  0.1000 (reduced coord)
  -7.48061    3.20738    3.60509    3.60509    6.34869    7.49388    7.49388    8.12688
 kpt#   6, nband=  8, wtk=  1.00000, kpt=  0.1250  0.0000  0.1250 (reduced coord)
  -7.38437    2.80964    3.37936    3.37936    6.17975    7.83734    7.83734    8.36244
```

- The same information is stored in the *_DS2_EBANDS.agr output data file, that can be plotted directly with xmgrace (http://plasma-gate.weizmann.ac.il/Grace/)

Band structure (Kohn-Sham) in ABINIT (4/4)





Density-Of-States (Kohn-Sham)

- Goal: obtain Density-Of-States (DOS) vs ε (including empty states) by solving eigenvalue equation $\widehat{H}_{\mathrm{KS}}\ \phi_{n\mathbf{k}}(\mathbf{r}) = \varepsilon_{n\mathbf{k}}\phi_{n\mathbf{k}}(\mathbf{r})$
- Catch: ground-state calculation of electron density and total energy only need occupied states and a “coarse” grid of k-points to converge, but we are also interested in empty states and need a fine grid to get reasonable DOS values!
- Approach:
  1. Ground-state calculation without (redundant) empty states that would only slow down the self-consistent cycle and “coarse” grid of k-points
  2. Non-self-consistent calculation including empty states on a fine grid of the Brillouin Zone

Density-Of-States (Kohn-Sham) in ABINIT (1/3)

- The approach is the same as the one used for the band structure: a self-consistent GS calculation on a “coarse” BZ grid (dataset 1) followed by a non-self-consistent calculation on a much finer grid and including empty states (dataset 2)
- Inside ABINIT input file something.abi, the following settings are introduced for dataset 2:

```
#########################################################################
# DENSITY-OF-STATES AND FERMI SURFACE (DATASET 2)
#########################################################################
ngkpt2 30 30 30                                  # number of grid points for DOS calculation (dataset 2);
nshiftk2 1                                       # many more points needed to have a fine sampling of BZ;
shiftk2 0.0 0.0 0.0
iscf2 -3                                         # self-consistency of the calculation;
                                                 # 0: SCF cycle,
                                                 # -2: non-SCF calculation
                                                 # (electron density from previous dataset - SCF cycle);
                                                 # -3: like -2, but initialize occupation
                                                 # (see https://docs.abinit.org/variables/basic/#iscf)
getden2 -1                                       # get density from previous step
tolwfr2 1.0d-12                                  # tolerance on wavefunction squared residual
nband2 6                                         # number of bands (if unsure, look at 'nband'
                                                 # in output file of SCF cycle and add some more)
enunit2 1                                        # output energies in eV instead of Ha
prtdos2 2                                        # save density-of-states;
                                                 # 2: use the tetrahedron method to calculate it
                                                 # (see https://docs.abinit.org/variables/files/#prtdos)
prtfsurf2 1                                      # save data for Fermi surface plot in BXSF format
                                                 # (readable by XCrysDen)
```

-3 instead of -2: non-self-consistent calculation + initialize occupation of energy levels

electron density from self-consistent GS calculation

save density-of-states

save Fermi surface

Recap: input for GS calculation followed by density-of-states and Fermi surface of aluminum

```
#########################################
# PREAMBLE
#########################################
outdata_prefix "./edos-fermi/edos-fermi"
ndtset 2

#########################################
# UNIT CELL
#########################################
acell 7.626 7.626 7.626
rprim 0.0 0.5 0.5
      0.5 0.0 0.5
      0.5 0.5 0.0

#########################################
# ATOM TYPES
#########################################
ntypat 1
znucl 13
pp_dirpath "../../../Pseudos/PBE-SR"
pseudos "Al.psp8"

#########################################
# ATOMS
#########################################
natom 1
typat 1
xred 0.0 0.0 0.0

#########################################
# PLANE-WAVE BASIS SET
#########################################
ecut 14.0
```

```
#############################
# OCCUPATION
#############################
occopt 4
```

smearing settings
(Al is a metal!)

```
tsmear 0.04



#############################
# k-POINT GRID (DATASET 1)
#############################
kptopt 1
ngkpt1 6 6 6
nshiftk1 4
shiftk1 0.5 0.5 0.5
       0.5 0.0 0.0
       0.0 0.5 0.0
       0.0 0.0 0.5

#############################
# SCF PROCEDURE (DATASET 1)
#############################
nstep1 20
toldfe1 1.0d-6
diemac1 1e6
prtden1   1
```

same for both datasets
(no integer appended)

save electron-density from dataset 1 so that can be used for dataset 2

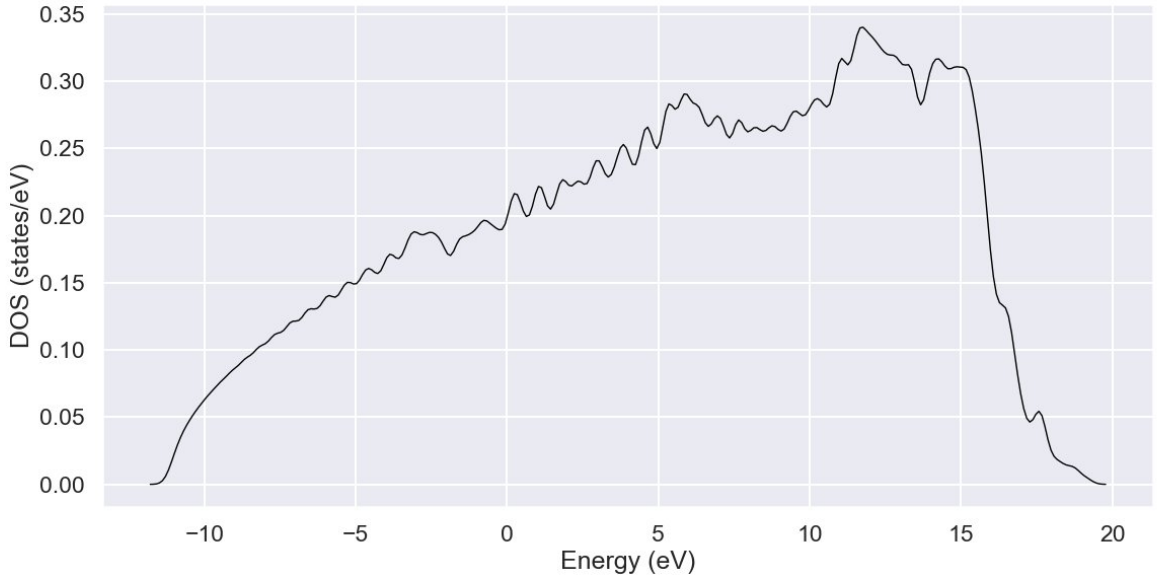
```
#####################################
# DENSITY-OF-STATES AND FERMI SURFACE
# (DATASET 2)
#####################################
ngkpt2 30 30 30
nshiftk2 1
shiftk2 0.0 0.0 0.0
iscf2 -3




getden2 -1
tolwfr2 1.0d-12
nband2 6

enunit2 1
prtdos2 2


prtfsurf2 1
```

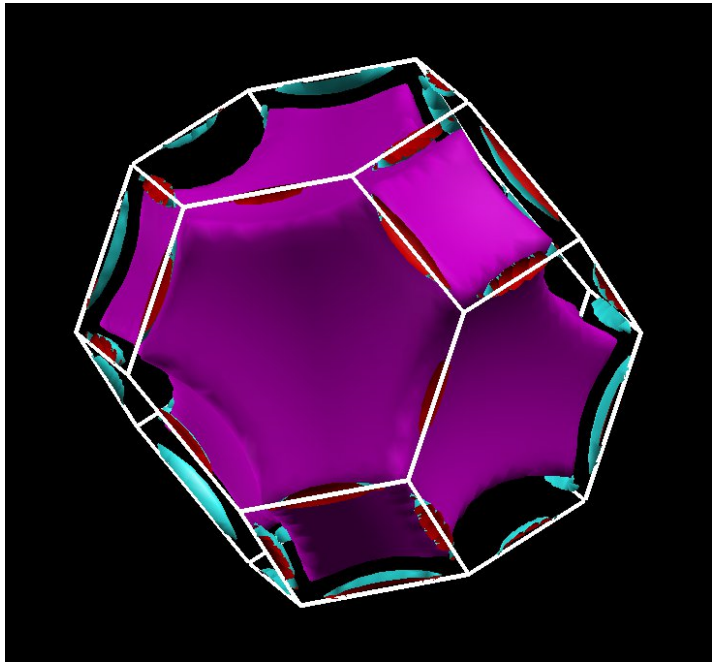
Density-Of-States (Kohn-Sham) in ABINIT (2/3)

- If prtdos=2, *_DS2_DOS output data file generated, which contains density-of-states calculated with the tetrahedron method and can be plot with the command xmgrace *_DS2_DOS
- If prtfsurf=1, *_DS2_BXSF output data file generated, which contains Fermi surface and can be directly plotted with XCrysDen () with the command xcrysden --bxsf *_DS2_BXSF

Density-Of-States (Kohn-Sham) in ABINIT (3/3)

- Aluminum:





Doping in ABINIT

- If you want to add or remove electrons, introduce the variable cellcharge in your input file.
- cellcharge specifies the unit cell charge in number of elementary charges.
- For example:
  - cellcharge 1 means one electron missing per unit cell.
  - cellcharge -1 means one electron added per unit cell.
- cellcharge specifies the average value. Also rational numbers can be used, meaning that one electron is added/removed only from a subset of the unit cells.

Phonon properties

Phonons with DFPT

- Phonon eigenvalues (ω) computed by solving secular equation:

$$\det\left|\frac{1}{\sqrt{M_s M_t}}\, C_{st}^{\alpha\beta}(\mathbf{q}) - \omega^2(\mathbf{q})\right| = 0$$

- Dynamical matrix given by:

$$C_{st}^{\alpha\beta}(\mathbf{q}) = \frac{1}{N_c}\frac{\partial^2 E}{\partial u_s^*(\mathbf{q})\ \partial u_t(\mathbf{q})}$$

- 2nd derivatives related to electron density by:

- Linear response can be obtained by Density-Functional-Perturbation-Theory (DFPT)

Phonons with DFPT in ABINIT (1/8)

- Approach:
  1. Ground-state calculation (dataset 1) to get electron density $n(\mathbf{r})$
  2. Calculation of linear response and dynamical matrices on a grid of $\mathbf{q}$-vectors
     - ABINIT expects one and only one $\mathbf{q}$-vector per dataset
     - $\mathbf{q}$-vectors must be a subset of the k-vectors grid used in the GS calculation
  3. ABINIT will create a DDB (Derivative Data Base) file per $\mathbf{q}$-vector that must be merged into a single file with command line utility MRGDDB
  4. Interatomic-Force-Constants (IFC) $C_{st}^{\alpha\beta}(\mathbf{R}) = \frac{1}{N_c}\sum_{\mathbf{q}} C_{st}^{\alpha\beta}(\mathbf{q})\, e^{i\,\mathbf{q}\cdot\mathbf{R}}$ obtained with command line utility ANADDB
  5. Phonon eigenvalues for arbitrary $\mathbf{q}$-vectors (e.g., phonon band structure and DOS) obtained by Fourier interpolation with command line utility ANADDB

Phonons with DFPT in ABINIT (2/8)

- Steps 1-2: In ABINIT input file response.abi:

```
#########################################################################
# PREAMBLE
#########################################################################
outdata_prefix "./1-response/response"                # path prefix for output data files
ndtset 9                                              # number of datasets:
                                                      # 1: initial GS SCF calculation, 2-9: response to phonons
                                                      # the number of irreducible q-points has been checked
                                                      # with previous dummy run
```

- ABINIT will only retain the irreducible q-points, i.e. a subset of the q-point grid specified by the user
- However, number of datasets must be 1 (GS calculation) + number of q-points (1 dataset per q point)
- How to know ndtset before-hand?
- Do a dummy GS calculation with k-grid = desired q-grid and read the number of irreducible k-points in the output *.abo file

```
------------------------------------------------------------------------------
------------ Echo of variables that govern the present computation ------------
------------------------------------------------------------------------------

...

-outvars: echo values of preprocessed input variables --------
            acell      7.6260000000E+00  7.6260000000E+00  7.6260000000E+00 Bohr
              amu      2.69815390E+01
             ecut      1.40000000E+01 Hartree

...

nkpt           8
```

8 points

Phonons with DFPT in ABINIT (3/8)

```
#########################################################
# PHONON-RELATED DEFAULT SETTINGS
#########################################################
qptopt 1                                                # q-point grid generation
ngqpt 4 4 4                                             # same meanings as k-point related vars
nshiftq 1
shiftq 0 0 0

nqpt 1                                                  # read q-point (ABINIT considers one q point per dataset)
rfphon 1                                                # calculate response function to q-phonon
getwfk 1                                                # use wavefunction from dataset 1 (GS calculation)
tolvrs 1.0d-8                                           # convergence based on residual of potential

kptopt 3                                                # do not account for symmetry when generating k-vectors
                                                        # (needed for phonon calculations; overridden in dataset 1)

# specify which q-point from the grid to use for each dataset,
# by index in the list of q-points
iqpt: 0                                                 # start q-point: iqpt=0 => q = 0 0 0
iqpt+ 1                                                 # increment iqpt by 1 at each dataset
                                                        # iqpt=1 => q = 0 0 0
                                                        # at last dataset (9), iqpt=8 (max. index)
```

used for each dataset (no appended integer) unless overridden

q-point grid (same meaning as analogous k-point grid parameters)

read q-points? 0: no, 1: yes

calculate linear response? 0: no, 1: yes

get wavefunction from dataset 1 (GS calculation)

k-points in full BZ, not just IBZ (needed for DFPT)

ABINIT accepts only ONE q-point per dataset. We need to tell ABINIT which q-point to read from the generated grid at each dataset. iqpt=i with i>0 means “read the i-th q-point”. iqpt=0 means q=(0,0,0). iqpt: 0 iqpt+ 1 means start from 0 and increase value by 1 at each dataset. By starting from 0, we do not read from the q-point grid for dataset 1 (GS calculation), and we read all q-points one-by-one for the next datasets (see https://docs.abinit.org/guide/abinit/#34-the-multi-dataset-mode)

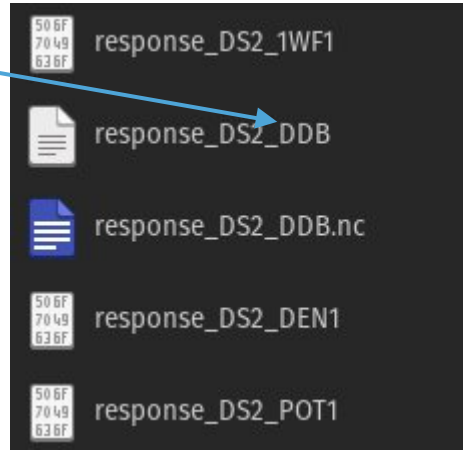
Phonons with DFPT in ABINIT (4/8)

```
#########################################################
# DATASET 1: GROUND-STATE CALCULATION - OVERRIDE DEFAULT SETTINGS ABOVE AND BELOW
#########################################################
getwfk1 0                                    # no wavefunction from previous calculation
nqpt1 0                                      # do not read q-point
rfphon1 0                                    # do not apply phonon perturbation
tolvrs1 1.0d-18                              # SCF convergence criterion: residual of the potential
                                             # (see https://docs.abinit.org/variables/basic/#tolvrs)
kptopt1 1                                    # override set-up of k-points to take symmetry into account
                                             # OK for GS, must be 3 for response calculation
```

Disable phonon-specific settings for GS calculation (dataset 1)

Everything else stays the same.

- One Derivative Data Base (*DDB) output data file generated per dataset, containing 2nd derivatives at the bottom:



q-vector

```
 2nd derivatives (non-stat.)  - # elements :      21
 qpt  0.00000000E+00  0.00000000E+00  0.00000000E+00   1.0
   1   1   1   1  0.25461565428637D+00  0.00000000000000D+00
   2   1   1   1  0.12730782714318D+00  0.00000000000000D+00
   3   1   1   1  0.12730782714318D+00  0.00000000000000D+00
   2   3   1   1  0.00000000000000D+00  0.00000000000000D+00
   3   3   1   1  0.00000000000000D+00  0.00000000000000D+00
```

Perturbations multi-index

Real and imaginary part of the 2nd derivative

Phonons with DFPT in ABINIT (5/8)

- Step 3: Merge the DDB files into a single database by running the command mrgddb < mrgddb.abi. Important: < is used because mrgddb.abi is not really an input file that is parsed by mrgddb. mrgddb is interactive: it asks a question and wait for the user input. (try to simply run mrgddb without anything after). By using <, we supply each line of the abi file as "answer" to the expected mrgddb "question". Therefore, the order of the lines must be exactly as below, and no comments are allowed, since they would be interpreted as input. They are only shown below to describe the abi file.

```
2-mrgddb.abo                                    # merged DDB filename
Al responses                                    # description
8                                               # number of DDBs to be merged
1-response/response_DS3_DDB                     # path to DDBs to be merged
1-response/response_DS4_DDB
1-response/response_DS5_DDB
1-response/response_DS6_DDB
1-response/response_DS7_DDB
1-response/response_DS8_DDB
1-response/response_DS9_DDB
1-response/response_DS10_DDB
```

Phonons with DFPT in ABINIT (6/8)

- Step 4-5: Calculate IFCs and interpolate phonon band structure and DOS with the command anaddb anaddb.abi, with the ANADDB input file anaddb.abi written as below:

```
#########################################################
# PREAMBLE
#########################################################
output_file = "3-anaddb.abo"                    # output file
ddb_filepath = "2-mrgddb.abo"                   # path to Derivative DataBase (DDB)
gkk_filepath = "dummy1"                         # path to GKK matrix elements database (only for e-ph)
eph_prefix = "dummy2"                           # prefix for e-ph output data (only for e-ph)
ddk_filepath = "dummy3"                         # not used here, but must be written
outdata_prefix "./3-anaddb/anaddb"              # output data prefix

#########################################################
# FLAGS
#########################################################
ifcflag 1     Activate IFC Fourier interpolation # calculate interatomic force constants and
                                                # use them to interpolate phonon spectrum
                                                # and dynamical matrices at every q-vector

#########################################################
# WAVE VECTOR (COARSE) GRID FROM CALCULATED DDB BASED ON WHICH IFCs CALCULATED
#########################################################
ngqpt 4 4 4                                     # Monkhorst-Pack grid to consider
                                                # (must correspond to DDB data or a subset of it)
nqshft 1                                        # number of q-shifts
q1shft 0 0 0                                    # shift vectors of coarse grid
```

Files management

(sub-)grid of DDB q-vectors based on which IFCs are calculated

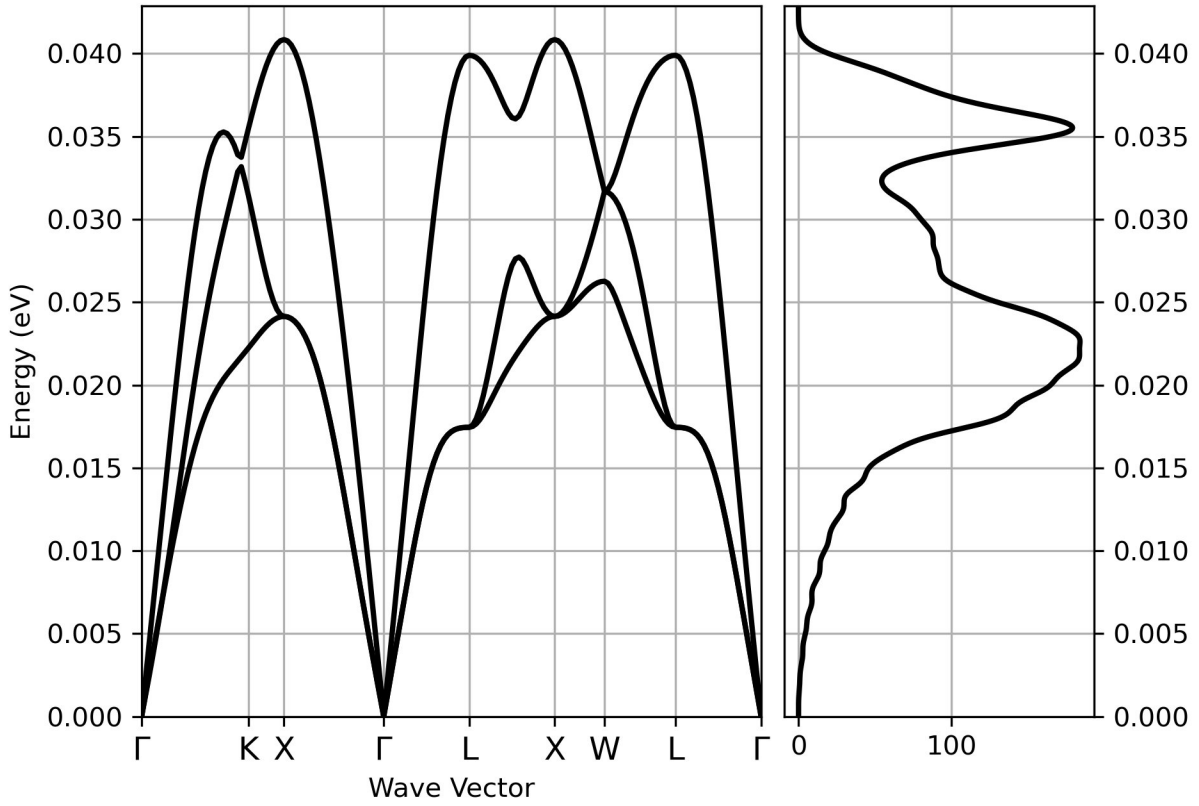
Phonons with DFPT in ABINIT (7/8)

```
##########################################################################
# BAND STRUCTURE CALCUATION BY FOURIER INTERPOLATION ON Q-VECTOR SEGMENTS
##########################################################################
nqpath 9                                                    # number of interpolation q-segment end-points
ndivsm 10                                                   # meaning of variables analogous to k-paths
                                                            # same segment not allowed twice (change cell)
qpath 0.0000  0.0000  0.0000                                # Gamma point
      0.3750  0.3750  0.7500                                # K point
      0.5000  0.5000  1.0000                                # X point
      1.0000  1.0000  1.0000                                # Gamma point
      0.5000  0.5000  0.5000                                # L point
      0.5000  0.0000  0.5000                                # X point
      0.5000  0.2500  0.7500                                # W point
      0.5000  0.5000  0.5000                                # L point
      0.0000  0.0000  0.0000                                # Gamma point

##########################################################################
# DOS CALCULATION BY FOURIER INTERPOLATION ON FINE GRID
##########################################################################
prtdos 1                                                    # calculate DOS?
                                                            # 0: no
                                                            # 1: by gaussian method
                                                            # 2: by tetrahedron method
dossmear 2e-5                                               # [Ha]
ng2qpt 30 30 30                                             # q-vector fine grid for DOS calculation
q2shft 0 0 0
```

Analogous to k-path definition

Phonons with DFPT in ABINIT (8/8)

- A _PHBANDS.agr file, containing the band structure information, that can be readily plotted with xmgrace, and a *_PHDOS file, containing the DOS information.
- For example, one gets for aluminum:



Electron-phonon coupling

Electron-phonon superconductivity

- Superconductor critical temperature:

$$T_c = \frac{\omega_{\log}}{1.2} \exp\left\{ -\frac{1.04(1+\lambda)}{\lambda\,(1-0.62\mu^*) - \mu^*} \right\}$$

- μ^*: Morel-Anderson pseudo-potential, typically treated empirically (value 0.1-0.13)
- λ and $\omega_{\log}$ computed from Eliashberg function $\alpha^2 F\,(\omega)$
- Eliashberg function $\alpha^2 F\,(\omega)$ calculated from electron-phonon matrix elements $g^{\nu}_{n\mathbf{k},\mathbf{k}+m\mathbf{q}}$, which describe the strength of the interaction between ions and electrons

e-ph. superconductivity in ABINIT (1/6)

- Approach very similar to phonon calculation (differences highlighted in red):
  1. Ground-state calculation (dataset 1) to get electron density $n(\mathbf{r})$
     - To be performed on full BZ and not only irreducible BZ (kptopt=3)
     - To get good e-ph matrix elements, you need fine k-grid (let's say 32x32x32) and tight tolerance (tolwfr=1e-20) (increase also max. number of SCF stesp (nstep=1000)
  2. Calculation of linear response, dynamical matrices, and e-ph matrix elements on a grid of $\mathbf{q}$-vectors
     - ABINIT expects one and only one $\mathbf{q}$-vector per dataset
     - $\mathbf{q}$-vectors must be a subset of the k-vectors grid used in the GS calculation
  3. ABINIT will create a DDB (Derivative Data Base) file and a GKK file (e-ph matrix element database) per $\mathbf{q}$-vector that must be merged into a single DDB and GKK file with command line utility MRGDDB and MRGGKK
  4. Superconducting properties obtained by analyzing GKK data with ANADDB

e-ph. superconductivity in ABINIT (2/6)

- Steps 1-2: In ABINIT input file response.abi:
  - Add the following variables for phonon response datasets:

```
# e-ph related default settings
prepgkk 1                               # do not reduce perturbations by symmetry
use_nonscf_gkk 0                        # calculate all GKK matrix elements self-consistently
                                        # not only the irreducible ones
prtgkk 1                                # print out e-ph matrix elements
prtwf 0                                 # do not print out wave function (save disk space)
```

  - Impose full BZ GS calculation, fine k-grid and tight wavefunction tolerance, and print out wavefunction for subsequent datasets:

```
kptopt 3        <-- Must be same grid as perturbations k-grid
ngkpt 32 32 32  <-- Fine grid needed for e-ph coupling

tolwfr1 1.0d-20

prtwf1 1
```

```
# do not account for symmetry when generating k-vectors
# number of grid points
# number of k vectors must be multiple of number of q vectors
# SCF convergence criterion: residual of the wavefunction
# a very tight value is needed to get good e-ph matrix elements
# print wavefunction for subsequent datasets
```

  - Allows more self-consistent cycles by setting nstep=1000
  - Everything else stays the same as for phonons!

e-ph. superconductivity in ABINIT (3/6)

- Steps 3: merge DDB files with mrgddb < mrgddb.abi and GKK files with mrggkk < mrggkk.abi. Example mrggkk.abi below (same "rules" as for mrgddb - remember, comments should NOT be included and are shown below just as explanation!).

```
3-mrggkk.abo                        # Name of output file
0                                   # output file format; 0: binary, 1: ascii (binary needed for ANADDB)
1-response/response_DS1_WFK         # ground-state wave function file
0 435 435                           # number of WF1 files (set to 0), of GKKs, and tot. num. of perts. (sum all GKKs)
1-response/response_DS2_GKK1        # name of 1WF and GKK files
1-response/response_DS2_GKK2
1-response/response_DS2_GKK3
1-response/response_DS3_GKK1
1-response/response_DS3_GKK2
1-response/response_DS3_GKK3
1-response/response_DS4_GKK1
1-response/response_DS4_GKK2
1-response/response_DS4_GKK3
```

Total number of perturbation.
In this case, each GKK file contains one q-vector. Hence, the number is the same as the one of GKK files.
If we would merge two merged GKK files, we would need to specify the total number of q-vectors inside the GKK files.

- I have used a 16x16x16 q-grid → Computationally demanding!

e-ph. superconductivity in ABINIT (4/6)

- Step 4: Calculate superconducting properties with command anaddb anaddb.abi, with the ANADDB input file anaddb.abi written as below:

```
#########################################################################
# PREAMBLE
#########################################################################
output_file = "4-anaddb.abo"
ddb_filepath = "2-mrgddb.abo"
gkk_filepath = "3-mrggkk.abo"
eph_prefix = "./4-anaddb/anaddb"
ddk_filepath = "dummy"
outdata_prefix "./4-anaddb/anaddb"


#########################################################################
# FLAGS
#########################################################################
ifcflag 1                                                    # calculate interatomic force constants and
                                                             # use them to interpolate phonon linewidths
                                                             # and dynamical matrices at every q-vector
elphflag 1                                                   # analyze e-ph coupling

#########################################################################
# WAVE VECTOR (COARSE) GRID FROM CALCULATED DDB BASED ON WHICH IFCs CALCULATED
#########################################################################
brav 2                                                       # 1: all the lattices (FCC included, but results less accurate)
                                                             # 2: FCC (help generation of special grid of q points)
ngqpt 16 16 16                                               # Monkhorst-Pack grid to consider
                                                             # (must correspond to DDB data or a subset of it)
nqshft 1                                                     # number of q-shifts
q1shft 0 0 0                                                 # shift vectors of coarse grid
```

e-ph. superconductivity in ABINIT (5/6)

```
#########################################################################
# SUPERCONDUCTIVITY SETTINGS
#########################################################################
mustar 0.136                                          # Coulomb pseudopotential parameter

#########################################################################
# PATH TO INTERPOLATE PHONON LINEWIDTHS
#########################################################################
nqpath 7
qpath 0.0 0.0 0.0
      1/2 1/2 0.0
      1   1   1
      1/2 1/2 1/2
      1/2 1/2 0.0
      1/2 3/4 1/4
      1/2 1/2 1/2
```

μ^*: Morel-Anderson pseudo-potential

Must be specified because calculation of phonon linewidths included in e-ph analysis. Error raised otherwise.

e-ph. superconductivity in ABINIT (6/6)

- Eliashberg function $\alpha^2 F(\omega)$ stored in *_A2F file

- Superconducting properties can be found in ANADDB output file anaddb.abo:

```
 Superconductivity : isotropic evaluation of parameters from electron-phonon coupling.

 mka2f: lambda <omega^2> =     4.709738E-07
 mka2f: lambda <omega^3> =     5.358574E-10
 mka2f: lambda <omega^4> =     6.386539E-13
 mka2f: lambda <omega^5> =     7.861400E-16
 mka2f: isotropic lambda =     4.700638E-01
 mka2f: omegalog  =     8.686447E-04 (Ha)     2.742960E+02 (Kelvin)
 mka2f: input mustar =     1.360000E-01
-mka2f: MacMillan Tc =     4.022671E-06 (Ha)     1.270258E+00 (Kelvin)
```

λ parameter

$\omega_{\log}$ parameter

T_c critical temperature

Remark

- Always check convergence of your phonon and e-ph related results with respect to variation of k- and q- grid size! You will get unphysical results (diverging Tc) if grids are too small.
- For e-ph coupling, you need large k- and q- grids because electron-phonon calculations presently implemented in ANADDB for superconductivity, determine isotropic quantities, which are averaged over the Fermi surface and summed over q-points.

Thank you for your attention!