

The ABINIT code



X. Gonze

Thanks to the > 50 ABINIT contributors, and especially to GM Rignanese for contributions to the slides

ABINIT software project

Ideas (1997) :

- 1) Softwares for first-principles simulations are more and more complex :
needs a worldwide collaboration, of specialized, complementary, groups
- 2) Linux software development : 'free software' model

Now (2022) :

Estimated >2000 users worldwide

>800 kLines of F90 + many python scripts (abipy)

about 50 contributors to ABINITv8/v9

Last release v9.6.2 used
in this school

<http://www.abinit.org>

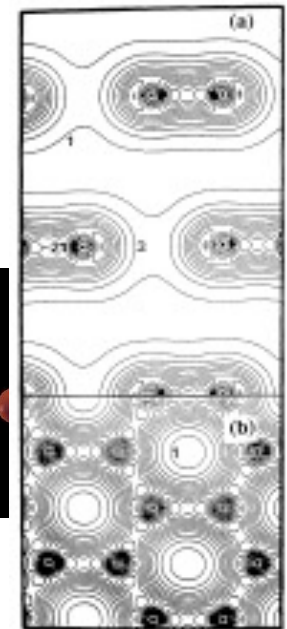
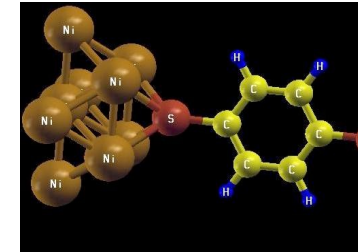
Available freely
(GPL, like Linux).



Properties from DFT+MBPT+ ...

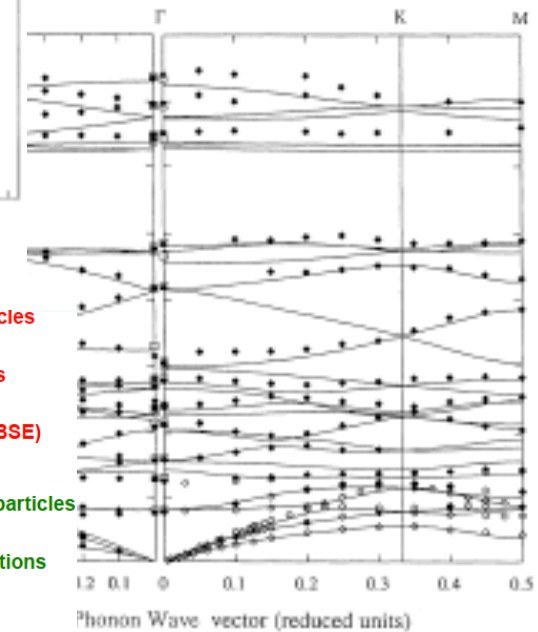
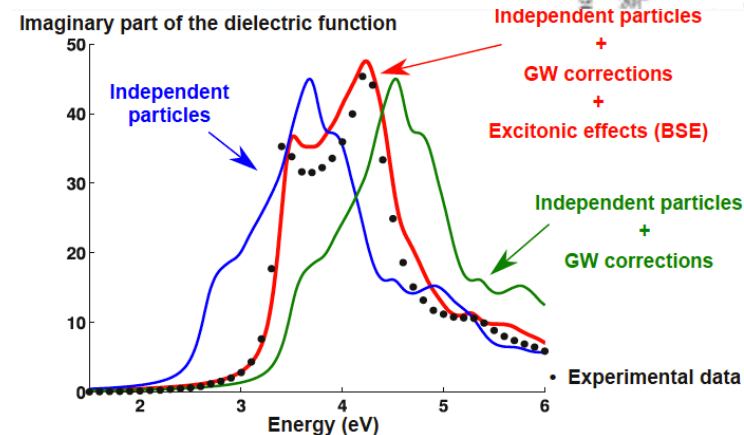
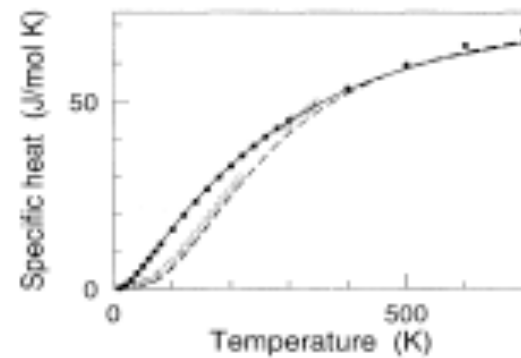
Computation of ...

interatomic distances, angles, total energies
electronic charge densities, electronic energies



A basis for the computation of ...

chemical reactions
electronic transport
vibrational properties
thermal capacity
dielectric behaviour
optical response
superconductivity
surface properties
spectroscopic responses
...



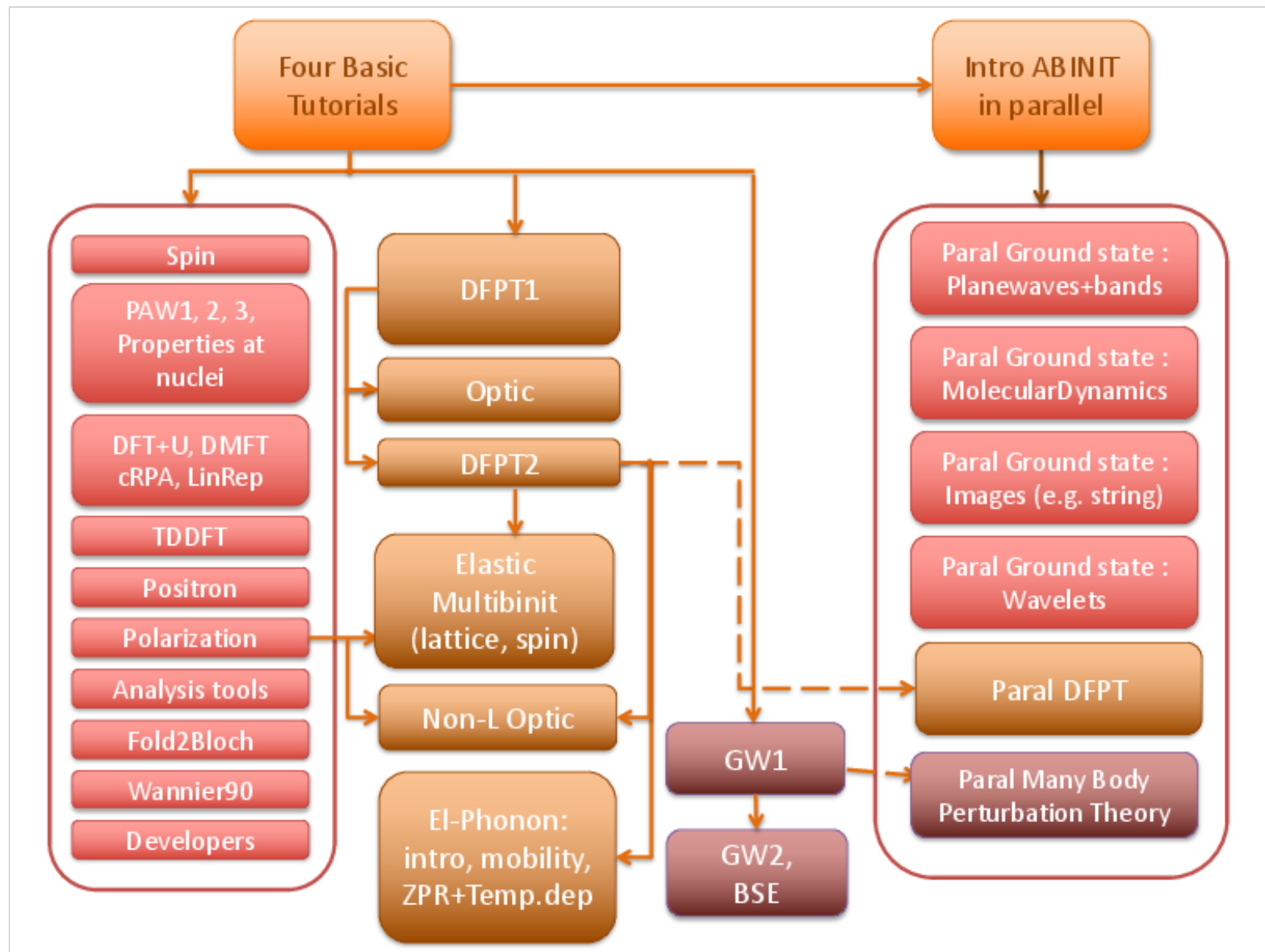
Basic Documentation

Web site <http://www.abinit.org> ; <http://docs.abinit.org>

- User's guides
- Installations notes
- List of input variables + description
- List of topics = a hub to input variables, files, tutorial, bibrefs
- over 800 example input files
- >30 tutorial lessons (each 1-2 hours)
<https://docs.abinit.org/tutorial>

+ Forum Web site <http://discourse.abinit.org>

ABINIT tutorial : layout + dependencies



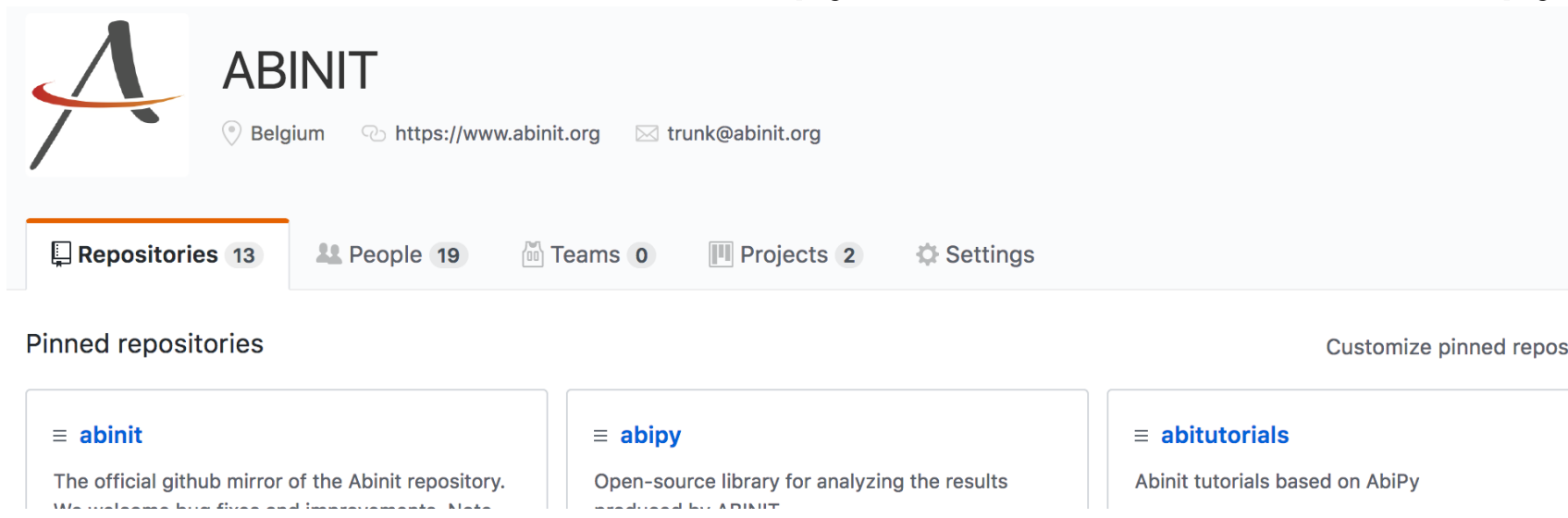
ABINIT + python : Abipy, Abitutorials ...

ABINIT organization on GitHub <https://github.com/abinit>

Abipy : python library for launching ABINIT jobs,
and analysing/plotting the results <http://pythonhosted.org/abipy>

=> e.g. connecting ABINIT with tools for high-throughput
calculations developed in the Materials Project context
(like Pymatgen, Fireworks).

Abitutorials : tutorial based on Jupyter notebooks ABINIT+python



ABINIT

Belgium <https://www.abinit.org> trunk@abinit.org

Repositories 13 People 19 Teams 0 Projects 2 Settings

Pinned repositories [Customize pinned repos](#)

- abinit**
The official github mirror of the Abinit repository.
- abipy**
Open-source library for analyzing the results
- abitutorials**
Abinit tutorials based on AbiPy

Running ABINIT : basics

Density Functional Theory calculations

In ABINIT ...

Representation of mathematical formalism
with a **Plane Wave** basis set :

- wavefunctions
- density, potential

Periodic boundary conditions

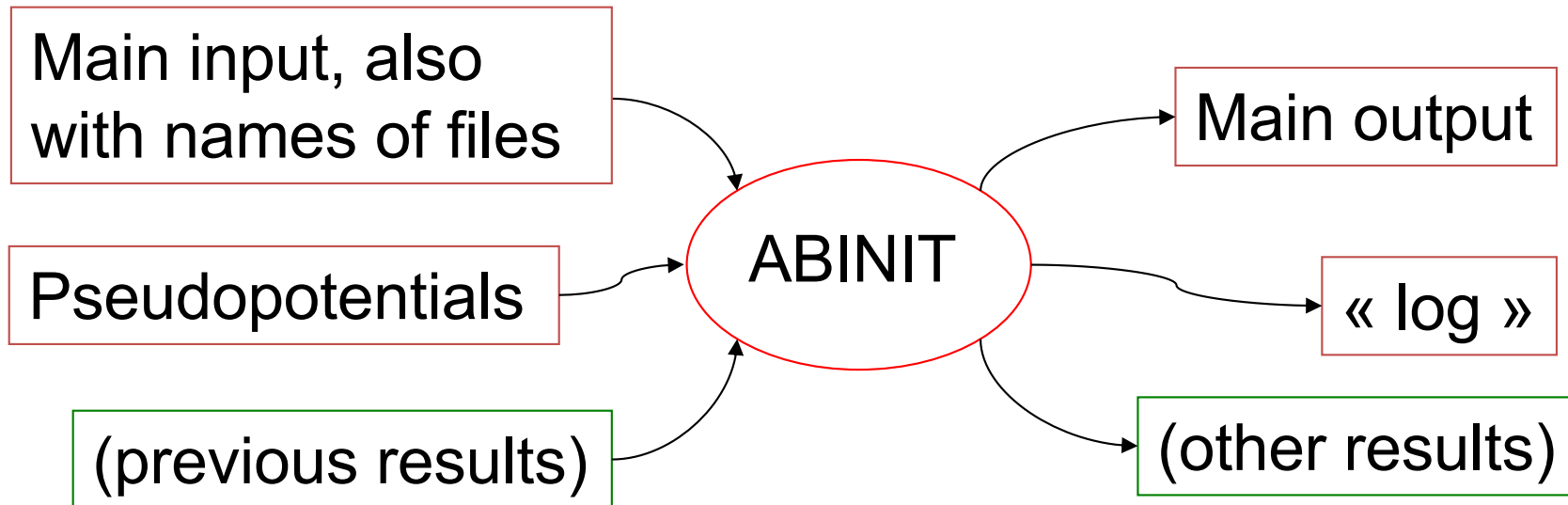
=> wavefunctions characterized by a **wavevector** (k-vector)

PseudoPotentials (or Projector Augmented Waves – PAW)

Iterative techniques to solve the equations

(Schrödinger equation ; DFT Self-consistency ; optimisation
of atomic positions)

External files in a ABINIT run



Results :

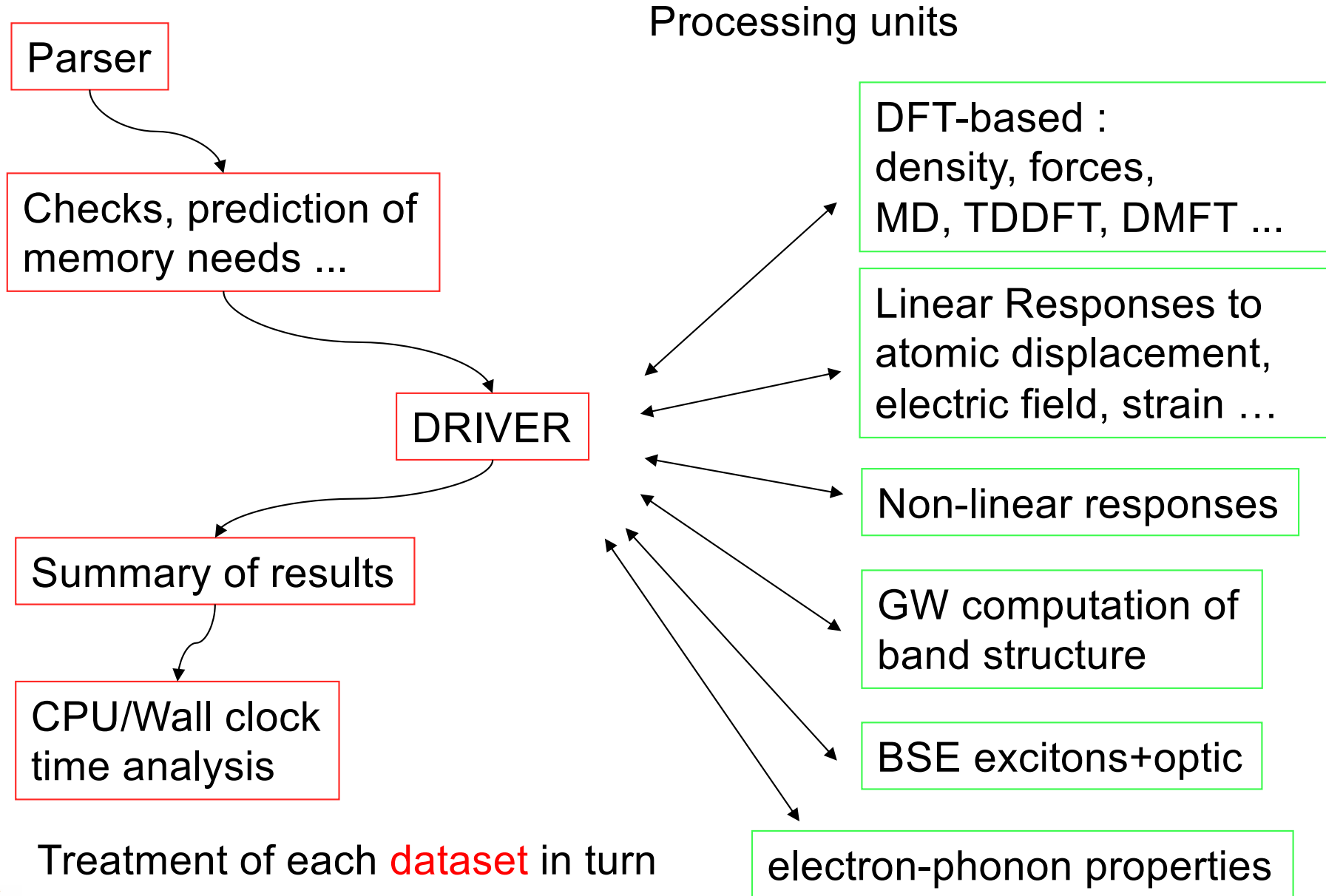
Text files : log, main output, energy derivatives (`_DDB`) ...

Binary F90 files : density (`_DEN`), potential (`_POT`), wavefunctions (`_WFK`), ...

netCDF files (similar to F90) : `_DEN.nc`, `_POT.nc`, `_WFK.nc`

Advantage of netCDF : portable, addressed by content, extensible, **Python-friendly**

ABINIT : the pipeline and the driver



A basic 'input' file : dihydrogen (I)

```
# H2 molecule in big cubic box
# Characters after '#' or after '!' are comments, will be ignored.
# Keywords followed by values. Order of keywords in file is not important.

# Definition of the unit cell
acell 10 10 10      # Keyword "acell" refers to lengths of primitive vectors (default in Bohr)
# Definition of the atom types
ntypat 1           # Only one type of atom
znucl 1            # Keyword "znucl" refers to atomic number of possible type(s) of atoms.
pseudos "Pseudodojo_nc_sr_04_pw_standard_psp8/H.psp8"
    # Pseudopotential file name, for the only type of atom, hydrogen.
    # It comes from pseudodojo site http://www.pseudo-dojo.org/ (NC SR LDA standard),
    # and was generated using the LDA XC functional (PW=Perdew-Wang, ixc -1012).
    # By default, abinit uses same XC functional than the one of input pseudopotential(s)
# Definition of the atoms
natom 2           # Two atoms
typat 1 1         # Both are of type 1, that is, Hydrogen
xcart             # Keyword "xcart" indicates that location of the atoms
                # will follow, one triplet of numbers for each atom
-0.7 0.0 0.0     # Triplet giving cartesian coordinates of atom 1, in Bohr
0.7 0.0 0.0     # Triplet giving cartesian coordinates of atom 2, in Bohr
```

A basic input file : dihydrogen (II)

Definition of **planewave basis set**

ecut 10.0 # Maximal plane-wave kinetic energy cut-off, in Hartree

Definition of **k-point grid**

kptopt 0 # Enter k points manually

nkpt 1 # Only one k point is needed for isolated system,
taken by default to be 0.0 0.0 0.0

#Definition of **SCF (self-consistent field) procedure**

nstep 10 # Maximal number of SCF cycles

toldfe 1.0d-6 # Will stop when, twice in a row, the difference
between two consecutive evaluations of total energy
differs by less than toldfe (default in Hartree)

diemac 2.0 # Although this is not mandatory, it is worth to precondition the
SCF cycle. A model dielectric function, used as standard
preconditioner, is described in "dielng" input variable section.
Here, we follow prescriptions for molecules in a big box

After modifying the following section, one might need to ...

##<BEGIN TEST_INFO> **Metadata ... to be ignored in the tutorial !**

Specification of the atomic geometry

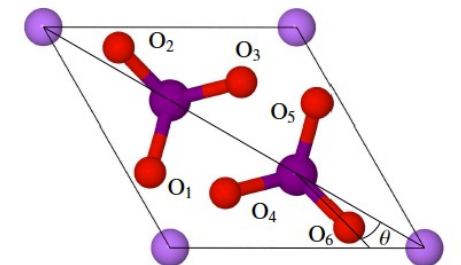
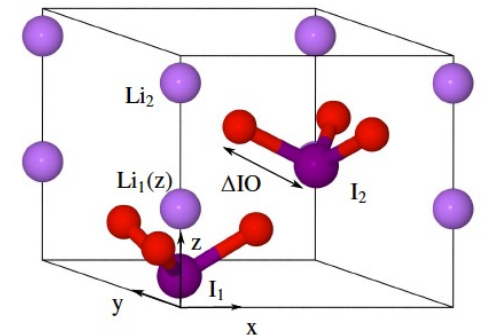
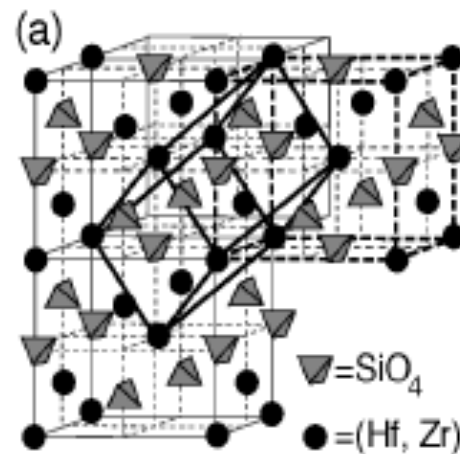
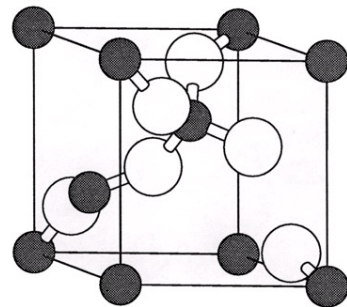
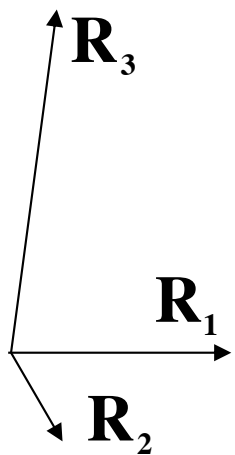
ABINIT : treatment of periodic systems

Plane waves $e^{i\mathbf{K}\mathbf{r}}$: particularly simple and efficient (when used with pseudopotentials), but infinite spatial extent.

Cannot use a finite set of planewaves for finite systems !

Need periodic boundary conditions.

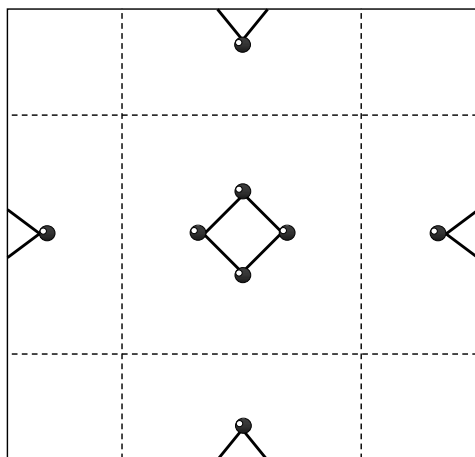
Primitive vectors \mathbf{R}_j , primitive cell volume Ω_0



OK for crystalline solids

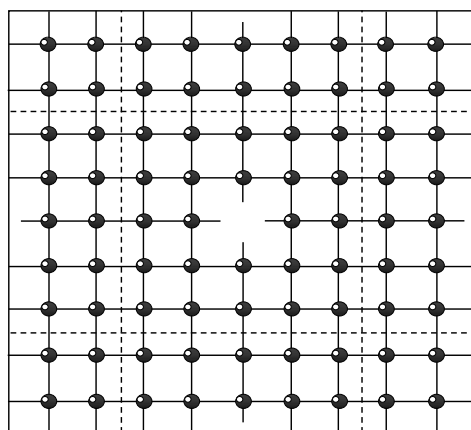
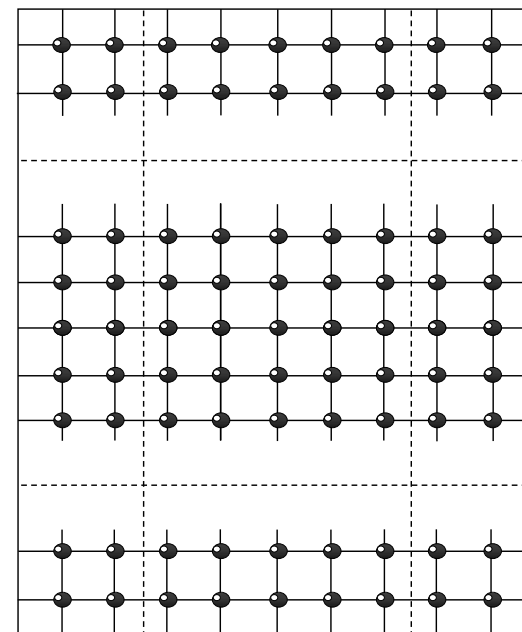
But : finite systems, surfaces, defects, polymers, nanosystems ... ?

Solution : the supercell technique



Molecule,
cluster

Surface : treatment
of a slab
Interface

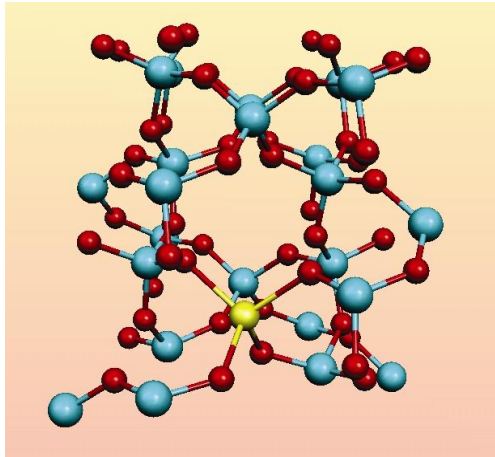


Point defect in a bulk solid

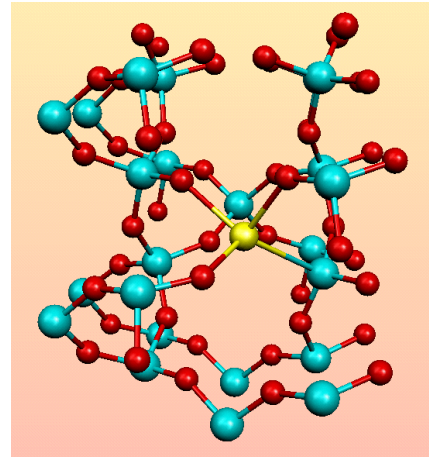
The supercell must be sufficiently big : convergence study

Examples of defects SiO₂-quartz : Pb

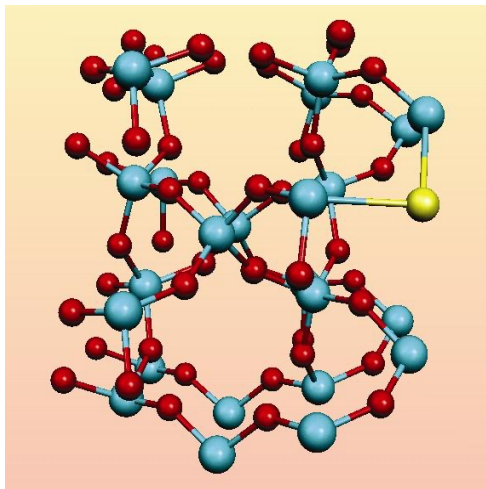
72-atom supercell of quartz



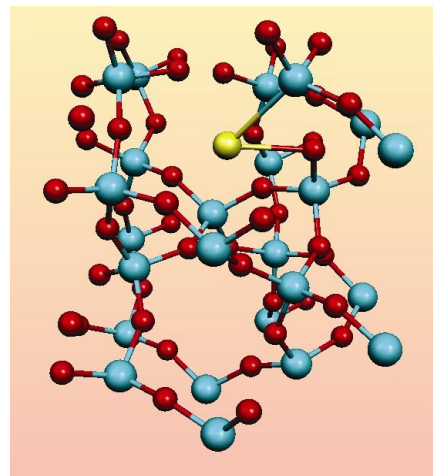
Pb^{Si}



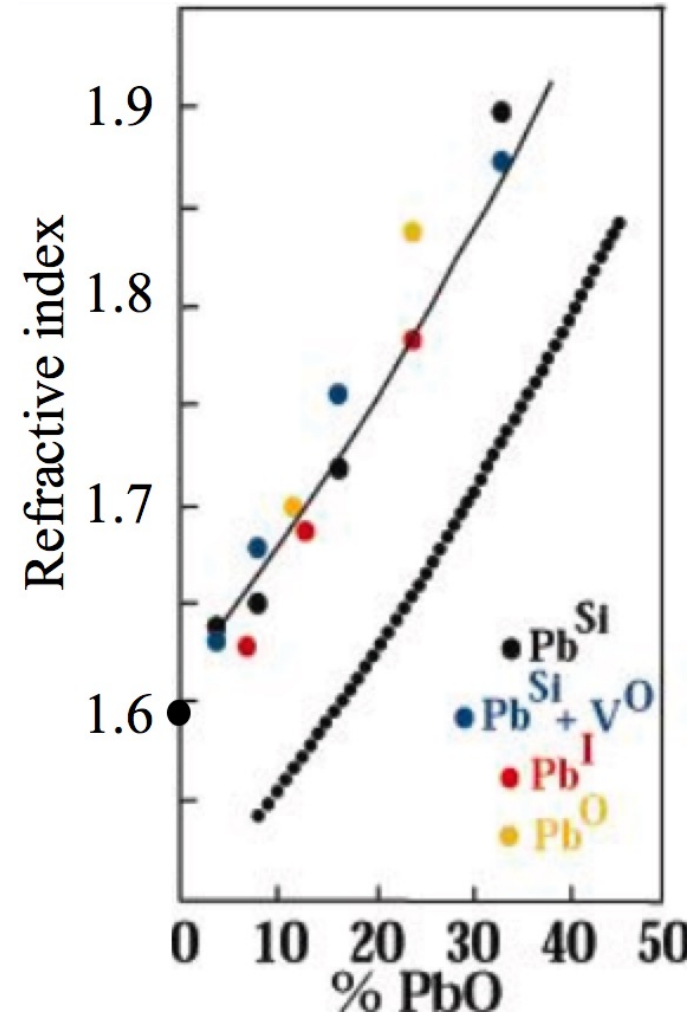
Pb^{Si} + V⁰



Pb⁰



Pb^I

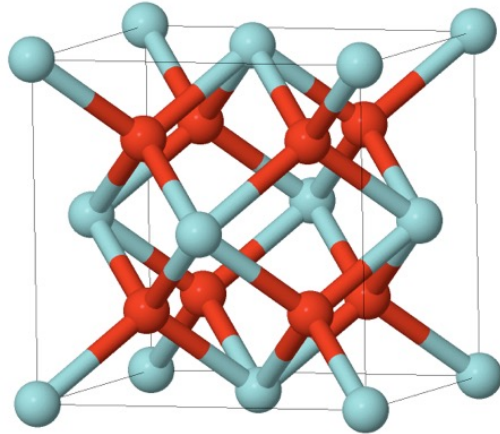


Comparison with amorphous SiO₂

Main input file : input variable flexibility

- cell primitive vectors → **rprim**
 - ... or angle (degrees) between primitive vectors → **angdeg**
 - + scale cell vector lengths → **acell**
 - + scale cartesian coordinates → **scalectart**
- number of atoms → **natom**
- reduced coordinates → **xred** (initial guess ...)
 - ... or cartesian → **xcart** (default in Bohr but Å if specified)
- type of atoms → **typat**
- space group either automatically recognized from coords,
 - ... or → **spgroup** + **natrd**
 - ... or number of symmetries → **nsym**
 - + symmetry operations → **symrel** + **tnons**

Example : cubic zirconium dioxide



$Fm-3m$
 $a=5.010\text{\AA}$
 $b=5.010\text{\AA}$
 $c=5.010\text{\AA}$
 $\alpha=90.0^\circ$
 $\beta=90.0^\circ$
 $\gamma=90.0^\circ$

Bilbao Crystallographic Server → Assignment of Wyckoff Positions

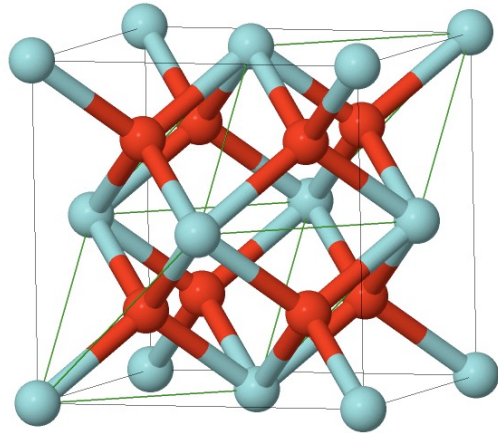
Assignment of Wyckoff Positions

Atoms Data:

AT.	WP	SS	Representative	Atomic orbit
Zr1	4a (0,0,0)	m-3m	(0.000000,0.000000,0.000000)	(0.000000,0.000000,0.000000) (0.000000,0.500000,0.500000) (0.500000,0.000000,0.500000) (0.500000,0.500000,0.000000)
O2	8c (1/4,1/4,1/4)	-43m	(0.250000,0.250000,0.250000)	(0.250000,0.250000,0.250000) (0.750000,0.750000,0.250000) (0.750000,0.250000,0.750000) (0.250000,0.750000,0.750000) (0.250000,0.250000,0.750000) (0.750000,0.750000,0.750000) (0.250000,0.750000,0.250000) (0.750000,0.250000,0.250000)

Face-centered cubic, with three atoms per primitive cell

Example : cubic zirconium dioxide



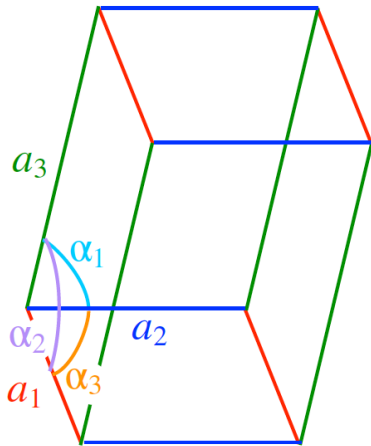
```
natom 3
typat 1 2 2
acell 3*5.01 Angst    NOTE "*" is a repeater
rprim 0.0 0.5 0.5
        0.5 0.0 0.5
        0.5 0.5 0.0
xred 3*0.0 3*0.25 3*0.75
=> symmetries are found automatically
```

OR

```
natom 3
typat 1 2 2
acell 3*5.01 Angst
rprim 0.0 0.5 0.5
        0.5 0.0 0.5
        0.5 0.5 0.0
spgroup 225    natrd 2
xred 3*0.0 3*0.25 => the set of atoms is completed automatically
```

Primitive vectors in ABINIT (rprimd)

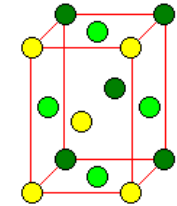
- $R_i(j) \rightarrow \text{rprimd}(j,i) = \text{scalecart}(j) \times \text{rprim}(j,i) \times \text{acell}(i)$



```

scalecart  9.5  9.8  10.0
rprim      0.0  0.5  0.5
           0.5  0.0  0.5
           0.5  0.5  0.0
    
```

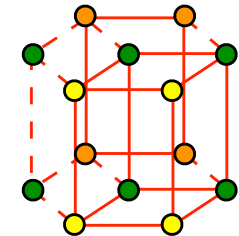
face-centered
orthorhombic



```

acell      9.5  9.5  10.0
rprim     0.8660254038E+00  0.5  0.0
          -0.8660254038E+00  0.5  0.0
           0.0  0.0  1.0
    
```

hexagonal



- $a_i \rightarrow \text{acell}(i) / \alpha_i \rightarrow \text{angdeg}(i)$

```

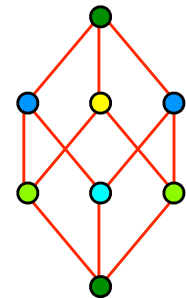
acell      9.5  9.5  10.0
angdeg     120  90  90
    
```

hexagonal

```

acell      9.0  9.0  9.0
angdeg     48  48  48
    
```

trigonal



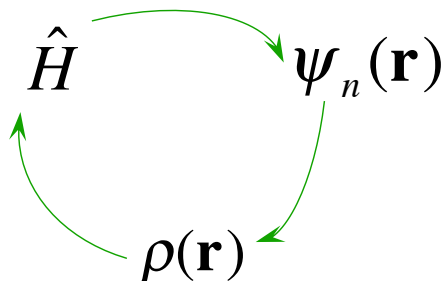
The plane wave basis set

$$\psi_{\mathbf{k}}(\mathbf{r}) = (N\Omega_0)^{-1/2} \sum_{\mathbf{G}} u_{\mathbf{k}}(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G})\mathbf{r}}$$

A reminder : basic equations in DFT

Solve self-consistently the Kohn-Sham equation

$$\left\{ \begin{array}{l} \hat{H} |\psi_n\rangle = \epsilon_n |\psi_n\rangle \\ \hat{H} = \hat{T} + \hat{V} + \hat{V}_{Hxc}[\rho] \\ \rho(\vec{r}) = \sum_n^{occ} \psi_n^*(\vec{r}) \psi_n(\vec{r}) \end{array} \right.$$



$$\delta_{mn} = \langle \psi_m | \psi_n \rangle \text{ for } m, n \in \text{occupied set}$$

or minimize

$$E_{el} \{ \psi \} = \sum_n^{occ} \langle \psi_n | \hat{T} + \hat{V} | \psi_n \rangle + E_{Hxc}[\rho]$$

with

$$\hat{V}(\vec{r}) = \sum_{aK} - \frac{Z_K}{|\vec{r} - \vec{R}_K^a|}$$

Periodic system : wavevectors

For a **periodic** Hamiltonian : wavefunctions characterized by a wavevector \mathbf{k} (crystal momentum) in Brillouin Zone

Bloch's theorem $\psi_{m,\mathbf{k}}(\mathbf{r}+\mathbf{R}_j) = e^{i\mathbf{k}\cdot\mathbf{R}_j}\psi_{m,\mathbf{k}}(\mathbf{r})$

$$\psi_{m,\mathbf{k}}(\mathbf{r}) = (N\Omega_0)^{-1/2} e^{i\mathbf{k}\cdot\mathbf{r}} u_{m,\mathbf{k}}(\mathbf{r}) \quad u_{m,\mathbf{k}}(\mathbf{r}+\mathbf{R}_j) = u_{m,\mathbf{k}}(\mathbf{r})$$

Normalization ?

Born-von Karman supercell supercell vectors $N_j\mathbf{R}_j$ with $N=N_1N_2N_3$

$$\psi_{m,\mathbf{k}}(\mathbf{r}+N_j\mathbf{R}_j) = \psi_{m,\mathbf{k}}(\mathbf{r})$$

Planewave basis set

Reciprocal lattice : set of \mathbf{G} vectors such that $e^{i\mathbf{G}\mathbf{R}_j} = 1$
 $e^{i\mathbf{G}\mathbf{r}}$ has the periodicity of the real lattice

$$u_{\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{G}} u_{\mathbf{k}}(\mathbf{G}) e^{i\mathbf{G}\mathbf{r}}$$

$$\psi_{\mathbf{k}}(\mathbf{r}) = (N\Omega_0)^{-1/2} \sum_{\mathbf{G}} u_{\mathbf{k}}(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G})\mathbf{r}}$$

$$u_{\mathbf{k}}(\mathbf{G}) = \frac{1}{\Omega_0} \int_{\Omega_0} e^{-i\mathbf{G}\mathbf{r}} u_{\mathbf{k}}(\mathbf{r}) d\mathbf{r} \quad (\text{Fourier transform})$$

Kinetic energy of a plane wave $-\frac{\nabla^2}{2} \rightarrow \frac{(\mathbf{k}+\mathbf{G})^2}{2}$

The coefficients $u_{\mathbf{k}}(\mathbf{G})$ for the lowest eigenvectors decrease exponentially with the kinetic energy $\frac{(\mathbf{k} + \mathbf{G})^2}{2}$

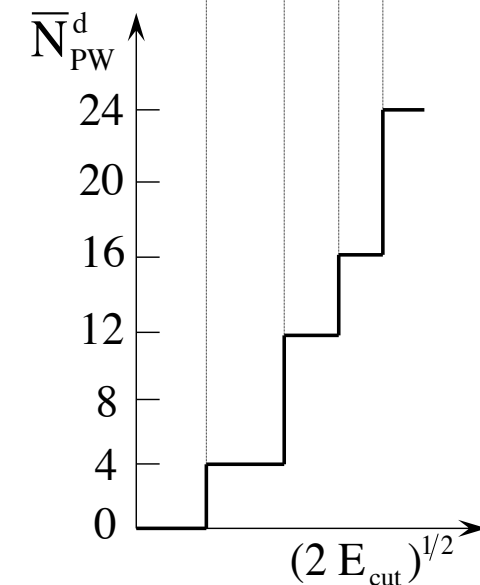
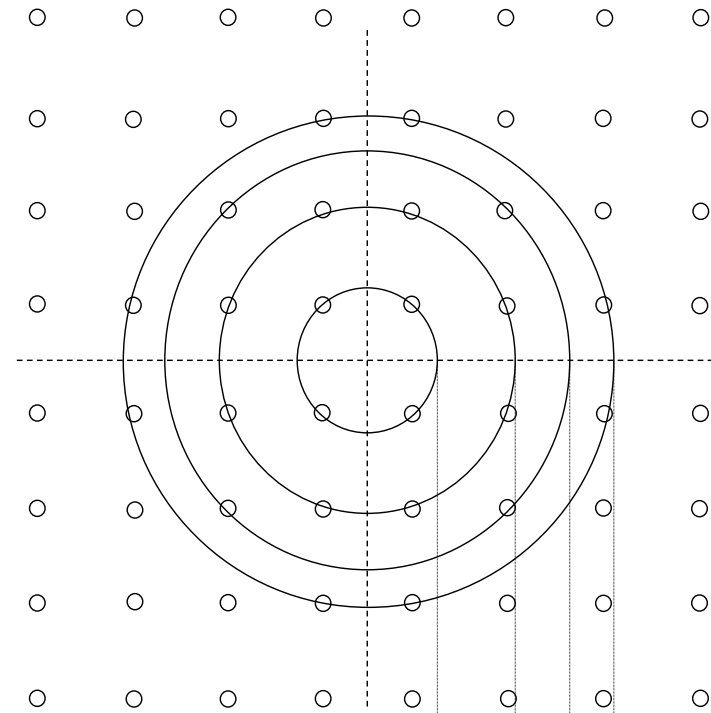
Selection of plane waves determined by a cut-off energy E_{cut}

$$\frac{(\mathbf{k} + \mathbf{G})^2}{2} < E_{\text{cut}} \quad \text{Plane wave sphere}$$

e_{cut}

Number of planewaves

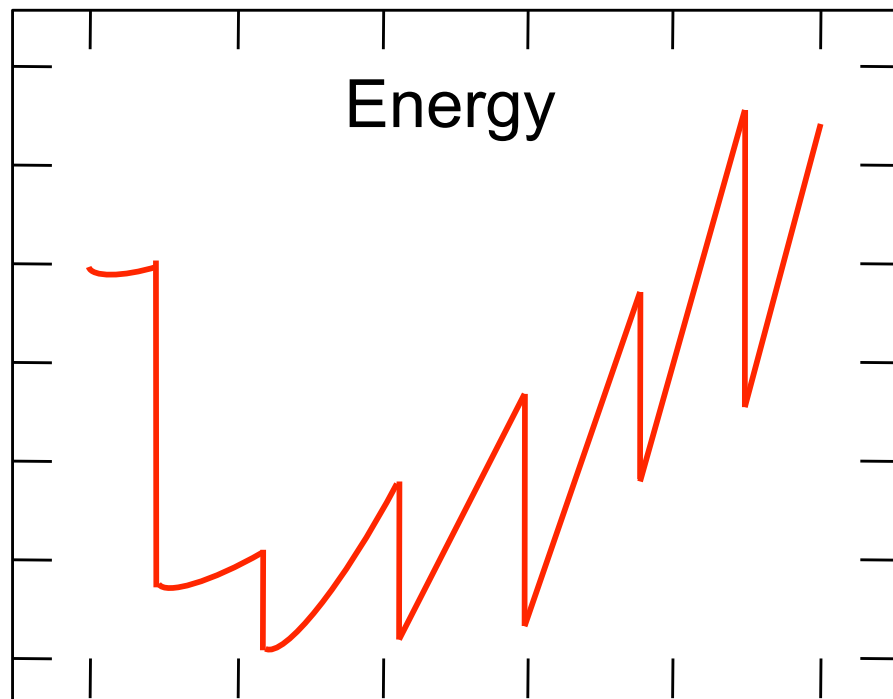
Number of plane waves
= function of the
kinetic energy cut-off
... **not continuous**



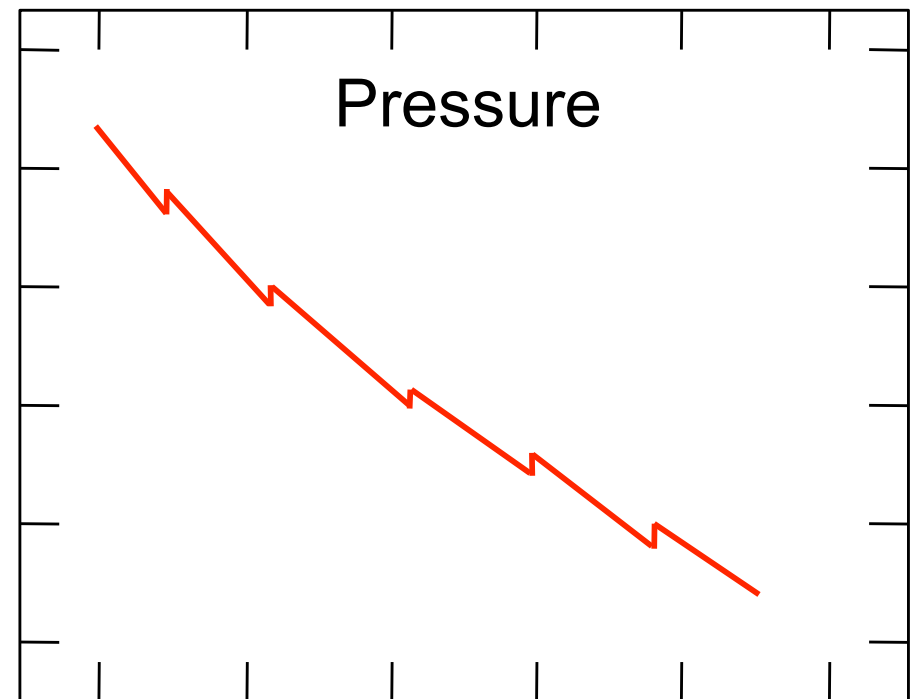
Also, a (discontinuous) function of
lattice parameter at fixed kinetic energy

Discontinuities in energy and pressure

=> Energy (and pressure) also (discontinuous) functions of lattice parameter at fixed kinetic energy



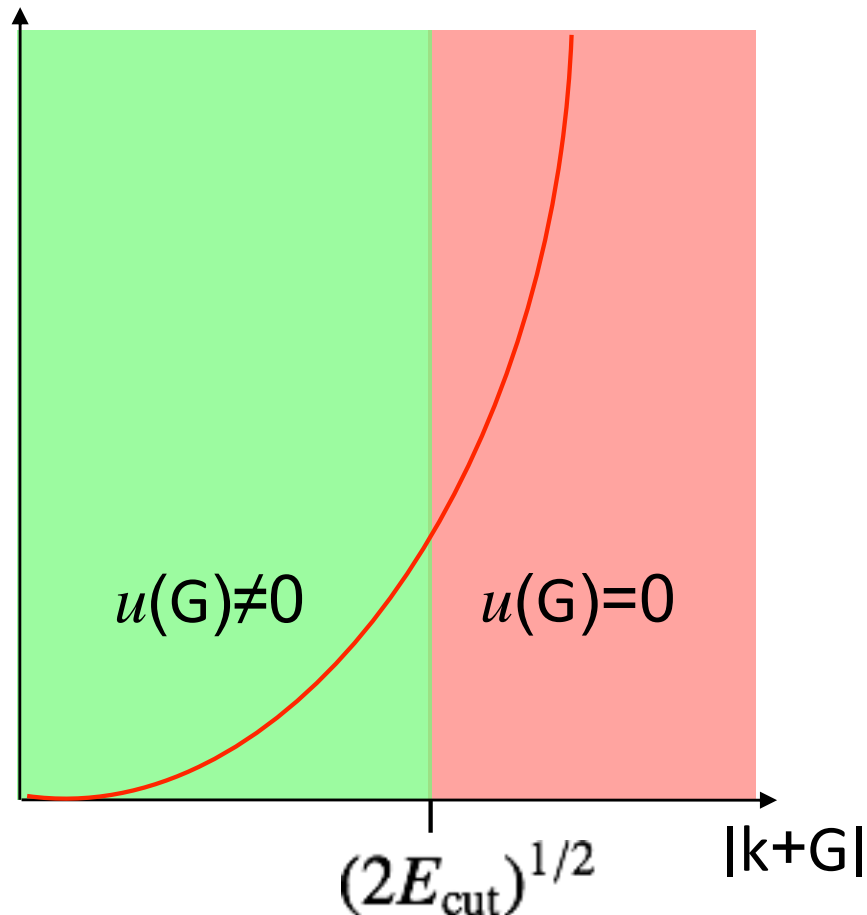
Lattice parameter



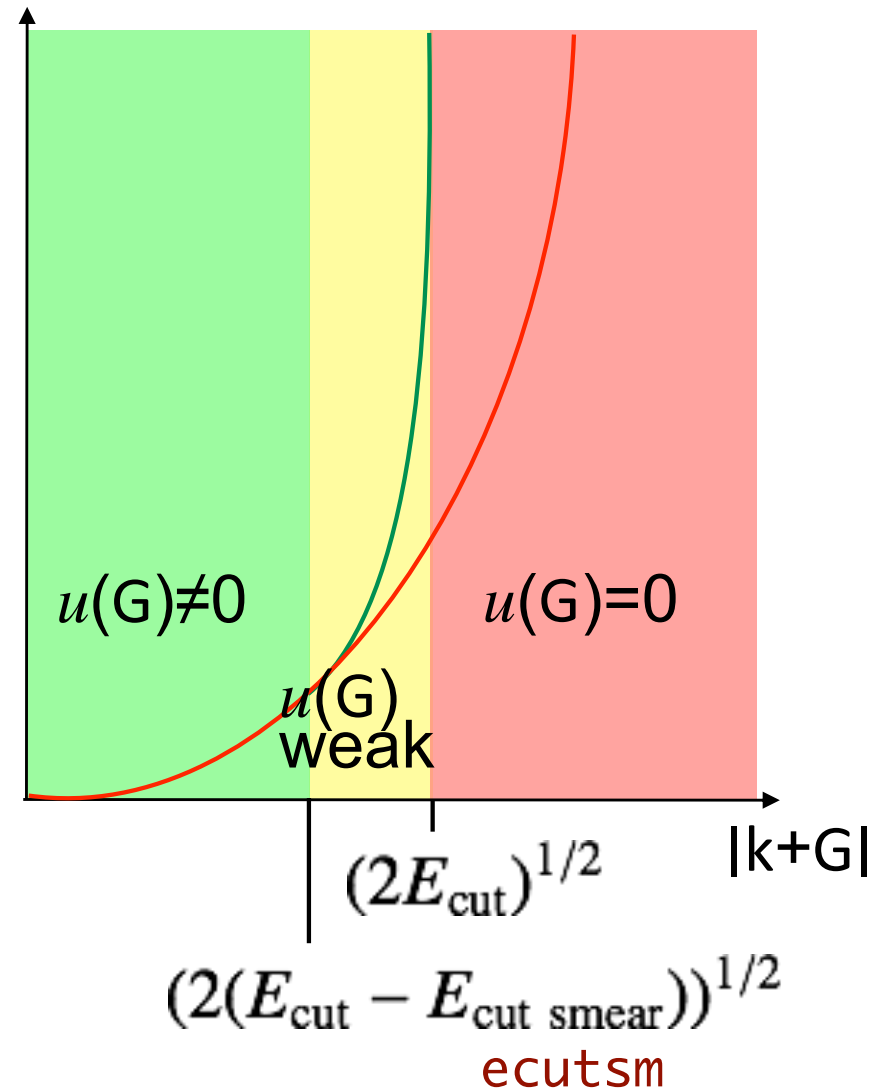
Lattice parameter

Removing discontinuities

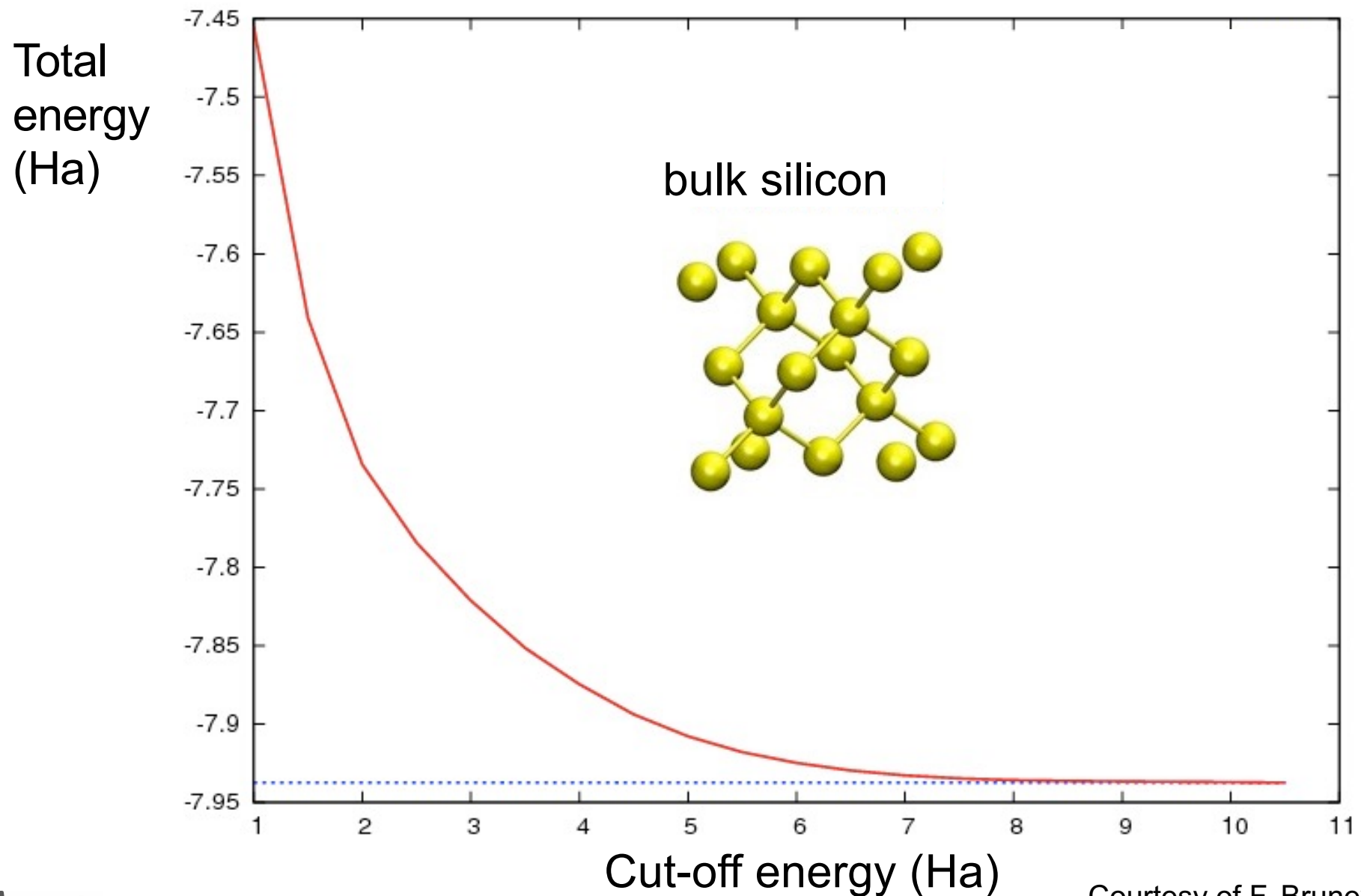
Kinetic energy



Kinetic energy



Convergence wrt to kinetic energy cutoff



Courtesy of F. Bruneval

Plane waves : the density and potential

Fourier transform of a periodic function $f(\mathbf{r})$

$$f(\mathbf{G}) = \frac{1}{\Omega_{\text{or}}} \int_{\Omega_{\text{or}}} e^{-i\mathbf{G}\mathbf{r}} f(\mathbf{r}) d\mathbf{r} \quad f(\mathbf{r}) = \sum_{\bar{\mathbf{G}}} e^{i\mathbf{G}\mathbf{r}} f(\mathbf{G})$$

Poisson equation $\Rightarrow n(\mathbf{G})$ and $V_H(\mathbf{G})$

$$V_H(\mathbf{r}) = \int \frac{n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}' \Leftrightarrow \nabla^2 V_H|_{\mathbf{r}} = -4\pi n(\mathbf{r})$$

Relation between Fourier coefficients:

$$G^2 V_H(\mathbf{G}) = 4\pi n(\mathbf{G}) \quad V_H(\mathbf{G}) = \frac{4\pi}{G^2} n(\mathbf{G})$$

For $G^2 = 0$ ($\mathbf{G} = 0$) divergence of V_H ($\mathbf{G} = 0$)

$$n(\mathbf{G} = 0) = \frac{1}{\Omega_{\text{or}}} \int_{\Omega_{\text{or}}} n(\mathbf{r}) d\mathbf{r} \quad \text{Average}$$

Representation of the density

Density associated with one eigenfunction :

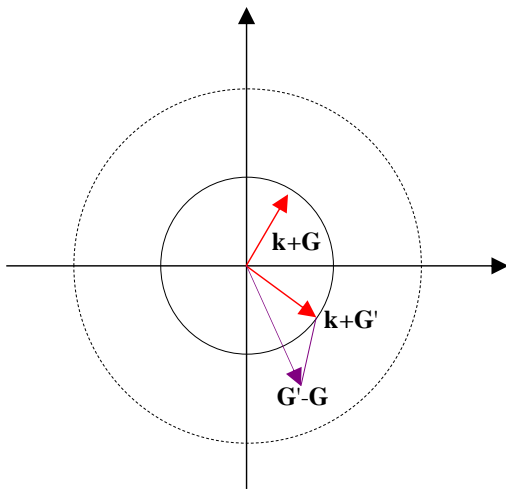
$$n_{nk}(\mathbf{r}) = u_{nk}^*(\mathbf{r}) u_{nk}(\mathbf{r})$$

Computation of

$$u_{nk}^*(\mathbf{r}) u_{nk}(\mathbf{r})$$

$$= \left(\sum_{\mathbf{G}} u_{nk}^*(\mathbf{G}) e^{i\mathbf{G}\mathbf{r}} \right) \left(\sum_{\mathbf{G}'} u_{nk}(\mathbf{G}') e^{-i\mathbf{G}'\mathbf{r}} \right)$$

$$= \sum_{\mathbf{G}\mathbf{G}'} \left[u_{nk}^*(\mathbf{G}) u_{nk}(\mathbf{G}') \right] e^{i(\mathbf{G}'-\mathbf{G})\mathbf{r}}$$



Non-zero coefficients for $\mathbf{k}+\mathbf{G} \in \text{sphere}$

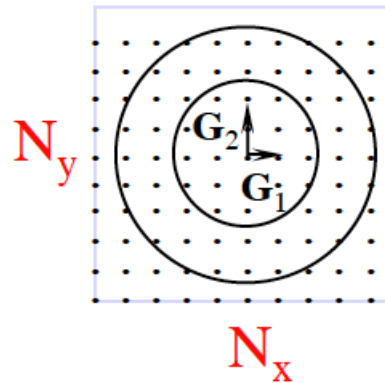
$\mathbf{k}+\mathbf{G}' \in \text{sphere}$

The sphere for $n(\mathbf{G})$ has a double radius

From real space to reciprocal space

$$n(\mathbf{r}) = \sum_{\mathbf{G} \in \text{sphere}(2)} n(\mathbf{G}) e^{i\mathbf{G}\mathbf{r}}$$

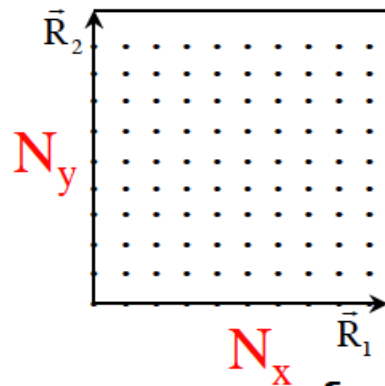
Use of the discrete Fourier transform $\{\mathbf{r}_i\} \leftrightarrow \{\mathbf{G}\}$



Reciprocal lattice

$$n(\mathbf{G}) = \frac{1}{N_{\mathbf{r}_i}} \sum_{\{\mathbf{r}_i\}} n(\mathbf{r}_i) e^{-i\mathbf{G}\mathbf{r}_i}$$

Fast Fourier Transform algorithm



Real lattice: original cell

Representation : wrap-up

- Choice of a basis (e.g. Plane waves)
- Truncating of the basis -> finite basis

$$\frac{(\mathbf{k}+\mathbf{G})^2}{2} < E_{\text{cut}} \quad \text{Sphere of plane waves}$$

- Discontinuous increase of the number of plane waves ?

Smearing of $u(\mathbf{G})$

-> Progressive incorporation of new \mathbf{G} vectors

- Representation of the density

Sphere with a double radius in the reciprocal space

- Going from the real space to reciprocal space

Discrete Fourier transform

Grid of points + Fast Fourier Transform

$$\{\mathbf{r}_i\} \leftrightarrow \{\mathbf{G}\}$$

Sampling the Brillouin zone

From discrete states to Brillouin zone

Discrete summations over states :

$$\text{Total kinetic energy} \quad \sum_i \langle \psi_i | -\frac{1}{2} \nabla^2 | \psi_i \rangle$$

$$\text{Density} \quad n(\mathbf{r}) = \sum_i \psi_i^*(\mathbf{r}) \psi_i(\mathbf{r})$$

In the periodic case : summation over energy bands +
integration over the Brillouin zone

$$\text{Total kinetic energy} \quad \sum_n \frac{1}{\Omega_{0\mathbf{k}}} \int_{\Omega_{0\mathbf{k}}} f(\varepsilon_F - \varepsilon_{\mathbf{n}\mathbf{k}}) \langle \psi_{\mathbf{n}\mathbf{k}} | -\frac{1}{2} \nabla^2 | \psi_{\mathbf{n}\mathbf{k}} \rangle d\mathbf{k}$$

$$\text{Density} \quad n(\mathbf{r}) = \sum_n \frac{1}{\Omega_{0\mathbf{k}}} \int_{\Omega_{0\mathbf{k}}} f(\varepsilon_F - \varepsilon_{\mathbf{n}\mathbf{k}}) \psi_{\mathbf{n}\mathbf{k}}^*(\mathbf{r}) \psi_{\mathbf{n}\mathbf{k}}(\mathbf{r}) d\mathbf{k}$$

How to treat $\frac{1}{\Omega_{0\mathbf{k}}} \int_{\Omega_{0\mathbf{k}}} X_{\mathbf{k}} d\mathbf{k}$?

Brillouin zone integration

$$\frac{1}{\Omega_{ok}} \int_{\Omega_{ok}} X_{\mathbf{k}} d\mathbf{k} \Rightarrow \sum_{\{\mathbf{k}\}} w_{\mathbf{k}} X_{\mathbf{k}} \quad [\text{with } \sum_{\{\mathbf{k}\}} w_{\mathbf{k}} = 1]$$

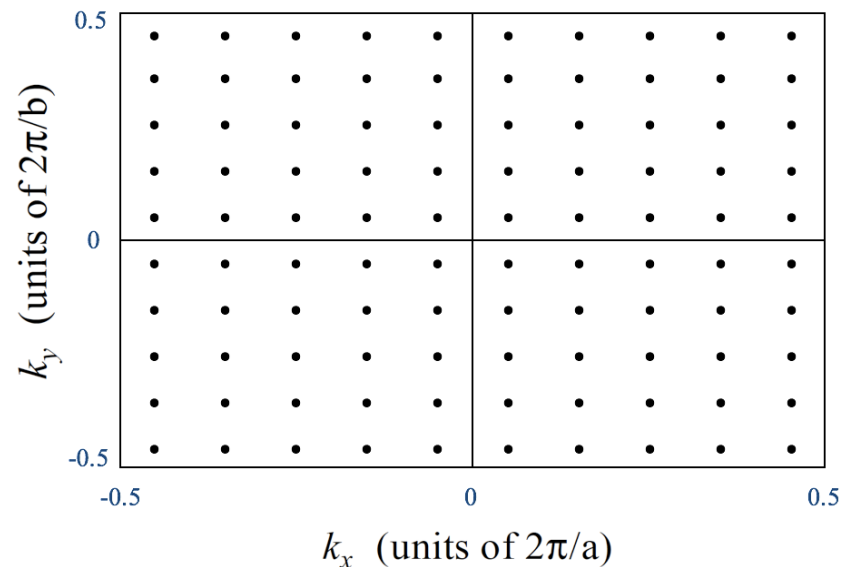
How to chose $\{\mathbf{k}\}$ and $\{w_{\mathbf{k}}\}$?

Special points

Weights

Simple answer : Homogeneous grid (1D - 2D - 3D) and equal weights

Homogeneous sampling of the Brillouin zone



Brillouin zone integration

Theorem :

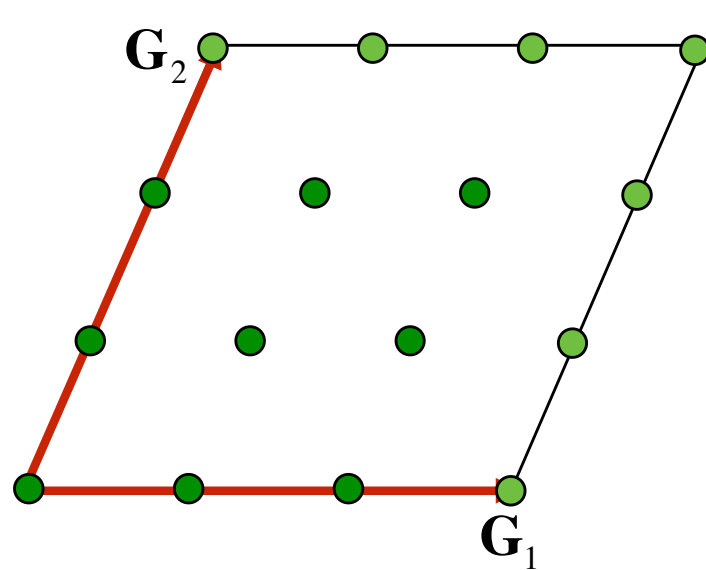
- If
- the integrand is periodic
 - the integrand is continuous + derivable at all orders ($C^\infty D^\infty$)
 - $\{\mathbf{k}\}$ homogeneous grid (1D - 2D - 3D) and $w_{\mathbf{k}}$ all equal

Then exponential convergence, with respect to $\Delta\mathbf{k}$

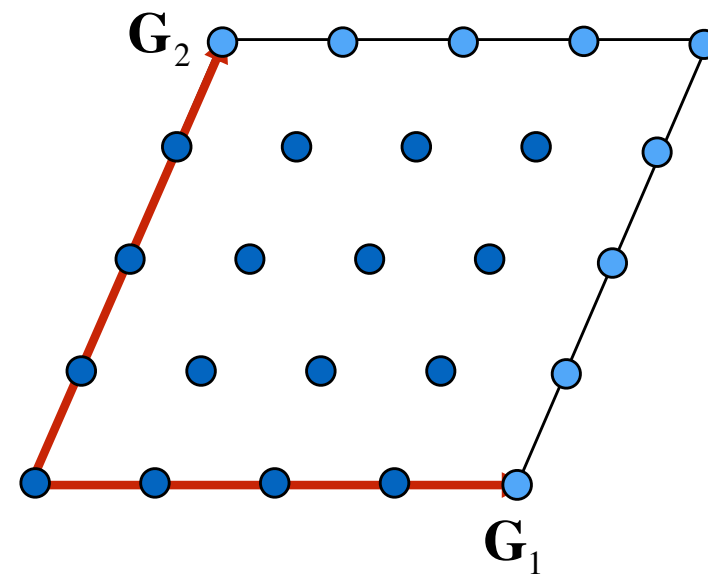
- OK for semiconductors/insulators where the occupation number is independent of k within a band
- Convergence : one ought to test several grids with different $\Delta\mathbf{k}$
- Monkhorst & Pack grids (Phys. Rev. B 13, 5188 (1976))
 $k_1 \times k_2 \times k_3$ points + simple cubic, FCC, BCC ...
- Other techniques ... (tetrahedron method)

BZ integration : Monkhorst-Pack grid

- Uniformly spaced grid of $n_{k1} \times n_{k2} \times n_{k3}$ points in the first Brillouin Zone [Monkhorst & Pack, Phys. Rev. B 13, 5188 (1976)]



$$n_{k1} = n_{k2} = 3$$

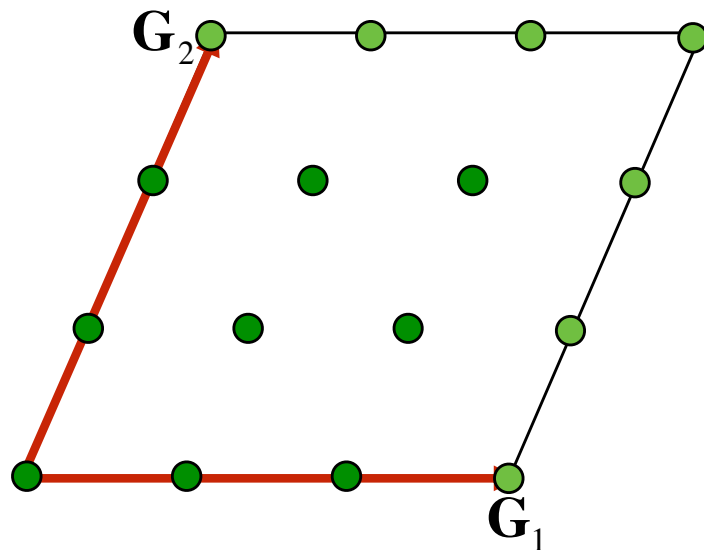


$$n_{k1} = n_{k2} = 4$$

ngkpt nk1 nk2 nk3

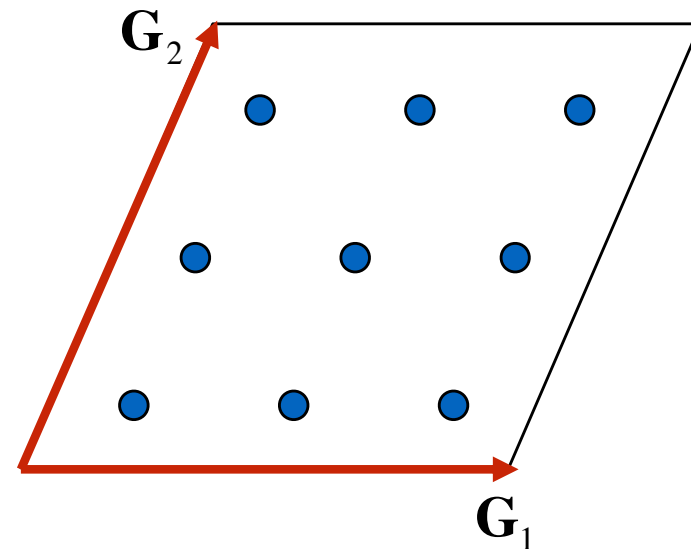
Unshifted and shifted grids

- k-points grid can be chosen to be shifted : not centered at Γ .
- Advantage : comparable accuracy can be obtained with fewer k-points in IBZ (especially for highly symmetric cases)



$$n_{k1} = n_{k2} = 3$$

unshifted



$$n_{k1} = n_{k2} = 3$$

shifted by $(1/2, 1/2)$

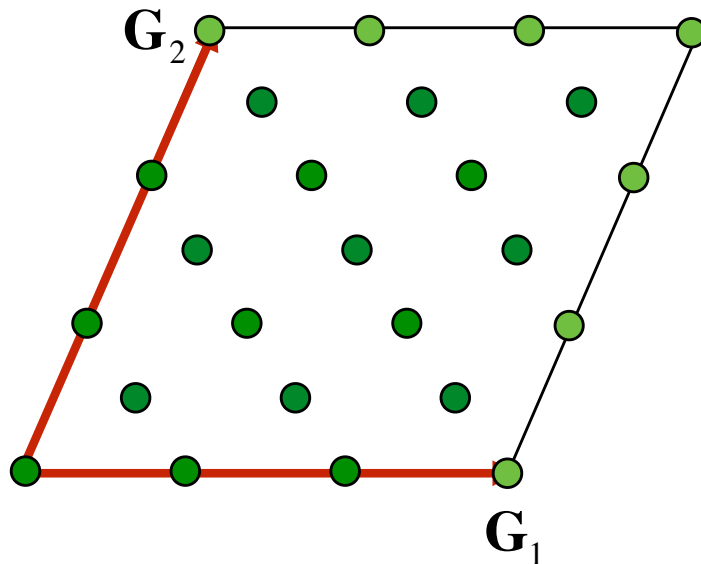
ngkpt nk1 nk2 nk3

shftk sk1 sk2 sk3

(default: 0.5 0.5 0.5)

Combining grids with various shifts

- k-points grid with various shifts can also be combined.



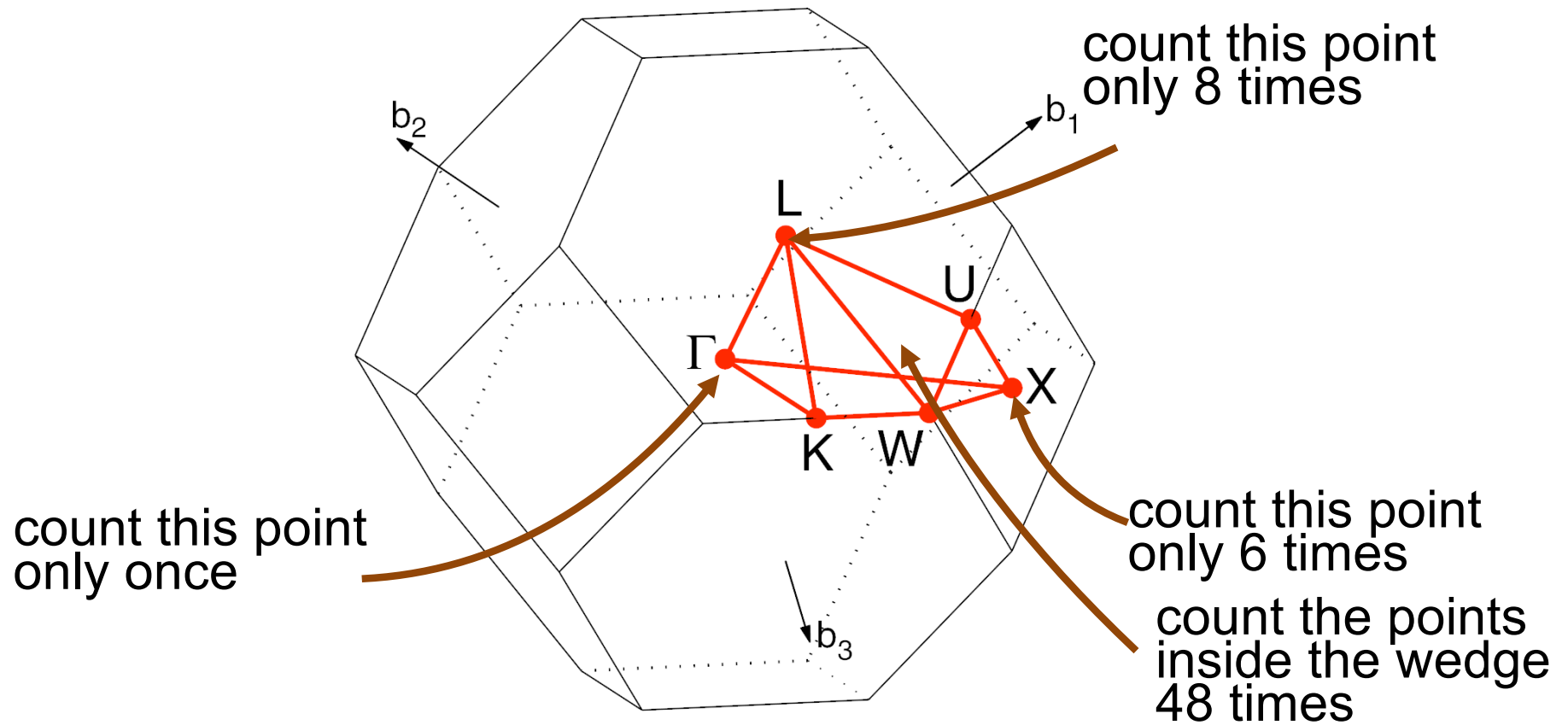
combining unshifted
and shifted by $(1/2, 1/2)$
for $n_{k1} = n_{k2} = 3$

```

ngkpt nk1 nk2 nk3
nshiftk nsk
shiftk sk1(1) sk2(1) sk3(1)
      sk1(2) sk2(2) sk3(2)
      ...
      sk1(nsk) sk2(nsk) sk3(nsk)
    
```

Irreducible wedge

- Using symmetries to avoid summing entire BZ :
- Restrict the sum to the Irreducible Brillouin zone (IBZ) provided that weights are adapted.

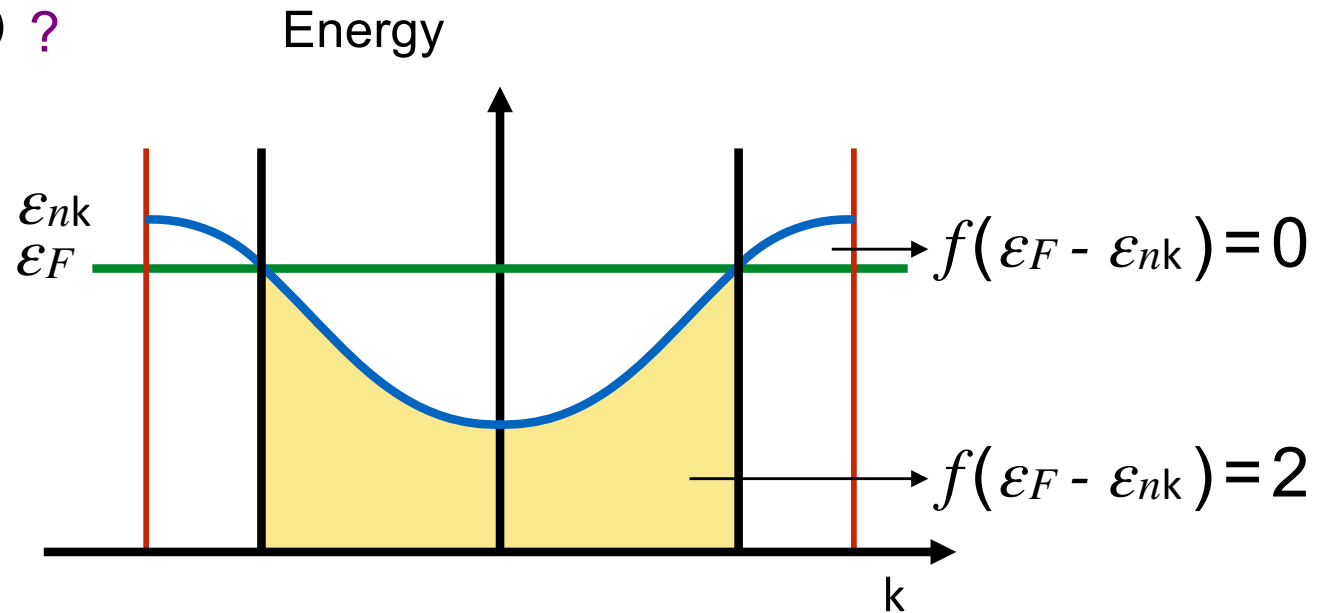


Treatment of metals (I)

Behaviour of $f(\epsilon_F - \epsilon_{nk})$?

Discontinuity of integrand at Fermi level

Smearing technique



First trial : generalisation of DFT to finite temperature

$$f(\epsilon_{nk}) = \frac{1}{1 + e^{(\epsilon_{nk} - \epsilon_F)/kT}}$$

f goes from 0 to 2 in an energy range $\sigma = k_B T$

$$E(T) \cong E(T=0) + \alpha T^2 + \dots$$

$$F(T) = E - TS$$

occpot 3
tsmear σ

Problem : T needed to recover the same convergence as for semiconductors is very high ($\gg 2000$ K)

Treatment of metals (II)

Better technique : obtain $E(\sigma = 0)$ from total energy expression $E(\sigma)$ with **modified occupation numbers**, and σ **similar** to a temperature

$$E(\sigma) = E(\sigma = 0) + \alpha\sigma^2 + O(\sigma^3) \quad \text{with } \alpha \text{ small}$$

$$\text{or } E(\sigma) = E(\sigma = 0) + \alpha\sigma^n + O(\sigma^{n+1}) \quad \text{with } n > 2$$

$$f_{nk}(\epsilon_{nk}) = s \cdot \int_{t=\frac{\epsilon_{nk}-\epsilon_F}{\sigma}}^{\infty} \tilde{\delta}(t) dt \quad [\text{with } \int_{-\infty}^{\infty} \tilde{\delta}(t) dt = 1]$$

Spin factor

Gaussian smearing $\tilde{\delta}(x) = \frac{1}{\sqrt{\pi}} e^{-x^2} \Rightarrow \alpha \text{ small}$ occopt 7

Gauss - Hermite smearing $\tilde{\delta}(x) = \frac{1}{\sqrt{\pi}} \left(\frac{3}{2} - x^2\right) e^{-x^2}$

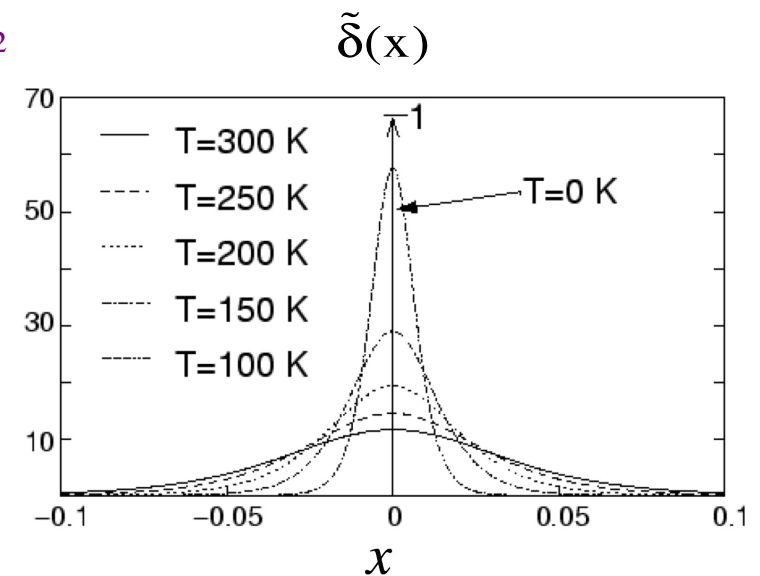
$\Rightarrow n = 4$ but occupations can be negative

...

'Cold Smearing' occopt 4/5

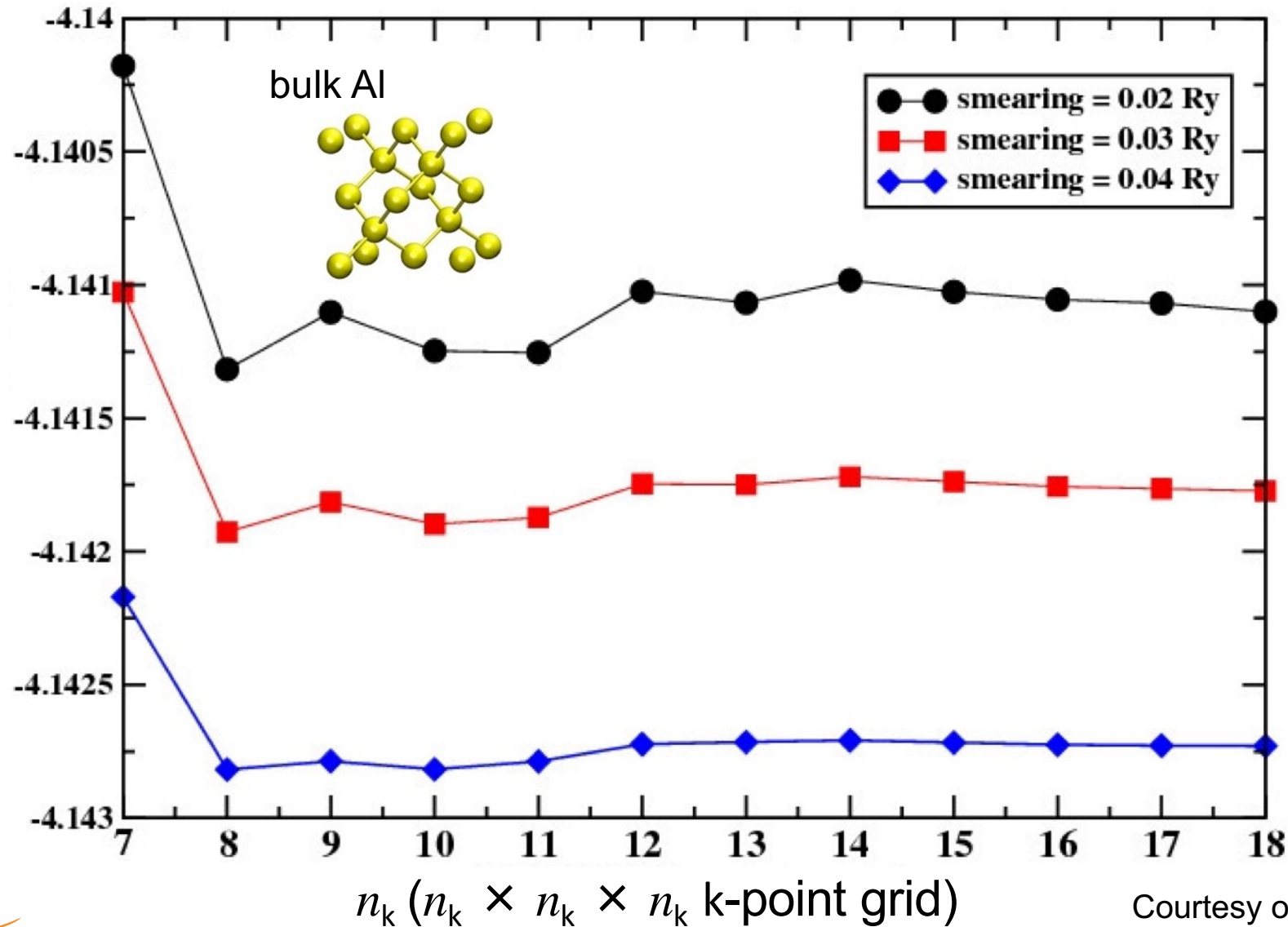
(Marzari et al, Phys. Rev. Lett. 82, 3296 (1999))

$\Rightarrow n = 3$ with positive occupations



Convergence wrt k-points and smearing

Total energy (Ry)



Courtesy of S. Narasimhan

How many k points ? Smearing width ?

Rule of thumb ! Goal : lattice parameter converged better than 0.5 %

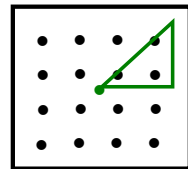
Semiconductors - Insulators # $\mathbf{k} \times N_{atoms}$ 50 ... 500

Metals # $\mathbf{k} \times N_{atoms}$ 1000 ... 2000



Use symmetries \Rightarrow integration in the **irreducible** Brillouin zone

2D Example



grid $4 \times 4 = 16$

\Downarrow
3 points in the irreducible Brillouin Zone

Smearing : depends on the density of electronic states (DOS) at the Fermi level

s-p Metal (Al, Na ...) ~ 0.04 Ha

d Metal (Cu, Ag...) ~ 0.01 Ha



magnetism needs small σ

Pseudopotentials

Core and valence electrons (I)

Core electrons occupy orbitals that are « the same » in the atomic environment or in the bonding environment

It depends on the accuracy of the calculation !

Separation between core and valence orbitals : the density...

$$n(\mathbf{r}) = \sum_i^N \psi_i^*(\mathbf{r}) \psi_i(\mathbf{r})$$
$$= \sum_{i \in \text{core}}^{N_{\text{core}}} \psi_i^*(\mathbf{r}) \psi_i(\mathbf{r}) + \sum_{i \in \text{val}}^{N_{\text{val}}} \psi_i^*(\mathbf{r}) \psi_i(\mathbf{r}) = n_{\text{core}}(\mathbf{r}) + n_{\text{val}}(\mathbf{r})$$

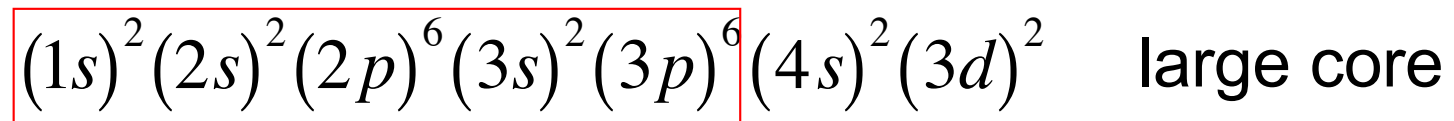
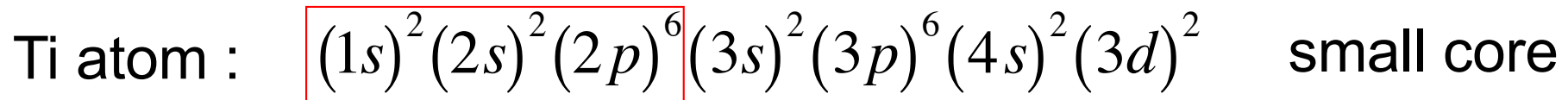
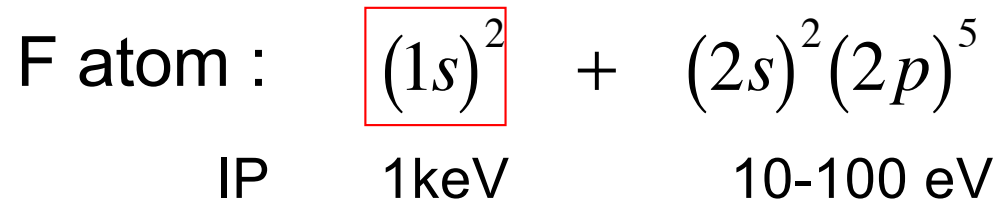
« Frozen core » for $i \in \text{core}$: $\psi_i = \psi_i^{\text{atom}}$

Small core / Large core

It depends on the target accuracy of the calculation !

(remark also valid for pseudopotentials, with similar cores)

For some elements, the core/valence partitioning is obvious, for some others, it is not.



IP 99.2 eV 43.3eV

Gd atom : small core with n=1,2,3 shells , might include 4s, 4p, and 4d in the core. 4f partially filled

Core and valence electrons (II)

Separation between core and valence orbitals : the energy ...

$$E_{\text{KS}}[\{\psi_i\}] = \sum_i \langle \psi_i | -\frac{1}{2} \nabla^2 | \psi_i \rangle + \int V_{\text{ext}}(\mathbf{r}) n(\mathbf{r}) d\mathbf{r} + \frac{1}{2} \int \frac{n(\mathbf{r}_1) n(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2 + E_{\text{xc}}[n]$$



$$E_{\text{KS}}[\{\psi_i\}] = \sum_{i \in \text{core}}^{N_{\text{core}}} \langle \psi_i | -\frac{1}{2} \nabla^2 | \psi_i \rangle + \int V_{\text{ext}}(\mathbf{r}) n_{\text{core}}(\mathbf{r}) d\mathbf{r} + \frac{1}{2} \int \frac{n_{\text{core}}(\mathbf{r}_1) n_{\text{core}}(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2$$

$$+ \sum_{i \in \text{val}}^{N_{\text{val}}} \langle \psi_i | -\frac{1}{2} \nabla^2 | \psi_i \rangle + \int V_{\text{ext}}(\mathbf{r}) n_{\text{val}}(\mathbf{r}) d\mathbf{r} + \frac{1}{2} \int \frac{n_{\text{val}}(\mathbf{r}_1) n_{\text{val}}(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2$$

$$V_{\text{ion}} = V_{\text{ext}} + n_{\text{core}}$$

$$+ \int \frac{n_{\text{core}}(\mathbf{r}_1) n_{\text{val}}(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2$$

$$+ E_{\text{xc}}[n_{\text{core}} + n_{\text{val}}]$$

Removing core electrons (I)

From the previous construction : valence orbitals must still be orthogonal to core orbitals
(=> oscillations, slope at the nucleus ...)

Pseudopotentials try to remove completely the core orbitals from the simulation

Problem with the number of nodes

This is a strong modification of the system ...

Pseudopotentials confine the strong changes within a « cut-off radius »

Removing core electrons (II)

Going from $\left(-\frac{1}{2}\nabla^2 + v\right) |\psi_i\rangle = \varepsilon_i |\psi_i\rangle$

To $\left(-\frac{1}{2}\nabla^2 + v_{ps}\right) |\psi_{ps,i}\rangle = \varepsilon_{ps,i} |\psi_{ps,i}\rangle$

Possible set of conditions (norm-conserving pseudopotentials)

NCPP - Hamann D.R., Schlüter M., Chiang C, Phys.Rev.Lett. 43, 1494 (1979)

$$\varepsilon_i = \varepsilon_{ps,i}$$

$$\psi_i(\mathbf{r}) = \psi_{ps,i}(\mathbf{r}) \quad \text{for } r > r_c$$

$$\int_{r < r_c} |\psi_i(\mathbf{r})|^2 d\mathbf{r} = \int_{r < r_c} |\psi_{ps,i}(\mathbf{r})|^2 d\mathbf{r}$$

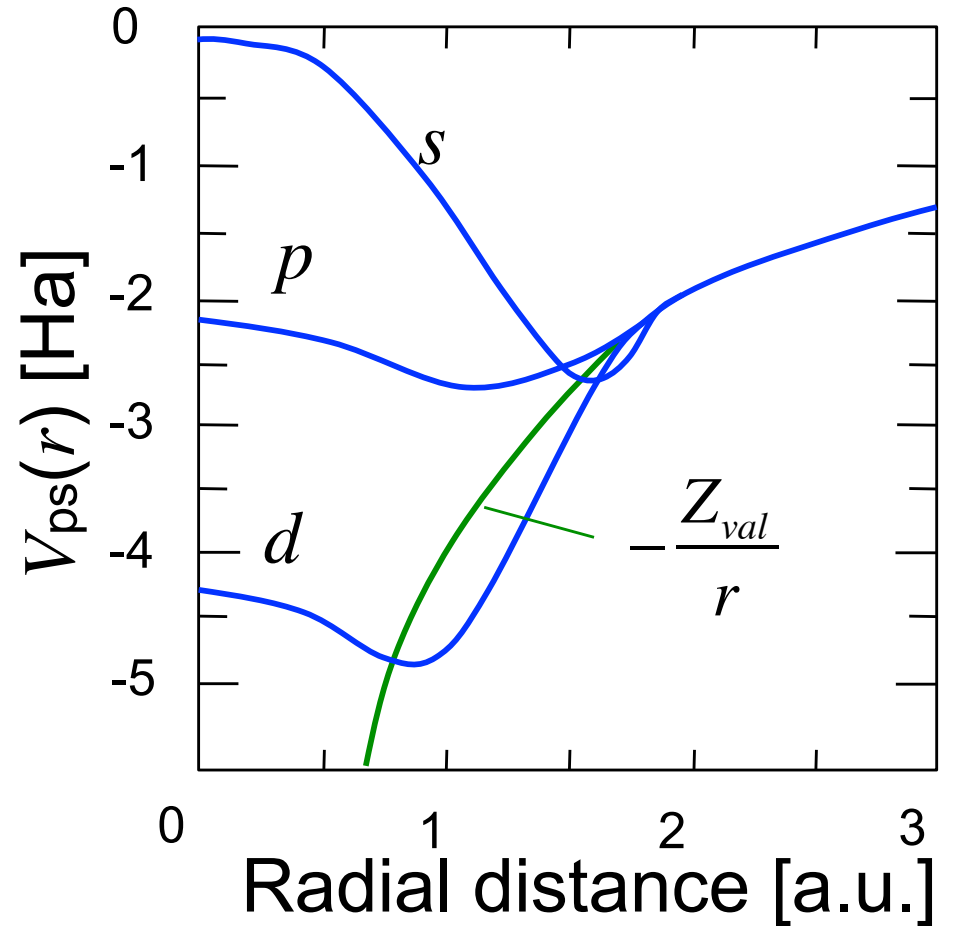
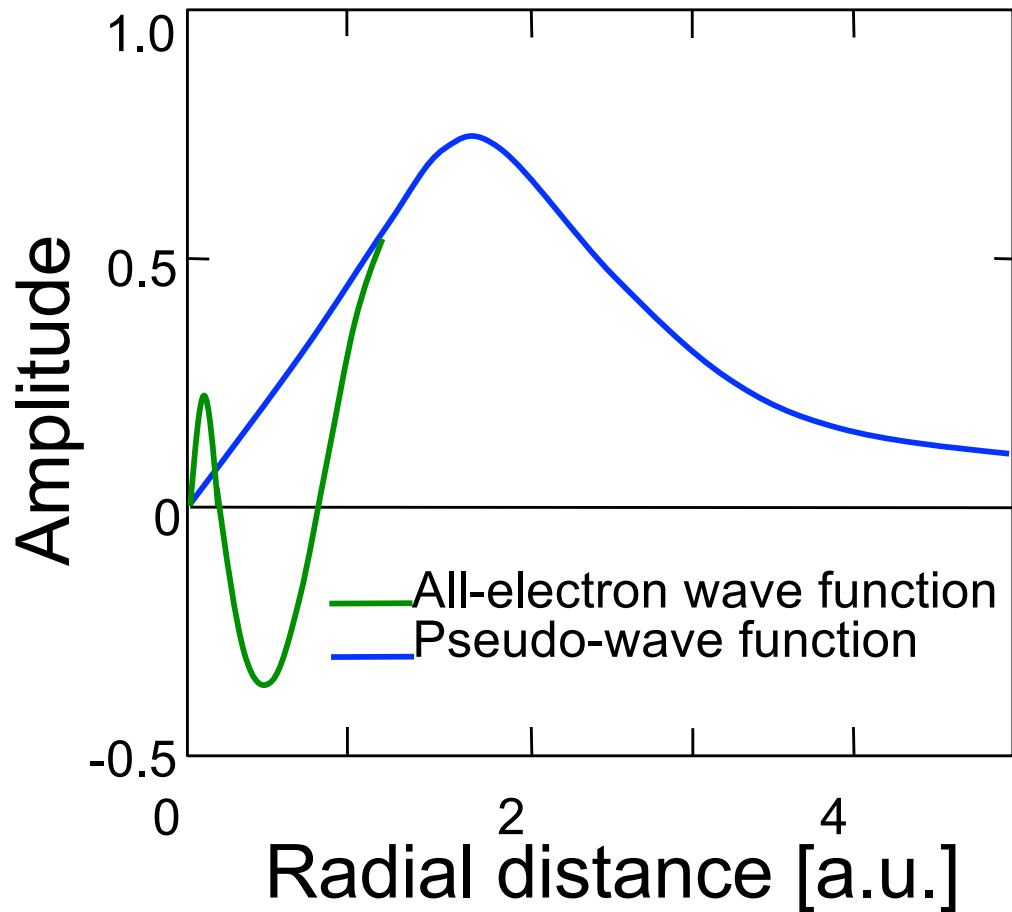
For the lowest
angular momentum
channels (s + p ... d ...f)

Generalisation : ultra-soft pseudopotentials (USPP),
projector-augmented plane waves (PAW)

Warning : be it NCPP, USPP or PAW, regions within cut-off spheres of different atoms forming solid or molecule should not overlap. Uncontrolled approximation !

Example of pseudopotential

3s Radial wave function of Si



Pseudopotentials/PAW data in ABINIT

- Preferred PAW atomic dataset table : JTH

Jollet, Torrent, Holzwarth, Computer Physics Comm. 185, 1246 (2014)

<https://www.abinit.org/psp-tables>

H																		He
Li	Be											B	C	N	O	F	Ne	
Na	Mg											Al	Si	P	S	Cl	Ar	
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr	
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe	
Cs	Ba		Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn	
Fr	Ra		Rf	Ha	Sg	Ns	Hs	Mt										
			La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu	
			Ac	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr	

■ Atomic data available
■ Atomic data non available

Also, possibility to use : GPAW table, GBRV v1.0 table, or norm-conserving pseudopotentials (e.g. ONCVSP pseudo generator), or many others

Pseudopotentials/PAW data in ABINIT

- Norm-conserving pseudos : pseudo-dojo approach

Van Setten et al , Computer Physics Comm. 226, 39 (2018)

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

The screenshot shows the Pseudo-Dojo website interface. At the top, there is a navigation bar with links for Home, F.A.Q., Contribute, and About. The main content area features a large logo for "PSEUDO DOJO" and a "Download" button. Below the logo, there are several filters: "Type" (NC (ONCVSP v0.4)), "XC" (PBE), "Accuracy" (standard), and "Format" (psp8, upf, psml, html, djrepo). A "Mean" box displays the following data:

3.13
Mean
hints tests
32.74 0.95
37.25 2.20
43.36 -0.09

The periodic table is displayed with various elements and their corresponding pseudopotential data. The data for each element is presented in a grid format, with columns for element symbol, atomic number, and pseudopotential values. The elements are color-coded by group: s-block (red), d-block (blue), p-block (green), and noble gases (purple).

Computing the forces

Computing the forces (I)

Born - Oppenheimer approx. \Rightarrow find electronic ground state in potential created by nuclei.

A starting configuration of nuclei $\{\mathbf{R}_\kappa\}$ is usually NOT in equilibrium geometry.

$$F_{\kappa,\alpha} = - \left. \frac{\partial E}{\partial R_{\kappa,\alpha}} \right|_{\{\bar{\mathbf{R}}_\kappa\}} \quad (\text{principle of virtual works})$$

Forces can be computed by finite differences.

Better approach : compute the response to a perturbation

\Rightarrow What is the energy change ?

$$\{\mathbf{R}_{\kappa,\alpha}\} \rightarrow \{\mathbf{R}_{\kappa,\alpha} + \lambda \delta \mathbf{R}_{\kappa,\alpha}\} \quad \text{Small parameter}$$

Computing the forces (II)

To simplify, let's compute the derivative of an electronic eigenvalue

Perturbation theory : Hellmann - Feynman theorem

$$\frac{d\varepsilon_n}{d\lambda} = \left\langle \psi_n^{(0)} \left| \frac{d\hat{H}}{d\lambda} \right| \psi_n^{(0)} \right\rangle$$

$$\frac{d\psi_n}{d\lambda} \text{ not needed !}$$

Application to the derivative with respect to an atomic displacement :

$$\hat{H} = \hat{T} + \hat{V}_{ext} \{ \hat{\mathbf{R}} \} \Rightarrow \frac{\partial \hat{H}}{\partial R_{\kappa, \alpha}} = \frac{\partial \hat{V}_{ext}}{\partial R_{\kappa, \alpha}}$$

$$\frac{\partial \varepsilon_n}{\partial R_{\kappa, \alpha}} = \left\langle \psi_n \left| \frac{\partial \hat{H}}{\partial R_{\kappa, \alpha}} \right| \psi_n \right\rangle = \int n(\mathbf{r}) \frac{\partial \hat{V}_{ext}(\mathbf{r})}{\partial R_{\kappa, \alpha}} d\mathbf{r}$$

Computing the forces (III)

Generalisation to density functional theory

Reminder :
$$E[\psi_i] = \sum_n \langle \psi_i | \hat{T} | \psi_i \rangle + \int n(\mathbf{r}) V_{\text{ext}}(\mathbf{r}) d\mathbf{r} + E_{\text{Hxc}}[n]$$

If change of atomic positions ...

$$V_{\text{ext}}(\vec{r}) = \sum_{k'} -\frac{Z_{k'}}{|\vec{r} - \vec{R}_{k'}|} \quad (\text{can be generalized to pseudopotential case})$$

$$\frac{\partial V_{\text{ext}}(\vec{r})}{\partial R_{k,\alpha}} = +\frac{Z_{k'}}{|\vec{r} - \vec{R}_k|^2} \cdot \frac{\partial |\vec{r} - \vec{R}_k|}{\partial R_{k,\alpha}} = -\frac{Z_{k'}}{|\vec{r} - \vec{R}_k|^3} \cdot (\vec{r} - \vec{R}_k)_\alpha$$

$$\frac{\partial E}{\partial R_{k,\alpha}} = \int n(\mathbf{r}') \frac{\partial V_{\text{ext}}(\mathbf{r}')}{\partial R_{k,\alpha}} d\mathbf{r}' = - \int \frac{n(\mathbf{r}')}{|\vec{r}' - \vec{R}_k|^3} \cdot (\vec{r}' - \vec{R}_k)_\alpha d\vec{r}'$$

Forces can be computed directly from the density !

Iterative algorithms

Algorithmics : problems to be solved

(1) Kohn - Sham equation

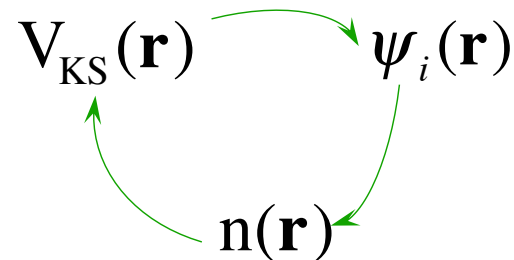
$$\underline{\underline{A}} \underline{x}_i = \lambda_i \underline{x}_i$$

$$\left[-\frac{1}{2} \nabla^2 + V_{\text{KS}}(\mathbf{r}) \right] \psi_i(\mathbf{r}) = \varepsilon_i \psi_i(\mathbf{r})$$

$\left\{ \mathbf{G}_j \right\} \quad \left\{ \mathbf{r}_j \right\}$

Size of the system	[2 atoms... 600 atoms...]	+ vacuum ?
Dimension of the vectors \underline{x}_i	300... 100 000...	(if planewaves)
# of (occupied) eigenvectors	4... 1200...	

(2) Self-consistency



(3) Geometry optimization

Find the positions $\{\mathbf{R}_\kappa\}$ of ions such that the forces $\{\mathbf{F}_\kappa\}$ vanish
 [= Minimization of energy]

Current practice : iterative approaches

The 'steepest-descent' algorithm

Forces are gradients of the energy : moving the atoms along gradients is the steepest descent of the energy surface.

=> Iterative algorithm.

Choose a starting geometry, then a parameter λ , and iterately update the geometry, following the forces :

$$\mathbf{R}_{K,\alpha}^{(n+1)} = \mathbf{R}_{K,\alpha}^{(n)} + \lambda \mathbf{F}_{K,\alpha}^{(n)}$$

Equivalent to the simple mixing algorithm of SCF (see later)

Energy+forces around equilib. geometry

Let us denote the equilibrium geometry as $\mathbf{R}_{\kappa,\alpha}^*$

Analysis of forces close to the equilibrium geometry, at which forces vanish, thanks to a Taylor expansion :

$$F_{\kappa,\alpha}(\mathbf{R}_{\kappa',\alpha'}) = F_{\kappa,\alpha}(\mathbf{R}_{\kappa',\alpha'}^*) + \sum_{\kappa',\alpha'} \frac{\partial F_{\kappa,\alpha}}{\partial \mathbf{R}_{\kappa',\alpha'}} \Big|_{\{\mathbf{R}^*\}} (\mathbf{R}_{\kappa',\alpha'} - \mathbf{R}_{\kappa',\alpha'}^*) + O(\mathbf{R}_{\kappa',\alpha'} - \mathbf{R}_{\kappa',\alpha'}^*)^2$$

Moreover, $F_{\kappa,\alpha} = -\frac{\partial E^{\text{BO}}}{\partial \mathbf{R}_{\kappa,\alpha}}$ $\frac{\partial F_{\kappa',\alpha'}}{\partial \mathbf{R}_{\kappa,\alpha}} = -\frac{\partial^2 E^{\text{BO}}}{\partial \mathbf{R}_{\kappa,\alpha} \partial \mathbf{R}_{\kappa',\alpha'}}$

Vector and matrix notation

$$\begin{array}{l} \mathbf{R}_{\kappa,\alpha}^* \rightarrow \underline{\mathbf{R}}^* \\ \mathbf{R}_{\kappa,\alpha} \rightarrow \underline{\mathbf{R}} \end{array} \quad F_{\kappa,\alpha} \rightarrow \underline{\mathbf{F}} \quad \frac{\partial^2 E^{\text{BO}}}{\partial \mathbf{R}_{\kappa,\alpha} \partial \mathbf{R}_{\kappa',\alpha'}} \Big|_{\{\mathbf{R}_{\kappa,\alpha}^*\}} \rightarrow \underline{\underline{\mathbf{H}}} \quad (\text{the Hessian})$$

Steepest-descent : analysis (I)

$$\underline{\mathbf{R}}_{K,\alpha}^{(n+1)} = \underline{\mathbf{R}}_{K,\alpha}^{(n)} + \lambda \underline{\mathbf{F}}_{K,\alpha}^{(n)}$$

Analysis of this algorithm, in the linear regime :

$$\underline{\mathbf{F}}(\underline{\mathbf{R}}) = \cancel{\underline{\mathbf{F}}(\underline{\mathbf{R}}^*)} - \underline{\mathbf{H}}(\underline{\mathbf{R}} - \underline{\mathbf{R}}^*) + \cancel{O(\underline{\mathbf{R}} - \underline{\mathbf{R}}^*)^2}$$

$$\underline{\mathbf{R}}^{(n+1)} = \underline{\mathbf{R}}^{(n)} + \lambda \underline{\mathbf{F}}^{(n)} \longrightarrow (\underline{\mathbf{R}}^{(n+1)} - \underline{\mathbf{R}}^*) = (\underline{\mathbf{R}}^{(n)} - \underline{\mathbf{R}}^*) - \lambda \underline{\mathbf{H}}(\underline{\mathbf{R}}^{(n)} - \underline{\mathbf{R}}^*)$$

$$(\underline{\mathbf{R}}^{(n+1)} - \underline{\mathbf{R}}^*) = (\underline{\mathbf{1}} - \lambda \underline{\mathbf{H}})(\underline{\mathbf{R}}^{(n)} - \underline{\mathbf{R}}^*)$$

For convergence of the iterative procedure, the "distance" between trial geometry and equilibrium geometry must decrease.

- 1) Can we predict conditions for convergence ?
- 2) Can we make convergence faster ?

Need to understand the action of the matrix (or operator)

$$\underline{\mathbf{1}} - \lambda \underline{\mathbf{H}}$$

Steepest-descent : analysis (II)

What are the eigenvectors and eigenvalues of $\underline{\underline{H}}$?

$\underline{\underline{H}}$ symmetric,
positive definite matrix

$$\left(= \frac{\partial^2 E^{BO}}{\partial R_{\kappa,\alpha} \partial R_{\kappa',\alpha'}} \Big|_{\{R_{\kappa,\alpha}^*\}} \right)$$

$\underline{\underline{H}} \underline{f}_i = h_i \underline{f}_i$ where $\{\underline{f}_i\}$ form a complete, orthonormal, basis set

Discrepancy decomposed as $(\underline{R}^{(n)} - \underline{R}^*) = \sum c_i^{(n)} \underline{f}_i$

and $(\underline{R}^{(n+1)} - \underline{R}^*) = (\underline{1} - \lambda \underline{\underline{H}}) \sum_i c_i^{(n)} \underline{f}_i = \sum_i c_i^{(n)} (1 - \lambda h_i) \underline{f}_i$

The coefficient of \underline{f}_i is multiplied by $1 - \lambda h_i$

Iteratively : $(\underline{R}^{(n)} - \underline{R}^*) = \sum_i c_i^{(0)} (1 - \lambda h_i)^{(n)} \underline{f}_i$

Steepest-descent : analysis (III)

$$\left(\underline{\mathbf{R}}^{(n)} - \underline{\mathbf{R}}^*\right) = \sum_i c_i^{(0)} (1 - \lambda h_i)^{(n)} \underline{\mathbf{f}}_i$$

The size of the discrepancy decreases if $|1 - \lambda h_i| < 1$

Is it possible to have $|1 - \lambda h_i| < 1$, for all eigenvalues ?

H positive definite \Rightarrow all h_i are positive

Yes ! If λ positive, sufficiently small ...

Steepest-descent : analysis (IV)

$$\left(\underline{\mathbf{R}}^{(n)} - \underline{\mathbf{R}}^*\right) = \sum_i c_i^{(0)} (1 - \lambda h_i)^{(n)} \underline{\mathbf{f}}_i$$

How to determine the optimal value of λ ?

The maximum of all $|1 - \lambda h_i|$ should be as small as possible.

At the optimal value of λ , what will be the convergence rate ?

(= by which factor is reduced the worst component of $\left(\underline{\mathbf{R}}^{(n)} - \underline{\mathbf{R}}^*\right)$?)

As an exercise : suppose $\left. \begin{array}{l} h_1 = 0.2 \\ h_2 = 1.0 \\ h_3 = 5.0 \end{array} \right\} \Rightarrow$ what is the best value of λ ?
+ what is the convergence rate ?

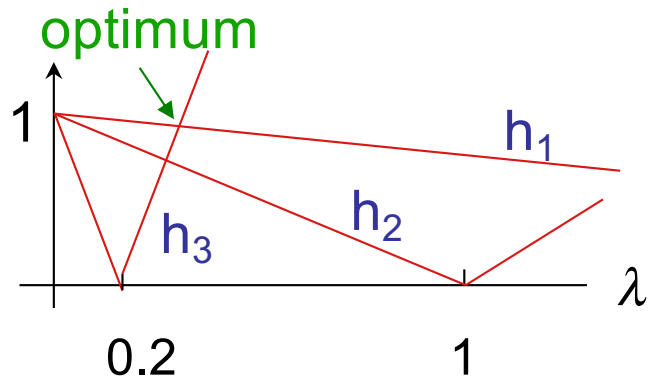
Hint : draw the three functions $|1 - \lambda h_i|$ as a function of λ . Then, find the location of λ where the largest of the three curves is the smallest.
Find the coordinates of this point.

Steepest-descent : analysis (V)

Minimise the maximum of $|1 - \lambda h_i|$

$$\begin{array}{l} h_1 = 0.2 \\ h_2 = 1.0 \\ h_3 = 5.0 \end{array} \quad \begin{array}{l} |1 - \lambda \cdot 0.2| \\ |1 - \lambda \cdot 1| \\ |1 - \lambda \cdot 5| \end{array}$$

$$\begin{array}{l} \text{optimum} \Rightarrow \lambda = 5 \\ \text{optimum} \Rightarrow \lambda = 1 \\ \text{optimum} \Rightarrow \lambda = 0.2 \end{array} \quad \left. \vphantom{\begin{array}{l} \text{optimum} \Rightarrow \lambda = 5 \\ \text{optimum} \Rightarrow \lambda = 1 \\ \text{optimum} \Rightarrow \lambda = 0.2 \end{array}} \right\} ?$$



$$\mu = |1 - \underbrace{\lambda \cdot 0.2}_{\text{positive}}| = |1 - \underbrace{\lambda \cdot 5}_{\text{negative}}|$$

$$1 - \lambda \cdot 0.2 = -(1 - \lambda \cdot 5)$$

$$2 - \lambda (0.2 + 5) = 0 \Rightarrow \lambda = 2 / 5.2$$

$$\mu = 1 - 2 \cdot (0.2 / 5.2)$$

Only ~ 8% decrease of the error, per iteration ! Hundreds of iterations will be needed to reach a reduction of the error by 1000 or more.

Note : the second eigenvalue does not play any role.

The convergence is limited by the extremal eigenvalues : if the parameter is too large, the smallest eigenvalue will cause divergence, but for that small parameter, the largest eigenvalue lead to slow decrease of the error...

The condition number

In general, $\lambda_{\text{opt}} = 2 / (h_{\text{min}} + h_{\text{max}})$
 $\mu_{\text{opt}} = 2 / [1 + (h_{\text{max}}/h_{\text{min}})] - 1 = [(h_{\text{max}}/h_{\text{min}}) - 1] / [(h_{\text{max}}/h_{\text{min}}) + 1]$

Perfect if $h_{\text{max}} = h_{\text{min}}$. Bad if $h_{\text{max}} \gg h_{\text{min}}$.

$h_{\text{max}}/h_{\text{min}}$ called the "condition" number. A problem is "ill-conditioned" if the condition number is large. It does **not** depend on the intermediate eigenvalues.

Suppose we start from a configuration with forces on the order of 1 Ha/Bohr, and we want to reach the target $1e-4$ Ha/Bohr. The mixing parameter is optimal.

How many iterations are needed ?

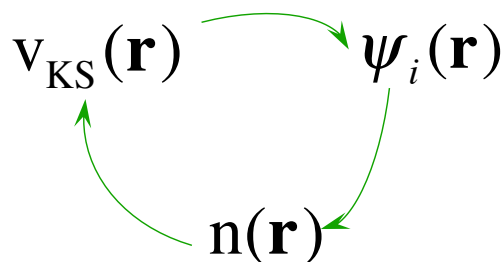
For a generic decrease factor Δ , with "n" the number of iterations.

$$\|\underline{\mathbf{F}}^{(n)}\| \approx \left(\frac{h_{\text{max}}/h_{\text{min}} - 1}{h_{\text{max}}/h_{\text{min}} + 1} \right)^n \|\underline{\mathbf{F}}^{(0)}\| \quad \Delta \approx \left(\frac{h_{\text{max}}/h_{\text{min}} - 1}{h_{\text{max}}/h_{\text{min}} + 1} \right)^n$$

$$n \approx \left[\ln \left(\frac{h_{\text{max}}/h_{\text{min}} + 1}{h_{\text{max}}/h_{\text{min}} - 1} \right) \right]^{-1} \ln \Delta \approx 0.5 (h_{\text{max}}/h_{\text{min}}) \ln \frac{1}{\Delta}$$

(The latter approximate equality supposes a large condition number)

Analysis of self-consistency



Natural iterative methodology (KS : in => out) :

$$v_{in}(\mathbf{r}) \rightarrow \psi_i(\mathbf{r}) \rightarrow n(\mathbf{r}) \rightarrow v_{out}(\mathbf{r})$$

Which quantity plays the role of a force, that should vanish at the solution ?

The difference

$$v_{out}(\mathbf{r}) - v_{in}(\mathbf{r}) \text{ (generic name : a "residual")}$$

Simple mixing algorithm
(\approx steepest - descent)

$$\underline{v}_{in}^{(n+1)} = \underline{v}_{in}^{(n)} + \lambda \left(\underline{v}_{out}^{(n)} - \underline{v}_{in}^{(n)} \right)$$

Analysis ... $\underline{v}_{out} \left[\underline{v}_{in} \right] = \underline{v}_{out} \left[\underline{v}^* \right] + \frac{\delta \underline{v}_{out}}{\delta \underline{v}_{in}} \left(\underline{v}_{in} - \underline{v}^* \right)$

\searrow
H

Like the steepest-descent algorithm, this leads to the requirement to minimize $|1 - \lambda h_i|$ where h_i are eigenvalues of

$$\frac{\delta \underline{v}_{out}}{\delta \underline{v}_{in}}$$

(actually, the dielectric matrix)

Sherbrooke, June 6-7, 2022

Modify the condition number (II)

$$\underline{\mathbf{R}}^{(n+1)} = \underline{\mathbf{R}}^{(n)} + \lambda \left(\underline{\mathbf{H}}^{-1} \right)_{\text{approx}} \underline{\mathbf{F}}^{(n)}$$

$$\underline{\mathbf{F}}(\underline{\mathbf{R}}) = -\underline{\mathbf{H}}(\underline{\mathbf{R}} - \underline{\mathbf{R}}^*) \implies \left(\underline{\mathbf{R}}^{(n+1)} - \underline{\mathbf{R}}^* \right) = \left(1 - \lambda \left(\underline{\mathbf{H}}^{-1} \right)_{\text{approx}} \underline{\mathbf{H}} \right) \left(\underline{\mathbf{R}}^{(n)} - \underline{\mathbf{R}}^* \right)$$

Notes : 1) If approximate inverse Hessian perfect, optimal geometry is reached in one step, with $\lambda = 1$.

Steepest-descent NOT the best direction.

2) Non-linear effects not taken into account. For geometry optimization, might be quite large. Even with perfect hessian, need 5-6 steps to optimize a water molecule.

3) Approximating inverse hessian by a multiple of the unit matrix is equivalent to changing the λ value.

4) Eigenvalues and eigenvectors of $\left(\underline{\mathbf{H}}^{-1} \right)_{\text{approx}} \underline{\mathbf{H}}$ govern the convergence : the condition number can be changed.

$\left(\underline{\mathbf{H}}^{-1} \right)_{\text{approx}}$ often called a "**pre-conditioner**".

5) Generalisation to other optimization problems is trivial.

(The Hessian is referred to as the Jacobian if it is not symmetric.)

Modify the condition number (III)

Approximate Hessian can be generated on a case-by-case basis.

Selfconsistent determination of the Kohn-Sham potential :

Jacobian = dielectric matrix.

Lowest eigenvalue **close to 1**.

Largest eigenvalue :

= **1.5 ... 2.5** for small close-shell molecules, and small unit cell solids

(Simple mixing will sometimes converge with parameter set to 1 !)

= the **macroscopic dielectric constant** (e.g. 12 for silicon),

for larger close-shell molecules and large unit cell insulators,

= **diverge** for large-unit cell metals, or open-shell molecules !

Model dielectric matrices known for rather homogeneous systems.

Knowledge of approx. macroscopic dielectric constant

=> efficient preconditioner

Work in progress for inhomogeneous systems

(e.g. metals/vacuum systems).

Advanced algorithms : using the history

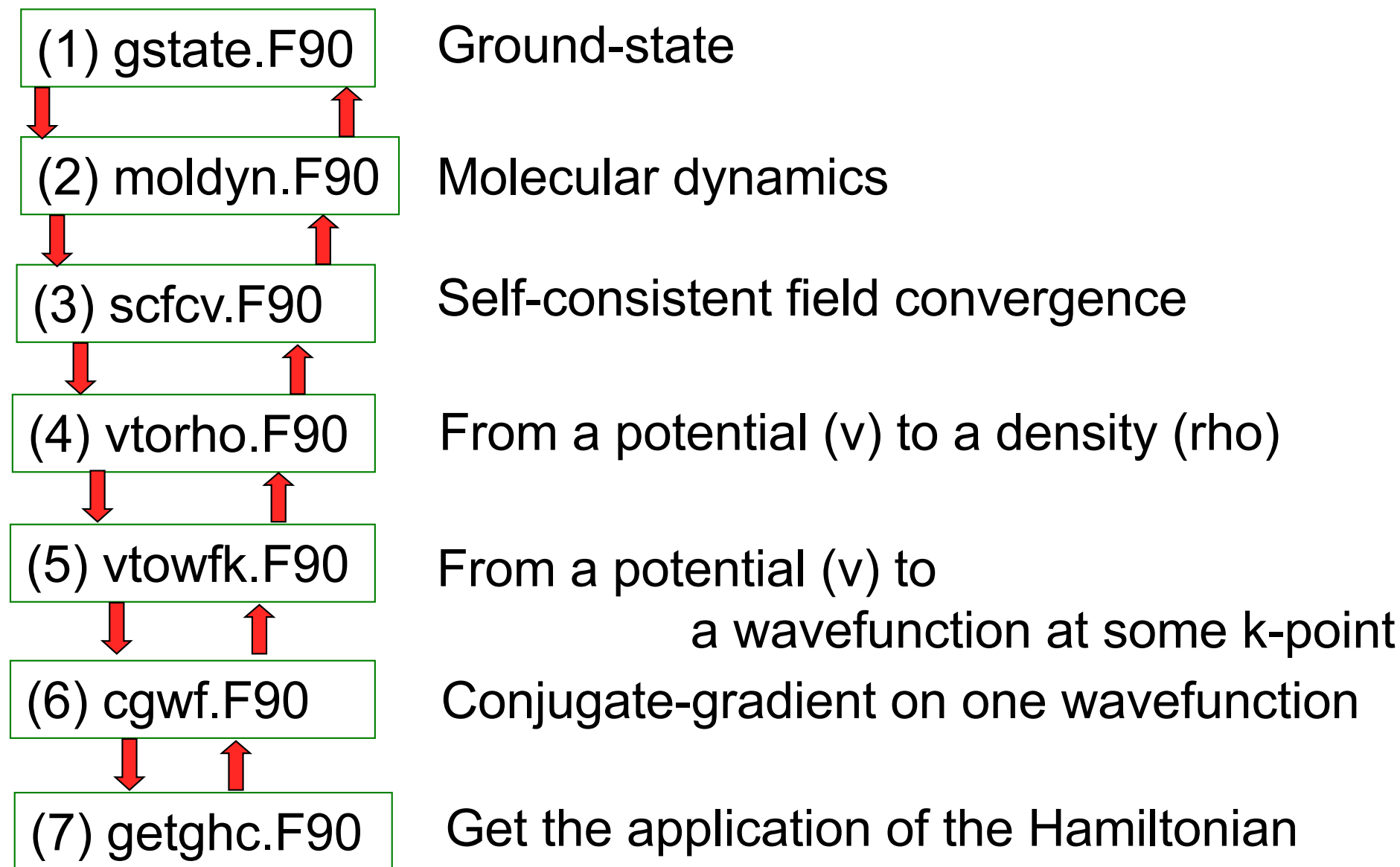
Instead of using only previously computed forces, take into account past forces for past positions

Large class of methods :

- Broyden (quasi-Newton-type),
- Davidson,
- conjugate gradients,
- Lanczos ...

Approximate Hessian can be combined with usage of history

Stages in the main processing unit



In practice ...

(1) Kohn - Sham equation
$$\left[-\frac{1}{2}\nabla^2 + V_{\text{KS}}(\mathbf{r}) \right] \psi_i(\mathbf{r}) = \varepsilon_i \psi_i(\mathbf{r})$$

Details are usually hidden to the user `nline`

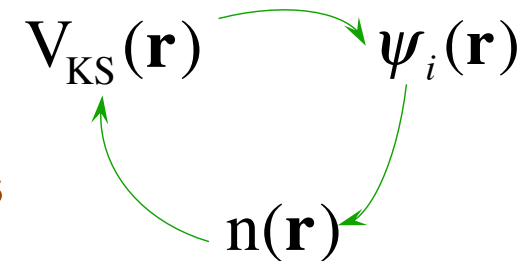
Note that scaling with size of system is quadratic or even cubic

(2) Self-consistency

Target tolerance `toldfe, toldff, tolrff, tolvrs`

+ Maximal number of loops `nstep`

Preconditioner `diemac, diemix, ...`



(3) Geometry optimization / molecular dynamics

Find the positions $\{\mathbf{R}_k\}$ of ions such that the forces $\{\mathbf{F}_k\}$ vanish

Target tolerance `tolmxf`

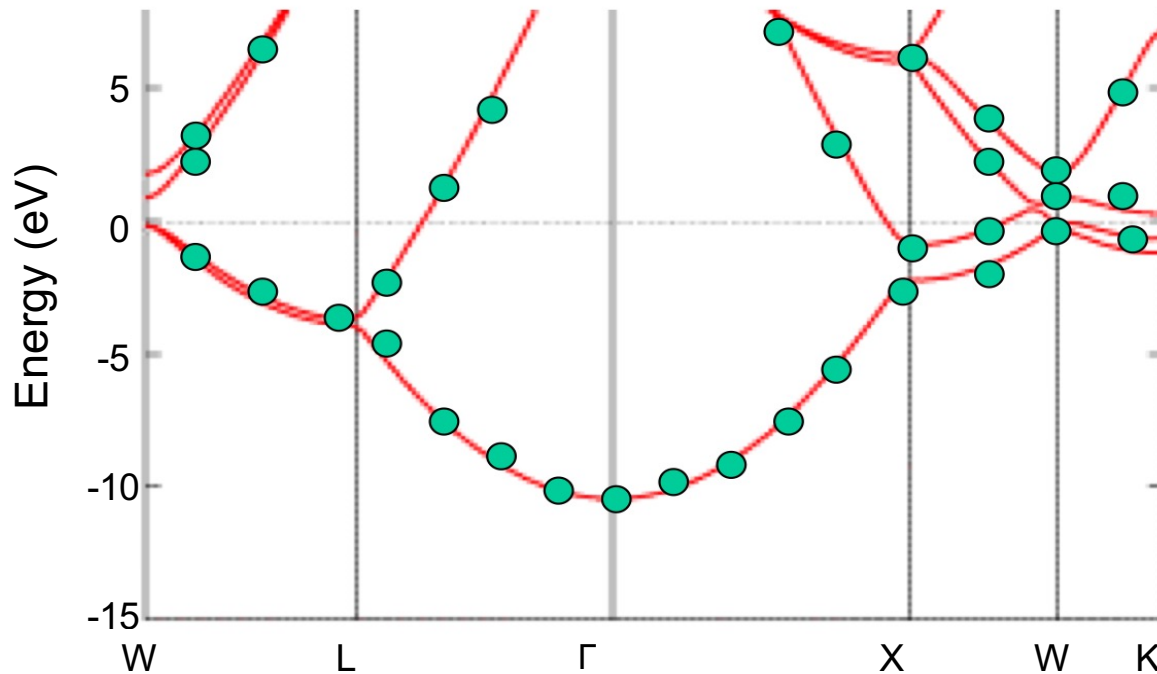
+ Maximal number of loops `ntime`

Algorithm `ionmov`

Computing band structure and density of states

Non-self consistent calculations

- Once the density has been determined self-consistently, it is possible to compute the eigenenergies/eigenfunctions rapidly for a large number of wavevectors, at fixed KS potential
- Band structure :
non self-consistent calculation (`iscf -2`)
k-points along high-symmetry directions (`kptopt<0;kptbounds;ndivk`).



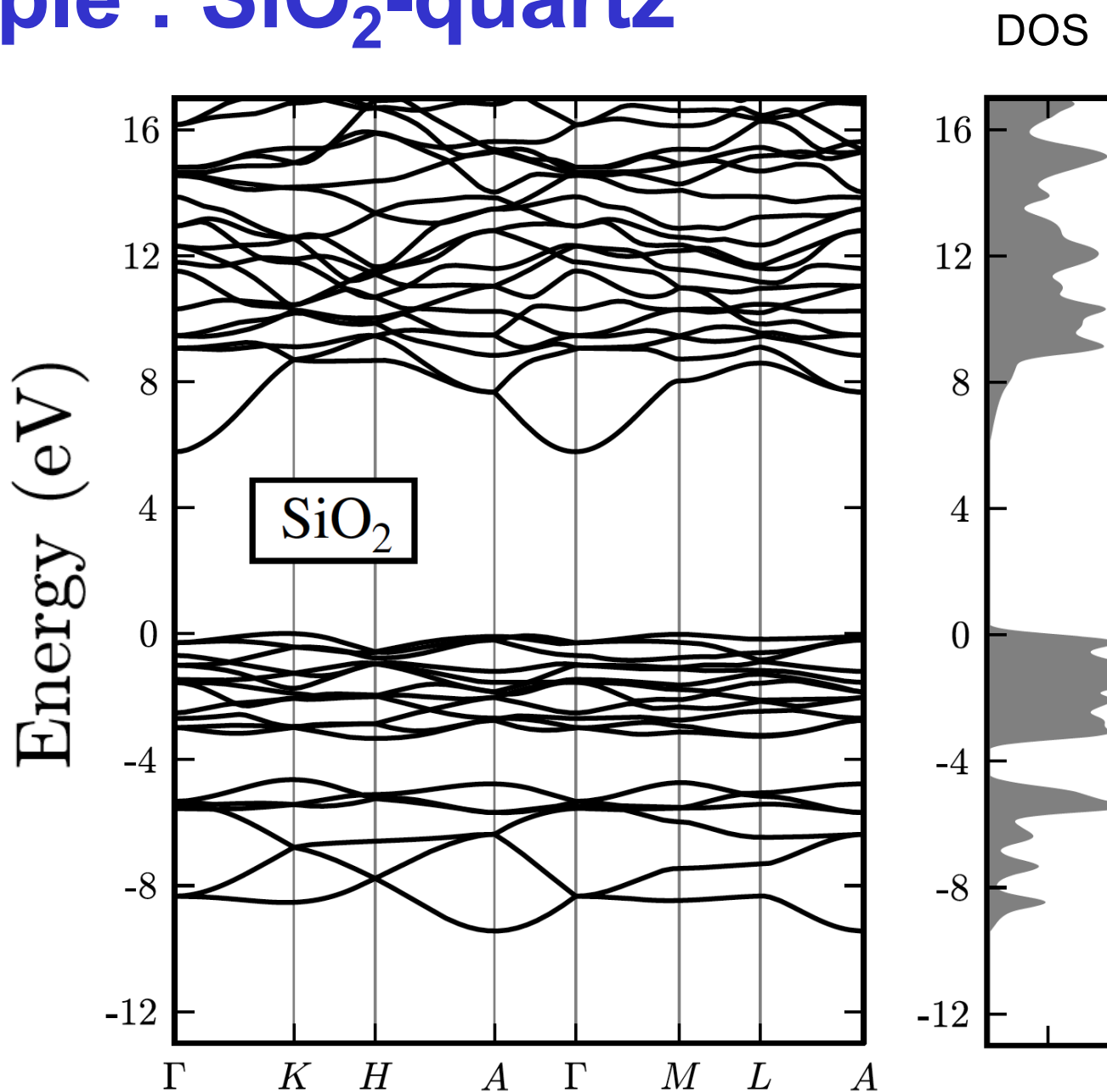
Density of states

- Density of states (DOS) defined as number of states available in energy range E to $E+dE$:

$$g(E) = \sum_n \frac{1}{\Omega_{0k}} \int_{\Omega_{0k}} \delta(E - \varepsilon_{nk}) d\mathbf{k}$$

- Recipe : determine ε_{nk} on a grid of k-points in the BZ using a non self-consistent procedure (`iscf -3`).
- The δ -function is approximated by smeared-out function (typically a Gaussian) with a width σ (`prtdos 1`).
- Very fine grid of k-points needed to get sharp features accurately. For faster convergence, use tetrahedron method to interpolate between k-points (`prtdos 2`).

Example : SiO₂-quartz

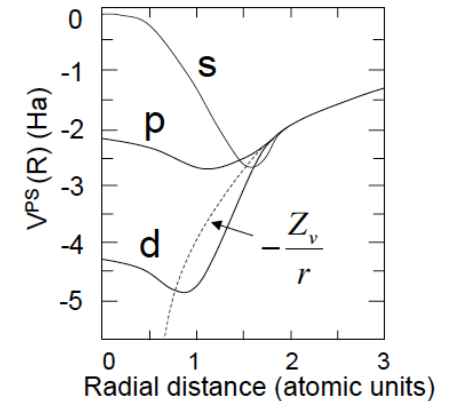
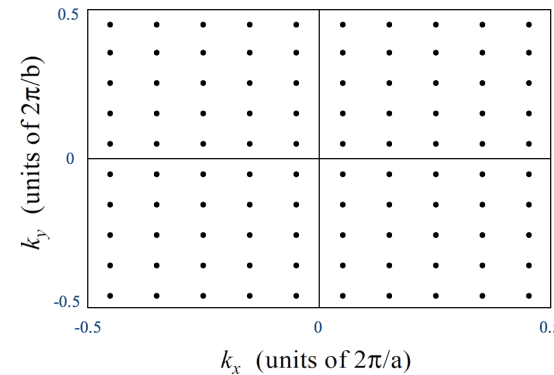


Summary

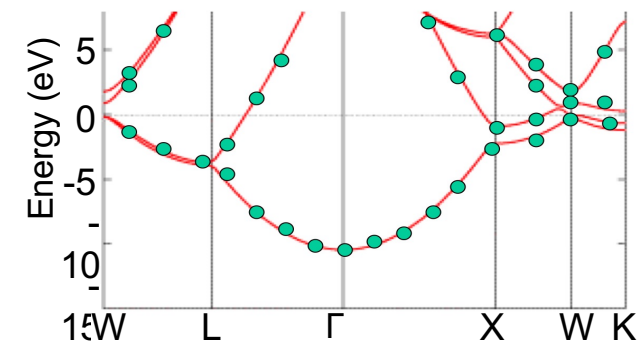
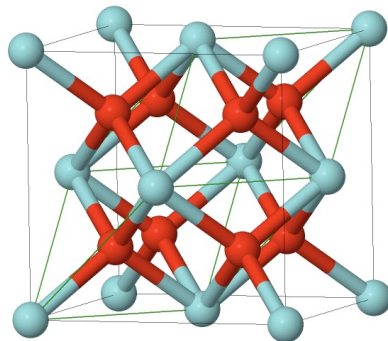
- Plane waves basis set
- Brillouin zone integration
- PW need pseudopotentials
- Easy computation of forces

$$\psi_{\mathbf{k}}(\mathbf{r}) = (N\Omega_0)^{-1/2} \sum_{\mathbf{G}} u_{\mathbf{k}}(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G})\mathbf{r}}$$

Homogeneous sampling of the Brillouin zone



=> relaxation of geometry, or molecular dynamics

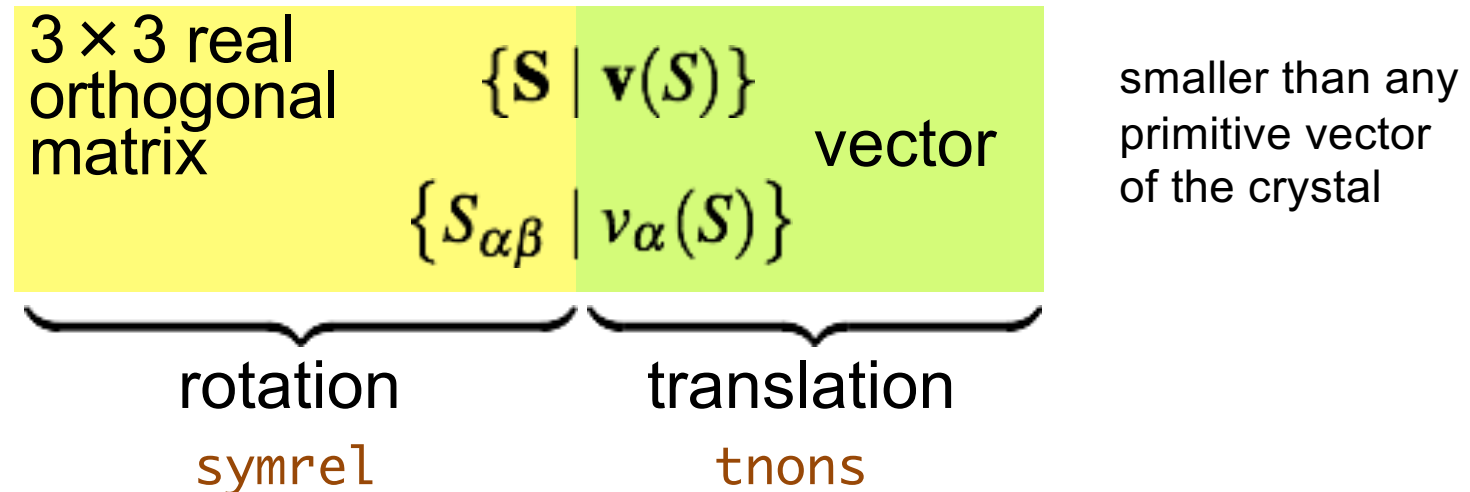


Beyond the basics



Symmetries in ABINIT

- Seitz notation for the symmetry operations of crystal :



- Applied to the equilibrium position vector of atom κ relative to the origin of the cell τ_κ , this symmetry transforms it as:

$$\{\mathbf{S} \mid \mathbf{v}(\mathcal{S})\} \tau_\kappa = \mathbf{S}\tau_\kappa + \mathbf{v}(\mathcal{S}) = \tau_{\kappa'} + \mathbf{R}^a$$

$$\{S_{\alpha\beta} \mid v_\alpha(\mathcal{S})\} \tau_{\kappa\alpha} = S_{\alpha\beta} \tau_{\kappa\alpha} + v_\alpha(\mathcal{S}) = \tau_{\kappa'\alpha} + R_\alpha^a$$

where R^a belongs to the real space lattice.

Documentation : abipy galleries

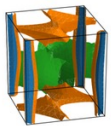
Web site

- Plot gallery
- Flow gallery

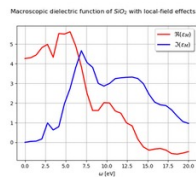
<https://github.com/abinit/abipy>

<http://abinit.github.io/abipy/gallery/index.html>

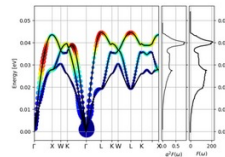
http://abinit.github.io/abipy/flow_gallery/index.html



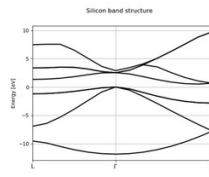
MgB2 Fermi surface



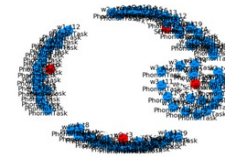
Dielectric function with LFE



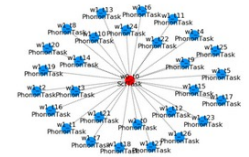
Eliashberg function



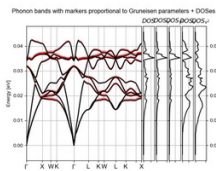
Band structure plot



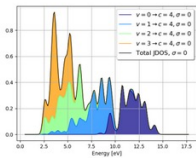
Flow to analyze the convergence of phonons in metals wrt ngkpt and tsmear



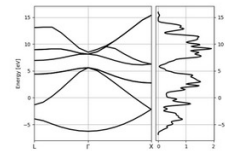
Flow for phonons with DFPT



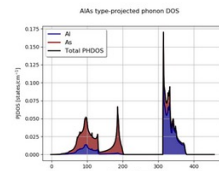
Grüneisen parameters



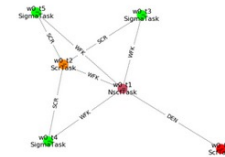
Joint Density of States



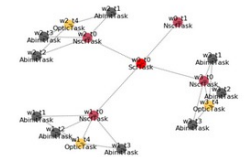
Bands + DOS



Projected phonon DOS

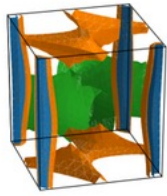


G0W0 Flow with convergence study wrt nband

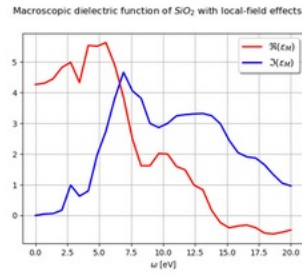


Optic Flow

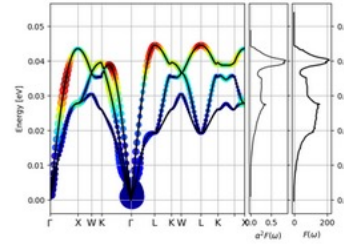
Documentation : abipy galleries



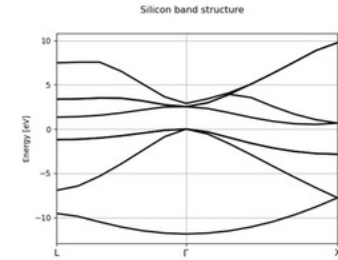
MgB2 Fermi surface



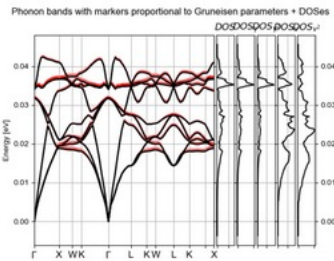
Dielectric function with LFE



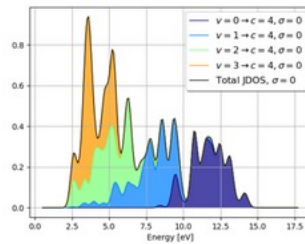
Eliashberg function



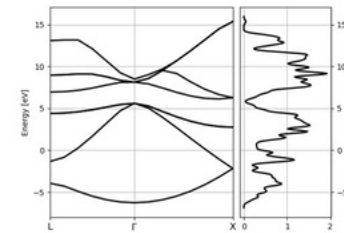
Band structure plot



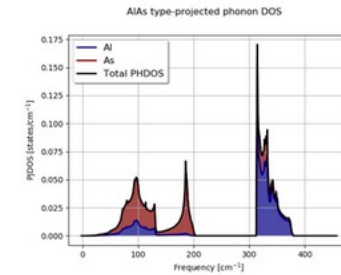
Gruneisen parameters



Joint Density of States

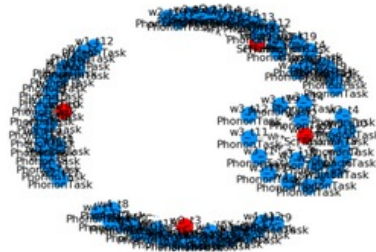


Bands + DOS

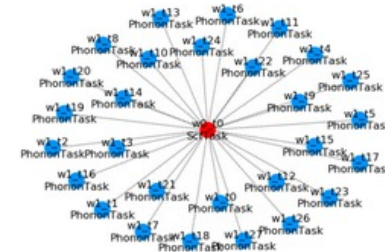


Projected phonon DOS

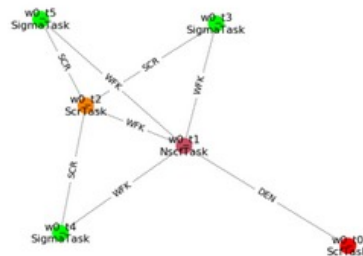
Documentation : abipy galleries



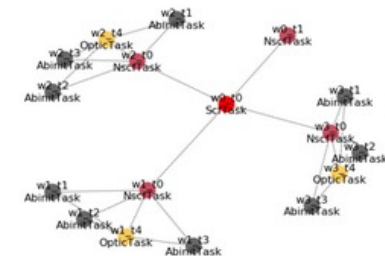
Flow to analyze the convergence of phonons in metals wrt ngkpt and tsmear



Flow for phonons with DFPT



G0W0 Flow with convergence study wrt nband



Optic Flow

Documentation : abitutorials

Web site

<https://github.com/abinit/abitutorials>

- Jupyter notebook : very fast execution of tutorial, so student can grasp the whole story, then come back to details later
- Easier if familiarized with python
- Recent, 7 lessons available

Abinit + AbiPy Lessons

- [The H₂ molecule](#)
- [Crystalline silicon](#)
- [Phonons, dielectric tensor and Born effective charges from DFPT](#)
- [G₀W₀ band structure](#)
- [Bethe-Salpeter equation and excitonic effects](#)
- [E-PH self-energy and T-dependent band structures](#)
- [Phonon linewidths and Eliashberg function of Al](#)

Documentation : central ABINIT doc

Web site <https://docs.abinit.org> Based on markdown+mkdocs

- User's guide + Installations notes
- List of topics, input variables, tutorial (>30 lessons)
- Theory documents including bibliography
- >1000 example input files + reference output (from test set)

User Guide NewUser

User Guide Topics Variables Lessons Theory Developers About

User Guide

NewUser

Abinit

Aim

Anaddd

Cut3d

Fold2Bloch

Mrgddb

Mrgscr

Optic

Respfn

Spacegroup

Installation ▾

New user help file

This page gives a beginner's introduction to the ABINIT resources, the package, and the main ABINIT applications.

Foreword

The ABINIT project is a group effort of dozens of people worldwide, whose central outcome is the main ABINIT application, delivered with many other files in the ABINIT package. The ABINIT project includes also resources provided on the [ABINIT Web site](#) and the [Github organization](#).

Before reading the present page, and get some grasp about the main ABINIT application, you should get some theoretical background. In case you have already used another electronic structure code, or a quantum chemistry code, it might be sufficient to read the introduction of [this](#)

Table of contents

- Foreword
- Introduction
- The main executable: abinit
- Other programs in the package
- Other resources outside the ABINIT package
- Input variables to abinit
- Output files
- What does the code do?
- Versioning logic