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 (2018) :

>2000 registered people on the forum
>800 kLines of F90 + many python scripts (abipy)
about 50 contributors to ABINITv8

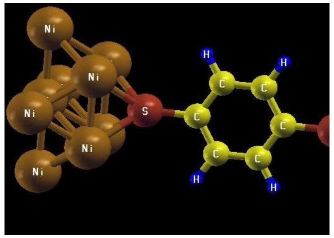
last release v8.8.2 used
in this school
http://www.abinit.org

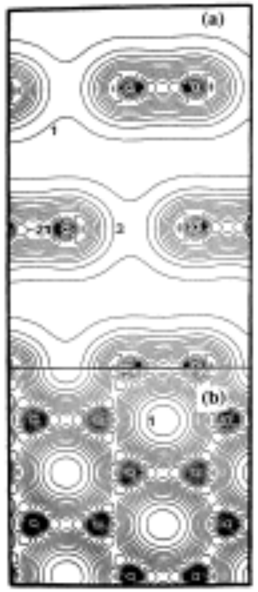
Available freely
(GPL, like Linux).

Properties from DFT+MBPT+ …

Computation of ...

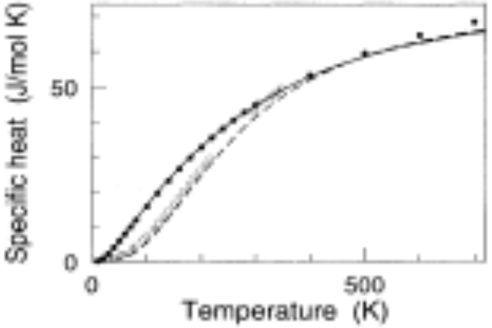
interatomic distances, angles, total energies
electronic charge densities, electronic energies

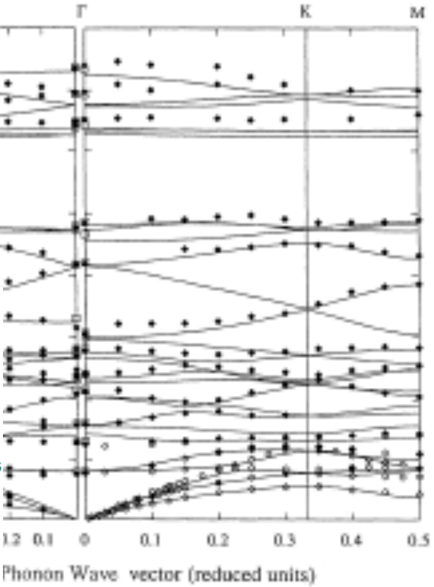


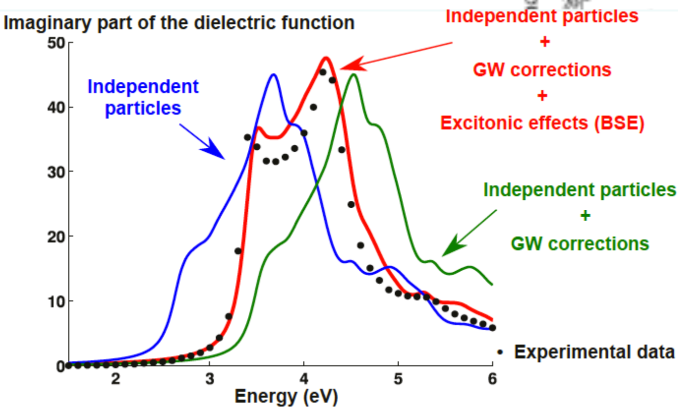


A basis for the computation of ...

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







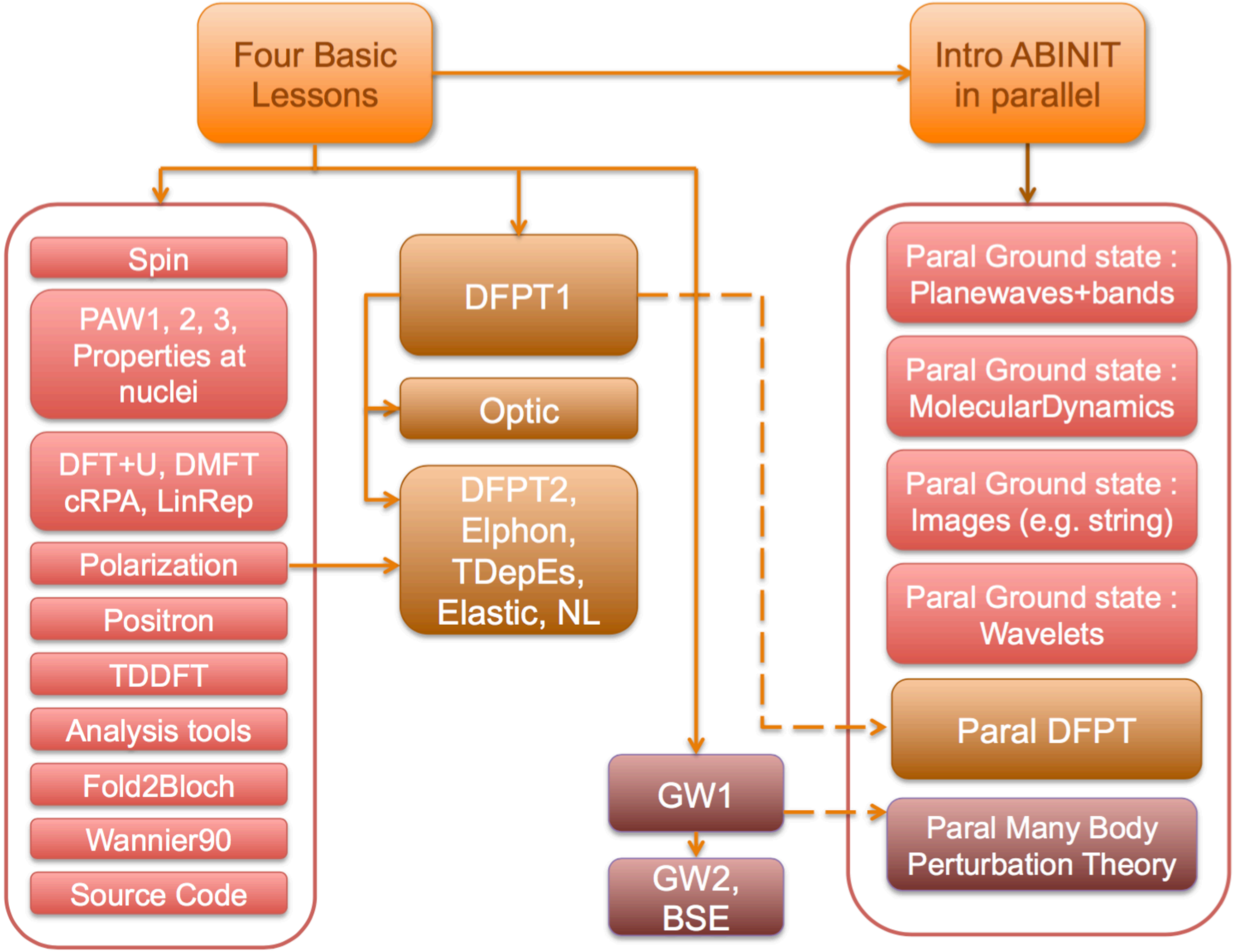
Basic Documentation

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://forum.abinit.org

ABINIT tutorial : layout + dependencies



Four Basic Lessons

Intro ABINIT in parallel

- Spin
- PAW1, 2, 3, Properties at nuclei
- DFT+U, DMFT cRPA, LinRep
- Polarization
- Positron
- TDDFT
- Analysis tools
- Fold2Bloch
- Wannier90
- Source Code

DFPT1

Optic

DFPT2, Elphon, TDepEs, Elastic, NL

GW1

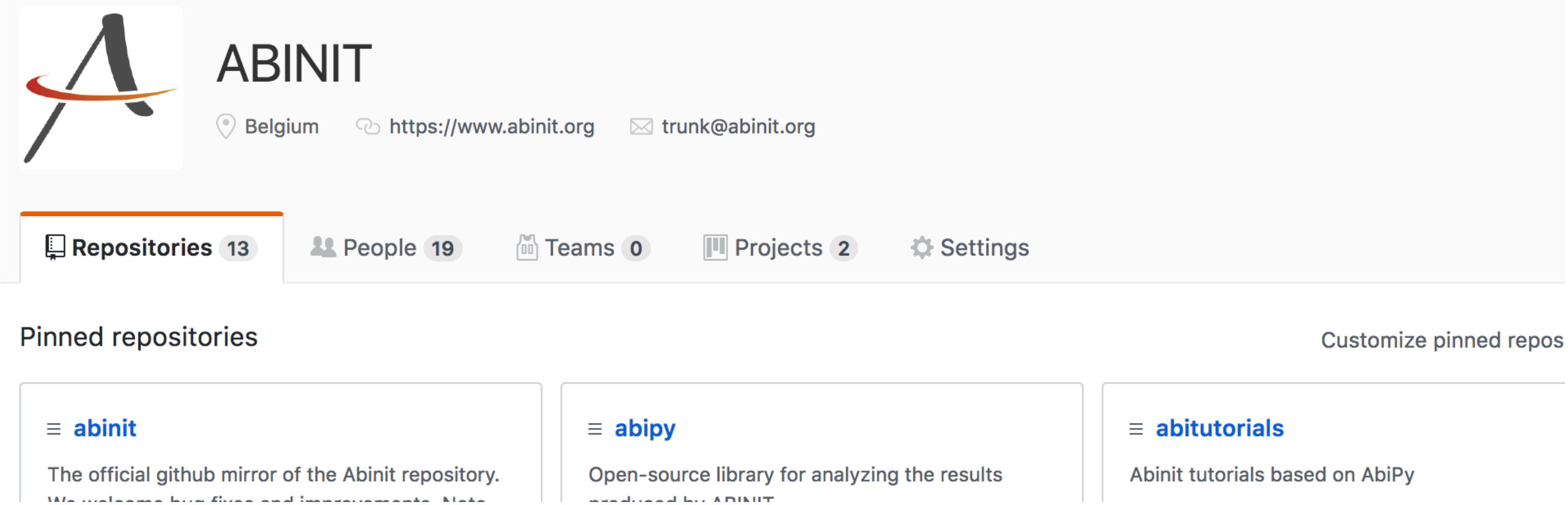
GW2, BSE

- Paral Ground state : Planewaves+bands
- Paral Ground state : MolecularDynamics
- Paral Ground state : Images (e.g. string)
- Paral Ground state : Wavelets
- Paral DFPT
- Paral Many Body Perturbation Theory

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

Pinned repositories

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 …

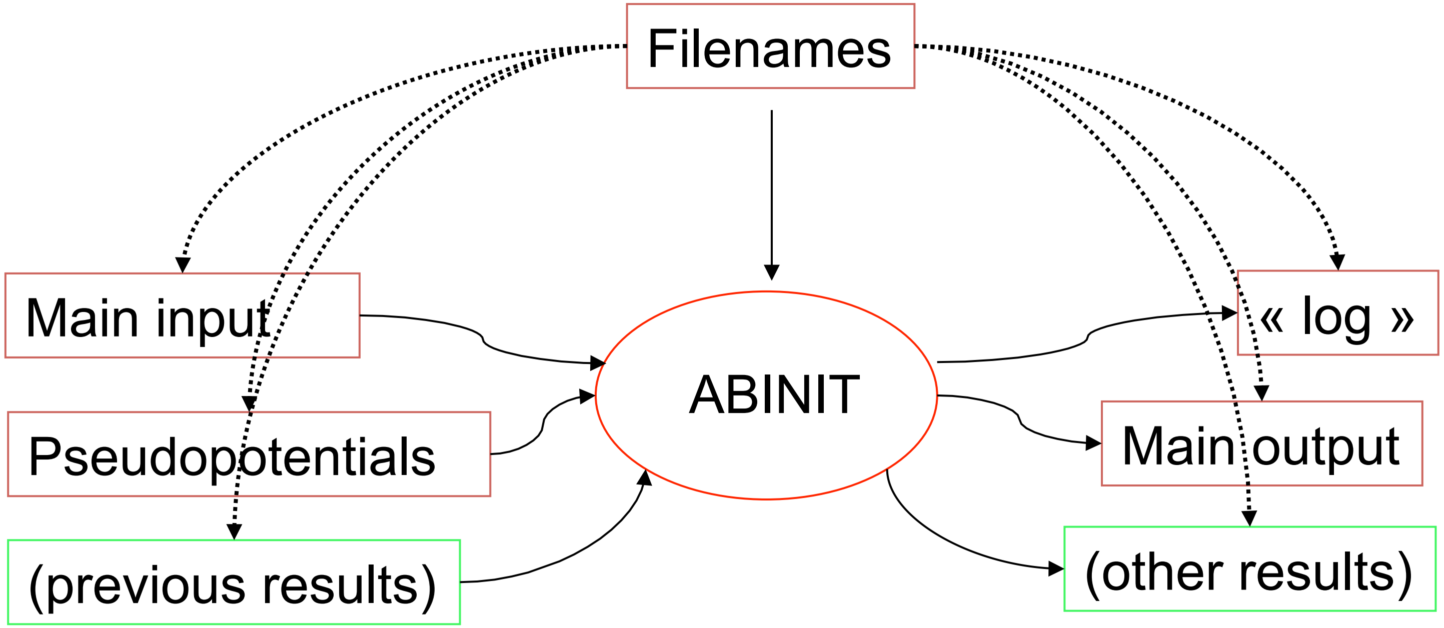
Representation of mathematical formalism
with a Plane Wave basis set :
- wavefunctions
- density, potential

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 :

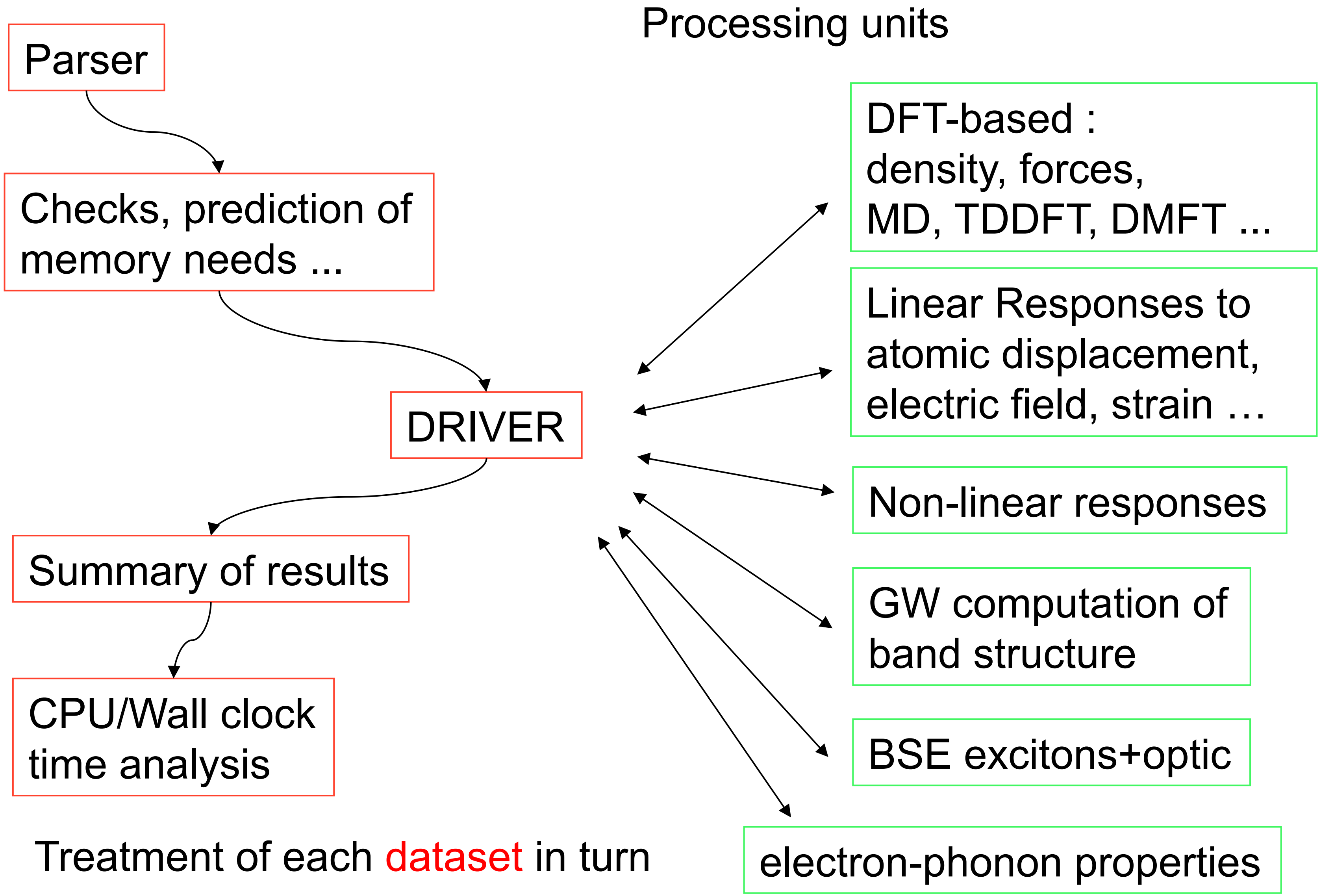
log, main output, energy derivatives (_DDB), ... – text files

density (_DEN), potential (_POT), wavefunctions (_WFK), ... – binary F90 files

or similar files in netCDF (_DEN.nc, _POT.nc, _WFK.nc)

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

ABINIT : the pipeline and the driver



Processing units

Parser

Checks, prediction of memory needs ...

DRIVER

Summary of results

CPU/Wall clock time analysis

DFT-based :
density, forces,
MD, TDDFT, DMFT ...

Linear Responses to atomic displacement, electric field, strain …

Non-linear responses

GW computation of band structure

BSE excitons+optic

electron-phonon properties

Treatment of each dataset in turn

Basic 'files' file : delivers filenames

h2.in	Name of input file
h2.out	Name of main output file
h2i	'Root' name for possibly other input files
h2o	'Root' name for possibly other output files
h2	'Root' name for temporary files
hydrogen.hgh	Name for the pseudopotential file for atoms of type 1
[oxygen.pspnc	Name for the pseudopotential file for atoms of type 2]
[92u.psp	Name for the pseudopotential file for atoms of type 3]

Made of at least 6 lines (more if > 1 type of atoms) with one name/address specified on each of these lines.

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 atom. Pseudopotential(s)
                        # mentioned in "filenames" file must correspond
                        # to type(s) of atom. Here, the only type is Hydrogen.
# Definition of the atoms
natom 2                 # Two atoms
typat 1 1               # Both are of type 1, that is, Hydrogen
xcart                   # This keyword indicate that location of the atoms
                        # will follow, one triplet of number 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>
```

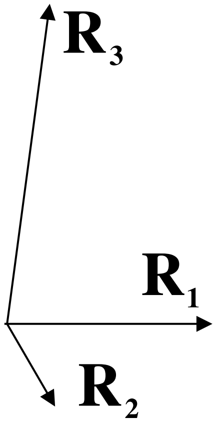
Metadata … to be ignored in the tutorial !

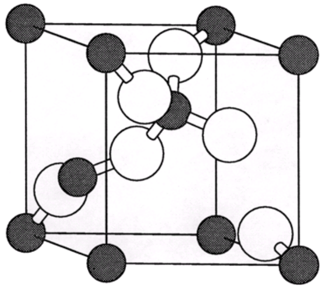
Specification of the atomic geometry

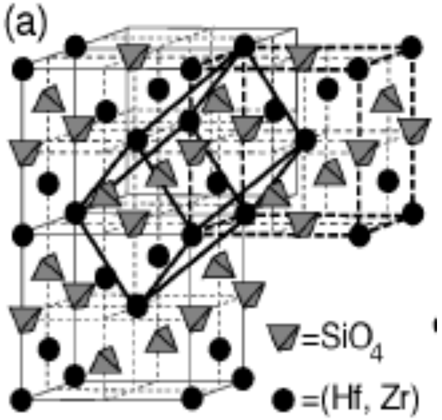
ABINIT : a periodic code

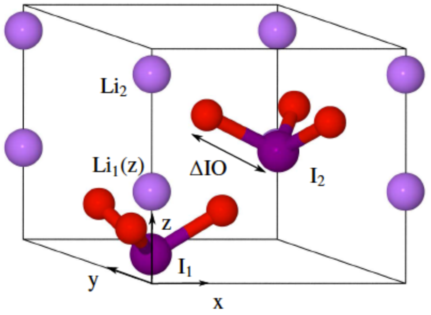
Plane waves $e^{i\mathbf{Kr}}$: particularly simple and efficient (when used with pseudopotentials), but infinite spatial extent.

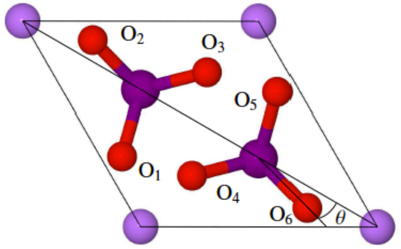
Cannot use a finite set of planewaves for finite systems !
Need periodic boundary conditions.
Primitive vectors $\mathbf{R}_j$, primitive cell volume Ω_0





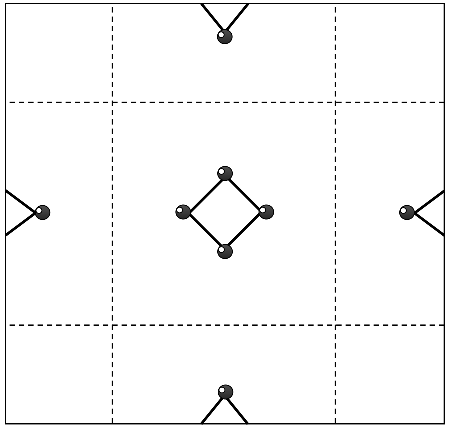




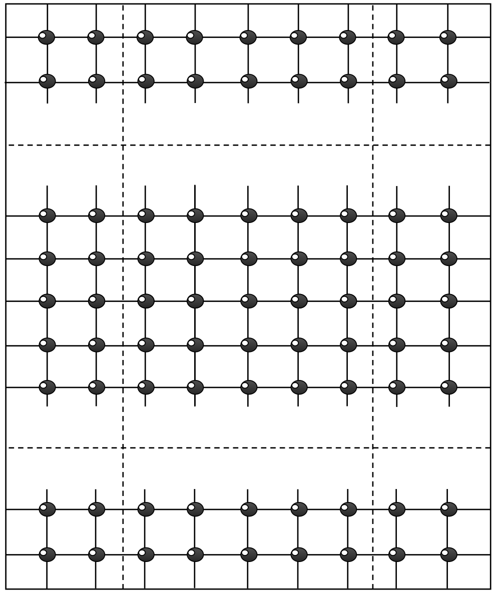


OK for crystalline solids
But : finite systems, surfaces, defects, polymers, nanosystems ... ?

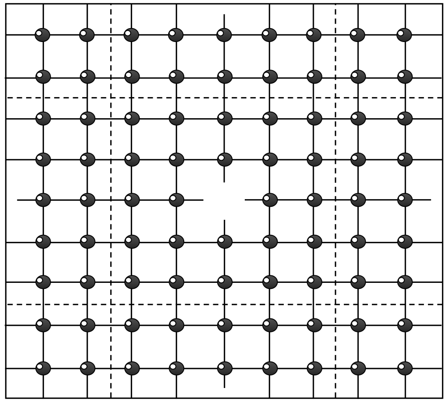
Solution : the supercell technique



Molecule,
cluster



Surface : treatment
of a slab
Interface

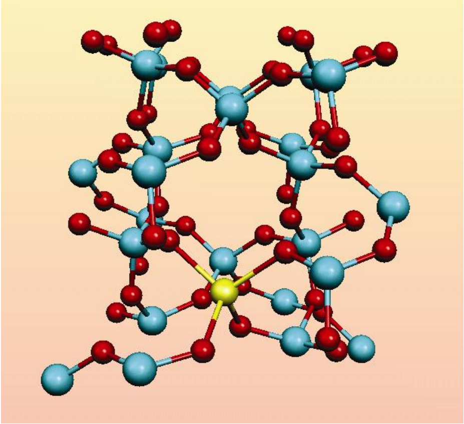


Point defect in a bulk solid

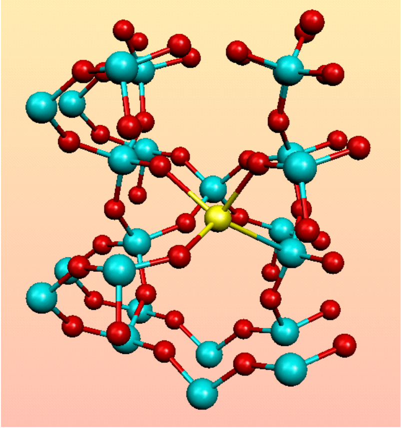
The supercell must be sufficiently big : convergence study

Examples of defects SiO_2-quartz : Pb

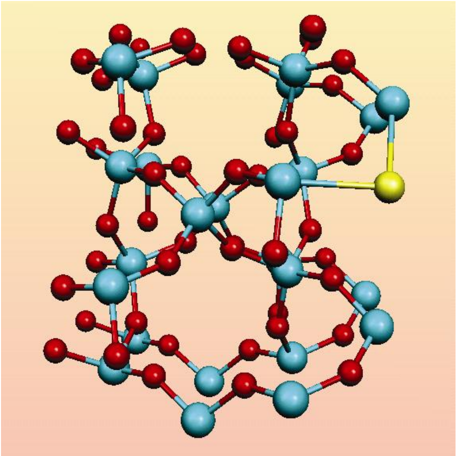
72-atom supercell of quartz



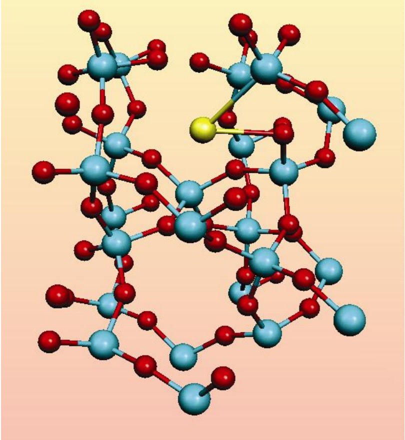
Pb^{Si}



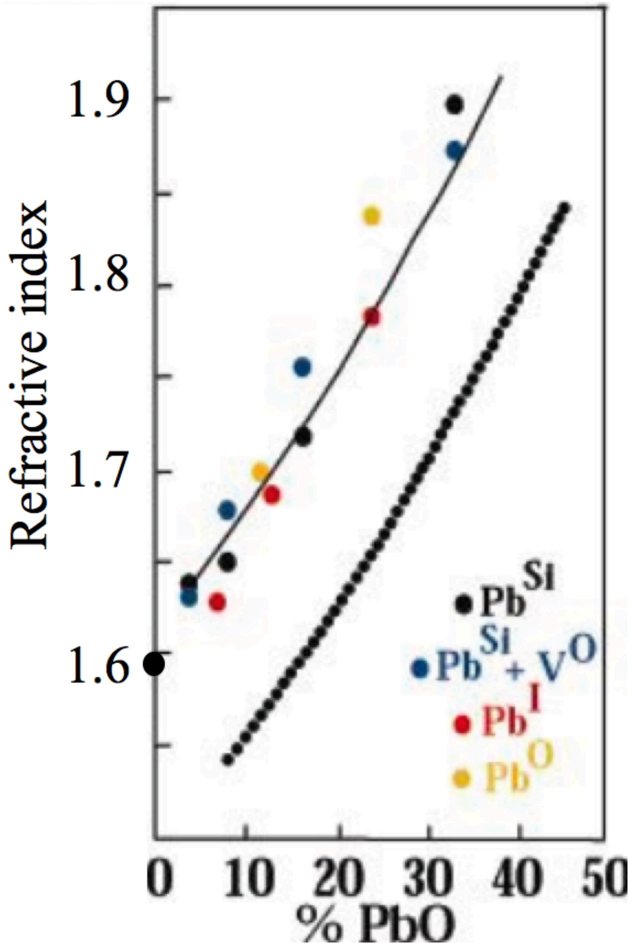
$Pb^{Si} + V^{O}$



Pb^{O}



Pb^{I}

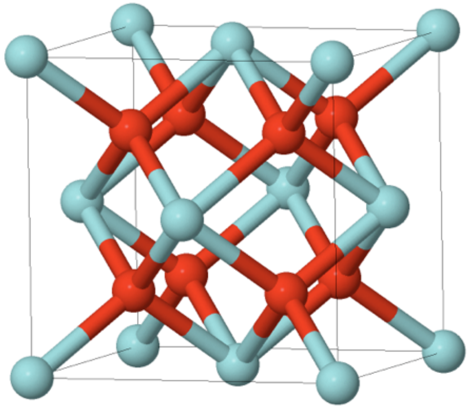


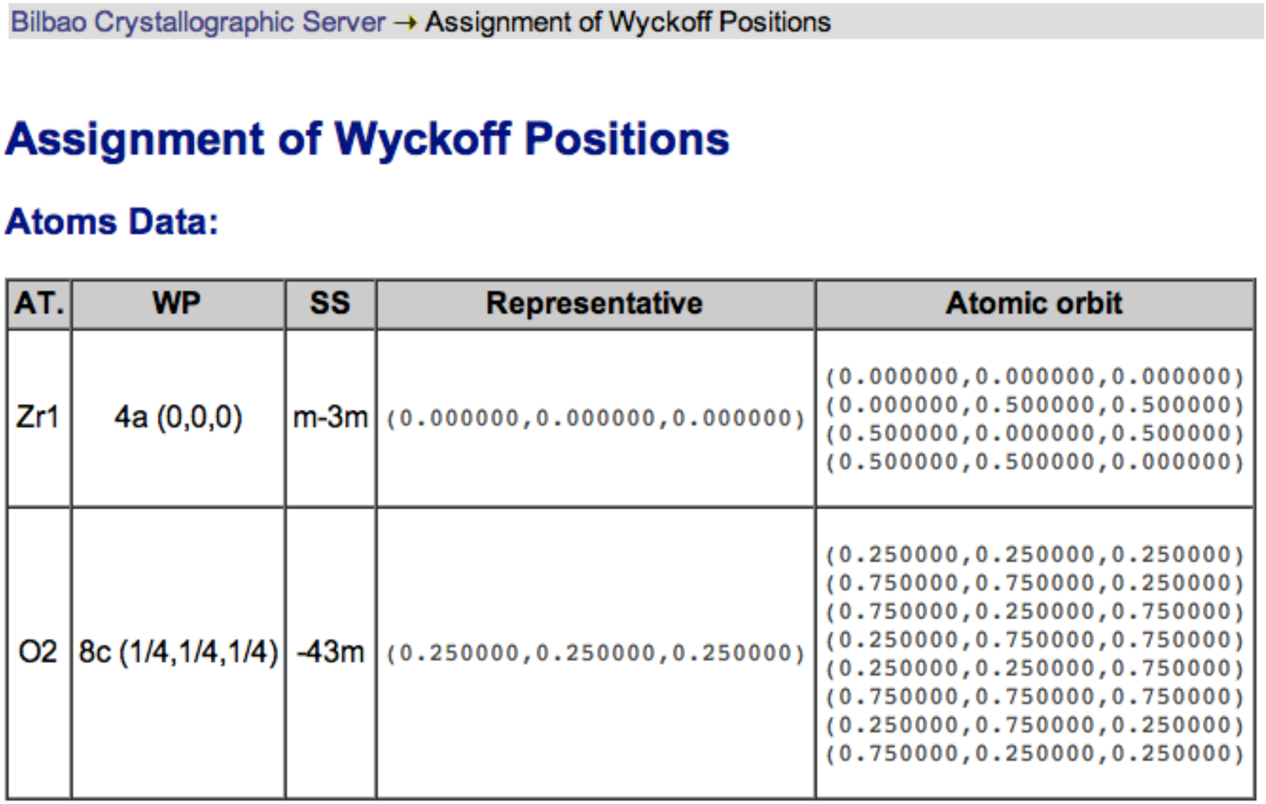
Comparison with amorphous SiO_2

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 → scalecart
- number of atoms → natom
- reduced coordinates → xred (initial guess …)
  … or cartesian coordinates → xcart (in Bohr) / xangst (in Å)
- type of atoms → typat
- space group → spgroup + natrd
  … or number of symmetries → nsym
  + symmetry operations → symrel + tnons

Example : cubic zirconium dioxide





Bilbao Crystallographic Server → Assignment of Wyckoff Positions

Assignment of Wyckoff Positions

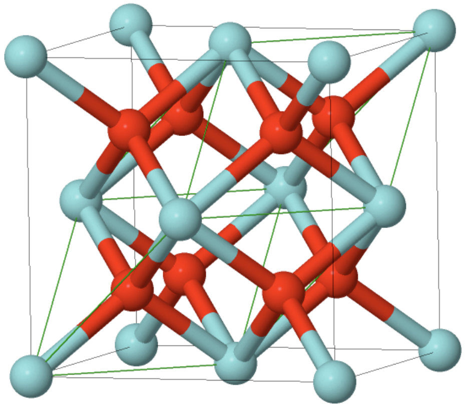
Atoms Data:

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)

Fm-3m
a=5.010Å
b=5.010Å
c=5.010Å
α=90.0°
β=90.0°
γ=90.0°

Face-centered cubic, with three atoms per primitive cell

Example : cubic zirconium dioxide



```
natom 3
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
typat 1 2 2
xred 3*0.0  3*0.25 3*0.75
=> symmetries are found automatically
```

OR

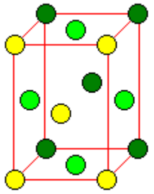
```
natom 3
acell 3*5.01 Angst
rprim  0.0 0.5 0.5
       0.5 0.0 0.5
       0.5 0.5 0.0
typat 1 2 2
spgroup 225       natrd 2
xred 3*0.0  3*0.25          => the set of atoms is completed automatically
```

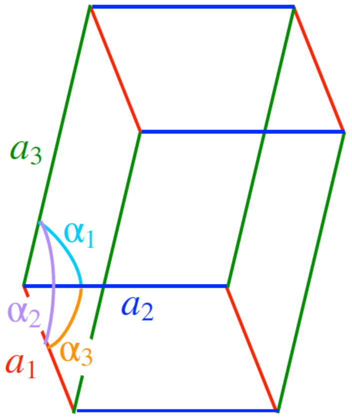
Primitive vectors in ABINIT (rprimd)

- R_i(j) → rprimd(j,i)=scalecart(j)×rprim(j,i)×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
```

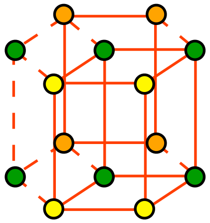
face-centered orthorhombic





```
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



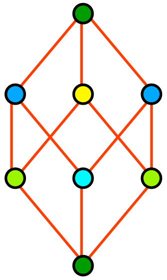
- a_i → acell(i) / α_i → angdeg(i)

```
 acell    9.5  9.5  10.0
angdeg    120 90 90
```

hexagonal

```
 acell    9.0  9.0  9.0
angdeg    48 48 48
```

trigonal



Symmetries in ABINIT

- Seitz notation for the symmetry operations of crystal :

3×3 real orthogonal matrix

vector

$$\underbrace{\{\mathbf{S}}_{\text{rotation}} \mid \underbrace{\mathbf{v}(S)\}}_{\text{translation}}$$

$$\underbrace{\{S_{\alpha\beta}}_{\text{rotation}} \mid \underbrace{v_\alpha(S)\}}_{\text{translation}}$$

`symrel` `tnons`

smaller than any primitive vector of the crystal

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

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

$$\{S_{\alpha\beta} \mid v_\alpha(S)\}\, \tau_{\kappa\alpha} = S_{\alpha\beta}\tau_{\kappa\alpha} + v_\alpha(S) = \tau_{\kappa'\alpha} + R^a_\alpha$$

where R^a belongs to the real space lattice.

The plane wave basis set

$$\boxed{\psi_{\mathbf{k}}(\mathbf{r}) = \left(N\Omega_0\right)^{-1/2} \sum_{\mathbf{G}} \mathrm{u}_{\mathbf{k}}(\mathbf{G})\, \mathrm{e}^{\mathrm{i}(\mathbf{k}+\mathbf{G})\mathbf{r}}}$$

A reminder : basic equations in DFT

Solve self-consistently the Kohn-Sham equation

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

$$\begin{tikzcd}
\hat{H} \arrow[rr, bend left] & & \psi_n(\mathbf{r}) \arrow[dl, bend left] \\
& \rho(\mathbf{r}) \arrow[ul, bend left] &
\end{tikzcd}$$

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

or minimize

$$E_{el}\{\psi\} = \sum_{n}^{occ} \left\langle \psi_n \middle| \hat{T} + \hat{V} \middle| \psi_n \right\rangle + E_{Hxc}[\rho]$$

with

$$\hat{V}(\vec{r}) = \sum_{a\kappa} -\frac{Z_\kappa}{\left|\vec{r} - \vec{R}_\kappa^a\right|}$$

Periodic system : wavevectors

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

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

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

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

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

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

$$u_{\mathbf{k}}(\mathbf{G}) = \frac{1}{\Omega_o} \int_{\Omega_o} e^{-i\mathbf{G}\mathbf{r}}\, u_{\mathbf{k}}(\mathbf{r})\, d\mathbf{r}$$ (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 $\frac{(\mathbf{k}+\mathbf{G})^2}{2}$

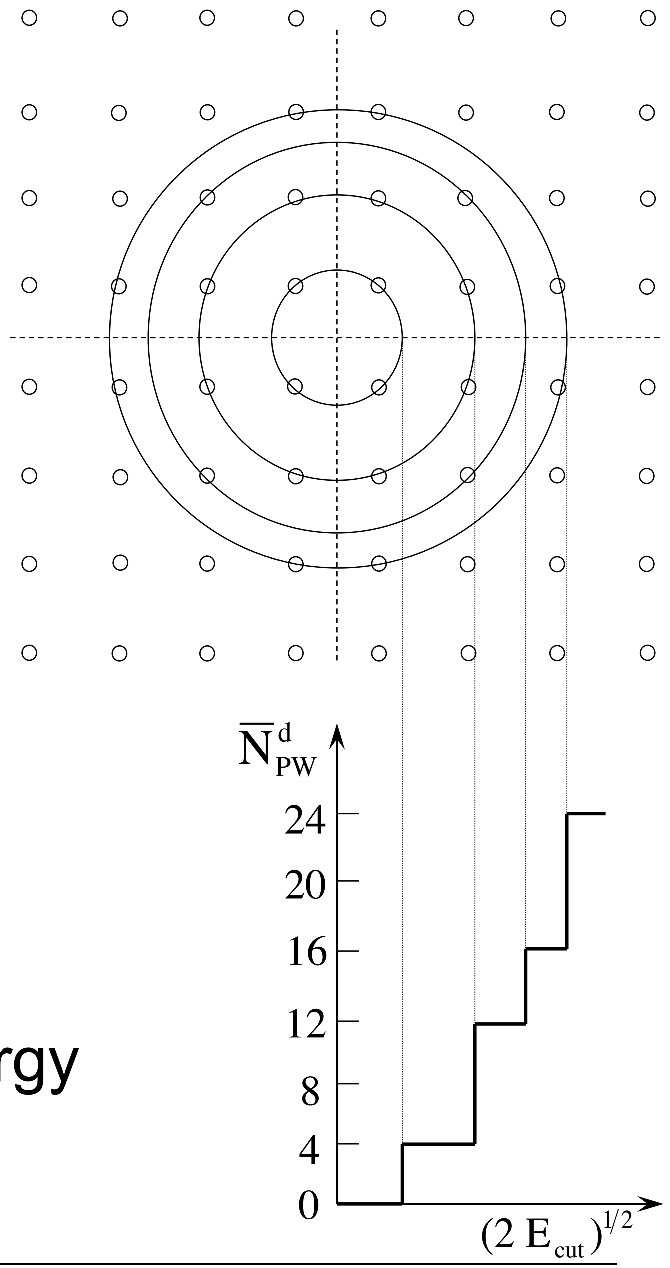
Selection of plane waves determined by a cut-off energy E_{cut}

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

`ecut`

Number of planewaves

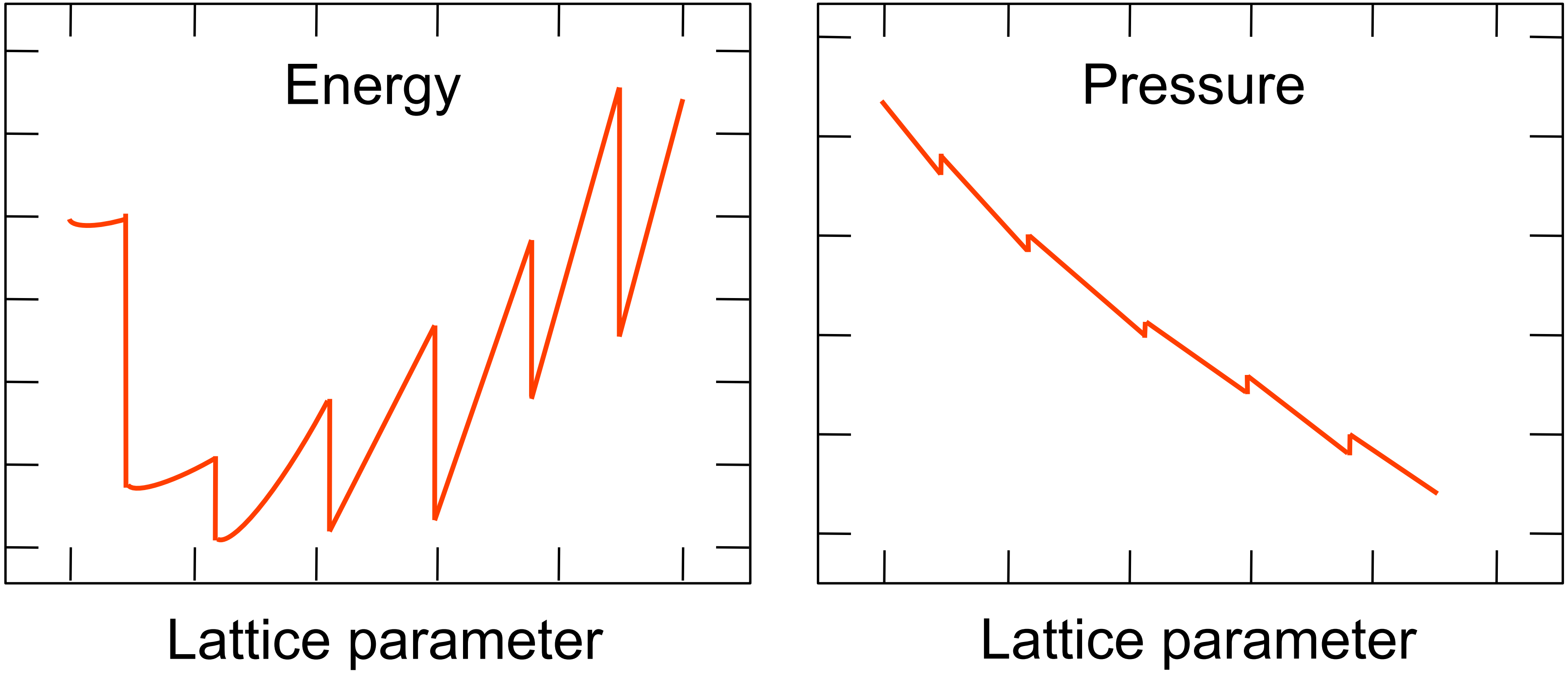
Number of plane waves
= function of the
kinetic energy cut-off
… not continuous



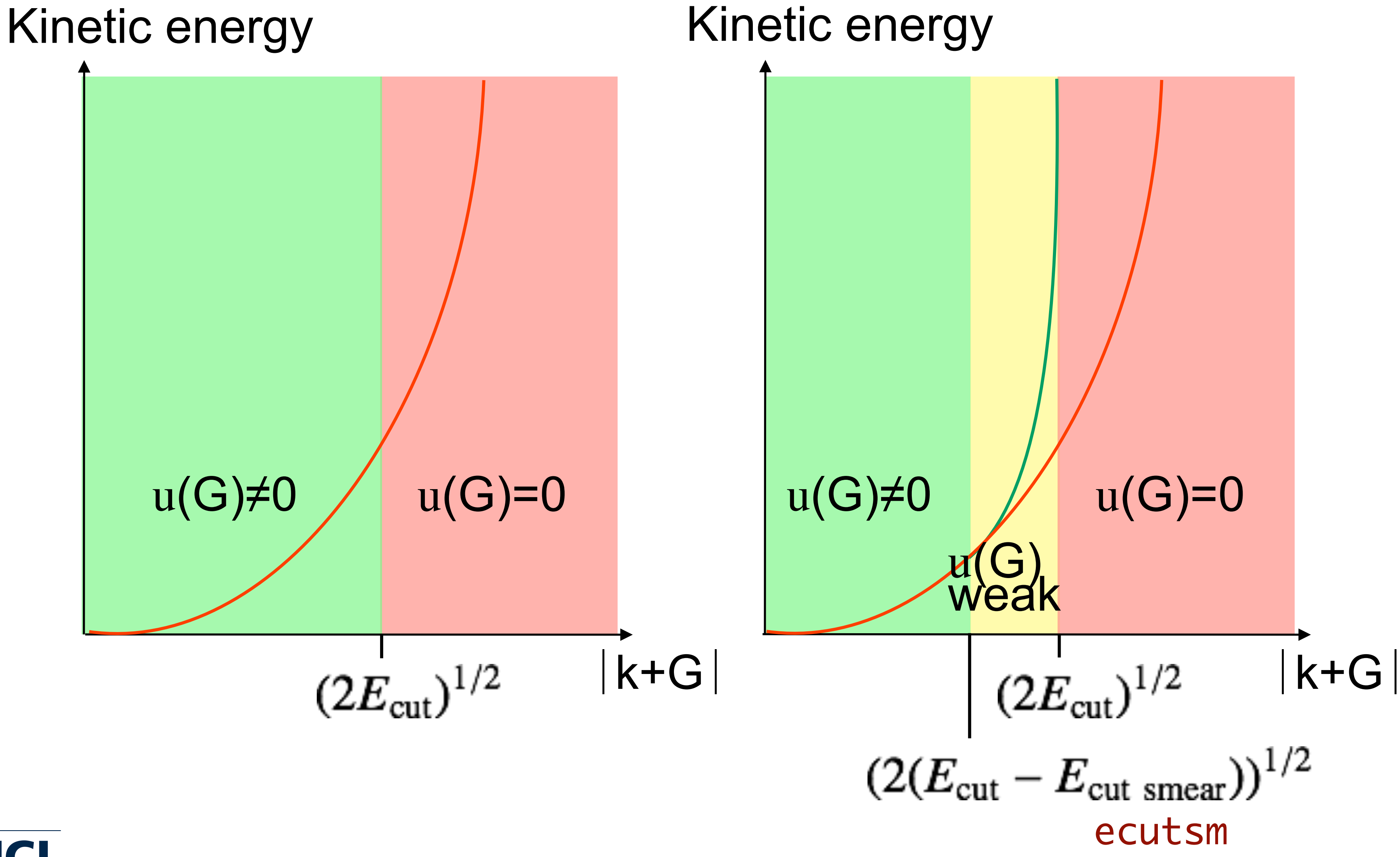
Also, a (discontinuous) function of
lattice parameter at fixed kinetic energy

Discontinuities in energy and pressure

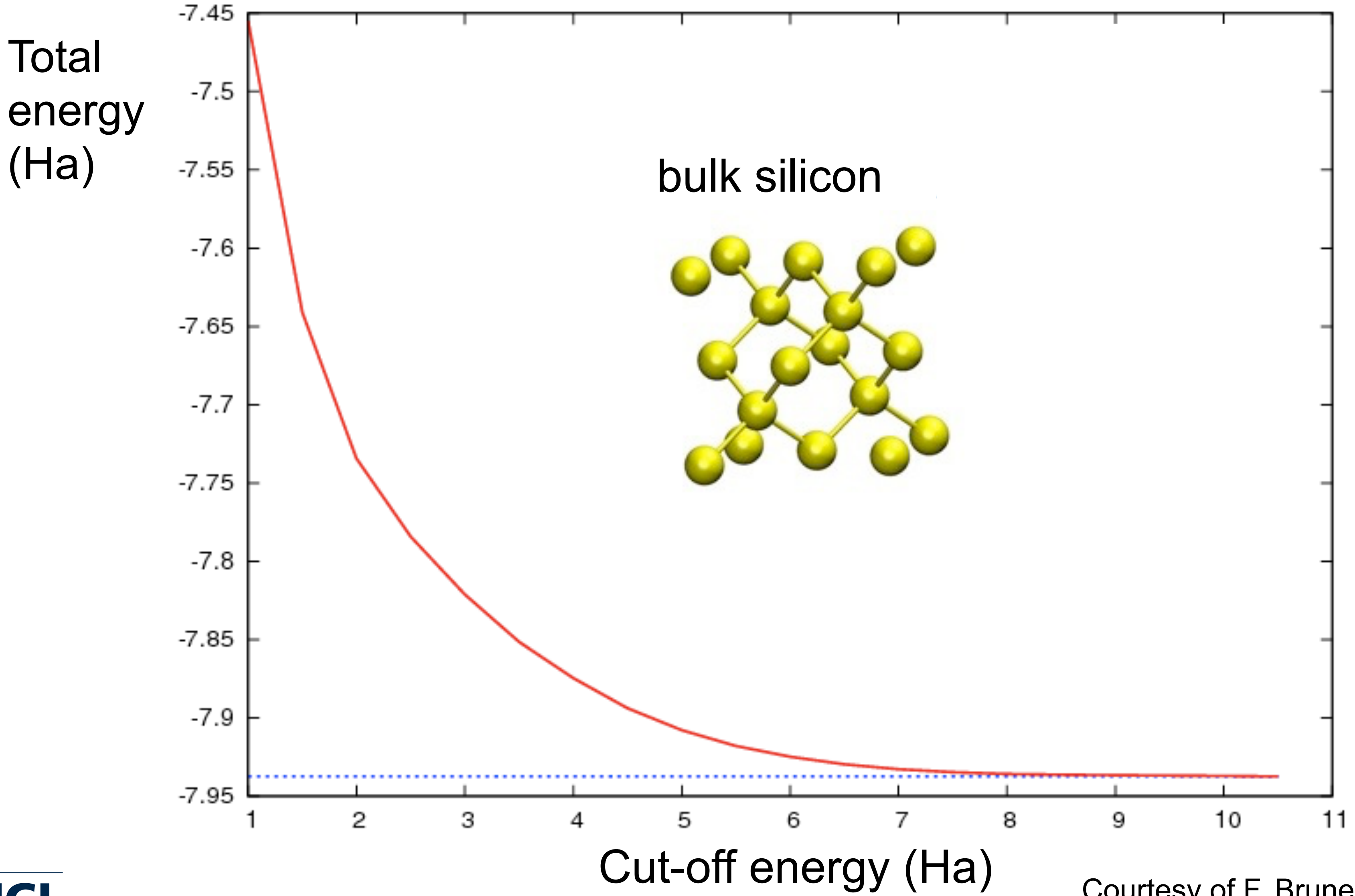
=> Energy (and pressure) also (discontinuous) functions of lattice parameter at fixed kinetic energy



Removing discontinuities



Convergence wrt to kinetic energy cutoff



Courtesy of F. Bruneval

Plane waves : the density and potential

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

$$f(\mathbf{G}) = \frac{1}{\Omega_{o\vec{r}}} \int_{\Omega_{o\vec{r}}} e^{-i\mathbf{G}\mathbf{r}} \, f(\mathbf{r}) \, d\mathbf{r} \qquad f(\mathbf{r}) = \sum_{\vec{G}} e^{i\mathbf{G}\mathbf{r}} \, f(\mathbf{G})$$

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

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

Relation between Fourier coefficients:

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

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

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

Average

Representation of the density

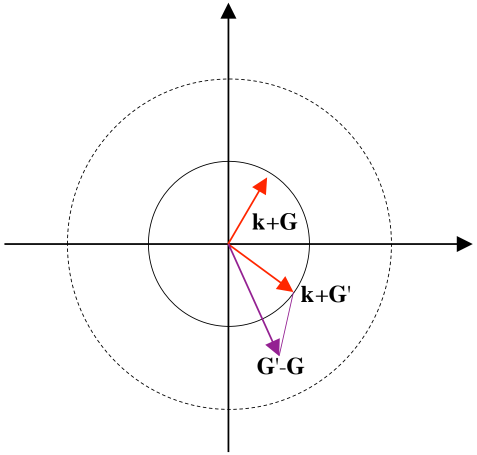
Density associated with one eigenfunction :

$$n_{n\mathbf{k}}(\mathbf{r}) = u^*_{n\mathbf{k}}(\mathbf{r})\, u_{n\mathbf{k}}(\mathbf{r})$$

Computation of $u^*_{n\mathbf{k}}(\mathbf{r})\, u_{n\mathbf{k}}(\mathbf{r})$

$$= \left(\sum_{\mathbf{G}} u^*_{n\mathbf{k}}(\mathbf{G})\, e^{-i\mathbf{G}\mathbf{r}}\right)\left(\sum_{\mathbf{G}'} u_{n\mathbf{k}}(\mathbf{G}')\, e^{-i\mathbf{G}'\mathbf{r}}\right)$$

$$= \sum_{\mathbf{G}\mathbf{G}'}\left[u^*_{n\mathbf{k}}(\mathbf{G})\, u_{n\mathbf{k}}(\mathbf{G}')\right] e^{i(\mathbf{G}'-\mathbf{G})\mathbf{r}}$$



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

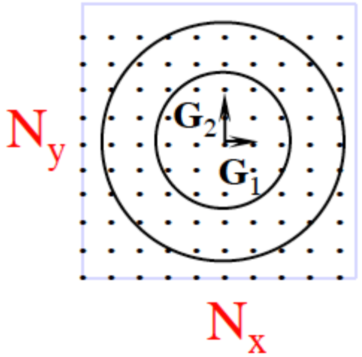
$\mathbf{k}+\mathbf{G}' \in$ sphere

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

From real space to reciprocal space

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

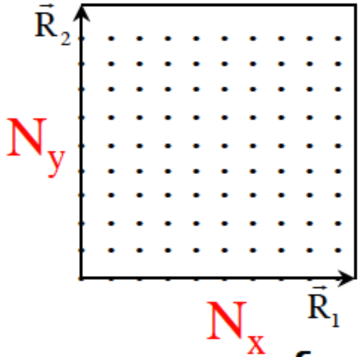
Use of the discrete Fourier transform $\{\mathbf{r}_i\} \leftrightarrow \{\mathbf{G}\}$



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

$$\frac{(\mathbf{k}+\mathbf{G})^2}{2} < E_{\text{cut}}$$ 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 :

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

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

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

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

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

How to treat $\frac{1}{\Omega_{o\mathbf{k}}} \int_{\Omega_{o\mathbf{k}}} \mathrm{X}_{\mathbf{k}} \; \mathrm{d}\mathbf{k}$?

Brillouin zone integration

$$\frac{1}{\Omega_{o\mathbf{k}}} \int_{\Omega_{o\mathbf{k}}} X_{\mathbf{k}} \, d\mathbf{k} \;\Rightarrow\; \sum_{\{\mathbf{k}\}} w_{\mathbf{k}} X_{\mathbf{k}} \qquad [\text{ with } \sum_{\{\mathbf{k}\}} w_{\mathbf{k}} = 1 \;]$$

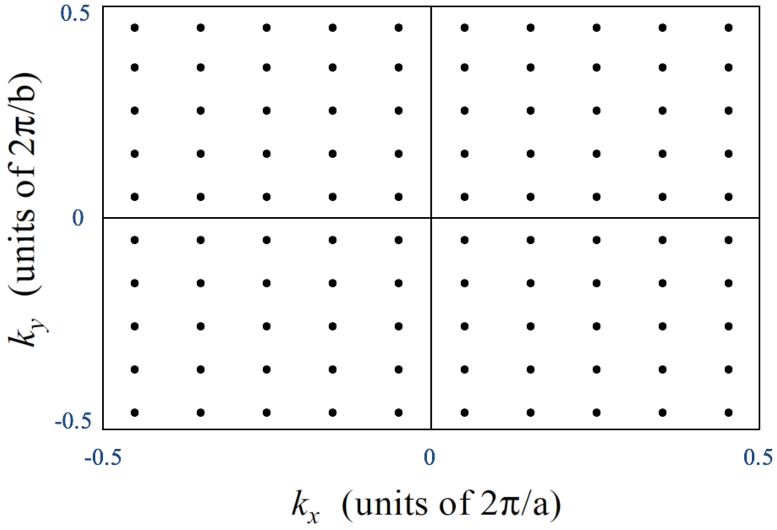
How to chose $\{\mathbf{k}\}$ and $\{w_{\mathbf{k}}\}$?

Special points

Weights

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

Homogeneous sampling of the Brillouin zone



Brillouin zone integration

Theorem :

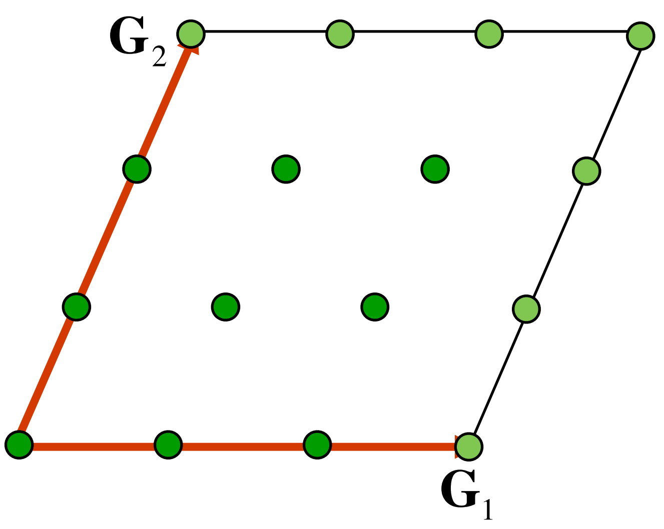
If
- the integrand is periodic
- the integrand is continuous + derivable at all orders $\left(C^{\infty} D^{\infty}\right)$
- $\{\mathbf{k}\}$ homogeneous grid (1D - 2D - 3D) and $w_{\mathbf{k}}$ all equal

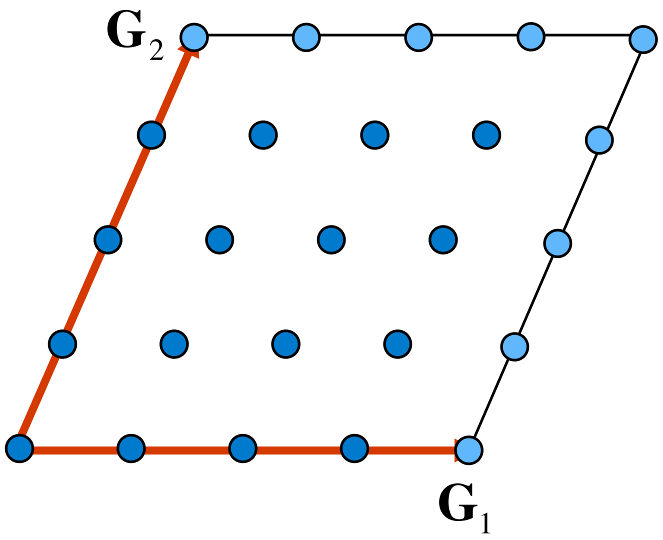
Then exponential convergence, with respect to $\Delta\mathbf{k}$

- 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)]





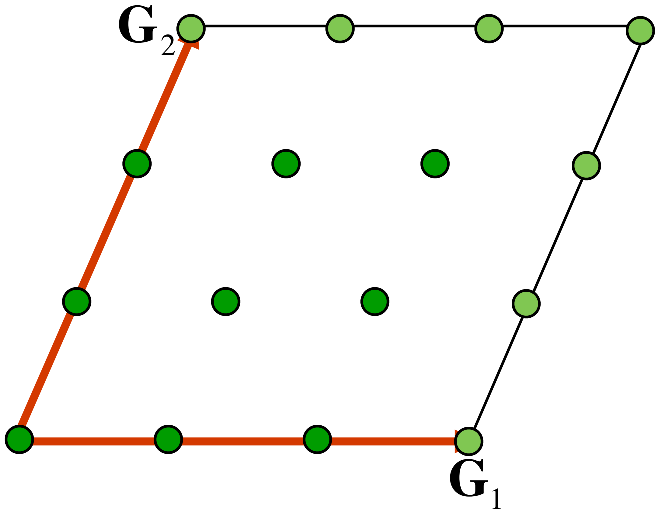
$n_{k1} = n_{k2} = 3$

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

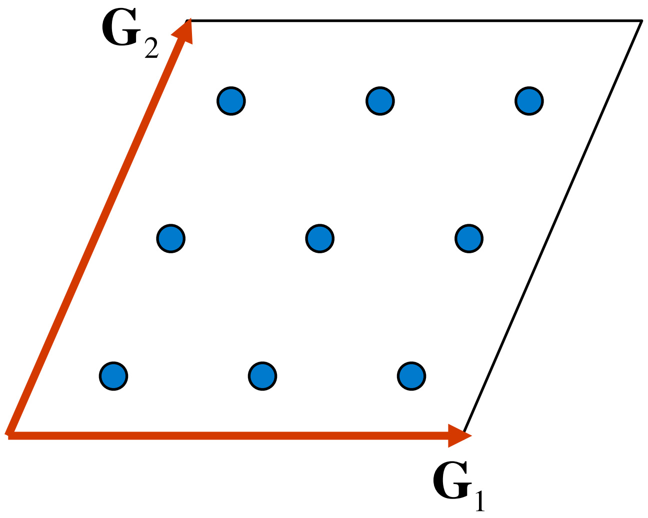
`ngkpt nk1 nk2 nk3`

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

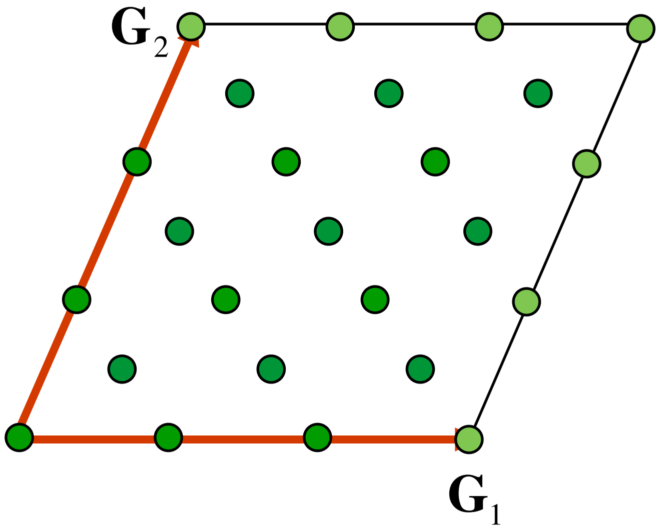


$n_{k1} = n_{k2} = 3$
shifted by (1/2,1/2)

```
ngkpt nk1 nk2 nk3
shiftk sk1 sk2 sk3   (defaut: 0.5 0.5 0.5)
```

Combining grids with various shifts

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

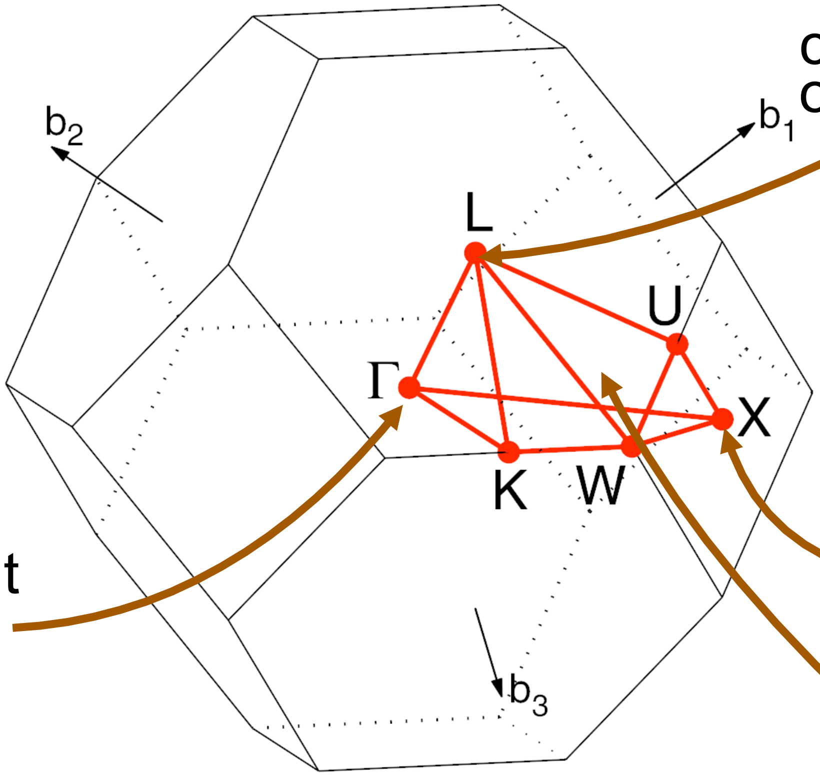


combining unshifted
and shifted by (1/2,1/2)
for $n_{k1} = n_{k2} = 3$

```
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.



count this point only 8 times

count this point only once

count this point only 6 times

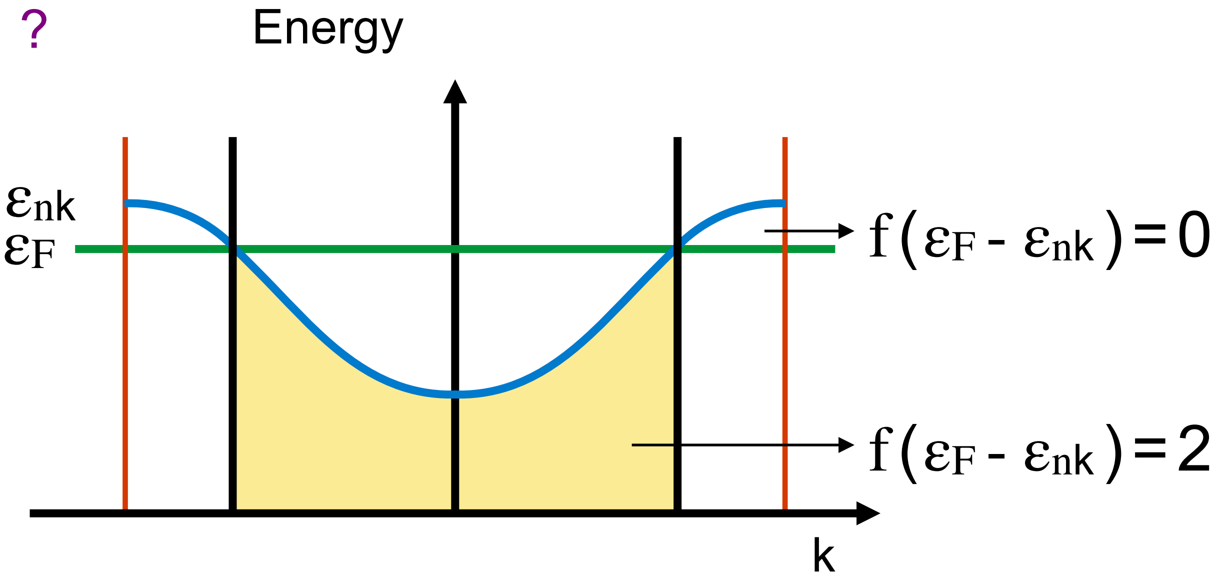
count the points inside the wedge 48 times

Treatment of metals (I)

Behaviour of $f(\varepsilon_F - \varepsilon_{\mathbf{nk}})$?

Discontinuity of
integrand
at Fermi level

Smearing technique



First trial : generalisation of DFT to finite temperature

$$f(\varepsilon_{n\mathbf{k}}) = \frac{1}{1+e^{(\varepsilon_{n\mathbf{k}}-\varepsilon_F)/kT}}$$

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

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

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

```
occopt 3
tsmear σ
```

Problem : T needed to recover the same convergence as for semiconductors is very high (>> 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

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

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

$$f_{\mathbf{nk}} \; (\varepsilon_{\mathbf{nk}}) \; = \; \underset{\text{Spin factor}}{s} \; . \int_{t=\frac{\varepsilon_{\mathbf{nk}}-\varepsilon_F}{\sigma}}^{\infty} \tilde{\delta}(t) \, dt \qquad [\text{ with } \int_{-\infty}^{\infty} \tilde{\delta}(t) \, dt \; = \; 1 \;]$$

Gaussian smearing $\tilde{\delta}(x) = \frac{1}{\sqrt{\pi}} e^{-x^2}$ ⇨ α small `occopt 7`

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

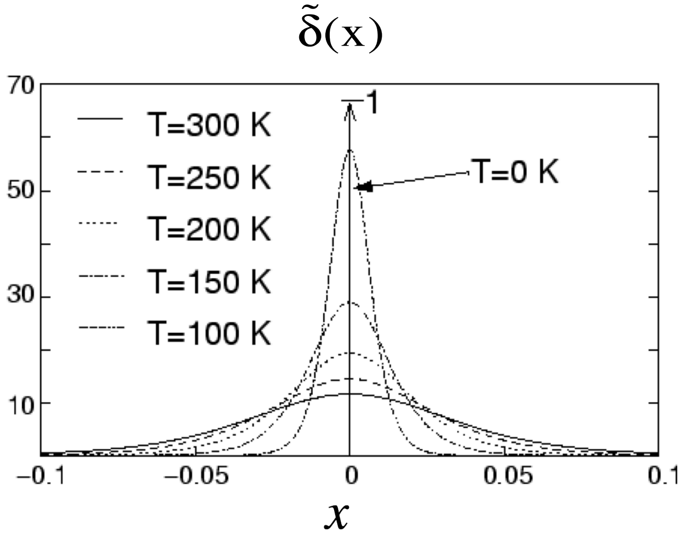
⇨ n = 4 but occupations can be negative

...

'Cold Smearing' `occopt 4/5`

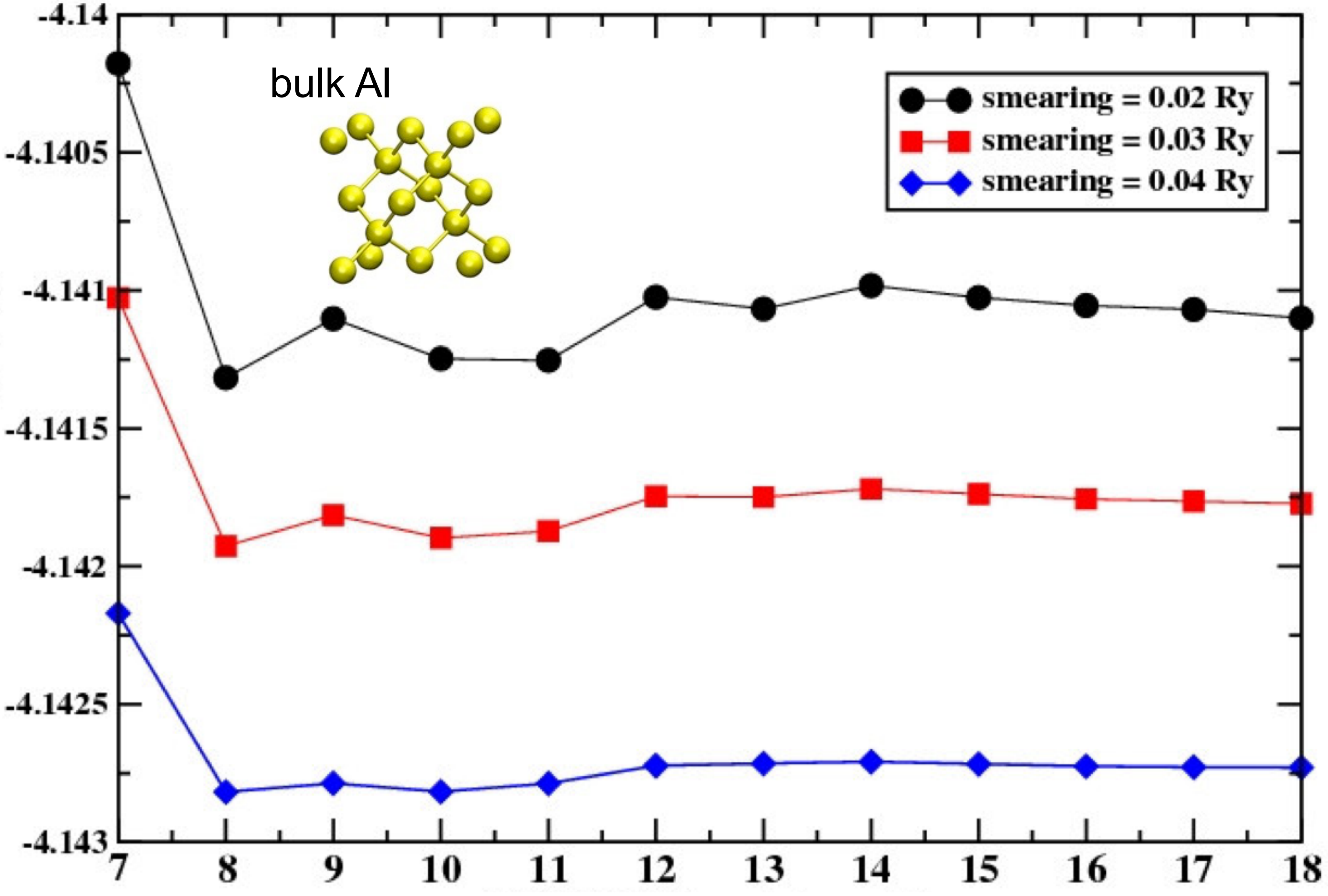
(Marzari et al, Phys. Rev. Lett. 82, 3296 (1999))

⇨ n = 3 with positive occupations



Convergence wrt k-points and smearing

Total energy (Ry)



n_k ($n_k \times n_k \times n_k$ k-point grid)

Courtesy of S. Narasimhan

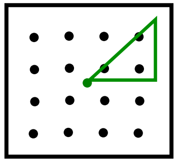
How many k points ? Smearing width ?

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 ⇨ integration in the irreducible Brillouin zone

2D Example



grid 4 x 4 = 16

⇩

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...

$$n(\mathbf{r}) = \sum_{i}^{N} \psi_i^*(\mathbf{r})\psi_i(\mathbf{r})$$

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

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

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.

F atom : $\boxed{(1s)^2} + (2s)^2(2p)^5$

IP 1keV 10-100 eV

Ti atom : $\boxed{(1s)^2(2s)^2(2p)^6}(3s)^2(3p)^6(4s)^2(3d)^2$ small core

$\boxed{(1s)^2(2s)^2(2p)^6(3s)^2(3p)^6}(4s)^2(3d)^2$ large core

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 ...

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

$$\begin{aligned}E_{\mathrm{KS}}\left[\{\psi_i\}\right]=&\sum_{i\in core}^{N_{core}}\langle\psi_i|-\frac{1}{2}\nabla^2|\psi_i\rangle+\int V_{ext}(\mathbf{r})n_{core}(\mathbf{r})\mathrm{d}\mathbf{r}+\frac{1}{2}\int\frac{n_{core}(\mathbf{r}_1)n_{core}(\mathbf{r}_2)}{|\mathbf{r}_1-\mathbf{r}_2|}\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2\\&+\sum_{i\in val}^{N_{val}}\langle\psi_i|-\frac{1}{2}\nabla^2|\psi_i\rangle+\int V_{ext}(\mathbf{r})n_{val}(\mathbf{r})\mathrm{d}\mathbf{r}+\frac{1}{2}\int\frac{n_{val}(\mathbf{r}_1)n_{val}(\mathbf{r}_2)}{|\mathbf{r}_1-\mathbf{r}_2|}\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2\\&+\int\frac{n_{val}(\mathbf{r}_1)n_{core}(\mathbf{r}_2)}{|\mathbf{r}_1-\mathbf{r}_2|}\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2+E_{\mathrm{xc}}\left[n_{core}+n_{val}\right]\end{aligned}$$

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 $\left(-\frac{1}{2}\nabla^2 + \mathrm{v}\right) \mid \psi_i > = \varepsilon_i \mid \psi_i >$

To $\left(-\frac{1}{2}\nabla^2 + \mathrm{v}_{ps}\right) \mid \psi_{ps,i} > = \varepsilon_{ps,i} \mid \psi_{ps,i} >$

Possible set of conditions (norm-conserving pseudopotentials)
NCPP - Hamann D.R., Schlüter M., Chiang C, Phys.Rev.Lett. 43, 1494 (1979)

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

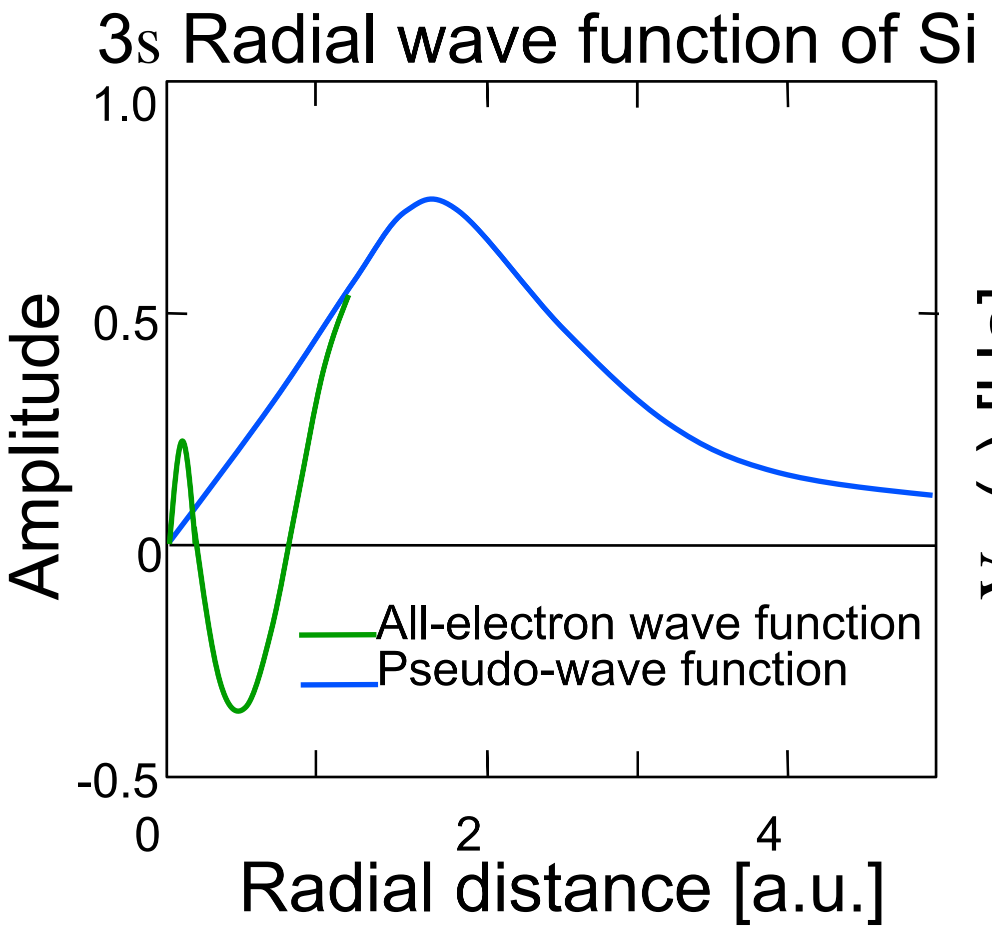
$$\psi_i(\mathbf{r}) = \psi_{ps,i}(\mathbf{r}) \quad \text{for} \quad r > r_c$$

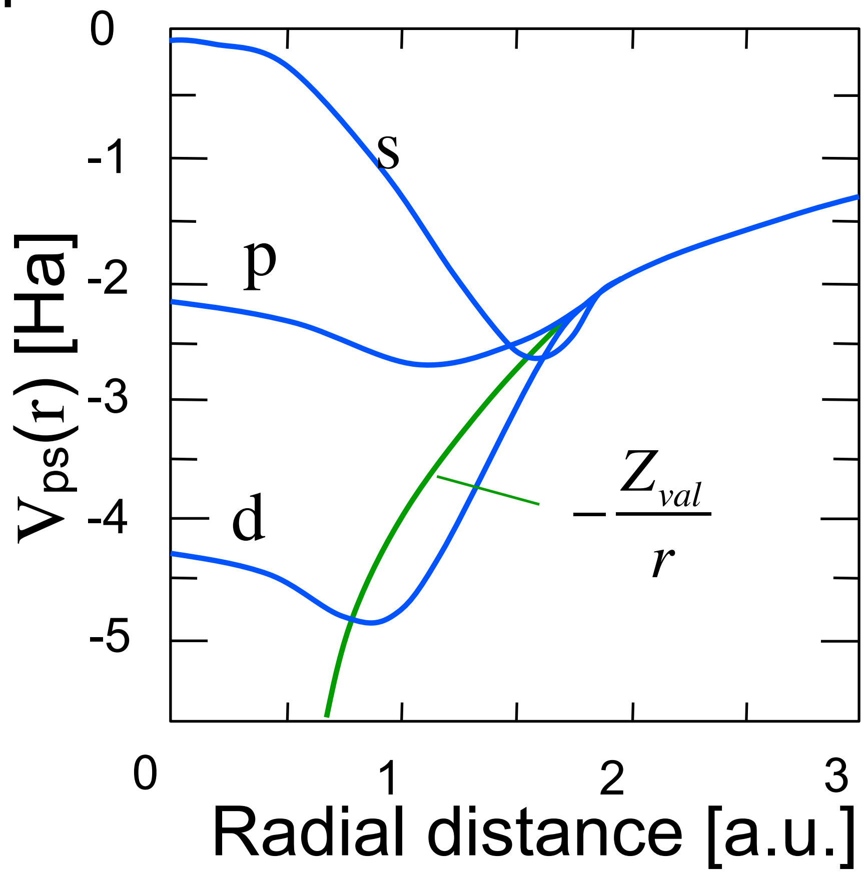
$$\int_{r<r_c} \left|\psi_i(\mathbf{r})\right|^2 d\mathbf{r} = \int_{r<r_c} \left|\psi_{ps,i}(\mathbf{r})\right|^2 d\mathbf{r}$$

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

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

Example of pseudopotential



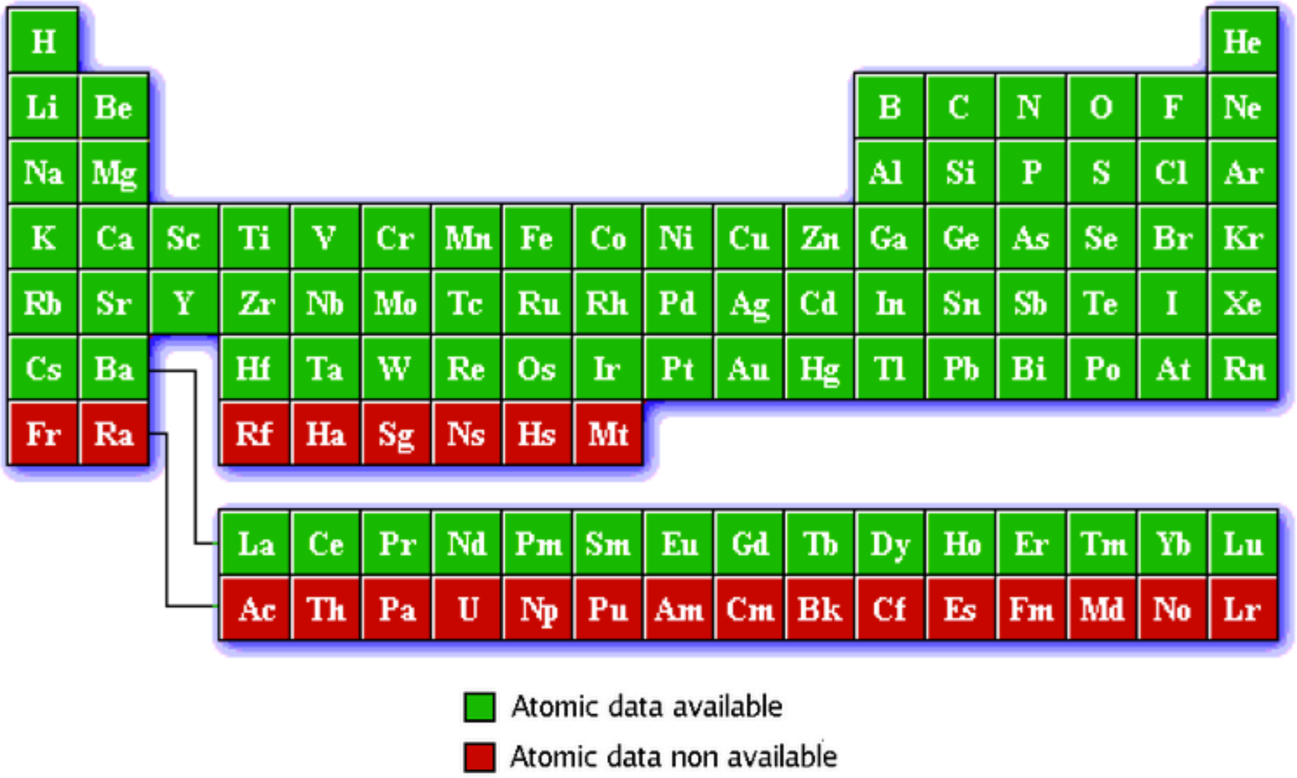


Pseudopotentials/PAW data in ABINIT

- 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

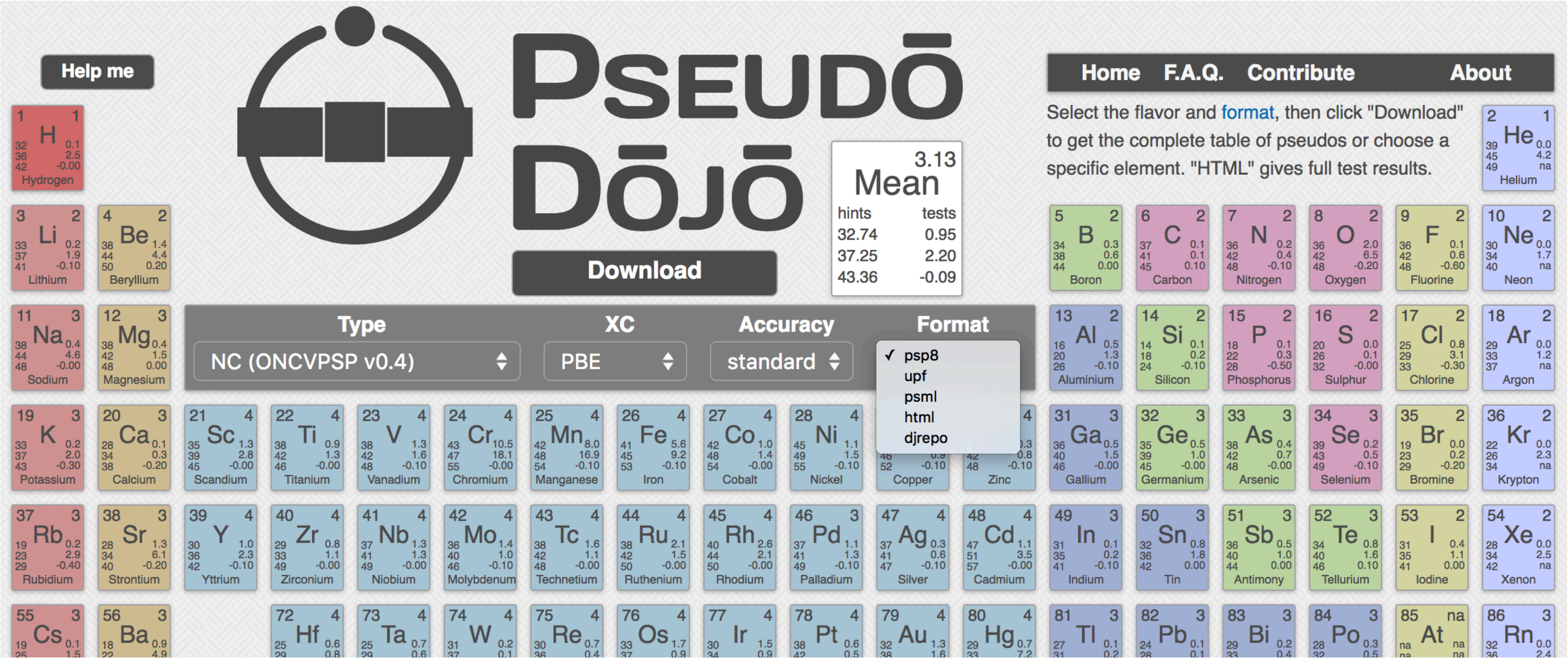
Also, possibility to use : GPAW table, GBRV v1.0 table, or norm-conserving pseudopotentials (e.g. ONCVPSP pseudo generator), or many others

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



Select the flavor and format, then click "Download" to get the complete table of pseudos or choose a specific element. "HTML" gives full test results.

Mean 3.13

hints	tests
32.74	0.95
37.25	2.20
43.36	-0.09

Type: NC (ONCVPSP v0.4)
XC: PBE
Accuracy: standard
Format: ✓ psp8

Computing the forces

Computing the forces (I)

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

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

$$F_{\kappa,\alpha} = -\left.\frac{\partial E}{\partial R_{\kappa,\alpha}}\right|_{\{\vec{R}_\kappa\}}$$

(principle of virtual works)

Forces can be computed by finite differences.

Better approach : compute the response to a perturbation

⇨ What is the energy change ?

$$\{R_{\kappa,\alpha}\} \rightarrow \{R_{\kappa,\alpha} + \lambda\delta R_{\kappa,\alpha}\}$$

Small parameter

Computing the forces (II)

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

Perturbation theory : Hellmann - Feynman theorem

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

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

Application to the derivative with respect to an atomic displacement :

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

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

Computing the forces (III)

Generalisation to density functional theory

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

If change of atomic positions ...

$$V_{ext}(\vec{r}) = \sum_{k'} -\frac{Z_{k'}}{\left|\vec{r}-\vec{R}_{k'}\right|}$$

(can be generalized to pseudopotential case)

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

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

Forces can be computed directly from the density !

Iterative algorithms

Algorithmics : problems to be solved

(1) Kohn - Sham equation

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

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

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

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

$$\begin{tikzcd}
V_{KS}(\mathbf{r}) \arrow[rr, bend left] & & \psi_i(\mathbf{r}) \arrow[dl, bend left] \\
& n(\mathbf{r}) \arrow[ul, bend left] &
\end{tikzcd}$$

(3) Geometry optimization

Find the positions $\left\{\mathbf{R}_\kappa\right\}$ of ions such that the forces $\left\{\mathbf{F}_\kappa\right\}$ 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 :

$$\boxed{R_{\kappa,\alpha}^{(n+1)} = R_{\kappa,\alpha}^{(n)} + \lambda F_{\kappa,\alpha}^{(n)}}$$

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

Energy+forces around equilib. geometry

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

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

$$\mathrm{F}_{\kappa,\alpha}(\mathrm{R}_{\kappa',\alpha'}) = \cancel{\mathrm{F}_{\kappa,\alpha}(\mathrm{R}^{*}_{\kappa',\alpha'})} + \sum_{\kappa',\alpha'} \left.\frac{\partial \mathrm{F}_{\kappa,\alpha}}{\partial \mathrm{R}_{\kappa',\alpha'}}\right|_{\{\mathrm{R}^{*}\}} \left(\mathrm{R}_{\kappa',\alpha'} - \mathrm{R}^{*}_{\kappa',\alpha'}\right) + O\left(\mathrm{R}_{\kappa',\alpha'} - \mathrm{R}^{*}_{\kappa',\alpha'}\right)^{2}$$

Moreover, $\mathrm{F}_{\kappa,\alpha} = -\dfrac{\partial \mathrm{E}^{\mathrm{BO}}}{\partial \mathrm{R}_{\kappa,\alpha}}$ $\qquad \dfrac{\partial \mathrm{F}_{\kappa',\alpha'}}{\partial \mathrm{R}_{\kappa,\alpha}} = -\dfrac{\partial^{2} \mathrm{E}^{\mathrm{BO}}}{\partial \mathrm{R}_{\kappa,\alpha}\partial \mathrm{R}_{\kappa',\alpha'}}$

Vector and matrix notation

$$\mathrm{R}^{*}_{\kappa,\alpha} \rightarrow \underline{\mathrm{R}}^{*} \qquad \mathrm{F}_{\kappa,\alpha} \rightarrow \underline{\mathrm{F}} \qquad \left.\frac{\partial^{2} \mathrm{E}^{\mathrm{BO}}}{\partial \mathrm{R}_{\kappa,\alpha}\partial \mathrm{R}_{\kappa',\alpha'}}\right|_{\{\mathrm{R}^{*}_{\kappa,\alpha}\}} \rightarrow \underline{\underline{\mathrm{H}}}$$

$$\mathrm{R}_{\kappa,\alpha} \rightarrow \underline{\mathrm{R}}$$

(the Hessian)

Steepest-descent : analysis (I)

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

Analysis of this algorithm, in the linear regime :

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

$$\underline{R}^{(n+1)} = \underline{R}^{(n)} + \lambda \underline{F}^{(n)} \Longrightarrow \left(\underline{R}^{(n+1)} - \underline{R}^{*}\right) = \left(\underline{R}^{(n)} - \underline{R}^{*}\right) - \lambda \underline{\underline{H}}\left(\underline{R}^{(n)} - \underline{R}^{*}\right)$$

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

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) $\boxed{\underline{\underline{1}} - \lambda \underline{\underline{H}}}$

Steepest-descent : analysis (II)

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

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

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

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

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

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

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

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

Steepest-descent : analysis (III)

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

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 ?

$\underline{\underline{H}}$ positive definite => all h_i are positive

Yes ! If λ positive, sufficiently small ...

Steepest-descent : analysis (IV)

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

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 $\left(\underline{R}^{(n)}-\underline{R}^{*}\right)$?)

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

\+ what is the convergence rate ?

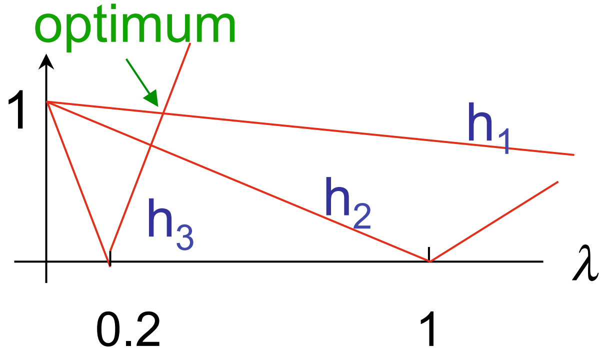
Hint : draw the three functions $|1-\lambda h_i|$ as a function of λ . Then, find the location of λ where the largest of the three curves is the smallest.
Find the coordinates of this point.

Steepest-descent : analysis (V)

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

$h_1 = 0.2$ $|1-\lambda .0.2|$ optimum $\Rightarrow \lambda = 5$
$h_2 = 1.0$ $|1-\lambda . 1|$ optimum $\Rightarrow \lambda = 1$
$h_3 = 5.0$ $|1-\lambda . 5|$ optimum $\Rightarrow \lambda = 0.2$

$\Big\}$?



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

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

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

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

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, $\lambda_{opt} = 2 / (h_{min} + h_{max})$

$\mu_{opt} = 2 / [1+ (h_{max}/h_{min})] - 1 = [(h_{max}/h_{min}) -1] / [(h_{max}/h_{min}) +1]$

Perfect if $h_{max} = h_{min}$. Bad if $h_{max} >> h_{min}$.
h_{max}/h_{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.

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

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

(The latter approximate equality supposes a large condition number)

Analysis of self-consistency

$$\begin{tikzcd}
\mathrm{v}_{\mathrm{KS}}(\mathbf{r}) \arrow[rr, bend left] & & \psi_i(\mathbf{r}) \arrow[dl, bend left] \\
& \mathrm{n}(\mathbf{r}) \arrow[ul, bend left] &
\end{tikzcd}$$

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

$$\mathrm{v}_{\mathrm{in}}(\mathbf{r}) \rightarrow \psi_i(\mathbf{r}) \rightarrow \mathrm{n}(\mathbf{r}) \rightarrow \mathrm{v}_{\mathrm{out}}(\mathbf{r})$$

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

The difference $\mathrm{v}_{\mathrm{out}}(\mathbf{r}) - \mathrm{v}_{\mathrm{in}}(\mathbf{r})$ (generic name : a "residual")

Simple mixing algorithm
(≈ steepest - descent)

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

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

$$\searrow \underline{\underline{\mathrm{H}}}$$

Like the steepest-descent algorithm, this leads to the requirement to minimize |1- λ h$_i$| where h$_i$ are eigenvalues of $\dfrac{\delta \underline{\mathrm{v}}_{\mathrm{out}}}{\delta \underline{\mathrm{v}}_{\mathrm{in}}}$

(actually, the dielectric matrix)

Modify the condition number (II)

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

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

Notes : 1) If approximate inverse Hessian perfect, optimal geometry is reached in one step, with λ =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 $\left(\underline{\underline{H}}^{-1}\right)_{\text{approx}} \underline{\underline{H}}$ govern the convergence : the condition number can be changed. $\left(\underline{\underline{H}}^{-1}\right)_{\text{approx}}$ 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),
forlarger 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

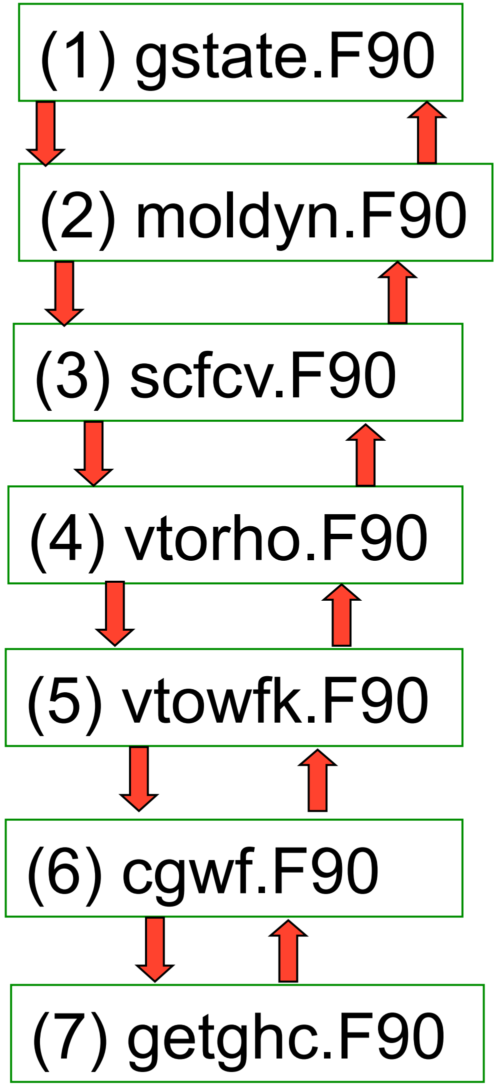
Instead of using only previously computed forces,
take into account past forces for past positions

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



(1) gstate.F90	Ground-state
(2) moldyn.F90	Molecular dynamics
(3) scfcv.F90	Self-consistent field convergence
(4) vtorho.F90	From a potential (v) to a density (rho)
(5) vtowfk.F90	From a potential (v) to a wavefunction at some k-point
(6) cgwf.F90	Conjugate-gradient on one wavefunction
(7) getghc.F90	Get the application of the Hamiltonian

In practice …

(1) Kohn - Sham equation

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

Details are usually hidden to the user nline

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

(2) Self-consistency

$$\begin{tikzcd}
V_{KS}(\mathbf{r}) \arrow[rr, bend left] & & \psi_i(\mathbf{r}) \arrow[dl, bend left] \\
& n(\mathbf{r}) \arrow[ul, bend left] &
\end{tikzcd}$$

Target tolerance	toldfe, toldff, tolrff, tolvrs
+ Maximal number of loops	nstep
Preconditioner	diemac, diemix, …

(3) Geometry optimization / molecular dynamics

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

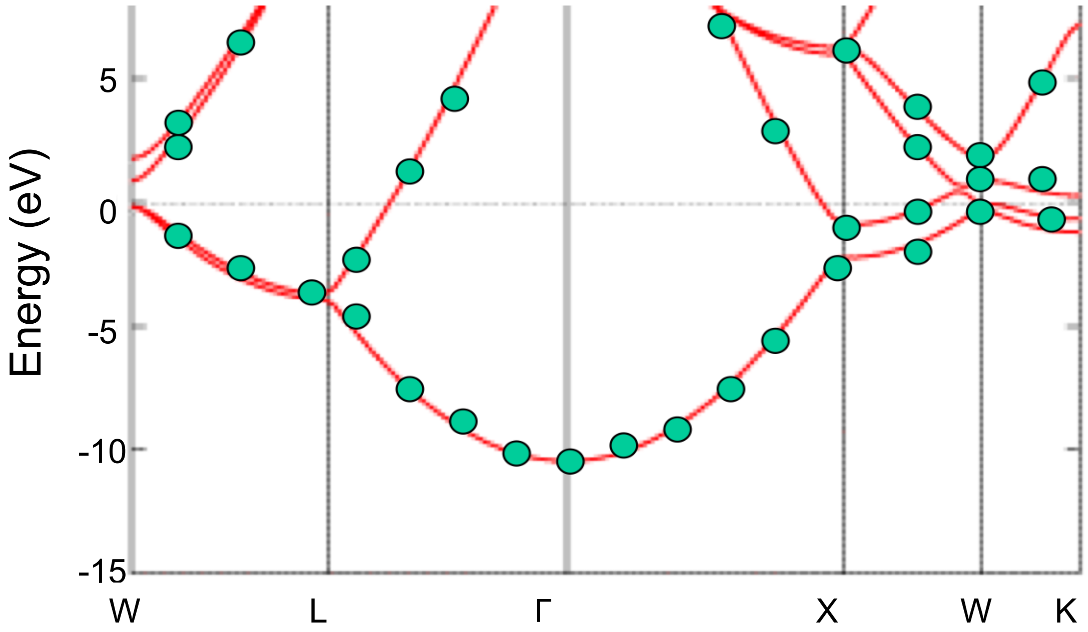
Target tolerance	tolmxf
+ Maximal number of loops	ntime
Algorithm	ionmov

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 $E+dE$:

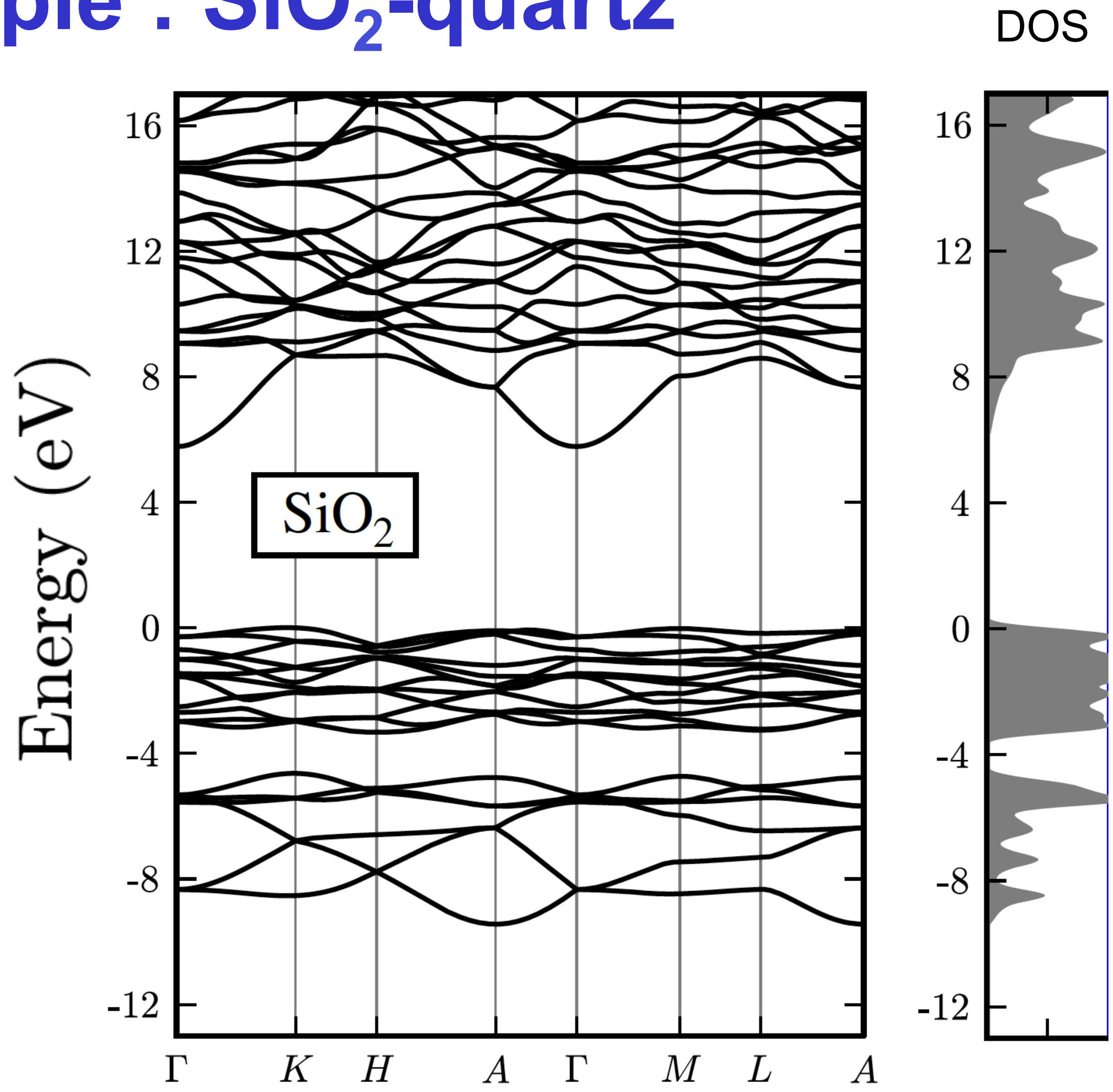
$$g(E) = \sum_n \frac{1}{\Omega_{0\mathbf{k}}} \int_{\Omega_{0\mathbf{k}}} \delta(E - \varepsilon_{n\mathbf{k}})\, d\mathbf{k}$$

- 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`).

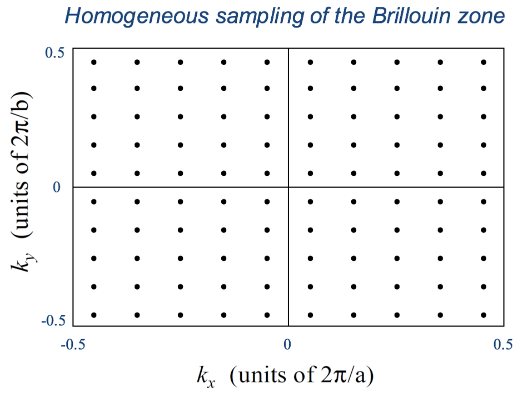
Example : SiO_2-quartz

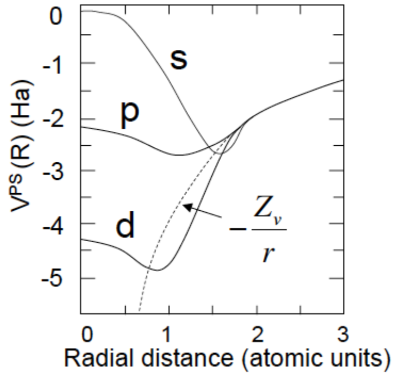


Summary

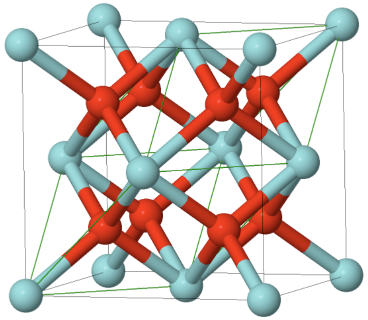
- Plane waves basis set
- Brillouin zone integration
- PW need pseudopotentials
- Easy computation of forces

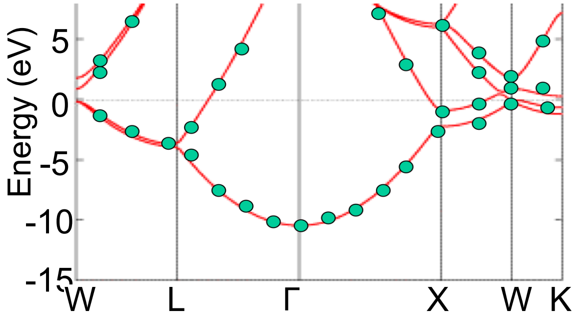
$$\psi_{\mathbf{k}}(\mathbf{r}) = \left(N\Omega_0\right)^{-1/2} \sum_{\mathbf{G}} \mathrm{u}_{\mathbf{k}}\ (\mathbf{G})\ \mathrm{e}^{\mathrm{i}(\mathbf{k}+\mathbf{G})\mathbf{r}}$$





=> relaxation of geometry, or molecular dynamics



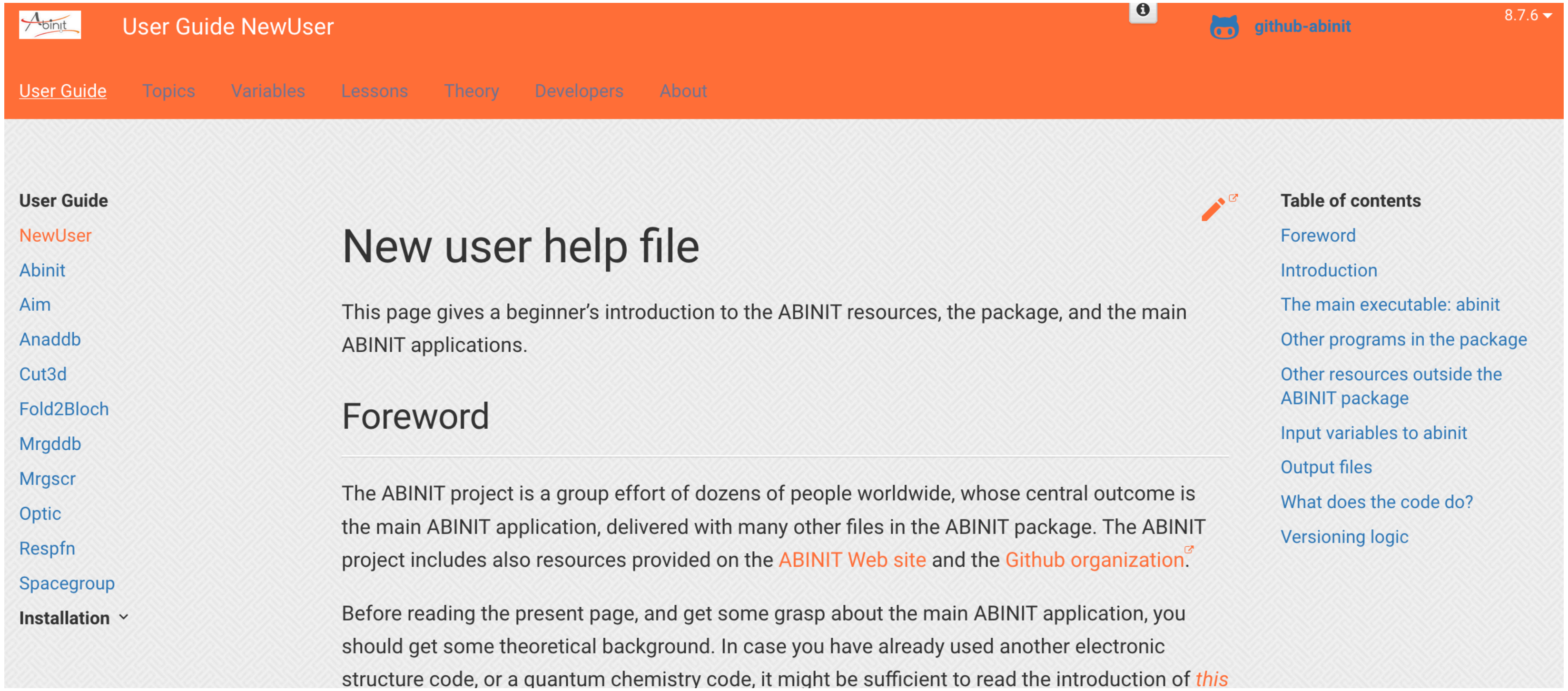


Beyond the basics

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)



New user help file

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

Foreword

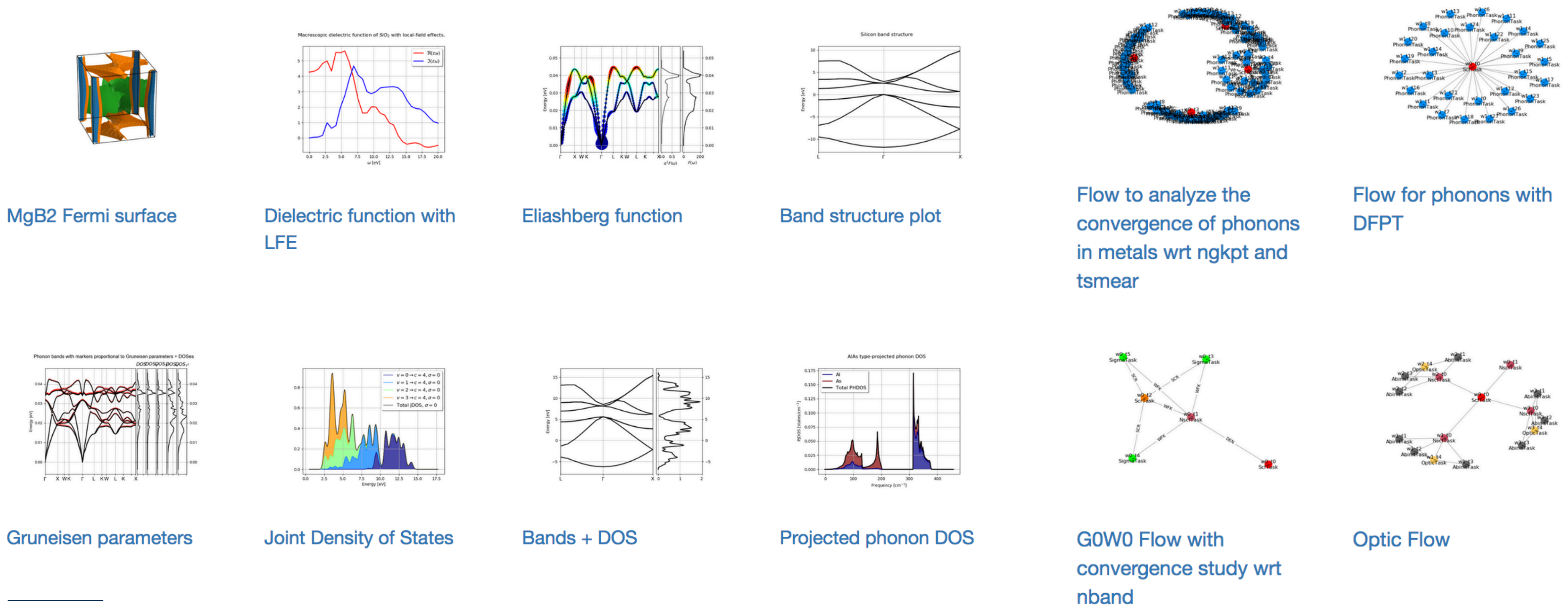
The ABINIT project is a group effort of dozens of people worldwide, whose central outcome is the main ABINIT application, delivered with many other files in the ABINIT package. The ABINIT project includes also resources provided on the ABINIT Web site and the Github organization.

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*

Documentation : abipy galleries

Web site https://github.com/abinit/abipy

- Plot gallery http://abinit.github.io/abipy/gallery/index.html
- Flow gallery 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

Gruneisen parameters

Joint Density of States

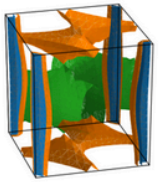
Bands + DOS

Projected phonon DOS

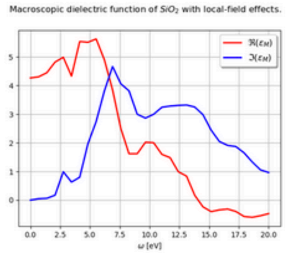
G0W0 Flow with convergence study wrt nband

Optic Flow

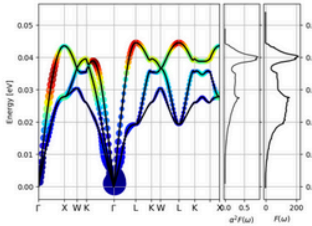
Documentation : abipy galleries



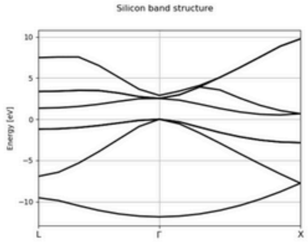
MgB2 Fermi surface



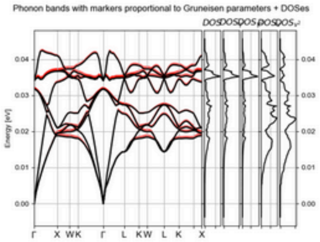
Dielectric function with LFE



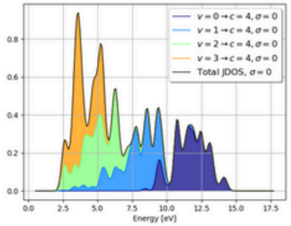
Eliashberg function



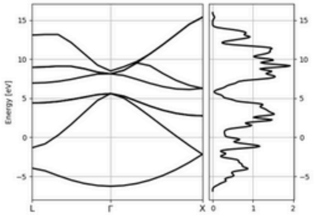
Band structure plot



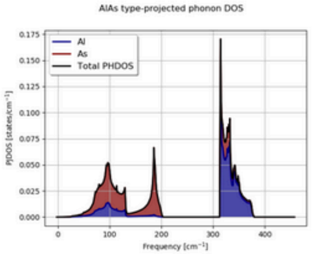
Gruneisen parameters



Joint Density of States

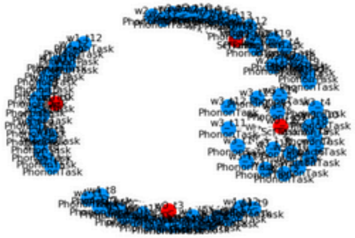


Bands + DOS



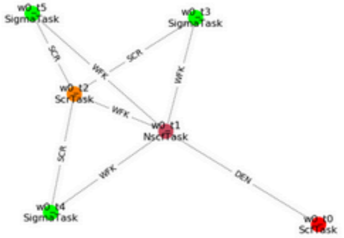
Projected phonon DOS

Documentation : abipy galleries



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

Flow for phonons with DFPT



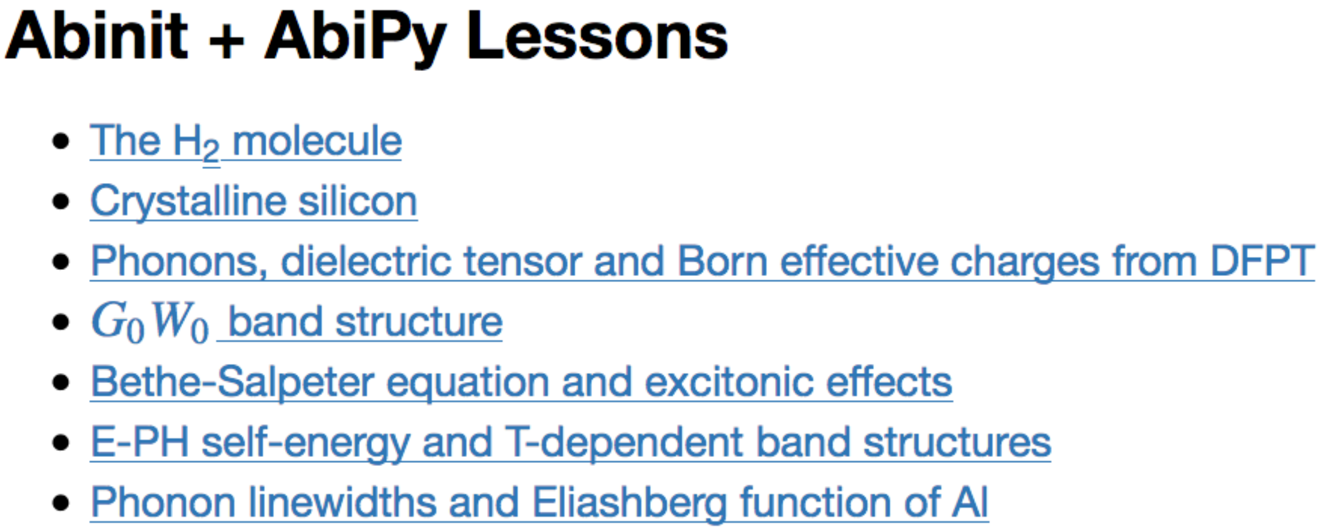
G0W0 Flow with convergence study wrt nband

Optic Flow

Documentation : abitutorials

Web site https://github.com/abinit/abitutorials

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



Abinit + AbiPy Lessons

- The H_2 molecule
- Crystalline silicon
- Phonons, dielectric tensor and Born effective charges from DFPT
- G_0W_0 band structure
- Bethe-Salpeter equation and excitonic effects
- E-PH self-energy and T-dependent band structures
- Phonon linewidths and Eliashberg function of Al