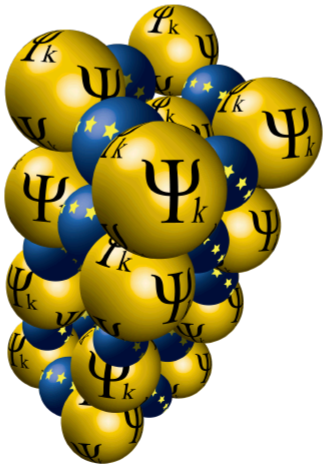
ICTP/Psi-k/CECAM School on Electron-Phonon Physics from First Principles

Trieste, 19-23 March 2018



Centre Européen de Calcul Atomique et Moléculaire

Introduction to ABINIT

X. Gonze

Many thanks to the > 50 ABINIT contributors.
Especially to GM Rignanese for contributions to the slides and M Giantomassi
for the python+ABINIT developments

ABINIT software project

Ideas (1997) :

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

Now (2018) :

>1900 registered people on the forum
>900 kLines of F90 + many python scripts (abipy)
about 50 contributors to ABINITv8,
last (unofficial) version v8.7.6 used in this school

ABINIT milestones

- Precursor : the Corning PW code (commercialized 1992-1995 by Biosym)
- 1997 : beginning of the ABINIT project
- Dec 2000 : release of ABINITv3 under the GNU General Public License (GPL)
- Nov 2002 – May 2017 : 8 international ABINIT developer workshops
  (between 35 and 60 participants each)
- Jan 2010 : launch of the Forum

Participants to the Frejus 2017 Int. ABINIT developer workshop

Overview

1. ABINIT : a state-of-the-art, robust, flexible, free software
2. Running ABINIT : basics
3. Beyond the basics
4. Introducing DFPT and e-ph lessons
5. ABINIT phonon band structures on the Materials Project web site

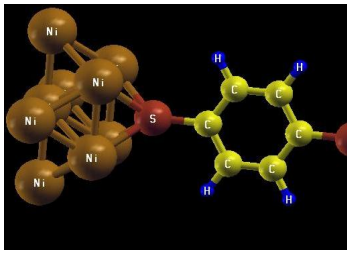
Properties from DFT+MBPT+ ...

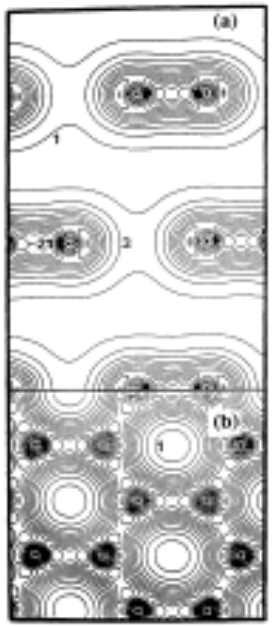
Computation of ...

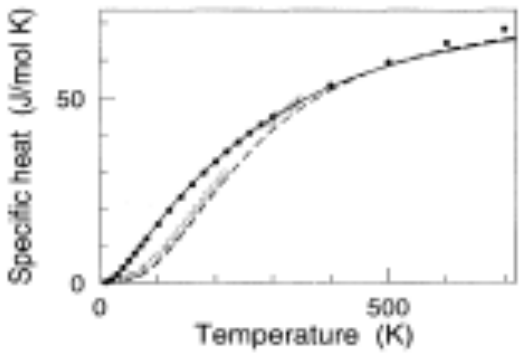
- interatomic distances, angles, total energies
- electronic charge densities, electronic energies

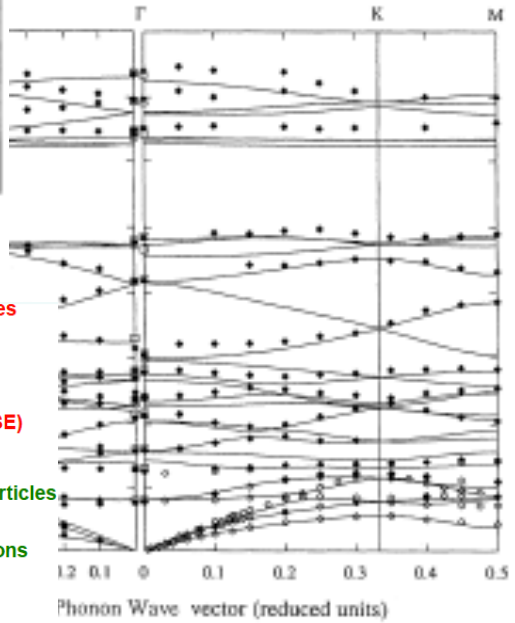
A basis for the computation of ...

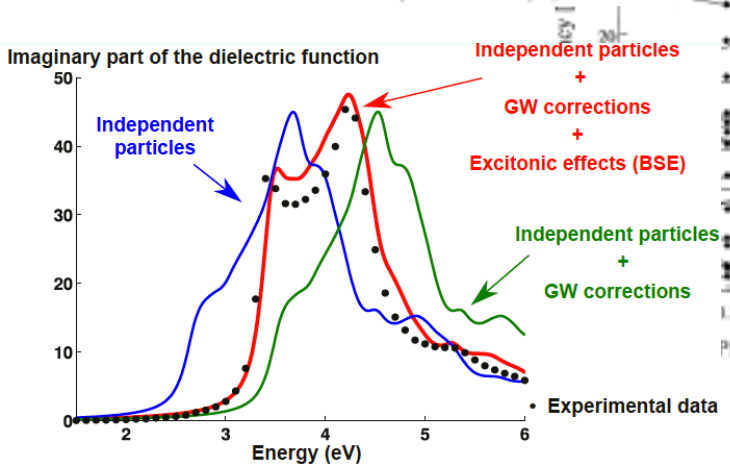
- chemical reactions
- electronic transport
- vibrational properties
- thermal capacity
- dielectric behaviour
- optical response
- superconductivity
- surface properties
- spectroscopic responses
- ...











ABINIT v8 capabilities (I)

Methodologies

Pseudopotentials/Plane Waves
+ Projector Augmented Waves (for selected capabilities)
    Many pseudopotential types, different PAW generators
        (e.g. ATOMPAW is tightly connected to ABINIT, with ATOMPAW tests)

Density functionals : LDA, GGA (many : PBE and variations, HCTH, ...),
    LDA+U (or GGA+U)
    hybrid functionals + some advanced functionals (exact exchange + RPA or ...)
LR-TDDFT for finite systems excitation energies (Casida)
GW for accurate electronic eigenenergies
    (4 plasmon-pole models or contour integration ; non-self-consistent / partly self-consistent / quasiparticle self-consistent ; spin-polarized)
Bethe-Salpeter for accurate optical properties calculations
Dynamical mean field-theory for strongly-correlated materials

ABINIT v8 capabilities (II)

Insulators/metals - smearings : Fermi, Gaussian, Gauss-Hermite ...

Collinear spin / non-collinear spin / spin-orbit coupling

Forces, stresses, automatic optimisation of atomic positions and unit cell parameters (Broyden and Molecular dynamics with damping)

Molecular dynamics (Verlet or Numerov), Nosé thermostat, Langevin dynamics

Path-Integral Molecular Dynamics, String / NEB method for saddle points

Susceptibility matrix by sum over states (Adler-Wiser)

Optical (linear + non-linear) spectra by sum over states

Polarization, finite electric field calculations

Electric field gradients

Positron lifetime

Symmetry analyser (database of the 230 spatial groups and the 1191 Shubnikov magnetic groups)

ABINIT v8 capabilities (III)

Density-Functional Perturbation Theory :

- Responses to atomic displacements, to static homogeneous electric field, to strain perturbations, to magnetic field (Zeeman)

- Second-order derivatives of the energy, giving direct access to :
  dynamical matrices at any q, phonon frequencies, force constants ;
  phonon DOS, thermodynamic properties (quasi-harmonic approximation) ;
  dielectric tensor, Born effective charges ;
  elastic constants, internal strain ;
  piezoelectric tensor ...
- Matrix elements, giving direct access to :
  electron-phonon coupling, deformation potentials, superconductivity,
  temperature-dependence of the electronic structure
- Non-linear responses thanks to the 2n+1 theorem - at present :
  non-linear dielectric susceptibility; Raman cross-section ;
  electro-optic tensor

Quality control : test suite + test farm

How to secure existing capabilitites despite the development efforts (by diverse groups) and associated bug generation ?

Test suite : >1000 automatic tests (+ new added for each capability)

Test farm : >12 computers (4 to 48 cores)

with 4 compilers (gfort, Intel, NAG, XLF) => over 20 'builders'

Name	Brand	CPU / Freq	# cores	RAM	OS	misc
abiref	**HP DL360 gen9**	**Xeon E5-2670v3/ 2.30**	**2 x 24**	**32GB**	**CentOS 7.2**	**Ref**
bob	**Dell R430**	**Xeon E5-2603v3/ 1.60**	**2 x 6**	**8GB**	**Fedora 23**	
buda	SuperMicro	Xeon X5570/ 2.7	2 x 4	12GB	CentOS 6.8	**2xGPU K40 2xGPU C1060**
coba2	HP Z400	Xeon W3520/ 2.7	4	12GB	CentOS 6.5	
cronos	HP DL185 G7	AMD Opteron 6276/ 2.3	2 x 16	16GB	Debian 5.0	
graphene	**Apple MacPro**	**Xeon E5-2697/ 2.7**	**1 x 12**	**64GB**	**MacOS X 10.12**	
ibm8	**IBM Power S824**	**Power8/ 3.0**	**4**	**8GB**	**AIX 7.2**	
inca	virtual machine	Opteron 6276/ 2.3	12	30GB	**CentOS 6.9**	
max2	HP DL185	Opteron 6140/ 2.6	2 x 8	12GB	Slinux 6.1	
petrus	Intel	Core i7 3930/ 3.2	6	16GB	openSUSE 12.1	
testf	Bull Novascale	Xeon X5570/ 2.9	2 x 4	12GB	CentOS 5.11	
tikal	Dell T5500	Xeon X5647/ 3.0	8	8GB	**Slinux 6.9**	
ubu	**HP DL360 gen9**	**Xeon E5-2670v3/ 2.30**	**2 x 24**	**32GB**	**Ubuntu 16.04**	

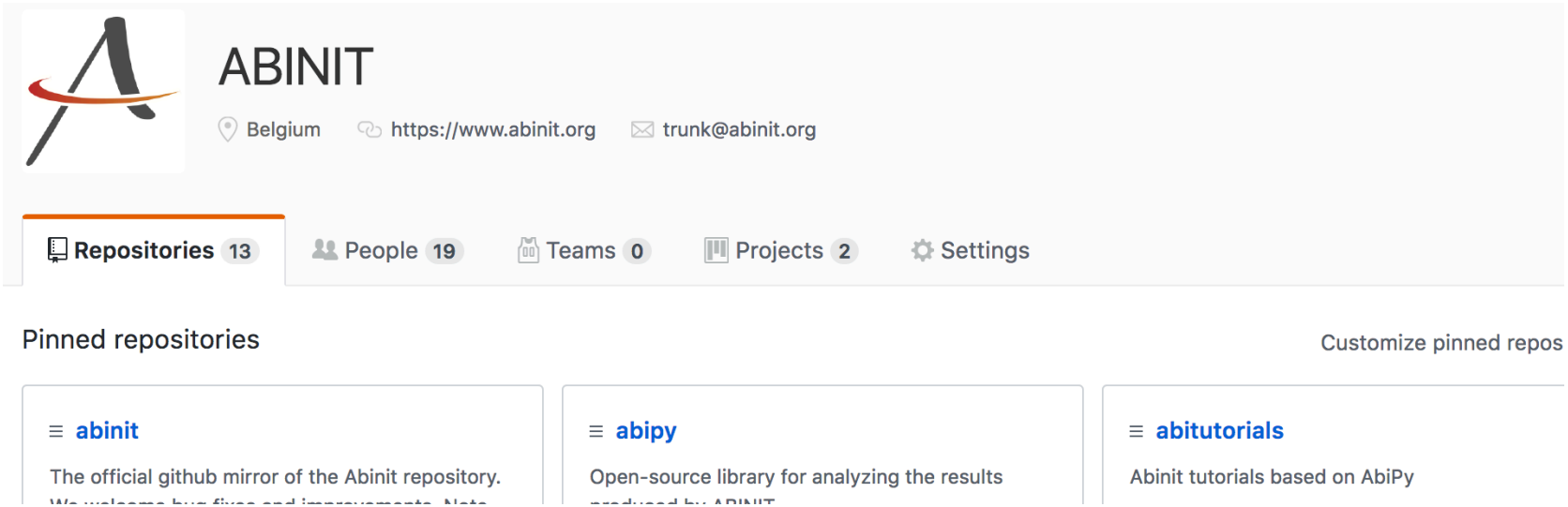
Software : buildbot. ABINIT pioneered its usage in computational condensed matter.

Pouillon et al, Computing in Science and Engineering 13, 62 (2011)

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



The “Free” software concept

Free for freedom (also price ...)

- freedom 1 : unlimited use for any purpose
- freedom 2 : study and modify for your needs (need source access !)
- freedom 3 : copy
- freedom 4 : distribute modifications

From copyright to freedom (“copyleft”)

- copyright allows licensing
- licenses grants freedom

Terminology : Free software=Open source=Libre software

ABINIT pioneered the use of the GPL « Free software license » in the computational condensed matter community (2000)

http://www.abinit.org : download, documentation ...

A few references

- Description of the ABINIT project and ABINIT capabilities

  X. Gonze et al, Comput. Mat. Science 25, 478 (2002)

  X. Gonze et al, Z. Kristallogr. 220, 558 (2005)

  X. Gonze et al, Comp. Phys. Comm. 180, 2582 (2009)

  X. Gonze et al, Comp. Phys. Comm. 205, 106 (2016)

- Software engineering and management techniques used in ABINIT

  Y. Pouillon et al, Computing in Science and Engineering 13, 62 (2011)

- Pseudopotentials & PAW datasets

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

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

- File format description (netCDF ETSF-IO)

  X. Gonze et al, Comput. Mat. Science 43, 1056 (2008)

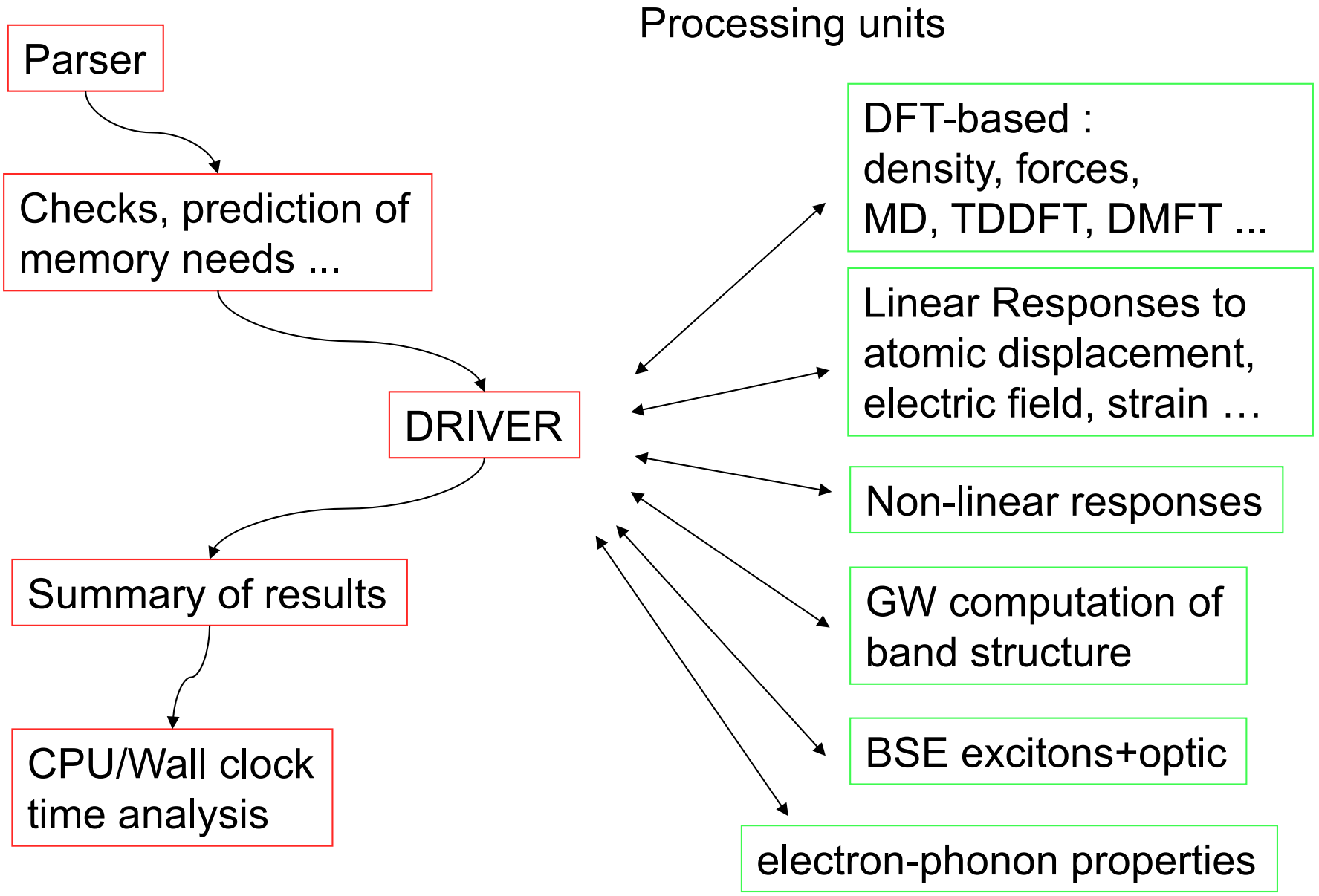
- >1500 phonon band structures from ABINIT

  G. Petretto et al, Comput. Mat. Science 144, 331 (2018)

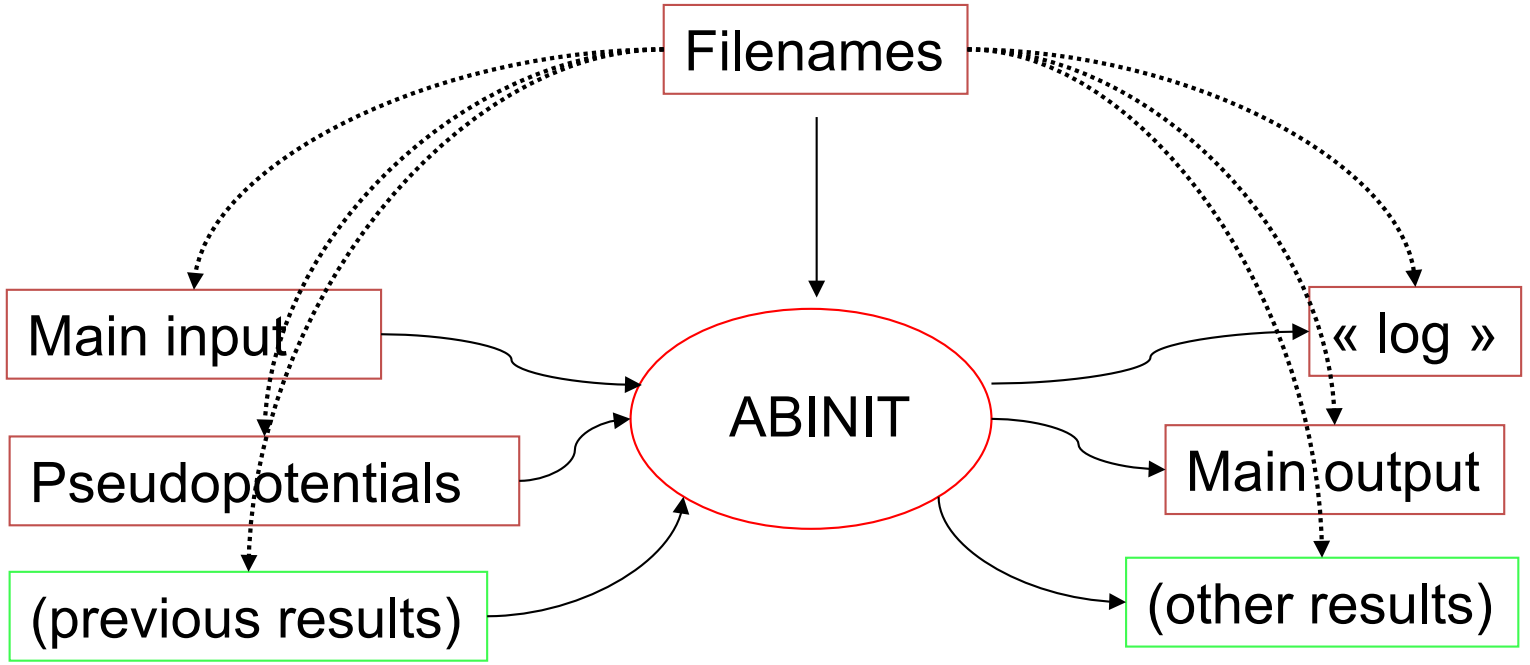
  G. Petretto et al, Scientific data SDATA 17-00473 (2018)

Running ABINIT : basics

ABINIT : the pipeline and the driver



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

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]

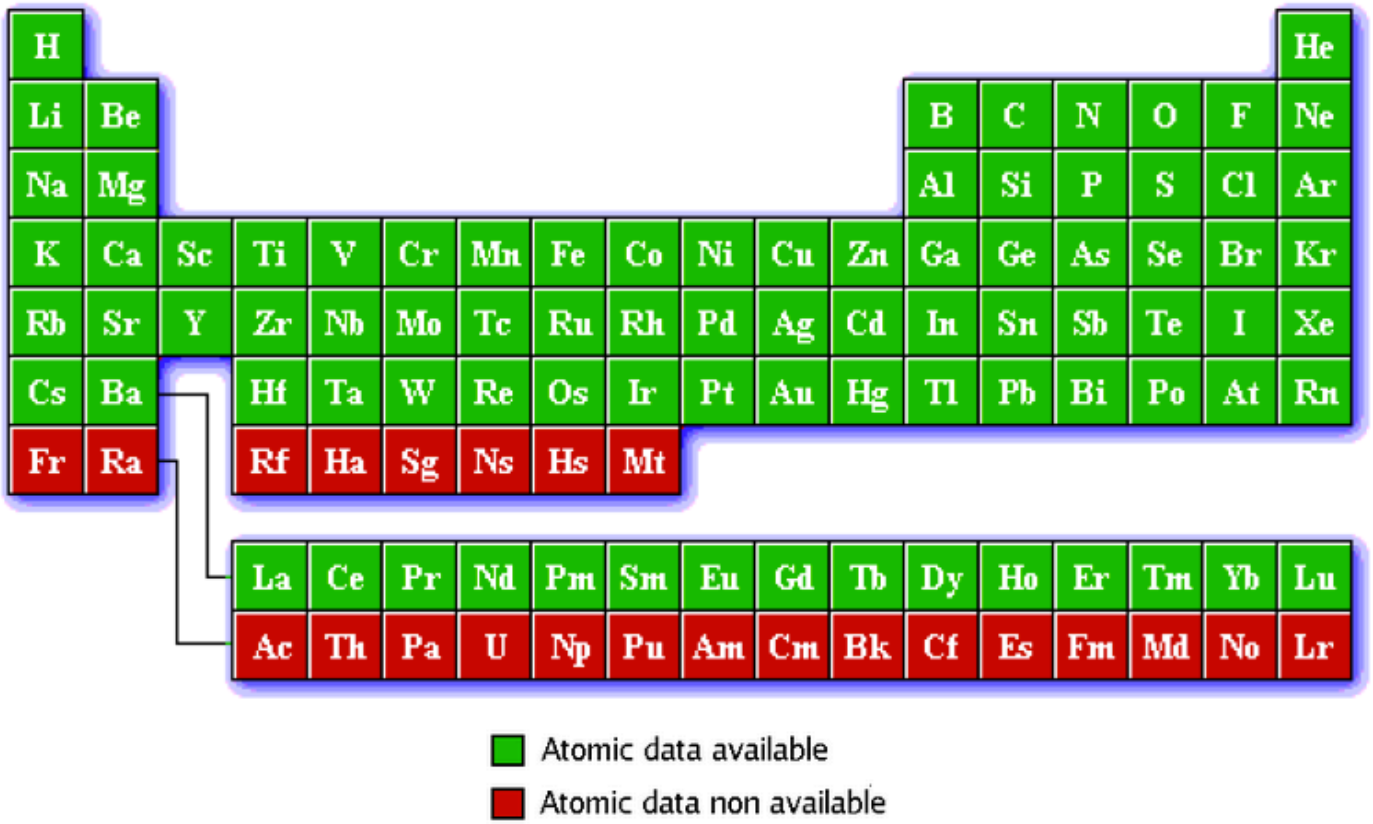
Made of at least 6 lines (more if > 1 type of atoms) with one name/address specified on each of these lines.

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



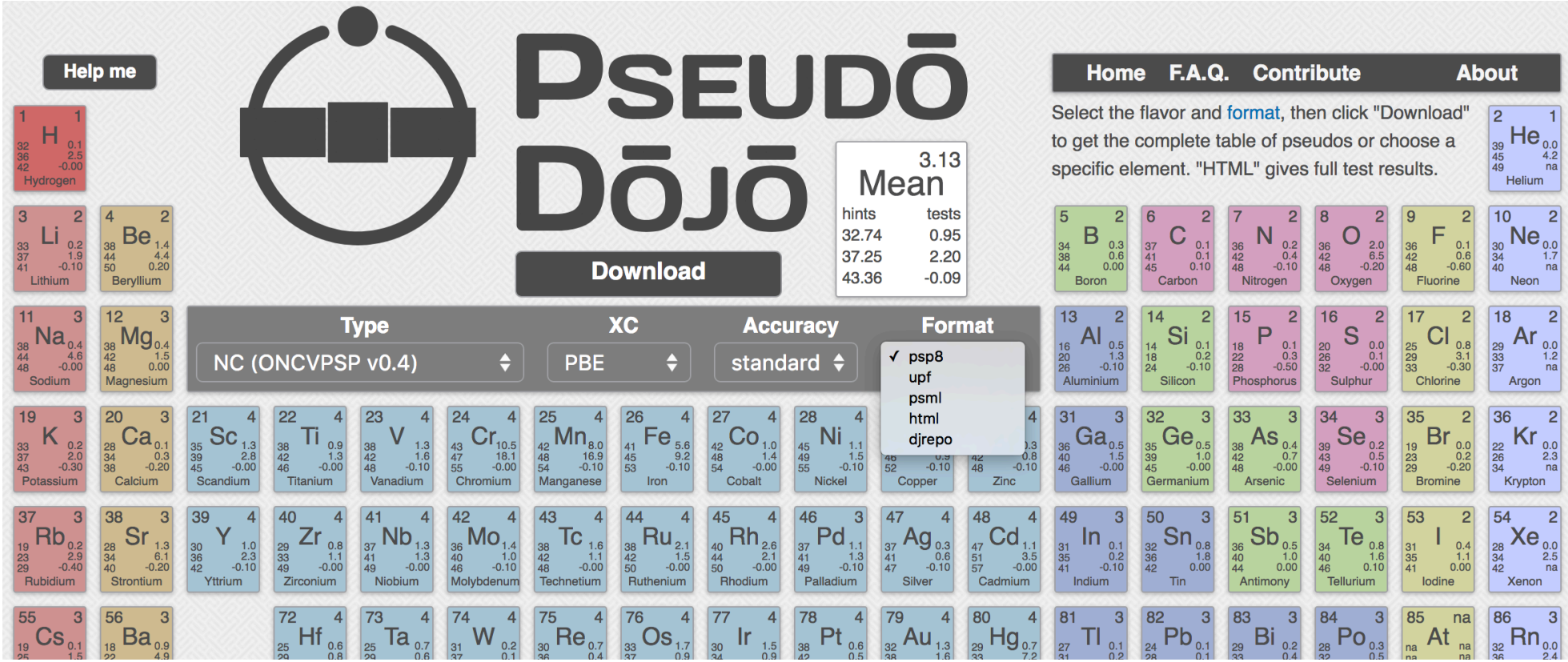
Also, possibility to use : GPAW table, GBRV v1.0 table, or norm-conserving pseudopotentials (e.g. ONCVPSP pseudo generator), or many others

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



A basic ‘input’ file : dihydrogen (I)

```
# H2 molecule in big cubic box
# Characters after  ’#’ or after ‘!’ are comments, will be ignored.
# Keywords + 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)

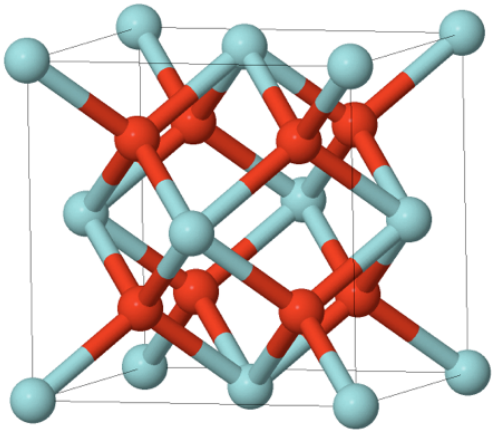
```
# Definition of planewave basis set
ecut 10.0          # Maximal plane-wave kinetic energy cut-off, in Hartree

# 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 !

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 … might be optimized)
  … 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



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

Bilbao Crystallographic Server → Assignment of Wyckoff Positions

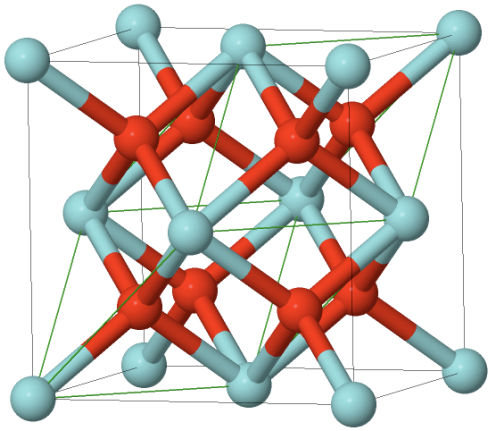
Assignment of Wyckoff Positions

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 3 atoms per primitive cell

Example : cubic zirconium dioxide

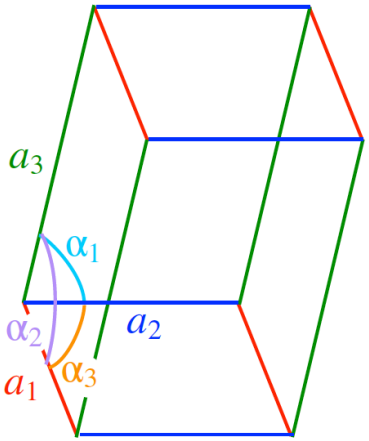


```
natom 3
acell 3*5.01 Angst      NOTE : "*" is a repeater
rprim  0.0 0.5 0.5
       0.5 0.0 0.5
       0.5 0.5 0.0
typat 1 2 2
xred 3*0.0  3*0.25 3*0.75
=> symmetries are found automatically
```

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
```

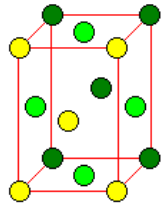
Primitive vectors in ABINIT (rprimd)



- $R_i(j) \rightarrow$ rprimd(j,i)=scalecart(j) × rprim(j,i) × acell(i)

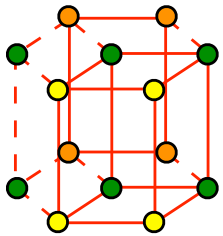
```
scalecart    9.5  9.8  10.0
    rprim    0.0  0.5  0.5
             0.5  0.0  0.5
             0.5  0.5  0.0
```

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



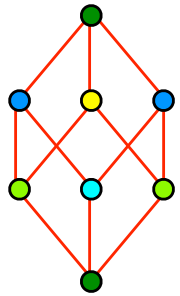
- $a_i \rightarrow$ acell(i) / $\alpha_i \rightarrow$ angdeg(i)

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

hexagonal

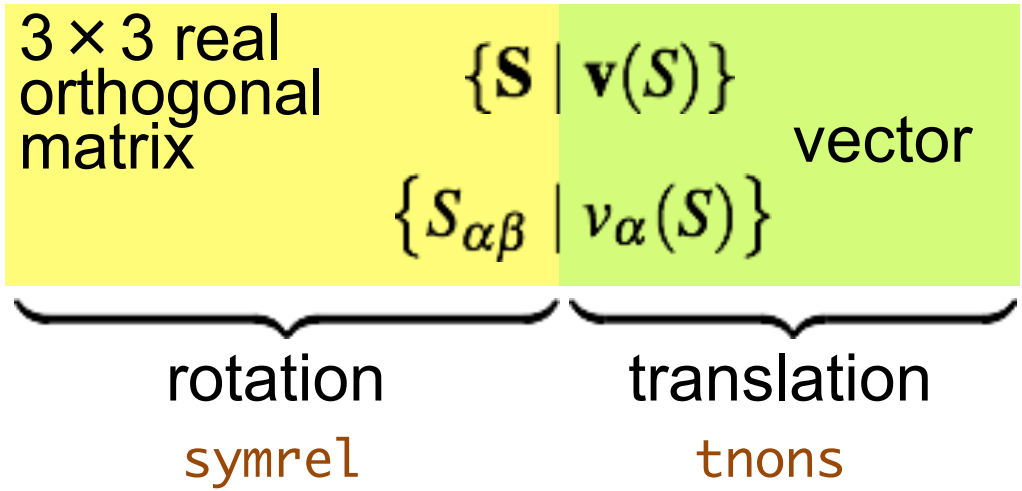
```
 acell    9.0  9.0  9.0
angdeg    48 48 48
```

trigonal



Symmetries in ABINIT

- Seitz notation for the symmetry operations of crystal :



smaller than any primitive vector of the crystal

- Applied to equilibrium position vector of atom κ relative to origin of the cell $\boldsymbol{\tau}_\kappa$, 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 real space lattice.

Planewave basis set

Reciprocal lattice : set of $\mathbf{G}$ vectors such that $e^{iGR_j} = 1$

e^{iGr} has the periodicity of real space lattice

$$u_k(\mathbf{r}) = \sum_G u_k(\mathbf{G})\, e^{iGr}$$

$$\psi_k(\mathbf{r}) = (N\Omega_0)^{-1/2} \sum_G u_k(\mathbf{G})\, e^{i(k+G)r}$$

$$u_k(\mathbf{G}) = \frac{1}{\Omega_o} \int_{\Omega_o} e^{-iGr}\, u_k(\mathbf{r})\, d\mathbf{r}$$

(Fourier transform)

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

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

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

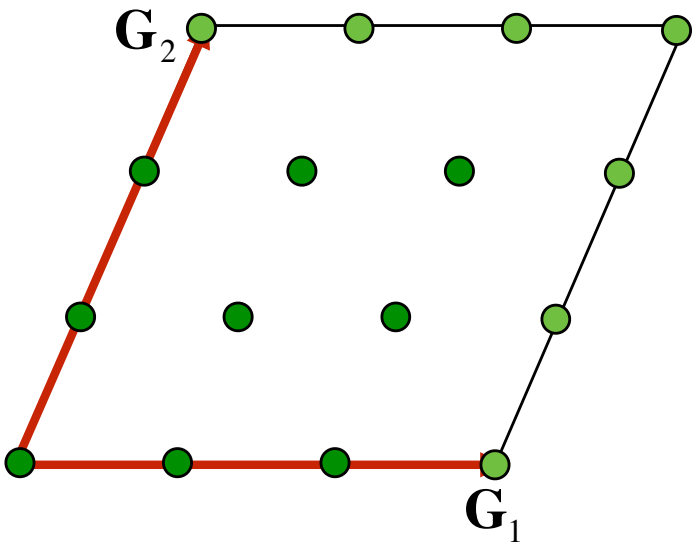
$$\frac{(\mathbf{k}+\mathbf{G})^2}{2} < E_{cut}$$

Plane wave sphere

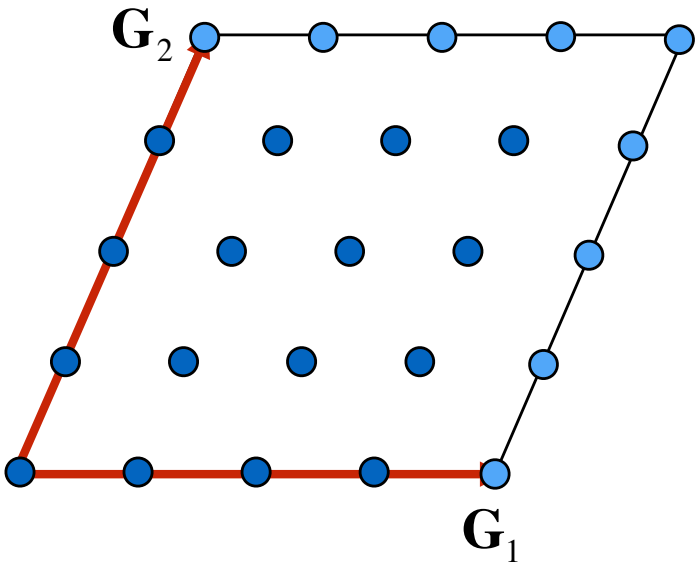
ecut

BZ integration : Monkhorst-Pack grid

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



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

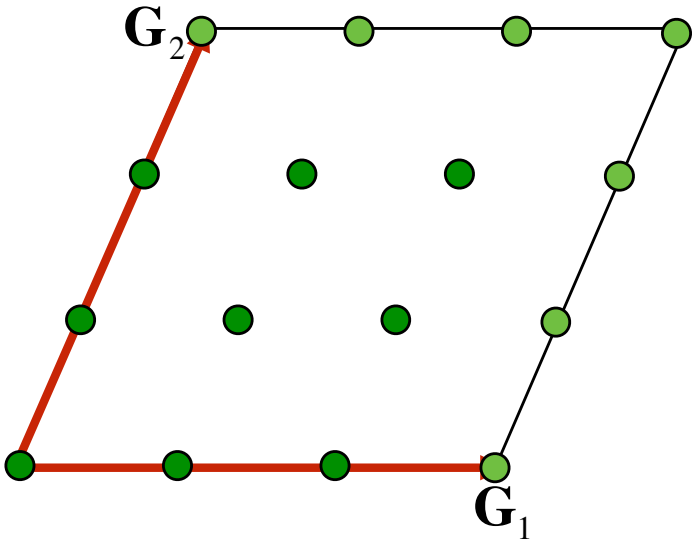


$n_{k1} = n_{k2} = 4$

`ngkpt` nk1 nk2 nk3

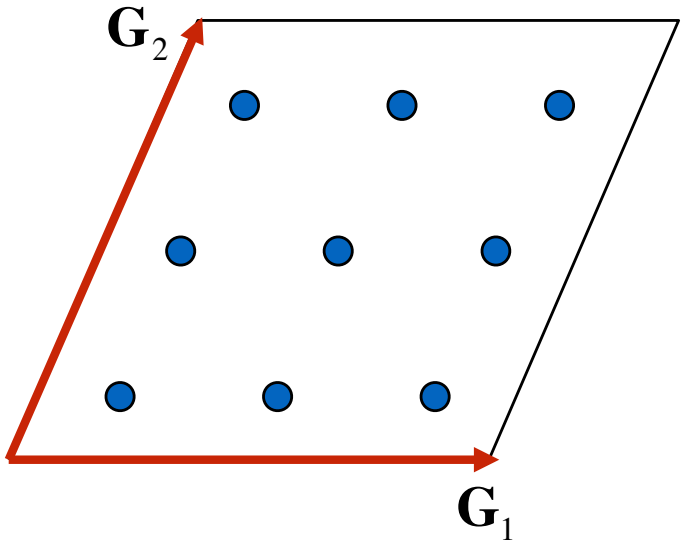
Unshifted and shifted grids

- 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



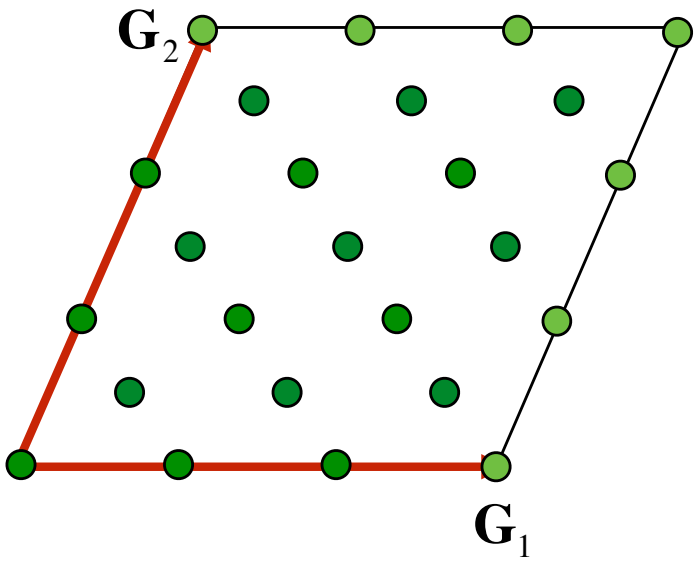
$n_{k1} = n_{k2} = 3$

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.

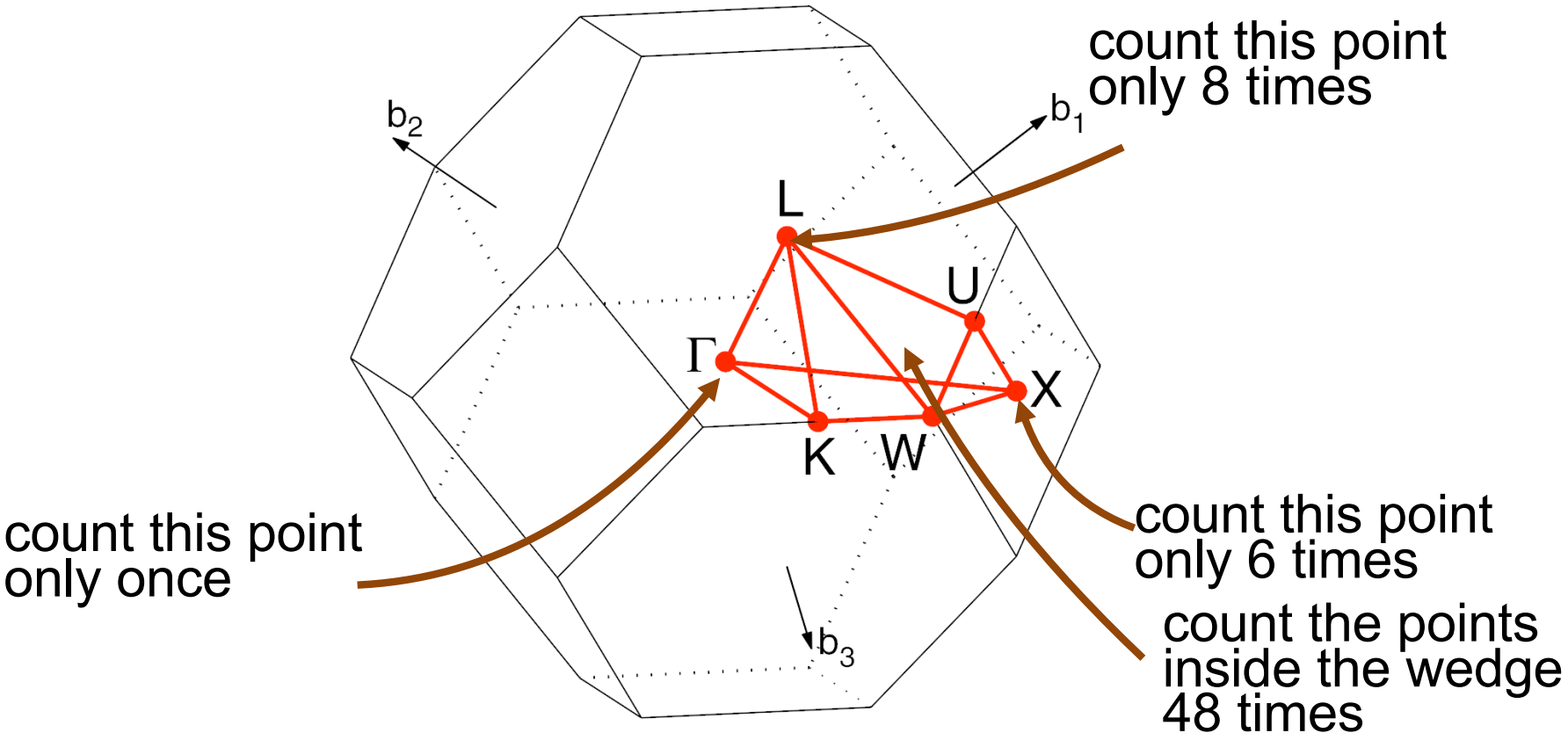


combining unshifted
and shifted by (1/2,1/2)
for $n_{k1} = n_{k2} = 3$

```
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.



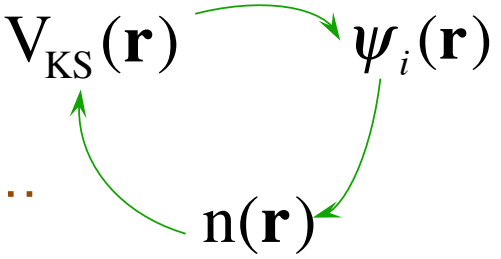
In practice …

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

Details are usually hidden to the user nline, tolrde

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

Target tolerance tolmxf, strfact

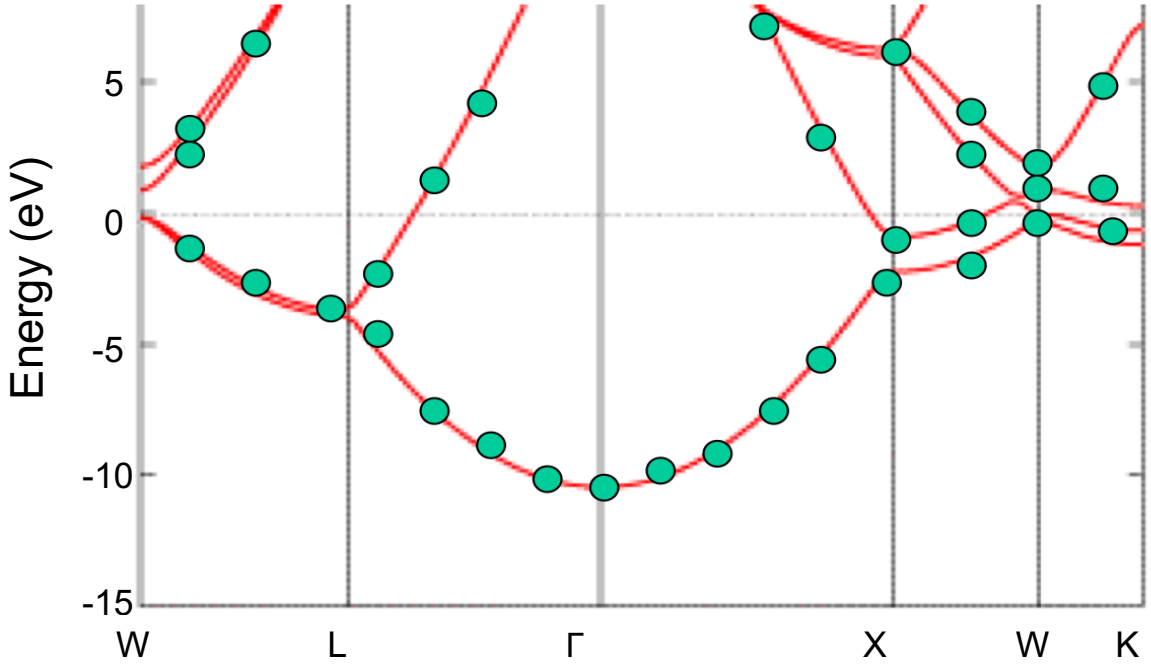
+ Maximal number of loops ntime

Algorithm ionmov

Non-self consistent calculations

- Once the density has been determined self-consistently, 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`).

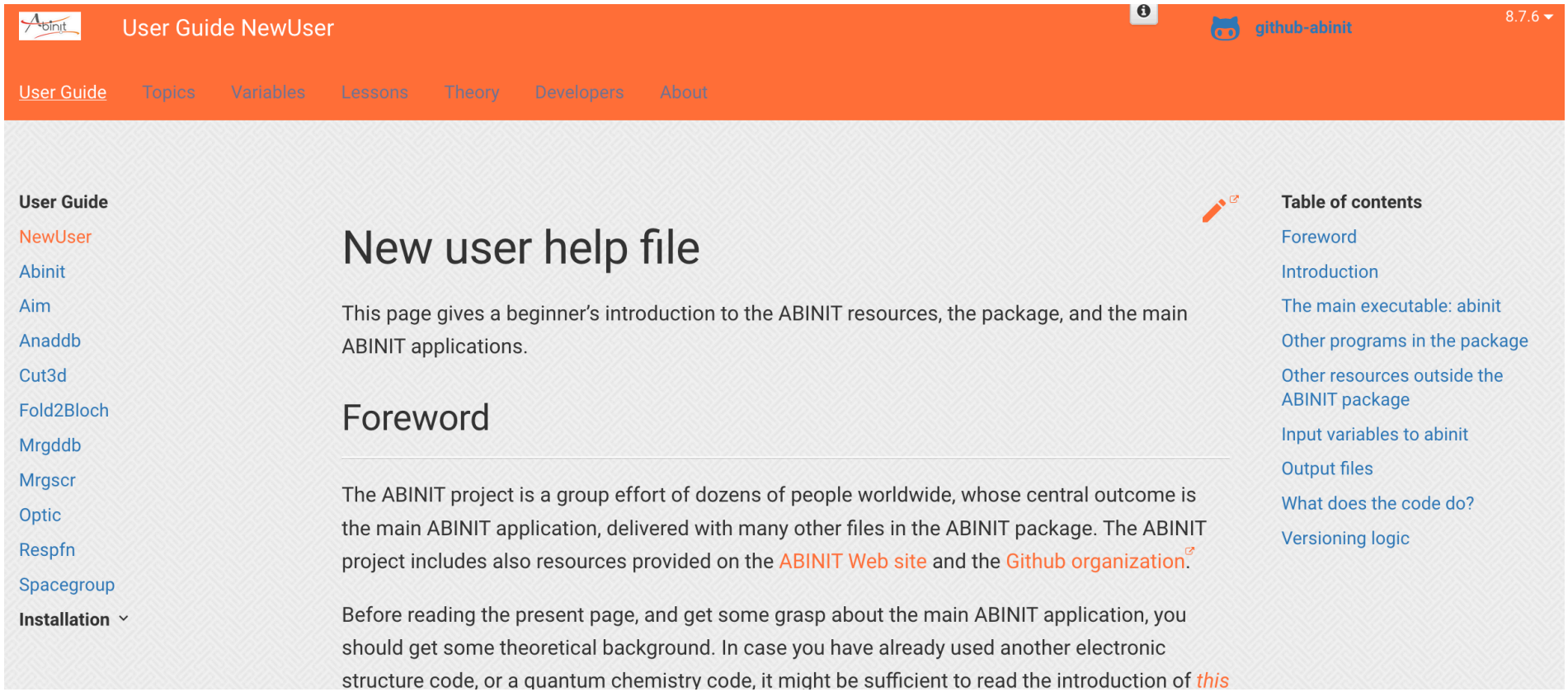


Beyond the basics

Documentation : central ABINIT doc

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

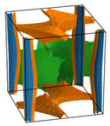
- User's guide + Installations notes
- List of topics, input variables, tutorial (>30 lessons)
- Theory documents including bibliography
- >1000 example input files + reference output (from test set)



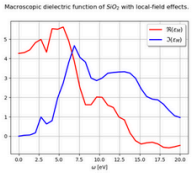
Documentation : abipy galleries

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

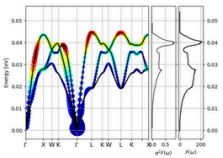
- Plot gallery http://abinit.github.io/abipy/gallery/index.html
- Flow gallery http://abinit.github.io/abipy/flow_gallery/index.html



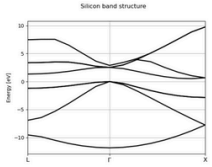
MgB2 Fermi surface



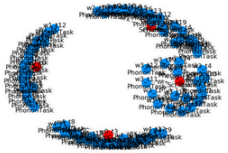
Dielectric function with LFE



Eliashberg function

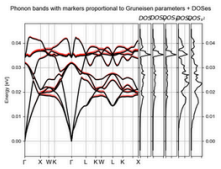


Band structure plot

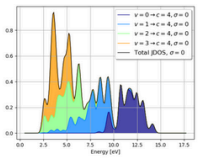


Flow to analyze the convergence of phonons in metals wrt ngkpt and tsmear

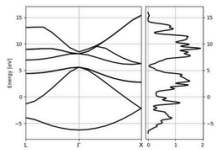
Flow for phonons with DFPT



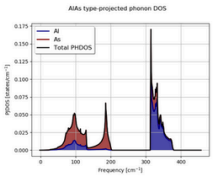
Gruneisen parameters



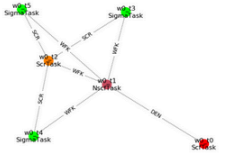
Joint Density of States



Bands + DOS



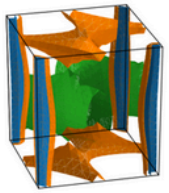
Projected phonon DOS



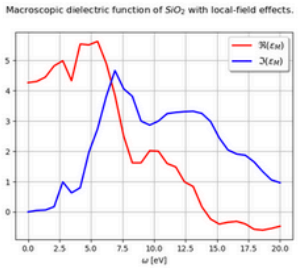
G0W0 Flow with convergence study wrt nband

Optic Flow

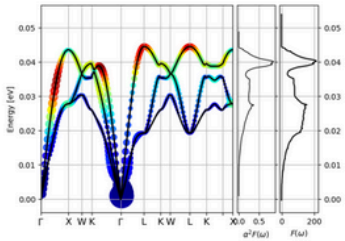
Documentation : abipy galleries



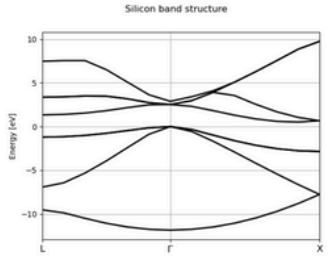
MgB2 Fermi surface



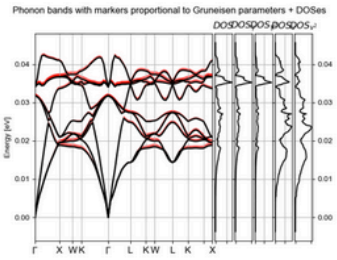
Dielectric function with LFE



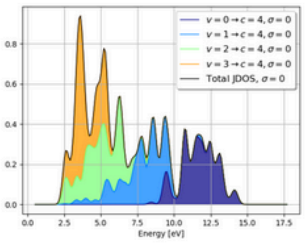
Eliashberg function



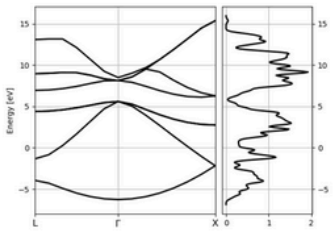
Band structure plot



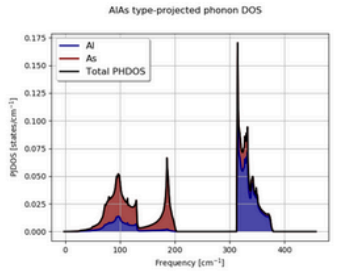
Gruneisen parameters



Joint Density of States

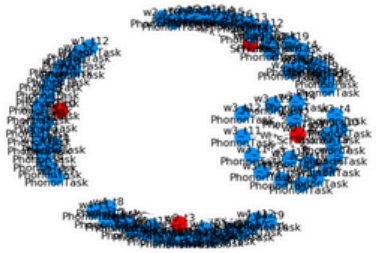


Bands + DOS



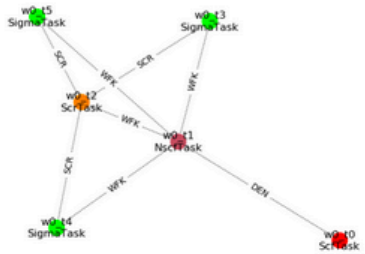
Projected phonon DOS

Documentation : abipy galleries

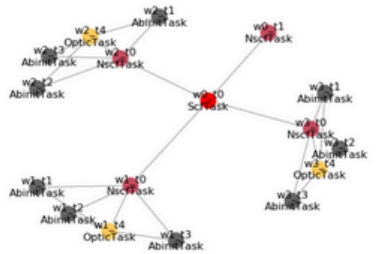


Flow to analyze the convergence of phonons in metals wrt ngkpt and tsmear

Flow for phonons with DFPT



G0W0 Flow with convergence study wrt nband



Optic Flow

Documentation : abitutorials

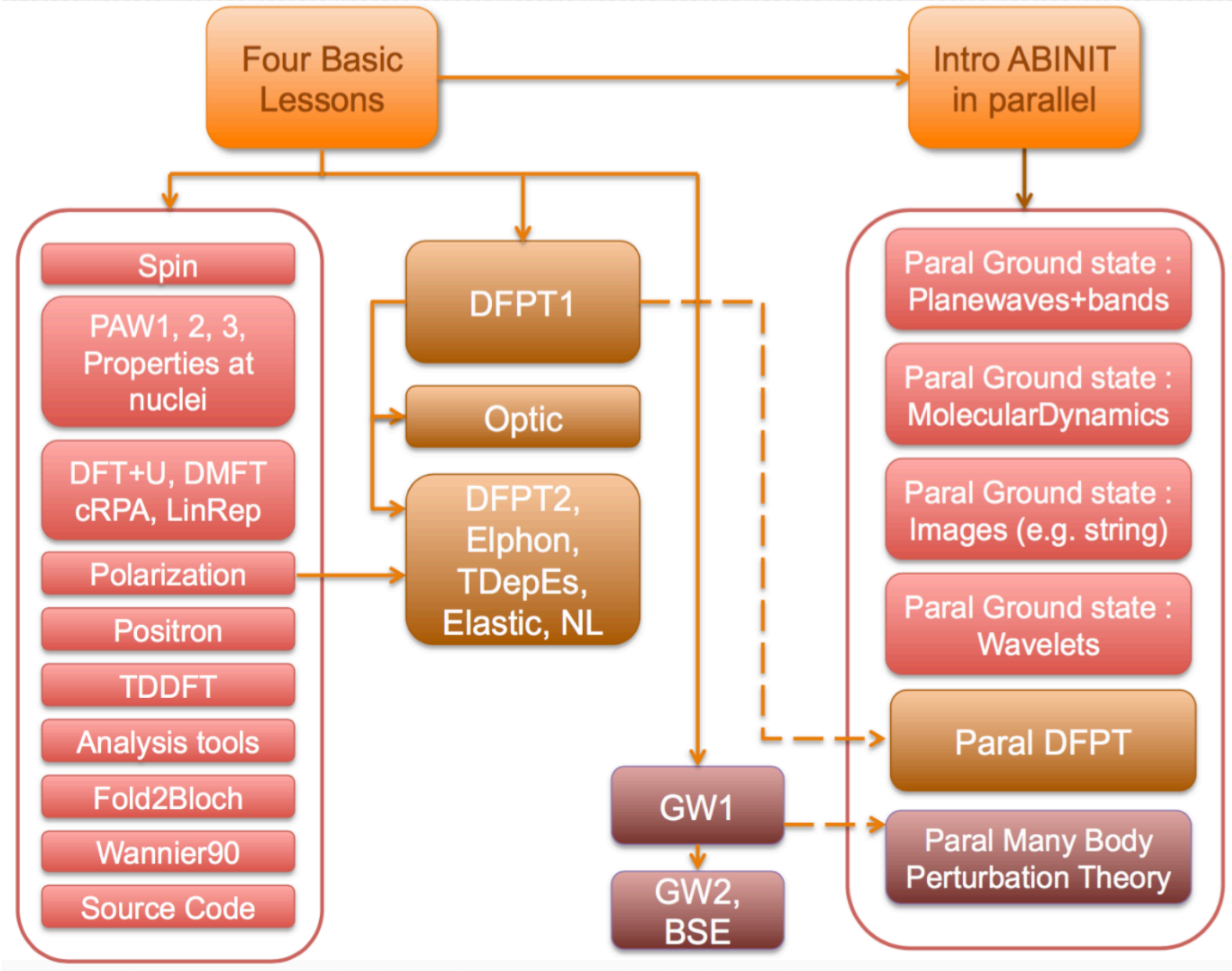
Web site https://github.com/abinit/abitutorials

- 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, DFPT and E-PH abitutorial available for present school

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

Conventional tutorial : global view



This afternoon …

Goal : Temperature Dependence of the Electronic Structure

Track using conventional tutorial http://docs.abinit.org

4 basic lessons + DFPT (RF1+RF2) + TDepES

includes reading abinit user's guide and respfn user's guide

… in principle about 10-15 hours for newcomers.

But "only" 3 hours

⇒ bypass the 4 basic lessons + abinit user's guide

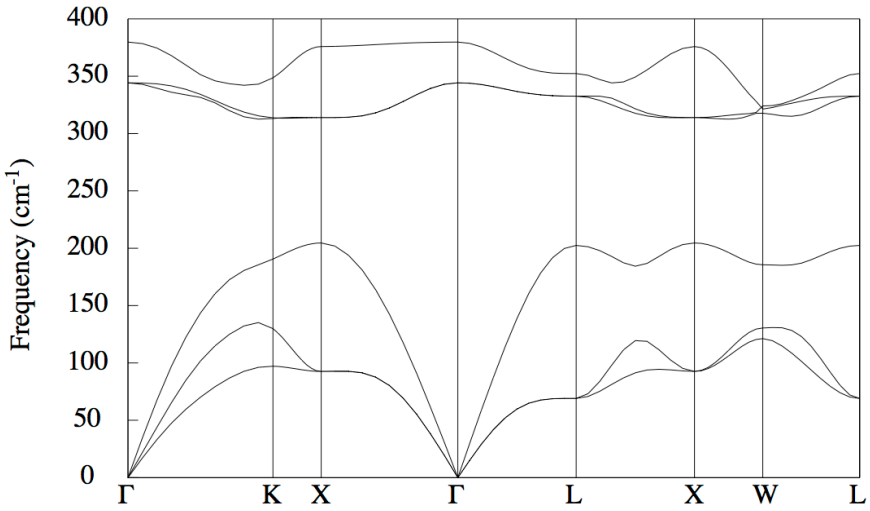
⇒ focus on DFPT (RF1+RF2) + TDepES and respfn user's guide … later you can come back to the "normal" learning of ABINIT ...

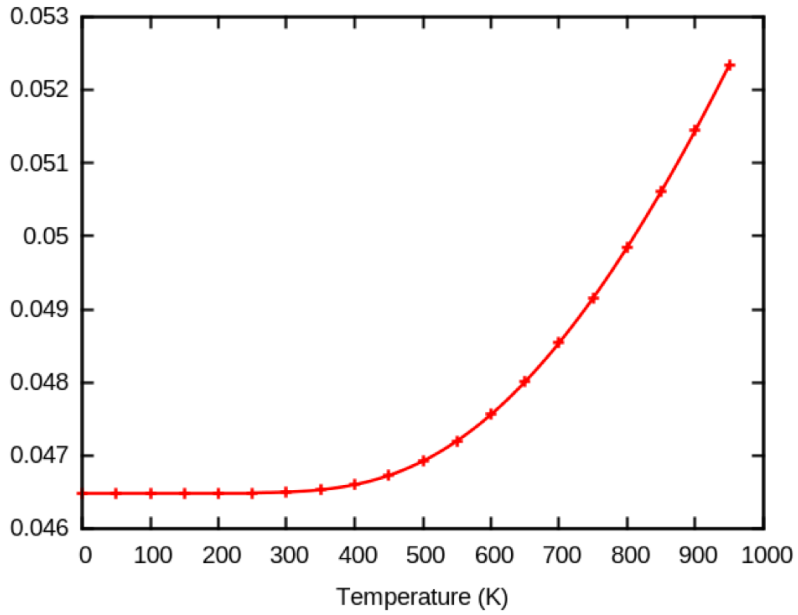
Track with abitutorials : Jupyter notebook based

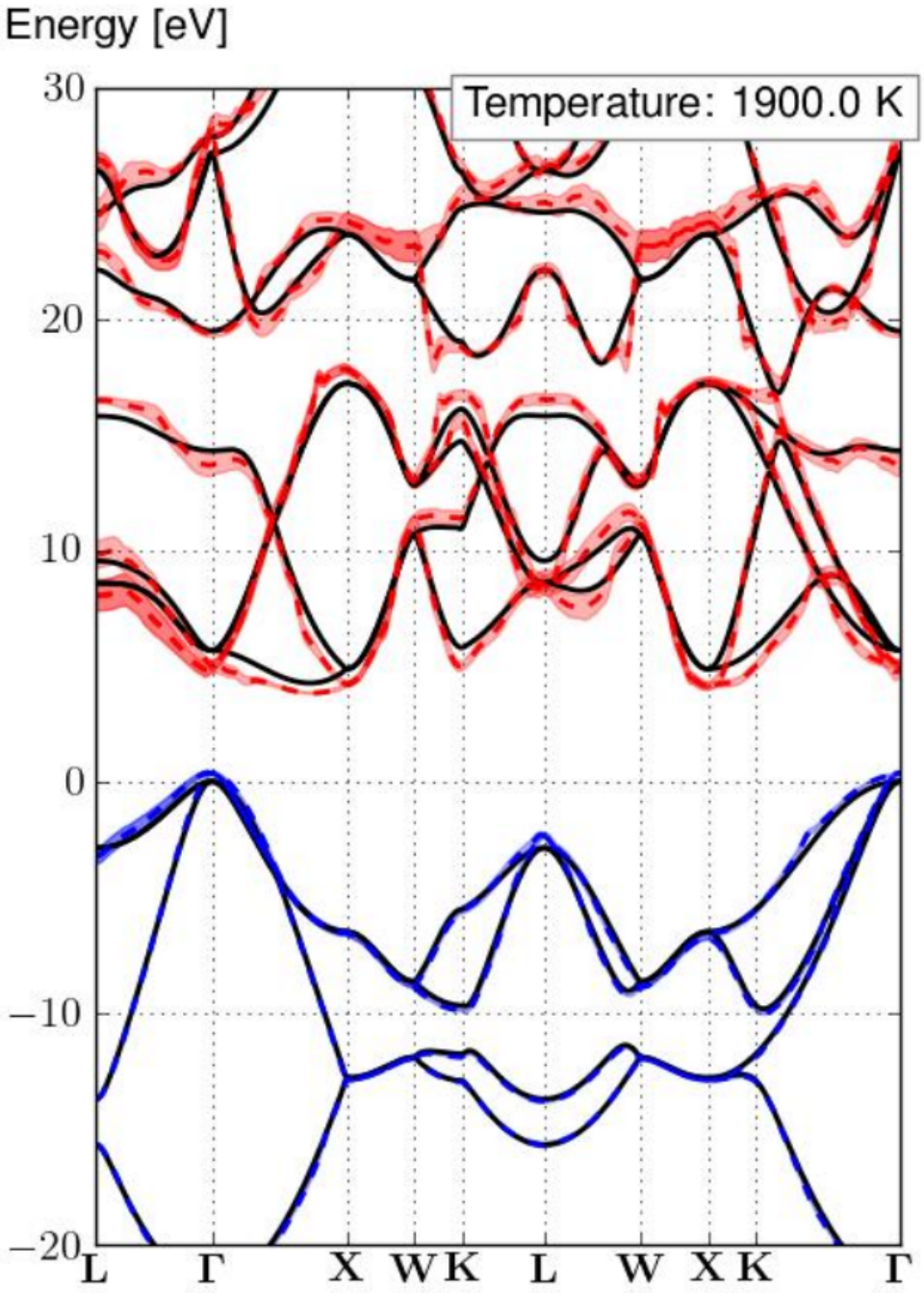
See ABINIT Github Web site

https://github.com/abinit/abitutorials

Conventional DFPT + TDepES tutorial

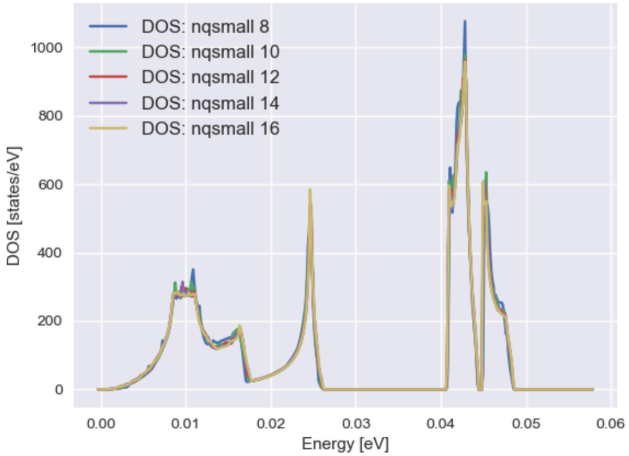


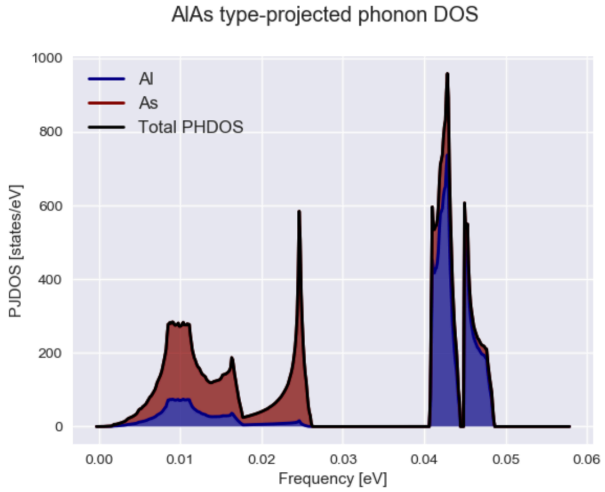


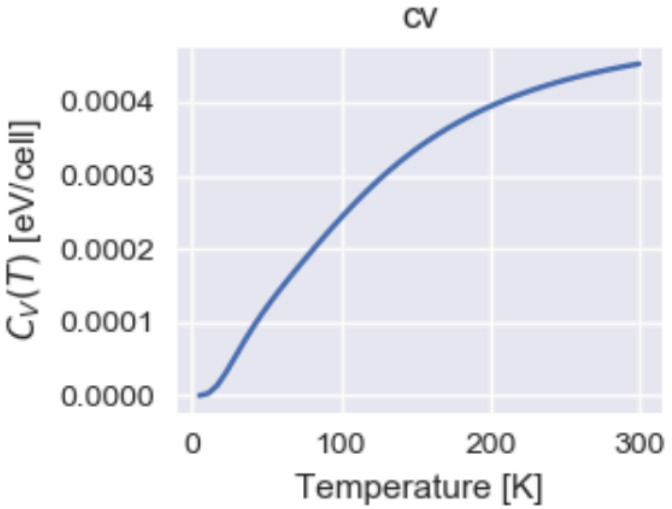


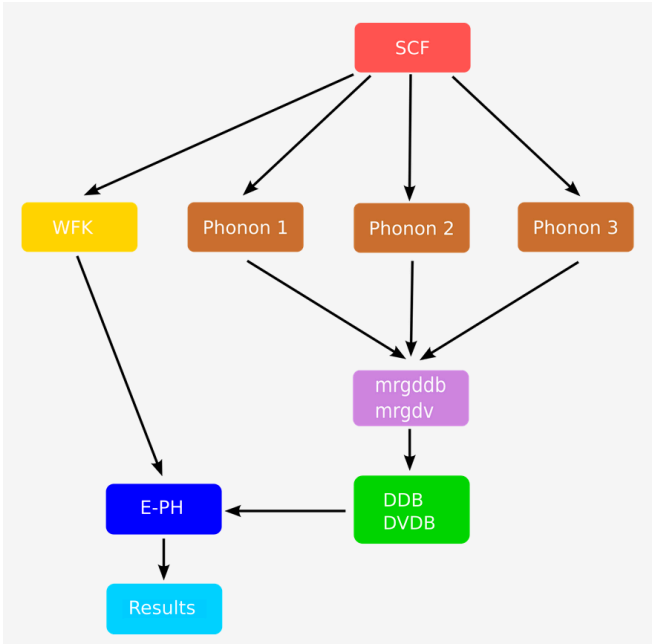
TDepES abitutorial

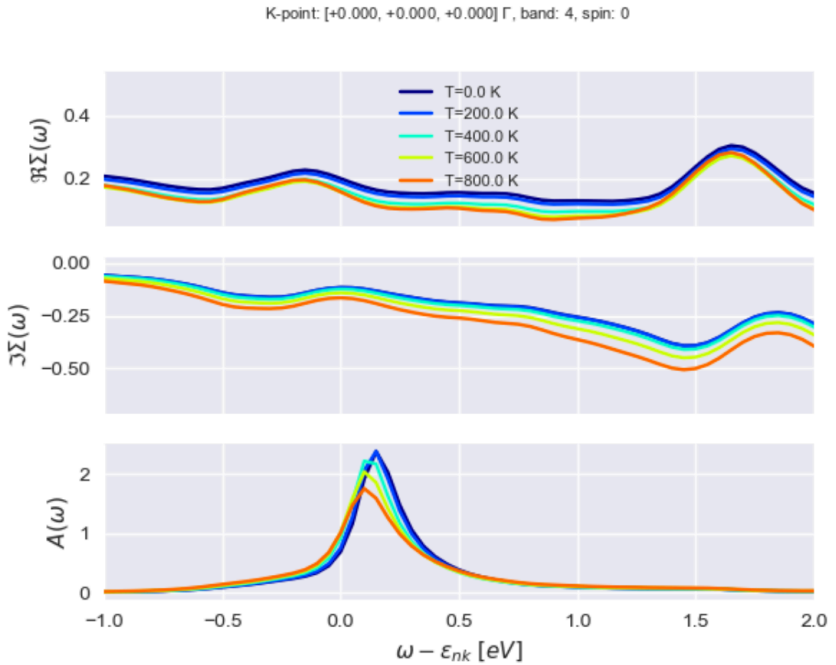
More results visualized ... + more powerful











Introducing
DFPT and e-ph lessons

Basic quantities and their handling

$$\delta\varepsilon_{\Gamma n}^{ZPM} = \frac{1}{N_{\vec{q}}} \sum_{\vec{q}j} \frac{\partial \varepsilon_{\Gamma n}}{\partial n_{\vec{q}j}} \frac{1}{2}$$

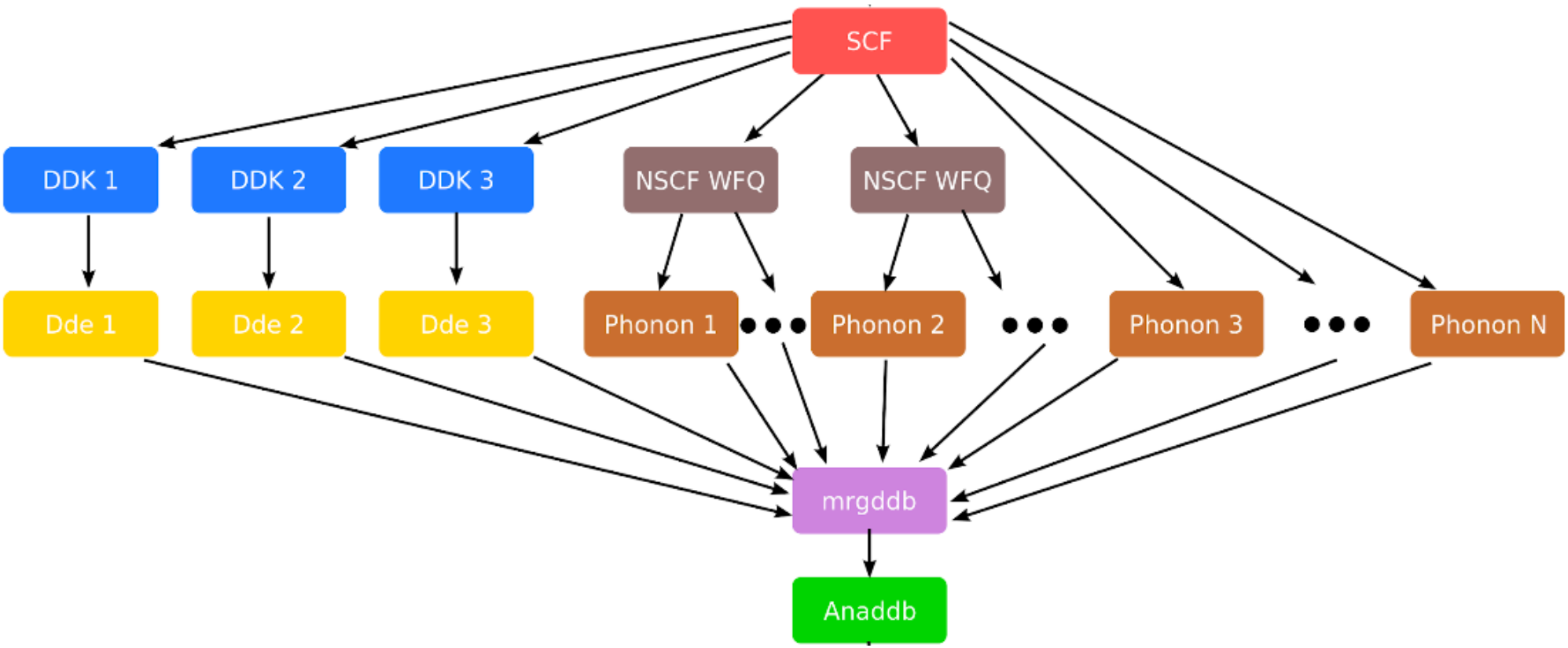
$$\frac{\partial \varepsilon_{\Gamma n}(Fan)}{\partial n_{\vec{q}j}} = \frac{1}{\omega_{\vec{q}j}} \Re \sum_{\kappa a \kappa' b n'} \frac{\left\langle \phi_{\Gamma n} \middle| \nabla_{\kappa a} H_{\kappa} \middle| \phi_{\vec{q}n'} \right\rangle \left\langle \phi_{\vec{q}n'} \middle| \nabla_{\kappa' b} H_{\kappa'} \middle| \phi_{\Gamma n} \right\rangle}{\varepsilon_{\Gamma n} - \varepsilon_{\vec{q}n'}} \frac{\xi_{\kappa a}(\vec{q}j)\xi_{\kappa' b}(-\vec{q}j)}{\sqrt{M_{\kappa} M_{\kappa'}}} e^{iq.(R_{\kappa' b} - R_{\kappa a})}$$

Files :

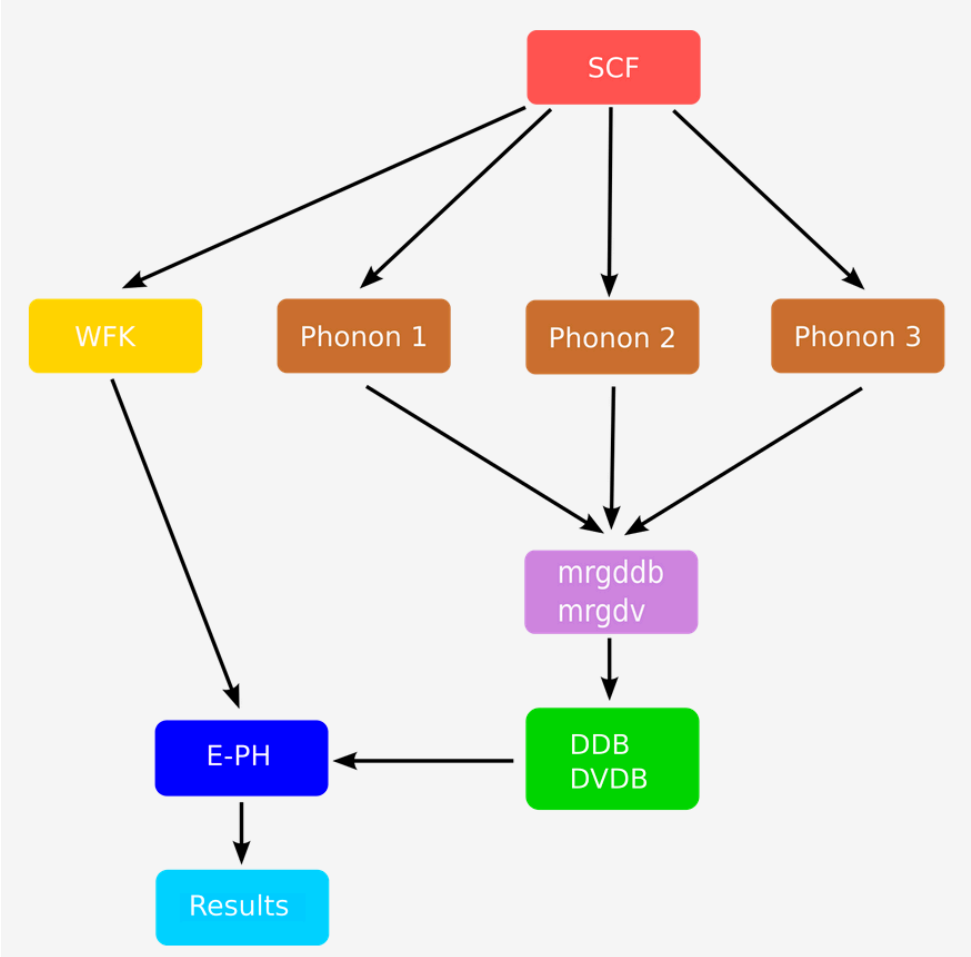
From unperturbed ground-state calculation
… eigenenergies (_EIG.nc) and wavefunctions at k and k+q (_WFK)

From phonon DFPT calculations with wavevector q
… energy derivatives (_DDB), electron-phonon matrix elements (_GKK.nc),
first-order wavefunctions (_1WFx), and
2nd-derivative of eigenenergies - factor in blue (_EIGR2D.nc &_EIGI2D.nc),

Flow for phonons



Flow for electron-phonon (simplified)



Two implementations !

Common : ABINIT calculations of phonons (F90).
Then :

(1) Extraction of electron-phonon matrix elements + postprocessing python scripts
(well-established)
conventional tutorial

(2) Separate (fast) calculations of electron-phonon matrix elements on interpolated potential, and (fast) treatment, all inside ABINIT
(under development)
abitutorial `optdriver` 7

Some input variables

In ABINIT:
DFPT implemented by considering perturbations in which one atom is moved along one real space direction.

`ipert` number of the atom whose position is perturbed (max value `natom`)
`idir` 1, 2 or 3, for the three primitive vector directions

`pertcase=3*(ipert-1)+idir` could run from 1 to `natom`

Input variables define a subset of perturbations.
`rfpert` : gives lower an upper bound for `ipert`
`rfdir` : 3 integers 0 or 1, to indicate whether corresponding `idir` must be considered.

Symmetries are used to reduce the number of independent such perturbations, and possibly reduce number of k points (little group).

Datasets : built-in flow capability of ABINIT

`ndtset` number of datasets

-Input variables are by default vallid for all datasets
-If appended with a number, valid only for that dataset
-Both can be present : the specific value supercedes the generic value for the particular dataset
-Series can be defined with metacharacters.
‘:’ is used to initialize the series
‘+’ is used to increment the series

```
ecut   10
ecut1   10
```

```
ecut:   10      ecut+   2
```

equivalent to

```
ecut1   10
ecut2   12
ecut3   14
ecut4   16
```

- Chain calculations : output of one dataset can be input of next dataset
  `getwfk2   -1` take the wavefunctions from previous dataset as input

Datasets : complex flow with double-loop

ndtset number of datasets
udtset maximum indices of an inner and an outer loop
Metacharacter '?' to indicate full validity for one of the indices

Example

```
ndtset 6     udtset 3 2
ecut1?  10          ecut2?  15          ecut3?  20
optdriver?1  0      optdriver?2  1
```

… equivalent to …

```
ndtset 6
ecut11  10      optdriver11  0
ecut12  10      optdriver12  1
ecut21  15      optdriver21  0
ecut22  15      optdriver22  1
ecut31  20      optdriver31  0
ecut32  20      optdriver32  1
```

File naming convention

‘Root name’ _DS ’datasetnumber’ _’File specifier’

Example :

trf2_1o_DS99_DDB

Root name given in the ‘files’ file

Number of the dataset

This is a ’Derivative DataBase’

For first-order wavefunction files, the pertcase is also mentioned
‘Root name’ _DS ’datasetnumber’ _1WF’pertcase’

Example :

trf2_1o_DS99_1WF12

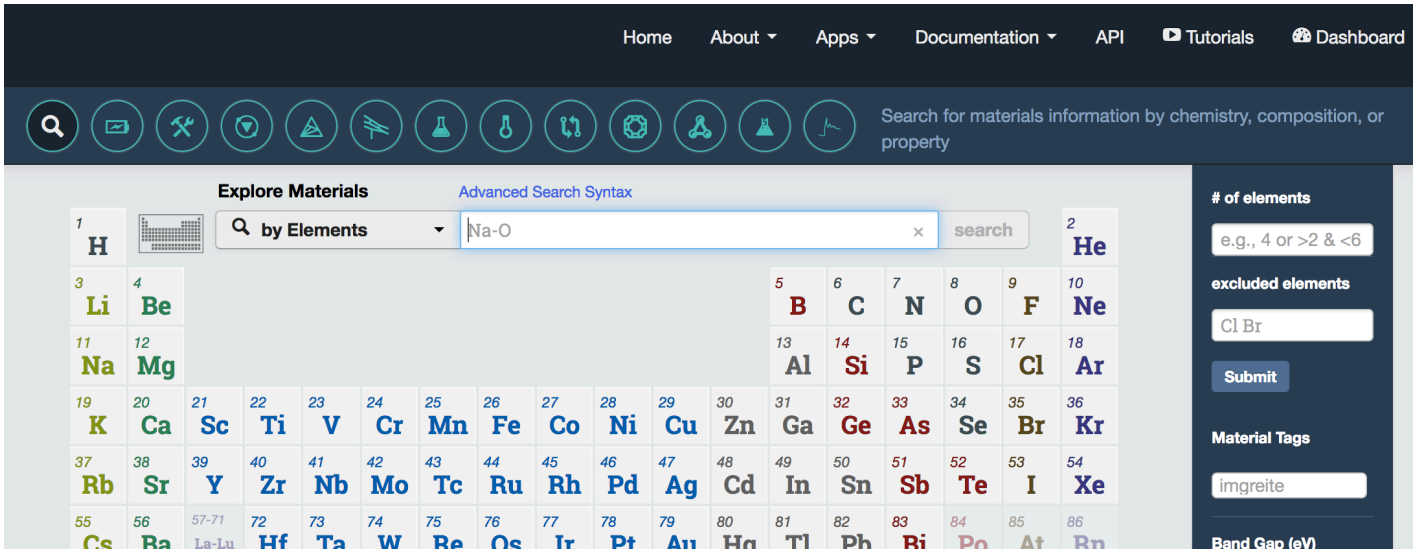
12 for ipert=4 idir=3
The fourth atom is moved in the third direction

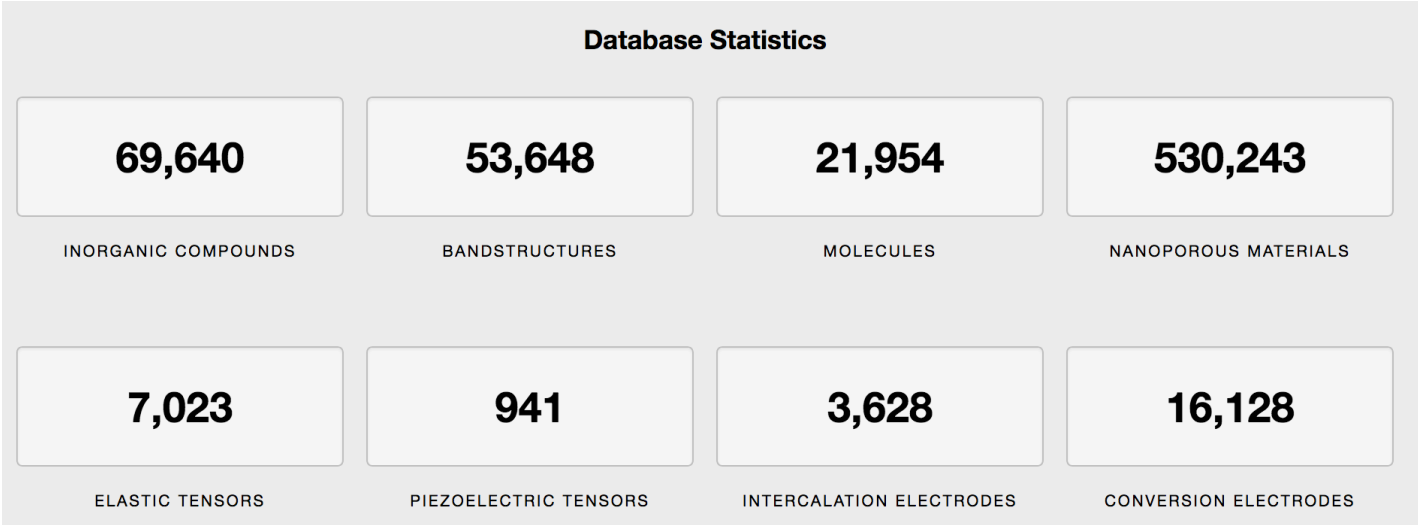
ABINIT
phonon band structures
from the
Materials Project web site

Materials Project

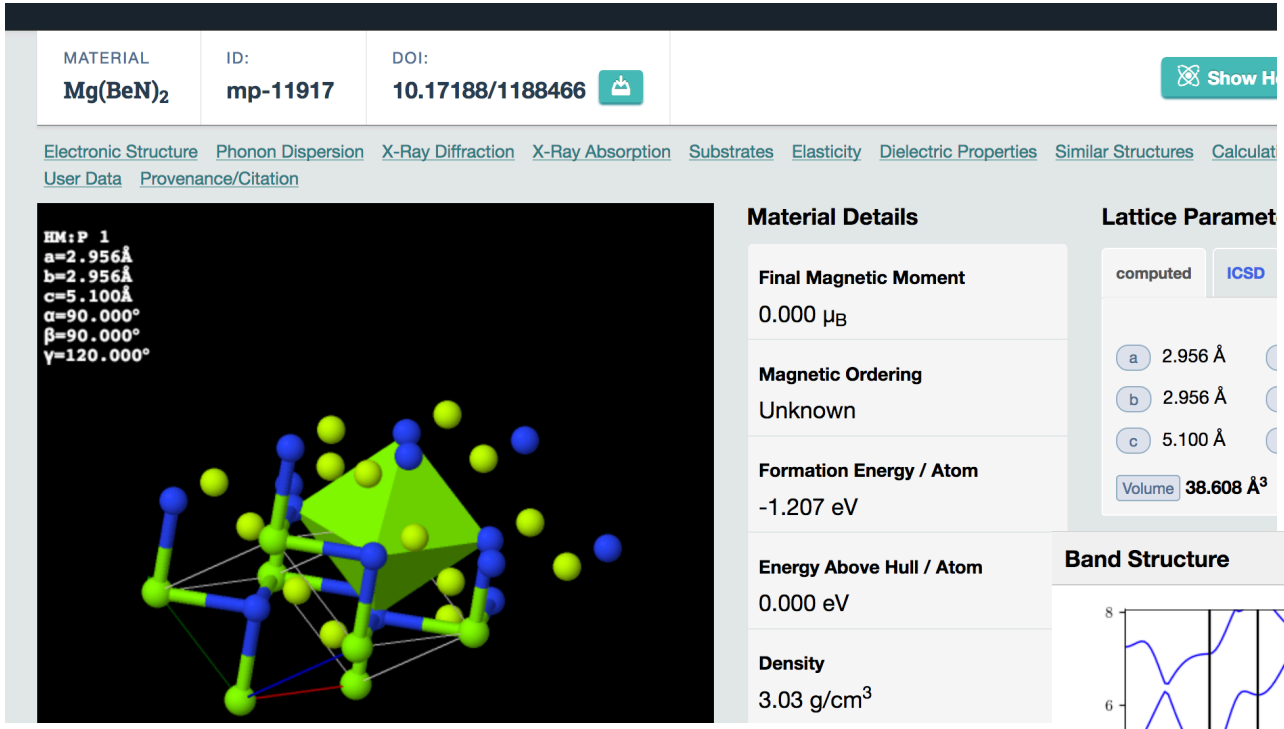
https://materialsproject.org

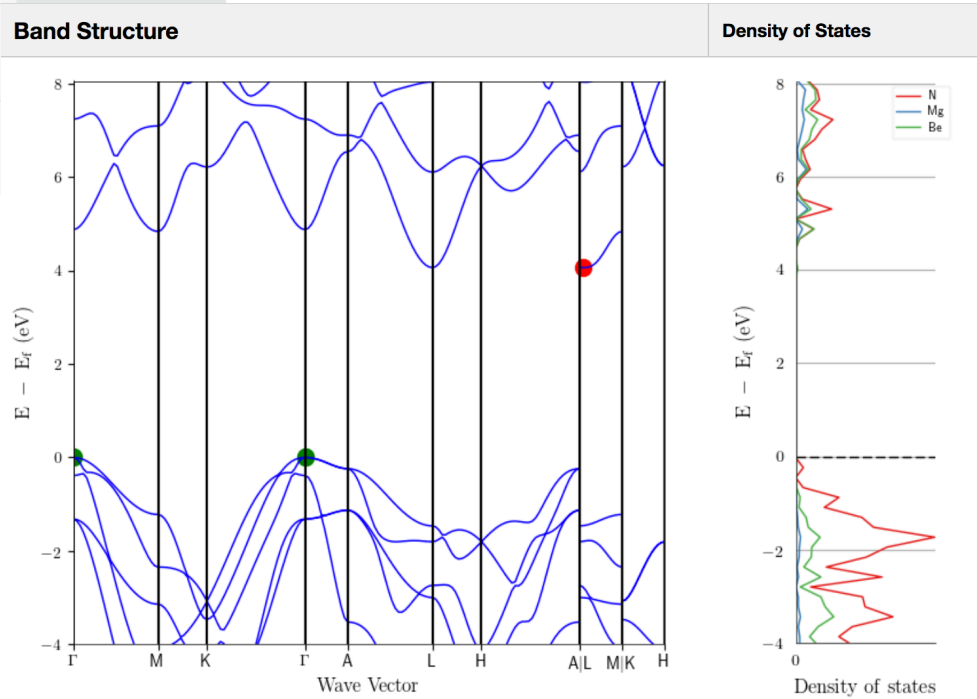
Database of computed properties of materials (ICSD set)





Materials Project : an example

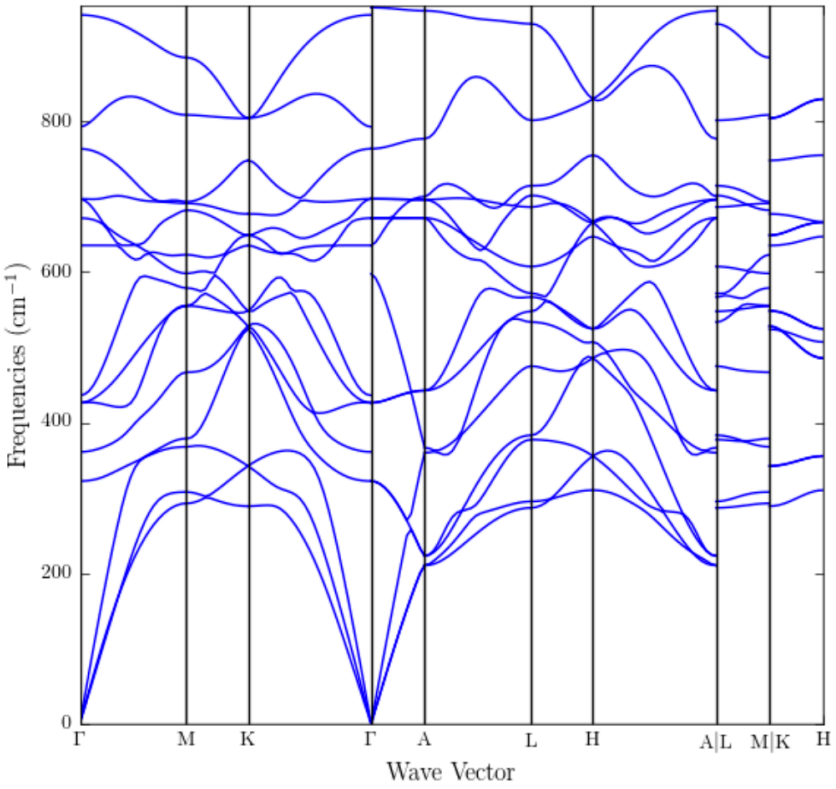


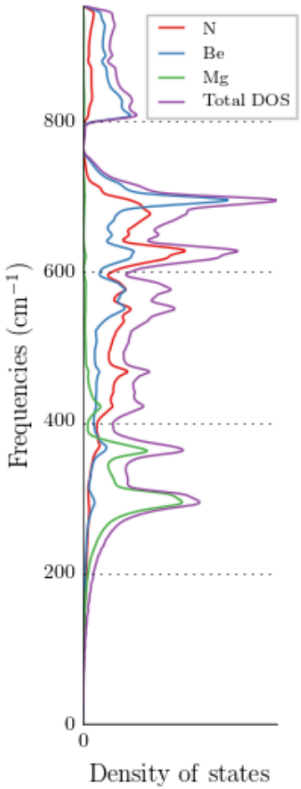


https://materialsproject.org/materials/mp-11917

Materials Project, ABINIT phonons

Phonon dispersion | **Density of States**





Warning! These calculations were performed using a PBEsol exchange correlation functional in the framework of DFPT using the Abinit code. Please see the wiki for more info. ×

Availability of > 1000 consolidated ABINIT phonon band structures

https://materialsproject.org/materials/mp-11917/#phonon-dispersion

Technology : ABINIT, Abipy, pymatgen, fireworks
Need to be in « Django » phonon group. Soon publicly available.
https://materialsproject.org/groupadd/mirandamagic

Taking advantage of DDB files

jupyter nbviewer

JUPYTER FAQ

abitutorials / abitutorials

Back to the main Index

Post-processing DFPT calculations with the DDB file

This notebook explains how to use AbiPy and the DDB file produced by Abinit to analyze:

- Phonon band structures including the LO-TO splitting in heteropolar semiconductors
- Phonon fatbands, phonon DOS and projected DOS
- Born effectives charges $Z^*_{\kappa,\alpha\beta}$ and the dielectric tensors $\epsilon^{\infty}_{\alpha\beta}$, $\epsilon^{0}_{\alpha\beta}$
- Thermodynamic properties in the harmonic approximation

In the last part, we discuss how to use the `DdbRobot` to analyze multiple DDB files and perform typical convergence studies.

Table of Contents

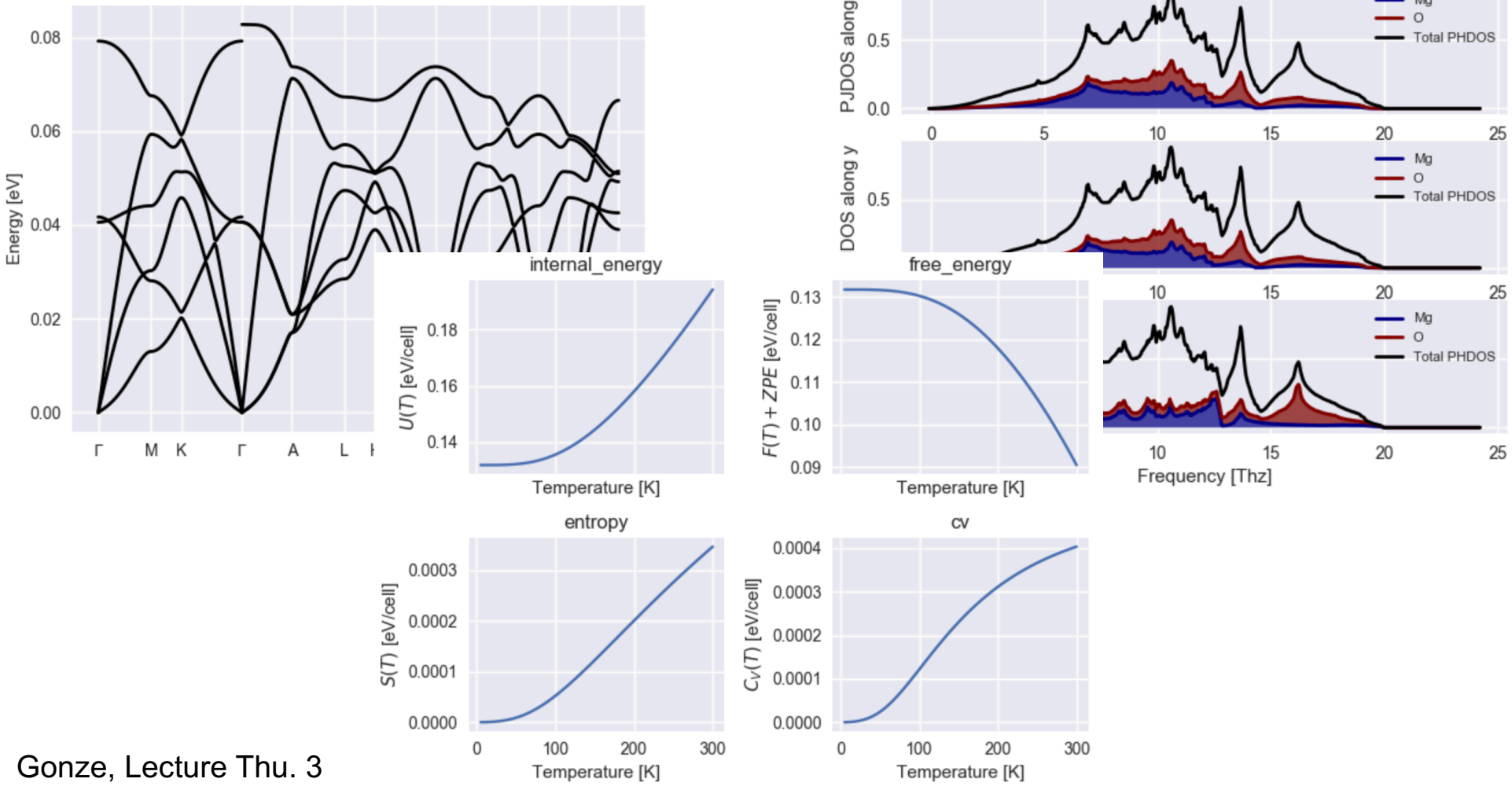
Downloading DDB files from MP

Integration with the materials project database

AbiPy, pymatgen and fireworks have been used by Petretto et al to compute the vibrational properties of more than 1500 compounds with Abinit. The results are available on the materials project website. The results for this phase of MgO are available at https://materialsproject.org/materials/mp-1009129/

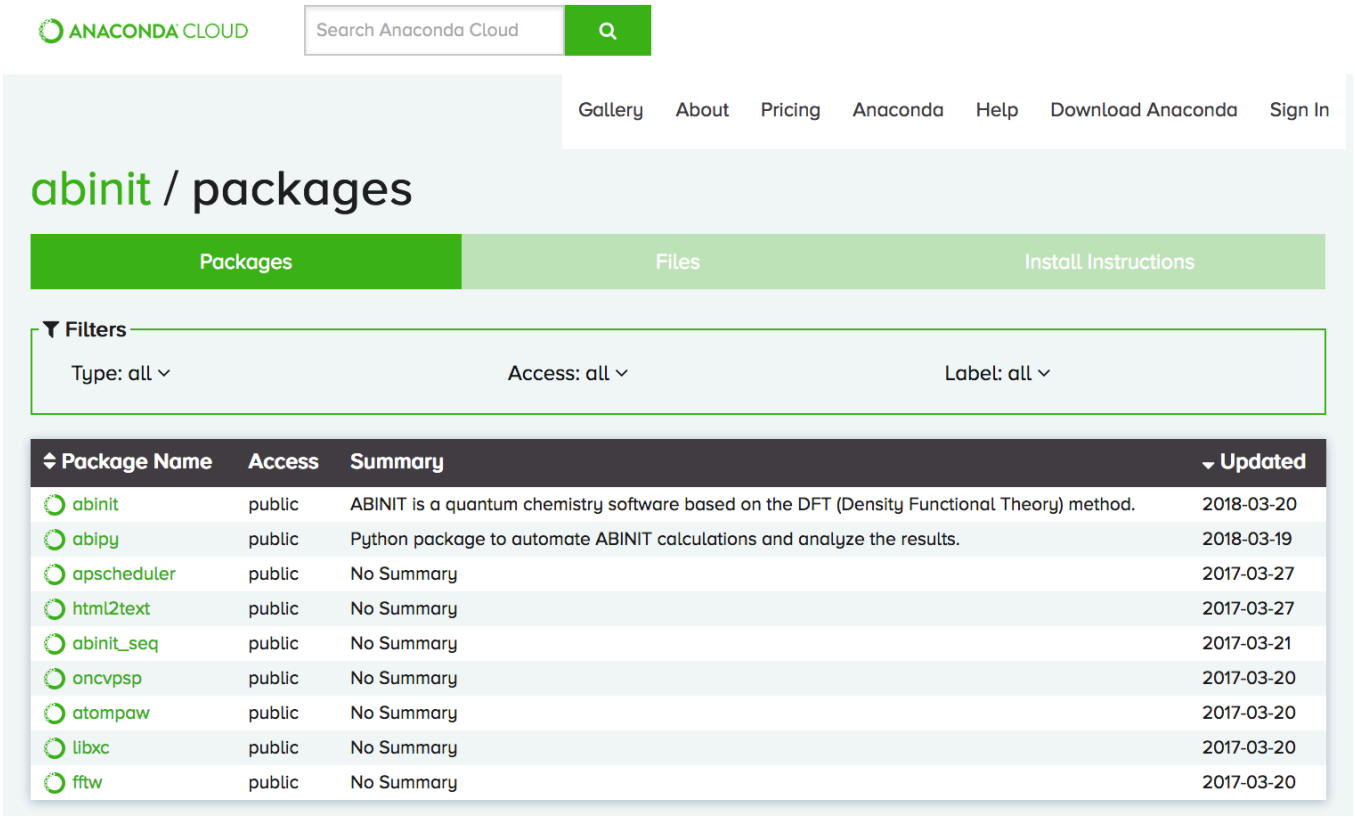
To fetch the DDB file from the materials project database and build a `DdbFile` object, use:

```
ddb = abilab.DdbFile.from_mpid("mp-1009129")
```



Installing abinit+python+netCDF+ … ?

ABINIT+python is a powerful approach, but needs many libraries …
Solution : use Anaconda. An abinit channel has been developed.
With respect to ‘pip install’ : availability of precompiled executable & library, including netCDF.
https: //anaconda.org/abinit/repo



ABINIT

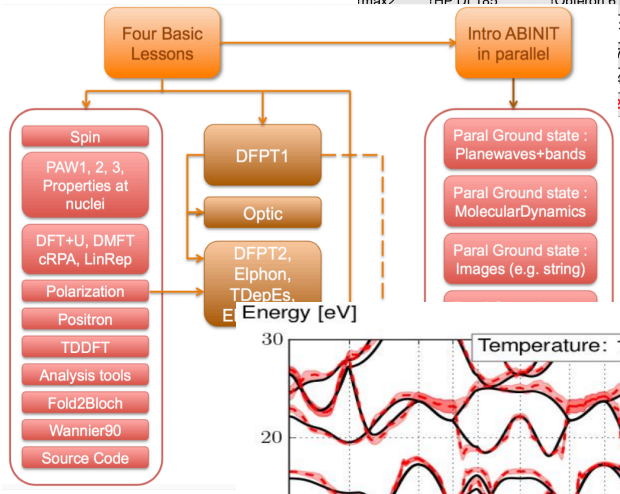
Name	Brand	CPU / Freq	# cores	RAM	OS	misc
abiref	HP DL360 gen9	Xeon E5-2670v3/ 2.30	2 x 24	32GB	CentOS 7.2	Ref
bob	Dell R430	Xeon E5-2603v3/ 1.60	2 x 6	8GB	Fedora 23	
buda	SuperMicro	Xeon X5570/ 2.7	2 x 4	12GB	CentOS 6.8	2xGPU K40 2xGPU C1060
coba2	HP Z400	Xeon W3520/ 2.7	4	12GB	CentOS 6.5	
cronos	HP DL185 G7	AMD Opteron 6276/ 2.3	2 x 16	16GB	Debian 5.0	
graphene	Apple MacPro	Xeon E5-2697/ 2.7	1 x 12	64GB	MacOS X 10.12	
ibm8	IBM Power S824	Power8/ 3.0	4	8GB	AIX 7.2	
inca	virtual machine	Opteron 6276/ 2.3	12	30GB	CentOS 6.9	
max2	HP DL185	Opteron 6140/ 2.6	2 x 8	12GB	Slinux 6.1	
		30/ 3.2	6	16GB	openSUSE 12.1	
		70/ 2.9	2 x 4	12GB	CentOS 5.11	
		47/ 3.0	8	8GB	Slinux 6.9	
		2670v3/ 2.30	2 x 24	32GB	Ubuntu 16.04	

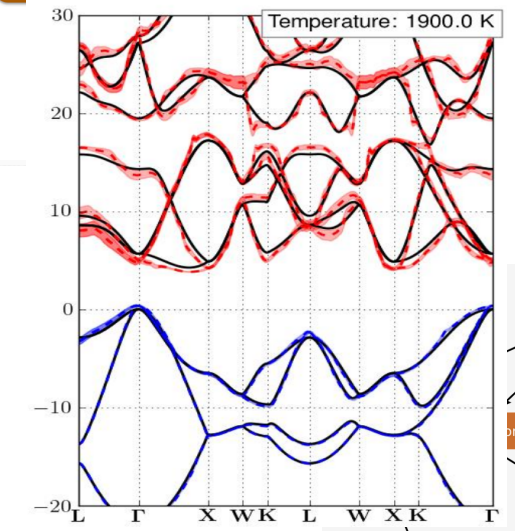
- 20 year old project, thousands users
- active community of developers

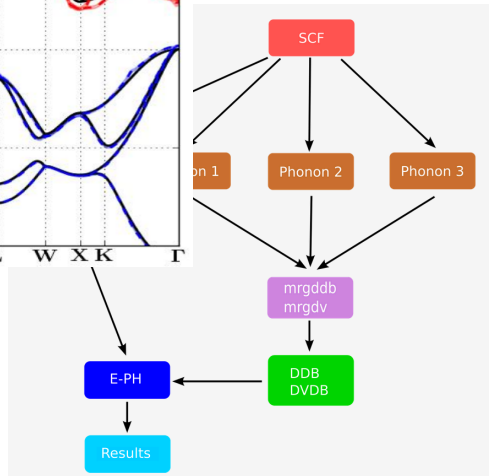
- large set of capabilities
- intensive quality control
- free licence

- basic input variables were detailed
- pointer to advanced documentation
- powerful ABINIT+Python approach

- capability to compute T-dep of electronic structure
- >1000 phonon band structure and derivative databases available from Materials Project







ABINIT Hands-on

(1) In SCRATCH, create your own 'home' abinit directory (referred to as ~abinit)

cd /scratch/$USER ; mkdir abinit ; cd abinit

(2) The pristine abinit package is available at /home/nfs3/smr3191/Abinit/github_abinit

You need the input files as well as the scripts :

cp –r /home/nfs3/smr3191/Abinit/github_abinit/scripts . (do not forget the final dot)

mkdir tests ; cd tests ; mkdir tutorespfn ; cd tutorespfn

cp –r /home/nfs3/smr3191/Abinit/github_abinit/tests/tutorespfn . (do not forget the final dot)

(3) Abinit executables (abinit, mrgddb …) are available at /home/nfs3/smr3191/Abinit/local/bin

but will be available directly in your PATH if you execute :

cp –r /home/nfs3/smr3191/Abinit/local/bin /scratch/$USER/abinit_bins

export PATH=/scratch/$USER/abinit_bins:$PATH

(4) Familiarize with the ABINIT doc

connect to docs.abinit.org

click in the 'i' icon on the right-hand side of the banner (a 30-second guided tour)

click on the 'lessons' keyword in the banner, then spend one-two minutes there

Now, you can follow the 'conventional' track, or the 'abitutorial' track

Conventional track

from 'lessons', click on the 'DFPT1' keyword inside the schema

follow the DFPT1(=RF1) tutorial … i.e. go back to linux and alternate with Web window

cd tutorespfn/Input

mkdir Work_rf1 ; cd Work_rf1 (as suggested in section 1 of DFPT1 lesson)

(do not follow all links, because you will not have time today !)

then continue with tutorial DFPT2 (=RF2), and finally the tutorial TDepES (in EPH section)

ENJOY !

ABINIT Hands-on

Abitutorial track

(1) Connect to https://github.com/abinit

click on the abitutorials rectangle, go to the 'ABOUT' section, spend 5…10 mins there :

- examine 'AbiPy plot gallery' and find e.g. the Eliashberg function and have a look
- examine 'AbiPy flow gallery' and find e.g. the 'flow for phonons with DFPT'
- examine 'How to use the tutorials'

(2) If you are working with your own PC, then either :

- follow the tutorial using the static HTML version (find the link in 'How to use the tutorials')

  => you will not be able to really run ABINIT, but you will have a nice guided tour

  Indeed : simply go to the 'Abinit+Abipy Lessons', and look at

  * Phonons, dielectric tensor and Born effective charges from DFPT
  * E-PH self-energy and T-dependent band structures

  ENJOY !

- install ABINIT and Abipy on your machine following 'If you opt for the last option …

  (30-60 minutes ?!) Hint : use miniconda, as advised on this abitutorial page !!

  => you are a real Pythonista !

  Then continue by executing a tutorial, as suggested in the abitutorials page…

  ENJOY !