



ABINIT School 2026
Learning electronic structure calculations using ABINIT
Feb. 2 - 6 2026 - Bruyères-le-Châtel, France



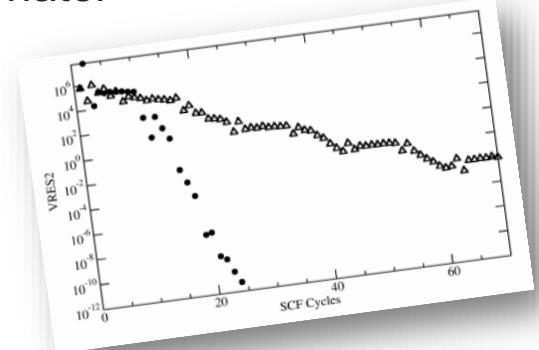
Tuning ABINIT

Convergence, precision, speed-up

Marc Torrent

CEA, DAM, DIF, F-91297 Arpajon cedex
Université Paris-Saclay, LMCE, F-91680 Bruyères-le-Châtel

accuracy 2
autoparal 1
natom 36
ntypat 3



Outline

DFT - Precision vs speed

Precision, accuracy, speed

What can be tuned in a plane-wave DFT code?

Automatic tuning

Speeding-up ABINIT

Basics: discretization, size

Advanced

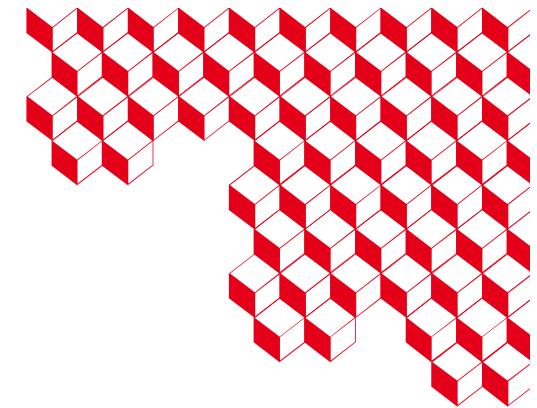
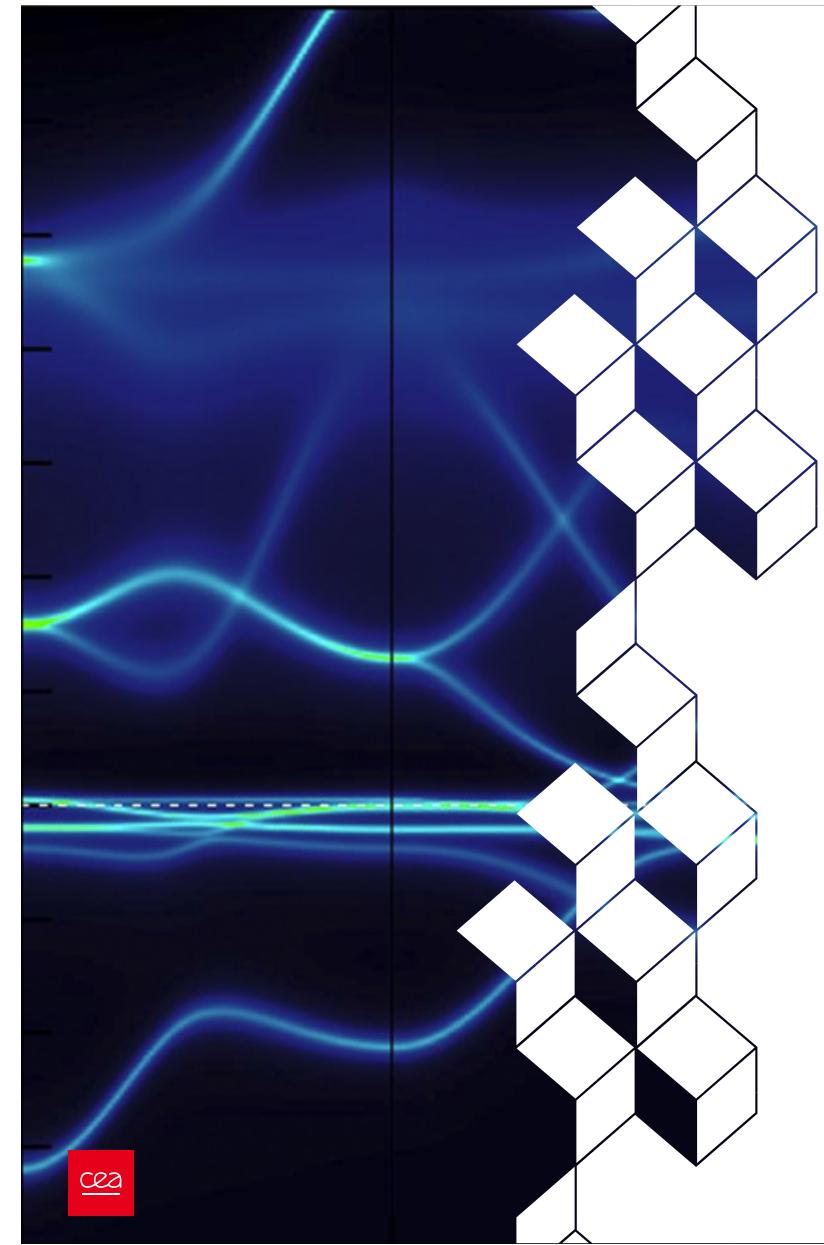
Helping ABINIT to converge

Mixing of the density

Optimization of the wave-functions

Brillouin zone sampling





Precision vs speed



Precision vs speed?

■ Properties needing precision

- Fine comparisons of energies
- Derivatives of the energy: forces, stresses, phonons, pressure...
- Structural relaxation (forces)
- Wave-functions in view of excited states/response function
- Magnetism
- Etc.

■ Properties needing speed

- Molecular Dynamics
- High-throughput computing (mass calculations)

By default, ABINIT
settings favor
precision





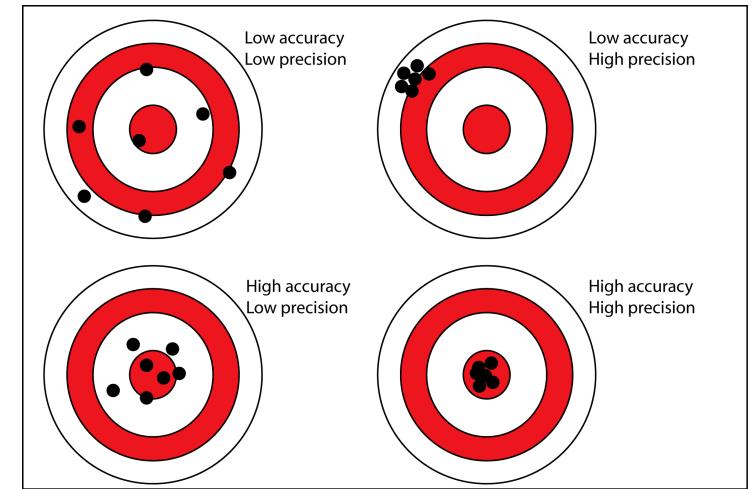
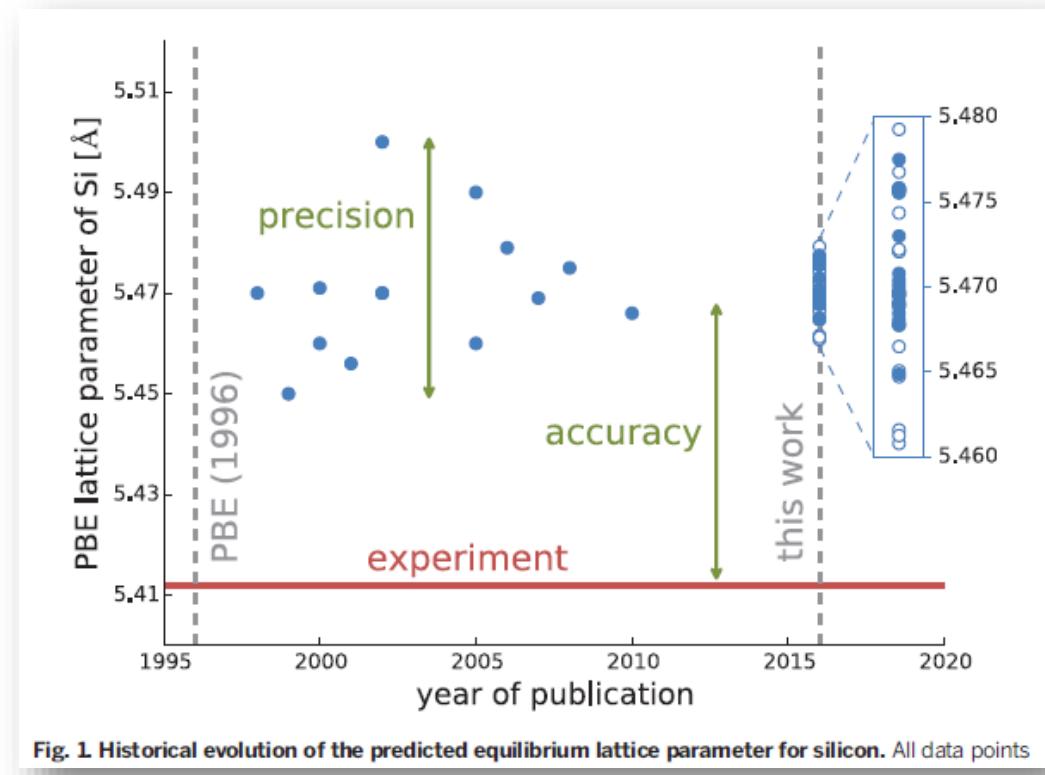
Precision vs accuracy

- By tuning the input parameters, you can increase precision...
 - Get more significant digits
 - Increase the size of the basis
 - Do more iterations
 - ...

- ... not accuracy
 - Calculations (PAW) are supposed to match all-electron calculations, not experiment
 - Precision can be improved by:
 - changing the pseudopotential (change frozen-core)
 - adding more physics in the theory (e.g. *more sophisticated exchange/correlation, exact exchange, ...*)
 - ...



Accuracy vs precision



Lejaeghere et al., Science 351 (2016)



Precision in DFT codes

Discretization

Electronic density formula

$$\rho(\vec{r}) = \sum_{\sigma} \sum_{\text{spins}}^n \int_{\text{Reciprocal space}} f_{nk} \left(\left| \sum_{\vec{g}} \left(C_{n,k}(\vec{g}) \cdot e^{i \cdot (\vec{k} + \vec{g}) \cdot \vec{r}} \right) \right|^2 \right) \cdot d\vec{k}$$

Discretization of reciprocal space
Number of k points
Electronic occupations f_{nk}

Discretization of the real space
Number of grid points (FFT grid)

Size of the plane wave basis
Determined by the cut-off energy

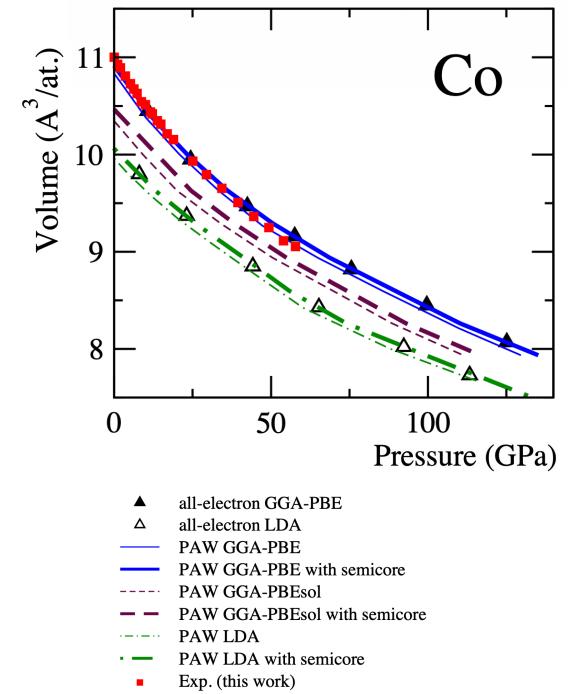
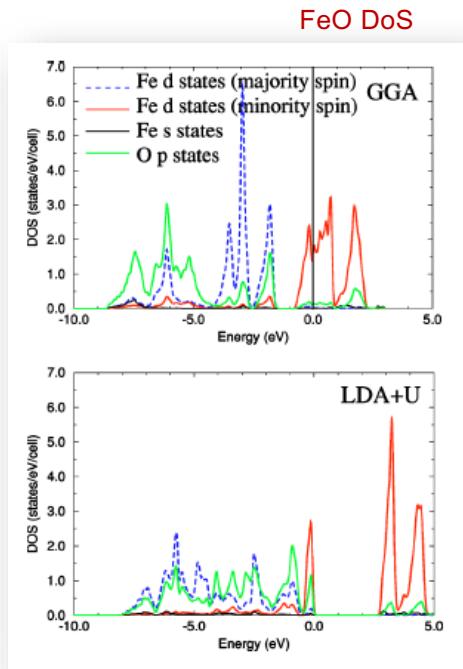
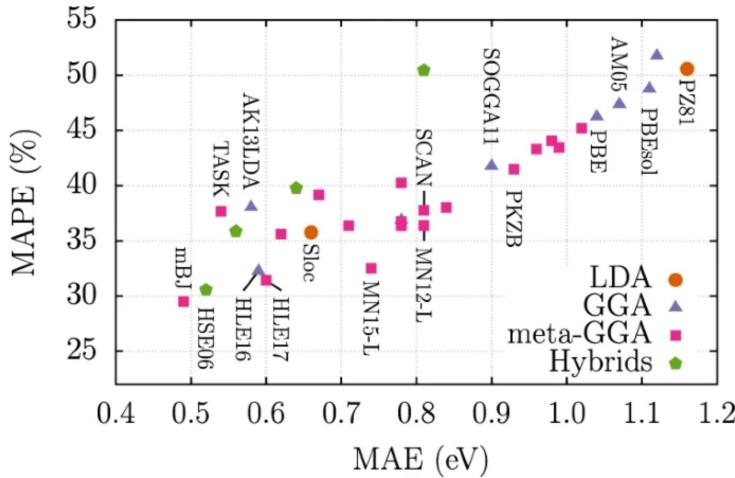
Discretization in numerical
schemes, in fitting procedures, etc.



Accuracy in DFT codes

Exchange-correlation, and co ...

Fig. 4: Overview of the MAE and MAPE of all studied functionals.



Borlido et al., *npj Comput Mater* **6**, 96 (2020)

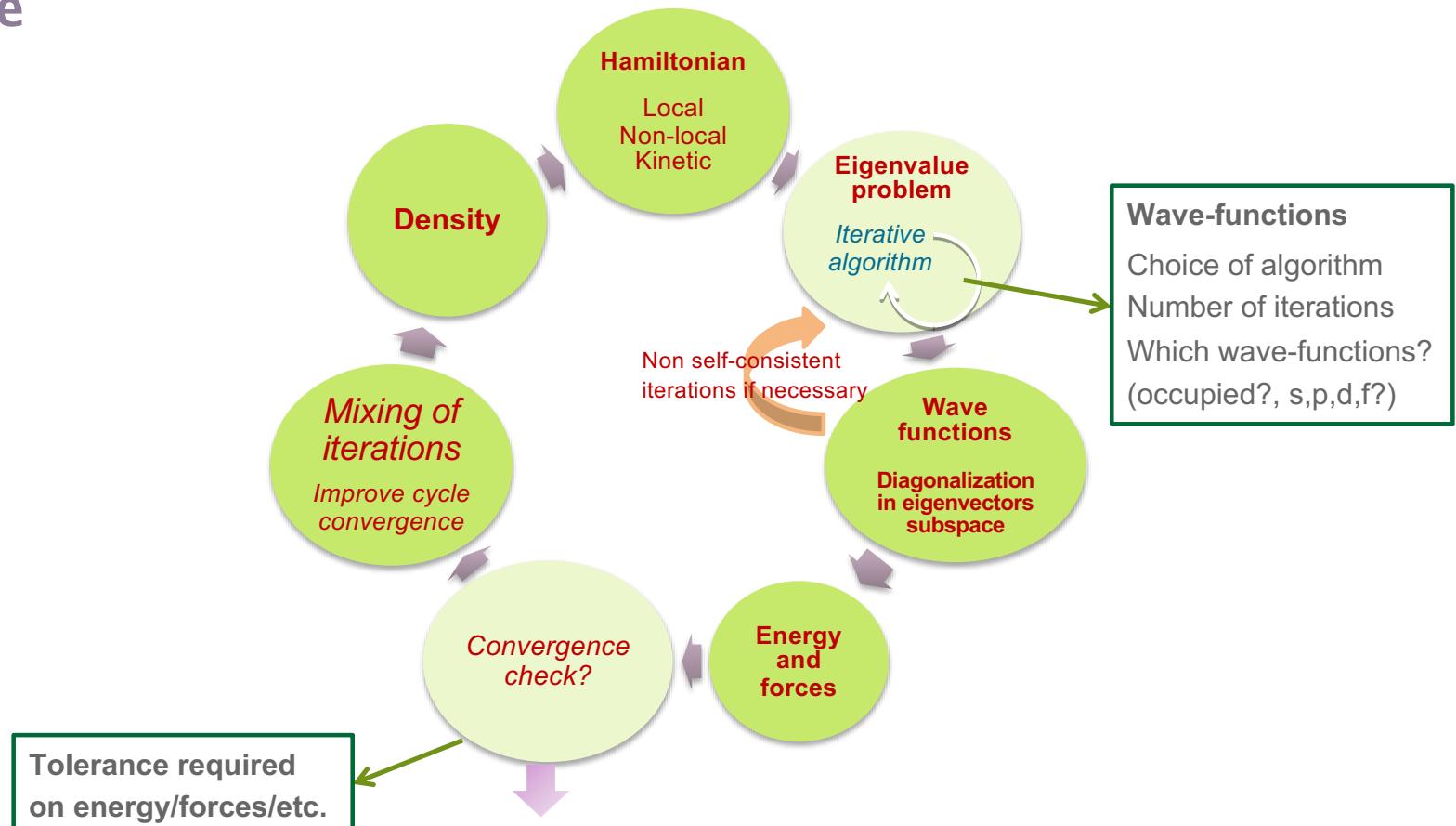
Cococcioni, de Gironcoli, PRB **71**, 035105 (2005)

Dewaele et al., PRB **78**, 104102 (2008)



Precision in DFT codes

Convergence

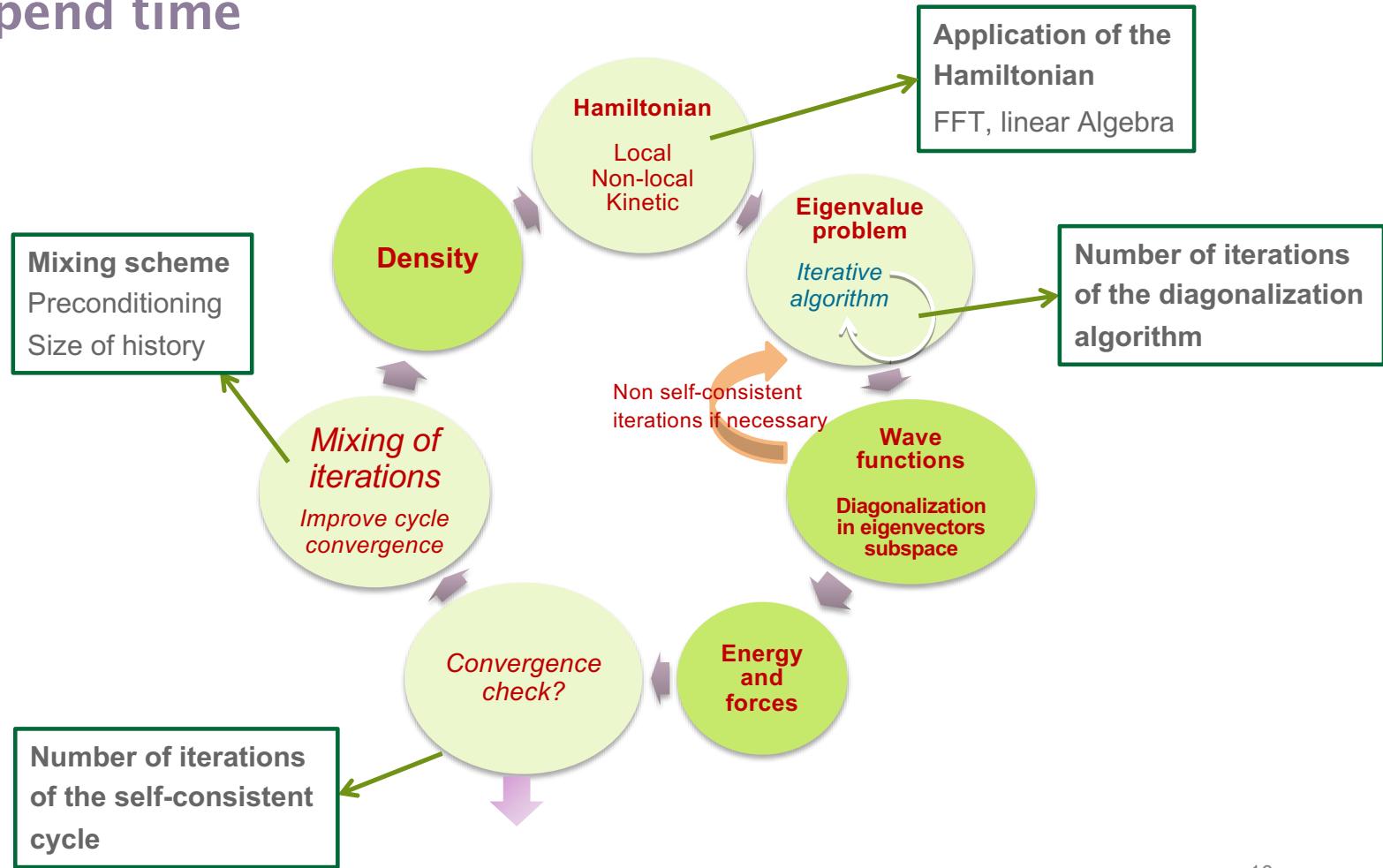


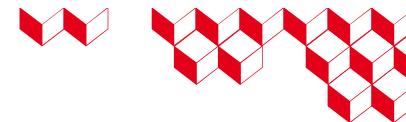


Plane-wave DFT

Where do we spend time

Time=
(number of iterations) x
(time spent in iterations)





Precision vs speed in DFT codes

What can we tune

■ Improve speed, decrease precision...

- Decrease the cost of the Hamiltonian application
 - Less plane-waves
 - Smaller FFTs
- Decrease the sampling of the reciprocal space
 - Less k-points
- Decrease the number of required significant digits
 - Increase the tolerance(s)

Warning!

Tolerance should always
be chosen according to the
property of interest

...or the contrary

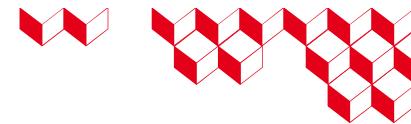


Precision vs speed in DFT codes

What can we tune

■ Improve speed... without decreasing precision

- Use parallelism
- Decrease the number of empty bands
- Improve the efficiency of the SCF cycle
 - Improve efficiency of the mixing scheme
 - Decrease the number of iterations
- Improve the efficiency of the iterative diagonalization
 - Fine tune the parameters of the algorithm



ABINIT tuning

Automatic or manual?

■ Automatic tuning - Pros

- It's convenient!
- It's a good starting point
- Manual tuning is a challenging task

■ Automatic tuning - Cons

- It is not optimal for all systems
- Some problem can be hidden



ABINIT tuning

accuracy input variable

accuracy

Mnemonics: ACCURACY
Mentioned in topic(s): [topic_Planewaves](#), [topic_SCFControl](#)
Variable type: integer
Dimensions: scalar
Default value: 0

[Test list \(click to open\). Moderately used, \[16/998\] in all abinit tests, \[3/117\] in abinit tutorials](#)

Allows to tune the accuracy of a calculation by setting automatically the variables according to the following table:

accuracy	1	2	3	5	6
----------	---	---	---	---	---

accuracy=4
Default ABINIT behavior

- **accuracy = 1 : precision is low**
Intended for use in Molecular Dynamics
- **accuracy = 6 : precision is high**
Designed to prepare response function calculations



ABINIT tuning

Choice of plane-wave cut-off energy



Used if accuracy is activated and ecut not in input file

Can be used as ecut estimate

```
<?xml version="1.0"?>
-<paw_dataset version="0.7">
<atom valence=<< 3.00" core="10.00" z=<< 13.00" symbol=<< Al"/>
<b><pw_ecut high="15.00" medium="12.00" low="10.00"/>
</b>
<xc_functional name="PBE" type="GGA"/>
<generator name="atompaw-4.0.0.12" type="scalar-relativistic"/>
```



ABINIT tuning

autoparal input variable

autoparal

Mnemonics: AUTOmatisation of the PARALLELism

Characteristics: DEVELOP

Mentioned in topic(s): topic_parallelism

Variable type: integer

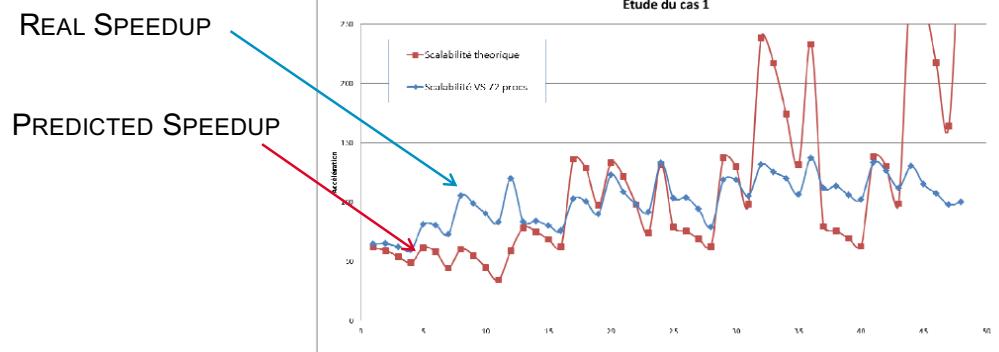
Dimensions: scalar

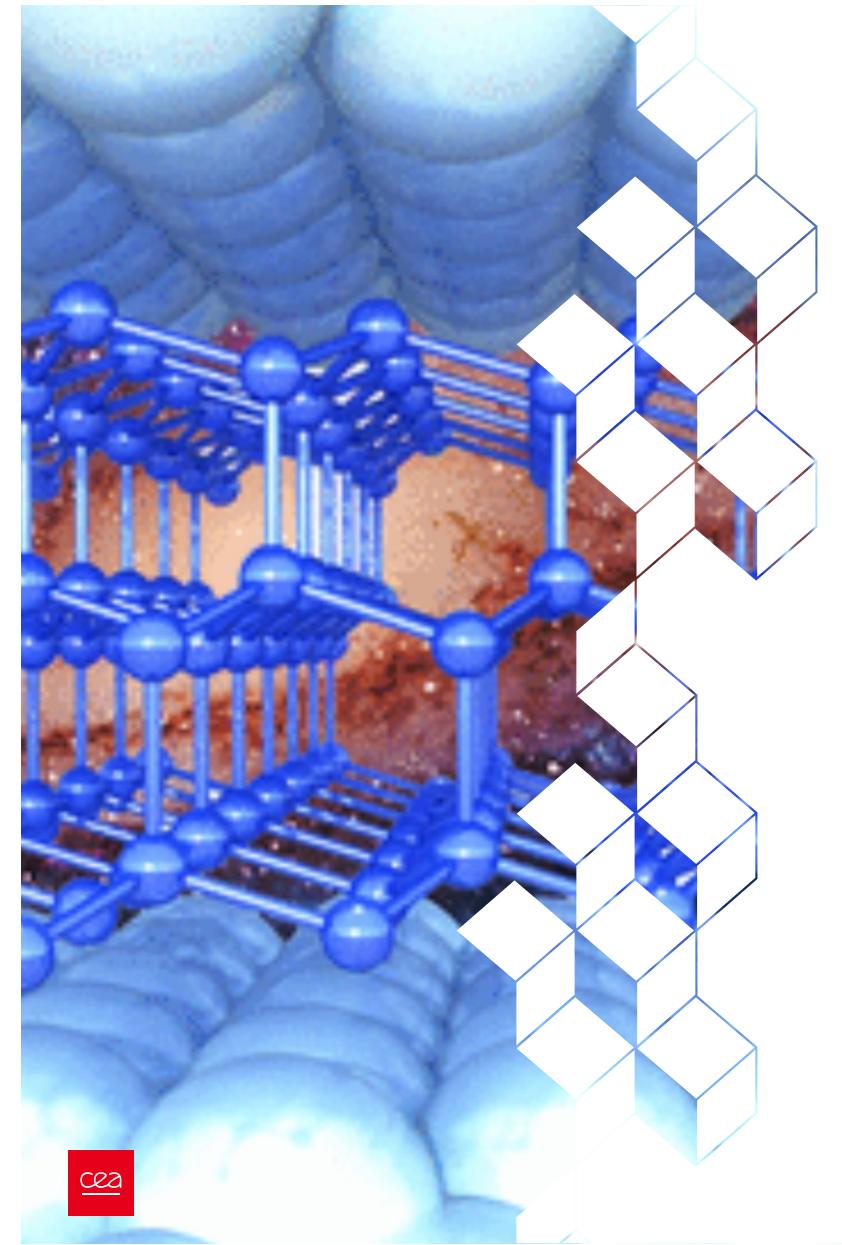
Default value: 0

Test list (click to open). Moderately used, [11/998] in all abinit tests, [1/117] in abinit tutorials

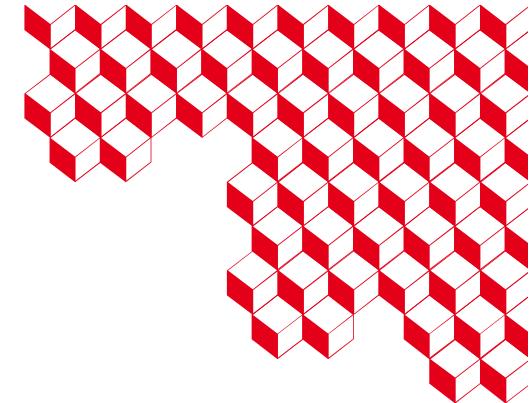
This input variable is used only when running ABINIT in parallel and for Ground-State calculations. It controls the automatic determination of parameters related to parallel work distribution (if not imposed in input file). Given a

- **autoparal = 1 : automatic parallelization is on**
ABINIT tries to determine the best distribution of processors on the different parallelization levels taking into account their respective efficient
- A simple heuristics is used





Speed-up ABINIT





Speeding-up ABINIT

Finding the best compromise...

... between

- the time required for one SCF iteration
- the number of iterations

Increasing the precision required for one iteration usually decreases the number of iterations
but takes longer per iteration!

See “*Helping ABINIT to converge*” section



Speeding-up ABINIT

Basics 1

■ Using parallelism

See lecture on parallelism – See `autoparal` input variable

- ABINIT takes fully advantage of the parallelism
- Can be activated even on scalar computers
Modern “processors” are multicore
- `openMP` (multi-threading) is MANDATORY with ABINIT 10, when available

■ Deactivating useless file access

- If you run only Ground-State calculations, wave-function file is not required
- Molecular Dynamics can be restarted without density/wave-functions, etc.
- Some files are used by specific post-processes (ex.: GSR file)

■ ABINIT input variables

- `prtwf/prtden`: printing of wave-function/density (can be deactivated)
- `write-files "none"` : no file written
- `prtgsr`: printing of GSR file (can be deactivated if `abipy` is not used)



Speeding-up ABINIT

Basics 2

■ Discretization

- Decrease the plane-wave basis size
- Decrease the sampling of the Brillouin zone
- Decrease the size of the real space/FFT grid

ecut: PW cut-off energy

ngkpt: k-points sampling

ngfft: size of FFT grid

■ Size of the system

- Decrease the number of electronic states
(do not compute useless empty bands)
- Use a suitable pseudopotential
(do not treat semi-core states when unnecessary)

nband: number of bands

Warnings!

- Always check convergence (don't use rule of thumb)
- Always check convergence with the studied property
- Some parameters depend on the chemical specie, some on the simulation cell



Speeding-up ABINIT

Convergence criteria

- Always confirm convergence relative to the target property!
 - Energy – Use tolerance on the total energy (**toldfe**)
Cohesive energy, barrier, phase stability, etc.
 - Forces (1) – Use tolerance on all forces (**toldff** / **tolrff**)
Molecular Dynamics
 - Forces (2) – Use tolerance on the maximal force precision (**tolmxrf**)
Structural relaxation
 - Wave-functions – Use tolerance on the WF residual (**tolwfr**)
Preparation of excited states runs or response function runs
 - Other – Use tolerance on the density/potential residual (**tolvrs**)
A generalist criterion (good compromise)

■ ABINIT input variables

- **toldfe**, **toldff**, **tolrff**, **tolmxrf**, **tolwfr**, **tolvrs**:

Tolerance criteria

Except for tolwfr, only one criterion can be used



Speeding-up ABINIT

Advanced 1

■ Decrease the number of SCF iterations – tuning the mixing

See “Helping ABINIT to converge” section

- Adapt the density/potential mixing to the simulation cell
Metal, insulator, cristal, inhomogeneous material, collinear magnetism, spin-orbit coupling, etc...
- The numerical behavior of the density during the SCF cycle is highly system dependent
- Number of iterations can be strongly decreased with a tuning of the mixing scheme

■ ABINIT input variables

- **diemix, diemac**: parameters for density residual preconditioning
- **iscf, npulayit**: parameters for density mixing



Speeding-up ABINIT

Advanced 2

■ Decreasing the size of the real space/FFT grid

$$\rho(\mathbf{r}) = \sum_i |\Psi_i(r)|^2$$

For an exact calculation, if the wave-function Ψ is expressed with all wave vectors up to $G_{\text{cut}}^2 \leq 2E_{\text{cut}}$, density should be expressed with wave-vectors up to $2G_{\text{cut}}$.

- N_{FFT} for density should be **2x** N_{FFT} for wave-function
- This ratio (`boxcutmin` input variable) can be decreased without a significant loss of precision.
- Always check!
- Not suitable if you prepare excited state or response function...
- **Highly efficient in terms of execution time!**

■ ABINIT input variables

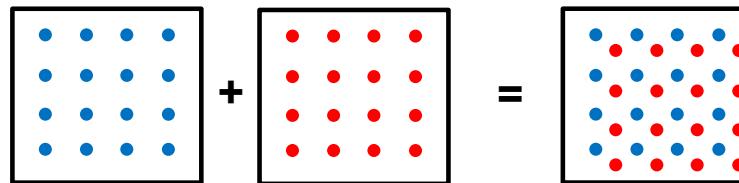
- **boxcutmin**: ratio between wave-function and density FFT grids



Speeding-up ABINIT

Advanced 3

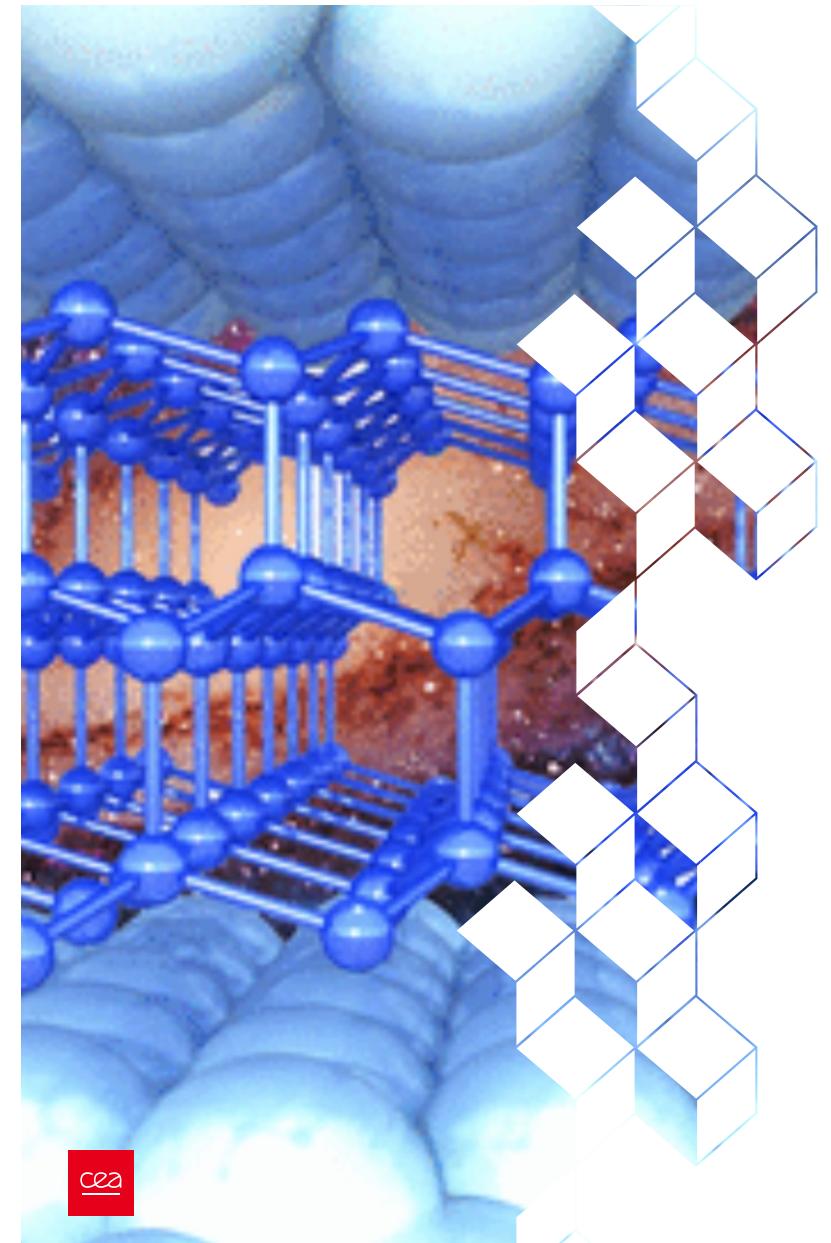
■ Use shifted k-points grids



- There are optimal shifts for each Bravais lattice
- Much more efficient and less consuming than increasing the k-point density of one grid

■ ABINIT input variables

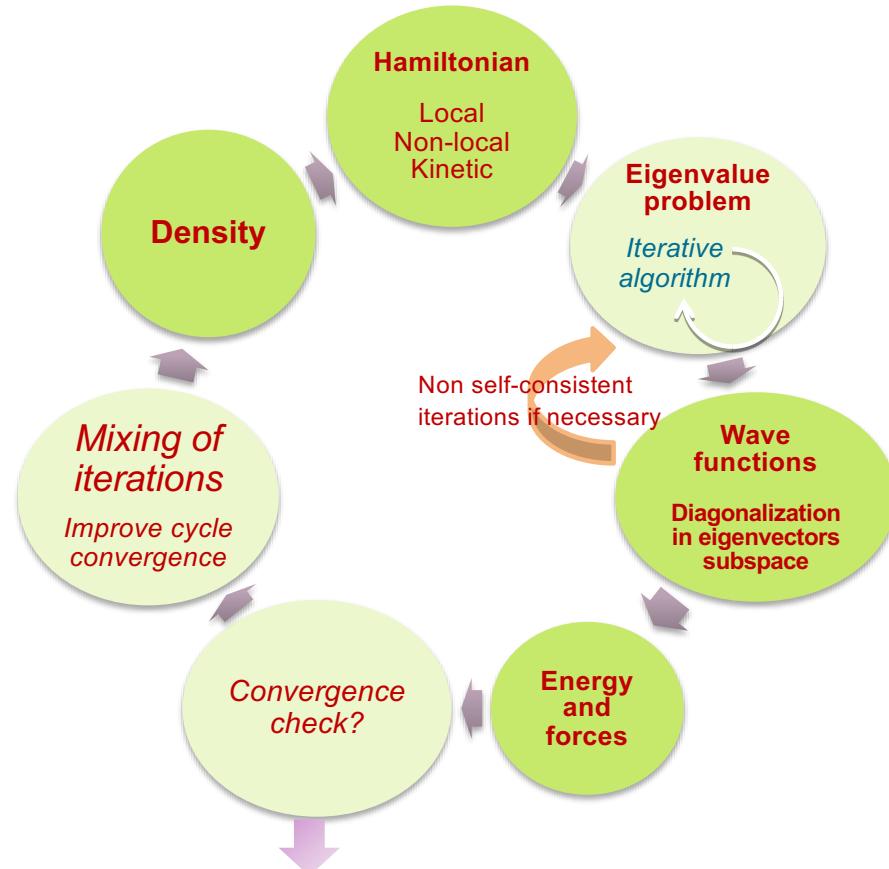
- **shiftk, nshiftk**: shifts to be applied to the k-points grid
Default : one shift ($\frac{1}{2}, \frac{1}{2}, \frac{1}{2}$)



**Helping ABINIT
to converge**



Self-consistent cycle

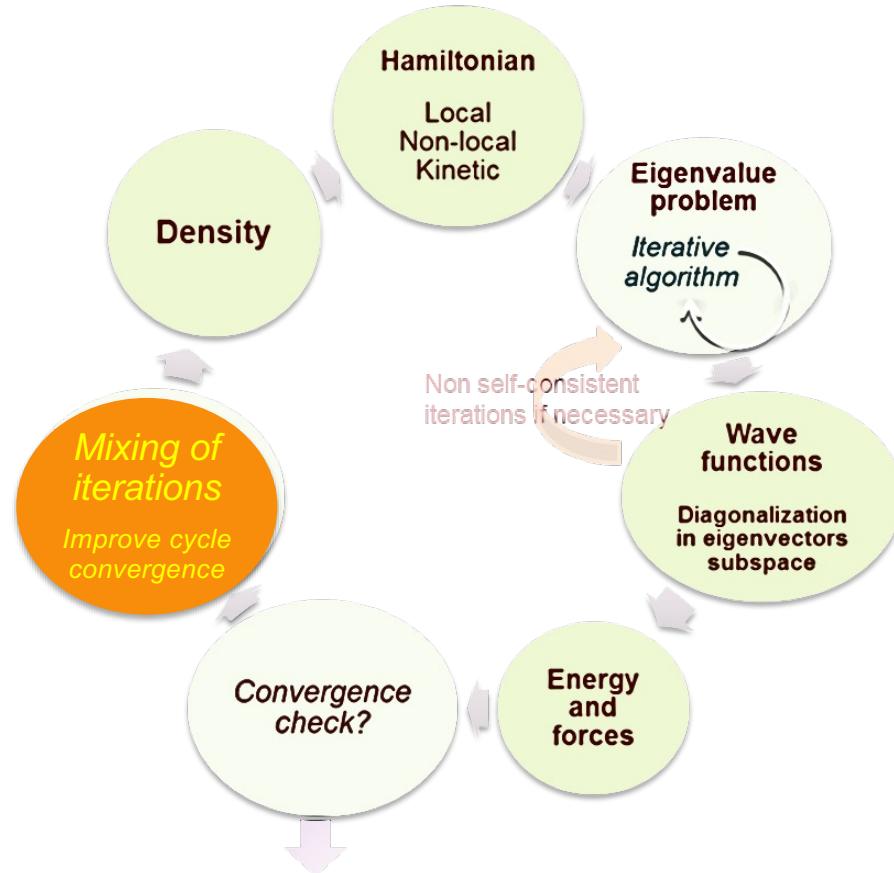




Self-consistent cycle

Mixing

Mixing the density
with the densities of
previous iterations
=> Damping of the
oscillations





Mixing/preconditionning the density

Mixing the new iteration and previous one(s)

Favor the iterations near the convergence

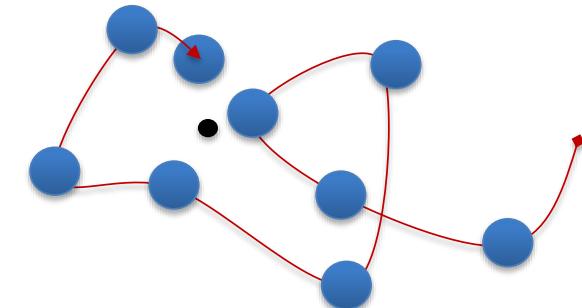
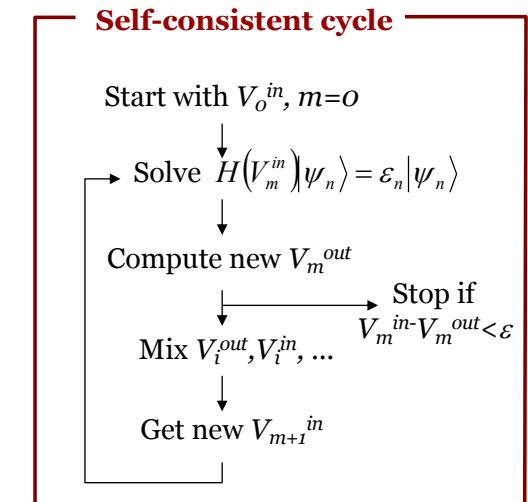
- Key quantity: the density residual:

$$R(V^{in}) = V^{out}(V^{in}) - V^{in}$$

- Goal for a mixing scheme:

Minimize the norm of the density residual $|R(V^{in})|^2$

- Independent of the mixing space (real or reciprocal)



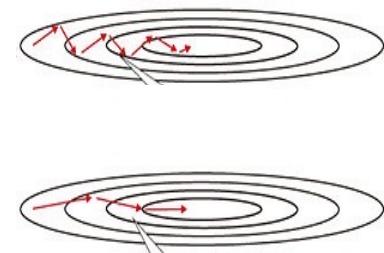


Mixing/preconditionning the density

Preconditionning the density residual

*Use the relevant components and attenuate
the oscillations of the residual*

- Apply a P operator on density residual: $P(R(V)) \leftarrow R(V)$
- The inverse of the dielectric function $\left(\frac{dV}{d\rho}\right)^{-1}$ is the ideal preconditioner \rightarrow But too heavy to compute
- We just need an approximation, not the exact preconditioner
Use of « model » preconditioner
- Do we precondition in real or reciprocal space?





Preconditionning the density

ABINIT input variables

■ Preconditioning

Use inverse of a model dielectric matrix

$$P(K) = \varepsilon^{-1}(K)$$

$$= \text{diemix} \cdot \left(\frac{1}{\text{diemac}} + \text{dielng}^2 K^2 \right) / (1 + \text{dielng}^2 K^2)$$

$$\varepsilon = \frac{d\rho}{dV}$$

■ ABINIT input variables

- **diemix**: decrease to help convergence (but will slow it)
- **diemac**: huge for metals, 5-10 for insulators
- **dielng**: not really important (fine tuning)



Mixing the density

ABINIT input variables

■ Mixing

Can mix density or potential – Default algorithm : Pulay algorithm

Can modify the size of the history (but a large history is memory-consuming)

$$\text{new } \rho_{i+1}(\mathbf{r}) = \rho_i^{IN}(\mathbf{r}) + \underset{j \leq i}{\text{mix}} [\rho^{OUT}(\mathbf{r}), \mathbf{P}\rho^{RES}(\mathbf{r})]$$

$$\text{new } V_{i+1}(\mathbf{r}) = V_i^{IN}(\mathbf{r}) + \underset{j \leq i}{\text{mix}} [V^{OUT}(\mathbf{r}), \mathbf{P}V^{RES}(\mathbf{r})]$$

■ ABINIT input variables

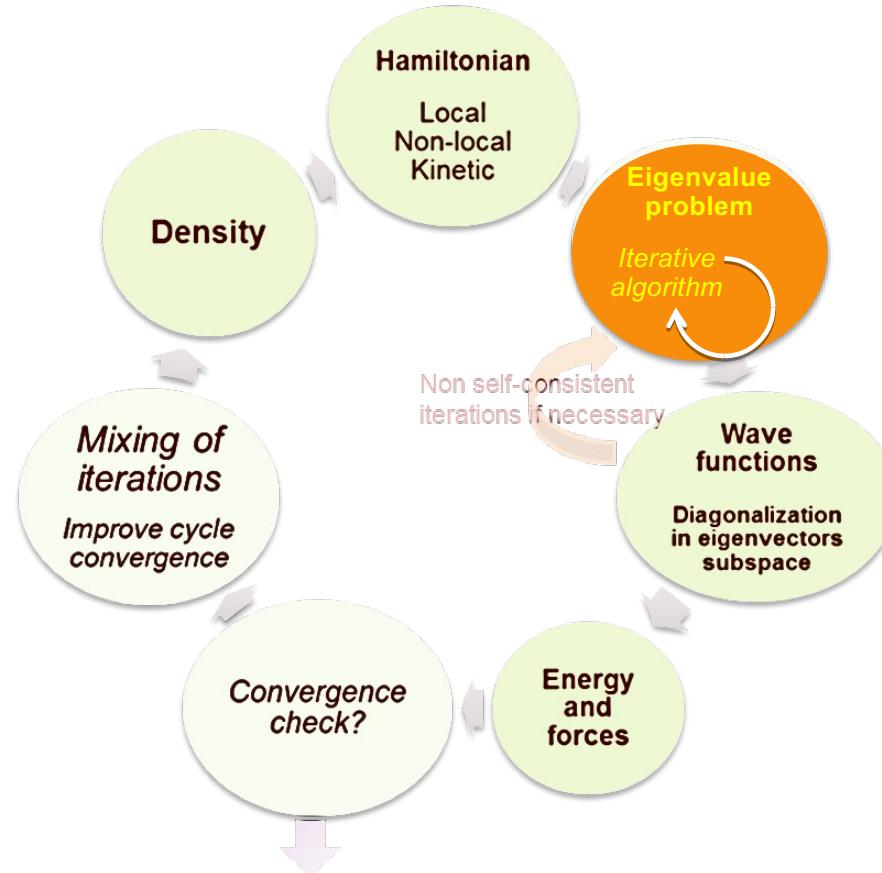
- **iscf**: 7 → mix the potential, 17 → mix the density
Can change the behavior of the convergence
- **npulayit**: size of history of Pulay algorithm (default=7)
Increasing the size can be very efficient (→ 30)



Optimization of the wave-functions

Most consuming part

- Better converge the wave-functions
 - => Can reduce the Number of iterations
 - => Each iteration takes more time





Optimization of the wave-functions

Algorithm and parameters

- Convergence depends on:
 - System of interest
 - Iterative diagonalization(minimization) algorithm

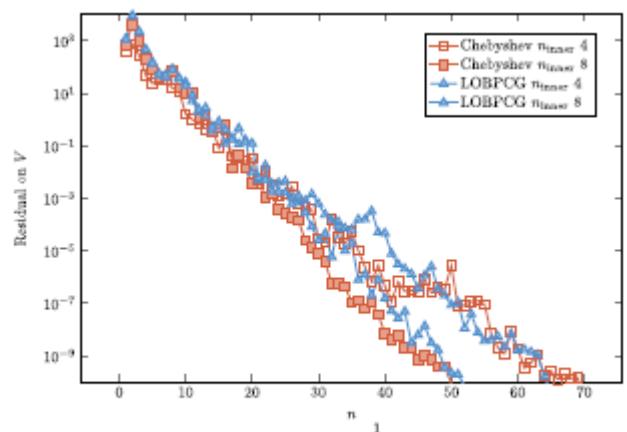
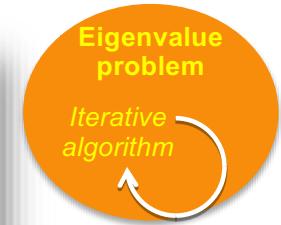


Fig. 6. Self-consistent convergence. The blocksize for LOBPCG was 128.

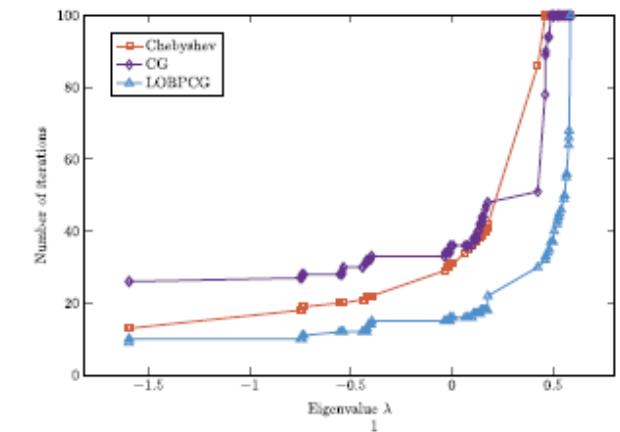


Fig. 4. Number of iterations to obtain a precision of 10^{-10} , BaTiO₃, 100 bands.



Optimization of the wave-functions

Algorithm and parameters

■ Choice of algorithm

- **Conjugate Gradient**

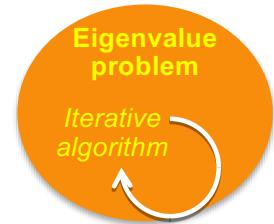
Default when no parallelization or k-points parallelization

- **Block conjugate gradient** (LOBPCG)

Default when Band-FFT parallelization

- **Chebyshev Filtering**

For a larger number of processors, for GPUs



■ ABINIT input variables

- **wfoptalg:** **0** → conjugate gradient
114 → block conjugate gradient
111 → Chebyshev filtering

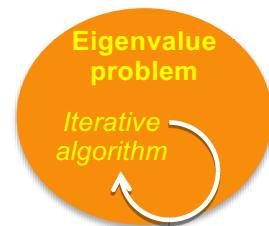


Optimization of the wave-functions

Algorithm and parameters

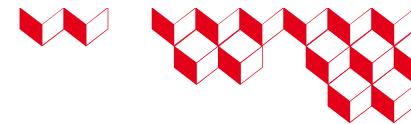
■ Algorithm parameters

- Required tolerance on wave-functions
- Number of max. iterations
- Block Conjugate Gradient: size of the blocks
 - One block (size N_{band}) converges better than N_{band} blocks (size 1)
 - But can takes longer



■ ABINIT input variables

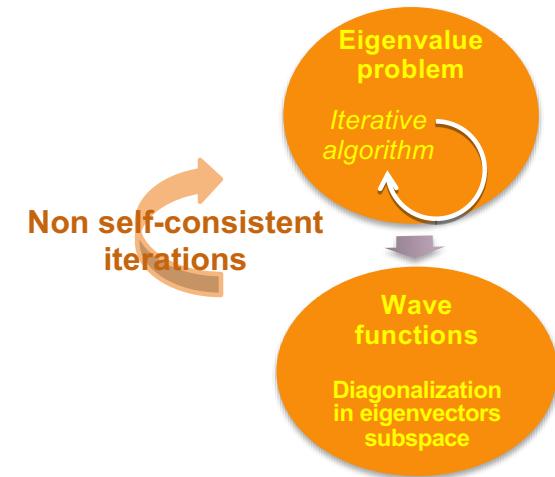
- **tolwfr**: Tolerance of Wave-Function residual
Decreasing it improve convergence but takes longer.
- **nline**: number of iterations of the diagonalization algorithm
Increasing it improves convergence but takes longer
- **nblock_lobpcg**: Number of BLOCKs for LOBPCG
Should be small on supercomputers



Optimization of the wave-functions

Non self-consistent iterations

- Last chance if still no convergence...
 - Run the diagonalization algorithm several times, resetting it
 - **Increase the number of Non-Self-Consistent Iterations**



- ABINIT input variables
 - **nnsclo**: Number of Non Self-COnsistent iterations
Default: 2 iterations for steps 1 and 2, then 1 iteration

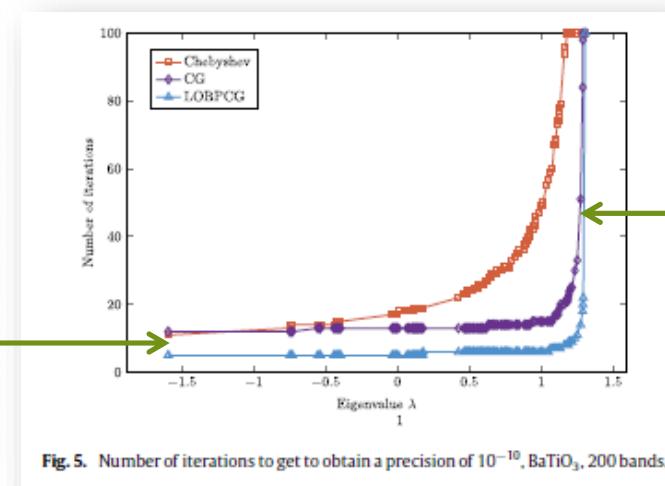


Optimization of the wave-functions

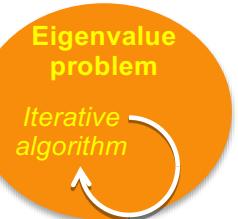
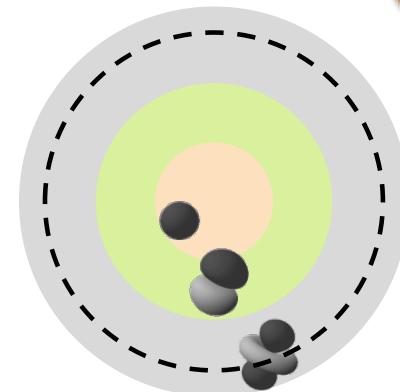
Empty bands (metals)

- Empty bands do not converge as fast as occupied ones
 - Depends on algorithm (CG, LOBPCG...)
 - Last bands can really not converge
 - Worst case: optimize partially an electron shell

Occupied bands
Few iterations
to converge



Empty bands
Many iterations
to converge



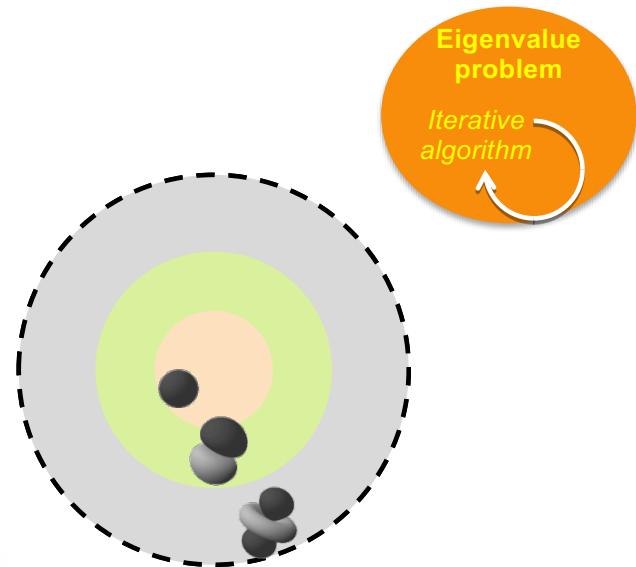


Optimization of the wave-functions

Empty bands (metals)

- Solution :
 - Change N_{band} in order to include all the states of the last shell

- ABINIT input variables
 - **nband**: Number of BANDs to optimize
Warning (metals): check that last states are really empty!



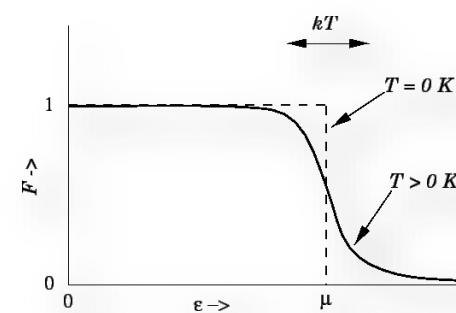
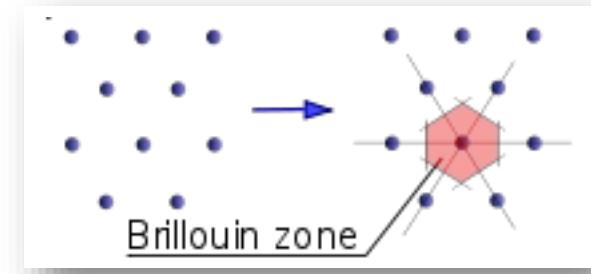


Metals / Magnetism

Sampling of Brillouin zone

- Impact of sampling of Brillouin Zone:
 - Many properties are obtained from an integral over the Brillouin Zone
 - Magnetic moment is numerically sensitive

- Smearing method:
 - In metals, integral over Brillouin Zone are discontinuous at Fermi level
 - Replace step function by a smoother function:
→ use a artificial electronic temperature
 - Modification of the electronic occupations

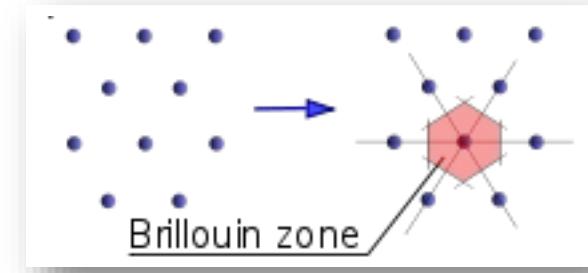




Metals / Magnetism

Sampling of Brillouin zone

- Impact of sampling of Brillouin Zone:
 - Many properties are obtained from an integral over the Brillouin Zone
 - Magnetic moment is numerically sensitive



- Solution:
 - **Find the right balance between smearing and sampling density**

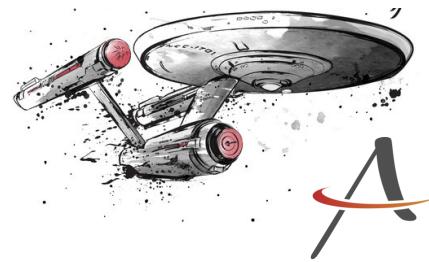
- ABINIT input variables
 - **nkpt/ngkpt/etc.**: number of k-points
 - **occopt**: smearing scheme for occupations ($3 \leq \text{occopt} \leq 7$)
 - **tsmear**: smearing temperature



Conclusion

Tuning ABINIT – Key points

- ABINIT default settings favor precision
- To run ABINIT faster, you can use the automatic method or the manual one
- To help ABINIT to converge it is necessary to have a minimal knowledge of the internal algorithms
 - Convergence cycles
 - Mixing algorithms
 - Discretization



In the 21st century
they used **ABINIT** in
the **PAW** approach!

