FROM RESEARCH TO INDUSTRY

cea

ABINIT School 2019

A newcomer-oriented school to ab initio nanoscience simulations

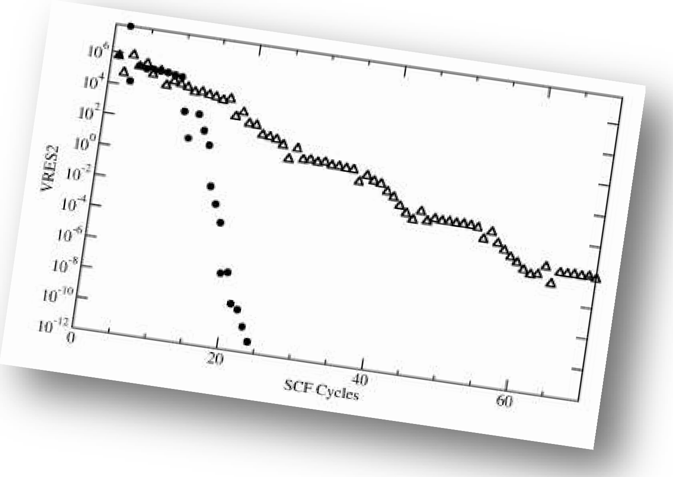
January 21-25, 2019 - Bruyères-le-Châtel, France

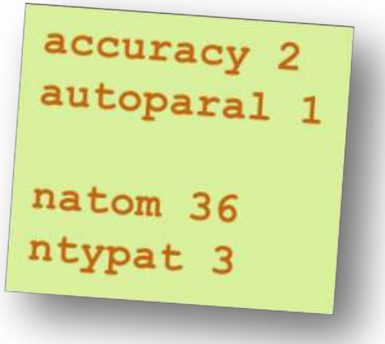
TUNING ABINIT

SPEED-UP PRECISION CONVERGENCE

Marc Torrent

CEA, DAM, DIF, France





```
accuracy 2
autoparal 1

natom 36
ntypat 3
```

www.cea.fr

OUTLINE

ABINIT: 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-funtctions

Brillouin zone sampling

ABINIT
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 <u>precision</u>

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
  - Accuracy 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

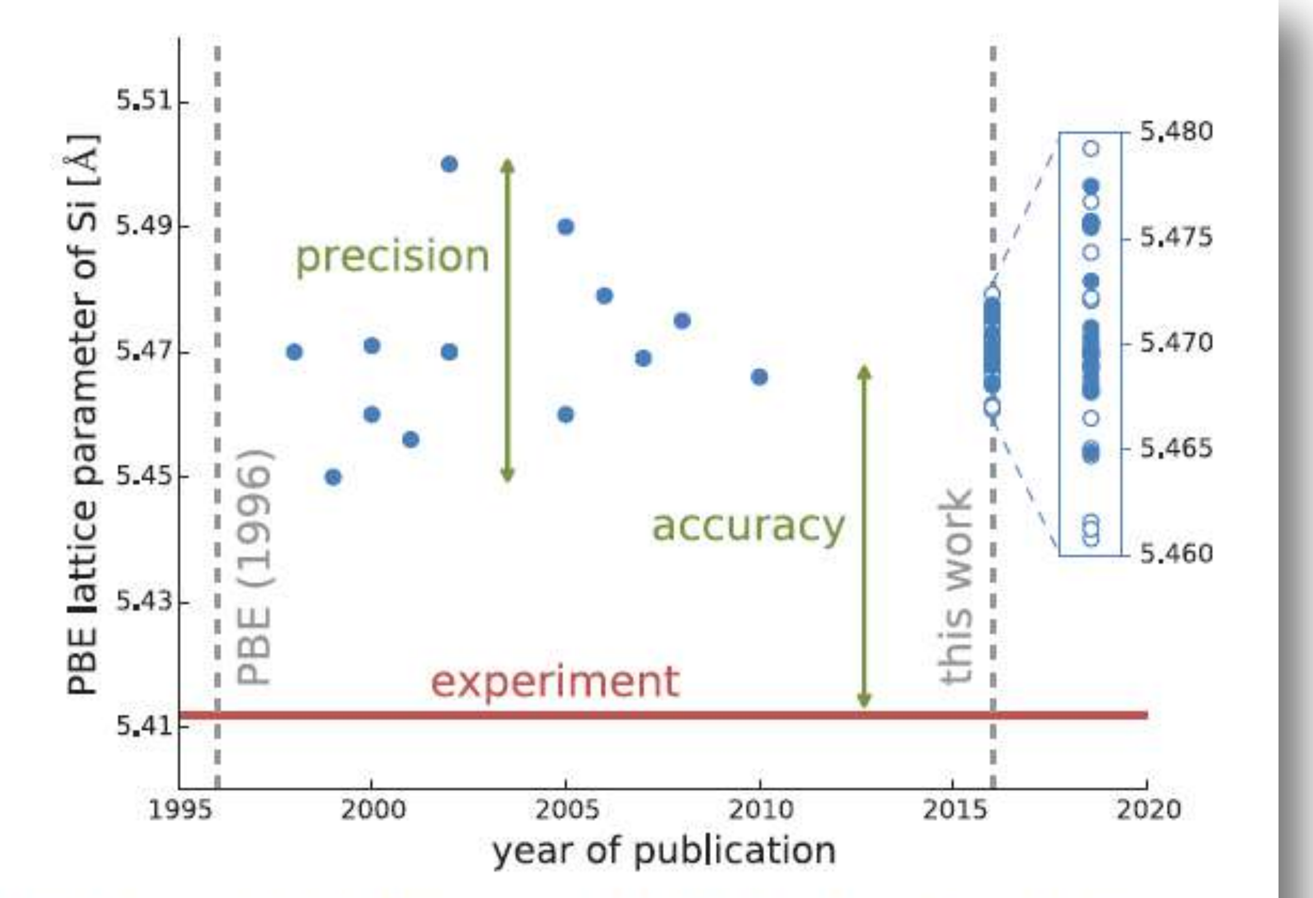


Fig. 1 Historical evolution of the predicted equilibrium lattice parameter for silicon.

Lejaeghere et al., Science 351 (2016)

PRECISION IN DFT CODES - DISCRETISATION

Electronic density formula

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

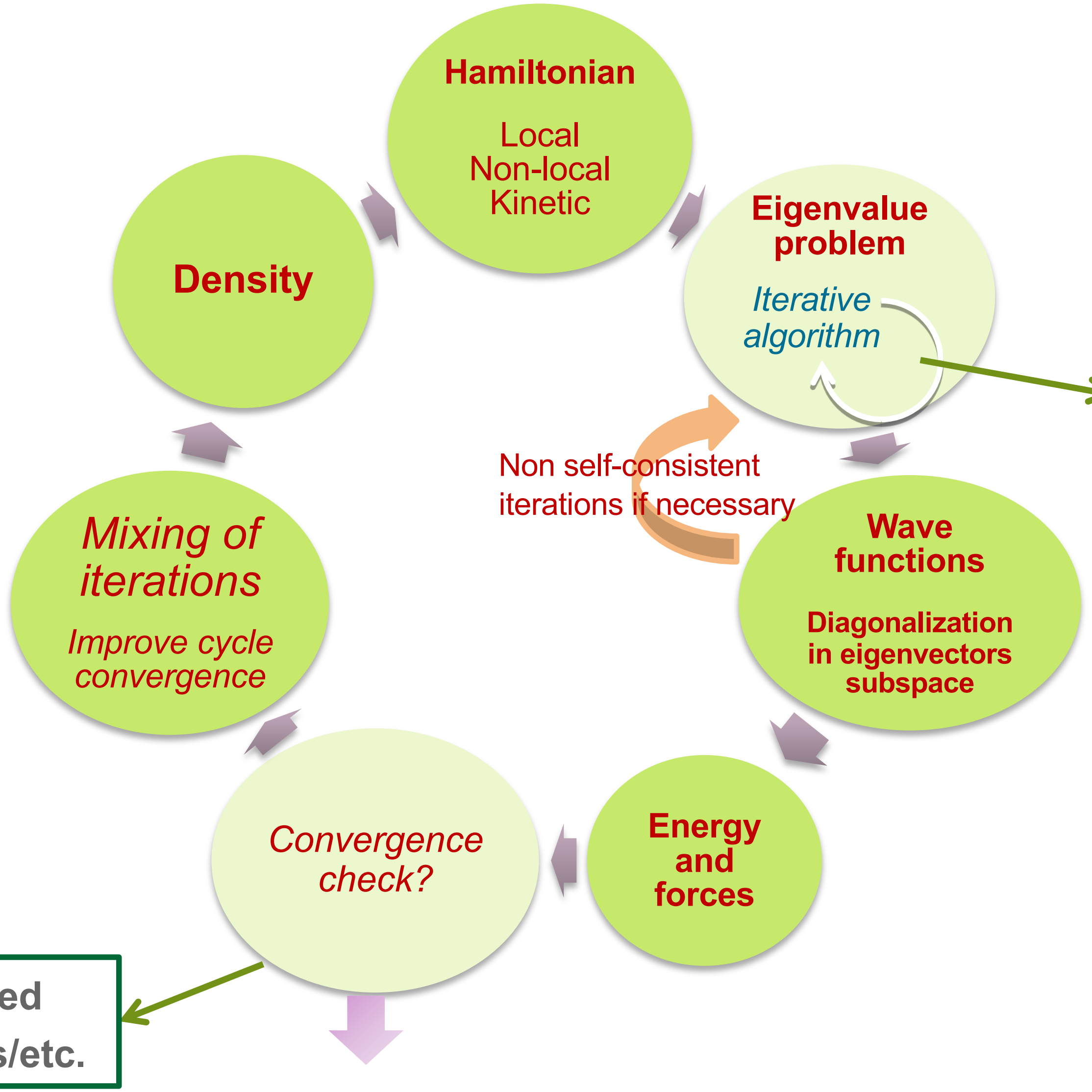
Discretization of reciprocal space
Number of k points
Electronic occupations f_{nk}

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.

PRECISION IN DFT CODES - CONVERGENCE



Wave-functions

Choice of algorithm

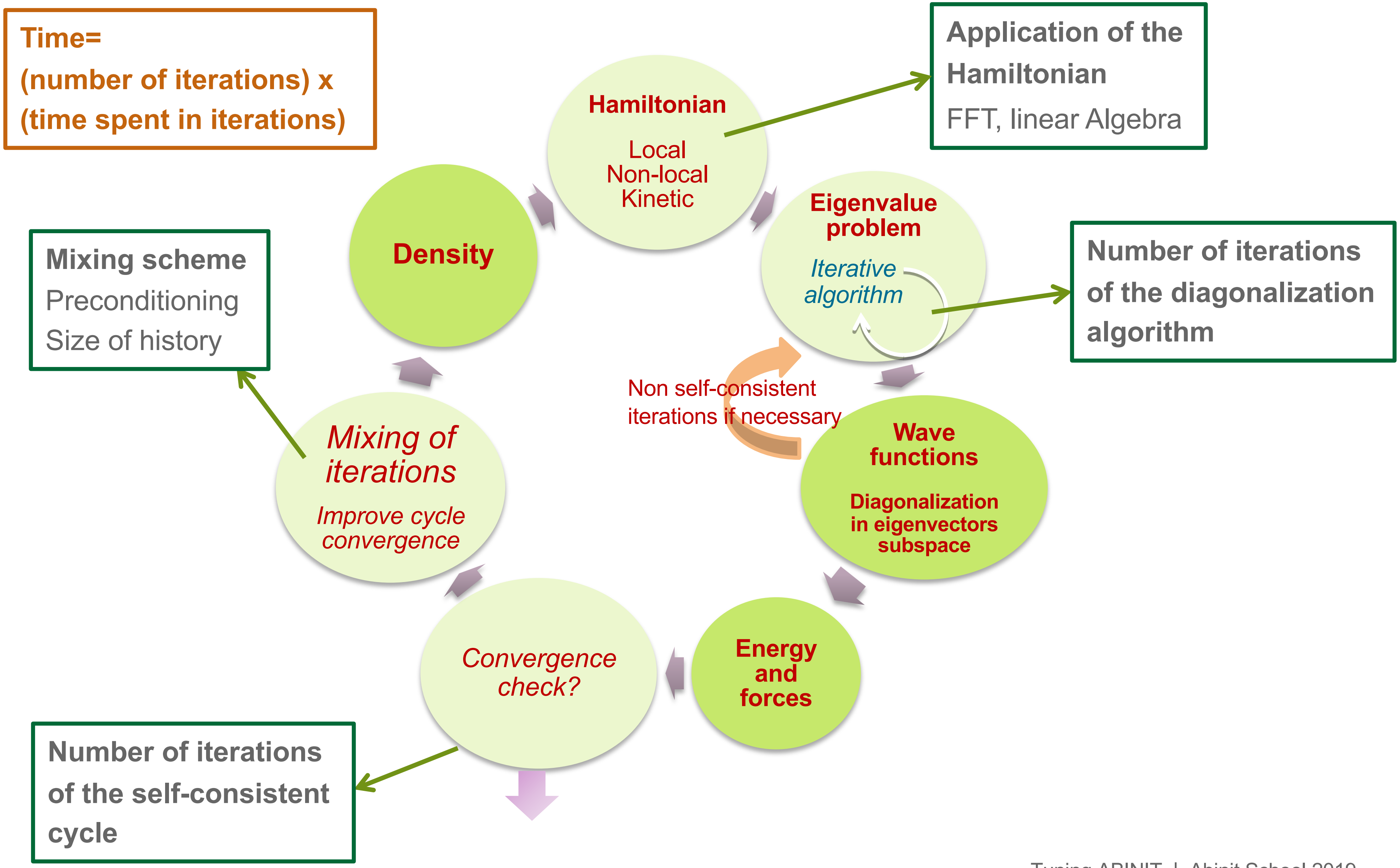
Number of iterations

Which wave-functions?

(occupied?, s,p,d,f?)

Tolerance required on energy/forces/etc.

PLANE WAVE DFT - WHERE DO WE SPEND TIME?



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

Hamiltonian
Local
Non-local
Kinetic

Application of the Hamiltonian
FFT, linear Algebra

Eigenvalue problem
Iterative algorithm

Number of iterations of the diagonalization algorithm

Non self-consistent iterations if necessary

Wave functions
Diagonalization in eigenvectors subspace

Energy and forces

Convergence check?

Number of iterations of the self-consistent cycle

Mixing of iterations
Improve cycle convergence

Mixing scheme
Preconditioning
Size of history

Density

PRECISION VS SPEED - 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)

…or the contrary

Warning!
Tolerance should always be chosen according to the property of interest

PRECISION VS SPEED - 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?

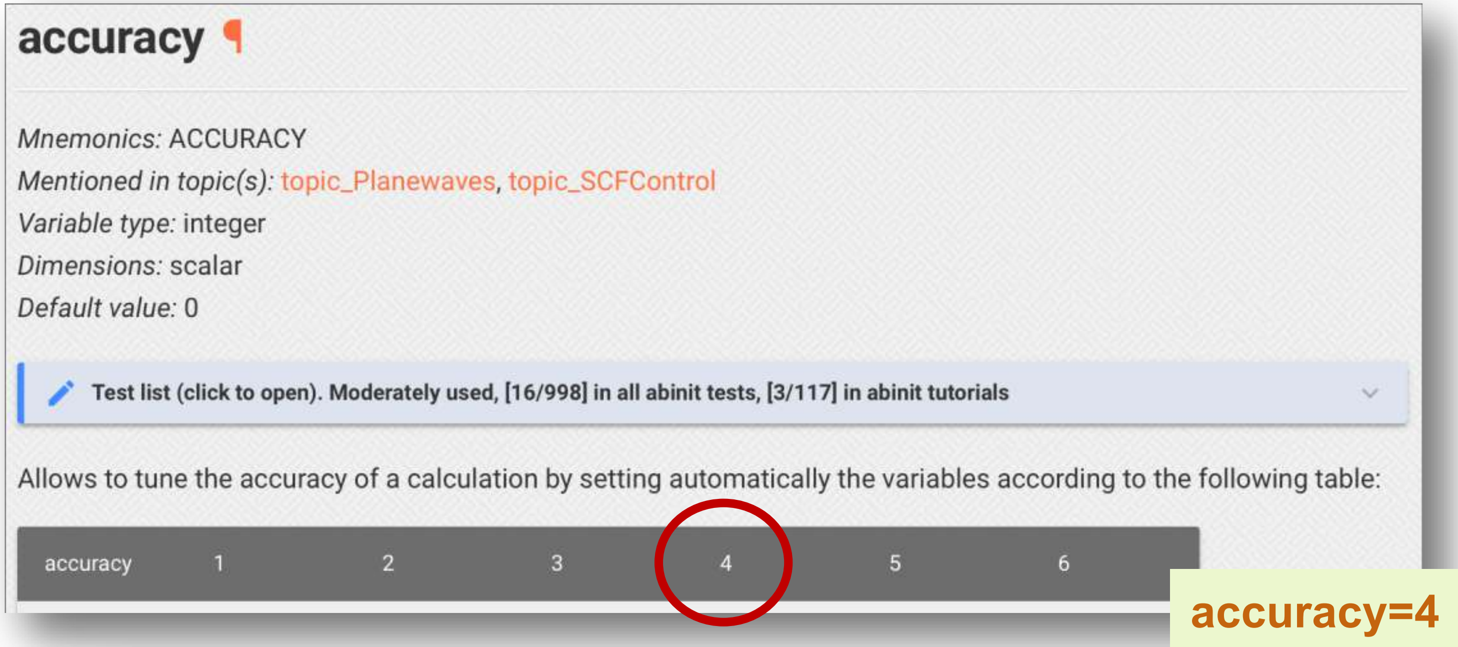
Automatic tuning - Pros

- 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 – « 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	4	5	6

accuracy=4
Default ABINIT behavior

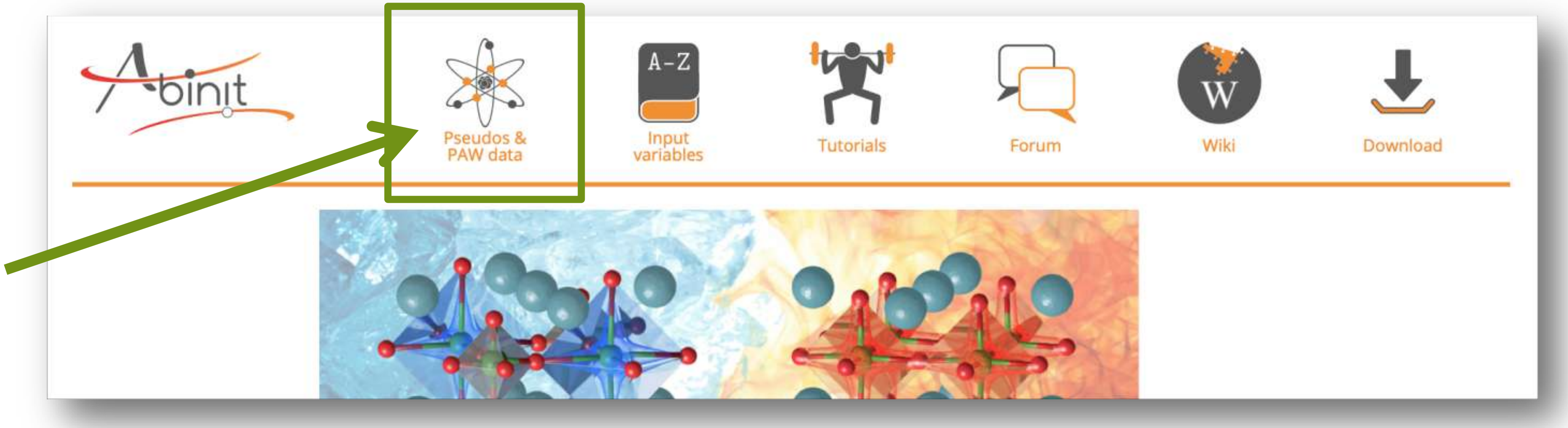
- **accuracy = 1 : precision is low**
  Intended for use in Molecular Dynamics
- **accuracy = 6 : precision is high**
  Designed to prepare response function calculations
- **accuracy = 0 : precision is medium**
  “Automatic *ecut*” not activated

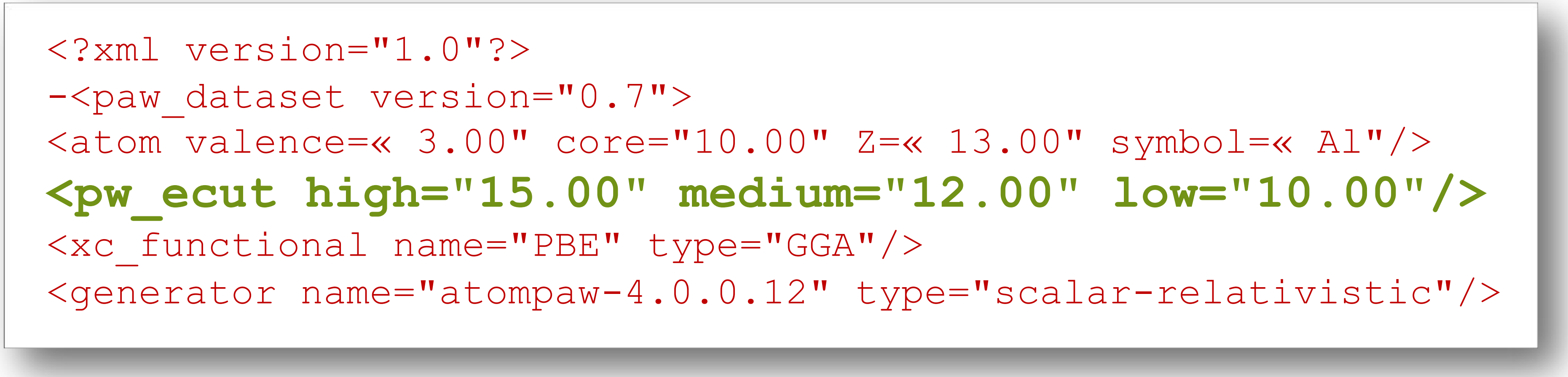
ABINIT – « ACCURACY » INPUT VARIABLE

Default values for plane wave cut-off enregy

- Default values for `ecut` variable **can be** read from the PAW pseudo potential file
- Only if `accuracy` and PAW are activated
- Only with the "JTH" atomic data table (www.abinit.org)
- 3 default precision values: *low, medium, high*
  Choice made according to `accuracy` input variable
- Values based on the comparison of the pure material Equation of States with all-electron results
- No guarantee for alloys, oxides…
- To activate it: **don't put `ecut` in the input file**

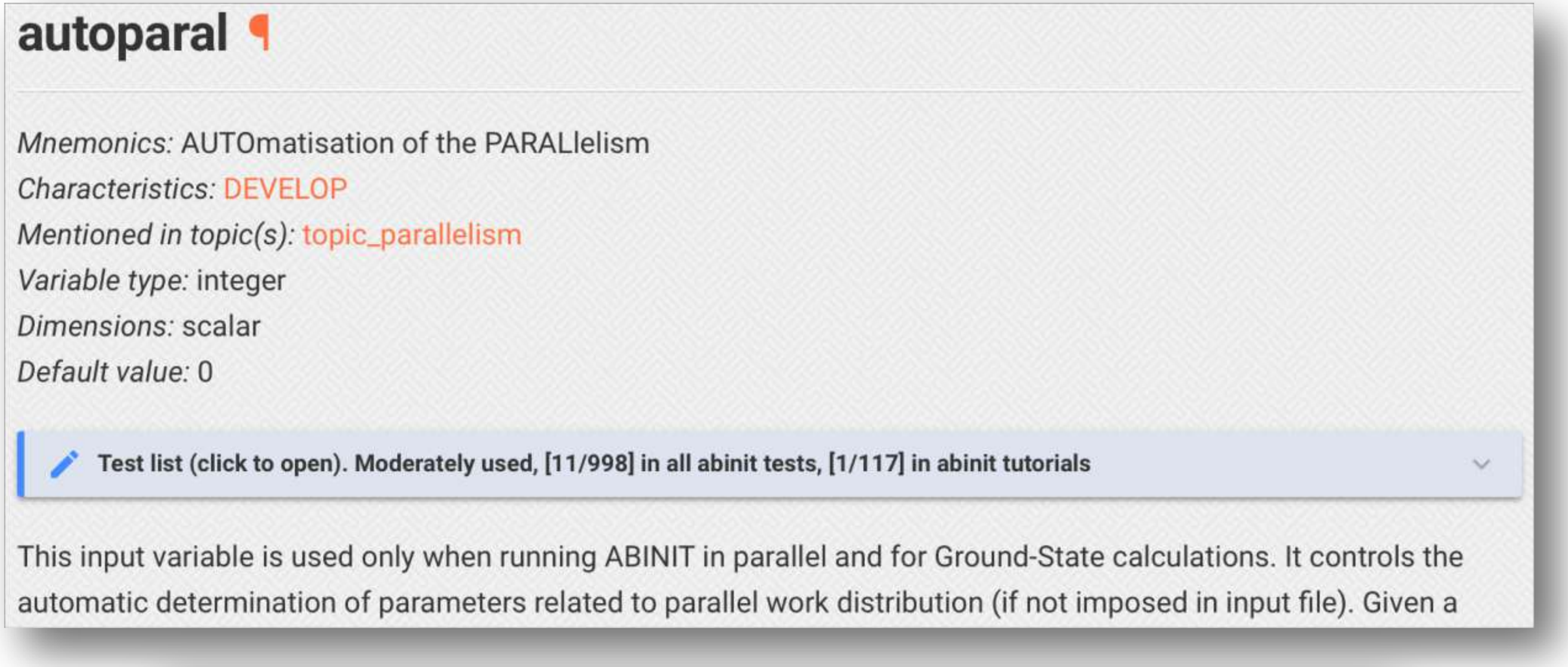
ABINIT – « ACCURACY » INPUT VARIABLE





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

ABINIT – « 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

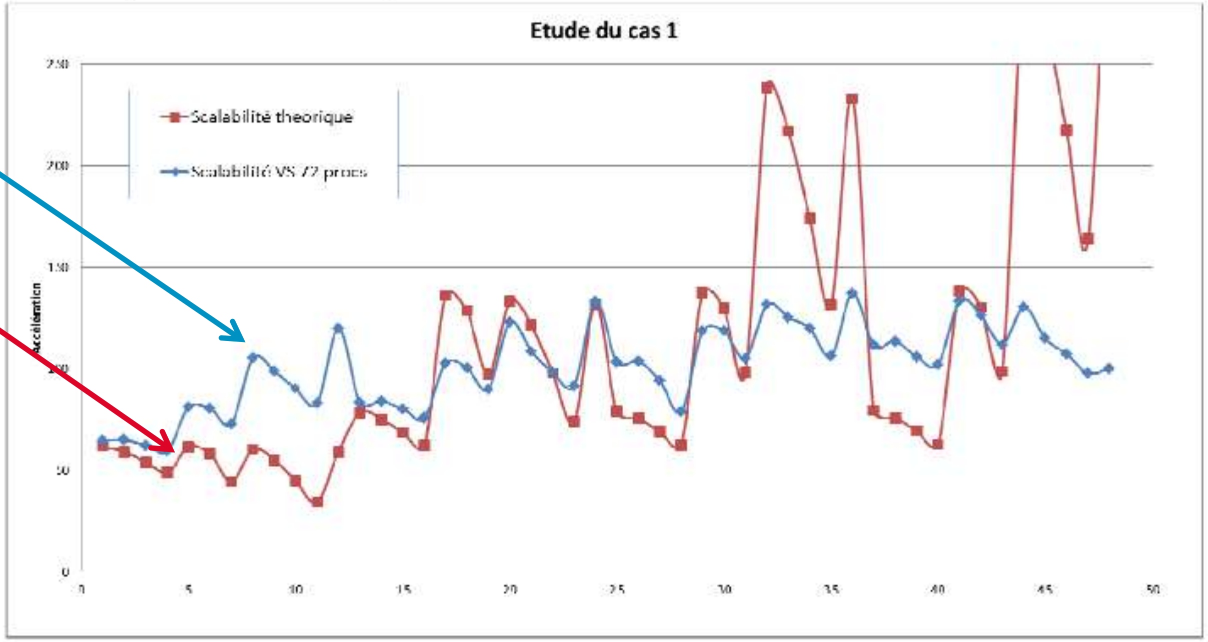
This input variable is used only when running ABINIT in parallel and for Ground-State calculations. It controls the automatic determination of parameters related to parallel work distribution (if not imposed in input file). Given a

- **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

ABINIT – « AUTOPARAL » INPUT VARIABLE

Real Speedup

Predicted Speedup



SPEEDING-UP ABINIT

SPEEDING-UP ABINIT

- **Find 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

SPEED-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 v8.10, 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`**: printing of wave-function (can be deactivated)
- **`prtden`**: printing of wave-function (can be deactivated most of the time)
- **`prtgsr`**: printing of GSR file (can be deactivated if abipy is not used)

SPEED-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

SPEED-UP ABINIT – CONVERGENCE CRITERIA

Always check the convergence with the studied properties!

- **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 (`tolmxf`)
  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, tolmxf, tolwfr, tolvrs`**:
  Tolerance criteria
  Except for `tolwfr`*, only one criterion can be used*

SPEED-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

SPEED-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 $\mathbf{G}^2_{\mathbf{cut}} \leq \mathbf{2E_{cut}}$, density should be expressed with wave-vectors up to $\mathbf{2G_{cut}}$.

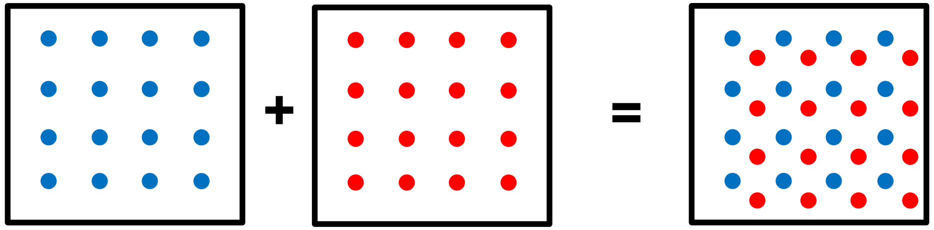
- N_{FFT} for density should be **2x** N_{FFT} for wave-function
- This ratio (`boxcutmin` input variable) can be decreased without a significant loss of precision.
- Always check!
- Not suitable if you prepare excited state or response function…
- **Very efficient on the execution time!**

ABINIT input variables

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

SPEED-UP ABINIT – ADVANCED 3

Use shifted k-points grids



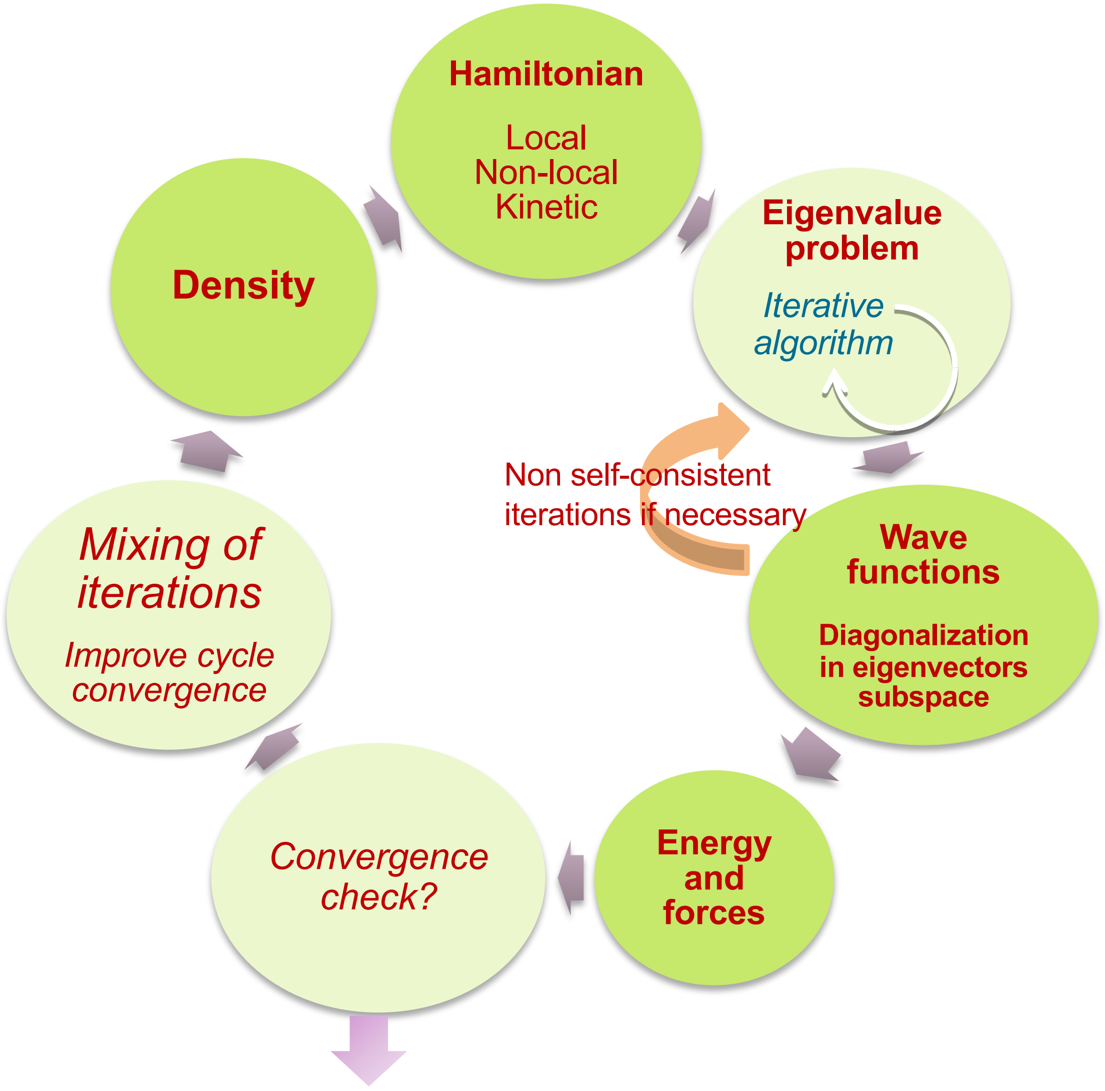
- There are optimal shifts for each Bravais lattice
- Much more efficient and less consuming than increasing the k-point density of one grid

ABINIT input variables

- `shiftk, nshiftk`: shifts to be applied to the k-points grid
  Default : one shift (½, ½, ½)

HELPING ABINIT TO CONVERGE

SELF-CONSISTENT CYCLE



Hamiltonian

Local
Non-local
Kinetic

Eigenvalue problem

Iterative algorithm

Non self-consistent iterations if necessary

Wave functions

Diagonalization in eigenvectors subspace

Energy and forces

Convergence check?

Mixing of iterations

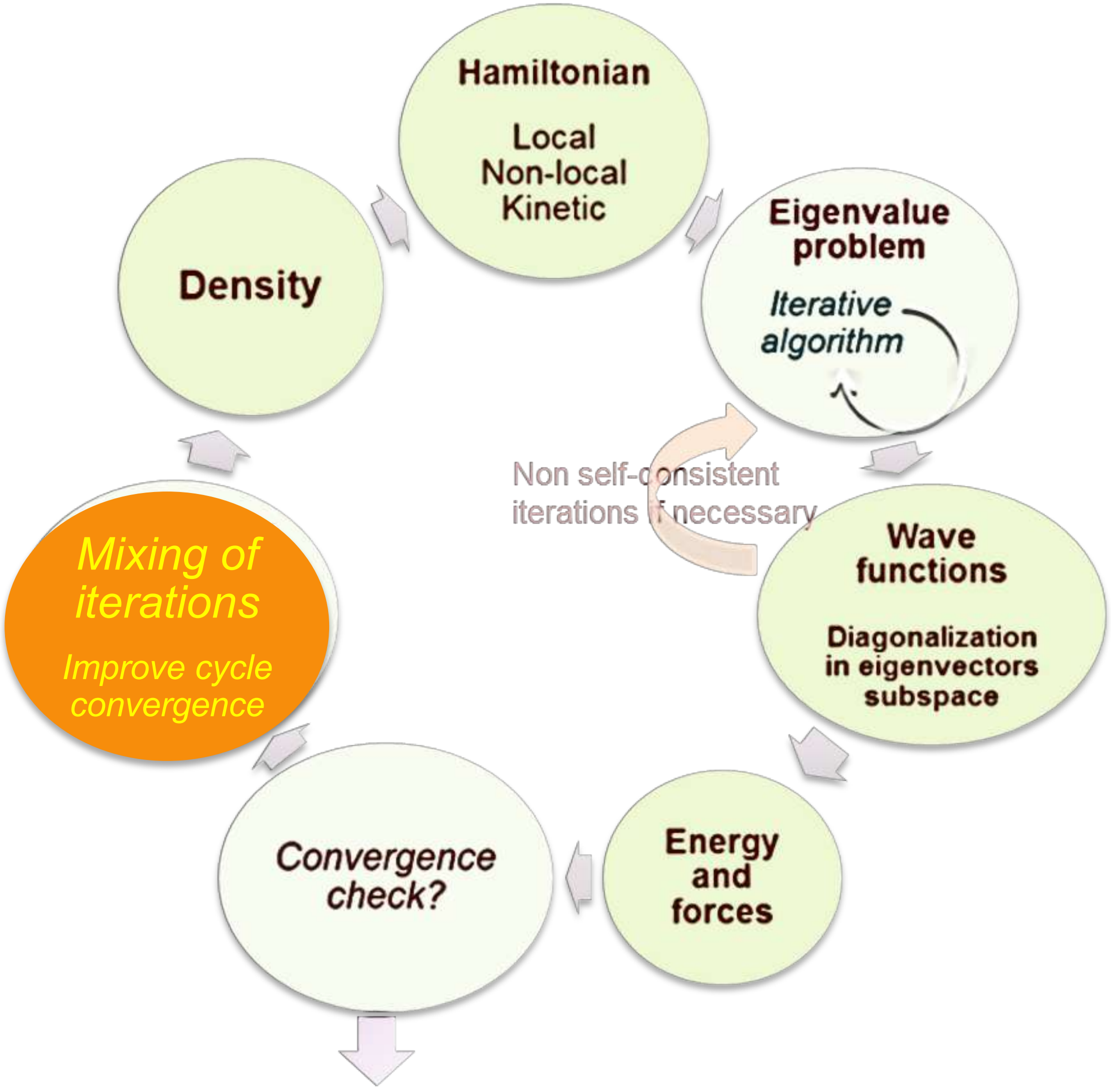
Improve cycle convergence

Density

SELF-CONSISTENT CYCLE - MIXING

Mixing the density with the densities of previous iterations

=> Damping of the oscillations



MIXING THE DENSITY

Mixing the new iteration and previous one

- Density residual : $\rho^{RES}(\mathbf{r}) = \rho^{OUT}(\mathbf{r}) - \rho^{IN}(\mathbf{r})$
- If no mixing : $new\ \rho(\mathbf{r}) = \rho^{IN}(\mathbf{r}) + \rho^{RES}(\mathbf{r}) = \rho^{OUT}(\mathbf{r})$

- **Preconditioning**

  Applying a transformation to $\rho^{RES}(\mathbf{r})$ in order to damp instabilities

$$\rho^{RES}(\mathbf{r}) \leftarrow \mathbf{P}\rho^{RES}(\mathbf{r})$$

- **Mixing**

  Mixing the new density and previous ones (favor small residuals)

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

MIXING THE DENSITY – ABINIT INPUT VARIABLES

Preconditioning

Use inverse of a model dielectric matrix

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

$$\mathrm{P(K)} = \varepsilon^{-1}(K) = \mathbf{diemix}\,.\left(\frac{1}{\mathbf{diemac}} + \mathbf{dielng}^2 K^2\right)/(1 + \mathbf{dielng}^2 K^2)$$

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)

Note:

- **iprcell**: use a more accurate “dielectric constant…

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)

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

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

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)

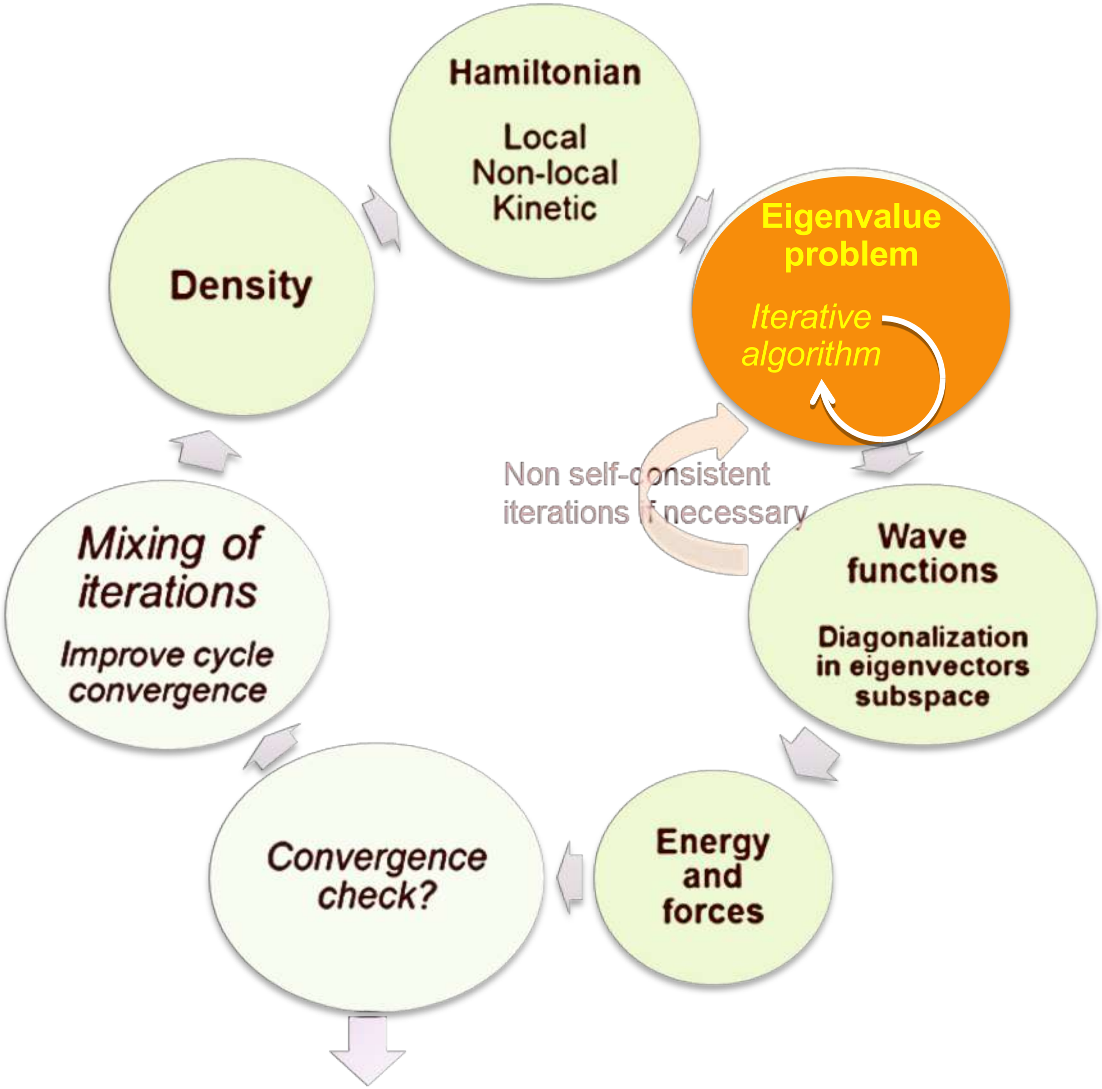
  Increasing the size can be very efficient (→ 30)

OPTIMIZATION OF WAVE-FUNCTIONS

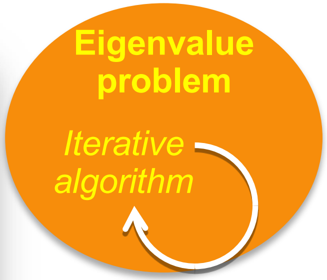
Better converge the wave-functions

=> Can reduce the Number of iterations

=> Each iteration takes more time



OPTIMIZATION OF WAVE-FUNCTIONS ALGORITHM AND PARAMETERS


Eigenvalue problem

Iterative algorithm

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

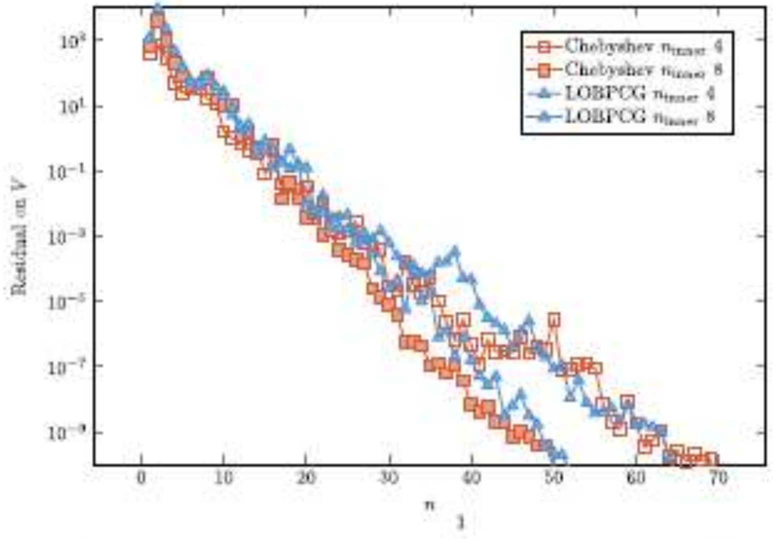


Fig. 6. Self-consistent convergence. The blocksize for LOBPCG was 128.

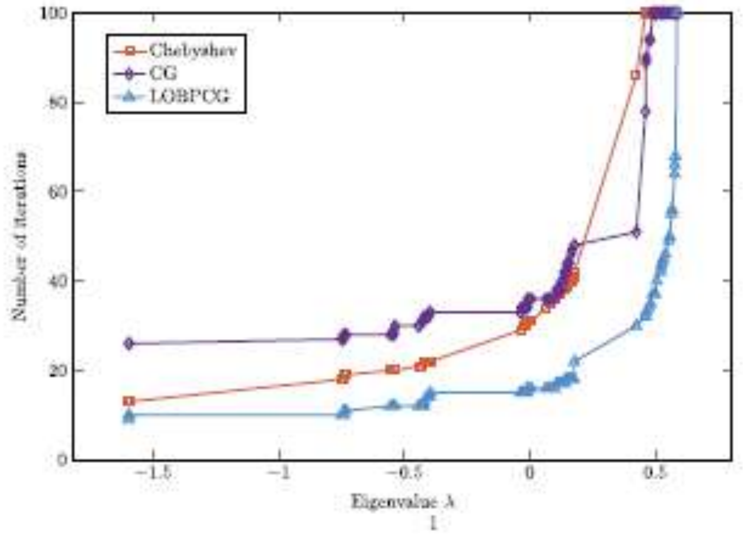
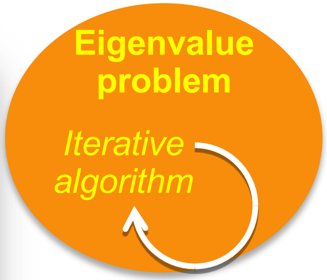


Fig. 4. Number of iterations to obtain a precision of 10^{-10}, $BaTiO_3$, 100 bands.

OPTIMIZATION OF WAVE-FUNCTIONS ALGORITHM AND PARAMETERS


Eigenvalue problem

Iterative algorithm

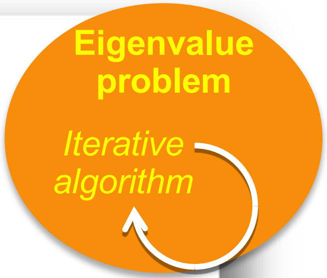
Choice of algorithm

- **Conjugate Gradient**
  Default when no parallelization or k-points parallelization
- **Block conjugate gradient** (LOBPCG)
  Default when Band-FFT parallelization
- **Chebyshev Filtering**
  For a very large number of processors

ABINIT input variables

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

OPTIMIZATION OF WAVE-FUNCTIONS
ALGORITHM AND PARAMETERS



Eigenvalue problem

Iterative algorithm

Algorithm parameters

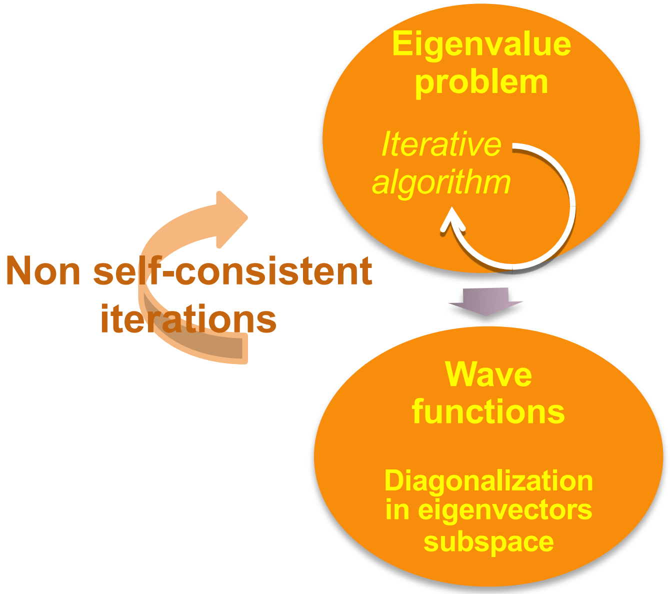
- Required tolerance on wave-functions
- Number of max. iterations
- Block Conjugate Gradient: size of the blocks
  – One block (size N_{band}) converges better than N_{band} blocks (size 1)
  – But can takes longer

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
- **`bandpp`**: Bands Per Proc
  Increase it to increase the size of blocks

OPTIMIZATION OF WAVE-FUNCTIONS
NON SELF-CONSISTENT ITERATIONS

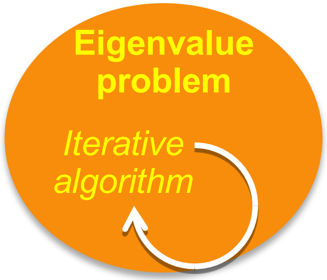
- Last chance if still no convergence…
  - Run the diagonalization algorithm several times, resetting it
  - **Increase the number of Non-Self-Consistent Iterations**

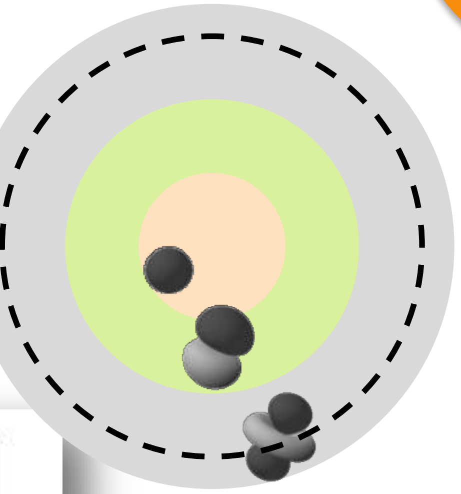

Eigenvalue problem
Iterative algorithm
Non self-consistent iterations
Wave functions
Diagonalization in eigenvectors subspace

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

OPTIMIZATION OF 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





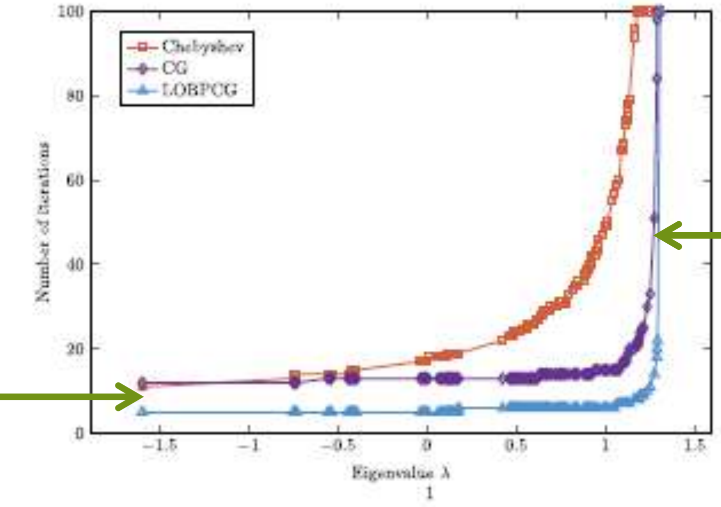
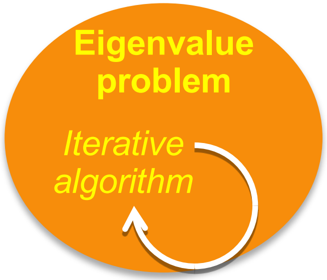


Fig. 5. Number of iterations to get to obtain a precision of 10^{-10}, BaTiO$_3$, 200 bands.

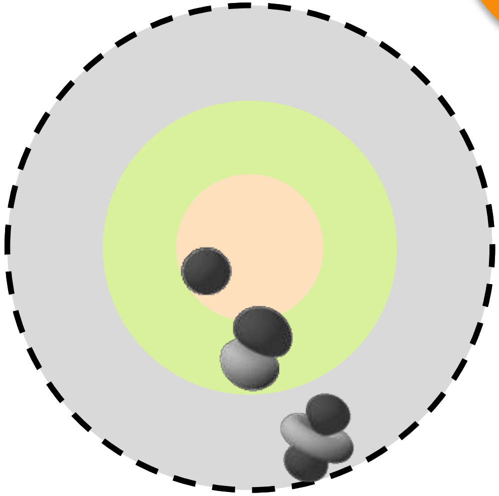
Occupied bands
Few iterations
to converge

Empty bands
Many iterations
to converge

OPTIMIZATION OF WAVE-FUNCTIONS EMPTY BANDS (METALS)

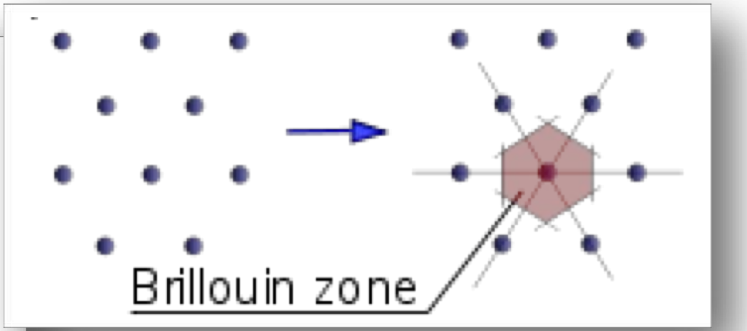


- Solution :
  - Change N_{band} in order to include all the states of the last shell



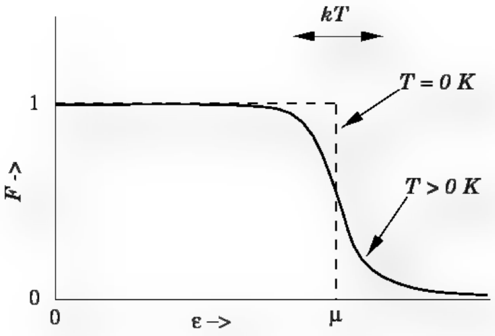
- ABINIT input variables
  - **nband**: Number of BANDs to optimize
    Warning (metals): check that last states are really empty!

METALS OR MAGNETISM – SAMPLING OF BRILLOUIN ZONE



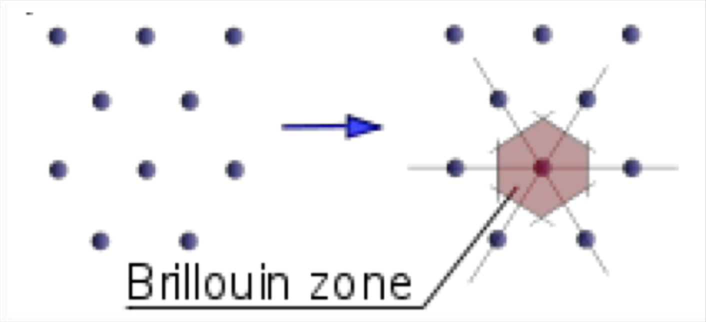
- Impact of sampling of Brillouin Zone:
  - Many properties are obtained from an integral over the Brillouin Zone
  - Magnetic moment is numerically sensitive

- 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 OR 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<=occopt<=7`)
  - **`tsmear`**: smearing temperature

CONCLUSION

TUNING ABINIT – KEYS 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

Commissariat à l’énergie atomique et aux énergies alternatives

Etablissement public à caractère industriel et commercial | RCS Paris B 775 685 019