The ABINIT code

X. Gonze

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

Ideas (1997):
1) Softwares for first-principles simulations are more and more sophisticated:
   one needs a worldwide collaboration, of specialized, complementary, groups
2) Linux software development: ‘free software’ model

Now (2016):
>1600 registered people on the forum
>800 kLines of F90 + many python scripts (abipy)
about 50 contributors to ABINITv7,
last release v8.0.6 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 – April 2015: 7 international ABINIT developer workshops (between 40 and 60 participants each)
- Jan 2010: launch of the Forum
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
...
Documentation

Web site  http://www.abinit.org

- User’s guide
- Installations notes
- List of input variables
- >30 tutorial lessons (each 1-2 hours)
  http://www.abinit.org/documentation/helpfiles/for-v8.10/tutorial/welcome.html

+ Forum Web site  http://forum.abinit.org

Videos

How to install on Linux ?
http://www.youtube.com/watch?v=DppLQ-KQA68

How to install on Windows 64bits ?
http://www.youtube.com/watch?v=EfJcYi1MNBg

First ABINIT tutorial (H₂ molecule)
http://www.youtube.com/watch?v=gcbfb_Mteo4

Sherbrooke, May 30, 2016
ABINIT tutorial: layout + dependencies

Four Basic Lessons

Spin
PAW1, 2, 3, Properties at nuclei
Wannier90
Polarization
Analysis tools
DFT+U, DMFT Determine U
TDDFT
Source Code

Resp Func 1
Optic
Resp Func 2, elphon elastic, NL

GW1
GW2, BSE

Parallel Ground state MD, Images
Parallel Ground state wavelets
Parallel Response Functions
Parallel Many Body Perturb. Theory

Intro ABINIT in parallel
Running ABINIT: basics
External files in a ABINIT run

Results: density (_DEN), potential (_POT), wavefunctions (_WFK), ...
ABINIT: the pipeline and the driver

Parser

Checks, prediction of memory needs ...

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

Linear Responses to atomic displacement electric field

Non-linear responses

GW computation of band structure

BSE excitons+optic

Processing units

Summary of results

CPU/Wall clock time analysis

Treatment of each dataset in turn

Sherbrooke, May 30, 2016
Density Functional Theory calculations

In ABINIT …

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

Periodic boundary conditions
=> wavefunctions characterized by a wavevector (k-vector)

PseudoPotentials (or Projector Augmented Waves – PAW)

Iterative techniques to solve the equations
(Schrödinger equation; DFT Self-consistency; optimisation of atomic positions)
A basic input file: dihydrogen (I)

# H2 molecule in a big box

# Definition of the unit cell
acell 10 10 10  # The keyword "acell" refers to the lengths of the primitive vectors (default in Bohr)

# Definition of the atom types
ntypat 1  # There is only one type of atom
znucl 1  # The keyword "znucl" refers to the atomic number of the possible type(s) of atom. The pseudopotential(s) mentioned in the "filenames" file must correspond to the type(s) of atom. Here, the only type is Hydrogen.

# Definition of the atoms
natom 2  # There are two atoms
typat 1 1  # They both are of type 1, that is, Hydrogen
xicart  # This keyword indicate that the location of the atoms will follow, one triplet of number for each atom
-0.7 0.0 0.0  # Triplet giving the cartesian coordinates of atom 1, in Bohr
0.7 0.0 0.0  # Triplet giving the cartesian coordinates of atom 2, in Bohr
A basic input file: dihydrogen (II)

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

# Definition of the k-point grid
kptopt 0 # Enter the 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 the 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
              # differ by less than toldfe (default in Hartree)
diemac 2.0 # Although this is not mandatory, it is worth to
             # precondition the SCF cycle. The model dielectric
             # function used as the standard preconditioner
             # is described in the "dielng" input variable section.
             # Here, we follow the prescriptions for molecules
             # in a big box
Specification of the atomic geometry
Prerequisites of plane waves

Plane waves $e^{iK \cdot r}$: particularly simple and efficient (when used with pseudopotentials), but infinite spatial extent. Cannot use a finite set of planewaves for finite systems! Need periodic boundary conditions. Primitive vectors $\mathbf{R}_j$, primitive cell volume $\Omega_0$.

OK for crystalline solids
But: finite systems, surfaces, defects, polymers, nanosystems ... ?
Solution: the supercell technique

The supercell must be sufficiently big: convergence study
Examples of defects SiO$_2$-quartz : Pb

72-atom supercell of quartz

Comparison with amorphous SiO$_2$
Main input file: input variable flexibility

- cell primitive vectors → rprim
  ... or angle (degrees) between primitive vectors → angdeg
  + scale cell vector lengths → acell
  + scale cartesian coordinates → scalecart
- number of atoms → natom
- reduced coordinates → xred (initial guess ... might be optimized)
  ... or cartesian coordinates → xcart (in Bohr) / xangstrom (in Å)
- type of atoms → typat
- space group → spgroup + natrd
  ... or number of symmetries → nsym
  + symmetry operations → symrel + t nons
Example: cubic zirconium dioxide

Assignment of Wyckoff Positions

Atoms Data:

<table>
<thead>
<tr>
<th>AT.</th>
<th>WP</th>
<th>SS</th>
<th>Representative</th>
<th>Atomic orbit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zr1</td>
<td>4a (0,0,0)</td>
<td>m-3m</td>
<td>(0.000000, 0.000000, 0.000000)</td>
<td>(0.000000, 0.000000, 0.000000)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.000000, 0.500000, 0.500000)</td>
<td>(0.500000, 0.000000, 0.500000)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.500000, 0.500000, 0.000000)</td>
<td>(0.000000, 0.500000, 0.000000)</td>
</tr>
<tr>
<td>O2</td>
<td>8c (1/4,1/4,1/4)</td>
<td>-43m</td>
<td>(0.250000, 0.250000, 0.250000)</td>
<td>(0.250000, 0.250000, 0.250000)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.750000, 0.750000, 0.750000)</td>
<td>(0.750000, 0.750000, 0.750000)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.750000, 0.250000, 0.250000)</td>
<td>(0.250000, 0.750000, 0.750000)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.250000, 0.750000, 0.250000)</td>
<td>(0.750000, 0.250000, 0.250000)</td>
</tr>
</tbody>
</table>

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

Face-centered cubic, with three atoms per primitive cell
Example: cubic zirconium dioxide

natom 3
acell 3*5.01 Angst
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)

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

<table>
<thead>
<tr>
<th>scalecart</th>
<th>9.5</th>
<th>9.8</th>
<th>10.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>rprim</td>
<td>0.0</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.5</td>
<td>0.0</td>
</tr>
</tbody>
</table>

face-centered orthorhombic

<table>
<thead>
<tr>
<th>acell</th>
<th>9.5</th>
<th>9.5</th>
<th>10.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>rprim</td>
<td>0.8660254038E+00</td>
<td>0.5</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>-0.8660254038E+00</td>
<td>0.5</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

hexagonal

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

<table>
<thead>
<tr>
<th>acell</th>
<th>9.5</th>
<th>9.5</th>
<th>10.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>angdeg</td>
<td>120</td>
<td>90</td>
<td>90</td>
</tr>
</tbody>
</table>

hexagonal

<table>
<thead>
<tr>
<th>acell</th>
<th>9.0</th>
<th>9.0</th>
<th>9.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>angdeg</td>
<td>48</td>
<td>48</td>
<td>48</td>
</tr>
</tbody>
</table>

trigonal
Symmetries in ABINIT

• **Seitz notation** for the symmetry operations of crystal:

\[
\begin{align*}
3 \times 3 \text{ real orthogonal matrix} & \quad \{ S \mid v(S) \} \\
\{ S_{\alpha\beta} \mid v_{\alpha}(S) \} & \quad \text{vector}
\end{align*}
\]

\begin{itemize}
  \item rotation \ symmetric \ \text{rel} \ (\text{symrel})
  \item translation \ \text{translation} \ \text{non-symmetry} \ (\text{tnons})
\end{itemize}

• Applied to the equilibrium position vector of atom \( \kappa \) relative to the origin of the cell \( \tau_\kappa \), this symmetry transforms it as:

\[
\begin{align*}
\{ S \mid v(S) \} \ \tau_\kappa &= S \tau_\kappa + v(S) = \tau_{\kappa'} + R^a \\
\{ S_{\alpha\beta} \mid v_{\alpha}(S) \} \ \tau_{\kappa\alpha} &= S_{\alpha\beta} \tau_{\kappa\alpha} + v_\alpha(S) = \tau_{\kappa'\alpha} + R^a
\end{align*}
\]

where \( R^a \) belongs to the real space lattice.
The plane wave basis set

\[ \psi_{k}(r) = \left( N\Omega_0 \right)^{-1/2} \sum_{G} u_k(G) e^{i(k+G)r} \]
A reminder: basic equations in DFT

Solve self-consistently the Kohn-Sham equation

\[
\begin{align*}
\hat{H} \psi_n &= \varepsilon_n \psi_n \\
\hat{H} &= \hat{T} + \hat{V} + \hat{V}_{Hxc}[\rho] \\
\rho(\vec{r}) &= \sum_{n}^{\text{occ}} \psi_n^{*}(\vec{r}) \psi_n(\vec{r}) \\
\delta_{mn} &= \langle \psi_m | \psi_n \rangle \text{ for } m, n \in \text{occupied set}
\end{align*}
\]

or minimize

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

with

\[
\hat{V}(\vec{r}) = \sum_{a\kappa} - \frac{Z_{\kappa}}{|\vec{r} - \vec{R}_a|}
\]

Sherbrooke, May 30, 2016
Periodic system : wavevectors

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

Bloch’s theorem

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

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

Normalization ?

Born-von Karman supercell

supercell vectors $N_j \mathbf{R}_j$ with $N=N_1N_2N_3$

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

Reciprocal lattice: set of \( G \) vectors such that

\[
e^{iG \mathbf{r}} \quad \text{has the periodicity of the real lattice}
\]

\[
u_k(\mathbf{r}) = \sum_G u_k(G) \ e^{iG \mathbf{r}}
\]

\[
u_k(G) = \frac{1}{\Omega_0} \int_{\Omega_0} e^{-iG \mathbf{r}} \ u_k(\mathbf{r}) \ \text{d}\mathbf{r}
\]

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

Kinetic energy of a plane wave

\[
- \frac{\nabla^2}{2} \rightarrow \frac{(k+G)^2}{2}
\]

The coefficients \( u_k(G) \) for the lowest eigenvectors decrease exponentially with the kinetic energy

Selection of plane waves determined by a cut-off energy \( E_{\text{cut}} \)

\[
\frac{(k + G)^2}{2} < E_{\text{cut}}
\]

Plane wave sphere
Number of plane waves

= function of the kinetic energy cut-off

… not continuous

Also, a (discontinuous) function of lattice parameter at fixed kinetic energy
Discontinuities in energy and pressure

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

Kinetic energy

\[ u(G) \neq 0 \quad u(G) = 0 \]

\[ (2E_{\text{cut}})^{1/2} \]

\[ k + G \]

Kinetic energy

\[ u(G) \neq 0 \quad u(G) = 0 \]

\[ (2E_{\text{cut}})^{1/2} \]

\[ (2(E_{\text{cut}} - E_{\text{cut smear}}))^{1/2} \]

ecutsm
Convergence wrt to kinetic energy cutoff

Total energy (Ha)

Cut-off energy (Ha)

bulk silicon

Courtesy of F. Bruneval

Sherbrooke, May 30, 2016
Plane waves: the density and potential

Fourier transform of a periodic function $f(r)$

$$f(G) = \frac{1}{\Omega_{or}} \int_{\Omega_{or}} e^{-iGr} f(r) \, dr$$

$$f(r) = \sum_{G} e^{iGr} f(G)$$

Poisson equation

$$V_H(r) = \int \frac{n(r')}{|r-r'|} \, dr' \quad \Rightarrow \quad \nabla^2 V_H|_r = -4\pi n(r)$$

Relation between Fourier coefficients:

$$G^2 V_H(G) = 4\pi n(G)$$

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

For $G^2 = 0$ (i.e., $G=0$) divergence of $V_H(G=0)$

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

Average
Representation of the density

Density associated with one eigenfunction:

\[ n_{nk}(r) = u^*_{nk}(r) u_{nk}(r) \]

Computation of:

\[ u^*_{nk}(r) u_{nk}(r) \]

\[ = \left( \sum_{G} u^*_{nk}(G) e^{-iGr} \right) \left( \sum_{G'} u_{nk}(G') e^{-iG'r} \right) \]

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

Non-zero coefficients for \( k+G \in \) sphere
\( k+G' \in \) sphere

The sphere for \( n(G) \) has a double radius
From real space to reciprocal space

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

Use of the discrete Fourier transform

$$\{r_i\} \leftrightarrow \{G\}$$

Reciprocal lattice

$$n(G) = \frac{1}{N_{r_i}} \sum_{\{r_i\}} n(r_i) e^{-iGr_i}$$

Real lattice: original cell

Fast Fourier Transform algorithm
Representation : wrap-up

- Choice of a basis (e.g. Plane waves)
- Truncating of the basis -> finite basis
  \[ \frac{(k+G)^2}{2} < E_{cut} \] Sphere of plane waves
- Discontinuous increase of the number of plane waves?
  Smearing of \( u(G) \)
  -> Progressive incorporation of new \( G \) vectors
- Representation of the density
  Sphere with a double radius in the reciprocal space
- Going from the real space to reciprocal space
  Discrete Fourier transform
  Grid of points + Fast Fourier Transform
  \[ \{ r_i \} \leftrightarrow \{ G \} \]
Sampling the Brillouin zone
From discrete states to Brillouin zone

Discrete summations over states:

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

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

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

**Total kinetic energy**
$$\sum_n \frac{1}{\Omega_{0k}} \int_{\Omega_{0k}} f(\varepsilon_F - \varepsilon_{nk}) \langle \psi_{nk} | -\frac{1}{2} \nabla^2 | \psi_{nk} \rangle dk$$

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

How to treat
$$\frac{1}{\Omega_{0k}} \int_{\Omega_{0k}} X_k dk$$

Sherbrooke, May 30, 2016
Brillouin zone integration

\[ \frac{1}{\Omega_\omega} \int_{\Omega_\omega} X_k \, dk \Rightarrow \sum_{\{k\}} w_k X_k \quad \text{[ with } \sum_{\{k\}} w_k = 1 \text{ ]} \]

How to chose \( \{k\} \) and \( \{w_k\} \) ?

Special points \( \rightarrow \) Weights

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

Homogeneous sampling of the Brillouin zone
Brillouin zone integration

Theorem:

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

Then exponential convergence, with respect to \(\Delta k\)

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

- Uniformly spaced grid of $n_{k1} \times n_{k2} \times n_{k3}$ points in the first Brillouin Zone

\[
\begin{align*}
G_2 & = G_1 \\
G_2 & = G_1 \\
n_{k1} = n_{k2} = 3 & \quad n_{k1} = n_{k2} = 4
\end{align*}
\]

ngkpt  nk1  nk2  nk3
Unshifted and shifted grids

- k-points grid can be chosen to be shifted: not centered at \( \Gamma \).
- 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 \((1/2,1/2)\)

\[ \text{ngkpt \ nk1 \ nk2 \ nk3} \]
\[ \text{shiftk \ sk1 \ sk2 \ sk3} \]  (default: 0.5 0.5 0.5)
Combining grids with various shifts

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

\[ G_2 \]

\[ G_1 \]

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

\[
\begin{array}{cccc}
ngkpt & nk1 & nk2 & nk3 \\
nshiftk & nsk \\
shiftk & sk1(1) & sk2(1) & sk3(1) \\
 & sk1(2) & sk2(2) & sk3(2) \\
 & \vdots & \vdots & \vdots \\
 & sk1(nsk) & sk2(nsk) & sk3(nsk)
\end{array}
\]
Irreducible wedge

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

![Diagram showing the irreducible wedge with points labeled and counts indicated]
**Treatment of metals (I)**

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

Discontinuity of integrand at Fermi level

Smearing technique

First trial: generalisation of DFT to finite temperature

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

$f$ goes from 0 to 2 in an energy range

$$E(T) \approx E(T=0) + \alpha T^2 + ...$$

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

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

Sherbrooke, May 30, 2016
Treatment of metals (II)

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

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

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

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

Gaussian smearing \[ \tilde{\delta}(x) = \frac{1}{\sqrt{\pi}} \, e^{-x^2} \quad \Rightarrow \alpha \quad \text{small} \]

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

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

'Cold Smearing' \[ \text{occopt 4/5} \]

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

$\Rightarrow n = 3$ with positive occupations
Convergence wrt k-points and smearing

Total energy (Ry)

bulk Al

$E_{\text{total}}(n_k (n_k \times n_k \times n_k \text{ k-point grid}))$

Courtesy of S. Narasimhan

Sherbrooke, May 30, 2016
How many k points? Smearing width?

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

Semiconductors - Insulators
\[ \# k \times N_{\text{atoms}} \quad 50 \ldots 500 \]

Metals
\[ \# k \times N_{\text{atoms}} \quad 1000 \ldots 2000 \]

⚠️ Use symmetries ⇒ integration in the irreducible Brillouin zone

2D Example

\[
\begin{array}{cccc}
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\end{array}
\]
grid \( 4 \times 4 = 16 \)

3 points in the irreducible Brillouin Zone

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

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

d Metal (Cu, Ag...) \( \sim 0.01 \) Ha

⚠️ magnetism needs small \( \sigma \)
Pseudopotentials
Core and valence electrons (I)

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

It depends on the accuracy of the calculation!

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

\[ n(r) = \sum_{i}^{N} \psi_i^*(r) \psi_i(r) \]

\[ = \sum_{i \in \text{core}}^{N_{\text{core}}} \psi_i^*(r) \psi_i(r) + \sum_{i \in \text{val}}^{N_{\text{val}}} \psi_i^*(r) \psi_i(r) = n_{\text{core}}(r) + n_{\text{val}}(r) \]

« Frozen core » for \( i \in \text{core} : \psi_i = \psi_i^{\text{atom}} \)
Small core / Large core

It depends on the target accuracy of the calculation!
(remark also valid for pseudopotentials, with similar cores)
For some elements, the core/valence partitioning is obvious, for some others, it is not.

F atom:
\[
(1s)^2 + (2s)^2(2p)^5
\]
IP 1keV 10-100 eV

Ti atom:
\[
(1s)^2(2s)^2(2p)^6(3s)^2(3p)^6(4s)^2(3d)^2 \quad \text{small core}
\]
\[
(1s)^2(2s)^2(2p)^6(3s)^2(3p)^6(4s)^2(3d)^2 \quad \text{large core}
\]
IP 99.2 eV 43.3 eV

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

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

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

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

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

\[ + \int \frac{n_{\text{val}}(\mathbf{r}_1)n_{\text{core}}(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} \, d\mathbf{r}_1 d\mathbf{r}_2 + E_{xc}[n_{\text{core}} + n_{\text{val}}] \]
Removing core electrons (I)

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

Pseudopotentials try to remove completely the core orbitals from the simulation

Problem with the number of nodes
This is a strong modification of the system ...

Pseudopotentials confine the strong changes within a « cut-off radius »
Removing core electrons (II)

Going from \[\left(-\frac{1}{2} \nabla^2 + v\right) |\psi_i> = \epsilon_i |\psi_i>\]

To \[\left(-\frac{1}{2} \nabla^2 + v_{ps}\right) |\psi_{ps, i}> = \epsilon_{ps, i} |\psi_{ps, i}>\]

Possible set of conditions (norm-conserving pseudopotentials)

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

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

3s Radial wave function of Si

\[ V_{ps}(r) [\text{Ha}] \]

Radial distance [a.u.]

Radial distance [a.u.]
Pseudopotentials/PAW data in ABINIT

- Preferred PAW atomic dataset table: JTH v0.2
  Jollet, Torrent, Holzwarth, Computer Physics Comm. 185, 1246 (2014)
  http://www.abinit.org/downloads/PAW2

Also, possibility to use: GPAW table, GBRV v1.0 table, or norm-conserving pseudopotentials (e.g. ONCVPSP pseudo generator), or many other pseudos...
Computing the forces
Computing the forces (I)

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

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

\[
F_{\kappa,\alpha} = -\frac{\partial E}{\partial R_{\kappa,\alpha}} |_{\{ \mathbf{R}_\kappa \}} \quad \text{(principle of virtual works)}
\]

Forces can be computed by finite differences.

Better approach: compute the response to a perturbation

⇒ What is the energy change?

\[
\left\{ \mathbf{R}_{\kappa,\alpha} \right\} \rightarrow \left\{ \mathbf{R}_{\kappa,\alpha} + \lambda \delta \mathbf{R}_{\kappa,\alpha} \right\} \quad \text{Small parameter}
\]
Computing the forces (II)

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

Perturbation theory: Hellmann - Feynman theorem

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

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

Application to the derivative with respect to an atomic displacement:

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

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

Generalisation to density functional theory

Reminder:

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

If change of atomic positions ...

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

(can be generalized to pseudopotential case)

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

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

Forces can be computed directly from the density!
Iterative algorithms
Algorithmics : problems to be solved

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

Size of the system
[2 atoms… 600 atoms…] + vacuum ?
Dimension of the vectors \( x_i \) 300… 100 000… (if planewaves)
# of (occupied) eigenvectors 4… 1200…

(2) Self-consistency

(3) Geometry optimization

Find the positions \( \{ \mathbf{R}_k \} \) of ions such that the forces \( \{ \mathbf{F}_k \} \) vanish
[ = Minimization of energy ]

Current practice : iterative approaches
The ‘steepest-descent’ algorithm

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

=> Iterative algorithm.
Choose a starting geometry, then a parameter $\lambda$, and iterately update the geometry, following the forces:

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

Equivalent to the simple mixing algorithm of SCF (see later)
Energy+forces around equilib. geometry

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

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

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

Moreover,

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

Vector and matrix notation

$$
\mathbf{R}_{\kappa,\alpha}^* \rightarrow \mathbf{R}^* \\
\mathbf{F}_{\kappa,\alpha} \rightarrow \mathbf{F} \\
\mathbf{R}_{\kappa',\alpha'} \rightarrow \mathbf{R} \\
\frac{\partial^2 E_{BO}}{\partial \mathbf{R}_{\kappa,\alpha} \partial \mathbf{R}_{\kappa',\alpha'}} \left\{ \mathbf{R}_{\kappa,\alpha}^* \right\} \rightarrow \mathbf{H} \quad \text{(the Hessian)}
$$
Steepest-descent : analysis (I)

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

Analysis of this algorithm, in the linear regime:

\[ F(R) = F(R^*) - H(R - R^*) + O(R - R^*)^2 \]

\[ R^{(n+1)} = R^{(n)} + \lambda F^{(n)} \quad \Rightarrow \quad (R^{(n+1)} - R^*) = (R^{(n)} - R^*) - \lambda H (R^{(n)} - R^*) \]

\[ (R^{(n+1)} - R^*) = (1 - \lambda H) (R^{(n)} - R^*) \]

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

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

Need to understand the action of the matrix (or operator) \( 1 - \lambda H \)
What are the eigenvectors and eigenvalues of $H$?

$H$ is a symmetric, positive definite matrix

$$
\begin{bmatrix}
\frac{\partial^2 E^{BO}}{\partial R_{\kappa,\alpha} \partial R_{\kappa',\alpha'}}
\end{bmatrix}
$$

$$
H f_i = h_i f_i \quad \text{where} \quad \{f_i\} \quad \text{form a complete, orthonormal, basis set}
$$

Discrepancy decomposed as

$$
(R^{(n)} - R^*) = \sum c_i^{(n)} f_i
$$

and

$$
(R^{(n+1)} - R^*) = (1 - \lambda H) \sum c_i^{(n)} f_i = \sum c_i^{(n)} (1 - \lambda h_i) f_i
$$

The coefficient of $f_i$ is multiplied by $1 - \lambda h_i$

Iteratively:

$$
(R^{(n)} - R^*) = \sum c_i^{(0)} (1 - \lambda h_i)^{(n)} f_i
$$
Steepest-descent : analysis (III)

\[
\left( R^{(n)} - R^* \right) = \sum_i c_i^{(0)} (1 - \lambda h_i)^{(n)} f_i
\]

The size of the discrepancy decreases if \( |1 - \lambda h_i| < 1 \).

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

Hence positive definite \( \implies \) all \( h_i \) are positive

Yes! If \( \lambda \) positive, sufficiently small ...
Steepest-descent: analysis (IV)

\[
\left( R^{(n)} - R^* \right) = \sum_i c_i^{(0)} (1 - \lambda h_i)^{(n)} f_i
\]

How to determine the optimal value of \( \lambda \)?

The maximum of all \(|1 - \lambda h_i|\) should be as small as possible.
At the optimal value of \( \lambda \), what will be the convergence rate?
( = by which factor is reduced the worst component of \( R^{(n)} - R^* \) ? )

As an exercise: suppose \( h_1 = 0.2 \)
\( h_2 = 1.0 \)
\( h_3 = 5.0 \) => what is the best value of \( \lambda \)?

+ what is the convergence rate?

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

\[
\begin{align*}
    h_1 &= 0.2 \quad |1 - \lambda \cdot 0.2| & \text{optimum} \Rightarrow \lambda &= 5 \\
    h_2 &= 1.0 \quad |1 - \lambda \cdot 1| & \text{optimum} \Rightarrow \lambda &= 1 \\
    h_3 &= 5.0 \quad |1 - \lambda \cdot 5| & \text{optimum} \Rightarrow \lambda &= 0.2
\end{align*}
\]

\[
\mu = |1 - \lambda \cdot 0.2| = |1 - \lambda \cdot 5| \\
\text{positive} \quad 1 - \lambda \cdot 0.2 = -(1 - \lambda \cdot 0.5) \\
\text{negative} \quad 2 - \lambda \cdot (0.2 + 5) = 0 \Rightarrow \lambda = 2/5.2 \\
\mu = 1 - 2 \cdot (0.2 / 5.2)
\]

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

Note: the second eigenvalue does not play any role.

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

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

Perfect if \( h_{\text{max}} = h_{\text{min}} \). Bad if \( h_{\text{max}} >> h_{\text{min}} \).

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

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

For a generic decrease factor \( \Delta \), with "n" the number of iterations.

\[
\|F^{(n)}\| \approx \left(\frac{h_{\text{max}}/h_{\text{min}} - 1}{h_{\text{max}}/h_{\text{min}} + 1}\right)^n \|F^{(0)}\| \\
\Delta \approx \left(\frac{h_{\text{max}}/h_{\text{min}} - 1}{h_{\text{max}}/h_{\text{min}} + 1}\right)^n
\]

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

(The latter approximate equality supposes a large condition number)
Analysis of self-consistency

Natural iterative methodology (KS: in $\Rightarrow$ out):

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

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

The difference

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

Simple mixing algorithm ($\approx$ steepest-descent)

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

Analysis...

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

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

$$\frac{\delta v_{out}}{\delta v_{in}}$$
Modify the condition number (II)

\[ \mathbf{R}^{(n+1)} = \mathbf{R}^{(n)} + \lambda (H^{-1})_{\text{approx}} \mathbf{F}^{(n)} \]

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

Notes:
1) If approximate inverse Hessian perfect, optimal geometry is reached in one step, with \( \lambda = 1 \). Steepest-descent NOT the best direction.
2) Non-linear effects not taken into account. For geometry optimization, might be quite large. Even with perfect hessian, need 5-6 steps to optimize a water molecule.
3) Approximating inverse hessian by a multiple of the unit matrix is equivalent to changing the \( \lambda \) value.
4) Eigenvalues and eigenvectors of \((H^{-1})_{\text{approx}} H\) govern the convergence: the condition number can be changed. Often called a "pre-conditioner".
5) Generalisation to other optimization problems is trivial.

(The Hessian is referred to as the Jacobian if it is not symmetric.)
Modify the condition number (III)

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

Selfconsistent determination of the Kohn-Sham potential:

Jacobian = dielectric matrix.
Lowest eigenvalue close to 1.
Largest eigenvalue:
\[ = 1.5 \ldots 2.5 \] for small close-shell molecules, and small unit cell solids
(Simple mixing will sometimes converge with parameter set to 1 !)
\[ = \text{the macroscopic dielectric constant} \] (e.g. 12 for silicon),
for larger close-shell molecules and large unit cell insulators,
\[ = \text{diverge} \] for large-unit cell metals, or open-shell molecules!

Model dielectric matrices known for rather homogeneous systems.
Knowledge of approx. macroscopic dielectric constant
\[ => \text{efficient preconditioner} \]
Work in progress for inhomogeneous systems
(e.g. metals/vacuum systems).
Advanced algorithms : using the history

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

Large class of methods :
- Broyden (quasi-Newton-type),
- Davidson,
- conjugate gradients,
- Lanczos ...

Approximate Hessian can be combined with usage of history
Stages in the main processing unit

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

(1) Kohn - Sham equation

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

Details are usually hidden to the user

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

(2) Self-consistency

\[
V_{KS}(\mathbf{r}) \xrightarrow{\text{self-consistency}} \psi_i(\mathbf{r})
\]

Target tolerance \( \text{toldfe, toldff, tolrff, tolvrs} \)

+ Maximal number of loops \( \text{nstep} \)

Preconditioner \( \text{diemac, diemix, …} \)

(3) Geometry optimization / molecular dynamics

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

Target tolerance \( \text{tolmxf} \)

+ Maximal number of loops \( \text{ntime} \)

Algorithm \( \text{ionmov} \)
Computing band structure and density of states
Non-self consistent calculations

- Once the density has been determined self-consistently, it is possible to compute the eigenenergies/eigenfunctions rapidly for a large number of wavevectors, at fixed KS potential
- Band structure:

  non self-consistent calculation (iscf -2)
  k-points along high-symmetry directions (kptopt<0;kptbounds;ndivk).
Density of states

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

$$g(E) = \sum_n \frac{1}{\Omega_{0k}} \int_{\Omega_{0k}} \delta(E - \epsilon_{nk}) dk$$

- Recipe: determine $\epsilon_{nk}$ on a grid of k-points in the BZ using a non self-consistent procedure ($iscf$ -3).

- The $\delta$-function is approximated by smeared-out function (typically a Gaussian) with a width $\sigma$ ($prtdos$ 1).

- Very fine grid of k-points needed to get sharp features accurately. For faster convergence, use tetrahedron method to interpolate between k-points ($prtdos$ 2).
Example: $\text{SiO}_2$-quartz
Summary

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

=> relaxation of geometry, or molecular dynamics