Plane waves and pseudopotentials
(Part II)

X. Gonze
Université Catholique de Louvain,
Louvain-la-neuve, Belgium
Outline

I. **Iterative techniques**
   (the Kohn-Sham equation, the SCF convergence, optimization of the geometry)
Algorithmics : problems to be solved

(1) Kohn - Sham equation

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

\[\Lambda \mathbf{x}_i = \lambda_i \mathbf{x}_i\]

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

(2) Self-consistency

\[V_{KS}(\mathbf{r}) \xrightarrow{\psi_i(\mathbf{r})} \xrightarrow{n(\mathbf{r})}\]

(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
Analysis of simple iterative algorithms
The ‘steepest-descent’ algorithm (I)

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
Analysis of self-consistency

\[ V_{KS}(\mathbf{r}) \xrightarrow{\psi_i(\mathbf{r})} n(\mathbf{r}) \]

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

\[ V_{in}(\mathbf{r}) \rightarrow \psi_i(\mathbf{r}) \rightarrow n(\mathbf{r}) \rightarrow V_{out}(\mathbf{r}) \]

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

The difference \[ V_{out}(\mathbf{r}) - V_{in}(\mathbf{r}) \] (generic name : a "residual")

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

\[ V_{in}^{(n+1)} = V_{in}^{(n)} + \lambda \left( V_{out}^{(n)} - V_{in}^{(n)} \right) \]
Energy and forces close to the equilibrium geometry

Let us denote the equilibrium geometry as \( R_{k,\alpha}^* \).

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

\[
F_{k,\alpha}(R_{k',\alpha'}) = F_{k,\alpha}(R_{k',\alpha'}^*) + \sum_{k',\alpha'} \left. \frac{\partial F_{k,\alpha}}{\partial R_{k',\alpha'}} \right|_{\{R^*\}} (R_{k',\alpha'} - R_{k',\alpha'}^*) + O\left( (R_{k',\alpha'} - R_{k',\alpha'}^*)^2 \right)
\]

Moreover,

\[
F_{k,\alpha} = -\frac{\partial E^{\text{BO}}}{\partial R_{k,\alpha}} \quad \frac{\partial F_{k',\alpha'}}{\partial R_{k,\alpha}} = -\frac{\partial^2 E^{\text{BO}}}{\partial R_{k,\alpha} \partial R_{k',\alpha'}}
\]

Vector and matrix notation

\[
R_{k,\alpha}^* \rightarrow \mathbf{R}^* \quad \mathbf{F}_{k,\alpha} \rightarrow \mathbf{F} \quad \frac{\partial^2 E^{\text{BO}}}{\partial R_{k,\alpha} \partial R_{k',\alpha'}} \left\{ R_{k,\alpha}^* \right\} \rightarrow \mathbf{H} \quad \text{(the Hessian)}
\]
The ‘steepest-descent’ algorithm (II)

\[ R^{(n+1)}_{k,\alpha} = R^{(n)}_{k,\alpha} + \lambda F^{(n)}_{k,\alpha} \]

Analysis of this algorithm, in the linear regime:

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

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

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

For convergence of the iterative procedure, the "distance" between the trial geometry and the 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)
The ‘steepest-descent’ algorithm (III)

What are the eigenvectors and eigenvalues of $H$?

$H$ symmetric, positive definite matrix

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

Thus the discrepancy can be decomposed as

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

and

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

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

Iteratively:

$$R^{(n)} - R^* = \sum_i c_i^{(0)} (1 - \lambda h_i)^{(n)} f_i$$

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?

$H$ positive definite $\Rightarrow$ all $h_i$ are positive

Yes! If $\lambda$ positive, sufficiently small ...
The ‘steepest-descent’ algorithm (IV)

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

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 \(\mu\) ?
( = by which factor is reduced the worst component of \((R^{(n)} - R^*)\) ? )

As an exercise: suppose
\[
\begin{align*}
  h_1 &= 0.2 \\
  h_2 &= 1.0 \\
  h_3 &= 5.0
\end{align*}
\]

\[\Rightarrow\] what is the best value of \(\lambda\) ?

+ what is the convergence rate \(\mu\) ?

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.
The ‘steepest-descent’ algorithm (V)

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

$h_1 = 0.2 \quad |1 - \lambda \cdot 0.2| \quad \text{optimum} \Rightarrow \lambda = 5$

$h_2 = 1.0 \quad |1 - \lambda \cdot 1| \quad \text{optimum} \Rightarrow \lambda = 1$

$h_3 = 5.0 \quad |1 - \lambda \cdot 5| \quad \text{optimum} \Rightarrow \lambda = 0.2$

\[
\begin{align*}
\mu &= |1 - \lambda \cdot 0.2| = |1 - \lambda \cdot 5| \\
1 - \lambda \cdot 0.2 &= -(1 - \lambda \cdot .5) \\
2 - \lambda (0.2+5) &= 0 \Rightarrow \lambda = 2/5.2 \\
\mu &= 1 - 2 \cdot (0.2 / 5.2)
\end{align*}
\]

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

Note: the second eigenvalue does not play any role.
The convergence is limited by the extremal eigenvalues: if the parameter is too large, the smallest eigenvalue will cause divergence, but for that small parameter, the largest eigenvalue lead to slow decrease of the error...
The condition number

In general, 

\[ \lambda_{\text{opt}} = \frac{2}{(h_{\text{min}} + h_{\text{max}})} \]

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

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?

Let us work this out for a generic decrease factor \( \Delta \), with "n" the number of iterations.

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

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

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

\[ \ln \Delta \approx 0.5 \left( \frac{h_{\text{max}}}{h_{\text{min}}} \right) \ln \frac{1}{\Delta} \]

(The latter approximate equality suppose a large condition number)
How to do a better job?

1) Taking advantage of the history

Simple mixing
Steepest descent
Power + shift \[ \{ \text{very primitive algorithms} \} \]
\[ v^{(n)} \rightarrow v^{(n+1)} \]

The information about previous iterations is completely ignored
We already know several vector/residual pairs \( (v^{(n)}, r^{(n)}) \)
We should try to use them!

\[
\begin{align*}
R & \rightarrow F \\
V_{\text{in}} & \rightarrow V_{\text{out}} - V_{\text{in}} \\
\left| \psi^T \right> & \rightarrow \left\{ \begin{array}{l}
\langle \epsilon \rangle = \left| \psi^T \right| \hat{H} \left| \psi^T \right>
nR = \left( \hat{H} - \langle \epsilon \rangle \right) \left| \psi^T \right>
\end{array} \right.
\end{align*}
\]

The residual vector is the nul vector at convergence in all three cases.

2) Decreasing the condition number
Minimisation of the residual (I)

Suppose we know that $v^{(p)}$ gives $r^{(p)}$ for $p=1, 2 \ldots n$

We try to find the best $v$ that can be obtained by combining the latest one with its differences with other $v^{(p)}$.

$$v = v^{(n)} + \sum_{p=1}^{n-1} s_p \left( v^{(p)} - v^{(n)} \right)$$

This is equivalent to

$$v = \sum_{p=1}^{n} s_p v^{(p)} \quad \text{with} \quad 1 = \sum_{p=1}^{n} s_p$$

What is the residual associated with $v$, if we make a linear approximation?

$$r = H \left( v - v^* \right) = H \left( \sum_{p=1}^{n} s_p v^{(p)} - \left( \sum_{p=1}^{n} s_p \right) v^* \right)$$

$$= \left( \sum_{p=1}^{n} s_p \left( v^{(p)} - v^* \right) \right) = \sum_{p=1}^{n} s_p H \left( v^{(p)} - v^* \right) = \sum_{p=1}^{n} s_p r^{(p)}$$

⇒ The new residual is a linear combination of the old ones, with the same coefficients as those of the potential.

An excellent strategy is select the $s_p$ such as to minimize the norm of the residual (RMM = residual minimisation method - Pulay). Then one mixes part of the predicted residual to the predicted vector.
Minimisation of the residual (II)

Characteristics of the RMM method:
(1) it takes advantage of the whole history
(2) it makes a linear hypothesis
(3) one needs to store all previous vectors and residuals
(4) it does not modify the condition number

Point (3): memory problem if all wavefunctions are concerned, and the basis set is large (plane waves, or discretized grids). Might sometimes also be a problem for potential-residual pairs, represented on grids, especially for a large number of iterations. No problem of memory for geometries and forces.

Simplified RMM method: Anderson's method, where only two previous pairs are kept.

Modifying the condition number (I)

Back to the optimization of geometry, with the linearized relation between forces, hessian and nuclei configuration:

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

Steepest-descent:
\[ R^{(n+1)} = R^{(n)} + \lambda F^{(n)} \]

Now, suppose an approximate inverse Hessian \( H^{-1} \)_{approx}

Then, applying \( H^{-1} \)_{approx} on the forces, and moving the nuclei along these modified forces gives
\[ R^{(n+1)} = R^{(n)} + \lambda \left( H^{-1} \right)_{approx} F^{(n)} \]

The difference between trial configuration and equilibrium configuration, in the linear approximation, behaves like

\[ \left( R^{(n+1)} - R^* \right) = \left( 1 - \lambda \left( H^{-1} \right)_{approx} H \right) \left( R^{(n)} - R^* \right) \]
Modifying the condition number (II)

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

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

Notes:
1) If the approximate inverse Hessian is perfect, the optimal geometry is reached in one step, with \( \lambda = 1 \). Thus the steepest-descent is NOT the best direction in general.
2) Non-linear effects are not taken into account. For geometry optimization, they might be quite large. Even with the perfect hessian, one needs 5-6 steps to optimize a water molecule.
3) Approximating the inverse hessian by a multiple of the unit matrix is equivalent to changing the \( \lambda \) value.
4) Eigenvalues and eigenvectors of \( \left( \mathbf{H}^{-1} \right)_{\text{approx}} \mathbf{H} \) will govern the convergence: the condition number can be changed. \( \left( \mathbf{H}^{-1} \right)_{\text{approx}} \) is 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.)
Modifying the condition number (III)

The approximate Hessian can be generated on a case-by-case basis.

Selfconsistent determination of the Kohn-Sham potential:

The Hessian matrix is the dielectric matrix.
Lowest eigenvalue close to 1 usually. Largest eigenvalue for small close-shell molecules, and small unit cell solids, is 1.5 ... 2.5.
(Simple mixing will sometimes converge with parameter set to 1 !)
For larger close-shell molecules and large unit cell insulators, the largest eigenvalue is on the order of the macroscopic dielectric constant (e.g. 12 for silicon).
For large-unit cell metals, or open-shell molecules the largest eigenvalue might diverge!

Model dielectric matrices are known for rather homogeneous systems. With knowledge of the approximate macroscopic dielectric constant, preconditioners can be made very efficient. Work is still in progress for inhomogeneous systems (e.g. metals/vacuum systems).
Modifying the condition number (V)

The approximate Hessian can be further improved by using the history.

Large class of methods:
- Broyden (quasi-Newton-type),
- Davidson,
- conjugate gradients,
- Lanczos ...
(although the three latter methods are not often presented in this way !).
In ABINIT, in practice ...

One needs to select an “history” algorithm
(for SCF, default is Pulay ;
for geometry optimization, no default ;
for Schrödinger equation, no other choice than conjugate-gradients)

One needs to select a pre-conditioner
(for SCF, default is Kerker, if non-metallic, it is better to provide the macroscopic dielectric constant ;
for geometry optimization, no preconditioner ;
for Schrödinger equation, no other choice than inverse kinetic operator)

One needs to provide stopping criteria !!
Also, a maximum number of iterations ... (again, for Schrödinger equation, these parameters are not visible)
ABINIT tutorials

ABINIT Web site : http://www.abinit.org
Tutorial page :
http://www.abinit.org/Infos_v5.3/tutorial/welcome.html
19 lessons (from 30 minutes to 2h work) ...
4 basic lessons, then independent lessons on :
GW (2), phonons/dielectric constant (2), PAW (2),
spin, TDDFT, optics, elastic constants, electron-phonon,
nonlinear properties, parallelism, ...

The goal of the hands-on in Bristol :
a glimpse on these lessons (only lessons 1 and 3).

⇒ Convergence studies brought to a minimum !!
(see lesson 2)