Iterative techniques

X. Gonze, U.C.Louvain

1. Computing the forces
2. Analysis of simple iterative algorithms
3. Advanced iterative algorithms

Computing the forces

Computing the forces (I)

Born - Oppenheimer approximation ⇨ one finds the electronic ground state in the potential created by the nuclei.

Consider a given configuration of nuclei $\{ \mathrm{R}_\kappa \}$.
Usually it is NOT the equilibrium geometry.

$$\mathrm{F}_{\kappa,\alpha} = -\left.\frac{\partial E}{\partial R_{\kappa,\alpha}}\right|_{\{\vec{\mathrm{R}}_\kappa\}}$$ (principle of virtual works)

Forces can be computed by finite differences.
Better approach : compute the response to a perturbation

$$\{\mathrm{R}_{\kappa,\alpha}\} \rightarrow \{\mathrm{R}_{\kappa,\alpha} + \lambda\delta\mathrm{R}_{\kappa,\alpha}\}$$

Small parameter

⇨ What is the energy change ?

Computing the forces (II)

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

Perturbation theory : Hellmann - Feynman theorem

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

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

Application to the derivative with respect to an atomic displacement :

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

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

Computing the forces (III)

Generalisation to density functional theory

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

If change of atomic positions ...

$$V_{ext}(\vec{r}) = \sum_{k'} -\frac{Z_{k'}}{|\vec{r} - \vec{R}_{k'}|}$$ (can be generalized to pseudopotential case)

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

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

Computing the forces (IV)

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

Forces can be computed directly from the density !
No need for additional work (like solving the Kohn-Sham eq. or self-consistency) - at variance with finite-difference

Pseudopotentials instead of Coulomb potential
⇨ additional term, involving also wavefunctions ...
... Needed for PW/PP approach

If basis set depends on the atomic positions, and not complete
⇨ additional term (Pulay correction, or IBSC, incomplete basis set correction)
... Not needed for PW/PP approach

Analysis of simple iterative algorithms

Algorithmics : problems to be solved

(1) Kohn - Sham equation

$$\underline{\underline{A}}\ \underline{x}_i = \lambda_i \underline{x}_i \qquad \{\mathbf{G}_j\} \qquad \{\mathbf{r}_j\}$$

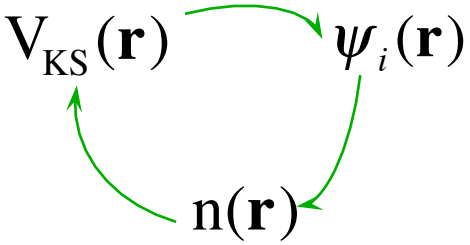
$$\left[-\frac{1}{2}\nabla^2 + V_{KS}(\mathbf{r})\right]\psi_i(\mathbf{r}) = \varepsilon_i\,\psi_i(\mathbf{r})$$

Size of the system [2 atoms… 600 atoms…] + vacuum ?

Dimension of the vectors $\underline{x}_i$ 300… 100 000… (if planewaves)

of (occupied) eigenvectors 4… 1200…

(2) Self-consistency



(3) Geometry optimization

Find the positions $\{\mathbf{R}_\kappa\}$ of ions such that the forces $\{\mathbf{F}_\kappa\}$ 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 λ ,
and iterately update the geometry, following the forces :

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

Equivalent to the simple mixing algorithm of SCF (see later)

Energy+forces around equilib. geometry

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

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

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

Moreover, $F_{\kappa,\alpha} = -\frac{\partial E^{BO}}{\partial R_{\kappa,\alpha}}$ $\qquad \frac{\partial F_{\kappa',\alpha'}}{\partial R_{\kappa,\alpha}} = -\frac{\partial^2 E^{BO}}{\partial R_{\kappa,\alpha}\partial R_{\kappa',\alpha'}}$

Vector and matrix notation

$R^*_{\kappa,\alpha} \rightarrow \underline{R}^*$ $\qquad$ $R_{\kappa,\alpha} \rightarrow \underline{R}$ $\qquad$ $F_{\kappa,\alpha} \rightarrow \underline{F}$ $\qquad$ $\left.\frac{\partial^2 E^{BO}}{\partial R_{\kappa,\alpha}\partial R_{\kappa',\alpha'}}\right|_{\{R^*_{\kappa,\alpha}\}} \rightarrow \underline{\underline{H}}$ (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 :

$$\underline{F}(\underline{R}) = \underline{F}(\underline{R}^*) - \underline{\underline{H}}\left(\underline{R} - \underline{R}^*\right) + O\left(\underline{R} - \underline{R}^*\right)^2$$

$$\underline{R}^{(n+1)} = \underline{R}^{(n)} + \lambda \underline{F}^{(n)} \Rightarrow \left(\underline{R}^{(n+1)} - \underline{R}^*\right) = \left(\underline{R}^{(n)} - \underline{R}^*\right) - \lambda \underline{\underline{H}}\left(\underline{R}^{(n)} - \underline{R}^*\right)$$

$$\left(\underline{R}^{(n+1)} - \underline{R}^*\right) = \left(\underline{\underline{1}} - \lambda \underline{\underline{H}}\right)\left(\underline{R}^{(n)} - \underline{R}^*\right)$$

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) $\underline{\underline{1}} - \lambda \underline{\underline{H}}$

Steepest-descent : analysis (II)

What are the eigenvectors and eigenvalues of $\underline{\underline{H}}$?

$\underline{\underline{H}}$ symmetric,
positive definite matrix $\left(= \left. \frac{\partial^2 E^{BO}}{\partial R_{\kappa,\alpha} \partial R_{\kappa',\alpha'}} \right|_{\{R^*_{\kappa,\alpha}\}} \right)$

$\underline{\underline{H}}\,\underline{f}_i = h_i\,\underline{f}_i$ where $\{\underline{f}_i\}$ form a complete, orthonormal, basis set

Discrepancy decomposed as $\left(\underline{R}^{(n)} - \underline{R}^*\right) = \sum_i c_i^{(n)}\,\underline{f}_i$

and $\left(\underline{R}^{(n+1)} - \underline{R}^*\right) = \left(\underline{\underline{1}} - \lambda\underline{\underline{H}}\right)\sum_i c_i^{(n)}\,\underline{f}_i = \sum_i c_i^{(n)}\left(1 - \lambda h_i\right)\underline{f}_i$

The coefficient of $\underline{f}_i$ is multiplied by 1- $\lambda\, h_i$

Iteratively : $\boxed{\left(\underline{R}^{(n)} - \underline{R}^*\right) = \sum_i c_i^{(0)}\left(1 - \lambda h_i\right)^{(n)}\underline{f}_i}$

Steepest-descent : analysis (III)

$$\left(\underline{R}^{(n)} - \underline{R}^{*}\right) = \sum_{i} c_i^{(0)} \left(1 - \lambda h_i\right)^{(n)} \underline{f}_i$$

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 ?

$\underline{\underline{H}}$ positive definite => all h_i are positive

Yes ! If λ positive, sufficiently small ...

Steepest-descent : analysis (IV)

$$\left(\underline{R}^{(n)} - \underline{R}^{*}\right) = \sum_i c_i^{(0)} \left(1 - \lambda h_i\right)^{(n)} \underline{f}_i$$

How to determine the optimal value of λ ?

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

As an exercise : suppose $h_1 = 0.2$, $h_2 = 1.0$, $h_3 = 5.0$ => what is the best value of λ ?

\+ what is the convergence rate ?

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

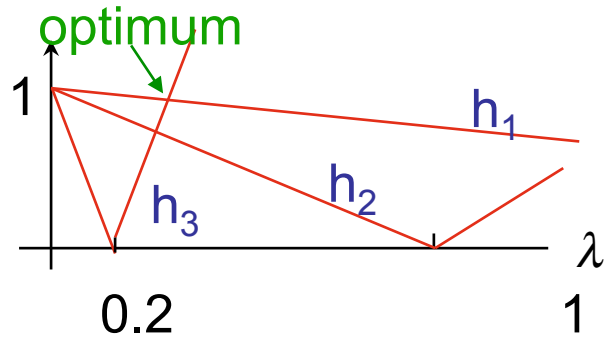
Steepest-descent : analysis (V)

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

$h_1 = 0.2$ $\quad |1- \lambda .0.2|$ $\quad$ optimum => $\lambda = 5$

$h_2 = 1.0$ $\quad |1- \lambda . 1 |$ $\quad$ optimum => $\lambda = 1$ } ?

$h_3 = 5.0$ $\quad |1- \lambda . 5 |$ $\quad$ optimum => $\lambda = 0.2$



$\mu = |1- \lambda 0.2| = |1- \lambda 5|$

positive negative

$1- \lambda . 0.2 = -(1- \lambda .5)$

$2 - \lambda (0.2+5)=0 \Rightarrow \lambda = 2/5.2$

$\mu = 1 - 2. (0.2 / 5.2)$

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, $\lambda_{opt} = 2 / (h_{min} + h_{max})$

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

Perfect if $h_{max} = h_{min}$. Bad if $h_{max} >> h_{min}$.
h_{max}/h_{min} called the "condition" number. A problem is "ill-conditioned" if the condition number is large. It does not depend on the intermediate eigenvalues.

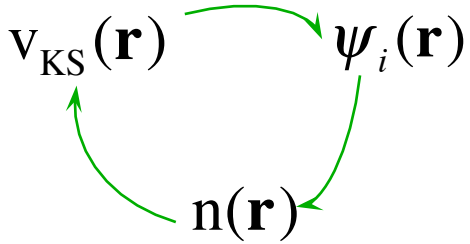
Suppose we start from a configuration with forces on the order of 1 Ha/Bohr, and we want to reach the target 1e-4 Ha/Bohr. The mixing parameter is optimal.
How many iterations are needed ?
For a generic decrease factor Δ , with "n" the number of iterations.

$$\left\|\underline{F}^{(n)}\right\| \approx \left(\frac{h_{max}/h_{min}-1}{h_{max}/h_{min}+1}\right)^{n} \left\|\underline{F}^{(0)}\right\| \qquad \Delta \approx \left(\frac{h_{max}/h_{min}-1}{h_{max}/h_{min}+1}\right)^{n}$$

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

(The latter approximate equality suppose a large condition number)

Analysis of self-consistency



Natural iterative methodology (KS : in => 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
(≈ steepest - descent)

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

Analysis ... $\underline{v}_{out}\left[\underline{v}_{in}\right] = \underline{v}_{out}\left[\underline{v}^{*}\right] + \dfrac{\delta \underline{v}_{out}}{\delta \underline{v}_{in}}\left(\underline{v}_{in} - \underline{v}^{*}\right)$

$\dfrac{\delta \underline{v}_{out}}{\delta \underline{v}_{in}} \rightarrow \underline{\underline{H}}$

Like the steepest-descent algorithm, this leads to the requirement to minimize $|1- \lambda\ h_i|$ where h_i are eigenvalues of $\dfrac{\delta \underline{v}_{out}}{\delta \underline{v}_{in}}$

(actually, the dielectric matrix)

Variational (minimum) principle for the eigenfunctions/eigenvalues

$$\left[-\frac{1}{2}\nabla^2 + v_{KS}(\vec{r})\right]\psi_i(\vec{r}) = \varepsilon_i\,\psi_i(\vec{r})$$

Or, using Dirac notations : $\hat{H}\left|\psi_i\right\rangle = \varepsilon_i\left|\psi_i\right\rangle$

Lowest eigenfunction can be found by a minimum principle for the expectation value of the eigenenergy

under constraint $\left\langle \psi_0^T \middle| \psi_0^T \right\rangle = 1$ (*T* stands for "trial") $\min \left\langle \psi_0^T \middle| \hat{H} \middle| \psi_0^T \right\rangle = \varepsilon_0^T$

Lowest eigenvectors can be found by a minimum principle for the expectation value of the sum of eigenenergies

$$\min \sum_i^N \left\langle \psi_i^T \middle| \hat{H} \middle| \psi_i^T \right\rangle = \sum_i^N \varepsilon_i^T$$ under constraints $\left\langle \psi_i^T \middle| \psi_j^T \right\rangle = \delta_{ij}$

Finding eigenfunctions

$$\hat{H}|\psi_i\rangle = \varepsilon_i|\psi_i\rangle$$

Label the lowest energy state as i=0,
then states with increasing energy, i = 1, i = 2, i = 3 …

1) Direct methods : in finite basis (planewaves, mesh points, local orbitals ...)
an Hamiltonian is nothing else than a matrix, that can be treated by direct methods developed by numericians (e.g. Choleski-Householder)

- H matrix N_{basis} x N_{basis}
- T_{CPU} scales as N^3_{basis}
- Deliver all eigenvalues and eigenvectors i = 0 … (N_{basis} -1)

2) Iterative techniques : better adapted to electronic structure !

- Might focus only on the lowest eigenstates (= # bands N_{bd})
- T_{CPU} scales as N_{bd} * T ($\hat{H}|\varphi\rangle$)
  In particular, for plane waves, or a discretized representation on a grid, the CPU time needed to apply the Hamiltonian to a wavefunction does not scale like a matrix-vector product N_{basis} x N_{basis}, but like N_{basis} or N_{basis} log N_{basis}

The power method

Suppose we want to find the eigenvector associated with the largest eigenvalue of $\hat{\mathrm{H}}\left|\psi_i\right\rangle = \varepsilon_i \left|\psi_i\right\rangle$

We start from a trial vector $\left|\psi^T\right\rangle$, and apply iteratively the operator.
At the end (or during the procedure, to avoid divergence),
we renormalize the vector.

Analysis : $\left|\psi^T\right\rangle = \sum_i c_i^{(0)} \left|\psi_i\right\rangle$

$$\left|\psi^{(1)}\right\rangle = \hat{\mathrm{H}}\left|\psi^T\right\rangle = \sum_i c_i^{(0)} \varepsilon_i \left|\psi_i\right\rangle$$

$$\left|\psi^{(2)}\right\rangle = \hat{\mathrm{H}}^2\left|\psi^T\right\rangle = \sum_i c_i^{(0)} \varepsilon_i^2 \left|\psi_i\right\rangle$$

...

The component with the largest $\left|\varepsilon_{\mathrm{i}}\right|$ increases the most rapidly.

After renormalisation, $\lim_{n\to\infty}\left|\psi^{(n)}\right\rangle = \sum_i \left| c_i^{(0)} \frac{\varepsilon_i^n}{\varepsilon_{\max}^n} \right| \left|\psi_i\right\rangle$

Only the component with maximal eigenvalue will survive !

Advanced iterative algorithms

How to do a better job ?

1) Take advantage of the history

Simple mixing
Steepest descent
Power + shift

very primitive algorithms $\underline{v}^{(n)} \rightarrow \underline{v}^{(n+1)}$

The information about previous iterations is completely ignored

We already know several vector/residual pairs $\left(\underline{v}^{(n)}, \underline{r}^{(n)}\right)$
We should try to use them !

$\underline{R} \rightarrow \underline{F}$ $\qquad$ $\underline{v}_{in} \rightarrow \underline{v}_{out} - \underline{v}_{in}$ $\qquad$ $\left|\psi^T\right\rangle \rightarrow \begin{cases} \langle\varepsilon\rangle = \left\langle\psi^T\right|\hat{H}\left|\psi^T\right\rangle \\ \left|R\right\rangle = \left(\hat{H} - \langle\varepsilon\rangle\right)\left|\psi^T\right\rangle \end{cases}$

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

2) Decrease the condition number

Minimisation of the residual (I)

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

Find the best $\underline{v}$ that can be obtained by combining the latest one with its differences with other $\underline{v}^{(p)}$.

Equivalent to $\underline{v} = \sum_{p=1}^{n} s_p \underline{v}^{(p)}$ with $1 = \sum_{p=1}^{n} s_p$ (like a normalisation)

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

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

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

=> New residual is a linear combination of the old ones, with same coefficients as those of the potential.

Excellent strategy : select the s_p such as to minimize the norm of residual (RMM =residual minimisation method - Pulay).

Then mix 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.

(D.G. Anderson, J. Assoc. Comput. Mach. 12, 547 (1964))

Modify the condition number (I)

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

$$\underline{F}(\underline{R}) = -\underline{\underline{H}}\left(\underline{R} - \underline{R}^{*}\right)$$

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

giving $\left(\underline{R}^{(n+1)} - \underline{R}^{*}\right) = \left(\underline{\underline{1}} - \lambda \underline{\underline{H}}\right)\left(\underline{R}^{(n)} - \underline{R}^{*}\right)$

Now, suppose an approximate inverse Hessian $\left(\underline{\underline{H}}^{-1}\right)_{\text{approx}}$

Then, applying $\left(\underline{\underline{H}}^{-1}\right)_{\text{approx}}$ on the forces, and moving the nuclei

along these modified forces gives $\underline{R}^{(n+1)} = \underline{R}^{(n)} + \lambda \left(\underline{\underline{H}}^{-1}\right)_{\text{approx}} \underline{F}^{(n)}$

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

$$\left(\underline{R}^{(n+1)} - \underline{R}^{*}\right) = \left(1 - \lambda \left(\underline{\underline{H}}^{-1}\right)_{\text{approx}} \underline{\underline{H}}\right)\left(\underline{R}^{(n)} - \underline{R}^{*}\right)$$

Modify the condition number (II)

$$\underline{R}^{(n+1)} = \underline{R}^{(n)} + \lambda \left(\underline{\underline{H}}^{-1} \right)_{approx} \underline{F}^{(n)}$$

$$\underline{F}(\underline{R}) = -\underline{\underline{H}} \left(\underline{R} - \underline{R}^{*} \right) \Longrightarrow \quad \left(\underline{R}^{(n+1)} - \underline{R}^{*} \right) = \left(1 - \lambda \left(\underline{\underline{H}}^{-1} \right)_{approx} \underline{\underline{H}} \right) \left(\underline{R}^{(n)} - \underline{R}^{*} \right)$$

Notes : 1) If approximate inverse Hessian perfect, optimal geometry is reached in one step, with λ =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 λ value.

4) Eigenvalues and eigenvectors of $\left(\underline{\underline{H}}^{-1} \right)_{approx} \underline{\underline{H}}$ govern the convergence : the condition number can be changed. $\left(\underline{\underline{H}}^{-1} \right)_{approx}$ 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 ... 2.5 for small close-shell molecules, and small unit cell solids
(Simple mixing will sometimes converge with parameter set to 1 !)
= the macroscopic dielectric constant (e.g. 12 for silicon),
forlarger close-shell molecules and large unit cell insulators,
= diverge for large-unit cell metals, or open-shell molecules !

Model dielectric matrices known for rather homogeneous systems.
Knowledge of approx. macroscopic dielectric constant
=> efficient preconditioner
Work in progress for inhomogeneous systems
(e.g. metals/vacuum systems).

The history

Approximate Hessian can be 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 !).