Optimization techniques

Mathieu Salanne

Laboratoire PECSA
Université Pierre et Marie Curie / CNRS / ESPCI
www.pecsa.upmc.fr

CPMD–CP2K tutorial – 07/04/2010

Objective: solve the Kohn-Sham equations

Ground-state energy found by minimizing the functional $E^{KS}[\{\phi_i\}]$ w.r.t. the electronic degrees of freedom:

$$\psi = \min_{\{\phi_i\}} E^{KS}[\{\phi_i\}]$$

under the orbital orthonormality constraint:

$$\langle \phi_i \mid \phi_j \rangle = \delta_{ij}$$

We have the equivalent set of linear equations

$$H_e^{KS}\phi_i = \sum_j \Lambda_{ij}\phi_j (= \epsilon_i \phi_i)$$

where the Λ_{ij} are Lagrange multipliers

Plane wave basis set

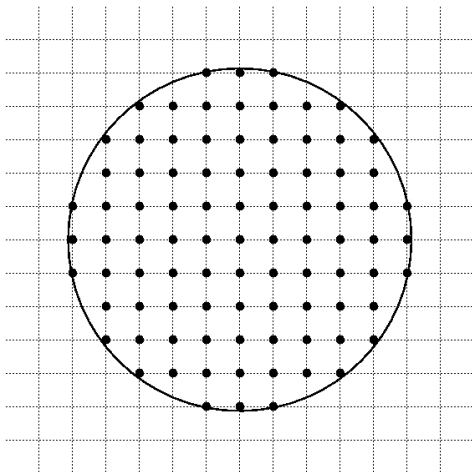
Expand Kohn-Sham orbitals $\phi_i(\mathbf{r})$ in plane waves

$$\phi_i(\mathbf{r}) = \frac{1}{\sqrt{\Omega}} \sum_{\mathbf{G}} c_i(\mathbf{G}) \exp[\imath \mathbf{G} \cdot \mathbf{r}]$$

where $c_i(\mathbf{G})$ are complex numbers

- Ω: Volume of the simulation cell
- Definition of a cutoff energy; $\frac{1}{2}G^2 \leq E_{cut}$
  $\mathbf{c}_i = \{c_i(\mathbf{G})\}_{\mathbf{G}}$
- Number of plane waves:

$$N_{PW} \approx \frac{1}{2\pi^2} \Omega E_{cut}^{3/2} [a.u.]$$



Fix point methods

1. Initial guess $n^{\mathrm{in}}(r)$
2. Calculate potential $V(r)$
3. Diagonalize KS matrix, get c^{out}
4. Calculate new density n^{out}
5. If $\mid n^{\mathrm{in}} - n^{\mathrm{out}} \mid \leq \epsilon$ stop
6. Calculate new density from n^{in} and n^{out} (mixing)
7. Go back to 2

Outline

Outline

Steepest descent

Suppose that the function to minimize is roughly approximated by a multidimensional quadratic form around some point $\mathbf{P}$:

$$f(\mathbf{x}) \approx c - \mathbf{b}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbb{A}\mathbf{x}$$

where

$$\begin{aligned} c &= f(\mathbf{P}) \\ b_i &= -\frac{\partial f}{\partial x_i}\,|_{\mathbf{P}} \\ A_{ij} &= \frac{\partial^2 f}{\partial x_i \partial x_j}\,|_{\mathbf{P}} \end{aligned}$$

with a symmetric positive-definite Hessian matrix $\mathbb{A}$. The problem is equivalent to solving

$$\mathbb{A}\mathbf{x} = \mathbf{b}$$

An iterative minimization procedure is defined by the sequence

$$\mathbf{P}^{(n+1)} = \mathbf{P}^{(n)} + \lambda^{(n)}\mathbf{g}^{(n)}$$

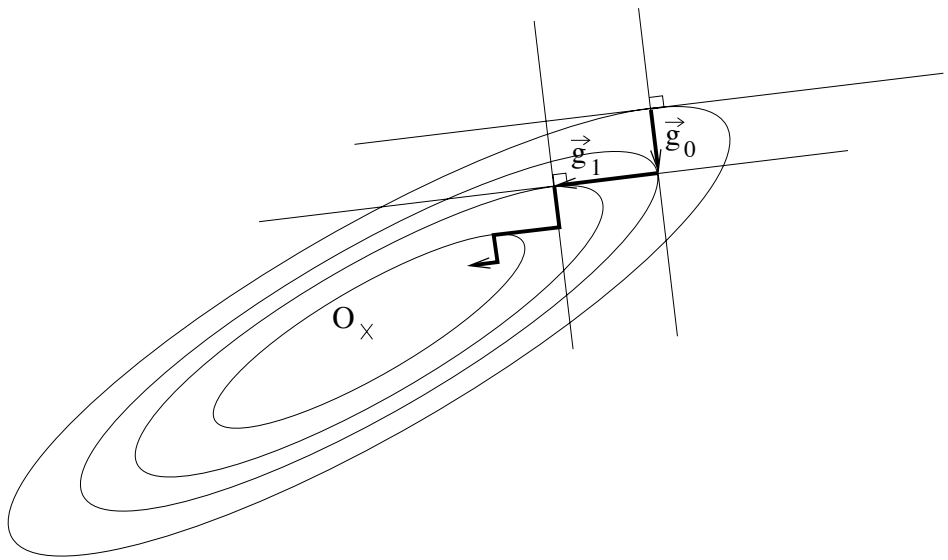
where

- $\mathbf{g}^{(n)}$ is the search direction. In steepest descent schemes one chooses

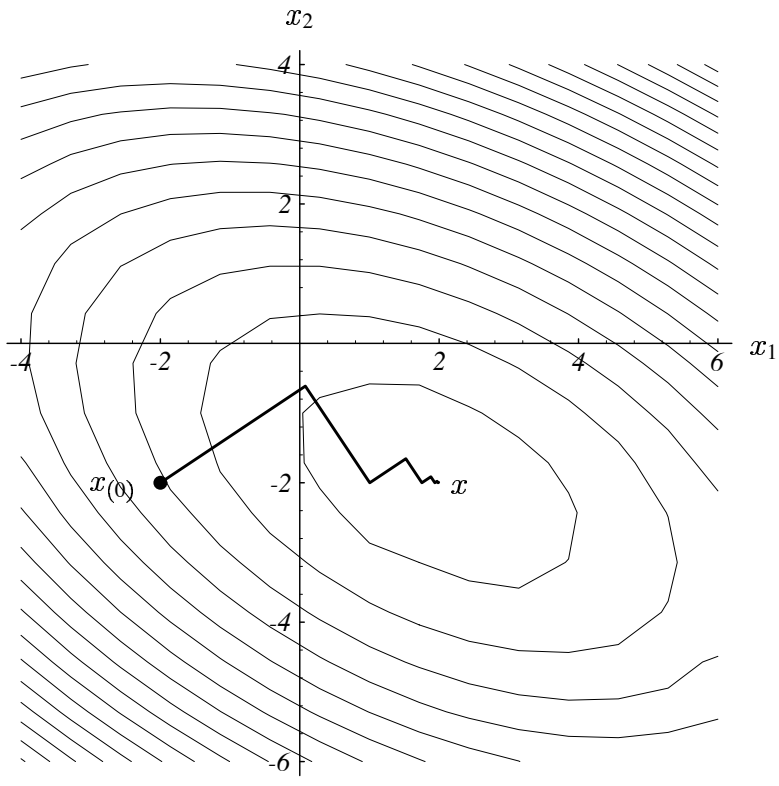
$$\mathbf{g}^{(n)} = -\boldsymbol{\nabla} f \mid_{\mathbf{P}^{(n)}}$$

- $\lambda^{(n)}$ is a scalar

$\lambda^{(n)}$ is generally chosen so that f is minimized along $\mathbf{g}^{(n)}$ direction. This is achieved when $\mathbf{g}^{(n)}$ and $\boldsymbol{\nabla} f \,|_{\mathbf{P}}^{(n+1)} = \mathbf{g}^{n+1}$ are orthogonal

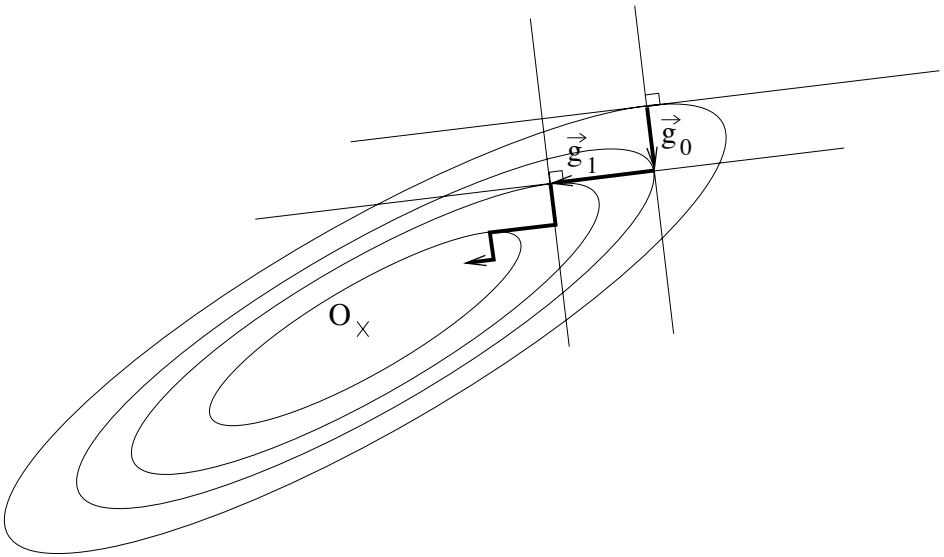


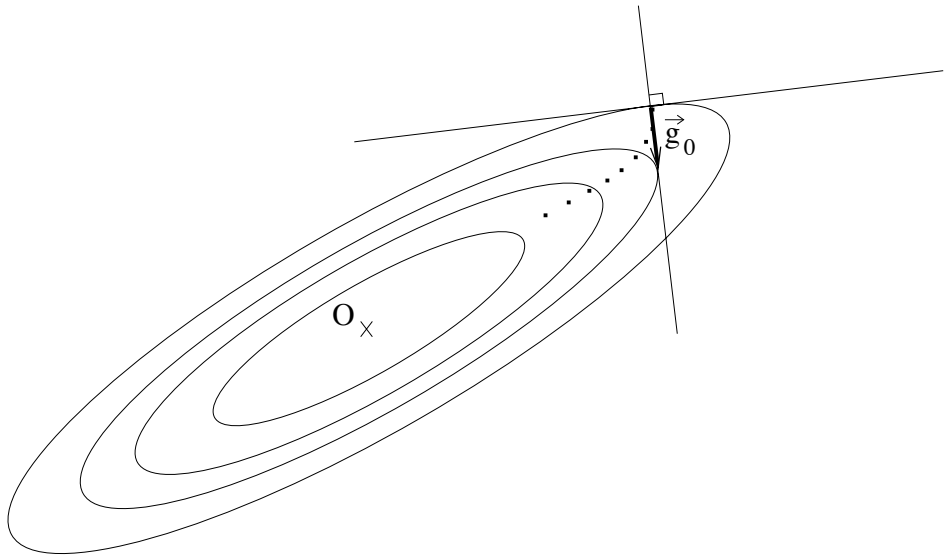
Example (starting point: $\begin{bmatrix} -2 \\ -2 \end{bmatrix}$; minimum: $\begin{bmatrix} 2 \\ -2 \end{bmatrix}$)



6 steps are necessary to find the minimum

Modified steepest descent (CPMD)





- No calculation of $\lambda^{(n)} \longrightarrow$ small value chosen
- Need for more iterations

Conjugate gradients

- Problem of steepest descent: Search along two directions only,

$$\mathbf{g}^{(n+1)T}\mathbf{g}^{(n)} = 0, \forall n$$

$\longrightarrow$ many steps necessary

- Conjugate gradients : search only once in a given direction. Search directions $\mathbf{h}^n$ are now given by:

$$\begin{aligned} \mathbf{h}^{(n)} &= \mathbf{g}^{(n)}, n = 0 \\ &= \mathbf{g}^{(n)} + \gamma^{(n-1)}\mathbf{h}^{(n-1)}, n > 0 \end{aligned}$$

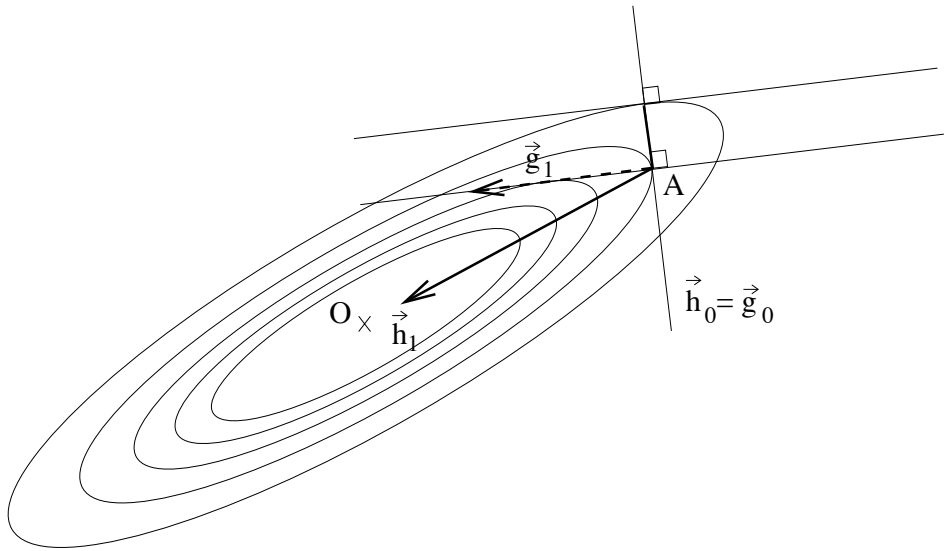
where

$$\gamma^{(n)} = \frac{\mathbf{g}^{(n+1)T}\mathbf{g}^{(n+1)}}{\mathbf{g}^{(n)T}\mathbf{g}^{(n)}}$$

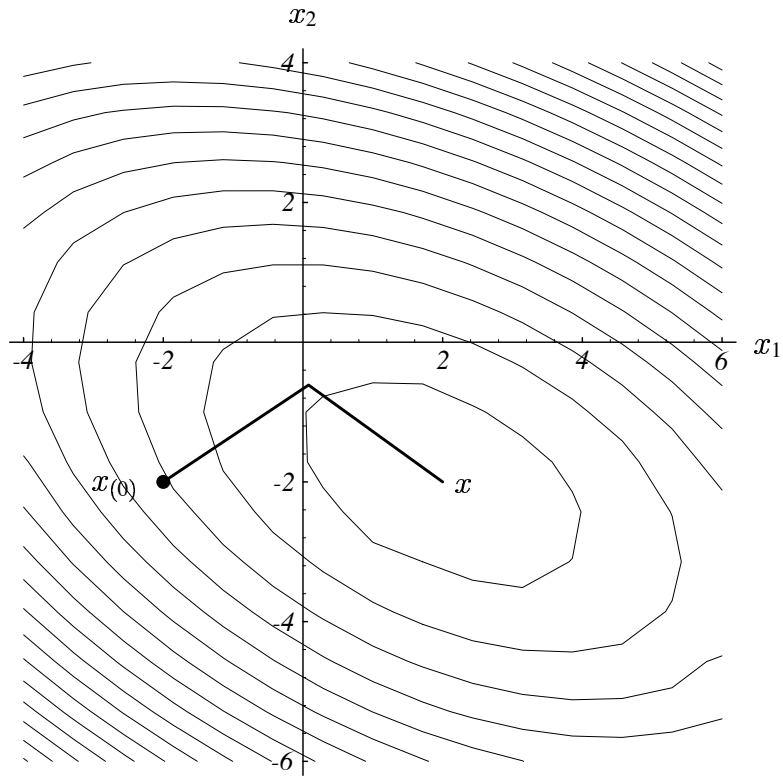
- Search directions $\mathbf{h}^n$ are now $\mathbb{A}$-orthogonal (conjugate), *i.e.*

$$\mathbf{h}^{(i)T}\mathbb{A}\mathbf{h}^{(j)} = 0, \forall i \neq j$$

$\lambda^{(n)}$ is again chosen so that f is minimized along $\mathbf{h}^{(n)}$ direction.



Example (starting point: $\begin{bmatrix} -2 \\ -2 \end{bmatrix}$; minimum: $\begin{bmatrix} 2 \\ -2 \end{bmatrix}$)



2 steps are necessary to find the minimum

Preconditioning

- Condition number associated with a problem is a measure of that problem's amenability to digital computation
- Preconditioning is a technique for improving the condition number of a matrix. Suppose that $\mathbb{M}$ is a symmetric, positive-definite matrix that approximates $\mathbb{A}$, but is easier to invert. We can solve $\mathbb{A}\mathbf{x} = \mathbf{b}$ indirectly by solving

$$\mathbb{M}^{-1}\mathbb{A}\mathbf{x} = \mathbb{M}^{-1}\mathbf{b}$$

- If the eigenvalues of $\mathbb{M}^{-1}\mathbb{A}$ are better clustered than those of $\mathbb{A}$, we can iteratively solve this equation more quickly than the original problem

Iterative minimization procedure is now defined by the sequence

$$\mathbf{P}^{(n+1)} = \mathbf{P}^{(n)} + \lambda^{(n)}\mathbf{h}^{(n)}$$

where search directions are

$$\begin{aligned} \mathbf{h}^{(n)} &= \mathbb{M}^{-1}\mathbf{g}^{(n)}, n = 0 \\ &= \mathbb{M}^{-1}\mathbf{g}^{(n)} + \gamma^{(n-1)}\mathbf{h}^{(n-1)}, n > 0 \end{aligned}$$

where (Fletcher-Reeves variant)

$$\gamma^{(n)} = \frac{\left(\mathbb{M}^{-1}\mathbf{g}^{(n+1)}\right)^T \mathbb{M}^{-1}\mathbf{g}^{(n+1)}}{\left(\mathbb{M}^{-1}\mathbf{g}^{(n)}\right)^T \mathbb{M}^{-1}\mathbf{g}^{(n)}}$$

Outline

Straightforward application of conjugate gradients

The iterative minimization procedure is defined by the sequence

$$\mathbf{c}_i^{(n+1)} = \mathbf{c}_i^{(n)} + \lambda^{(n)} \mathbf{h}_i^{(n)}$$

where the search direction is

$$\begin{aligned} \mathbf{h}_i^{(n)} &= \mathbf{g}_i^{(n)}, n = 0 \\ &= \mathbf{g}_i^{(n)} + \gamma^{(n-1)} \mathbf{h}_i^{(n-1)}, n > 0 \end{aligned}$$

with

$$\begin{aligned} g_i^{(n)}(\mathbf{G}) &= -\frac{\delta E^{KS}}{\delta c_i^{(n)*}(\mathbf{G})} \\ \gamma^{(n)} &= \frac{\sum_i \mathbf{g}_i^{(n+1)T} \mathbf{g}_i^{(n+1)}}{\sum_i \mathbf{g}_i^{(n)T} \mathbf{g}_i^{(n)}} \end{aligned}$$

Preconditioner

Kohn-Sham Hamiltonian $\mathbb{H}^{KS}$ is dominated by kinetic energy term for large $\mathbf{G} \rightarrow \mathbb{H}^{KS}$ is diagonally dominant for large $\mathbf{G} \rightarrow$ diagonal part of $\mathbb{H}^{KS}$ is a good preconditioner and can be inverted.

For small values of $\mathbf{G}$ we can not take the diagonal part of $\mathbb{H}^{KS}$ but should choose some constant value $\rightarrow$ interpolate between the two extremes.

$$\begin{aligned} \mathbb{K}_{\mathbf{G},\mathbf{G}'} &= \mathbb{H}^{KS}_{\mathbf{G},\mathbf{G}}\delta_{\mathbf{G},\mathbf{G}'} \qquad \text{if } \mid \mathbf{G} \mid \geq G_c \\ &= \mathbb{H}^{KS}_{\mathbf{G}_c,\mathbf{G}_c}\delta_{\mathbf{G},\mathbf{G}'} \qquad \text{if } \mid \mathbf{G} \mid \leq G_c \end{aligned}$$

where G_c is a free parameter that can be adjusted to accelerate convergence.

PCG minimize

New iterative minimization procedure:

$$\mathbf{c}_i^{(n+1)} = \mathbf{c}_i^{(n)} + \lambda^{(n)} \mathbf{h}_i^{(n)}$$

where the search direction is

$$\begin{aligned} \mathbf{h}_i^{(n)} &= \mathbb{K}^{-1}\mathbf{g}_i^{(n)}, n = 0 \\ &= \mathbb{K}^{-1}\mathbf{g}_i^{(n)} + \gamma^{(n-1)}\mathbf{h}_i^{(n-1)}, n > 0 \end{aligned}$$

with

$$\begin{aligned} g_i^{(n)}(\mathbf{G}) &= -\frac{\partial E^{KS}}{\partial c_i^{(n)*}(\mathbf{G})} \\ \gamma^{(n)} &= \frac{\sum_i (\mathbb{K}^{-1}\mathbf{g}_i^{(n+1)})^T (\mathbb{K}^{-1}\mathbf{g}_i^{(n+1)})}{\sum_i (\mathbb{K}^{-1}\mathbf{g}_i^{(n)})^T (\mathbb{K}^{-1}\mathbf{g}_i^{(n)})} \end{aligned}$$

Orthonormalization

We now note $\tilde{\mathbf{c}}_i^{(n)}$ the vectors obtained by the CG procedure. They do not satisfy orthonormality condition:

$$\tilde{\mathbf{c}}_i^T \tilde{\mathbf{c}}_j = \sum_{\mathbf{G}} \tilde{c}_i^*(\mathbf{G}) \tilde{c}_j(\mathbf{G}) \neq \delta_{ij}$$

Orthonormalization can be performed through a Gram-Schmidt procedure:

$$\begin{aligned} \mathbf{c}'_i &= \tilde{\mathbf{c}}_i - \sum_{j<i} (\mathbf{c}_j^T \tilde{\mathbf{c}}_i) \mathbf{c}_j \\ \mathbf{c}_i &= \frac{\mathbf{c}'_i}{| \mathbf{c}'_i |} \end{aligned}$$

Outline

DIIS

Extrapolation method which uses the information of n previous steps

Assume we have generated a sequence of n parameter vectors $\{\mathbf{c}_i^{(k)}\}_{k\in[1,n]}$ and that we are able to guess for each of these n vectors its difference $\mathbf{e}_i^{(k)}$ to the stationary point $\mathbf{c}_i$

$$\mathbf{e}_i^{(k)} = \mathbf{c}_i^{(k)} - \mathbf{c}_i$$

Ansatz: find the best linear combination of vectors $\mathbf{c}_i^{(k)}$

$$\mathbf{c}_i^{(n+1)} = \sum_{k=1}^{n} d_k \mathbf{c}_i^{(k)}$$

In the ideal case this would be

$$\begin{aligned}
\sum_{k=1}^{n} d_k \mathbf{c}_i^{(k)} &= \mathbf{c}_i \\
\sum_{k=1}^{n} d_k (\mathbf{c}_i + \mathbf{e}_i^{(k)}) &= \mathbf{c}_i \\
\sum_{k=1}^{n} d_k \mathbf{c}_i + \sum_{k=1}^{n} d_k \mathbf{e}_i^{(k)} &= \mathbf{c}_i
\end{aligned}$$

The last line can be fulfilled by setting

$$\begin{aligned}
\sum_{k=1}^{n} d_k &= 1 \\
\sum_{k=1}^{n} d_k \mathbf{e}_i^{(k)} &= 0
\end{aligned}$$

We can fulfill the last condition only approximately

Objective: minimize the norm of $\sum_{k=1}^{n} d_k \mathbf{e}_i^{(k)}$ with $\sum_{k=1}^{n} d_k = 1$

This leads to a system of linear equations from which the $\{d_k\}_{k\in[1,n]}$ can be calculated:

$$\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} & -1 \\ b_{21} & b_{22} & \cdots & b_{2n} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} & -1 \\ -1 & -1 & \cdots & -1 & 0 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix}$$

where $b_{kl} = \sum_i \mathbf{e}_i^{(k)T} \mathbf{e}_i^{(l)}$

Choice of the error vectors (GDIIS):

$$\mathbf{e}_i^{(k)} = -\mathbb{K}^{-1}\mathbf{g}_i^{(k)}$$

where $\mathbb{K}^{-1}$ is the same preconditionner as for PCG method

From the set of $\{d_k\}_{k\in[1,n]}$ we can determine the new $\mathbf{c}_i^{(n+1)}$ and $\mathbf{e}_i^{(n+1)}$, and thus a new estimate for $\tilde{\mathbf{c}}_i$,

$$\tilde{\mathbf{c}}_i \quad = \quad \mathbf{c}_i^{(n+1)} - \mathbf{e}_i^{(n+1)}$$

Again the set of $\tilde{\mathbf{c}}_i$ has to be orthonormalised

Outline

Simulated annealing

In fact the SD equations

$$\mathbf{c}_i^{(n+1)} = \mathbf{c}_i^{(n)} - \lambda^{(n)} \left(\mathbb{H}^{KS} \mathbf{c}_i^{(n)} - \sum_j \Lambda_{ij} \mathbf{c}_j^{(n)} \right)$$

can be thought of as the discrete version of the first-order differential equation

$$\dot{\mathbf{c}}_i = -\mathbb{H}^{KS} \mathbf{c}_i + \sum_j \Lambda_{ij} \mathbf{c}_j$$

with $\lambda^{(n)}$ taken as a time-like parameter

Alternative to direct minimization: perform an annealing procedure, which corresponds to a damped second order dynamics:

$$\mu \ddot{\mathbf{c}}_i + \eta \dot{\mathbf{c}}_i = -\mathbb{H}^{KS} \mathbf{c}_i + \sum_j \Lambda_{ij} \mathbf{c}_j$$

with μ a mass-like coefficient and $\eta > 0$ a friction coefficient ensuring that the energy always decreases during the dynamical evolution of the orbitals

This second set of differential equations can be integrated numerically using the following discretized algorithm:

$$\begin{aligned} \mathbf{c}_i^{(n+1)} &= \frac{1}{1+\tilde{\Delta}}\left(2\mathbf{c}_i^{(n)} - (1-\tilde{\Delta})\mathbf{c}_i^{(n-1)}\right) \\ &\quad -\frac{1}{1+\tilde{\Delta}}\frac{\Delta^2}{\mu}\left(\mathbb{H}^{KS}\mathbf{c}_i^{(n)} - \sum_j \Lambda_{ij}\mathbf{c}_j^{(n)}\right) \end{aligned} \tag{1}$$

with $\tilde{\Delta} = \eta\Delta/2\mu$, Δ is the time step

- Relaxation time $\tau = \mu/\eta$
- The SD algorithm is recovered for $\tilde{\Delta} = 1$
- It is possible to dynamically adjust the friction coefficient η

Efficiency equivalent to conjugate gradients method

Outline

Direct minimization methods

1. Initial guess c^{in}
2. Calculate new density
3. Calculate potential $V(r)$
4. Calculate gradient g
5. If $\mid g \mid^2 \leq \epsilon$ stop
6. Calculate new c (conjugate gradients, DIIS)
7. Orthonormalize c
8. Go back to 2

Memory requirements

- Conjugate gradients: 2 wavefunctions; 1 gradient
- DIIS: m wavefunctions; m gradients, where m is typically 4–10
- Old gradients are only used as $\mathbb{K}^{-1}\mathbf{g}$. $\mathbb{K}$ is only very approximate $\rightarrow$ only store approximate gradient. In CPMD gradients are stored as 8 bit (instead of 64 bit).
- "Converged" wavefunctions are accurate to ≈ 6 digits. Store additional wavefunctions only at 32 bit (also used to reduce file size).

Other minimization problem: geometry optimization

- Nuclear gradient
- Quasi-Newton methods
- DIIS
- Simulated annealing

Further reading: books

- Dominik Marx and Jürg Hutter,
  Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods
  Cambridge University Press, Cambridge 2009
- Jorge Kohanoff,
  Electronic Structure Calculations for Solids and Molecules
  Cambridge University Press, Cambridge 2006
- Richard M. Martin,
  Electronic Structure: Basic Theory and Practical Methods
  Cambridge University Press, Cambridge 2004

Further reading: articles

- P. Pulay, *J. Comput. Chem.* **3** (4), 556–560 (1982)
- M.J. Gillan, *J. Phys.: Condens. Matter* **1**, 689–711 (1988)
- I. Stich, R. Car, M. Parrinello & S. baroni, *Phys. Rev. B* **39** (8), 4997–5004 (1989)
- D.K. Remler & P.A. Madden, *Mol. Phys.* **70** (6), 921–966 (1990)
- M.C. Payne, M.P. Teter, D.C. Allan, T.A. Arias & J.D. Joannopoulos, *Rev. Mod. Phys.* **64** (4), 1045–1097 (1992)
- J. Hutter, H.P. Lüthi & M. Parrinello, *Comput. Mat. Sci.* **2**, 244–248 (1994)
- F. Tassone, F. Mauri & R. Car, *Phys. Rev. B* **50** (15), 10561–10573 (1994)