Lecture on First-principles Computations (15): Algorithms for solving self-consistency & optimization problems.

任新国 (Xinguo Ren)

中国科学技术大学
量子信息重点实验室
Key Laboratory of Quantum Information, USTC

Hefei, 2018.11.07

Solving the Kohn-Sham equation iteratively

i: iteration step

ϵ: convergence threshold

$0 < \alpha < 1$: mixing parameter

$$n^{(1)}(\boldsymbol{r}) = n_{trial}(\boldsymbol{r})$$

$$V_{eff}^{(i)}(\boldsymbol{r}) = V_{ext}(\boldsymbol{r}) + \int d^3\, r' \frac{n^{(i)}(\mathbf{r}')}{|\, \boldsymbol{r} - \boldsymbol{r}'\, |} + V_{xc}[n^{(i)}](\boldsymbol{r})$$

$$\left(-\frac{\nabla^2}{2\mathrm{m}} + V_{eff}^{(i)}(\boldsymbol{r})\right)\psi_l^{(i)}(\boldsymbol{r}) = \epsilon_l \psi_l^{(i)}(\boldsymbol{r})$$

$$n_{out}^{(i)}(\mathbf{r}) = \sum_{l=1}^{N} \left|\psi_l^{(i)}(\boldsymbol{r})\right|^2$$

$$\int \left|n_{out}^{(i)}(\mathbf{r}) - n^{(i)}(\boldsymbol{r})\right| dr < \epsilon ?$$

no

$$n^{(i+1)} = \alpha n_{out}^{(i)} + (1-\alpha) n^{(i)}$$

yes

exit

Simple linear mixing here;
More sophisticated mixing scheme exists.

How to determine the optimal n_{in}?

The mathematical problem can be formulated as:

$$n_{out} = f[n_{in}] \quad \Longrightarrow$$

Find the fixed point of the nonlinear equation $n - f[n] = 0$!

At each iteration, n_{out} is a functional of n_{in}

$$n_{out} = n_{out}[n_{in}]$$

Define the charge density residual:

$$R[n_{in}] = n_{out}[n_{in}] - n_{in}$$

and the norm of the residual:

$$\langle R[n_{in}]|R[n_{in}]\rangle = \int d^3\, r(R[n_{in}])^2$$

$\langle R[n_{in}]|R[n_{in}]\rangle \rightarrow 0$ at self-consistency.

Linear mixing and the Kerker preconditioner

Simple linear mixing:

$$n_{in}^{m+1} = \alpha n_{out}[n_{in}^{m}] + (1-\alpha) n_{in}^{m} = n_{in}^{m} + \alpha R[n_{in}^{m}]$$

The Kerker preconditioner (momentum-resolved linear mixing）

*G. P. Kerker Phys. Rev. B **23**, 3082 (1981)*

$$n(\boldsymbol{q}) = \frac{1}{\Omega}\int d^3\, r n(\boldsymbol{r}) e^{i\boldsymbol{q}\cdot\boldsymbol{r}}$$

The mixing parameter is ***q***-dependent

$$n_{in}^{m+1}(\mathbf{q}) = n_{in}^{m}(\mathbf{q}) + G^1(\mathbf{q}) R[n_{in}^{m}(\mathbf{q})]$$

$$G^1(\mathbf{q}) = \alpha \frac{q^2}{q_0^2 + q^2}$$

To prevent “charge sloshing” in metal slabs, arising from diverging dielectronic function at q->0.

Pulay mixer

P. Pulay, Chem. Phys. Lett. ***73****, 393 (1980)*

$$n_{in}^{opt} = \sum_{i=1}^{m} \alpha_i n_{in}^{i}$$

Assumption: the residual is linear with respect to the input charge density.

$$R\left[n_{in}^{opt}\right] = R\left[\sum_{i=1}^{m} \alpha_i n_{in}^{i}\right] = \sum_{i=1}^{m} \alpha_i R\left[n_{in}^{i}\right]$$

Minimizing the norm $\left\langle R\left[n_{in}^{opt}\right] \middle| R\left[n_{in}^{opt}\right] \right\rangle$ with respect α_i
under the constraint $\sum_{i=1}^{m} \alpha_i = 1$

$$\Rightarrow \quad \alpha_i = \sum_j A_{ij}^{-1} \Big/ \sum_{i,j} A_{ij}^{-1} \text{ , with } A_{ij} = \left\langle R\left[n_{in}^{i}\right] \middle| R\left[n_{in}^{j}\right] \right\rangle$$

$$n_{in}^{m+1} = n_{in}^{m} + GR\left[n_{in}^{opt}\right] \qquad 0 < G < 1\text{: length of step}$$

(Pulay method =“Direct inversion of the iterative sub-space (DIIS) method”)

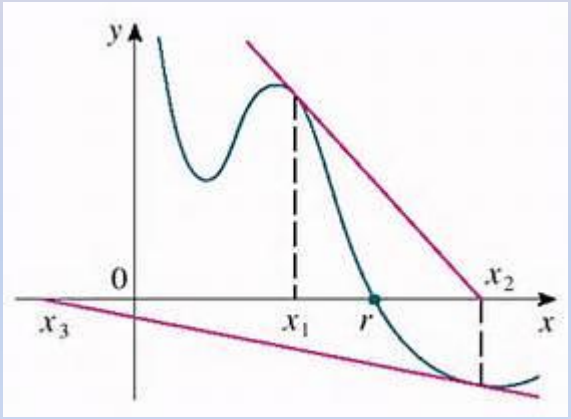
Newton-Raphson method

A method to find successively better approximation to the zeros (roots) of a real-valued function.

$x: f(x) = 0$

The Newton-Raphson algorithm:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



$(x_{n+1},0)$ is the intersection of the x -axis and the tangent of the curve at $(x_n, f(x_n))$

The algorithm can be extended to complex numbers and multi-valued functions.

Quasi Newton-Raphson method

Consider the problem

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{x},\ \text{e.g.,}\ n_{out}[n_{in}] = n_{in}$$

The residual: $\boldsymbol{R}(\boldsymbol{x}) = \boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{x}$

Solving the problem amounts to minimizing the norm $\langle \boldsymbol{R}(\boldsymbol{x}) \mid \boldsymbol{R}(\boldsymbol{x}) \rangle$

Quasi Newton-Raphson method:

$$x_{m+1} = x_m - J_m^{-1} \cdot R_m$$

Jacobian: $\boldsymbol{J} = \dfrac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}$ $(J_{mn} = \partial R_m / \partial x_n)$

Broyden mixing

Approximate way of obtaining successive Jacobians J's

Start with a reasonable guess

$$\boldsymbol{J_0^{-1}}, \qquad \text{e.g. } \boldsymbol{J_0^{-1} = \alpha I}$$

The prediction from J_{m-1}^{-1}: $\delta \boldsymbol{x}_m = \boldsymbol{x}_m - \boldsymbol{x}_{m-1} = -\boldsymbol{J}_{m-1}^{-1} \cdot \boldsymbol{R}_{m-1}$

The change of the residual from actual calculations:

$$\delta \boldsymbol{R}_m = \boldsymbol{R}_m - \boldsymbol{R}_{m-1}$$

The updated Jacobian satisfy: $\delta \boldsymbol{x}_m - \boldsymbol{J}_m^{-1} \cdot \delta \boldsymbol{R}_m = 0$

Further requiring $||\boldsymbol{J}_m^{-1} - \boldsymbol{J}_{m-1}^{-1}||$ to be minimized

$$\Rightarrow \quad \boldsymbol{J}_m^{-1} = \boldsymbol{J}_{m-1}^{-1} + \frac{(\delta \boldsymbol{x}_m - \boldsymbol{J}_{m-1}^{-1} \cdot \delta \boldsymbol{R}_m)\delta \boldsymbol{R}_m}{\langle \delta \boldsymbol{R}_m \mid \delta \boldsymbol{R}_m \rangle}$$

The minimization problem

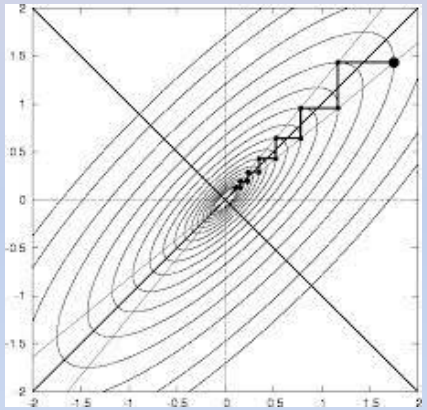
Minimizing a function

$F(x_1, x_2, \cdots, x_N)$ with respect to $x_i, i = 1, \cdots, N$

. Steepest Descend (SD) method

At a starting point in space $\{x_i^0\}$

"The steepest direction": $g_i^0 = -\frac{\partial F}{\partial x_i}\Big|_{x_i = x_i^0}$



Move along the direction: $\boldsymbol{g^0} = \{g_i^0\}$

by a distance α, $x_i^1 = x_i^0 + \alpha g_i^0$

Now $F = F(\alpha)$, the optimal $\alpha = \alpha^1$ is found by "line search".

Repeat the steps : $g_i^m = -\frac{\partial F}{\partial x_i}|_{x_i = x_i^m}, \quad x_i^{m+1} = x_i^m + \alpha^{m+1} g_i^m$

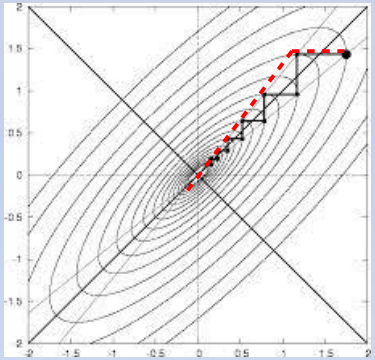
The Conjugate Gradient (CG) method

$$F = F(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}\cdot\boldsymbol{H}\cdot\boldsymbol{x}, \qquad \boldsymbol{g}(\boldsymbol{x}) = -\frac{\partial F}{\partial \boldsymbol{x}} = -\boldsymbol{H}\cdot\boldsymbol{x}, \qquad \boldsymbol{x} = \{x_i\}$$

Starting point: $\boldsymbol{x^0}, \boldsymbol{d}^0 = \boldsymbol{g}^0$

First step: $\boldsymbol{x}^1 = \boldsymbol{x}^0 + \alpha^1\boldsymbol{d}^0$ (same as SD)

From now on: $\boldsymbol{d}^m \cdot \boldsymbol{H} \cdot \boldsymbol{d}^{m+1} = 0$



$$\boldsymbol{d}^{m+1} = \boldsymbol{g}^{m+1} + \gamma^{m+1}\boldsymbol{d}^m$$

$$\gamma^{m+1} = -\frac{\boldsymbol{d}^m \cdot \boldsymbol{H} \cdot \boldsymbol{g}^{m+1}}{\boldsymbol{d}^m \cdot \boldsymbol{H} \cdot \boldsymbol{d}^m}$$

For quadratic functions, the minimum can be reached exactly with *N* steps for N-variable functions!

Direct minimization of the KS functional

The Kohn-Sham functional

$$E_{KS}[\{\phi_n\}] = \sum_n f_n \langle \phi_n \mid \hat{T} \mid \phi_n \rangle + E_{ext}[n] + E_H[n] + E_{xc}[n]$$

Minimizing E_{KS} with respect to ϕ_n, under the condition

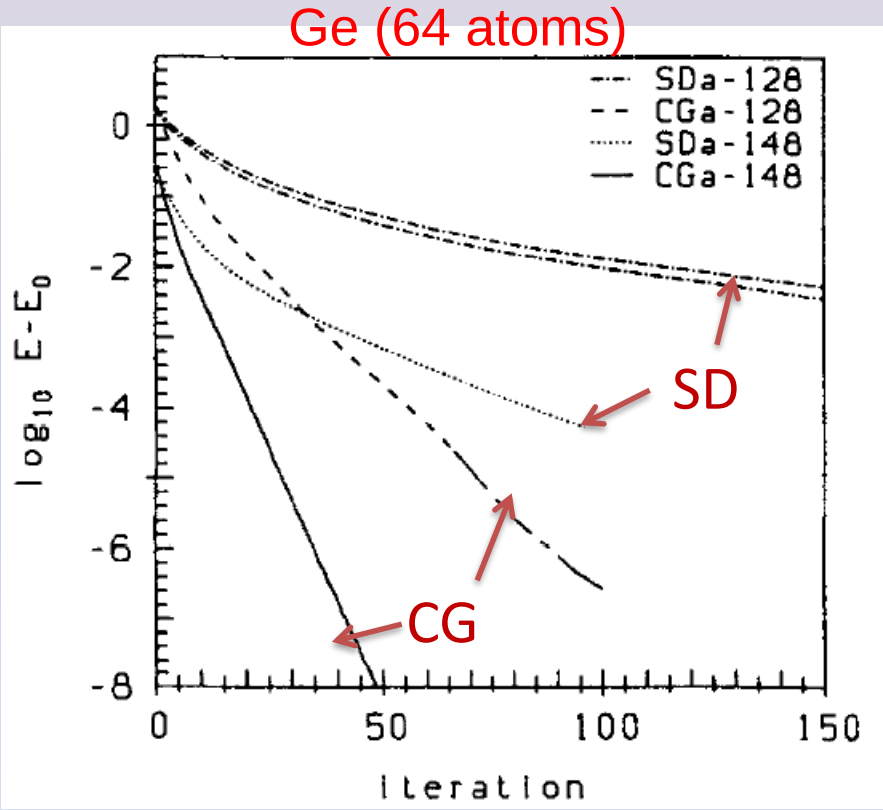
$$\langle \phi_n \mid \phi_{n'} \rangle = \delta_{n,n'}$$

$$\left(-\frac{\nabla^2}{2\mathrm{m}_e} + V_{eff}(\mathbf{r}) \right) |\phi_n\rangle = \epsilon_n |\phi_n\rangle$$

Directly minimizing $E_{KS}[\{\phi_n\}]$ amounts to solving the KS equation !

CG is the method to use!

Example



G. Kresse, Furthmüller, CMS (1996)