

OpenMXによる第一原理計算 #4

- グラフェンのSTMシミュレーション
- アセチレン分子のXPS計算
- 始状態と終状態との差電子密度解析
- 炭素鎖の透過率計算
- その他の計算における留意事項

尾崎泰助

東京大学物性研究所

第一原理電子状態計算の基礎と応用 2021年9月

Tersoff-Hamann法によるSTM像の計算

Tersoff-Hamann法によって走査型トンネル顕微鏡 (STM) 像を計算する。この方法は、化学ポテンシャルから計ったエネルギーウィンド内の部分電荷密度の計算と等価である。以下webに従い計算を実施すれば、以下の様な像が得られる。

http://www.openmx-square.org/openmx_man3.9jp/node163.html

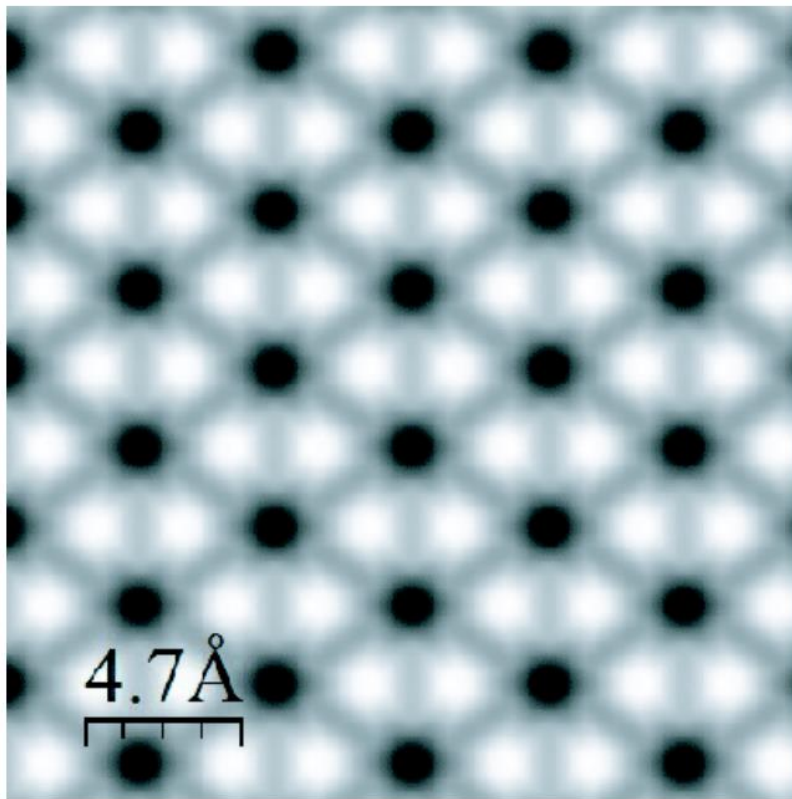


Figure 59: Tersoff-Hamann近似によるグラフェン層のSTM像。計算では「partial.charge.energy.window」に2eVに設定。入力ファイルはディレクトリ「work」中の「Graphene_STM.dat」。ソフトウェアWSxM [153]を使い、cubeファイル「Graphene_STM.pden.cube」を等値面値0.0001で可視化。

Slackでの議論

WSxMのtwitterがあります。

https://twitter.com/wsxm_official

ここからlinkが張られています。

<http://www.wsxm.eu/download.html>

ここでWSxM v5.0 Develop 10.2をdownloadして下さい。

こちらのversionしか、Gaussian cubeファイルをサポートしていません。

プルタブメニューにおいて File -> Open でFile Typeを Gaussian Cube Fileを指定して、該当するファイルを読み込んで下さい。ファイルが読み込まれたら青い画面をクリックして下さい(添付ファイルを参考にして下さい)。

次に Process -> Filter -> Create STM image として、さらに Apply を押せばSTM像が出てきます。

VESTAでも切り出した面上の等高線図が描画できるかと思いませんので、同じような図が書けると思っています。

マニュアル中でGraphene_STM.dat と記載されておりますが、work中に存在しているファイル名は**Graphite_STM.dat**です。

アセチレン分子のC-1sの絶対束縛エネルギーの計算

アセチレン分子(C₂H₂)の内殻準位(C-1s)の絶対束縛エネルギーを計算し、実験値と比較する。

http://www.openmx-square.org/openmx_man3.9jp/node192.html

1. 始状態の計算 `$mpirun -np 4 ./openmx C2H2.dat | tee C2H2.std`
C2H2.dat中で `level.of.fileout 1` と指定しておく。

2. 終状態の計算 `$mpirun -np 4 ./openmx C2H2-CH.dat | tee C2H2-CH.std`
C2H2.dat-CH中で `level.of.fileout 1` と指定しておく。

3. エネルギー差の計算

outファイルから全エネルギー値を取得し、エネルギー差を計算する。

$$\begin{aligned} E_b^{(\text{gas})} &= E_f^{(0)}(N-1) - E_i^{(0)}(N), \\ &= -66.084858926233 - (-76.787732114928) = 10.702873188695(\text{Hartree}), \\ &\approx 291.24(\text{eV}). \end{aligned}$$

実験値は 291.14 eV

始状態と終状態 (アセチレン分子)の差電子密度解析

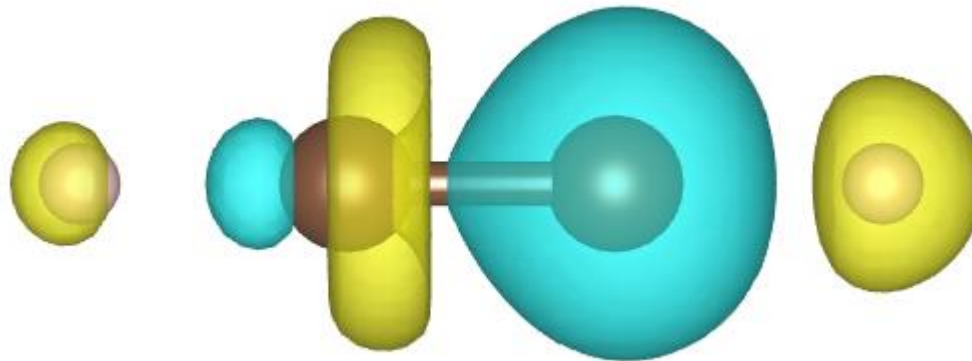
アセチレン分子(C₂H₂)の始状態と終状態の差電子密度を以下のwebに記載された手順に従い、解析する。

http://www.openmx-square.org/openmx_man3.9jp/node209.html

- (1) source中でdiff_gcubeをコンパイルし、workにコピーする。
- (2) diff_gcubeを用いて、始状態と終状態の差電子密度を計算する

```
$ ./diff_gcube C2H2.tden.cube C2H2-CH.tden.cube C2H2-diff.tden.cube
```

- (3) VESTAを用いて、C2H2-diff.tden.cubeを可視化する。

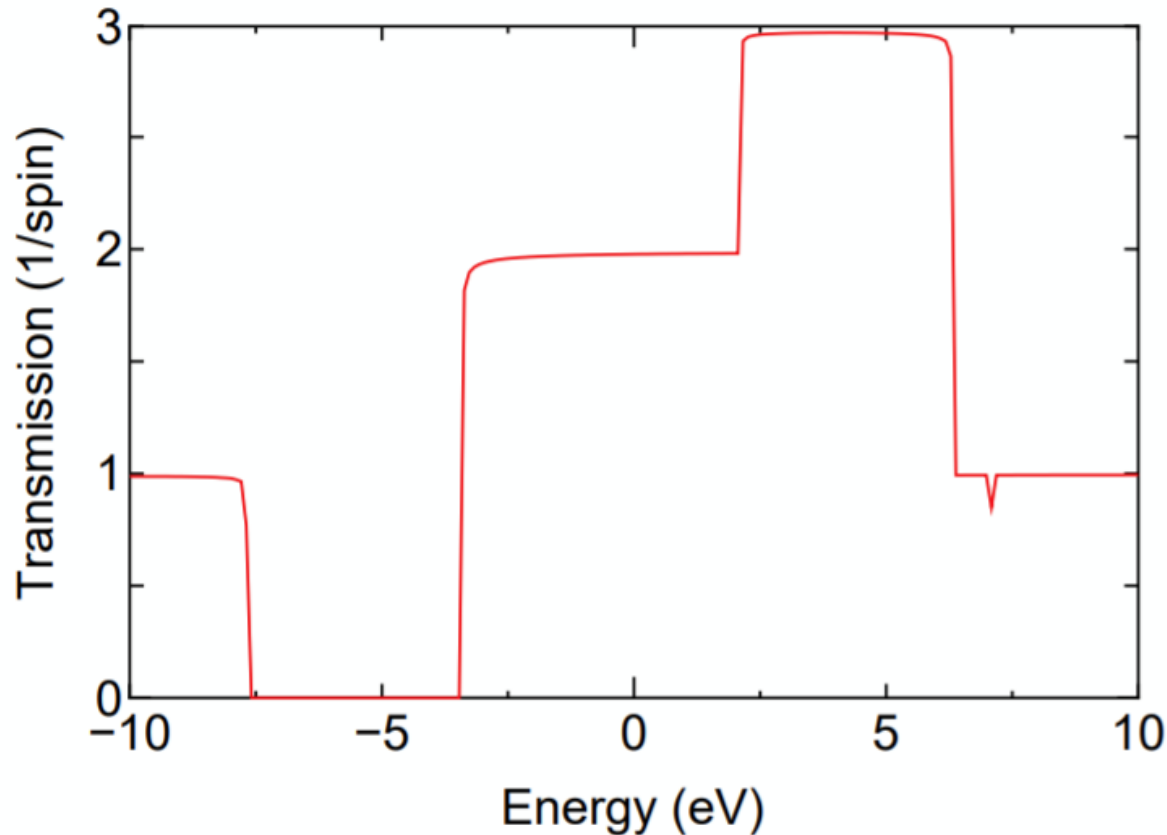


非平衡グリーン関数法による透過率計算

以下のwebの手順に従い、非平衡グリーン関数法を用いて炭素鎖の透過率を計算する。

http://www.openmx-square.org/openmx_man3.9jp/node127.html

計算が正常に実行されれば、透過率のエネルギー依存性は以下のようなになる。



計算で留意すべき点

- runtest(L)によるベンチマーク計算
- カットオフエネルギーの選択
- エネルギーカーブの計算
- SCF計算の理論と実際
- 構造最適化の理論と実際
- 基底関数の選択
- 表面状態やfloating状態
- リスタート
- バイナリモードでの出力

runtest(L)によるベンチマーク計算

OpenMXのインストールを正常に終了したとしても、プログラムが正常動作しない場合がある。この問題はコンパイラオプション(-O3)や数値演算ライブラリから、生じていることが多い。

runtestを実行することで、一通りの機能が正常に動作していることが確認できる。

For the MPI parallel running

```
% mpirun -np 8 openmx -runtest
```

For the OpenMP/MPI parallel running

```
% mpirun -np 8 openmx -runtest -nt 2
```

結果はruntest.resultに出力され、保存されている結果(全エネルギー、力)の差分が分かる。

1	input_example/Benzene.dat	Elapsed time(s)= 4.78	diff Utot= 0.000000000000	diff Force= 0.000000000002
2	input_example/C60.dat	Elapsed time(s)= 14.96	diff Utot= 0.000000000019	diff Force= 0.000000000004
3	input_example/CO.dat	Elapsed time(s)= 9.86	diff Utot= 0.000000000416	diff Force= 0.000000000490
4	input_example/Cr2.dat	Elapsed time(s)= 10.70	diff Utot= 0.000000000000	diff Force= 0.000000000044
5	input_example/Crys-MnO.dat	Elapsed time(s)= 19.98	diff Utot= 0.000000004126	diff Force= 0.000000001888
6	input_example/GaAs.dat	Elapsed time(s)= 26.39	diff Utot= 0.000000001030	diff Force= 0.000000000007
7	input_example/Glycine.dat	Elapsed time(s)= 5.48	diff Utot= 0.000000000001	diff Force= 0.000000000000
8	input_example/Graphite4.dat	Elapsed time(s)= 5.00	diff Utot= 0.000000002617	diff Force= 0.0000000015163
9	input_example/H2O-EF.dat	Elapsed time(s)= 4.88	diff Utot= 0.000000000000	diff Force= 0.000000000113
10	input_example/H2O.dat	Elapsed time(s)= 4.60	diff Utot= 0.000000000008	diff Force= 0.0000000013375
11	input_example/HMn.dat	Elapsed time(s)= 13.44	diff Utot= 0.000000000001	diff Force= 0.000000000001
12	input_example/Methane.dat	Elapsed time(s)= 3.64	diff Utot= 0.000000000001	diff Force= 0.000000002263
13	input_example/Mol_MnO.dat	Elapsed time(s)= 9.43	diff Utot= 0.000000003714	diff Force= 0.000000000540
14	input_example/Ndia2.dat	Elapsed time(s)= 5.67	diff Utot= 0.000000000004	diff Force= 0.000000000001

Total elapsed time (s) 138.79

runtestLは大きなジョブによるテスト。マシン性能を測る上で便利。様々な計算機上での結果が work/input_example 及び work/large_example 中にある。

カットオフエネルギーの選択

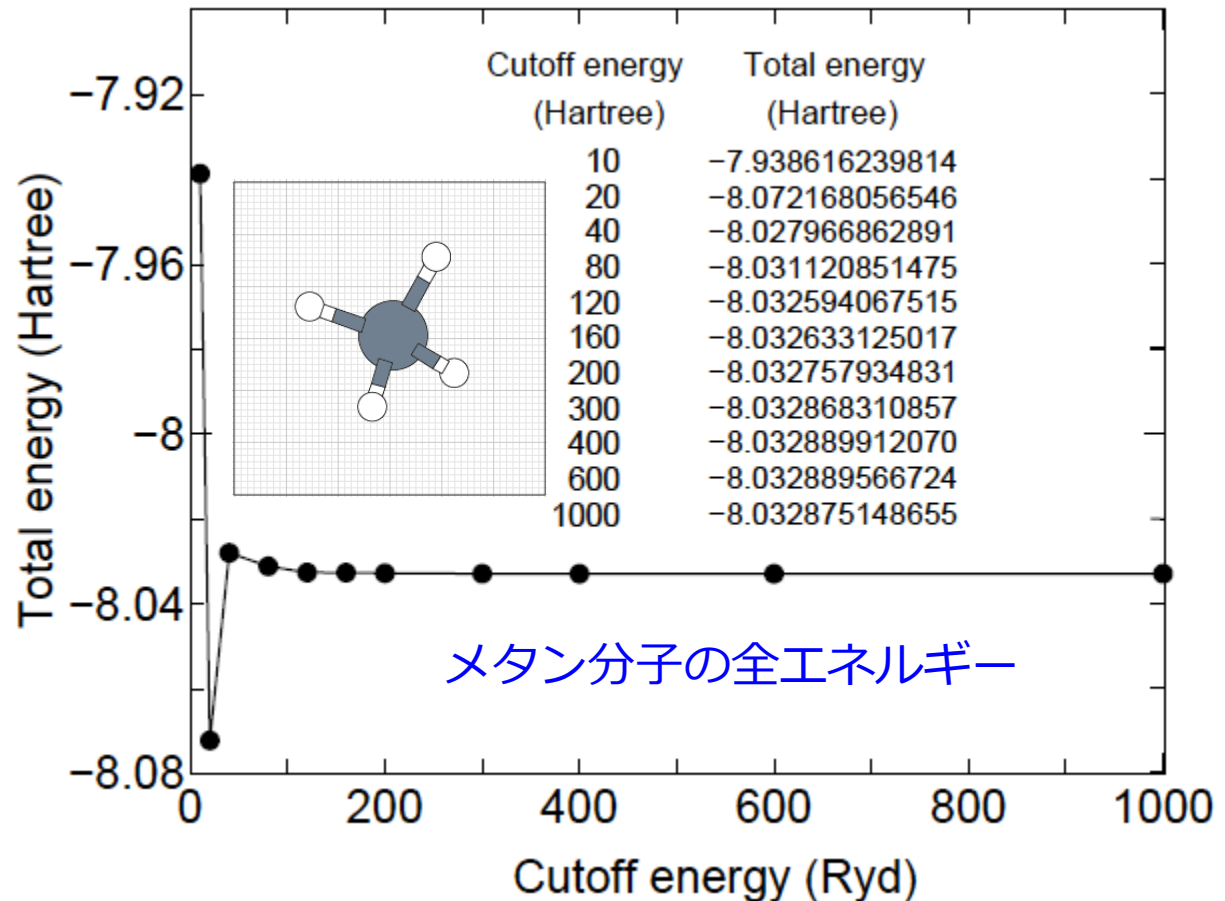
scf.energycutoff 200 # default=150 Ryd

実空間を離散化するFFTグリッドはscf.energycutoffで指定する。FFTグリッドはPoisson方程式の解法、 $E_{\delta_{ee}}$, E_{xc} の計算に使用する。

多くの系で200 Ryd程度で収束する。

擬ポテンシャルにセミコア状態を含む場合には、大きめのカットオフを取った方が安全。300-400 Ryd. 程度。

メモリの増加量:
 $O(E^{3/2})$



カットオフエネルギーの選択

H₂Oの最適化構造

水分子の最適化構造のscf.energycutoffに対する依存性。
180Ryd.でほぼ収束していることが分かる

Table 1: Convergence of structural parameters, dipole moment of a water molecule with respect to the cutoff energy. The input file is 'H2O.dat' in the directory 'work'.

Ecut(Ryd)	r(H-O) (Å)	∠ (H-O-H) (deg)	Dipole moment (Debye)
60	0.970	103.4	1.838
90	0.971	103.7	1.829
120	0.971	103.7	1.832
150	0.971	103.6	1.829
180	0.971	103.6	1.833
Exp.	0.957	104.5	1.85

エネルギーカーブの計算

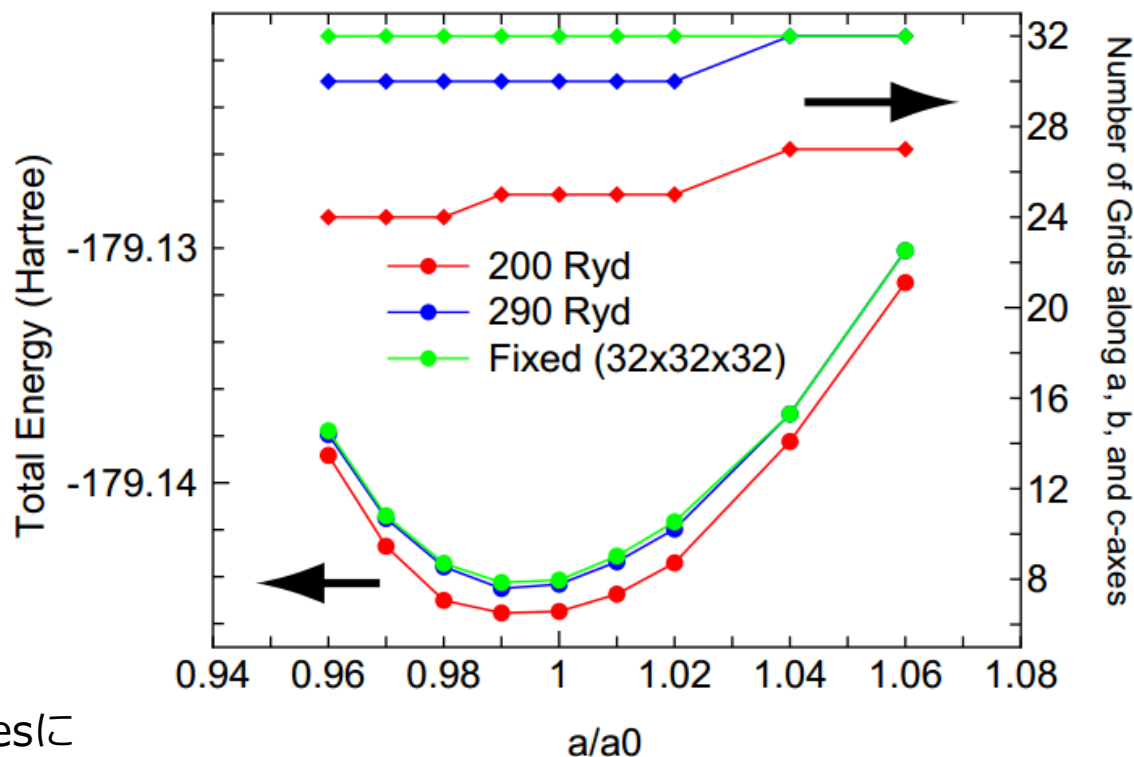
格子定数を変化させてエネルギー曲線を計算するには以下のkeywordが使用できる。

MD.Type	Evslc	#
MD.Evslc.Step	0.4	# default=0.4%
MD.maxIter	32	# default=1

scf.energycutoffによってグリッドを決定すると、格子定数の増加に応じて、不連続にグリッド数が変化する。このグリッド数の突然の変化はエネルギーカーブに不連続なジャンプを生じさせる。これを避けるためにはグリッド数を固定して計算を行うべきである。次のkeywordで明示的にグリッド数が指定できる。

scf.Ngrid 32 32 32

それぞれの数字はa-, b-, and c-axesに対するグリッド数。scf.energycutoffとscf.Ngridをどちらも指定した場合にはscf.Ngridが優先される。



MD.Evslc.flag 1 1 0 # a, b, c-軸に対して
1: 格子定数を変化, 0: 格子定数を変化させない。

SCF計算：電子密度の混合

KS有効ポテンシャルは ρ の関数である。
しかし本来、 ρ はKS方程式から求められる。

$$\hat{H}_{\text{KS}}\phi_i = \varepsilon_i\phi_i \quad \hat{H}_{\text{KS}} = -\frac{1}{2}\nabla^2 + v_{\text{eff}}$$

$$v_{\text{eff}} = v_{\text{ext}}(\mathbf{r}) + v_{\text{Hartree}}(\mathbf{r}) + \frac{\delta E_{\text{xc}}}{\delta\rho(\mathbf{r})}$$

$$\rho(\mathbf{r}) = \sum_i^{\text{occ}} \phi_i^*(\mathbf{r})\phi_i(\mathbf{r})$$

単純混合法

入力と出力の電子密度を単純に足し合わせて、次の入力電子密度とする。

$$\rho_{n+1}^{(\text{in})} = \alpha\rho_n^{(\text{in})} + (1 - \alpha)\rho_n^{(\text{out})},$$

大きなバンドギャップのある系、小さい系に有効な方法

OpenMXで利用可能なSCF収束法

Simple mixing (Simple)

関連するkeywords: scf.Init.Mixing.Weight, scf.Min.Mixing.Weight, scf.Max.Mixing.Weight

Residual minimization method in the direct inversion iterative subspace (RMM-DIIS)

関連するkeywords: scf.Init.Mixing.Weight, scf.Min.Mixing.Weight, scf.Max.Mixing.Weight, scf.Mixing.History, scf.Mixing.StartPulay

Guaranteed reduction Pulay method (GR-Pulay)

関連するkeywords: scf.Init.Mixing.Weight, scf.Min.Mixing.Weight, scf.Max.Mixing.Weight, scf.Mixing.History, scf.Mixing.StartPulay

Kerker mixing (Kerker)

関連するkeywords: scf.Init.Mixing.Weight, scf.Min.Mixing.Weight, scf.Max.Mixing.Weight, scf.Kerker.factor

RMM-DIIS with Kerker metric (RMM-DIISK)

関連するkeywords: scf.Init.Mixing.Weight, scf.Min.Mixing.Weight, scf.Max.Mixing.Weight, scf.Mixing.History, scf.Mixing.StartPulay, scf.Mixing.EveryPulay, scf.Kerker.factor

- 5つのSCF収束法が利用可能。
- 多くの系でRMM-DIISKがベスト。
- RMM-DIIS、RMM-DIISKはどちらもPulay混合法であるが、前者は密度行列、後者は電子密度のフーリエ変換に対してPulay混合を適用する。
- 非常に収束が難しい系はKerkerがベター。
- plus Uの場合にはRMM-DIISの方がRMM-DIISKより収束性に優れることもしばしばある。

RMM-DIISKによるSCF収束

アイデア

過去の(フーリエ空間での)残差ベクトルの線形和のノルムを最小化する。

$$\bar{R}_{n+1} = \sum_{m=n-(p-1)}^n \alpha_m R_m, \quad R_n(\mathbf{q}) \equiv \tilde{n}_n^{(\text{out})}(\mathbf{q}) - \tilde{n}_n^{(\text{in})}(\mathbf{q}),$$

$$F = \langle \bar{R}_{n+1} | \bar{R}_{n+1} \rangle - \lambda \left(1 - \sum_m^n a_m \right), \quad \langle R_m | R_{m'} \rangle \equiv \sum_{\mathbf{q}} \frac{R_m^*(\mathbf{q}) R_{m'}(\mathbf{q})}{w(\mathbf{q})},$$

$$= \sum_{m,m'} \alpha_m \alpha_{m'} \langle R_m | R_{m'} \rangle - \lambda \left(1 - \sum_m^n a_m \right). \quad \text{Kerker factor } w(\mathbf{q}) = \frac{|\mathbf{q}|^2}{|\mathbf{q}|^2 + q_0^2},$$

Fの最小化条件から次式が得られる。

$|\mathbf{q}|$ が小さい成分(長波長成分)に重きを置く。

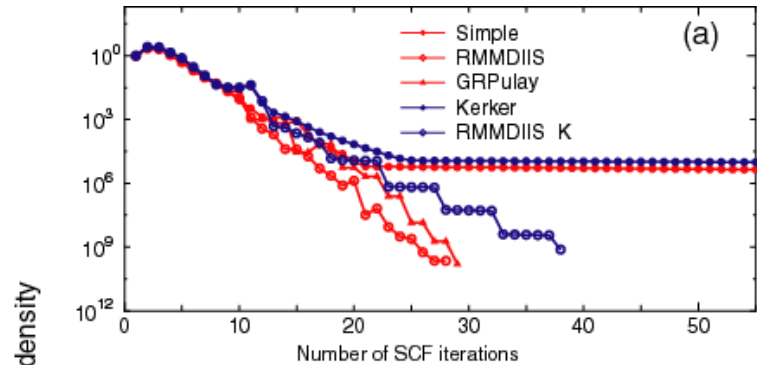
$$\frac{\partial F}{\partial \alpha_k} = 0 \quad \begin{pmatrix} \langle R_{n-(p-1)} | R_{n-(p-1)} \rangle & \cdots & \cdots & 1 \\ \cdots & \cdots & \cdots & 1 \\ \cdots & \cdots & \langle R_n | R_n \rangle & \cdots \\ 1 & \cdots & \cdots & 0 \end{pmatrix} \begin{pmatrix} \alpha_{n-(p-1)} \\ \alpha_{n-(p-1)+1} \\ \vdots \\ \frac{1}{2}\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}.$$

“最適”な入力電子密度は次式で与えられる。

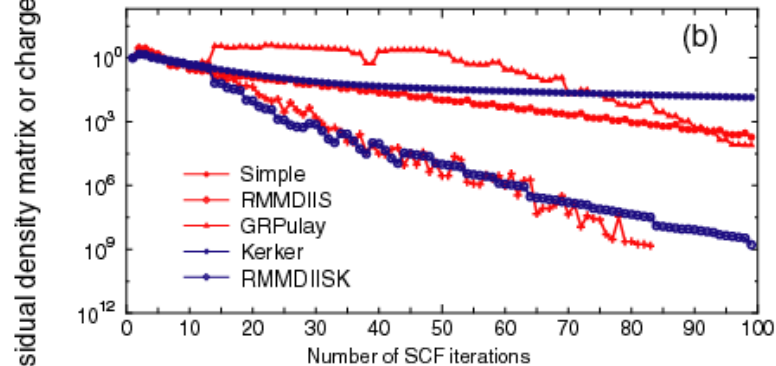
$$\rho_{n+1}^{(\text{in})} = \sum_{m=n-(p-1)}^n \alpha_m \rho_m^{(\text{in})} + \beta \sum_{m=n-(p-1)}^n \alpha_m R_m$$

様々な電荷密度混合法の比較

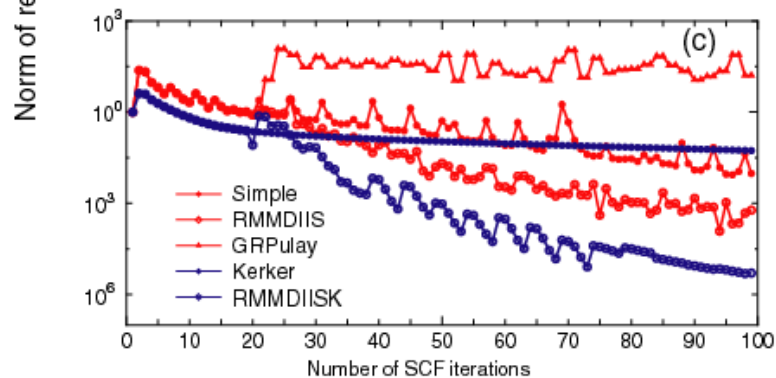
シアリック酸



Pt₁₃ クラスタ



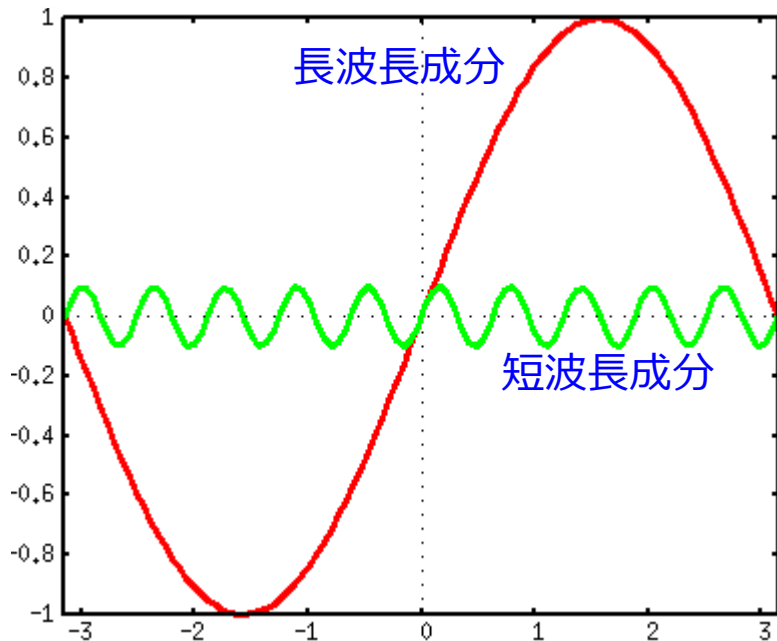
Pt₆₃ クラスタ



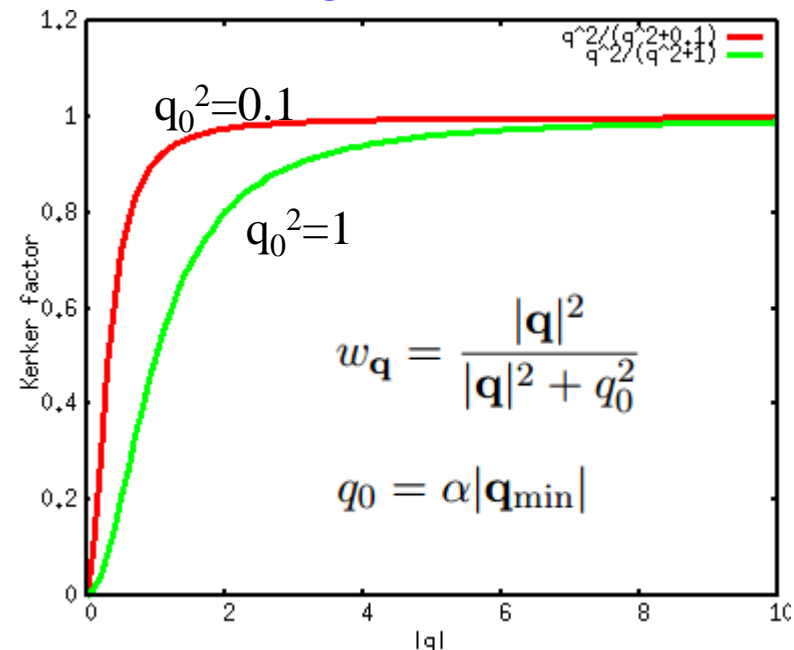
いずれの場合も
RMM-DIISもしくは
RMM-DIISKが
高速である。

Charge sloshingとKerker因子

SCF計算の際の大きな電子密度の揺らぎは密度の長波長成分(小さいq)から生じる



長波長成分から生じるcharge sloshingを低減するために、Kerker因子を導入する。 q_0 が大きいほど、sloshingが抑えられる。



Kerker因子 α はscf.Kerker.factor によって明示的に与えられるが、このkeywordを指定しない場合には自動で適切な値を推定する。

$$\alpha = \frac{0.5}{|\mathbf{b}_{\min}|^2} \left(4 \frac{Dq}{Aq} + 1.0 \right)$$

$$Aq = \frac{1}{3} \left(|\mathbf{b}_1|^2 + |\mathbf{b}_2|^2 + |\mathbf{b}_3|^2 \right),$$

$$Dq = \frac{1}{3} \sum_{i < j} \left| |\mathbf{b}_i|^2 - |\mathbf{b}_j|^2 \right|,$$

推定値は標準出力に以下の様に表示されるので、この値を手掛かりに調整すると良い。

Automatic determination of Kerker_factor: ****

RMM-DIISKのkeywords

RMM-DIISKは以下のkeywordsで制御される。

scf.Init.Mixing.Weight	0.01
scf.Min.Mixing.Weight	0.01
scf.Max.Mixing.Weight	0.2
scf.Mixing.History	40
scf.Mixing.StartPulay	20
scf.Mixing.EveryPulay	1
scf.Kerker.factor	5.0

- scf.Mixing.StartPulayはRMM-DIISKをスタートするSCFステップを指定する。
- それ以前のステップはKerker法が適用される。RMM-DIISK法は準ニュートン法であるから、ある程度の収束が達成された後に、乗り換えるべきである。さもないと収束しない場合がある。
- RMM-DIISKにおける履歴数はscf.Mixing.Historyで指定する。
- scf.Init.Mixing.Weight, scf.Min.Mixing.Weight, scf.Max.Mixing.Weight はKerker法を適用される際の混合比の初期値、最小値、最大値である。収束状況をモニターし、この範囲で自動的に調整される。
- Kerker因子はscf.Kerker.factorで与えられる。指定しない場合には自動的に最適値が推定され、使用される。
- RMM-DIISK法に乗り換えた後に、RMM-DIISKによる混合を実施する頻度をscf.Mixing.EveryPulayで指定する。1の場合には毎回、実施することに対応。

LDA+UにおけるSCF収束

LDA+U法のエネルギー表式

$$E = E_{\text{DFT}}[\rho] + E_{\text{U}}[\mathbf{n}]$$

$$E_{\text{U}} = \frac{1}{2} \sum_{\alpha} (U_{\alpha} - J_{\alpha}) \sum_{\sigma} \{ \text{Tr}(n_{\alpha}^{\sigma}) - \text{Tr}(n_{\alpha}^{\sigma} n_{\alpha}^{\sigma}) \}.$$

- LDA+U法のエネルギーは電子密度と密度行列の汎関数である。
- 電子密度 ρ に対する混合法, Kerker, RMM-DIISK法をLDA+U法に適用すると、密度行列 \mathbf{n} に対しては混合が行われずに、1ステップ前の密度行列 \mathbf{n} が使用されることになる。
- これは単純混合の100%混合に対応し、密度行列のsloshingが引き起こされる可能性がある。
- **そこでLDA+U法ではRMM-DIIS法の使用が望ましい。**
- 本来は電子密度と密度行列の混合を同時に実施する混合法を開発すべきであるが、現時点でのOpenMXには実装されていない。

他の電子密度混合法との比較

Anderson混合法

$$\bar{\rho}_n^{(\text{in})} = \rho_n^{(\text{in})} + \sum_{m=n-(p-1)}^{n-1} \theta_m (\rho_m^{(\text{in})} - \rho_n^{(\text{in})}), \quad \bar{R}_n^{(\text{in})} = R_n^{(\text{in})} + \sum_{m=n-(p-1)}^{n-1} \theta_m (R_m^{(\text{in})} - R_n^{(\text{in})}).$$

$$\rho_{n+1}^{(\text{in})} = \bar{\rho}_n^{(\text{in})} + \beta \bar{R}_n^{(\text{in})} \quad \rightarrow \text{equivalent to RMM-DIIS}$$

Broyden混合法

$$E = |G_n - G_{n-p}|^2 + \sum_{m=l+1-p}^{n-1} \langle \lambda_m | \{ (|n_{m+1}\rangle - |n_m\rangle) - G_n (|R_{m+1}\rangle - |R_m\rangle) \}$$

$$\frac{\partial E}{\partial G} = 0 \quad \frac{\partial E}{\partial \lambda} = 0$$

$$|n_{n+1}\rangle = |n_n\rangle - \sum_{m=n-p}^{n-1} \gamma_m^n (|n_{m+1}\rangle - |n_m\rangle) - G_{n-p} \left\{ |R_n\rangle - \sum_{m=n-p}^{n-1} \gamma_m^n (|n_{m+1}\rangle - |n_m\rangle) \right\}$$

$$G_{n-p} = -\beta \quad \rightarrow \quad \rho_{n+1}^{(\text{in})} = \bar{\rho}_n^{(\text{in})} + \beta \bar{R}_n^{(\text{in})} \quad \rightarrow \text{equivalent to RMM-DIIS}$$

RMM-DIIS, Anderson, Broyden 法は数学的には等価の方法であり、準Newton法に基づいている。

SCF収束性の改良に向けて

Broyden法

$$|n_{n+1}\rangle = |n_n\rangle - \sum_{m=n-p}^{n-1} \gamma_m^n (|n_{m+1}\rangle - |n_m\rangle) - G_{n-p} \left\{ |R_n\rangle - \sum_{m=n-p}^{n-1} \gamma_m^n (|n_{m+1}\rangle - |n_m\rangle) \right\}$$

$$G_{n-p} = -\beta \quad \rightarrow \quad \rho_{n+1}^{(\text{in})} = \bar{\rho}_n^{(\text{in})} + \beta \bar{R}_n^{(\text{in})}$$

もしGが保存できるならば、Broyden法がベストな方法であると考えられる。しかしながら、Gの行列サイズはq点に比例するため、保持しておくことは難しい。したがって現行のOpenMXにおいて、理論的な観点から**収束性を高める最も有効な方法は、履歴数を大きく取ること**である。

収束が難しい系に対しては30~50の履歴数をとると良い。

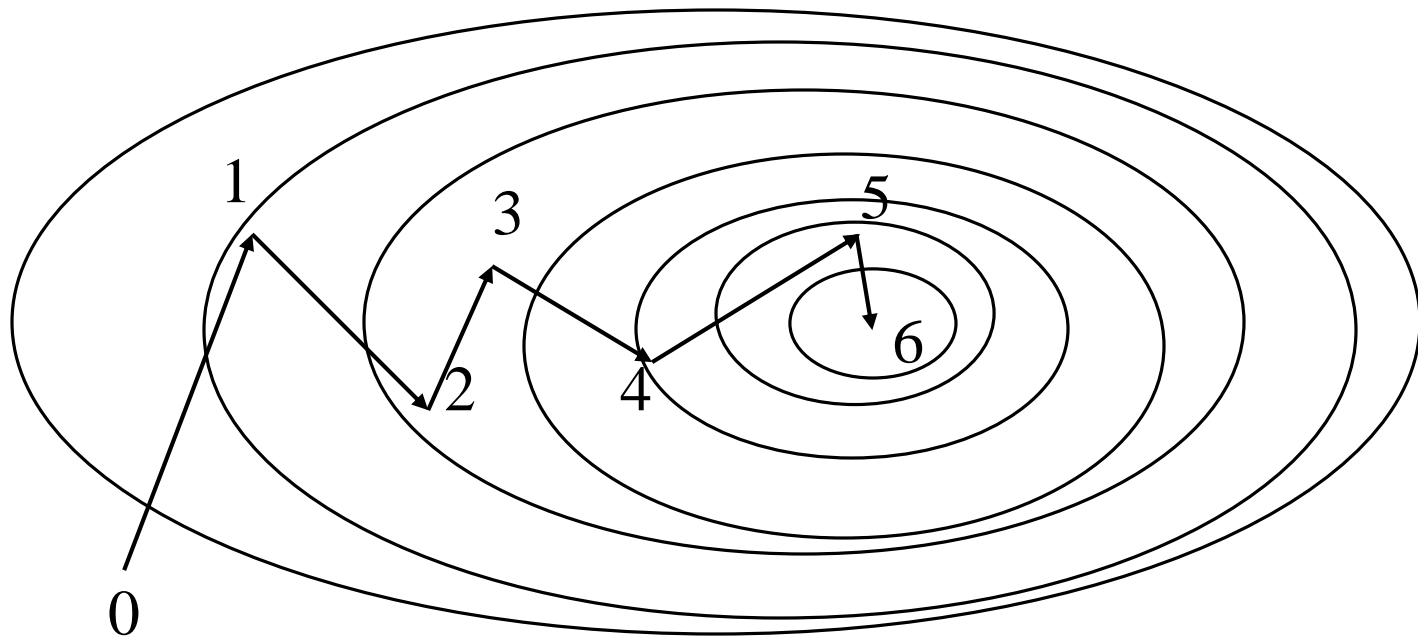
Broyden法の利点を生かしつつ、計算・メモリコストの小さい計算手法の開発が望まれる。L-BFGS法との組み合わせは有望であろう。

構造最適化

分子や固体の性質を精密に調べるためにはエネルギーの最小値を与える構造(最適構造)を計算する必要がある。

最急降下法(Steepest decent method)

エネルギーの勾配方向に少しずつ移動する。



あまり効率的ではない。

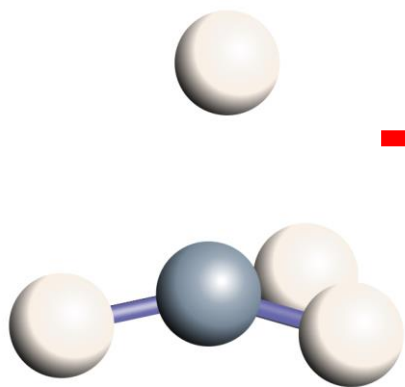
最急勾配法による構造最適化

メタン分子の炭素原子のx座標が
以下のように0.3 Åに変更

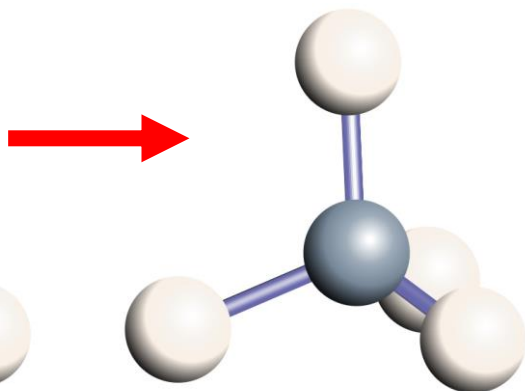
```
<Atoms.SpeciesAndCoordinates
 1 C  0.300000  0.000000  0.000000  2.0  2.0
 2 H -0.889981 -0.629312  0.000000  0.5  0.5
 3 H  0.000000  0.629312 -0.889981  0.5  0.5
 4 H  0.000000  0.629312  0.889981  0.5  0.5
 5 H  0.889981 -0.629312  0.000000  0.5  0.5
Atoms.SpeciesAndCoordinates>
```

Work中の「Methane2.dat」を
用いて本計算は再現できる。

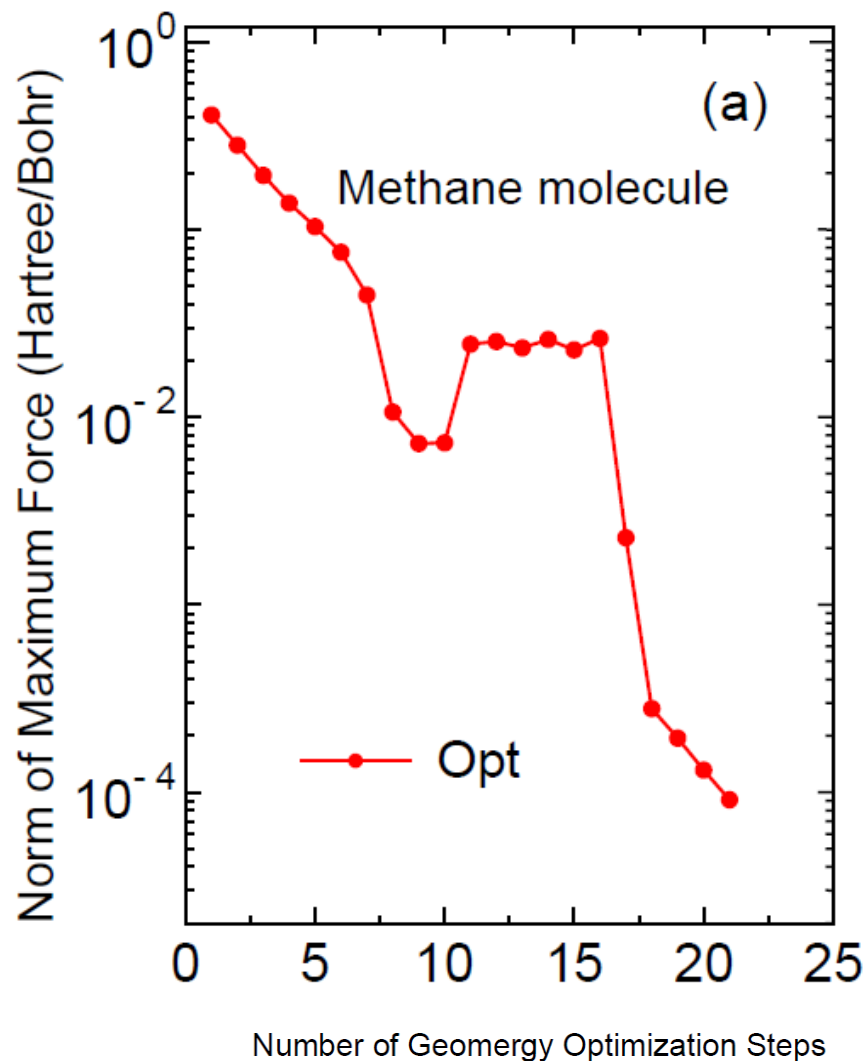
初期構造



最終構造



構造最適化の過程



解析に関する出力ファイル

met2.out中に構造最適化の履歴が確認できる。

```
*****  
*****  
History of geometry optimization  
*****  
*****
```

MD_iter	SD_scaling	Maximum force (Hartree/Bohr)	Maximum step (Ang)	Utot (Hartree)
1	1.25981733	0.40873710	0.10583545	-8.09571722
2	1.25981733	0.12000148	0.08000099	-8.16523842
3	1.25981733	0.06115237	0.04076824	-8.18400222
4	1.25981733	0.01811189	0.01207460	-8.18831175
5	3.14954331	0.01076505	0.01794175	-8.18898468
6	3.14954331	0.00576811	0.00961351	-8.18980390
7	3.14954331	0.00355034	0.00591724	-8.19010991
8	3.14954331	0.00255760	0.00426266	-8.19022887
9	7.87385828	0.00218904	0.00912099	-8.19027446
10	7.87385828	0.00863395	0.03597478	-8.19023586
11	1.57477166	0.04692594	0.03910495	-8.18816953
12	1.57477166	0.01357319	0.01131099	-8.19012703
13	3.93692914	0.00406864	0.00847634	-8.19029047
14	3.93692914	0.00915707	0.01907723	-8.19022710
15	0.78738583	0.02067017	0.00861257	-8.18990108
16	0.78738583	0.00721609	0.00300671	-8.19025699
17	1.96846457	0.00255359	0.00265999	-8.19030094
18	1.96846457	0.00163789	0.00170613	-8.19030460
19	1.96846457	0.00105501	0.00109897	-8.19030613
20	1.96846457	0.00068667	0.00071528	-8.19030671
21	4.92116143	0.00044434	0.00115713	-8.19030697
22	4.92116143	0.00138402	0.00360421	-8.19030537
23	0.98423229	0.00427346	0.00222576	-8.19028994
24	0.98423229	0.00078570	0.00040922	-8.19030656
25	2.46058071	0.00013628	0.00017745	-8.19030709
26	2.46058071	0.00014411	0.00018765	-8.19030714
27	0.49211614	0.00015118	0.00003937	-8.19030709
28	0.49211614	0.00008899	0.00002317	-8.19030713

met2.mdをOpenMX Viewerにドラッグ&ドロップすると構造最適化過程が容易に確認できる。

The screenshot shows a web browser window at www.openmx-square.org/viewer/index.html. The main content area displays a 3D ball-and-stick model of a molecule with four atoms (one blue, three white) connected by bonds. A dialog box titled "Structural Change" is overlaid on the model, showing "time= 0.000 (fs) Energy= -8.09572 (Hartree)" and "Frame 1/28". The dialog box contains navigation buttons: "First", "<<", "Stop", ">>", and "Last". Below these are fields for "Make animation" (Every 1 frames), "Delayed time (ms)" (0), and "File format" (not supp.). At the bottom of the browser window, a status bar shows various settings: "Supercell" (1x1x1), "Atoms" (rendering), "Bonds" (fine), "Number" (off), "Symbol" (off), "BG" (black), "Atom Size" (1.0), "Bond Thickness" (1.0), "Bond Factor" (1.0), "Cells" (off), "Axes" (off), "Perspective" (off), "Structure" (off), "Dynamics" (on), "Net Charge" (off), "Force" (off), "Velocity" (off), "Rot on x" (0), "Rot on y" (0), "Rot on z" (0). The status bar also includes the text "OpenMX website" and "Rotate: click+drag Zoom: scroll Translate: ctrl+click+drag Note: click the middle button for Safari".

ニュートン法による構造最適化

エネルギーをTaylor展開する

$$E = E_0 + \sum_i^{3N} \left(\frac{\partial E}{\partial x_i} \right)_0 (x_i - x_i^{(0)}) + \frac{1}{2} \sum_{i,j}^{3N} \left(\frac{\partial^2 E}{\partial x_i \partial x_j} \right)_0 (x_i - x_i^{(0)}) (x_j - x_j^{(0)}) + \dots,$$

座標の微分を計算

$$\frac{\partial E}{\partial x_k} = \left(\frac{\partial E}{\partial x_k} \right)_0 + \sum_i^{3N} \left(\frac{\partial^2 E}{\partial x_k \partial x_i} \right)_0 (x_i - x_i^{(0)}).$$

$$\frac{\partial E}{\partial x_k} = 0 \quad \longrightarrow \quad \begin{pmatrix} \left(\frac{\partial^2 E}{\partial x_1 \partial x_1} \right)_0 & \left(\frac{\partial^2 E}{\partial x_1 \partial x_2} \right)_0 & \cdots \\ \left(\frac{\partial^2 E}{\partial x_2 \partial x_1} \right)_0 & \left(\frac{\partial^2 E}{\partial x_2 \partial x_2} \right)_0 & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} \begin{pmatrix} (x_1 - x_1^{(0)}) \\ (x_2 - x_2^{(0)}) \\ \cdots \end{pmatrix} = - \begin{pmatrix} \left(\frac{\partial E}{\partial x_1} \right)_0 \\ \left(\frac{\partial E}{\partial x_2} \right)_0 \\ \cdots \end{pmatrix}.$$

$$H \Delta \mathbf{x} = -\mathbf{g},$$

これを $\Delta \mathbf{x}$ について解けば、

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - (H^{(n)})^{-1} \mathbf{g}^{(n)}.$$

もしHessian行列Hが求められるならば、この方法は効率的である。
しかし一般の場合に、Hを求めることは困難である。

準ニュートン法による構造最適化

OpenMXにおける構造最適化は準ニュートン法に基づいている。
OpenMX Ver. 3.7においては4つの最適化方法が実装されている。

$$\mathbf{r}_{\text{new}} = \mathbf{r}_{\text{DIIS}} + \Delta\mathbf{r}$$

$$\Delta\mathbf{r} = -H^{-1}\mathbf{g}_{\text{DIIS}}$$

近似ヘシアン行列Hの計算手法

DIIS

BFGS

RF(rational function)

EF(eigenvector following)

H=I

BFGS

BFGS+RF

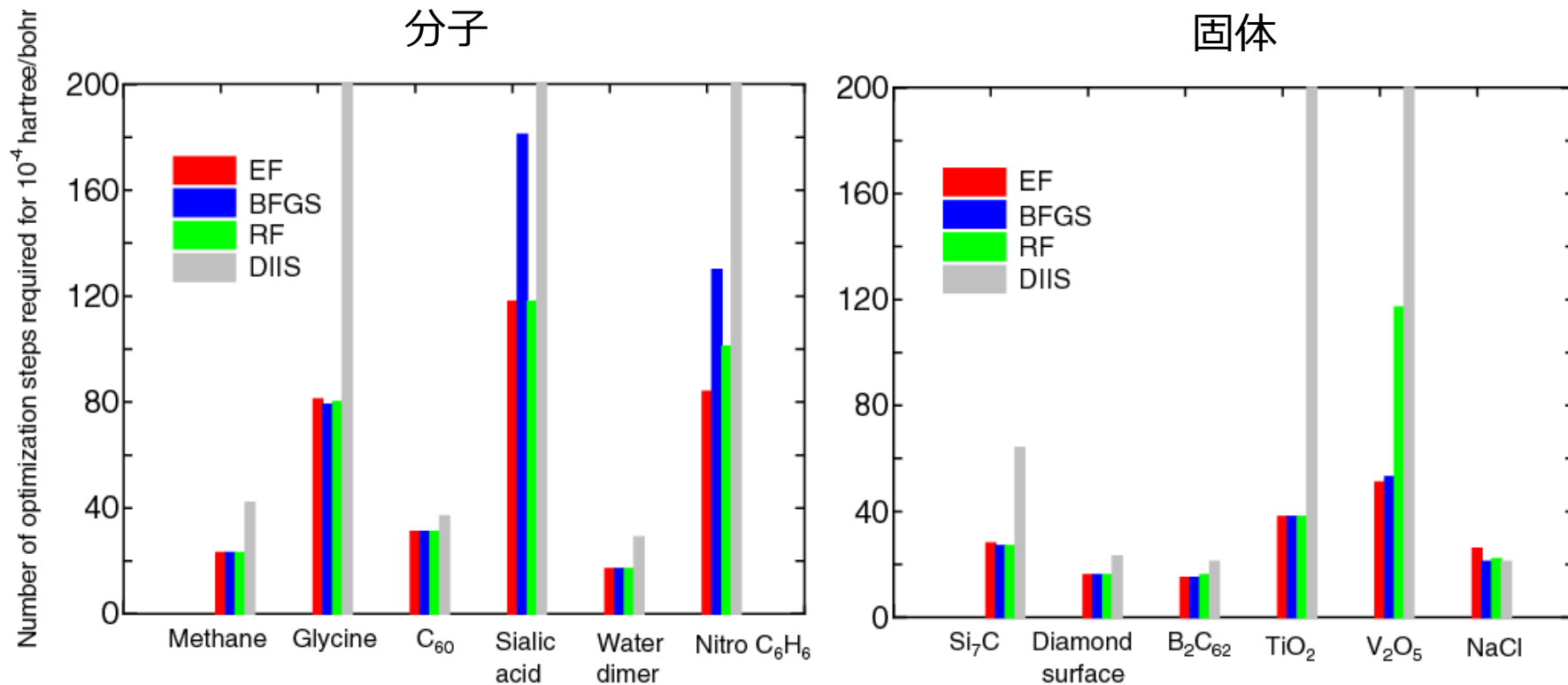
BFGS plus monitoring
of eigenvalues of H

Broyden-Fletcher-Goldfarb-Shanno (BFGS) 法

$$H_k = H_{k-1} + \frac{|\Delta\mathbf{g}_k\rangle\langle\Delta\mathbf{g}_k|}{\langle\Delta\mathbf{g}_k|\Delta\mathbf{r}_k\rangle} - \frac{|H_k\Delta\mathbf{r}_k\rangle\langle\Delta\mathbf{r}_kH_k|}{\langle\Delta\mathbf{r}_k|H_k|\Delta\mathbf{r}_k\rangle}$$

赤枠で囲まれた内積が正であるならば、Hの正定値性は保たれる。

構造最適化法の比較



- ベンチマーク計算からはEF法とRF法の性能が良いことが分かる。
- 構造に自由度が大きい分子系は収束しにくい。

このベンチマーク計算の入カファイルと出カファイルは以下のディレクトリにある。
openmx3.7/work/geoopt_example".

構造最適化に関連するkeywords

MD.Type	EF	# Opt DIIS BFGS RF EF
MD.Opt.DIIS.History	3	# default=3
MD.Opt.StartDIIS	5	# default=5
MD.Opt.EveryDIIS	200	# default=200
MD.maxIter	100	# default=1
MD.Opt.criterion	1.0e-4	# default=0.0003 (Hartree/Bohr)

特に、これらの準ニュートン法に基づいた構造最適化法は次の2種のキーワードにより制御することができます。

```
MD.Opt.DIIS.History 3 # default=3
MD.Opt.StartDIIS    5 # default=5
```

キーワード「MD.Opt.DIIS.History」は近似ヘシアンを更新するための過去の履歴ステップ数を指定します。デフォルト値は3です。また、「EF」、「BFGS」、「RF」、「DIIS」を開始する構造最適化ステップをキーワード「MD.Opt.StartDIIS」により指定します。これらの方法を開始する前の構造最適化ステップは、最急降下法により実行されます。デフォルト値は5です。

エネルギーカーブの計算

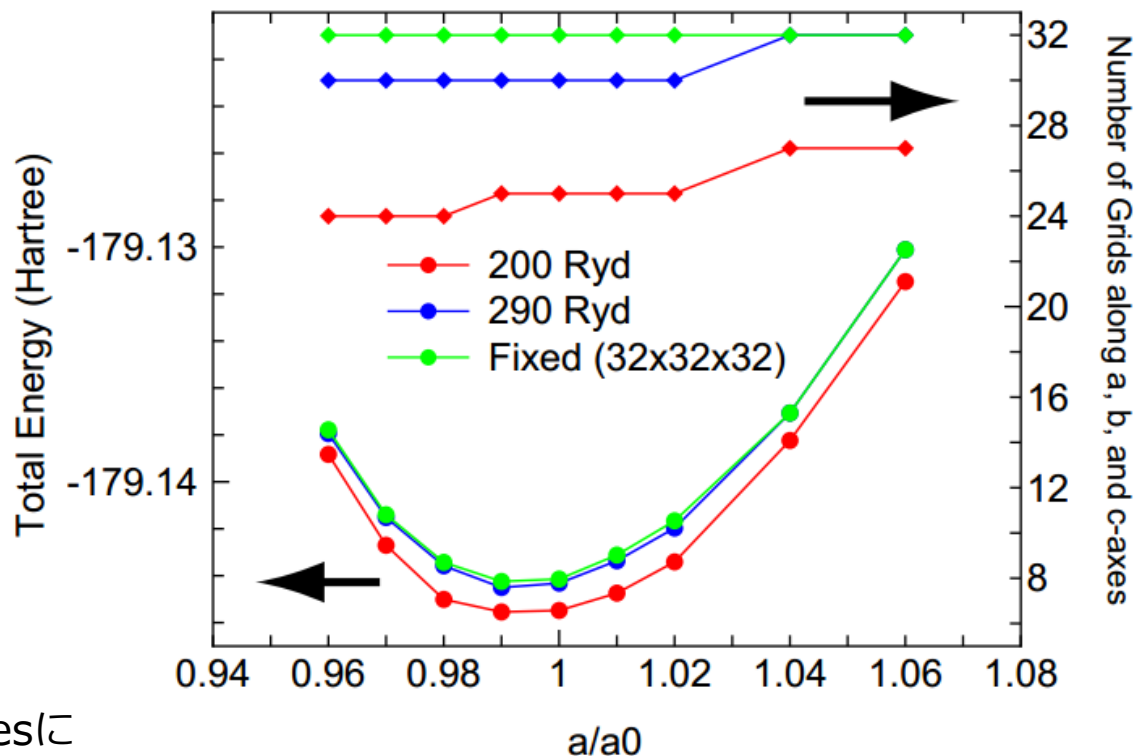
格子定数を変化させてエネルギー曲線を計算するには以下のkeywordが使用できる。

MD.Type	Evslc	#
MD.Evslc.Step	0.4	# default=0.4%
MD.maxIter	32	# default=1

scf.energycutoffによってグリッドを決定すると、格子定数の増加に応じて、不連続にグリッド数が変化する。このグリッド数の突然の変化はエネルギーカーブに不連続なジャンプを生じさせる。これを避けるためにはグリッド数を固定して計算を行うべきである。次のkeywordで明示的にグリッド数が指定できる。

scf.Ngrid 32 32 32

それぞれの数字はa-, b-, and c-axesに対するグリッド数。scf.energycutoffとscf.Ngridをどちらも指定した場合にはscf.Ngridが優先される。



MD.Evslc.flag 1 1 0 # a, b, c-軸に対して
1: 格子定数を変化, 0: 格子定数を変化させない。

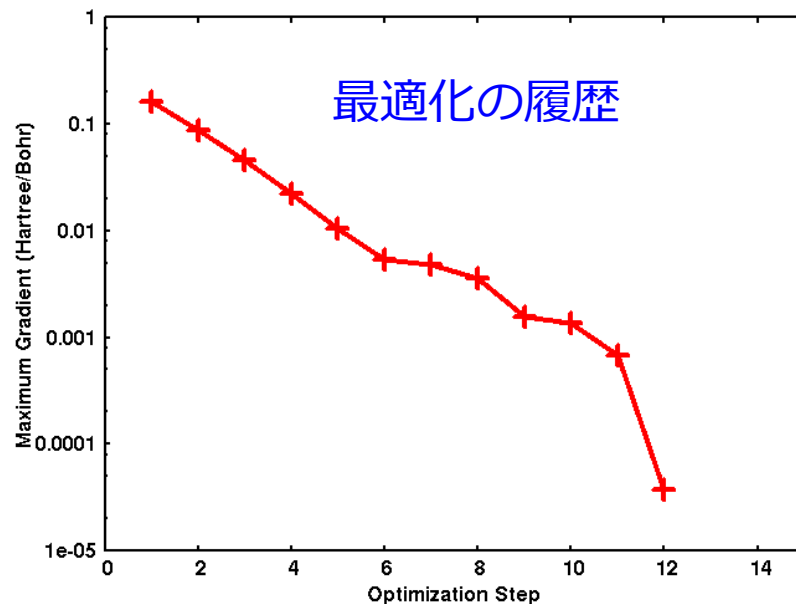
格子定数・内部座標の最適化

ダイヤモンド格子の内部座標を少しずらした所からスタートする。

```
Atoms.Number          2
Atoms.SpeciesAndCoordinates.Unit  frac # Ang|AU
<Atoms.SpeciesAndCoordinates
  1  C  0.10000000000000  0.00000000000000  -0.050000000000000  2.0  2.0
  2  C  0.25000000000000  0.25000000000000  0.250000000000000  2.0  2.0
Atoms.SpeciesAndCoordinates>
Atoms.UnitVectors.Unit          Ang # Ang|AU
<Atoms.UnitVectors
  1.6400  1.6400  0.0000
  1.6400  0.0000  1.6400
  0.0000  1.6400  1.6400
Atoms.UnitVectors>
```

最適化に関するkeywords:

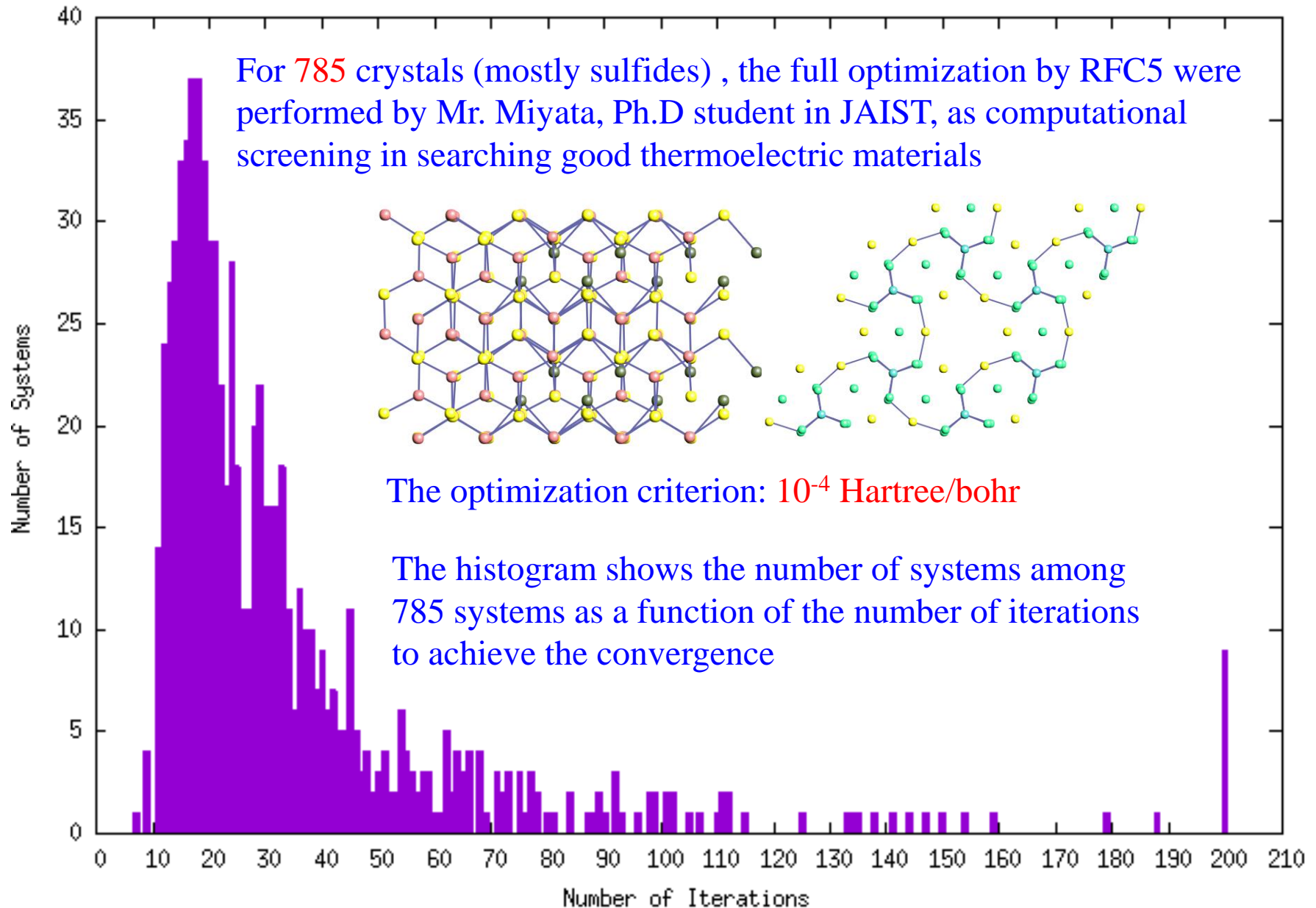
```
MD.Type          RFC5
MD.Opt.DIIS.History  3
MD.Opt.StartDIIS   7
MD.Opt.EveryDIIS  100000
MD.maxIter        100
MD.TimeStep       1.0
MD.Opt.criterion  0.0003
```



この計算はwork中のCdia-RF5.datで追試できる。

http://www.openmx-square.org/openmx_man3.8/node56.html も参照のこと

RFC5のベンチマーク計算



基底関数の指定

基底関数は以下の様に指定する。

```
<Definition.of.Atomic.Species
  O   O7.0-s2p2d1      O_PBE13
  H   H7.0-s2p1       H_PBE13
Definition.of.Atomic.Species>
```

- O7.0はO7.0.paoに対応する。
- -s2p2d1は動径関数として、s軌道を二つ、p軌道を2つ、d軌道を1つ選択することを意味する。
- この時、酸素Oに対して $2 \times 1 + 2 \times 3 + 1 \times 5 = 13$ 基底が割り当てられることになる。

O7.0.paoやO_PBE13.vpsの保存先は以下のkeywordで指定する。

```
DATA.PATH ../DFT_DATA13 # default=../DFT_DATA13
```

Default値は../DFT_DATA13である。

基底関数の選択: H₂Oの場合

Database(2013)においてH7.0.paoとO7.0.paoをクリックすると以下の情報が得られる。
http://www.jaist.ac.jp/~t-ozaki/vps_pao2013/H/index.html http://www.jaist.ac.jp/~t-ozaki/vps_pao2013/O/index.html

```
*****  
Eigen values(Hartree) of pseudo atomic orbitals  
*****
```

H7.0.pao

```
Eigenvalues  
Lmax= 3 Mul=15  
| mu 0 0 -0.23595211038442  
| mu 0 1 0.14109389991827  
| mu 0 2 0.61751730037441  
| mu 0 3 1.31890671598573  
| mu 0 4 2.24052765608302  
| mu 0 5 3.37954791544661  
| mu 0 6 4.73488369825610  
| mu 0 7 6.30608874470710  
| mu 0 8 8.09282718517299  
| mu 0 9 10.09464035732420  
| mu 0 10 12.31085267019158  
| mu 0 11 14.74057314485273  
| mu 0 12 17.38277845742691  
| mu 0 13 20.23645090753857  
| mu 0 14 23.30073926597344  
| mu 1 0 0.10914684890465  
| mu 1 1 0.47776040452236  
| mu 1 2 1.06988680483686  
| mu 1 3 1.88261331124981  
| mu 1 4 2.91175885838084  
| mu 1 5 4.15601184789448  
| mu 1 6 5.61454131060210  
| mu 1 7 7.28681796296307  
| mu 1 8 9.17254361476158  
| mu 1 9 11.27156766390586  
| mu 1 10 13.58381334569800  
| mu 1 11 16.10921637499960  
| mu 1 12 18.84767560575107  
| mu 1 13 21.79902110024685  
| mu 1 14 24.96300480798286  
| mu 2 0 0.27851528500170  
| mu 2 1 0.76970400958463  
| mu 2 2 1.47576814497054  
| mu 2 3 2.39881523735167  
| mu 2 4 3.53713346447754
```

↖3

↖6

↖5

```
*****  
Eigen values(Hartree) of pseudo atomic orbitals  
*****
```

O7.0.pao

```
Eigenvalues  
Lmax= 3 Mul=15  
| mu 0 0 -0.87913976280231  
| mu 0 1 0.06809061901229  
| mu 0 2 0.52709275865941  
| mu 0 3 1.24995140722317  
| mu 0 4 2.21829552402723  
| mu 0 5 3.41945468859267  
| mu 0 6 4.84509607059749  
| mu 0 7 6.48825090142865  
| mu 0 8 8.34207134316001  
| mu 0 9 10.39973244132192  
| mu 0 10 12.65513764955926  
| mu 0 11 15.10428688136984  
| mu 0 12 17.74677530362947  
| mu 0 13 20.58633940582683  
| mu 0 14 23.62957122674031  
| mu 1 0 -0.33075182895384  
| mu 1 1 0.16378499567753  
| mu 1 2 0.64129274864838  
| mu 1 3 1.35995471521821  
| mu 1 4 2.31377697411480  
| mu 1 5 3.50052618379026  
| mu 1 6 4.91841590421346  
| mu 1 7 6.56499793348396  
| mu 1 8 8.43634105410091  
| mu 1 9 10.52733540479795  
| mu 1 10 12.83276989780839  
| mu 1 11 15.34880171573570  
| mu 1 12 18.07414632417004  
| mu 1 13 21.01013110934429  
| mu 1 14 24.15933607166566  
| mu 2 0 0.26162948257116  
| mu 2 1 0.70705247436937  
| mu 2 2 1.34714706672243  
| mu 2 3 2.19799459356269  
| mu 2 4 3.26511989658328
```

↖1

↖4

↖2

↖7

↖8

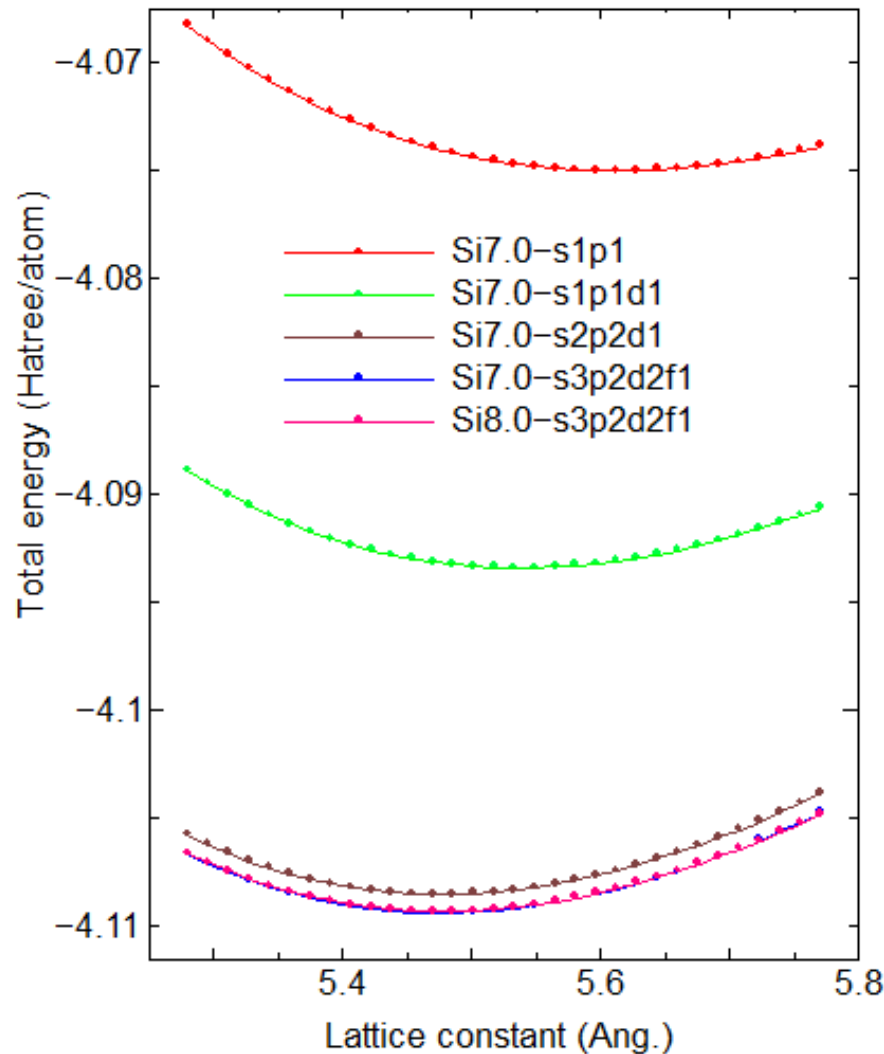
擬原子状態で低エネルギーの軌道から選択するとH7.0-s2p1, O7.0-s2p2d1となる。

基底関数の選択: Siの場合 (1)

Si7.0.pao

Eigenvalues		
Lmax= 3	Mul=15	
mu	0 0	-0.59320968623145
mu	0 1	0.01654247265872
mu	0 2	0.57915688461708
mu	0 3	1.39915528592998
mu	0 4	2.43106703909053
mu	0 5	3.63498884739274
mu	0 6	4.99885012002783
mu	0 7	6.55485759476897
mu	0 8	8.33541526908920
mu	0 9	10.34055461939939
mu	0 10	12.55818051232040
mu	0 11	14.98369777897096
mu	0 12	17.62014479571136
mu	0 13	20.47011746372617
mu	0 14	23.53245309073866
mu	1 0	-0.31460013133101
mu	1 1	0.12937640798489
mu	1 2	0.71637380083701
mu	1 3	1.54488697995006
mu	1 4	2.59211686526084
mu	1 5	3.84687324239366
mu	1 6	5.31246180826158
mu	1 7	6.99509799702661
mu	1 8	8.89454075291723
mu	1 9	11.00602102880752
mu	1 10	13.32692005826443
mu	1 11	15.85857455673626
mu	1 12	18.60272353142246
mu	1 13	21.55894843635971
mu	1 14	24.72602323954447
mu	2 0	0.01886411574821
mu	2 1	0.35893514996065
mu	2 2	0.94629918754692
mu	2 3	1.76201765644983
mu	2 4	2.81418723624388
mu	2 5	4.10656012645961
mu	2 6	5.63661971875011
mu	2 7	7.39522693820483
mu	2 8	9.37222098331867
mu	2 9	11.56227937764802
mu	2 10	13.96620568880690
mu	2 11	16.58651812641581
mu	2 12	19.42296574188659
mu	2 13	22.47308081363278
mu	2 14	25.73538272482773
mu	3 0	0.28356769151846
mu	3 1	0.79082836114569
mu	3 2	1.52992065308726
mu	3 3	2.52537261496132

Si7.0.pao中の固有エネルギーの低い方から順次、基底に含める。

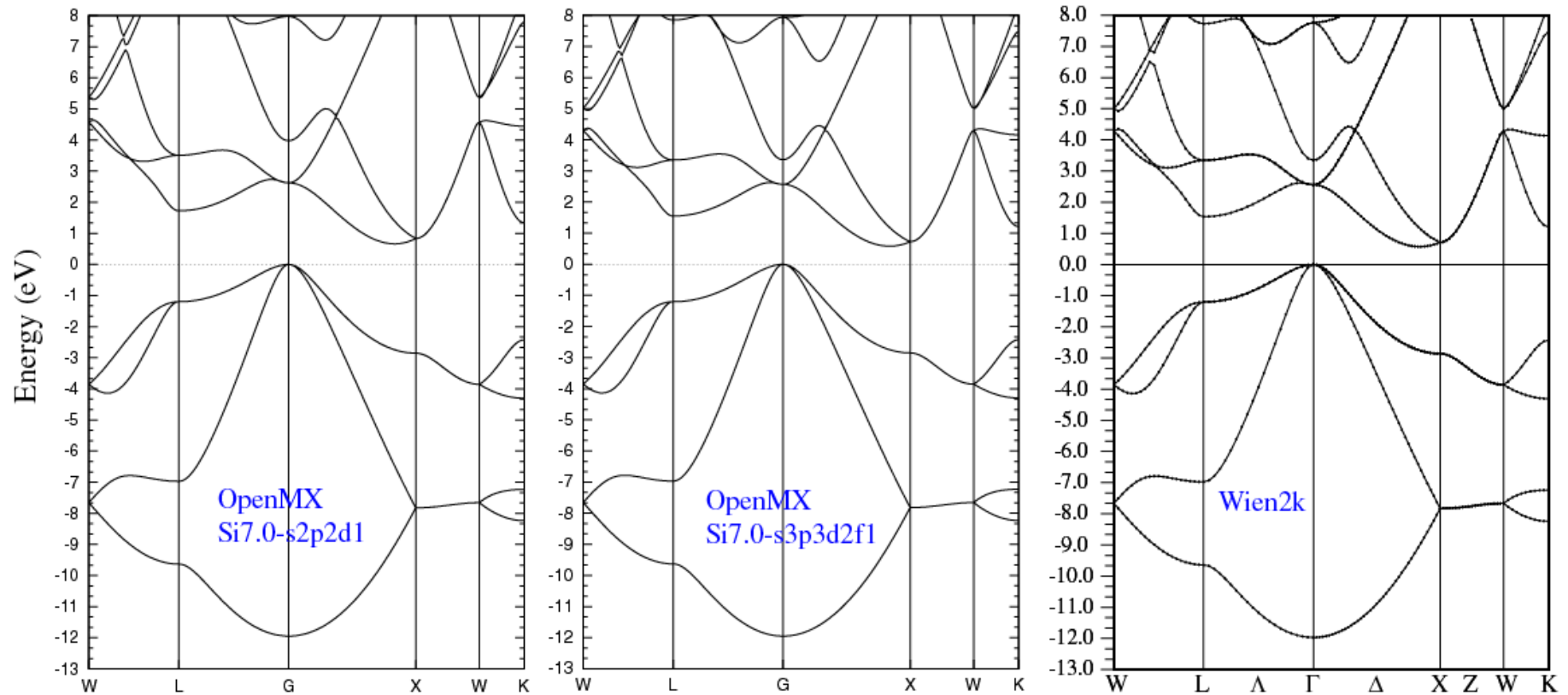


構造特性に関しては、Si7.0-s2p2d1で十分である。

Si7.0-s3p2d2f1とSi8.0-s3p2d2f1の比較から、カットオフ半径は7.0(a.u.)で収束していることが分かる。

基底関数の選択: Siの場合 (2)

バンド構造の比較からもSi7.0-s2p2d1でほぼ収束していることが分かる。



より完全な収束解を得るためにはSi7.0-s3p2d2f1(Si7.0-s3p3d2f1)を用いた方が良いが、大規模系を取り扱う場合や、広範囲な構造探索を行うためにはSi7.0-s2p2d1は精度とコストのバランスの取れた基底関数である。

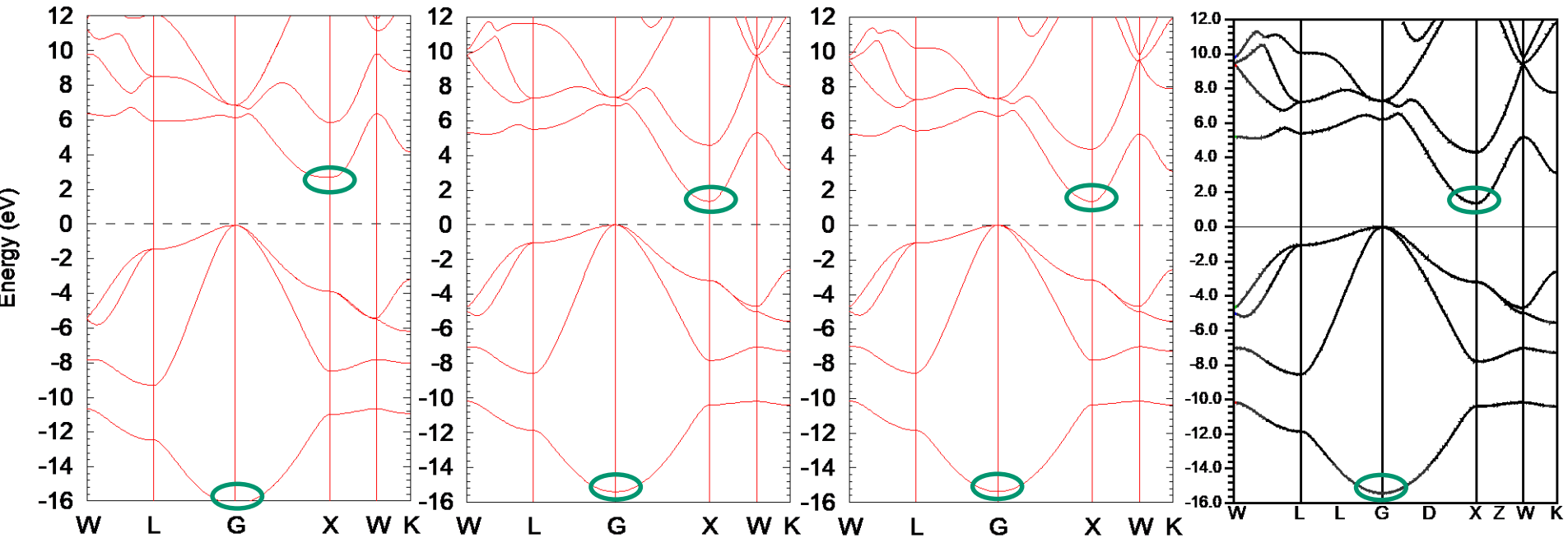
フローティング状態 in 3C-SiC

C6.0-s2p2
Si8.0-s2p2

C6.0-s2p2d1
Si8.0-s2p2d1

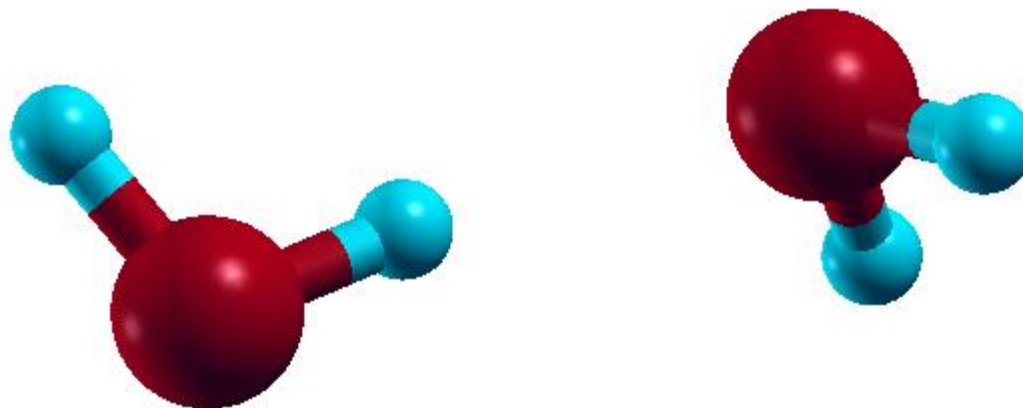
C6.0-s3p3d2f1
Si8.0-s3p3d2f1

Wien2k (FLAPW)



- ✓ d軌道を含めることが、バンド構造を再現する上で重要である。
- ✓ 少なくとも5eVまでのバンド構造はs2p2d1基底で再現されている。

基底関数重なり誤差 (BSSE)



	Equilibrium O-O distance (Ang.)	Dipole moment (Debye)	Binding energy (kcal/mol)	Binding energy (counterpoise corrected) (kcal/mol)
O7.0-s2p2d1, H7.0-s2p1	2.899	2.54	5.57	5.21
O7.0-s3p3d2, H7.0-s3p2	2.897	2.45	5.48	5.48
Other calc.	2.893 ^a		5.15 ^a	
Expt.	2.98 ^b	2.60 ^b	5.44 ^b	

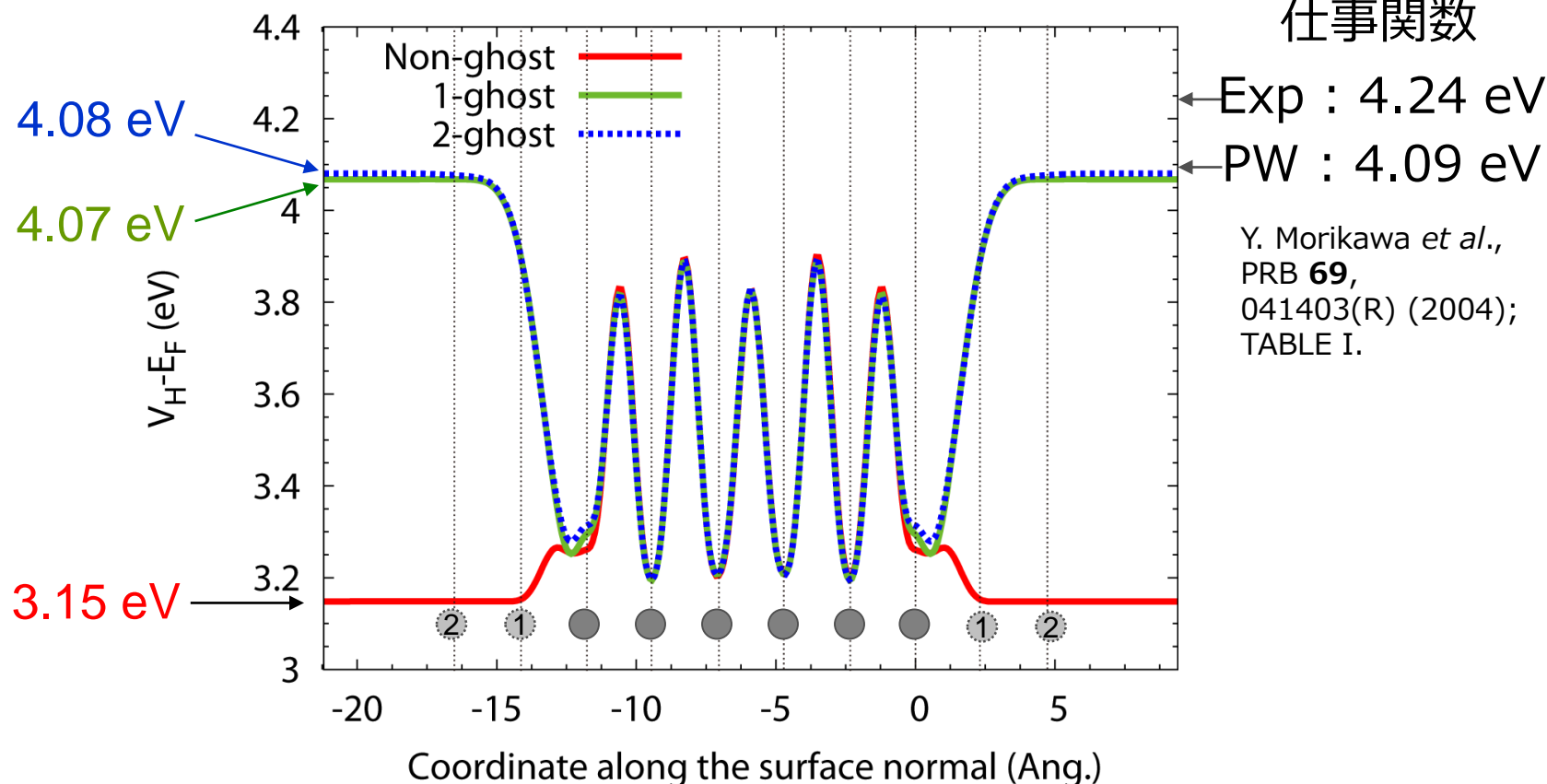
ベンチマーク計算から分子系に対するBSSEは ~ 0.5 kcal/mol

仕事関数の計算

fcc Al (111) surface

真空層にゴースト原子を置くことで仕事関数の高精度計算が可能である。
(本計算はDatabase(2006)の基底関数を用いて行われたため、
Database(2013)では状況が改善されている可能性がある。)

仕事関数



By Jippo and Ohfuti (Fujitsu)

Al(111) fcc 1x1 6L, GGA-PBE, Fix to the equilibrium lattice parameter
Al6.0-s1p2d1, Al_PBE (Database ver. 2006)
E6.0-s1p2d1, E (Database ver. 2006)

リスタート

- 計算が終了したにも関わらずSCFが収束していない場合、以下のkeywordを含めることで最終の電子密度分布を用いてSCF計算をリスタートすることが出来る。(ただしLDA+U法の場合には機能しない)

`scf.restart` `on` `# on|off,default=off`

- 構造最適化や分子動力学計算が途中で終了した場合、最終座標や最終速度を含んだ入力ファイル(*.dat#)が利用できる。このリスタート用の入力ファイルを用いて、計算を途中からリスタート出来る。

バイナリモードでの出力

- 大規模な計算を実行し、複数のcubeファイルを出力する場合、ファイルシステムの性能によっては長時間を要する場合がある。この場合、以下のキーワードを用いてバイナリモードで出力することで出力時間を低減することが可能である。

`OutData.bin.flag on # default=off, on|off`

- バイナリファイルはbin2txt.cを用いて、テキストファイルに変換できる。

コンパイル方法: `gcc bin2txt.c -lm -o bin2txt`

使用方法: `./bin2txt *.bin`