

RSDFT ユーザーガイド

Ver. 1.0

1	固体物理の基礎	5
1.1	原子単位系	5
1.2	単位胞	5
1.2.1	実格子ベクトル	5
1.2.2	逆格子ベクトルと <i>Brillouin</i> ゾーン	6
1.2.3	実空間グリッド	7
1.2.4	原子座標	8
1.3	BRILLOUIN ゾーン積分 (K 点サンプリング)	8
1.4	密度汎関数法	10
1.4.1	<i>Kohn-Sham</i> 方程式	10
1.4.2	全エネルギー	12
1.4.3	<i>Hellmann-Feynman</i> 力	14
1.4.4	背景電荷補正	15
2	実装済み機能一覧	17
2.1	交換相関汎関数	17
2.2	スピン	17
2.3	擬ポテンシャル	17
2.4	機能	17
2.5	境界条件	17
2.6	並列化	17
3	コンパイルと実行	19
3.1	MAKE.INC の編集	19
3.2	入力ファイル	20
3.3	実行	20
3.4	出力ファイル	21
3.5	パラメータファイル FORT.1 の形式	21
3.5.1	パラメーター一覧	22
3.6	原子座標ファイル FORT.970 の形式	32
3.7	初期スピン配置指定ファイル FORT.980 の形式	33
3.8	対称操作ファイルについて	33
3.9	分子・孤立系計算 (RSMOL) 用のパラメータ FORT.2	34

4	ユーティリティープログラム.....	35
4.1	原子座標作成	35
4.2	バンド図描画	35
4.3	原子座標, 密度・ポテンシャル, 波動関数の可視化.....	36
5	不具合.....	41

1 固体物理の基礎

電子状態計算に関連する基礎的事項について述べると同時に用語や記号の定義を行う。

1.1 原子単位系

Hartree 原子単位系 ($\hbar = m_e = e^2 = 1$) を採用する。ただしカットオフエネルギーだけは慣習に従い Rydberg 原子単位で表すことにする。

長さ	$1 (\text{bohr}) = 0.5291772083 (\text{\AA})$
エネルギー	$1 (\text{hartree}) = 27.21138344 (\text{eV})$ $1 (\text{Ry}) = 0.5 (\text{hartree}) = 13.60569172 (\text{eV})$
時間	$1 (\text{a.u.}) = 0.0241888433 (\text{fs})$

以下も単位換算等でよく用いられる。

ボルツマン定数	$0.08617342 (\text{meV/K})$
エネルギー	$1 (\text{kcal/mol}) = 0.04337 (\text{eV})$

1.2 単位胞

1.2.1 実格子ベクトル

結晶の単位胞の稜をなす 3 つのベクトルの x,y,z 成分を指定することで与える。

$$\mathbf{a}_1 = \begin{pmatrix} a_1^x \\ a_1^y \\ a_1^z \end{pmatrix} \quad \mathbf{a}_2 = \begin{pmatrix} a_2^x \\ a_2^y \\ a_2^z \end{pmatrix} \quad \mathbf{a}_3 = \begin{pmatrix} a_3^x \\ a_3^y \\ a_3^z \end{pmatrix} \quad (1.1)$$

実空間での任意の並進ベクトルは、整数 m_1, m_2, m_3 を用いて次のように書ける。

$$\mathbf{R} = m_1 \mathbf{a}_1 + m_2 \mathbf{a}_2 + m_3 \mathbf{a}_3 \quad (1.2)$$

1.2.2 逆格子ベクトルと Brillouin ゾーン

次のベクトルを逆格子ベクトルと呼ぶ。

$$\mathbf{b}_1 = 2\pi \frac{\mathbf{a}_2 \times \mathbf{a}_3}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)} \quad (1.3)$$

$$\mathbf{b}_2 = 2\pi \frac{\mathbf{a}_3 \times \mathbf{a}_1}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)} \quad (1.4)$$

$$\mathbf{b}_3 = 2\pi \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)} \quad (1.5)$$

Brillouin ゾーン (BZ) の体積は

$$\Omega_{BZ} = \mathbf{b}_1 \cdot (\mathbf{b}_2 \times \mathbf{b}_3) = \frac{(2\pi)^3}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)} = \frac{(2\pi)^3}{\Omega_{cell}} \quad (1.6)$$

となる。BZ (\mathbf{k} 空間と呼ぶ事もある) での体積要素は次のようになる

$$\Delta \mathbf{k} = \frac{\Omega_{BZ}}{N_{cell}} = \frac{(2\pi)^3}{N_{cell} \Omega_{cell}} = \frac{(2\pi)^3}{\Omega_{crystal}} \quad (1.7)$$

逆格子空間での並進ベクトルは、整数 n_1, n_2, n_3 を用いて

$$\mathbf{G} = n_1 \mathbf{b}_1 + n_2 \mathbf{b}_2 + n_3 \mathbf{b}_3 \quad (1.8)$$

と表せる。BZ 内の点は、実数 k_1, k_2, k_3 を用いて

$$\mathbf{k} = k_1 \mathbf{b}_1 + k_2 \mathbf{b}_2 + k_3 \mathbf{b}_3 \quad (1.9)$$

と表す。ベクトル \mathbf{k} は、ブロッホ関数の波数

$$\phi_{nk}(\mathbf{r}) = u_{nk}(\mathbf{r})e^{i\mathbf{k}\cdot\mathbf{r}} \quad (1.10)$$

であり、結晶中を（複数のユニットセルをまたいで）伝播する波の進行方向と波長（の逆数）の意味を持つ。ベクトル \mathbf{G} は、平面波展開

$$u_{nk}(\mathbf{r}) = \sum_{\mathbf{G}} c_{nk}(\mathbf{G})e^{i\mathbf{G}\cdot\mathbf{r}} \quad (1.11)$$

の基底関数の波数であり、ユニットセル内の範囲の波の詳細な（短波長の）変化を記述する。基底関数として採用する \mathbf{G} の大きさの最大値をカットオフ波数、あるいは単にカットオフと呼び、その 2 乗をカットオフエネルギーと呼ぶ。平面波展開による計算では

$$\begin{aligned} |\mathbf{G}| &\leq G_{cut} \\ |\mathbf{G}|^2 &\leq E_{cut} \quad (\text{Ry}) \end{aligned} \quad (1.12)$$

という範囲の波数を持つ平面波を基底関数として採用することになる。

1.2.3 実空間グリッド

フーリエ解析でよく知られているように、「エイリアシング（なりすまし）」による誤差を生じさせないという要請から、カットオフ波数が決まるとそれに対応する実空間グリッドの解像度が決まる。実空間法では逆に解像度、すなわちユニットセル内の空間を離散化するグリッド点の数を先に与える。グリッド点の数は、ユニットセルの一边を何分割するかで与えられる。計算の精度は分割数を増やす（刻み幅を小さくする）ことでコントロールする。これは平面波計算でカットオフエネルギーを上げていくことと等価である。カットオフと実空間グリッドの刻み幅 Δ は

$$G_{cut} \approx \frac{\pi}{\Delta} \quad (1.13)$$

という関係で結ばれている。有限の時間刻み幅で伝えることができる信号の周波数の最大値である「Nyquist 周波数」と同様のものである。RSDFT では、波動関数と密度・ポテンシャルに

同じグリッドを用いる。したがって平面波計算で十分なカットオフよりも少し大きめのカットオフに対応するグリッド幅をとる必要がある（グリッド幅に対する計算結果の収束性を常に意識するよう心がけること）。

1.2.4 原子座標

単位胞内の原子座標は、そのまま x, y, z 成分を指定する方法（xyz-表示）と、実格子ベクトルを基底として表す方法（ \mathbf{a} -表示または格子座標表示）がよく用いられる。

$$\begin{aligned}\mathbf{R}_{\text{ion}} &= X_{\text{ion}} \mathbf{e}_x + Y_{\text{ion}} \mathbf{e}_y + Z_{\text{ion}} \mathbf{e}_z \\ &= \alpha_{\text{ion}}^1 \mathbf{a}_1 + \alpha_{\text{ion}}^2 \mathbf{a}_2 + \alpha_{\text{ion}}^3 \mathbf{a}_3\end{aligned}\tag{1.14}$$

2 つの表示での座標成分は

$$\begin{pmatrix} X_{\text{ion}} \\ Y_{\text{ion}} \\ Z_{\text{ion}} \end{pmatrix} = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{pmatrix} \begin{pmatrix} \alpha_{\text{ion}}^1 \\ \alpha_{\text{ion}}^2 \\ \alpha_{\text{ion}}^3 \end{pmatrix}\tag{1.15}$$

という関係で結ばれている。

1.3 Brillouin ゾーン積分（ \mathbf{k} 点サンプリング）

固体物理の様々な場面で次の積分を評価する必要がある。

$$I = \int_{\Omega_{\text{BZ}}} f(\mathbf{k}) d\mathbf{k}\tag{1.16}$$

数値的には、これを BZ 内の有限個の \mathbf{k} 点の和で近似する。

$$I \approx \sum_{\mathbf{k}}^{\text{BZ}} w(\mathbf{k}) f(\mathbf{k}) \Delta \mathbf{k} \quad (1.17)$$

ゆえに、なるべく少ない \mathbf{k} 点数で積分の良い近似を得ることが求められる。近似精度を \mathbf{k} 点の選び方で向上させようとするのが **special points method** である。

通常 $f(\mathbf{k})$ として逆格子空間の周期関数を扱う。

$$f(\mathbf{k} + \mathbf{G}) = f(\mathbf{k}) \quad (1.18)$$

周期関数なので、フーリエ級数表示を用いて次のようにも表せる。

$$f(\mathbf{k}) = \sum_{\mathbf{R}} f(\mathbf{R}) e^{i\mathbf{k} \cdot \mathbf{R}} \quad (1.19)$$

このとき BZ 積分(1.16)は

$$I = \sum_{\mathbf{R}} f(\mathbf{R}) \int_{\Omega_{\text{BZ}}} e^{i\mathbf{k} \cdot \mathbf{R}} d\mathbf{k} = \Omega_{\text{BZ}} f(\mathbf{R} = 0) \quad (1.20)$$

となる。次に離散和(1.17)による計算を考える。もし $\mathbf{R} \neq 0$ の全ての実格子ベクトルに対して

$$\sum_{\mathbf{k}}^{\text{BZ}} w(\mathbf{k}) e^{i\mathbf{k} \cdot \mathbf{R}} = 0 \quad (1.21)$$

となる \mathbf{k} 点のセットがあったとすれば、離散和(1.17)で厳密な積分の結果を再現できることになる。しかし有限個の \mathbf{k} 点でこれを実現するのは不可能なので、現実的な方策として「なるべく多くの実格子ベクトルに対して」(1.21)を満たす有限個の \mathbf{k} 点のセットを用意するという近似を行うことになる。このとき、(1.19)の右辺のうち、(1.21)を満たさない実格子ベクトルを含む項が誤差となる。

金属的な系の場合には、(1.16)のような BZ 全域にわたる積分だけでなく、

$$\varepsilon(\mathbf{k}) = \varepsilon_F \quad (1.22)$$

で定義される曲面 (フェルミ面) の内側を積分範囲とする

$$I = \int_{\varepsilon(\mathbf{k}) \leq \varepsilon_F} f(\mathbf{k}) d\mathbf{k} \quad (1.23)$$

という計算も必要になる。このタイプの積分の数値計算には **smearing** 法やテトラヘドロン法が用いられる。

1.4 密度汎関数法

この節では計算の基礎理論となる密度汎関数法 (DFT = Density Functional Theory) についての簡単な説明を行う。

1.4.1 Kohn-Sham 方程式

スピン自由度を考慮しない場合の方程式を考える。解くべき方程式は

$$\left(-\frac{1}{2} \nabla^2 + v_{loc}(\mathbf{r}) + \hat{v}_{nloc}^{pp} \right) \phi_n(\mathbf{r}) = \varepsilon_n \phi_n(\mathbf{r}) \quad (1.24)$$

というものである。ここで、局所ポテンシャル

$$v_{loc}(\mathbf{r}) = v_{hartree}(\mathbf{r}) + v_{xc}(\mathbf{r}) + v_{loc}^{pp}(\mathbf{r}) \quad (1.25)$$

は、Hartree ポテンシャル、XC ポテンシャル、局所擬ポテンシャルの和であり、Hartree および XC ポテンシャルは電子密度 (あるいは Kohn-Sham 軌道) に依存する。非局所ポテンシャルは、通常は擬ポテンシャルのみから来るもので、一般には

$$\hat{v}_{nloc}^{pp} \phi_n(\mathbf{r}) = \int d\mathbf{r}' v_{nloc}^{pp}(\mathbf{r}, \mathbf{r}') \phi_n(\mathbf{r}') \quad (1.26)$$

という演算子として定義できる。ノルム保存擬ポテンシャルに分離型近似を用いた場合

$$\hat{v}_{nloc}^{pp} \phi_n(\mathbf{r}) = \sum_{a=1}^{N_{atom}} \sum_{l=0}^{l_{max}} \sum_{m=-l}^l c_{alm} \beta_{alm}^{pp}(\mathbf{r}) \int d\mathbf{r}' \beta_{alm}^{pp}(\mathbf{r}') \phi_n(\mathbf{r}') \quad (1.27)$$

という演算子となるが、これも形式的には(1.26)の形に書ける。

固体（周期系）の場合はブロッホ関数(1.10)を求める方程式

$$\left(-\frac{1}{2} \nabla^2 + v_{loc}(\mathbf{r}) + \hat{v}_{nloc}^{pp} \right) \phi_{n\mathbf{k}}(\mathbf{r}) = \varepsilon_{n\mathbf{k}} \phi_{n\mathbf{k}}(\mathbf{r}) \quad (1.28)$$

で考える事もできるが、RSDFT では周期関数部分 $u_{n\mathbf{k}}$ についての方程式

$$\left(-\frac{1}{2} \nabla^2 - i\mathbf{k} \cdot \nabla + \frac{k^2}{2} + v_{loc}(\mathbf{r}) + e^{-i\mathbf{k} \cdot \mathbf{r}} \hat{v}_{nloc}^{pp} e^{i\mathbf{k} \cdot \mathbf{r}} \right) u_{n\mathbf{k}}(\mathbf{r}) = \varepsilon_{n\mathbf{k}} u_{n\mathbf{k}}(\mathbf{r}) \quad (1.29)$$

を解くようになっている（差分化すると両式は等価でなくなることに注意）。規格化は

$$\int_{\Omega_{cell}} d\mathbf{r} |\phi_{n\mathbf{k}}(\mathbf{r})|^2 = \int_{\Omega_{cell}} d\mathbf{r} |u_{n\mathbf{k}}(\mathbf{r})|^2 = \langle u_{n\mathbf{k}} | u_{n\mathbf{k}} \rangle = 1 \quad (1.30)$$

としている。

以下、固体の場合を想定した式を書くが、 $\mathbf{k} = \mathbf{0}$ （ Γ 点）の場合を考えると分子の場合と全く同じ式となり、ほぼ並行に議論が進められる。

電子密度は

$$n(\mathbf{r}) = \sum_{n\mathbf{k}} f_{n\mathbf{k}} |\phi_{n\mathbf{k}}(\mathbf{r})|^2 = \sum_{n\mathbf{k}} f_{n\mathbf{k}} |u_{n\mathbf{k}}(\mathbf{r})|^2 \quad (1.31)$$

となる。ここで $f_{n\mathbf{k}}$ は占有数と呼ばれるもので、固有値の小さい軌道から順に電子が詰まるところを表すための数であり、

$$0 \leq f_{n\mathbf{k}} \leq 2 \quad (1.32)$$

$$\sum_{nk} f_{nk} = N_{electron}$$

を満たすようなものである。フェルミエネルギーを境に 0 or 2 を取るステップ関数

$$f_{nk} = 2\theta(\varepsilon_{nk} - \varepsilon_F) \quad (1.33)$$

とする場合と、これをより滑らかな関数で近似する

$$f_{nk} = 2f(\varepsilon_{nk} - \varepsilon_F) \quad (1.34)$$

という場合がある。後者を **smearing** 法と呼ぶ。Fermi-Dirac 分布関数や、誤差関数、エルミート多項式などが用いられる。

ポテンシャルが電子密度に依存しているために、(1.24)または(1.28)と、(1.25), (1.31) を自己無撞着に解く必要がある。これを、自己無撞着場 (SCF = Self-Consistent Field) を求める計算という意味で SCF 計算と呼ぶことがある。

1.4.2 全エネルギー

系の基底状態のエネルギーは次のように書ける。

$$E_{total} = T + E_{ee} + E_{ei} + E_{ii} \quad (1.35)$$

運動エネルギーは電子系の寄与のみであるとする (Born-Oppenheimer 近似)。最後の項はイオン (点電荷) 同士の相互作用のエネルギーで、周期系の場合は Ewald 和になる。

DFT では同じエネルギーを

$$E_{total} = T_s + E_{hartree} + E_{xc} + E_{ei} + E_{ii} \quad (1.36)$$

と表す。ここで xc 以外の項は、

$$T_s = \sum_{nk} f_{nk} \langle \phi_{nk} | -\frac{1}{2} \nabla^2 | \phi_{nk} \rangle \quad (1.37)$$

$$E_{hartree} = \frac{1}{2} \int_{\Omega_{cell}} d\mathbf{r} \int d\mathbf{r}' \frac{n(\mathbf{r})n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \quad (1.38)$$

$$E_{ei} = \int_{\Omega_{cell}} d\mathbf{r} v_{loc}^{pp}(\mathbf{r})n(\mathbf{r}) + \sum_{n\mathbf{k}} f_{n\mathbf{k}} \langle \phi_{n\mathbf{k}} | \hat{v}_{nloc}^{pp} | \phi_{n\mathbf{k}} \rangle \quad (1.39)$$

と明示的に表せる。二重積分の一方の積分範囲を Ω_{cell} としているのは単位胞あたりのエネルギーとするためである。分子の場合は Ω_{cell} は計算領域全体（すなわち全空間）であるとみなす。

xc 項は、電子密度あるいは Kohn-Sham 軌道の複雑な関数で、実際にはまず LDA や GGA といった近似関数形を選択してから具体的な計算を進めることになる。

規格化が一定値に保たれるという条件の下で(1.36)を最小化する軌道を求める。変分方程式として(1.28)の Kohn-Sham 方程式が得られる。このときポテンシャルは

$$\begin{aligned} v_{loc}(\mathbf{r}) &= \frac{\delta E_{hartree}}{\delta n(\mathbf{r})} + \frac{\delta E_{xc}}{\delta n(\mathbf{r})} + \frac{\delta E_{ei}}{\delta n(\mathbf{r})} \\ &= \int d\mathbf{r}' \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + \frac{\delta E_{xc}}{\delta n(\mathbf{r})} + v_{loc}^{pp}(\mathbf{r}) \\ &= v_{hartree}(\mathbf{r}) + v_{xc}(\mathbf{r}) + v_{loc}^{pp}(\mathbf{r}) \end{aligned} \quad (1.40)$$

となる。エネルギーを軌道について最小化することと、Kohn-Sham 方程式をセルフコンシステントに解くことは等価である。

SCF 反復の途中で全エネルギーを計算することを考える。ある与えられたポテンシャル $v_{hartree}^{in} + v_{xc}^{in}$ の下で計算を1ステップ進めた後得られた軌道を $\phi_{n\mathbf{k}}^{out}$ とし、この軌道で計算された密度を n^{out} とする。このとき全エネルギーは

$$E_{total} = \sum_{n\mathbf{k}} f_{n\mathbf{k}} \epsilon_{n\mathbf{k}} - \int_{\Omega} d\mathbf{r} \{ v_{hartree}^{in}(\mathbf{r}) + v_{xc}^{in}(\mathbf{r}) \} n^{out}(\mathbf{r}) + E_{hartree}[n^{out}] + E_{xc}[n^{out}] \quad (1.41)$$

となる。ここで $\epsilon_{n\mathbf{k}}$ は

$$\varepsilon_{\mathbf{nk}} = \left\langle \phi_{\mathbf{nk}}^{out} \left| -\frac{1}{2} \nabla^2 + v_{hartree}^{in} + v_{xc}^{in} + v_{loc}^{pp} + \hat{v}_{nloc}^{pp} \right| \phi_{\mathbf{nk}}^{out} \right\rangle \quad (1.42)$$

という期待値である。もちろん $\phi_{\mathbf{nk}}^{out}$ を用いて (1.36) の各項を全て計算し直してもよいが、通常 $\varepsilon_{\mathbf{nk}}$ は反復計算の過程で得られているはずなので、これを利用する方が軌道の数だけの内積計算を省略できる分、計算時間の節約になる。

また別種の全エネルギーの式として Harris の汎関数と呼ばれるものがある。

$$E_{total}^{Harris} = \sum_{\mathbf{nk}} f_{\mathbf{nk}} \varepsilon_{\mathbf{nk}} - \frac{1}{2} \int_{\Omega} d\mathbf{r} v_{hartree}^{in}(\mathbf{r}) n^{in}(\mathbf{r}) - \int_{\Omega} d\mathbf{r} v_{xc}^{in}(\mathbf{r}) n^{in}(\mathbf{r}) + E_{xc}[n^{in}] \quad (1.43)$$

これは SCF 反復の過程を経ず、1 ステップで収束解の良い近似となることを期待して作られたものである。収束解が得られていれば、(1.36), (1.41) および (1.43) は全て同じ値になる。

1.4.3 Hellmann-Feynman 力

原子(イオン)の配置を与えると擬ポテンシャルが定まり、エネルギー汎関数および Kohn-Sham 方程式が定まる。SCF 計算を実行すると、与えられた原子配置のもとでの電子系の基底状態エネルギーが得られる。様々な原子配置を試してエネルギーが最小となる構造を見出す事は、DFT の最も重要な応用の一つである。このような「原子構造最適化」を効率的に実行するためには、エネルギーの原子座標についての微分

$$\mathbf{F}_I = -\frac{\partial E_{total}}{\partial \mathbf{R}_I} \quad (1.44)$$

が必要となる。(1.36) および (1.37)~(1.40) を用いてこの微分を計算する。波動関数や密度は各構造での SCF 収束解を用いて計算すると、結果は

$$\mathbf{F}_I = -\int_{\Omega_{cell}} d\mathbf{r} \frac{\partial v_{loc}^{pp}(\mathbf{r})}{\partial \mathbf{R}_I} n(\mathbf{r}) - \sum_{\mathbf{nk}} f_{\mathbf{nk}} \left\langle \phi_{\mathbf{nk}} \left| \frac{\partial \hat{v}_{nloc}^{pp}}{\partial \mathbf{R}_I} \right| \phi_{\mathbf{nk}} \right\rangle \quad (1.45)$$

となる。波動関数や密度の原子座標微分が入ってこないのが (Hellmann-Feynman の定理)、この計算は大した手間をかけずに実行できる。

(1.44) は I 番目の原子に働く力とみなせるもので、この力を用いて分子動力学計算を実行することもある（第一原理分子動力学法）。

1.4.4 背景電荷補正

これは周期系で荷電状態の計算を行う場合にのみ考慮すべき事柄である。周期系ではユニットセル内で電荷中性でないと全エネルギーが発散する。これを回避するために、電子数の増減に応じ、空間的に一様な背景電荷を人為的に導入した計算を行う場合がある。このとき幾つかの付加項が現れる。電子との相互作用エネルギー

$$E_{ei} + E_{e-bg} = \int_{\Omega_{cell}} d\mathbf{r} v_{loc}^{pp}(\mathbf{r}) n(\mathbf{r}) + \int_{\Omega_{cell}} d\mathbf{r} n(\mathbf{r}) \int d\mathbf{r}' \frac{n_{bg}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \quad (1.46)$$

の右辺第二項から

$$\int_{\Omega_{cell}} d\mathbf{r} n(\mathbf{r}) \int d\mathbf{r}' \frac{n_{bg}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} = \frac{N_{electron} N_{bg}}{\Omega_{cell}} \lim_{G \rightarrow 0} \frac{4\pi}{G^2} \quad (1.47)$$

イオンとの相互作用エネルギー

$$\begin{aligned} E_{ii} + E_{i-bg} + E_{bg-bg} &= \frac{1}{2} \int_{\Omega_{cell}} d\mathbf{r} \int d\mathbf{r}' \frac{\{n_{ion}(\mathbf{r}) + n_{bg}(\mathbf{r})\} \{n_{ion}(\mathbf{r}') + n_{bg}(\mathbf{r}')\}}{|\mathbf{r} - \mathbf{r}'|} \\ &= E_{ewald} \\ &\quad + \int_{\Omega_{cell}} d\mathbf{r} \int d\mathbf{r}' \frac{n_{ion}(\mathbf{r}) n_{bg}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \\ &\quad + \frac{1}{2} \int_{\Omega_{cell}} d\mathbf{r} \int d\mathbf{r}' \frac{n_{bg}(\mathbf{r}) n_{bg}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \end{aligned} \quad (1.48)$$

の右辺第二項、三項から

$$\int_{\Omega_{cell}} d\mathbf{r} n_{ion}(\mathbf{r}) \int d\mathbf{r}' \frac{n_{bg}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} = \frac{N_{ion} N_{bg}}{\Omega_{cell}} \lim_{G \rightarrow 0} \frac{4\pi}{G^2} \quad (1.49)$$

および

$$\frac{1}{2} \int_{\Omega_{cell}} d\mathbf{r} \int d\mathbf{r}' \frac{n_{bg}(\mathbf{r})n_{bg}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} = \frac{1}{2} \frac{N_{bg}N_{bg}}{\Omega_{cell}} \lim_{G \rightarrow 0} \frac{4\pi}{G^2} \quad (1.50)$$

が現れる。これらの発散項は **Hartree** 項と局所ポテンシャル項の発散を打ち消すように働く。

点欠陥の荷電状態計算のような場合には、ユニットセルが電荷中性でも多極子によるスーパーセル間の相互作用が無視できない場合がある。このような場合の補正に関する議論も多数存在する。

2 実装済み機能一覧

2.1 交換相関汎関数

- 局所密度近似 (PZ81, PW92)
- 一般化勾配近似 (PBE96)
- 混成汎関数 (HSE, PBE0, HF)
- van der Waals 密度汎関数

2.2 スピン

- スピン依存計算 (ノンコリニアスピンは扱えない)

2.3 擬ポテンシャル

- ノルム保存擬ポテンシャル
- ウルトラソフト擬ポテンシャル (PSV 擬ポテンシャルのみ対応)

2.4 機能

- 原子構造最適化 (CG 法)
- バンド構造計算
- バンドアンフォールディング
- 荷電状態計算

2.5 境界条件

- 3次元周期境界条件 (このコードを「RSSOL」と呼ぶ)
- 0次元孤立系 (分子系) 境界条件 (このコードを「RSMOL」と呼ぶ)

2.6 並列化

- MPI による グリッド, バンド, k 点, スピン 並列
- OpenMP によるグリッド並列
- MPI と OpenMP を同時利用したハイブリッド並列

3 コンパイルと実行

3.1 make.inc の編集

ソースファイル群の中にある `make.inc.org` を `make.inc` という名前でコピーし、

- ◆ コンパイラ
- ◆ コンパイルオプション (`-cpp` 等のプリプロセッサオプションも含める)
- ◆ リンクオプション
- ◆ BLAS, LAPACK, ScaLAPACK
- ◆ プリプロセッサへ渡すキーワード

を指定する。`make.inc` 中の以下の項目を環境に合わせて編集する。幾つかのサンプルが用意されているので、該当する物がある場合はコメントを外すだけでよい。プリプロセッサを通すオプション (`-cpp` 等) は必須である

<code>FC =</code>	コンパイラ
<code>FFLAGS0 =</code>	コンパイルオプション
<code>LFLAGS0 =</code>	リンクオプション
<code>OMPFLAG =</code>	OpenMP を有効にする場合のオプション
<code>LAPACK_L =</code>	BLAS, LAPACK, ScaLAPACK へのリンク
<code>FFTW_L =</code>	FFTW へのリンク (使用する場合)
<code>INCLUDE_DIR =</code>	インクルードファイルへのパス
<code>CPP_FFLAGS =</code>	プリプロセッサへ渡すキーワード

プリプロセッサへ渡すキーワードには以下の物が指定可能である。

<code>-D_DRSDFT_</code>	倍精度実数版の実行モジュールを作成
<code>-D_FFTE_</code>	並列高速フーリエ変換 <code>FFTE</code> を使用
<code>-D_FFTW_</code>	高速フーリエ変換 <code>FFTW</code> を使用
<code>-D_LAPACK_</code>	ScaLAPACK を用いず、LAPACK のみ使用

現在、分子版 (RSMOL) 計算は倍精度実数版でのみ実行可能となっている。

3.2 入力ファイル

実行時に以下の入力ファイルが必要となる。

<code>fort.1</code>	実行に必要な種々のパラメータを記述
<code>fort.2</code>	分子版 (RSMOL) で実行するときのみ必要なパラメータ
<code>fort.970</code>	初期原子配置
<code>fort.980</code>	スピン計算実行時に初期スピン配置を指定するときのみ必要
擬ポテンシャル	ファイル名は <code>fort.1</code> に記述
<code>wf.dat1*</code>	前回の波動関数の計算結果を読み込む場合のみ必要
<code>vrho.dat1</code>	前回の密度・ポテンシャルの計算結果を読み込む場合のみ必要
<code>wopt.dat</code>	構造最適化の履歴を引き継いで計算再開する場合のみ必要

`fort.1`, `fort.970` および擬ポテンシャルファイルは常に必要である。

3.3 実行

入力ファイル (`fort.1` 等) が置いてあるディレクトリで、例えば

```
mpirun -np プロセス数 ./a.out
```

のように実行する。

3.4 出力ファイル

実行終了後、以下のようなファイルが生成される。構造最適化やバンド計算を実行したかどうかにより、必ずしも全てのファイルが出力されるわけではない。上述したように、いくつかのファイルはそのまま再計算を行う際の入力として必要になる。

<code>fort.99</code>	エネルギーや固有値等の最終結果
<code>fort.97</code>	構造最適化の線形探索中の原子座標
<code>fort.197</code>	線形探索終了後（CG 法 1 ステップ更新後）の原子座標
<code>wopt.dat</code>	構造最適化の履歴を引き継いで再計算するためのデータ
<code>band_eigv</code>	バンド計算結果（k 点, バンド, 固有値）
<code>band_dedk</code>	バンドの微分
<code>vrho.dat1*</code>	密度・ポテンシャルのデータ
<code>wf.dat1*</code>	波動関数のデータ

波動関数のデータは、1 つのファイル（`wf.dat1`）に全ての情報を書き出すモードと、MPI 並列の各ランクが担うデータを並列に書き出すモードが用意されており、後者の場合には `wf.dat1` には基本的なパラメータのみが格納され、波動関数のデータは、例えばランク 3 の担当分であれば `wf.dat1.00003` というファイルに出力される。

3.5 パラメータファイル `fort.1` の形式

入力パラメータは

```
KEYWORD      value1  value2  ...
```

という形式で与えるようになっている。省力した場合はデフォルト値が設定されるようになっているが、幾つかは必須の物もある。キーワードは大文字でも小文字でもよい。

複数の値が並ぶ場合、

```
KEYWORD      value1  value2  / value3 ...
```

のように書くと、同じ行の " / " 以降の値の読み込みを省略できる（これは Fortran の仕様である）。行末に常に " / " を書いておくことを推奨する。

3.5.1 パラメーター一覧

XCTYPE（交換相関汎関数）

交換相関汎関数を指定する。現在、

```
XCTYPE 'LDAPZ81'  
XCTYPE 'LDAPW92'  
XCTYPE 'GGAPBE96' ( or 'PBE' or 'PBE96' )  
XCTYPE 'PBEsol'  
XCTYPE 'revPBE'  
XCTYPE 'HSE' ( or 'HSE06' )  
XCTYPE 'HF'  
XCTYPE 'PBE0'  
XCTYPE 'VDWDF'
```

が指定可能である。

SYSTYPE（周期系/孤立系の選択）

周期系の計算か、孤立系の計算かを指定する。

```
SYSTYPE 0          (三次元周期系)  
SYSTYPE 1          (分子・孤立系(RSMOL))
```

分子・孤立系(RSMOL)の計算を行う場合には、入力ファイル `fort.2` も必要となる。また RSMOL は現在倍精度実数版でのみ実行可能となっている（3.1 参照）。

ISYM (対称性の利用)

結晶の対称性を利用するか否かを指定する（周期系計算の場合にだけ利用可能）。利用する場合（ISYM=1）の場合は、対称操作を記述するファイルを別途指定する必要がある。

ISYM	0 /	(対称性利用しない)
ISYM	1 'symmetry_file'	(対称性を利用する場合)

AX, A1, A2, A3 (実格子ベクトル)

ユニットセルの指定。以下は2原子Siダイヤモンド構造のユニットセルの例。

AX	10.261	(原子単位の格子定数)
A1	0.0 0.5 0.5	(\mathbf{a}_1 ベクトルの x,y,z 成分)
A2	0.5 0.0 0.5	(\mathbf{a}_2 ベクトルの x,y,z 成分)
A3	0.5 0.5 0.0	(\mathbf{a}_3 ベクトルの x,y,z 成分)

NGRID (実空間グリッド刻み数)

各格子ベクトル方向の空間刻みの数を指定。分子版(RSMOL)では無視される。

NGRID	12 12 12
-------	----------

MD (差分公式の次数)

差分公式の次数を指定。(2MD+1)-点公式を用いる。

MD	6
----	---

NBAND (バンド本数)

計算するバンド本数を指定。SCF 計算の場合電子数詰めるだけの本数よりも一割増程度をとっておくと収束性が向上する。

NBAND	16
-------	----

NSPIN (スピン数)

スピン自由度の指定。

NSPIN	1	(スピンなし計算)
NSPIN	2	(スピンあり計算)

NDSPIN (初期スピン差)

スピン電子数の差の初期値 (≥ 0.0 で指定する)。次の **NFIXED** パラメータで指定する回数だけこの差をキープして SCF 計算を行う。負の数を指定した場合、原子サイト毎の初期スピン配置を `fort.980` から読み込むモードになる。

NDSPIN	1.0	(アップとダウンの電子数差 1.0 に固定して計算を行う)
NSPIN	-1.0	(<code>fort.980</code> から初期スピン配置を読み込む)

NFIXED (初期スピン差をキープする反復回数)

NDSPIN で指定したスピン電子数差をしばらくキープして SCF 計算を行う。

NFIXED	10	(SCF 計算 10 反復の間スピン電子数差をキープ)
--------	----	-----------------------------

NEXTE (荷電状態)

外部から加える電子数を指定。正の数は負の荷電状態に対応する。

NEXTE	1.0	(電子数が中性よりも 1 個多い -1 荷の荷電状態の計算)
-------	-----	--------------------------------

NK (BZ 内のグリッド数)

BZ を NK^3 のグリッドに分割し、次の `mmm1` および `mmm2` でグリッド点のどこをサンプルするか指定する。

NK	8	(BZ を 8^3 のグリッドに分割する。座標は -4 ~ +4 ととる)
----	---	---

mmm1 (BZ 内のグリッド点のサンプル開始座標)

BZ を NK^3 のグリッドに分割し、サンプリングを座標 `mmm1` から開始する。座標の範囲を越える数字の指定は、周期境界条件で $-NK/2 \sim +NK/2$ の範囲の指定と解釈される。

mmm1	5	5	5	(BZ グリッドのサンプリング開始座標)
------	---	---	---	----------------------

mmm2 (BZ 内のグリッド点のサンプル間隔)

BZ を NK^3 のグリッドに分割し、サンプリングを座標 mmm1 から開始し、mmm2 点置きにサンプルする。

mmm2	2	2	2	(BZ グリッドの 2 点置き(1 つ飛ばし)でサンプルする)
------	---	---	---	---------------------------------

EKBT (smearing による BZ 積分の温度)

Methfessel-Paxton の方法による smearing による BZ 積分を行う際の温度。

EKBT	1.0d-5	(Hartree 原子単位)
------	--------	----------------

KINTEG (デルタ関数の展開)

Methfessel-Paxton の方法による smearing による BZ 積分を行う際の、デルタ関数をエルミート多項式で展開する際の次数。

KINTEG	5
--------	---

SWSCF (最初の SCF 計算を行うか否か)

読み込んだ初期原子座標 (fort.970) で SCF 計算を行うか否かを指定。

SWSCF	1	(行う)
SWSCF	0	(行わない)

DITER (SCF 反復のループ回数上限)

構造最適化時、バンド計算時用に別途指定可能。

DITER	100
-------	-----

NSWEEP (Sweep 計算のループ回数上限)

良い初期波動関数を得るために、初期ポテンシャルを固定した SCF 計算のようなものを実行し、

これを **Sweep** と呼んでいる。その **sweep** 計算時の反復回数の上限を指定。

NSWEEP	20
---------------	----

ICG (波動関数の更新アルゴリズム)

現在は共役勾配法によるレイリー商の最小化を採用している。以下の物が選択可能。

ICG	1	(Bylander-Kleinam-Lee の CG 法)
ICG	2	(LOBPCG 法)

IPC (波動関数の更新アルゴリズムの前処理)

CG 法の前処理として以下の物が選択可能。

IPC	1	(不完全な CG 反復)
ICG	2	(チューンされた不完全 CG)
ICG	4	(スレッド並列用不完全 CG)

MLOOP (不完全 CG 前処理の反復回数)

不完全 CG 前処理の反復回数上限。経験的に 3 くらいが良い。

MLOOP	3	(CG 法の前処理のための CG 法の反復回数)
--------------	---	--------------------------

NCG (波動関数更新用 CG の反復回数)

収束性と計算時間の兼ね合いで、経験的に 2 ~ 3 が良いように思われる。

NCG	2
------------	---

IMIX (密度/ポテンシャル混合)

SCF 収束のために、過去の反復で得たあるいはポテンシャルから外挿により次の入力を作る方法の指定。

IMIX	0	(密度で更新、単純混合)
IMIX	1	(ポテンシャルで更新、単純混合)
IMIX	10	(密度で更新、Pulay 法)
IMIX	11	(ポテンシャルで更新、Pulay 法)
IMIX	20	(密度で更新、Broyden 法)
IMIX	21	(ポテンシャルで更新、Broyden 法)

MMIX (密度/ポテンシャルの混合)

過去何回分のデータで外挿を行うか。

MMIX	4	(4 回前までの密度/ポテンシャルから次の入力を外挿)
------	---	-----------------------------

BETA (密度/ポテンシャルの混合)

BETA<1.0 のとき、外挿で作った物と現在の物をこの割合で混ぜて、最終的な次の入力とする。

BETA=1.0 のときは、計算中に調整しながら混ぜ率を決める。

BETA	1.0	(1.0 の場合は混ぜ率は自動調整になる)
------	-----	-----------------------

SCFCONV (SCF 収束判定条件)

ポテンシャルと密度それぞれで判定可能であるが、通常はポテンシャルのみで判定を行う。

SCFCONV	1.d-15	1.d-15	(ポテンシャル、密度の収束判定基準)
---------	--------	--------	--------------------

判定に用いる量は

$$\frac{1}{\Omega_{cell}} \int_{\Omega_{cell}} |f^{new}(\mathbf{r}) - f^{old}(\mathbf{r})|^2 d\mathbf{r} \quad (3.1)$$

である。

ETOTCONV (全エネルギーによる SCF 収束判定条件)

全エネルギーの誤差で SCF 計算の収束判定を行う

ETOTCONV	1.0d-12
----------	---------

FMAXCONV (力による SCF 収束判定条件)

SCF 反復計算途中で Hellman-Feynman 力を計算し、その最大値がほぼ一定値に収束した時点で SCF 反復を抜ける。

FMAXCONV	1.0d-5
-----------------	---------------

PP (擬ポテンシャルの種類とファイル名)

これを `fort.970` にある元素数の通し番号の順に元素数分書く。例えば SiC (2 元素) なら、

PP	2	'Si_psv.dat'	(データフォーマット、ファイル名)
PP	2	'C_psv.dat'	(データフォーマット、ファイル名)

のように書く。現在、読み込み可能な擬ポテンシャルデータフォーマットは以下の通りである。

PP	1	(TM フォーマット http://bohr.inesc-mn.pt/~jlm/pseudo.html)
PP	2	(PSV フォーマット)
PP	3	(YB フォーマット)
PP	4	(GTH, HGH フォーマット)
PP	5	(UPF フォーマット http://www.quantum-espresso.org/pseudopotentials)
PP	102	(PSV ウルトラスソフト擬ポテンシャル)

SWOPT (原子構造最適化)

原子構造最適化計算の制御。計算再開の場合は、前回の計算の最後に作られた `wopt.dat` が必要となる。

SWOPT	0	(構造最適化を行わない)
SWOPT	1	(構造最適化を実行する)
SWOPT	2	(構造最適化計算の再開。 <code>wopt.dat</code> が必要)

ATOMOPT1 (原子構造最適化)

共役勾配法による原子構造最適化計算の制御。CG 反復回数の上限、各 CG での線形探索のトライ数、CG のリフレッシュのタイミング。

ATOMOPT1	10	6	5	(CG 反復回数, 線形探索ステップ数, CG のリフレッシュ)
----------	----	---	---	----------------------------------

ATOMOPT2 (原子構造最適化)

共役勾配法による原子構造最適化計算の制御。「原子移動距離の上限、エネルギーの収束判定基準、力の収束判定基準、線形探索の収束判定基準」を指定。

ATOMOPT2	0.5	1.d-10	5.d-4	1.d-1	(原子移動最大許容値, エネルギー収束判定基準、力の収束判定基準、線形探索の収束判定基準)
----------	-----	--------	-------	-------	---

ATOMOPT3 (原子構造最適化中の SCF 反復回数)

原子構造最適化中の SCF 計算の反復回数上限。

ATOMOPT3	100
----------	-----

TIM (原子構造最適化における拘束条件)

このキーワードの次の行から 3 行ずつで、力の拘束条件を表す行列 (3×3) の要素を書く。原子座標 (fort.970) の 4 列目に拘束条件の番号を指定するパラメータがあり、その番号の最大値と同じ数だけ拘束行列を書く必要がある。省略すると単位行列が指定される。計算した原子毎の力に、指定した拘束行列を掛けた物を、実際に原子を動かす際の力として用いる。以下の例は z 成分のみを動かす場合の指定になっている。

TIM				
0.0	0.0	0.0	(拘束行列 C の Cxx, Cxy, Cxz 成分)	
0.0	0.0	0.0	(拘束行列 C の Cyx, Cyy, Cyz 成分)	
0.0	0.0	1.0	(拘束行列 C の Czx, Czy, Czz 成分)	

IC (ファイル I/O 制御)

計算の再開やバンド計算のための、波動関数や密度・ポテンシャルデータの読み込み。

IC	0	(何も読み込まない)
IC	1	(波動関数のみ読み込む)
IC	2	(密度&ポテンシャルのみ読み込む)
IC	3	(両方読み込む)

OC (ファイル I/O 制御)

波動関数や密度・ポテンシャルデータの書き出し。

OC	0	(何も書き出さない)
OC	1	(波動関数のみ書き出す)
OC	2	(密度&ポテンシャルのみ書き出す)
OC	3	(両方書き出す)
OC	4	(波動関数を単精度にして、波動関数のみ書き出す)
OC	5	(波動関数を単精度にして、両方書き出す)

OC2 (ファイル I/O 制御)

SCF 反復の何回に一回データを書き出すかを指定。

OC2	100	(100 反復毎に書き出しを行う)
-----	-----	-------------------

IOCTRL (ファイル I/O 制御)

並列計算時に波動関数データを書き出す場合、rank 0 に集めて書き出すか、rank 毎にそれぞれ書き出すか。rank 毎に書き出す場合は wf.dat1.00000 のように 5 桁の rank 番号を付けたファイル名にそれぞれ書き出す。

IOCTRL	0	(rank 0 が代表で書き出す)
IOCTRL	3	(各 rank がそれぞれ書き出す)

PROCS (MPI 並列数の指定)

グリッド、バンド、 \mathbf{k} 点、スピンの並列数を指定。全ての積が実行時に指定する MPI プロセス数に一致している必要がある。

```
PROCS    2  2  2  1  1  1    ( 空間を  $2 \times 2 \times 2$  で並列化する場合 )
```

ETLIMIT (経過時間の上限)

ここで指定した経過時間を過ぎたら、波動関数などのデータを書き出して計算を終了する。

```
ETLIMIT 86400.0    ( 24 時間 = 86,400 秒の指定 )
```

SWBAND (バンド計算実行の制御)

バンド計算を実行するか否かの指定。

```
SWBAND    0          (バンド計算を実行しない)
SWBAND    1          (バンド計算を実行する)
```

BAND (バンド計算のパラメータ)

キーワードとパラメータの次の行に \mathbf{k} 点と区間の刻み数も指定する。キーワードのある行には「バンドを描く区間の数」「計算に用いるバンド本数」「結果として取り出すバンド本数」「固有値による収束判定条件」「バンド計算中の反復回数上限」を指定する。その次の 3 行に、逆格子ベクトルを基底として \mathbf{k} 点区間の座標、

$$\mathbf{k} = k_1 \mathbf{b}_1 + k_2 \mathbf{b}_2 + k_3 \mathbf{b}_3 \quad (3.2)$$

を区間の数だけ書く。最後の行に各区間の刻み数を、やはり区間の数だけ指定する。入力例は 4 区間の場合である。

```

BAND      3  24  20  1.d-7  100
0.0  0.0  0.0  0.5
0.0  0.0  0.5  0.5
0.0  0.5  0.5  0.5
10  10  10  10

```

3.6 原子座標ファイル **fort.970** の形式

ユニットセルに含まれる元素数、原子数、原子番号、初期原子座標の指定。ユニットセル（格子ベクトル）の情報をこちらに書く事もでき、**fort.1** と両方に書いた場合は **fort.1** の方が優先される。例は 2 原子 Si ダイヤモンド構造の場合。

```

AX  10.261
A1  0.0  0.5  0.5
A2  0.5  0.0  0.5
A3  0.5  0.5  0.0
AA
1  2  14          (元素数、原子数、原子番号①, 原子番号②, ...)
1  0.00  0.00  0.00  1    (元素、座標,座標,座標、拘束条件)
1  0.25  0.25  0.25  1    (元素、座標,座標,座標、拘束条件)

```

このように書くと、原子座標は格子ベクトル表示で与えられたと解釈される。

AX	10.261			
A1	0.0	0.5	0.5	
A2	0.5	0.0	0.5	
A3	0.5	0.5	0.0	
XYZ				
1	2	14		(元素数、原子数、原子番号①, 原子番号②, …)
1	0.00000	0.00000	0.00000	1
1	2.56525	2.56525	2.56525	1

このように書くと原子座標 XYZ 表示で与えられたと解釈される。

3.7 初期スピン配置指定ファイル fort.980 の形式

原子座標ファイルに書いた順に原子に通し番号を付けたとして、その番号を用いて以下のよう
に指定する。

1	5	1.00	(1～5 番目の原子サイトでスピン電子数差 1.00 に)
6	6	0.50	(6 番目の原子サイトでは電子数差 0.50 に)
11	12	1.00	(11～20 番目の原子サイトでは電子数差 1.00 に)

指定したスピン電子数差を再現するように初期電子密度を用意する。

3.8 対称操作ファイルについて

以下のようなフォーマットのデータを用意する。ファイル名は任意で、**fort.1** で指定するようになっている。

nsym nnp (対称操作数 , 並進の単位の数)

$R_{11} \ R_{12} \ R_{13} \ R_{21} \ R_{22} \ R_{23} \ R_{31} \ R_{32} \ R_{33}$	$t_1 \ t_2 \ t_3$	(対称操作行列 , 並進ベクトル)
--	-------------------	------------------------

...

...

対称操作の行列と並進ベクトルは、格子座標表示で与える。すなわち位置ベクトルを

$$\begin{aligned}\mathbf{r} &= x\mathbf{e}_x + y\mathbf{e}_y + z\mathbf{e}_z \\ &= \alpha\mathbf{a}_1 + \beta\mathbf{a}_2 + \gamma\mathbf{a}_3\end{aligned}\tag{3.3}$$

と表したときの (α, β, γ) に対する変換

$$\begin{pmatrix} \alpha' \\ \beta' \\ \gamma' \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} + \frac{1}{\text{nnp}} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}\tag{3.4}$$

を与える。

3.9 分子・孤立系計算 (RSMOL) 用のパラメータ fort.2

RSMOL では計算領域やグリッドの指定の仕方が周期系 (RSSOL) の場合とは異なる。以下のパラメータが追加で必要となる。

1	(球状領域)
0.30 20.0	(格子間隔 0.3bohr, 半径 20 bohr の球状計算領域)
1	(負荷バランスを考慮したグリッド分割を使用)

3 行目「負荷バランスを考慮した空間分割」に 2 を指定すると、通信コストは増えるが負荷バランスがさらに良くなる分割が行われる。

RSMOL の場合、グリッドの解像度は fort.2 の格子間隔だけで決定される。また計算の収束性については、格子間隔の他に計算領域の大きさ (遠方で十分 0 となるように) にも留意する必要がある。

4 ユーティリティープログラム

`utility/` にあるプログラムの機能を説明する。

4.1 原子座標作成

sol_conf

ダイヤモンド構造の直方体のスーパーセルを作る。3つの整数 `nc1`, `nc2`, `nc3` を標準入力から読み込んで、 $8 \times nc1 \times nc2 \times nc3$ 原子のスーパーセルの原子座標を `fort.970` に書き出す。コンパイルと実行のイメージは以下の通りである。

```
[iwata@kaki utility]$ ifort sol_conf.f90
[iwata@kaki utility]$ ./a.out
input size of the unit cell : nc1,nc2,nc3=
( # of atoms is 8*nc1*nc2*nc3 )
11 11 11
# of nodes =          1
# of atoms =        10648
# of grid points =    5451776
# of eigenstates      23426
memory estimation (GB) = 1903.07653808594
```

4.2 バンド図描画

band2gp

RSDFE でバンド計算を実行すると `band_eigv` というファイルに `k` 点と固有値の情報が書き出される。これから `gnuplot` 用のデータとスクリプトを作るプログラムである。コンパイルと実行のイメージは以下の通りである。

```
> ls
band_eigv
> ifort band2gp.f90
> ./a.out
> ls
> plot_band fort.2001 fort.2002 .....
> gnuplot
gnuplot> load 'plot band'
```

4.3 原子座標, 密度・ポテンシャル, 波動関数の可視化

cubegen_simple

ユニットセル (格子ベクトル) の情報も含む形式の原子座標ファイルを標準入力から読み込み、**cube** 形式のファイルを生成する。出力ファイル名は **tmp.cube** である。**cube** ファイルを **VMD** や **VESTA** 等の描画ソフトに読み込ませれば、原子座標の可視化が行える。実行イメージは以下の通りである。

```
> ifort cubegen_simple.f90
> ./a.out < fort.970
> ls
> fort.970 tmp.cube
```

cubegen_vrho

ユニットセル (格子ベクトル) の情報も含む形式の原子座標ファイル **fort.970** と、密度およびポテンシャルの **RSDFT** の出力 **vrho.dat1** から、密度、ポテンシャル、スピン密度の描画データを含む **cube** ファイルを生成する。実行イメージは以下の通りである。

```
> ifort cubegen_vrho.f90
> ls
fort.970 vrho.dat1
> ./a.out
```

出力ファイルは、スピン自由を考慮していない場合には

```
> ls
rho_1.cub vloc_1.cub
```

スピンを考慮した場合には

```
> ls
dspin_1.cub rho_1.cub rho_2.cub vloc_1.cub vloc_2.cub
```

となる。これらを VMD や VESTA 等に読み込ませれば、原子座標に重ねて密度やポテンシャルの等値面、等高線の描画が行える。

cubegen_wf

ユニットセルの情報を含む **fort.970** および、分割書き出しをしていない (**IOCTRL=1**) 波動関数のデータファイル **wf.dat1** より、波動関数の描画データも含む **cube** ファイルを生成する。以下のような **cubegen_wf** 専用の入力を用意する必要がある。

```
R          2          / 実('r'or'R')か複素('c'or'C')か、iflag_abs
wf.dat1     / 波動関数データファイル名
1           / サンプル k 点数 ( Nbzsm )
1           / スピン自由度 ( Nspin )
4           / 何本分の波動関数のデータを生成するか
1 1 1       / n1,k1,s1 (バンド,k点,スピンを指定)
2 1 1       / n2,k2,s2 (データ数4としたので、以下4つ分記述)
3 1 1       / n3,k3,s3
4 1 1       / n4,k4,s4
```

iflag_abs は、0 ならば波動関数の符号も含めたデータを書き出す（実数版のみ可）。
iflag_abs=1 ならば絶対値、2 ならば絶対値二乗の描画データを書き出す。実行のイメージは以下の通りである。

```
# ls
fort.970  input_cubgen_wf  wf.dat1
# ./a.out < input_cubgen_wf
# ls
wf_00001_001_1.cub  wf_00002_001_1.cub ...
(wf_の後の数字は バンド_k 点_スピンを表す)
```

cubegen_wf3

ユニットセルの情報を含む **fort.970** および、分割書き出しされた波動関数のデータ (**IOCTRL = 3**) から、波動関数描画データを含む **cube** ファイルを生成する。このプログラムは **RSDFT** 実行時と同じ並列数の並列ジョブとして実行する必要がある。専用の入力ファイルを **fort.3** という名前で用意する。ファイルの中身は **cubegen_wf** とほぼ同じで、最後に並列数を指定する一行を追加する。

```
R          2          / 実('r'or'R')か複素('c'or'C')か、iflag_abs
wf.dat1     / 波動関数データファイル名
1           / サンプル k 点数 ( Nbzsm )
1           / スピン自由度 ( Nspin )
4           / 何本分の波動関数のデータを生成するか
1 1 1       / n1,k1,s1 (バンド,k 点,スピンを指定)
2 1 1       / n2,k2,s2 (データ数 4 としたので、以下 4 つ分記述)
3 1 1       / n3,k3,s3
4 1 1       / n4,k4,s4
PROCS      2 2 2 1 1 1 / RSDFT 実行時と同じキーワードと
                        並列数を指定
```

出力ファイルも **cubegen_wf** と全く同じである。実行のイメージは以下の通りである。

```
# ls
fort.3  fort.970  wf.dat1  wf.dat1.00000  wf.dat1.00001 ...
# mpirun -np 8 ./cubegen_wf3
# ls
wf_00001_001_1.cub  wf_00002_001_1.cub ...
(wf_の後の数字は バンド_k 点_スピン を表す)
```


5 不具合

- fcc および bcc 格子の斜交座標系において、差分近似の演算子が Γ 点での対称性を壊しているように思われる。したがって、この場合、対称性を回復させるためにより細かいグリッドをとる必要がある。