

# FLAPW のポテンシャルの平面波成分の扱い

柳瀬 章

平成 13 年 10 月 24 日

## 1 はじめに

FLAPW のプログラム KANSAI94 は, 多くの物質に適用されて成果をあげている. 最近では単位胞に 32 個とか 40 個の原子を含む系に適用されはじめている. このような応用にさいして見つかった, 不都合な点を改良したので, 報告する. 改良した点はいづれもポテンシャルの平面波成分の扱いにかかわっている. それらは

- 対称性によってベースがうまく生成されない.
- ベースを処理するサブルーチンの計算時間が長い

の二点である.

## 2 ポテンシャルの平面波成分のベースの生成

FLAPW のポテンシャルは次の形をしている.

$$V(r) = V_{PW}(r) \text{ for outside of muffin-tin spheres} \quad (1)$$

$$= V_{SW}(r) \text{ for inside of muffin-tin spheres} \quad (2)$$

$$V_{PW}(r) = \sum_i A_i G_i(r) \quad (3)$$

$$V_{SW}(r) = \sum_j B_j F_j(r) \quad (4)$$

ここでの議論は平面波成分  $V_{PW}(r)$  についてである。  $G_i$  は対称化された平面波の線形結合で結晶の対称性を持っている。

$$G_i(r) = \sum C_K \exp(iK \cdot r). \quad (5)$$

ポテンシャルは実関数であるから、  $G_i(r)$  を実関数にしておけば、  $A_i$  は実数になる。

一般に平面波の対称化には、 TSPACE のサブルーチンが使われる。これらのサブルーチンでは

$$P_{\mu\nu}^i = \frac{d_i}{n_k} \sum_{\alpha \in P_k} D_{\mu\nu}^{i*}(\alpha|u_\alpha) O(\alpha|u_\alpha) \quad (6)$$

で表される射影演算子が使われる。もしこれを作用する種（たね）が既約表現  $i$  の  $\nu$  成分を含んでおればその表現の  $\mu$  成分が生成される。  $P_k$  は  $k$  点群、  $d_i$  は既約表現  $i$  の次元である。また  $n_k$  は  $k$  点群の位数である (TSPACE P-152)。この射影演算子には重要な性質

$$P_{\kappa\lambda}^i P_{\mu\nu}^j = \delta_{i,j} \delta_{\lambda,\mu} P_{\kappa,\nu}^i \quad (7)$$

がある。この式が係数なしで成り立つように (6) の係数が選ばれている。したがって全対称な演算子  $H$  の行列要素の計算で

$$N_\Psi^* N_\Phi (P_{\kappa\lambda}^i \Psi | H | P_{\mu\nu}^j \Phi) = N_\Psi^* N_\Phi \delta_{ij} \delta_{\kappa\mu} (\Psi | H | P_{\lambda\nu}^i \Phi) \quad (8)$$

の変形が使える。ここで  $N_\Psi, N_\Phi$  は規格化とか、後に述べる実関数化のための調整位相因子のような、射影演算子のあたえる以外の係数である。この係数は左辺にそのまま移行する。ほとんどの場合射影演算子でつくったベースは  $k$  点群の位数だけの項を持っている。上の式の左辺と右辺では計算の手間がこのファクターの分だけ減っている。ここでの議論でもこの公式の効用が使われることになる。

ポテンシャルのベースを生成する場合には、平面波の  $k$  ベクトルは逆格子ベクトルとすることができるので事情は簡単になる。射影演算子を逆格子ベクトルの平面波にかけると、

$$\begin{aligned} P_{\kappa\lambda}^i \exp[iK \cdot r] &= \frac{d_i}{n_k} \sum_{\alpha \in P_k} D_{\mu\nu}^{i*}(\alpha|u_\alpha) \exp[i\alpha K \cdot r] \\ &\times \exp[-i\alpha K \cdot u_\alpha] \end{aligned} \quad (9)$$

となる。さらにここで使われる射影演算子は結晶が持っている対称性の空間群の恒等表現に対応するものである。つまり  $d = 1$  で、既約表現行列  $D$  はすべて 1、  $k$  点群が空間群の点群  $P_G$  になるので (6) は

$$P_{1,1}^1 = \frac{1}{n_G} \sum_{\alpha \in P_G} O(\alpha|u_\alpha) \quad (10)$$

と簡単になる. ここで  $n_G$  は空間群の点群の位数である. さらに (9) も

$$P_{1,1}^1 \exp[iK \cdot r] = \frac{1}{n_G} \sum_{\alpha \in P_G} \exp[i\alpha K \cdot r] \exp[-i\alpha K \cdot u_\alpha] \quad (11)$$

となる. ここで  $\alpha K$  は  $K$  が特殊であれば同じものが現れるが, その繰り返しの数は一つのベースのなかでは等しい.  $\alpha K = K$  になるような  $\alpha$  の集合は  $P_G$  の部分群である. それを  $P_K$  とし, その位数を  $n_K$  とすると,

$$P_G = \alpha_1 (= \epsilon) P_K + \alpha_2 P_K + \cdots + \alpha_{(n_G/n_K)} P_K \quad (12)$$

と剰余類に展開できる. この展開の代表元  $\alpha_i$  がそれぞれ異なる  $K$  に変換し, その剰余類に属する元はそれと同じ  $K$  に変換する. このとき (11) の位相因子が働いて, 同じ  $K$  の項の位相が異なることがあるが, その場合には互いに打ち消しあって 0 になる. この理由は  $\alpha$  の周期を  $p$  とすると,  $u_\alpha$  が格子の周期の  $1/p$  になっているからである. 結局 (11) の各平面波の係数の絶対値はすべて同じで, そのベースを形成する平面波の数の逆数になる.

さて (11) で与えられるベースは必ずしも実関数にはならない. しかし, もし結晶が反転対称を持ち, 原点が反転中心にとってあれば, (11) があたえる基底は自動的に実関数になる. 反転  $I$  があれば, 必ずそれが  $K$  を  $-K$  に変換し,  $u_I$  が 0 であれば, 係数が互いに複素共役になることが保証される.

しかし結晶が反転対称を持たなければ, 一般には同じベースの中に  $K$  と  $-K$  がいないことになる. ここで「一般には」としたのは, 選ばれた種の平面波  $K$  が特殊であれば, 結晶が持つ反転以外の回転の操作で  $-K$  に移される場合があるからである. 自明な例は  $K = (0\ 0\ 0)$  の場合で, つくられたベースは単項で全ての空間群で実関数になる. あまり自明でない例としては, 空間群の点群が  $T$  または  $T_d$  で  $K = (1\ 1\ 0)$  の場合である. この場合にはベースは 12 個のこの型の平面波の線形結合になるが, (11) の最後の因子のあたえる位相で実関数になったり, 純虚数の関数になったりする. もちろん後者は虚数単位  $i$  で割れば実関数になる.

一つのベースの中に  $K$  と  $-K$  が含まれれば, そのベースの中ではすべての平面波でこのような組ができていくことは, 次のように証明される. 「 $\gamma K = -K$  となる  $\gamma$  が存在すると, 任意の  $\alpha K$  に対して  $\alpha\gamma K = -\alpha K$

となる.  $P_G$  は一つの点群であるから,  $\alpha$  と  $\gamma$  が  $P_G$  に含まれれば,  $\alpha\gamma$  も  $P_G$  に含まれる.」これで, 一つのベースの中に  $K$  と  $-K$  が全て組で含まれるか,  $K$  と  $-K$  が全く含まれていないかのどちらかであることが証明された.

一つのベースの中に  $K$  と  $-K$  が全て組で含まれており, その係数が互いに複素共役であればそのベースは実関数である. しかし非共型な空間群の場合には, これは必ずしも保証されていない. しかし適当な調整位相因子を掛けて実関数にすることができる. つまり  $K$  が特殊の場合で二種類のベースがあることになる. 前者をタイプ 1, 後者をタイプ 2 と呼ぶことにする. つぎの表は空間群  $P2_13$  に現れるタイプ 2 の例である.

ISPW=	2	NPW=	12	AK=	.103162427869D+01	INDPW=	-1
( 0 1 1)	(	.000000000000D+00	-	.833333333333D-01	)		
( 0 -1 -1)	(	.000000000000D+00	.	.833333333333D-01	)		
( 0 -1 1)	(	.000000000000D+00	.	.833333333333D-01	)		
( 0 1 -1)	(	.000000000000D+00	-	.833333333333D-01	)		
( 1 0 1)	(	.000000000000D+00	-	.833333333333D-01	)		
( -1 0 -1)	(	.000000000000D+00	.	.833333333333D-01	)		
( -1 0 1)	(	.000000000000D+00	-	.833333333333D-01	)		
( 1 0 -1)	(	.000000000000D+00	.	.833333333333D-01	)		
( -1 1 0)	(	.000000000000D+00	.	.833333333333D-01	)		
( 1 -1 0)	(	.000000000000D+00	-	.833333333333D-01	)		
( -1 -1 0)	(	.000000000000D+00	.	.833333333333D-01	)		
( 1 1 0)	(	.000000000000D+00	-	.833333333333D-01	)		

$T$  または  $T_d$  では  $u_\alpha$  が格子の周期の半分になる空間群しかないので, (11) の位相因子は 1 か  $-1$  になり. 結果が実関数か純虚数の関数になる. 上の例では純虚数のベースを虚数単位  $i$  で割って実関数にしてある.

しかし一般の空間群では,  $K$  と  $-K$  が一つのベースに含まれていても一般の複素関数になる場合がある. この場合には実関数になるような適当な位相因子を探さなければならない.

空間群  $P3_121$  で現れるこのような例を次にあげる. TSSLPW が与える 3 番目のベースは次の形をしている.

3	8	13	0	1.19343	.40825		
	8	1	(	0	1	1)/	1 ( .16667 .00000)
	9	2	(	0	-1	-1)/	1 ( -.08333 -.14434)

10	3	(	1	0	-1)	/	1	(	.16667	.00000)
11	4	(	1	-1	1)	/	1	(	-.08333	.14434)
12	5	(	-1	0	1)	/	1	(	-.08333	-.14434)
13	6	(	-1	1	-1)	/	1	(	-.08333	.14434)

このベースは調整位相因子  $\exp(\pi i/3)$  を掛けると

```

ISPW= 3 NPW= 6 AK= .102782111673D+01 INDPW= -1
( 0 1 1) ( .833333333333D-01 .144337567297D+00 )
( 0 -1 -1) ( .833333333333D-01 -.144337567297D+00 )
( 1 -1 1) ( -.166666666667D+00 .000000000000D+00 )
( -1 1 -1) ( -.166666666667D+00 .000000000000D+00 )
( -1 0 1) ( .833333333333D-01 -.144337567297D+00 )
( 1 0 -1) ( .833333333333D-01 .144337567297D+00 )

```

となる。これで  $K$  と  $-K$  の係数がたがいに複素共役になって実関数になっている。新しい SYMPW0 では調整位相因子は次の方法で見つけている。

- ベースの最初の  $K$  の係数の位相が 0 になるように、位相因子を全体に掛ける。
- つぎに  $-K$  の係数の位相を求めて、その半分の位相を持つ位相因子の逆数を調整位相因子とする。

この方法は「調整位相因子があるとすれば、これかその符号を変えてものしかない。」とは言えるが、何時でも実関数を導くという証明ができていない。新しい SYMPW0 ではこの方法で実関数にならないと、その時点でプログラム止めるようにしてある。もちろん今まで試した例では無事に通過している。

反転の対称性があるのに、原点が反転中心にとってない場合は、当然反転で  $K$  が  $-K$  に移されるから、調整位相因子を必要とする例になる。もちろんこの場合には適当な位相因子がかならず見つかって、実関数にすることができる。

このような例外的な  $K$  以外の種、つまりどの回転操作によっても  $-K$  に移らない  $K$  に射影演算子をかけると、複素関数のベースができる。この場合には  $-K$  を種にするとこれと直交して、しかも複素共役なベースができる。この二つを同じ位相で加えたものと、逆位相で加えて虚数単位  $i$  で割ったものの二つは、互いに直交する実関数のベースになる。ここに

も二種類のベースができる. 前者をタイプ 3, 後者をタイプ 4 と呼ぶことにする. つぎの表は空間群  $P_{213}$  に現れるタイプ 3 とタイプ 4 の組の例である.

ISPW=	3	NPW=	8	AK=	.126347654453D+01	INDPW=	4
(	1	1	1)	(	.125000000000D+00	.000000000000D+00)	
(	1	-1	-1)	(	.125000000000D+00	.000000000000D+00)	
(	-1	1	-1)	(	.125000000000D+00	.000000000000D+00)	
(	-1	-1	1)	(	.125000000000D+00	.000000000000D+00)	
(	-1	-1	-1)	(	.125000000000D+00	.000000000000D+00)	
(	-1	1	1)	(	.125000000000D+00	.000000000000D+00)	
(	1	-1	1)	(	.125000000000D+00	.000000000000D+00)	
(	1	1	-1)	(	.125000000000D+00	.000000000000D+00)	
ISPW=	4	NPW=	8	AK=	.126347654453D+01	INDPW=	-3
(	1	1	1)	(	.000000000000D+00	-.125000000000D+00)	
(	1	-1	-1)	(	.000000000000D+00	-.125000000000D+00)	
(	-1	1	-1)	(	.000000000000D+00	-.125000000000D+00)	
(	-1	-1	1)	(	.000000000000D+00	-.125000000000D+00)	
(	-1	-1	-1)	(	.000000000000D+00	.125000000000D+00)	
(	-1	1	1)	(	.000000000000D+00	.125000000000D+00)	
(	1	-1	1)	(	.000000000000D+00	.125000000000D+00)	
(	1	1	-1)	(	.000000000000D+00	.125000000000D+00)	

以上のようにして作られたベースはつぎの形をしている.

$$(type1) G_i(r) = P_{1,1}^1 \exp(iK \cdot r) \quad (13)$$

$$(type2) G_i(r) = \exp(i\eta) P_{1,1}^1 \exp(iK \cdot r) \quad (14)$$

$$(type3) G_i(r) = \text{Re}[P_{1,1}^1 \exp(iK \cdot r)] \\ = \frac{1}{2}[P_{1,1}^1 \exp(iK \cdot r) + P_{1,1}^1 \exp(-iK \cdot r)] \quad (15)$$

$$(type4) G_i(r) = \text{Im}[P_{1,1}^1 \exp(iK \cdot r)] \\ = \frac{1}{2i}[P_{1,1}^1 \exp(iK \cdot r) - P_{1,1}^1 \exp(-iK \cdot r)] \quad (16)$$

タイプ 1 は, タイプ 2 でたまたま  $\eta = 0$  の場合になっている. しかし空間群が共型であるかまたは, 反転  $I$  を含んでいて  $u_I = 0$  であれば, (つまり普通の状況では) タイプ 2 は現れない. タイプ 2 の特殊性を強調する意味で, あえて別のタイプとして扱っている.

新しく作った SYMPW0 では

COMMON/SPW96/INDPW(LMNSPW)

を加えて、タイプの情報を他のサブルーチンに渡すようにしてある。この配列の値が 0 のベースはタイプ 1, -1 のものはタイプ 2 である。この値が正のものはタイプ 3 で、その値で組になっているタイプ 4 のベースの番号を示している。また -1 以外の負の値のものはタイプ 4 で、その値の絶対値が相手のタイプ 3 のベースの番号である。

### 3 ポテンシャルの自己無撞着性のテスト

マフィンティン球の外側でのポテンシャルの自己無撞着性のテストのために、古いポテンシャル  $V^o$  と新しいポテンシャル  $V^n$  の間で次の平均二乗偏差が計算される。

$$\Delta V_{PW} = \sqrt{\frac{1}{V_{out}} \int_{out} dr |V_{PW}^o(r) - V_{PW}^n(r)|^2} \quad (17)$$

この式の積分は

$$\begin{aligned} I &= \int_{out} dr |V_{PW}^o(r) - V_{PW}^n(r)|^2 \\ &= \int_{out} dr \left| \sum_i (A_i^o - A_i^n) G_i(r) \right|^2 \\ &= \sum_{i,j} (A_i^o - A_i^n)(A_j^o - A_j^n) \int_{out} dr G_i^*(r) G_j(r) \end{aligned} \quad (18)$$

と変形される。実関数  $G_i$  は複素共役をとっても意味がないように見える。しかし  $G_i$  を構成している指数関数は複素関数であり、その係数も複素数である。つまり \* をつけて考えることで、 $K$  に対して  $-K$  がどこにあるかを探す手間を省くことができる。

(18) の積分

$$I_{i,j} = \int_{out} dr G_i^*(r) G_j(r) \quad (19)$$

は結晶構造とマフィンティン球の半径および  $K$  のカットオフの値が変わらなければ、ポテンシャルの値にはよらずいつも同じものが使えるので、KANSAI94 ではイテレーションのループの外のサブルーチン SPSPW0 で計算し COMMON/SPW97/ の OSPW(LMNSPW,LMNSPW) に格納している。またこの値は次の実行に使用できるように fort.21 に書き出されている。

この(19)は一見すっきりしているが、各  $G_i(r)$  が(5)の平面波の線形結合であることから膨大な計算量になる。たとえば InP に不純物として Lu を In に置換した系の計算で、単位胞に 16 個の原子をとり、波動関数のベースを 1759 になるようにとったカットオフで、ポテンシャルのベースの数 524、これを構成する平面波の数が 17857 になる。平面波の数が 1759 の 8 倍より大きくなっているのは、この場合空間群が  $Td^2$  で多くのタイプ 3、タイプ 4 のベースが含まれており、(15)(16) で同じ平面波が 2 度使われているからである。

$$G_i(r) = \sum_K C_K \exp(iK \cdot r)$$

の形を(19)に代入すると、

$$\begin{aligned} I_{i,j} &= \sum_{K,K'} C_K^* C_{K'} \int_{out} dr \exp[i(K' - K) \cdot r] \\ &= \sum_{K,K'} C_K^* C_{K'} \left\{ \int_{cell} dr \exp[i(K' - K) \cdot r] \right. \\ &\quad \left. - \sum_a \int_{|r-r_a| < R_a} dr \exp[i(K' - K) \cdot r] \right\} \\ &= \sum_{K,K'} C_K^* C_{K'} \{ V_{cell} \delta_{K,K'} \\ &\quad - \sum_a \exp[i(K' - K) \cdot r_a] V_a F(|K' - K| R_a) \} \end{aligned} \quad (20)$$

$$F(x) = \frac{3(\sin(x) - x \cos(x))}{x^3} \quad (21)$$

ここで、 $V_{cell}$  は単位胞の体積、 $r_a$  は原子 a の位置ベクトル、 $R_a$  はそのマフィンティン球の半径で、 $V_a$  はその体積である。

KANSAI94 ではベースの具体的な形をそのまま使って計算を実行している。さきほどの例では  $F(x)$  を  $17857 \times 17857 \times (\text{原子の種類の数})$  回、(20)の原子位置に対する位相因子の計算を  $17857 \times 17857 \times 16$  回計算することになる。これは使用する計算機にもよるが数時間の CPU 時間を消費する。

しかし(13)-(16)の形を積極的に使えば大幅に計算時間を節約できる。まず  $G_i(r)$  は実関数であるから  $I_{i,j}$  は実数である。また明らかに

$$I_{i,j} = I_{j,i} \quad (22)$$

である。また組になっているタイプ 3 の  $G_i$  とタイプ 4 の  $G_{i'}$  は

$$G_i(r) = \text{Re}[P_{1,1}^1 \exp(iK \cdot r)] \quad (23)$$



$$G_{i'}(r) = \text{Im}[P_{1,1}^1 \exp(iK \cdot r)] \quad (24)$$

であるので

$$I_{i,j} = \text{Re} \int_{out} dr [P_{1,1}^1 \exp(iK \cdot r)]^* G_j(r) \quad (25)$$

$$I_{i',j} = -\text{Im} \int_{out} dr [P_{1,1}^1 \exp(iK \cdot r)]^* G_j(r) \quad (26)$$

が成り立っている。

さらに射影演算子の性質 (8) を使うと

$$\int_{out} dr [P_{1,1}^1 \exp(iK \cdot r)]^* G_j(r) = \int_{out} dr \exp(-iK \cdot r) G_j(r) \quad (27)$$

となる。この式の右辺では計算すべき積分の数が (1 / 点群の位数) だけになっている。この式を使って  $I_{i,j}$  を計算する時には (14) と (16) に含まれる位相因子、つまり (8) の  $N_{\Psi}^*$  を忘れないようにしなければならない。

式 (22) と (27) を使えば反転対称がある系で、計算時間が略回転の対称性の数  $\times 2$  のファクターだけ節約できる。また反転対称がない系ではさらに (25) (26) を使って略 1/4 にすることができる。新しい SPSPW0 には以上のアルゴリズムが組み込まれている。

ここで議論した  $I_{i,j}$  は全エネルギーの計算にもそのまま使える。新しい ETOTPW ではこのことが組み込まれているため、イテレーションの各ステップで計算時間を気にすることなく、全エネルギーの計算が行なえるようになっている。

## 4 ポテンシャルの平面波成分の行列要素

FLAPW が使う波動関数はマフィンティン球の外側では平面波

$$\exp[i(k + K) \cdot r]$$

である。したがってポテンシャルの平面波成分

$$V_{PW}(r) = \sum_j A_j G_j(r)$$

の行列要素は

$$\begin{aligned} & \sum_j A_j \int_{out} dr \exp[-i(k + K') \cdot r] G_j(r) \exp[i(k + K) \cdot r] \\ &= \sum_j A_j \int_{out} dr \exp[i(K - K') \cdot r] G_j(r) \end{aligned} \quad (28)$$

とブリルアンゾーンの中にとった  $k$  によらなくなっている。KANSAI94 では  $k$  のループの外側でサブルーチン VPW000 を走らせて必要な  $K - K'$  についてこの値を計算して fort.20 のファイルに落している。従来計算されていた比較的簡単な系ではブリルアンゾーンの中にとられる  $k$  の数が多く、この VPW000 の計算時間は全体の中で比較的小さなウエイトを占めていた。

最近 FLAPW 法を半導体の不純物状態の計算のような大きな系に使用するようになり、問題が表面化してきた。この種の計算では単位胞を大きくとり、含まれる原子数も 20,30,40 個のオーダーになっている。また  $k$  の数も極端に少なく 1 点だけというのがむしろ普通になっている。もちろんこのような系では波動関数のベースにする平面波の数も数 1000 のオーダーになる。 $K - K'$  の大きさは波動関数の  $K$  の大きさの 2 倍の範囲、つまり数では 8 倍が必要になり、VPW000 が計算の主要部分を占めるようになっていた。

さてここで必要になる積分値は (27) の右辺と同じものであることが分かる。つまり前の節で述べた  $I_{i,j}$  が計算されていれば、あらためて計算する必要がないことになる。具体的には、 $K - K'$  が、どの  $G_i$  のどこに含まれているかを見る逆引きの配列を SYMPW0 で用意しておけば、どの  $i$  の  $I_{i,j}$  を使えばよいかガストレートにわかる。ここでどこにが必要になるのは、同じベースに含まれる  $K$  でも、そのどれを種にしたかで  $G_i$  の位相が変わり、したがって  $I_{i,j}$  の位相が変わる。逆にいえば計算されている  $I_{i,j}$  に、着目した  $K$  の平面波の係数の位相因子の複素共役の逆数（つまりそのままと同じ）を掛けたものが求める積分値である。同じ  $K$  の平面波がタイプ 3 とタイプ 4 の二回含まれる場合は、もちろん両方の寄与の和が求める積分値である。あたらしい VPW000 はこのアルゴリズムを使用している。

## 5 あとがき

以上の改造で FLAPW の計算時間は大幅に削減される。上にあげた例で一回のイテレーションの計算時間が 26056 sec であったものが、この新バージョンを使うと 3732 sec になっている。この新バージョンを使う上で次の点を注意する必要がある。

- 新しい SYMPW0 が生成するポテンシャルのベースは、前のものと順序および符号が異なることがあるので注意する。

- 以前のプログラムで作った fort.21 の中間結果のファイルは使えない。具体的には新しいものでは、対称行列の左下半分をファイルに落とすことにしている。またマフィンティン半径の違いをチェックするようにしている。
- ここで改造したサブプログラムは、一応問題が起きそうな空間群でテストをしてあるが、FLAPW の計算を実際に行なった例はまだそれほど多くない。ぜひ問題のあるなしにかかわらず、結果を著者まで報告していただきたい。
- 以前のプログラムでは空間群  $G$  が反転対称を含んでおれば、原点を反転中心にとり、反転に附属する結晶の周期でないベクトルを 0 になるように原点をとる必要があった。ここでは、この制限を取り除いてある。しかしこの機能にたよって、計算対象の物質の空間原子配置の理解をおろそかにしてはいけない。

入れ換えるサブルーチンは

SYMPW0, SPSPW0, VPW000, ETOTPW

である。これらは

/u/yanase/lib2/flapw/wknew/

に入れてある。この4個のサブルーチンは KANSAI94 と異なるアルゴリズムを使用してるが結果的には同じ機能になるようにしてある。したがって元のソースから対応するものを取り除いて入れ換えるだけで使用できる。一部だけの入れ換えも可能であるが、詳しくは著者に問い合わせていただきたい。

古いベースを用いて作ったポテンシャルを新しいベースに変換するルーチン VPWTRS も同じディレクトリーに入れておく予定である。