

# Representation and conversion of one wavefunction

(X. Gonze, Y. Suzukawa, M. Mikami)

March 10, 2012

## 1 Notations and theoretical considerations

\* A Bloch wavefunction characterized by a wavevector  $\mathbf{k}$  is such that

$$\psi_{\mathbf{k}}(\mathbf{r}) = u_{\mathbf{k}}(\mathbf{r})e^{i2\pi\mathbf{k}\cdot\mathbf{r}}$$

where  $u_{\mathbf{k}}(\mathbf{r})$  is periodic, that is

$$u_{\mathbf{k}}(\mathbf{r} + \mathbf{R}_{latt}) = u_{\mathbf{k}}(\mathbf{r})$$

where  $\mathbf{R}_{latt}$  is a vector of the real space lattice.

\* Representation by plane waves

$$\begin{aligned} u_{\mathbf{k}}(\mathbf{r}) &= \sum_{\mathbf{G}} c_{\mathbf{k}}(\mathbf{G})e^{i2\pi\mathbf{G}\cdot\mathbf{r}} \\ \psi_{\mathbf{k}}(\mathbf{r}) &= \sum_{\mathbf{G}} c_{\mathbf{k}}(\mathbf{G})e^{i2\pi(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}} \end{aligned}$$

\* Normalisation

$$\sum_{\mathbf{G}} |c_{\mathbf{k}}(\mathbf{G})|^2 = 1$$

\* For a spinor wavefunction, there is an additional variable, the spin  $\sigma$  that can take two values, that is  $\sigma = \uparrow$  (spin up) or  $\sigma = \downarrow$  (spin down)  
The following relations hold :

$$u_{\mathbf{k}}(\mathbf{r}, \sigma) = \sum_{\mathbf{G}} c_{\mathbf{k}}(\mathbf{G}, \sigma)e^{i2\pi\mathbf{G}\cdot\mathbf{r}}$$

$$\psi_{\mathbf{k}}(\mathbf{r}, \sigma) = \sum_{\mathbf{G}} c_{\mathbf{k}}(\mathbf{G}, \sigma) e^{i2\pi(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}}$$

$$\sum_{\sigma} \sum_{\mathbf{G}} |c_{\mathbf{k}}(\mathbf{G}, \sigma)|^2 = 1$$

## 2 Properties of the wavefunctions (scalar case)

\* For ground-state wavefunctions, there is the Schrödinger equation

$$H|\psi_{n\mathbf{k}}\rangle = \varepsilon_{n\mathbf{k}}|\psi_{n\mathbf{k}}\rangle$$

where

$H$  is the Hamiltonian operator  
 $n$  labels the state (or the band)  
 $\varepsilon_{n\mathbf{k}}$  is the eigenvalue

\* As the wavevector labelling of an eigenstate comes from the property

$$\psi_{\mathbf{k}}(\mathbf{r} + \mathbf{R}_{latt}) = e^{i2\pi\mathbf{k}\mathbf{R}_{latt}}\psi_{\mathbf{k}}(\mathbf{r})$$

in which  $\mathbf{k}$  can be replaced by  $\mathbf{k} + \mathbf{G}_{latt}$  where  $\mathbf{G}_{latt}$  is any reciprocal space lattice vector, we can choose the wavefunctions at  $\mathbf{k}$  and  $\mathbf{k} + \mathbf{G}_{latt}$  to be equal, or to make a linear combination of wavefunctions with the same energy. We introduce the notation “L.C.” when linear combinations are allowed when equating two wavefunction.

$$\psi_{n(\mathbf{k}+\mathbf{G}_{latt})}(\mathbf{r}) \stackrel{\text{L.C.}}{=} \psi_{n\mathbf{k}}(\mathbf{r})$$

When there is no specific reason to prefer a linear combination, the equality relation will be used. This is a choice of “gauge”. Note that a phase factor might be allowed in taking the linear combination.

\* The  $\mathbf{k} \leftrightarrow \mathbf{k} + \mathbf{G}_{latt}$  correspondence translates to

$$u_{n(\mathbf{k}+\mathbf{G}_{latt})}(\mathbf{r}) \cdot e^{i2\pi\mathbf{G}_{latt}\cdot\mathbf{r}} \stackrel{\text{L.C.}}{=} u_{n\mathbf{k}}(\mathbf{r})$$

$$c_{n(\mathbf{k}+\mathbf{G}_{latt})}(\mathbf{G} - \mathbf{G}_{latt}) \stackrel{\text{L.C.}}{=} c_{n\mathbf{k}}(\mathbf{G})$$

\* The time-reversal symmetry (non-magnetic case) of the Hamiltonian gives the following relation

$$\psi_{n\mathbf{k}}(\mathbf{r}) \stackrel{\text{L.C.}}{=} \psi_{n(-\mathbf{k})}^*(\mathbf{r})$$

$$u_{n\mathbf{k}}(\mathbf{r}) \stackrel{\text{L.C.}}{=} u_{n(-\mathbf{k})}^*(\mathbf{r})$$

$$c_{n\mathbf{k}}(\mathbf{G}) \stackrel{\text{L.C.}}{=} c_{n(-\mathbf{k})}^*(-\mathbf{G})$$

\* For the  $\mathbf{k}$  wavevectors that are half a reciprocal lattice vector ( $2\mathbf{k} = \mathbf{G}_{latt}$ ), there is a special relationship between coefficients of the wavefunction :

$$c_{n\mathbf{k}}(\mathbf{G}) \stackrel{\text{L.C.}}{=} c_{n(\mathbf{k}-\mathbf{G}_{latt})}(\mathbf{G} + \mathbf{G}_{latt}) \stackrel{\text{L.C.}}{=} c_{n(-\mathbf{k})}(\mathbf{G} + \mathbf{G}_{latt}) \stackrel{\text{L.C.}}{=} c_{n\mathbf{k}}^*(-\mathbf{G} - \mathbf{G}_{latt})$$

That is, coefficients at  $\mathbf{G}$  and  $-\mathbf{G} - \mathbf{G}_{latt}$  are related. This will allow to decrease by a factor of 2 the storage space for these specific  $\mathbf{k}$  points.

### 3 Properties of the wavefunctions (spinor case)

\* One must distinguish two classes of Hamiltonians :

- the Hamiltonian is spin-diagonal
- the Hamiltonian mixes the spin components

In the first class, one finds usual non-spin-polarized, non-spin-orbit Hamiltonians, in which case the spin up-spin up and spin down-spin down parts of the Hamiltonian are equal, as well as spin-polarized Hamiltonian when the magnetic field varies in strength but not in direction.

In the second class, one finds Hamiltonians that include the spin-orbit splitting as well as non-collinear spin systems.

In the first class, the wavefunctions can be made entirely of either spin-up components or spin-down components, and treated independently of those made of opposite spin. This corresponds to `nspol = 2`.

In the second class, one must stay with spinor wavefunctions. This corresponds to `nspinor = 2`.

These two classes are mutually exclusive. The possibilities are thus :

nspol	nspinor	
1	1	scalar wavefunctions
2	1	spin-polarized wavefunctions
1	2	spinor wavefunctions

### 4 Plane wave basis set sphere

\* In order to avoid dealing with an infinite number of plane waves  $\{e^{i2\pi(\mathbf{k}+\mathbf{G})r}\}$  to represent Bloch wavefunctions, one selects those with a kinetic energy lower than some cut-off  $E_{kin-cut}$ . The set of allowed  $\mathbf{G}$  vectors will be noted by  $\{\mathbf{G}_{\mathbf{k}, E_{kin-cut}}\}$

$$\mathbf{G}_{latt} \in \{\mathbf{G}\}_{\mathbf{k}, E_{\text{kin-cut}}} \quad \text{if} \quad \frac{(2\pi)^2(\mathbf{G}_{latt} + \mathbf{k})^2}{2} < E_{\text{kin-cut}}$$

Expressed in reduced coordinates :

$$\frac{(2\pi)^2}{2} \sum_{ij} (\mathbf{G}_{latt,i}^{red} + \mathbf{k}_i^{red}) \mathbf{G}_{ij}^{met} (\mathbf{G}_{latt,j}^{red} + \mathbf{k}_j^{red}) < E_{\text{kin-cut}}$$

\* The kinetic energy cut-off is computed from the input variables `ecut` and `dilatmx`, to give the effective value :

$$\text{ecut\_eff} = \text{ecut} * (\text{dilatmx}) ** 2$$

\* For "time-reversal  $\mathbf{k}$ -points" ( $2\mathbf{k} = \mathbf{G}_{latt}$ , see section 2), not all coefficients must be stored. A specific storage mode, governed by the input variable `istwfk` has been introduced for the following  $\mathbf{k}$  points:

$$\left(000\right), \left(00\frac{1}{2}\right), \left(0\frac{1}{2}0\right), \left(0\frac{1}{2}\frac{1}{2}\right), \left(\frac{1}{2}00\right), \left(\frac{1}{2}0\frac{1}{2}\right), \left(\frac{1}{2}\frac{1}{2}0\right), \left(\frac{1}{2}\frac{1}{2}\frac{1}{2}\right)$$

For these points, the number of  $\mathbf{G}$  vectors to be taken into account, is decreased by about a factor of 2.

For the  $\mathbf{G}$ 's that are not treated, the coefficients  $c_{n\mathbf{k}}(\mathbf{G})$  can be recovered from those that are treated, thanks to

$$c_{n\mathbf{k}}(\mathbf{G}) = c_{n\mathbf{k}}^*(-\mathbf{G} - \mathbf{G}_{latt})$$

\* The number of plane waves is `npw`

For `ipw = 1 .. npw`, the reduced coordinates of  $\mathbf{G}$  are contained in the array `kg`:

$$\text{these are integer numbers} \begin{cases} \mathbf{G}_1^{red} = \text{kg}(1, \text{ipw}) \\ \mathbf{G}_2^{red} = \text{kg}(2, \text{ipw}) \\ \mathbf{G}_3^{red} = \text{kg}(3, \text{ipw}) \end{cases}$$

This list of  $\mathbf{G}$  vectors is computed in the routine `kpgsph.f`.

[To be continued : explain the time reversed  $k$ -point structure]

## 5 FFT grid and FFT box

\* For the generation of the density from wavefunctions, as well as for the application of the local part of the potential, one needs to be able to compute

$\psi_{n\mathbf{k}}(\mathbf{r})$  or  $u_{n\mathbf{k}}(\mathbf{r})$  for a 3D-mesh of  $\mathbf{r}$ -points, extremely fast, from the values  $c_{n\mathbf{k}}(\mathbf{G})$ .

[note : spin up and spin down parts can be treated separately in this operation, so they do not need to be specified otherwise in this section 5.]

\* The FFT algorithm starts from values of a function

$$z(j_1, j_2, j_3) \text{ for } j_1 = 0 \cdots (N_1 - 1), j_2 = 0 \cdots (N_2 - 1), j_3 = 0 \cdots (N_3 - 1)$$

and compute fast the transformed

$$\tilde{z}(l_1, l_2, l_3) \text{ for } l_1 = 0 \cdots (N_1 - 1), l_2 = 0 \cdots (N_2 - 1), l_3 = 0 \cdots (N_3 - 1)$$

with

$$\tilde{z}(l_1, l_2, l_3) = \sum_{j_1, j_2, j_3} z(j_1, j_2, j_3) e^{i2\pi \left( \frac{j_1 l_1}{N_1} + \frac{j_2 l_2}{N_2} + \frac{j_3 l_3}{N_3} \right)}$$

\* We want, on a FFT grid, the values of  $u_{\mathbf{k}}(\mathbf{r})$  for

$$\begin{aligned} r_1^{red} &= \frac{0}{N_1}, \frac{1}{N_1}, \dots, \frac{N_1 - 1}{N_1} \left( = \frac{l_1}{N_1} \right) \\ r_2^{red} &= \frac{0}{N_2}, \frac{1}{N_2}, \dots, \frac{N_2 - 1}{N_2} \left( = \frac{l_2}{N_2} \right) \\ r_3^{red} &= \frac{0}{N_3}, \frac{1}{N_3}, \dots, \frac{N_3 - 1}{N_3} \left( = \frac{l_3}{N_3} \right) \end{aligned}$$

(the choice of  $N_1, N_2, N_3$  is not discussed here.)

Note that we do not want  $u_{\mathbf{k}}(r)$  everywhere : these specific values allow to use the FFT algorithm. The effect of  $G_1^{red}$  or  $G_1^{red} + N_1$  (or any value of  $G_1^{red}$  modulo  $N$ ) will be similar.

\*

$$\begin{aligned} u_{\mathbf{k}}(\mathbf{r}) &= \sum_{\mathbf{G}} c_{\mathbf{k}}(\mathbf{G}) e^{i2\pi \mathbf{G} \cdot \mathbf{r}} \\ &= \sum_{\mathbf{G}} c_{\mathbf{k}}(\mathbf{G}) e^{i2\pi (G_1^{red} r_1^{red} + G_2^{red} r_2^{red} + G_3^{red} r_3^{red})} \end{aligned}$$

Let us represent  $u_{\mathbf{k}}(\mathbf{r})$  by the segment **wf\_real** (1 : 2, 1 :  $N_1$ , 1 :  $N_2$ , 1 :  $N_3$ ) where the first index refer to the real or imaginary part and the three others to the integer values  $l_1 + 1, l_2 + 1, l_3 + 1$

Let us map the  $c_{\mathbf{k}}(\mathbf{G})$  coefficients on a similar segment **wf\_reciprocal**(1 : 2, 1 :  $N_1$ , 1 :  $N_2$ , 1 :  $N_3$ ) with a similar meaning of **wf\_reciprocal**(1 : 2,  $j_1 + 1, j_2 + 1, j_3 + 1$ ):

$$\begin{aligned} j_1 &= \text{mod}(\mathbf{G}_1^{red}, N_1) [\Rightarrow j_1 \in [0, N_1 - 1]] \\ j_2 &= \text{mod}(\mathbf{G}_2^{red}, N_2) \\ j_3 &= \text{mod}(\mathbf{G}_3^{red}, N_3) \end{aligned}$$

Then :

$$\begin{aligned} & \mathbf{wf\_real}(\cdot, l_1 + 1, l_2 + 1, l_3 + 1) \\ &= \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \sum_{j_3=0}^{N_3-1} \mathbf{wf\_reciprocal}(\cdot, j_1 + 1, j_2 + 1, j_3 + 1) \times e^{i2\pi(\frac{j_1 l_1}{N_1} + \frac{j_2 l_2}{N_2} + \frac{j_3 l_3}{N_3})} \end{aligned}$$

This is, up to the array indexing convention, precisely the operation done by the FFT algorithm.

\* For FFT efficiency (minimisation of cache conflicts), the arrays `wf_real` and `wf_reciprocal` are not dimensioned `wf(2, N1, N2, N3)`, but `wf(2, N4, N5, N6)` where

if  $N_1$  even,  $N_4 = N_1 + 1$ ; if  $N_1$  odd,  $N_4 = N_1$   
 if  $N_2$  even,  $N_5 = N_2 + 1$ ; if  $N_2$  odd,  $N_5 = N_2$   
 if  $N_3$  even,  $N_6 = N_3 + 1$ ; if  $N_3$  odd,  $N_6 = N_3$

## 6 Wavefunctions and spatial symmetries.

\* If some spatial symmetry operation commutes with the Hamiltonian :

$$[H, S_t] = 0$$

then

$$\begin{aligned} H|\psi\rangle &= \varepsilon|\psi\rangle \Rightarrow S_t H|\psi\rangle = \varepsilon S_t |\psi\rangle \\ &\Rightarrow H[S_t |\psi\rangle] = \varepsilon[S_t |\psi\rangle] \end{aligned}$$

$S_t |\psi\rangle$  is also an eigenvector, with the same eigenvalue as  $|\psi\rangle$ .  
 However its wavevector is different :

$$\begin{aligned} \psi_{n\mathbf{k}}(\mathbf{r} + \mathbf{R}) &= e^{i2\pi\mathbf{k}\mathbf{R}} \psi_{n\mathbf{k}}(\mathbf{r}) \\ \Rightarrow (S_t \psi_{n\mathbf{k}})(\mathbf{r} + \mathbf{R}) &= \psi_{n\mathbf{k}}((S_t)^{-1}(\mathbf{r} + \mathbf{R})) \\ &= \psi_{n\mathbf{k}}\left(\sum_{\beta} S_{\alpha\beta}^{-1}(r_{\beta} + R_{\beta} - t_{\beta})\right) \\ &= \psi_{n\mathbf{k}}\left(\sum_{\beta} S_{\alpha\beta}^{-1}(r_{\beta} - t_{\beta}) + \sum_{\beta} S_{\alpha\beta}^{-1}R_{\beta}\right) \\ &= \psi_{n\mathbf{k}}((S_t)^{-1}(\mathbf{r}) + \sum_{\beta} S_{\alpha\beta}^{-1}R_{\beta}) \end{aligned}$$

( $S_{\alpha\beta}^{-1}R_{\beta}$  must be a vector of the real space lattice if  $S_t$  leaves the lattice invariant)

$$\begin{aligned} &= e^{i2\pi \sum_{\alpha\beta} k_{\alpha} S_{\alpha\beta}^{-1} R_{\beta}} \psi_{n\mathbf{k}}((S_t)^{-1}(\mathbf{r})) \\ &= e^{i2\pi\mathbf{k}'\cdot\mathbf{R}} (S_t \psi_{n\mathbf{k}})(\mathbf{r}) \end{aligned}$$

where  $(\mathbf{k}')_\alpha = \sum_\beta S_{\beta\alpha}^{-1} k_\beta$   
 For a vector in the reciprocal space

$$(\mathbf{k}')_\beta = (S_t(\mathbf{k}))_\beta = \sum_\alpha S_{\beta\alpha}^{-1} k_\alpha$$

i.e. the inverse transpose of  $S_{\alpha\beta}$  is used.

\* The preceding result means

$$\begin{aligned} \psi_{n(S^{-1}, t\mathbf{k})} &\stackrel{\text{L.C.}}{=} (S_t \psi_{n\mathbf{k}})(\mathbf{r}) \\ &\stackrel{\text{L.C.}}{=} \psi_{n\mathbf{k}}\left(\sum_\beta S_{\alpha\beta}^{-1}(r_\beta - t_\beta)\right) \end{aligned}$$

$$\begin{aligned} \Rightarrow u_{n(S^{-1}, t\mathbf{k})}(\mathbf{r}) &\stackrel{\text{L.C.}}{=} e^{i2\pi \sum_{\alpha\beta} S_{\alpha\beta}^{-1} k_\beta r_\alpha} \stackrel{\text{L.C.}}{=} e^{i2\pi \sum_{\alpha\beta} k_\alpha S_{\alpha\beta}^{-1}(r_\beta - t_\beta)} \times u_{n\mathbf{k}}\left(\sum_\beta S_{\alpha\beta}^{-1}(r_\beta - t_\beta)\right) \\ \Rightarrow u_{n(S^{-1}, t\mathbf{k})}(\mathbf{r}) &\stackrel{\text{L.C.}}{=} e^{-i2\pi \sum_{\alpha\beta} k_\alpha S_{\alpha\beta}^{-1} t_\beta} u_{n\mathbf{k}}\left(\sum_\beta S_{\alpha\beta}^{-1}(r_\beta - t_\beta)\right) \\ \Rightarrow \sum_{\mathbf{G}} c_{n(S^{-1}, t\mathbf{k})}(\mathbf{G}) e^{i2\pi \mathbf{G} \cdot \mathbf{r}} &\stackrel{\text{L.C.}}{=} e^{-i2\pi \sum_{\alpha\beta} k_\alpha S_{\alpha\beta}^{-1} t_\beta} \sum_{\mathbf{G}'} c_{n\mathbf{k}}(\mathbf{G}') e^{i2\pi \sum_{\alpha\beta} G'_\alpha S_{\alpha\beta}^{-1}(r_\beta - t_\beta)} \\ \Rightarrow c_{n(S^{-1}, t\mathbf{k})}\left(\sum_\alpha G'_\alpha S_{\alpha\beta}^{-1}\right) &\stackrel{\text{L.C.}}{=} e^{-i2\pi \sum_{\alpha\beta} (k_\alpha + G'_\alpha) S_{\alpha\beta}^{-1} t_\beta} c_{n\mathbf{k}}(\mathbf{G}') \end{aligned}$$

This formula allows to derive coefficients  $c_n$  at one  $\mathbf{k}$  point from these at a symmetric  $\mathbf{k}$  point.

## 7 Conversion of wavefunctions [routine wfconv.f]

\* The aim is to derive the wavefunction corresponding to a set of parameters, from the wavefunction corresponding to another set of parameters. This set of parameters is made of :

- nspinor (1 if scalar wavefunction, 2 if spinor wavefunction)
- kpt (the  $\mathbf{k}$ -point)
- kg (the set of plane waves, determined by  $E_{\text{kin-cut}}$ ,  $\mathbf{G}^{\text{met}}$  and  $\mathbf{k}$ )
- istwfk (the storage mode)

\* Changing nspinor :

- from nspinor=1 to nspinor=2: the scalar wavefunctions are used to generate

two spinor wavefunctions

$$\begin{aligned} c(\mathbf{G}) &\rightarrow c_1(\mathbf{G}, \sigma) = \begin{cases} c(\mathbf{G}) & (\text{if } \sigma = \uparrow) \\ 0 & (\text{if } \sigma = \downarrow) \end{cases} \\ &\rightarrow c_2(\mathbf{G}, \sigma) = \begin{cases} 0 & (\text{if } \sigma = \uparrow) \\ c(\mathbf{G}) & (\text{if } \sigma = \downarrow) \end{cases} \end{aligned}$$

- from nspinor=2 to nspinor=1: this is conceptually not well defined, as the natural "inverse" of the previous recipe

$$c_1(\mathbf{G}, \sigma) \rightarrow c(\mathbf{G}) = c_1(\mathbf{G}, \uparrow)$$

will not lead to a normalized wavefunction.

One state out of two must be ignored also.

Despite this criticism, this natural procedure is followed in `wfconv.f`.

\* Changing `kpt`, from `kpt1(k1)` to `kpt2(k2)`

Suppose (no time-reversal use)

$$\begin{aligned} (k_2^{red})_\alpha &= (\Delta G^{red})_\alpha + \sum_\beta S_{\beta\alpha}^{red} k_{1,\beta}^{red} \text{ [see listkk.f]} \\ (G_2^{red})_\alpha &= -(\Delta G^{red})_\alpha + \sum_\beta S_{\beta\alpha}^{red} G_{1,\beta}^{red} \end{aligned}$$

According to the results in sections 2 and 6,

$$c_{n\mathbf{k}_1}(\mathbf{G}_1) = e^{-i2\pi \sum_\alpha (\mathbf{k}_1 + \mathbf{G}_1)_\alpha^{red} t_\alpha^{red}} c_{n\mathbf{k}_2}(\mathbf{G}_2)$$

or equivalently

$$c_{n\mathbf{k}_2}(\mathbf{G}_2) = e^{i2\pi \sum_\alpha (\mathbf{k}_1 + \mathbf{G}_1)_\alpha^{red} t_\alpha^{red}} c_{n\mathbf{k}_1}(\mathbf{G}_1)$$

If the time-reversal symmetry is used, we have instead

$$\begin{aligned} (k_2^{red})_\alpha &= (\Delta G^{red})_\alpha - \sum_\beta S_{\beta\alpha}^{red} k_{1,\beta}^{red} \text{ [see listkk.f]} \\ (G_2^{red})_\alpha &= -(\Delta G^{red})_\alpha - \sum_\beta S_{\beta\alpha}^{red} G_{1,\beta}^{red} \end{aligned}$$

which leads to

$$c_{n\mathbf{k}_2}(\mathbf{G}_2) = (e^{i2\pi \sum_\alpha (\mathbf{k}_1 + \mathbf{G}_1)_\alpha^{red} t_\alpha^{red}} c_{n\mathbf{k}_1}(\mathbf{G}_1))^*$$

The phase factor is computed in `ph1d3d.f`

The resulting function, at  $\mathbf{G}_1$  is placed in a FFT box in `sphere.f` (`iflag=1`)  
The conversion from  $\mathbf{G}_1$  to  $\mathbf{G}_2$  is made when reading the coefficients in the FFT box, to place them in `cnk2(G2)`, in `sphere.f` also (`iflag=-1`).

\* The change of `istwfk` is accomplished when using `sphere.f`, as the representation in the FFT box is a full representation, where all the non-zero coefficients are assigned to their  $\mathbf{G}$  vector, even if they are the symmetric of another coefficient.