

Wannier90 library for PySCF : pyWannier90

Hung Q. Pham

January 12, 2018

1 Overview

Wannier function (WF) is an important theoretical tool in solid state physics and chemistry due to its highly localized nature.[1] WF have many applications: (i) Chemical bonding analysis, (ii) Theory of polarization, (iii) Correlation electron treatment for periodic systems using WF as basis.

WF is highly nonunique and the construction of WF is tricky due to two reasons: (i) The gauge indeterminacy of Bloch orbitals and (ii) The degeneracy of the occupied bands at high-symmetry points in the Brillouin zones.

In 1997, Marzari and Vanderbilt [2] introduced a robust scheme to construct the WFs with a “maximal-localization” of resulting WF. This WFs are defined as maximally-localized wannier functions (MLWFs). Years later, Souza, Marzari, and Vanderbilt developed a disentangled procedure to deal with entangled Bloch states. These schemes were implemented in the Wannier90,[3] a Fortran90 code that has been actively maintained and updated over time.

pyWannier90 is a Python interface of Wannier90 for PySCF to compute the maximally-localized Wannier functions (MLWF) from crystalline orbitals. Unlike other interfaces for Wannier90, with pyWannier90 one can straightforwardly manipulate the resulting MLWFs for either analysis purposes or for using them as basic for electronic structure calculation. In general, pyWannier90 can be incorporated into any python-based codes. It can be used as a standalone code where one has to provide the eigenvalue, overlap, and projections (optional) matrices from an electronic package or it can be used in the library mode in PySCF package where the crystallographic and electronic structure is provided by PySCF.

2 Theory

For detailed theory of MLWFs, the readers are highly recommended to see the original papers of MV [2] and SMV [4].

2.1 The Maximally-localized Wannier functions (MLWF):

For periodic systems, crystalline orbitals $\psi_m^{\mathbf{k}}(\mathbf{r})$ are defined by two quantum numbers m and \mathbf{k} in comparison with its molecular counterpart $\psi_m(\mathbf{r})$ where only one is needed. In Gaussian-function-based formulation, the crystalline orbitals or molecular Bloch orbitals/states/functions (BFs) or simply Bloch orbitals/states/functions can be written as linear combinations of N atomic BFs $\phi_\mu^{\mathbf{k}}(\mathbf{r})$:

$$\psi_m^{\mathbf{k}}(\mathbf{r}) = \sum_{\mu=1}^N C_{\mu,m}^{\mathbf{k}} \cdot \phi_\mu^{\mathbf{k}}(\mathbf{r}) \quad (2.1)$$

The atomic BF is obtained via the Fourier Transform (FT) of the local Gaussian functions $\chi_\mu^{\mathbf{g}}(\mathbf{r})$:

$$\phi_\mu^{\mathbf{k}}(\mathbf{r}) = \frac{1}{\sqrt{M}} \sum_{\mathbf{R}} e^{i \cdot \mathbf{k} \cdot \mathbf{R}} \chi_\mu^{\mathbf{g}}(\mathbf{r}) \quad (2.2)$$

where $\chi_\mu^{\mathbf{R}}(\mathbf{r})$ is the local Gaussian at the cell brought by the \mathbf{R} vector and M is the number of unit cells included in the FT. One should be note that here the BFs are normalized to one unit cell.

A WF, $\omega_n^{\mathbf{R}}(\mathbf{r})$, at a unit cell defined by a \mathbf{R} vector is computed from N_k Bloch states as:

$$\omega_n^{\mathbf{R}}(\mathbf{r}) = \frac{V}{(2\pi)^3} \int_{FBZ} \left[\sum_m^{N_k} U_{m,n}^{\mathbf{k}} \psi_m^{\mathbf{k}}(\mathbf{r}) \right] \cdot e^{-i \cdot \mathbf{k} \cdot \mathbf{R}} \cdot d\mathbf{k} \quad (2.3)$$

$$= \frac{V}{(2\pi)^3} \int_{FBZ} \left[\sum_m^{N_k} U_{m,n}^{\mathbf{k}} \cdot \sum_{\mu=1}^N C_{\mu,m}^{\mathbf{k}} \cdot \phi_\mu^{\mathbf{k}}(\mathbf{r}) \right] \cdot e^{-i \cdot \mathbf{k} \cdot \mathbf{R}} \cdot d\mathbf{k} \quad (2.4)$$

$$= \frac{V}{(2\pi)^3} \int_{FBZ} \left[\sum_{\mu=1}^N \tilde{C}_{\mu,n}^{\mathbf{k}} \cdot \phi_\mu^{\mathbf{k}}(\mathbf{r}) \right] \cdot e^{-i \cdot \mathbf{k} \cdot \mathbf{R}} \cdot d\mathbf{k} \quad (2.5)$$

For a mesh with N_{kp} number of k-point, equation (2.5) becomes:

$$\omega_n^{\mathbf{R}}(\mathbf{r}) = \frac{1}{N_{kp}} \sum_k^{FBZ} \left[\sum_{\mu=1}^N \tilde{C}_{\mu,n}^{\mathbf{k}} \cdot \phi_\mu^{\mathbf{k}}(\mathbf{r}) \right] \cdot e^{-i \cdot \mathbf{k} \cdot \mathbf{R}} \quad (2.6)$$

The main task in computing the MLWFs is to seek for a $U_{m,n}^{\mathbf{k}}$ matrix that minimizing the spread function Ω :

$$\Omega = \sum_n \left[\langle \mathbf{r}^2 \rangle_n - \bar{\mathbf{r}}_n^2 \right] \quad (2.7)$$

In Wannier90, the spread function and its derivative are calculated numerically using a procedure in the Appendix B of ref [2]. Within the k-space, Ω has the form:

$$\Omega = \Omega_I + \Omega_{OD} + \Omega_D \quad (2.8)$$

$$= \frac{1}{N} \sum_{\mathbf{k}, \mathbf{b}} \omega_{\mathbf{b}} \left(N_k - \sum_{m,n} |M_{m,n}^{(\mathbf{k}, \mathbf{b})}|^2 \right) + \frac{1}{N} \sum_{\mathbf{k}, \mathbf{b}} \omega_{\mathbf{b}} \sum_{m \neq n} |M_{m,n}^{(\mathbf{k}, \mathbf{b})}|^2 + \frac{1}{N} \sum_{\mathbf{k}, \mathbf{b}} \omega_{\mathbf{b}} \sum_n \left(-\text{Im} \ln M_{m,n}^{(\mathbf{k}, \mathbf{b})} - \mathbf{b} \cdot \bar{\mathbf{r}} \right)^2 \quad (2.9)$$

where Ω_I , Ω_{OD} , Ω_D are the invariant, off-diagonal, diagonal term of Ω , respectively. Ω_I is invariant in the sense that it does not depend on the unitary transformation. N_k is the number of bands. \mathbf{b} is the vector that brings a k-point \mathbf{k} to its nearest neighbor $\mathbf{k} + \mathbf{b}$.

One can see that the spread function is a function of the overlap matrix $M_{m,n}^{(\mathbf{k}, \mathbf{b})}$ and it is the main quantity needed for the computation of the MLWFs. In particular, Wannier90 needs the eigenvalues matrix $\epsilon_{n\mathbf{k}}$, overlap matrix $M_{m,n}^{(\mathbf{k}, \mathbf{b})}$ and projection matrix $A_{m,n}^{\mathbf{k}}$ from an electronic structure package, e.g. PySCF. The next sections will discuss about the meaning and how to get the overlap matrix and projection matrix.

2.2 The overlap matrix:

The overlap matrix is computed using the periodic part $u_m(\mathbf{k}, \mathbf{r})$ of the Bloch function (BF):

$$M_{m,n}^{(\mathbf{k}, \mathbf{b})} = \langle u_m^{\mathbf{k}}(\mathbf{r}) | u_n^{\mathbf{k}+\mathbf{b}}(\mathbf{r}) \rangle \quad (2.10)$$

where: $u_m^{\mathbf{k}}(\mathbf{r}) = e^{-i \cdot \mathbf{k} \cdot \mathbf{r}} \cdot \psi_m^{\mathbf{k}}(\mathbf{r})$ and \mathbf{b} is the vector that brings \mathbf{k} to its nearest neighbor. $M_{m,n}^{(\mathbf{k}, \mathbf{b})}$ is a $N_k \times N_k$ matrix at each $(\mathbf{k}, \mathbf{k} + \mathbf{b})$ pair.

$$M_{m,n}^{(\mathbf{k}, \mathbf{b})} = \sum_{\mu=1, \nu=1}^N C_{\mu,m}^{\mathbf{k}\dagger} C_{\nu,n}^{\mathbf{k}+\mathbf{b}} \langle u_\mu^{\mathbf{k}}(\mathbf{r}) | u_\nu^{\mathbf{k}+\mathbf{b}}(\mathbf{r}) \rangle \quad (2.11)$$

where $u_\mu^{\mathbf{k}}(\mathbf{r})$ is the periodic part of the atomic BF, defined similarly to its molecular counterpart.

2.3 The projection matrix:

In the wannierizing procedure of Wannier90, a set of N_k ($\leq N$) reference Bloch orbitals is prepared by projecting a set of L (i.e the number of desired MLWFs) localized trial orbitals $g_n(\mathbf{r})$ onto N_k Bloch eigenstates used to construct the MLWFs:

$$\tilde{\psi}_n^{\mathbf{k}}(\mathbf{r}) = \sum_m^{N_k} |\psi_m^{\mathbf{k}}(\mathbf{r})\rangle c = \sum_m^{N_k} A_{m,n}^{\mathbf{k}} |\psi_m^{\mathbf{k}}(\mathbf{r})\rangle \quad (2.12)$$

where $A_{m,n}^{\mathbf{k}} = \langle \psi_m^{\mathbf{k}}(\mathbf{r}) | g_n(\mathbf{r}) \rangle$ is the $N_k \times L$ projection matrix at each k-point and $g_n(\mathbf{r})$ is a Gaussian function at a certain position in the reference cell.

$$A_{m,n}^{\mathbf{k}} = \int_0^\infty \psi_m^{\mathbf{k}\dagger}(\mathbf{r}) \cdot g_n(\mathbf{r}) \cdot d\mathbf{r} \quad (2.13)$$

$$= \sum_{\mu=1}^N C_{\mu,m}^{\mathbf{k}\dagger} \int_0^\infty \phi_\mu^{\mathbf{k}\dagger}(\mathbf{r}) \cdot g_n(\mathbf{r}) \cdot d\mathbf{r} \quad (2.14)$$

The projection functions $g_n(r)$ used in Wannier90 are given by:

$$g_n(r) = \Theta_{l,m_r}(\theta, \phi) \times R_r(\mathbf{r}) \quad (2.15)$$

where $\Theta_{l,m_r}(\theta, \phi)$ are real functions obtained by a unitary transformation of canonical spherical harmonics $Y_{l,m}(\theta, \phi)$ from the solution of Schrödinger equation for hydrogen atom; $R_r(\mathbf{r})$ is the radial functions with regard to different values of r . The mathematic forms of the $\Theta_{l,m_r}(\theta, \phi)$ and $R_r(\mathbf{r})$ are given in the Table 3.1-3.3 of the [User Guide](#) of Wannier90. In pyWannier90, these quantities are evaluated using a grid. Bloch states can also be used as initial guess when constructing $A_{m,n}^{\mathbf{k}}$.

3 Examples

3.1 CH₄ in a box

A CH₄ molecule in a 6.35 Å cubic box is computed at the PBE/STO-3G level of theory. The Brillouin zone is sampled at Γ point. All nine Bloch states from DFT calculation are used to compute four MLWFs. The projection matrix is calculated using four sp³ orbitals center at the C atom. Similar calculation is carried out by VASP for the sake of comparison. All input and output files can be found in the released package.

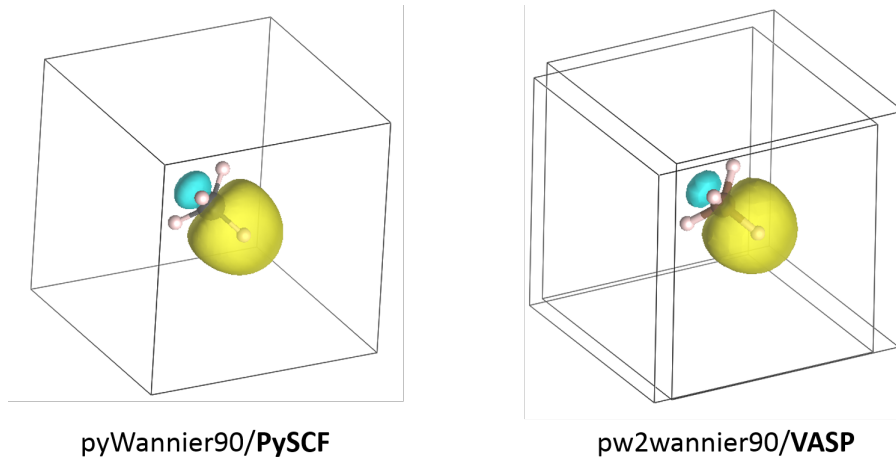


Figure 1. One of four sp³-like wannier functions calculated by pyWannier90/PySCF and pw2wannier90/VASP. The other three MLWFs orientate in different directions but look identical to each other.

3.2 H_2 in a box

A H_2 molecule in a 3.0 \AA cubic box is computed at the PBE/cc-pCVDZ level of theory. The Brillouin zone is sampled with a k-point mesh of $2 \times 2 \times 2$. Two Bloch states from DFT calculation are used to compute one MLWF. Bloch states are used to compute the projection matrix.

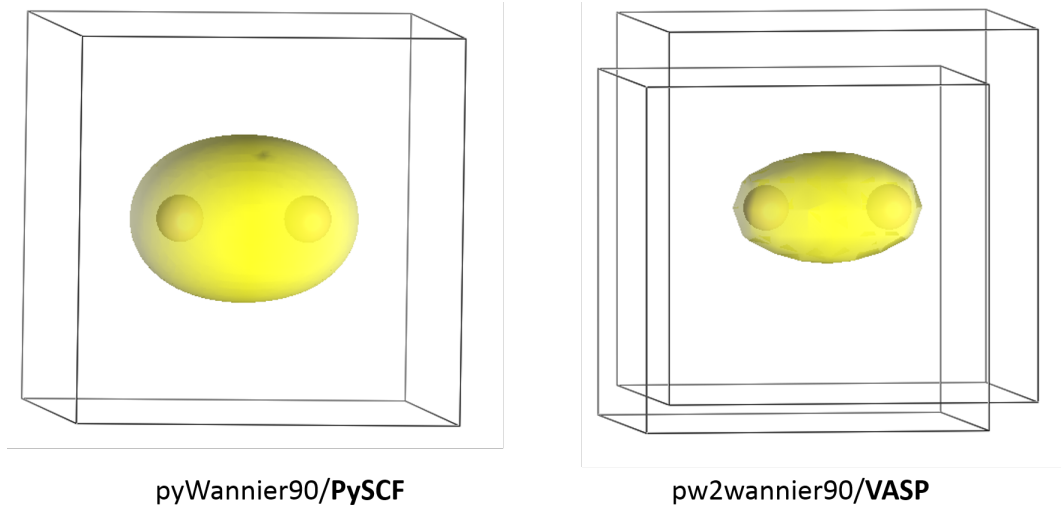


Figure 2. A σ -like wannier function calculated by pyWannier90/PySCF and pw2wannier90/VASP.

3.3 Polyynes

In this example, polyynes in a $2.65 \times 4.00 \times 4.00 \text{ \AA}$ rectangular box is computed at the PBE/6-31G level of theory. The Brillouin zone is sampled with a k-point mesh of $5 \times 1 \times 1$. Two highest occupied Bloch states from DFT calculation are used to compute two MLWFs. Bloch states are used as initial guess.

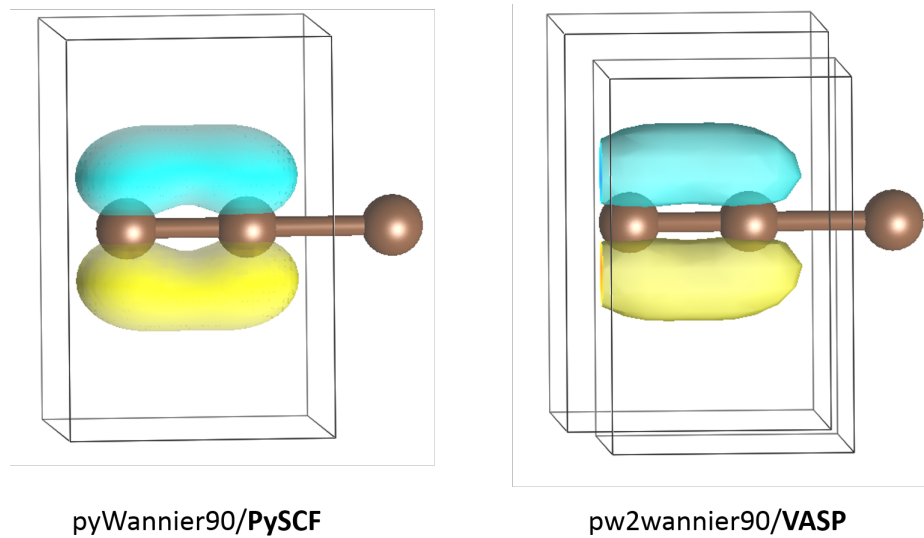


Figure 3. A π -like wannier function calculated by pyWannier90/PySCF and pw2wannier90/VASP. The other MLWF orientates in a different direction but look identical.

4 Appendix

4.1 The overlap matrix between atomic Bloch functions at a certain k-point

$$S_{\mu\nu}^{BB}(\mathbf{k}) = \langle \phi_{\mu}^{\mathbf{k}}(\mathbf{r}) | \phi_{\nu}^{\mathbf{k}}(\mathbf{r}) \rangle \quad (4.1)$$

$$= \int_0^{\infty} \phi_{\mu}^{\mathbf{k}\dagger}(\mathbf{r}) \cdot \phi_{\nu}^{\mathbf{k}}(\mathbf{r}) \cdot d\mathbf{r} \quad (4.2)$$

$$= \frac{1}{\sqrt{M}} \frac{1}{\sqrt{M}} \sum_{\mathbf{R}_1} \sum_{\mathbf{R}_2} e^{-i \cdot \mathbf{k} \cdot \mathbf{R}_1} \cdot e^{i \cdot \mathbf{k} \cdot \mathbf{R}_2} \int_0^{\infty} \chi_{\mu}^{\dagger}(\mathbf{r} - \mathbf{R}_1) \cdot \chi_{\nu}(\mathbf{r} - \mathbf{R}_2) \cdot d\mathbf{r} \quad (4.3)$$

noted that: $\chi_{\mu}(\mathbf{r} - \mathbf{R}) \equiv \chi_{\mu}^{\mathbf{R}}(\mathbf{r})$ and with $\mathbf{r} - \mathbf{R}_1 = \mathbf{r}'$ and $\mathbf{R}_2 - \mathbf{R}_1 = \mathbf{R}$, equation (4.3) becomes:

$$= \frac{1}{M} \sum_{\mathbf{R}_1} \sum_{\mathbf{R}} e^{-i \cdot \mathbf{k} \cdot \mathbf{R}_1} \cdot e^{i \cdot \mathbf{k} \cdot (\mathbf{R}_1 + \mathbf{R})} \int_0^{\infty} \chi_{\mu}^{\dagger}(\mathbf{r}') \cdot \chi_{\nu}(\mathbf{r}' - \mathbf{R}) \cdot d\mathbf{r}' \quad (4.4)$$

$$= \frac{1}{M} \sum_{\mathbf{R}_1} e^{-i \cdot (\mathbf{k} - \mathbf{k}) \cdot \mathbf{R}_1} \sum_{\mathbf{R}} e^{i \cdot \mathbf{k} \cdot \mathbf{R}} \int_0^{\infty} \chi_{\mu}^{\dagger}(\mathbf{r}') \cdot \chi_{\nu}(\mathbf{r}' - \mathbf{R}) \cdot d\mathbf{r}' \quad (4.5)$$

$$= \sum_{\mathbf{R}} e^{i \cdot \mathbf{k} \cdot \mathbf{R}} \int_0^{\infty} \chi_{\mu}^{\dagger}(\mathbf{r}') \cdot \chi_{\nu}(\mathbf{r}' - \mathbf{R}) \cdot d\mathbf{r}' \quad (4.6)$$

$$= \sum_{\mathbf{R}} e^{i \cdot \mathbf{k} \cdot \mathbf{R}} \langle \chi_{\mu} | \chi_{\nu}^{\mathbf{R}} \rangle \quad (4.7)$$

Hence:

$$S_{\mu\nu}^{BB}(\mathbf{k}) = \sum_{\mathbf{R}} e^{i \cdot \mathbf{k} \cdot \mathbf{R}} \langle \chi_{\mu} | \chi_{\nu}^{\mathbf{R}} \rangle \quad (4.8)$$

4.2 The overlap matrix between atomic Bloch functions at two different k-points

$$S_{\mu\nu}^{BB}(\mathbf{k}_1, \mathbf{k}_2) = \langle \phi_{\mu}^{\mathbf{k}_1}(\mathbf{r}) | \phi_{\nu}^{\mathbf{k}_2}(\mathbf{r}) \rangle \quad (4.9)$$

$$= \int_0^{\infty} \phi_{\mu}^{\mathbf{k}_1\dagger}(\mathbf{r}) \cdot \phi_{\nu}^{\mathbf{k}_2}(\mathbf{r}) \cdot d\mathbf{r} \quad (4.10)$$

$$= \frac{1}{\sqrt{M}} \frac{1}{\sqrt{M}} \sum_{\mathbf{R}_1} \sum_{\mathbf{R}_2} e^{-i \cdot \mathbf{k}_1 \cdot \mathbf{R}_1} \cdot e^{i \cdot \mathbf{k}_2 \cdot \mathbf{R}_2} \int_0^{\infty} \chi_{\mu}^{\dagger}(\mathbf{r} - \mathbf{R}_1) \cdot \chi_{\nu}(\mathbf{r} - \mathbf{R}_2) \cdot d\mathbf{r} \quad (4.11)$$

Similarly, with $\mathbf{r} - \mathbf{R}_1 = \mathbf{r}'$ and $\mathbf{R}_2 - \mathbf{R}_1 = \mathbf{L}$, equation (4.11) becomes:

$$S_{\mu\nu}^{BB}(\mathbf{k}_1, \mathbf{k}_2) = \frac{1}{M} \sum_{\mathbf{R}_1} \sum_{\mathbf{R}} e^{-i \cdot \mathbf{k}_1 \cdot \mathbf{R}_1} \cdot e^{i \cdot \mathbf{k}_2 \cdot (\mathbf{R}_1 + \mathbf{R})} \int_0^{\infty} \chi_{\mu}^{\dagger}(\mathbf{r}') \cdot \chi_{\nu}(\mathbf{r}' - \mathbf{R}) \cdot d\mathbf{r}' \quad (4.12)$$

$$= \frac{1}{M} \sum_{\mathbf{R}_1} e^{-i \cdot (\mathbf{k}_1 - \mathbf{k}_2) \cdot \mathbf{R}_1} \sum_{\mathbf{R}} e^{i \cdot \mathbf{k}_2 \cdot \mathbf{R}} \int_0^{\infty} \chi_{\mu}^{\dagger}(\mathbf{r}') \cdot \chi_{\nu}(\mathbf{r}' - \mathbf{R}) \cdot d\mathbf{r}' \quad (4.13)$$

$$= \frac{1}{M} \sum_{\mathbf{R}_1} e^{-i \cdot (\mathbf{k}_1 - \mathbf{k}_2) \cdot \mathbf{R}_1} \cdot S_{\mu\nu}^{BB}(\mathbf{k}_2) \quad (4.14)$$

Hence:

$$S_{\mu\nu}^{BB}(\mathbf{k}_1, \mathbf{k}_2) = \frac{1}{M} \sum_{\mathbf{R}_1} e^{-i \cdot (\mathbf{k}_1 - \mathbf{k}_2) \cdot \mathbf{R}_1} \cdot S_{\mu\nu}^{BB}(\mathbf{k}_2) \quad (4.15)$$

4.3 The overlap matrix between Bloch functions and Gaussian functions at the reference cell

$$S_{\mu\nu}^{BG}(\mathbf{k}) = \langle \chi_{\mu}^{\mathbf{R}_0}(\mathbf{r}) | \phi_{\nu}^{\mathbf{k}}(\mathbf{r}) \rangle \quad (4.16)$$

$$= \int_0^{\infty} \chi_{\mu}^{\mathbf{R}_0}(\mathbf{r}) \cdot \phi_{\nu}^{\mathbf{k}}(\mathbf{r}) \cdot d\mathbf{r} \quad (4.17)$$

$$= \frac{1}{\sqrt{M}} \sum_{\mathbf{R}} e^{i \cdot \mathbf{k} \cdot \mathbf{R}} \int_0^{\infty} \chi_{\mu}^{\dagger}(\mathbf{r}) \cdot \chi_{\nu}(\mathbf{r} - \mathbf{R}) d\mathbf{r} \quad (4.18)$$

$$= \frac{1}{\sqrt{M}} \sum_{\mathbf{R}} e^{i \cdot \mathbf{k} \cdot \mathbf{R}} \langle \chi_{\mu} | \chi_{\nu}^{\mathbf{R}} \rangle \quad (4.19)$$

Hence:

$$S_{\mu\nu}^{BG}(\mathbf{k}) = \frac{1}{\sqrt{M}} S_{\mu\nu}^{BB}(\mathbf{k}) \quad (4.20)$$

5 Acknowledgement

The author would like to thank Dr. Qiming Sun (Caltech) for his insightful discussion about PySCF package. This work cannot be done without his kind support.

References

- [1] Nicola Marzari et al. “Maximally localized Wannier functions: Theory and applications”. In: *Rev. Mod. Phys.* 84 (4 Oct. 2012), pp. 1419–1475. DOI: 10.1103/RevModPhys.84.1419. URL: <https://link.aps.org/doi/10.1103/RevModPhys.84.1419>.
- [2] Nicola Marzari and David Vanderbilt. “Maximally localized generalized Wannier functions for composite energy bands”. In: *Phys. Rev. B* 56 (20 Nov. 1997), pp. 12847–12865. DOI: 10.1103/PhysRevB.56.12847. URL: <https://link.aps.org/doi/10.1103/PhysRevB.56.12847>.
- [3] Arash A. Mostofi et al. “An updated version of wannier90: A tool for obtaining maximally-localised Wannier functions”. In: *Computer Physics Communications* 185.8 (2014), pp. 2309–2310. ISSN: 0010-4655. DOI: <https://doi.org/10.1016/j.cpc.2014.05.003>. URL: <http://www.sciencedirect.com/science/article/pii/S001046551400157X>.
- [4] Ivo Souza, Nicola Marzari, and David Vanderbilt. “Maximally localized Wannier functions for entangled energy bands”. In: *Phys. Rev. B* 65 (3 Dec. 2001), p. 035109. DOI: 10.1103/PhysRevB.65.035109. URL: <https://link.aps.org/doi/10.1103/PhysRevB.65.035109>.