

wannier90: A tool for obtaining maximally-localised Wannier functions [☆]

Arash A. Mostofi ^{a,*}, Jonathan R. Yates ^{b,c}, Young-Su Lee ^a, Ivo Souza ^{b,c}, David Vanderbilt ^d,
Nicola Marzari ^a

^a Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

^b Materials Science Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

^c Department of Physics, University of California, Berkeley, CA 94720, USA

^d Department of Physics and Astronomy, Rutgers University, Piscataway, NJ 08854-8019, USA

Received 5 August 2007; accepted 29 November 2007

Available online 23 December 2007

Abstract

We present `wannier90`, a program for calculating maximally-localised Wannier functions (MLWF) from a set of Bloch energy bands that may or may not be attached to or mixed with other bands. The formalism works by minimising the total spread of the MLWF in real space. This is done in the space of unitary matrices that describe rotations of the Bloch bands at each k -point. As a result, `wannier90` is independent of the basis set used in the underlying calculation to obtain the Bloch states. Therefore, it may be interfaced straightforwardly to any electronic structure code. The locality of MLWF can be exploited to compute band-structure, density of states and Fermi surfaces at modest computational cost. Furthermore, `wannier90` is able to output MLWF for visualisation and other post-processing purposes. Wannier functions are already used in a wide variety of applications. These include analysis of chemical bonding in real space; calculation of dielectric properties via the modern theory of polarisation; and as an accurate and minimal basis set in the construction of model Hamiltonians for large-scale systems, in linear-scaling quantum Monte Carlo calculations, and for efficient computation of material properties, such as the anomalous Hall coefficient. `wannier90` is freely available under the GNU General Public License from <http://www.wannier.org/>.

Program summary

Program title: `wannier90`

Catalogue identifier: AEAk_v1_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEAk_v1_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: Standard CPC licence, <http://cpc.cs.qub.ac.uk/licence/licence.html>

No. of lines in distributed program, including test data, etc.: 556 495

No. of bytes in distributed program, including test data, etc.: 5 709 419

Distribution format: `tar.gz`

Programming language: Fortran 90, perl

Computer: any architecture with a Fortran 90 compiler

Operating system: Linux, Windows, Solaris, AIX, Tru64 Unix, OSX

RAM: 10 MB

Word size: 32 or 64

Classification: 7.3

External routines:

- BLAS (<http://www.netlib.org/blas>).

[☆] This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Corresponding author.

E-mail address: a.mostofi@imperial.ac.uk (A.A. Mostofi).

- LAPACK (<http://www.netlib.org/lapack>).

Both available under open-source licenses.

Nature of problem: Obtaining maximally-localised Wannier functions from a set of Bloch energy bands that may or may not be entangled.

Solution method: In the case of entangled bands, the optimally-connected subspace of interest is determined by minimising a functional which measures the subspace dispersion across the Brillouin zone. The maximally-localised Wannier functions within this subspace are obtained by subsequent minimisation of a functional that represents the total spread of the Wannier functions in real space. For the case of isolated energy bands only the second step of the procedure is required.

Unusual features: Simple and user-friendly input system. Wannier functions and interpolated band structure output in a variety of file formats for visualisation.

Running time: Test cases take 1 minute.

References:

- [1] N. Marzari, D. Vanderbilt, Maximally localized generalized Wannier functions for composite energy bands, *Phys. Rev. B* 56 (1997) 12847.
 [2] I. Souza, N. Marzari, D. Vanderbilt, Maximally localized Wannier functions for entangled energy bands, *Phys. Rev. B* 65 (2001) 035109.

© 2007 Elsevier B.V. All rights reserved.

PACS: 71.15.-m; 75.15.Ap; 73.22.-f; 31.10.+z

Keywords: Electronic structure; Density-functional theory; Wannier function

1. Introduction

Within the independent-particle approximation, the electronic ground state of a periodic system may be solved in terms of a set of extended Bloch states $\psi_{n\mathbf{k}}(\mathbf{r})$. These states are characterised by the good quantum numbers n and \mathbf{k} , which refer to the band index and crystal momentum, respectively. Although this choice is widely used in electronic structure calculations, alternative representations are available. For example, the Wannier representation constitutes a description in terms of localised functions labeled by \mathbf{R} , the lattice vector of the cell in which the function is localised, and a band-like index n .

Wannier functions give a real-space picture of the electronic structure of a system. They provide insight into the nature of the chemical bonding and can be a powerful tool in the study of dielectric properties via the modern theory of polarisation.

The phase indeterminacy $e^{i\phi_n(\mathbf{k})}$ of an isolated Bloch state $\psi_{n\mathbf{k}}(\mathbf{r})$ at each wave-vector \mathbf{k} results in the Wannier functions being non-unique. For a group of isolated bands, such as the valence states of an insulator, this indeterminacy is more general as they may undergo arbitrary unitary transformations $U_{mn}^{(\mathbf{k})}$ amongst themselves at each \mathbf{k} . Two of the authors (N.M. and D.V.) developed a procedure that iteratively refines these degrees of freedom such that they lead to Wannier functions that are well-defined and localised (in the sense that they minimise the second moment around their centres) [1]. We refer to this procedure as the Marzari–Vanderbilt (MV) scheme and the resulting Wannier functions as maximally-localised Wannier functions (MLWF).

The MV method was designed to be applicable to isolated sub-groups of bands, i.e., a group of bands that, though they may have degeneracies with each other at certain points, are separated from all other bands by finite gaps throughout the Brillouin zone. An important example is the set of valence bands of an insulator. In many applications, however, the bands of interest are not isolated and one is interested, for instance, in the partially filled bands of a metal close to the Fermi level. In this case the bands of interest lie within a limited energy range but are attached to, or cross with, other bands that extend further out in energy. Such bands are referred to as *entangled* bands.

The complication associated with treating entangled bands is that the states of interest hybridise with unwanted bands. As a result, the number of bands at each \mathbf{k} -point in the relevant energy range may be greater than the number of Wannier functions N that are required. Before the MV localisation procedure may be applied, a prescription for extracting N bands at each \mathbf{k} -point from the entangled manifold of states is required. Three of the authors developed just such a *disentanglement* procedure, allowing MLWF to be determined from a set of entangled bands [2]. This Souza–Marzari–Vanderbilt (SMV) strategy breaks down the procedure into two steps: first, the correct N -dimensional subspace of bands at every \mathbf{k} is selected; and second, N MLWF are localised from this subspace, exactly in the same way as for an isolated set of bands.

wannier90 is a tool for obtaining MLWF from a set of (possibly entangled) energy bands using the methods of MV and SMV. The principal ingredient that is required from an electronic structure calculation is the overlap matrix between the periodic part $|u_{n\mathbf{k}}\rangle$ of Bloch states at neighbouring \mathbf{k} -points (described in more detail in Section 2). This matrix is small and independent of the basis used in the underlying calculation to obtain the Bloch states. As a result wannier90 may be interfaced to any electronic structure code. At the time of writing, wannier90 is able to:

1. Disentangle an optimally-connected N -dimensional subspace of bands at each \mathbf{k} -point and obtain the unitary transformations that generate MLWF from a given set of bands;
2. Output MLWF in a number of formats suitable for visualisation;

3. Generate interpolated band structures, densities of states and Fermi surfaces and output them in formats suitable for visualisation;
4. Output matrix elements of the Hamiltonian operator in the MLWF basis.

It is worth recording here a brief historical timeline of `wannier90`:

- **1997**: Two of the authors (N.M. and D.V.) develop and implement localisation algorithms in Fortran77; interface to the all-bands ensemble-DFT CASTEP code [3];
- **1998**: Algorithms implemented in the CPMD code [4] by Silvestrelli [5];
- **2001**: Three of the authors (I.S., N.M. and D.V.) develop the SMV disentanglement extension; interface to a full-potential-linearised-augmented-plane-wave (FLAPW) code [6] and to any generic electronic-structure code quickly followed;
- **2006**: Two of the authors (A.A.M. and J.R.Y.) completely rewrite the code in Fortran90, employing modern programming techniques and adding new functionality.

One significant benefit of the 2006 rewrite has been that in most cases `wannier90` is faster than the original Fortran77 version by an order of magnitude, and in some by more than two. Another is that now the source code is easy to follow, which makes interfacing it to electronic structure codes more straightforward. Indeed, the first such example is already in place within the QUANTUM-ESPRESSO package [7].

The formalism has seen many and diverse applications: linear-scaling quantum Monte Carlo [8], photonic crystals [9,10], metal-insulator interfaces [11], as an efficient interpolator for the anomalous Hall effect [12] and electron-phonon couplings [13], and a powerful tool for the study of large-scale systems [14,15], to cite only a few. In addition, MLWF are playing an increasing role in bridging density-functional approaches and strongly-correlated ones, to derive model Hamiltonians or as a starting point for LDA+U or LDA+DMFT [16–18]. They are also closely related to muffin-tin orbitals [19,20].

The remainder of this paper is organised as follows. In Section 2 we review briefly the theoretical background to the problem of obtaining MLWF and a recent theoretical development in using them to interpolate band structures on arbitrarily fine k-point meshes. In Section 3 we give an overview of some of the more technical aspects of the `wannier90` code. In Section 4 we describe the overall structure of the code. We demonstrate how to install and run `wannier90` in Sections 5 and 6, respectively. Finally, in Section 7 we provide a number of example calculations and applications of the code.

2. Theoretical background

For an isolated set of N Bloch bands $\psi_{n\mathbf{k}}(\mathbf{r})$, a set of N Wannier functions $w_{n\mathbf{R}}(\mathbf{r}) = w_n(\mathbf{r} - \mathbf{R})$, $n \in [1, N]$, labelled by Bravais lattice vectors \mathbf{R} , may be constructed as

$$|w_{n\mathbf{R}}\rangle = \frac{V}{(2\pi)^3} \int_{\text{BZ}} \left[\sum_{m=1}^N U_{mn}^{(\mathbf{k})} |\psi_{m\mathbf{k}}\rangle \right] e^{-i\mathbf{k}\cdot\mathbf{R}} d\mathbf{k}, \quad (1)$$

where $\mathbf{U}^{(\mathbf{k})}$ is a unitary matrix that mixes the bands at wave-vector \mathbf{k} , and the Brillouin zone (BZ) integral may also be interpreted as a unitary transformation. Different choices for $\mathbf{U}^{(\mathbf{k})}$ give rise to different Wannier functions, with different spatial spreads, that always describe the same manifold. The MV strategy consists of choosing the $\mathbf{U}^{(\mathbf{k})}$ that minimise the sum of the quadratic spreads of the Wannier functions about their centres:

$$\Omega = \sum_n \langle (\mathbf{r} - \bar{\mathbf{r}}_n)^2 \rangle_n = \sum_n \langle r^2 - 2\mathbf{r} \cdot \bar{\mathbf{r}}_n + |\bar{\mathbf{r}}_n|^2 \rangle_n = \sum_n [\langle r^2 \rangle_n - |\bar{\mathbf{r}}_n|^2], \quad (2)$$

where $\langle \hat{\mathcal{O}} \rangle_n \equiv \bar{\mathcal{O}}_n \equiv \langle w_{n\mathbf{0}} | \hat{\mathcal{O}} | w_{n\mathbf{0}} \rangle$.

It is worth noting that the spread functional may be decomposed into two contributions

$$\Omega = \Omega_I + \tilde{\Omega}, \quad (3)$$

where

$$\Omega_I = \sum_n \left[\langle w_{n\mathbf{0}} | r^2 | w_{n\mathbf{0}} \rangle - \sum_{m\mathbf{R}} |\langle w_{m\mathbf{R}} | \mathbf{r} | w_{n\mathbf{0}} \rangle|^2 \right] \quad (4)$$

and

$$\tilde{\Omega} = \sum_n \sum_{m\mathbf{R} \neq \mathbf{0}} |\langle w_{m\mathbf{R}} | \mathbf{r} | w_{n\mathbf{0}} \rangle|^2. \quad (5)$$

It can be shown that each of these quantities is non-negative and that Ω_I is *gauge invariant*, i.e., insensitive to changes in the matrices $U_{mn}^{(\mathbf{k})}$ [1]. Therefore, the minimisation of the spread functional for an isolated set of bands just corresponds to minimising $\tilde{\Omega}$.

2.1. Reciprocal-space formulation

As shown by Blount [21], matrix elements of the position operator between Wannier functions may be expressed in reciprocal-space as

$$\langle w_{n\mathbf{R}} | \mathbf{r} | w_{m\mathbf{0}} \rangle = i \frac{V}{(2\pi)^3} \int e^{i\mathbf{k}\cdot\mathbf{R}} \langle u_{n\mathbf{k}} | \nabla_{\mathbf{k}} | u_{m\mathbf{k}} \rangle d\mathbf{k} \quad (6)$$

and

$$\langle w_{n\mathbf{R}} | r^2 | w_{m\mathbf{0}} \rangle = - \frac{V}{(2\pi)^3} \int e^{i\mathbf{k}\cdot\mathbf{R}} \langle u_{n\mathbf{k}} | \nabla_{\mathbf{k}}^2 | u_{m\mathbf{k}} \rangle d\mathbf{k}, \quad (7)$$

where, as usual, the periodic part of the Bloch function is defined as $u_{n\mathbf{k}}(\mathbf{r}) = e^{-i\mathbf{k}\cdot\mathbf{r}} \psi_{n\mathbf{k}}(\mathbf{r})$. These expressions enable us to express the spread functional in terms of matrix elements of $\nabla_{\mathbf{k}}$ and $\nabla_{\mathbf{k}}^2$.

It is assumed that the Brillouin zone is discretised on a uniform Monkhorst–Pack mesh [22] on which the Bloch orbitals are determined. Thus, the gradient and Laplacian operators may be approximated by finite-difference formulas on the \mathbf{k} -space mesh. If $f(\mathbf{k})$ is a smooth function of \mathbf{k} , then

$$\nabla_{\mathbf{k}} f(\mathbf{k}) = \sum_{\mathbf{b}} \omega_b \mathbf{b} [f(\mathbf{k} + \mathbf{b}) - f(\mathbf{k})] \quad (8)$$

and

$$\langle f(\mathbf{k}) | \nabla_{\mathbf{k}}^2 | f(\mathbf{k}) \rangle = |\nabla_{\mathbf{k}} f(\mathbf{k})|^2 = \sum_{\mathbf{b}} \omega_b [f(\mathbf{k} + \mathbf{b}) - f(\mathbf{k})]^2, \quad (9)$$

where $\{\mathbf{b}\}$ are vectors connecting mesh-point \mathbf{k} to its nearest neighbours and ω_b is a weight factor associated with each shell of neighbours $b = |\mathbf{b}|$. The choice of \mathbf{b} -vectors and weights ω_b is discussed in further detail in Section 3.2.

It is worth noting that a finite \mathbf{k} -point mesh implies an approximation to both the self-consistent ground state that is obtained using that mesh *and* to the accuracy of the finite-difference representation of the operators $\nabla_{\mathbf{k}}$ and $\nabla_{\mathbf{k}}^2$. In principle, for a given mesh of \mathbf{k} -points, the latter may be improved by using higher-order finite-difference expressions.

Having discretised Eqs. (6) and (7) in reciprocal space, the only information required for computing the spread functional is the overlap matrix

$$M_{mn}^{(\mathbf{k},\mathbf{b})} = \langle u_{m\mathbf{k}} | u_{n,\mathbf{k}+\mathbf{b}} \rangle. \quad (10)$$

After some algebra, the two components of the spread functional may be expressed as

$$\Omega_I = \frac{1}{N_{\text{kp}}} \sum_{\mathbf{k},\mathbf{b}} \omega_b \sum_{m=1}^N \left[1 - \sum_{n=1}^N |M_{mn}^{(\mathbf{k},\mathbf{b})}|^2 \right] \quad (11)$$

and

$$\tilde{\Omega} = \frac{1}{N_{\text{kp}}} \sum_{\mathbf{k},\mathbf{b}} \omega_b \left[\sum_{n=1}^N (-\text{Im} \ln M_{nn}^{(\mathbf{k},\mathbf{b})} - \mathbf{b} \cdot \bar{\mathbf{r}}_n)^2 + \sum_{m \neq n}^N |M_{mn}^{(\mathbf{k},\mathbf{b})}|^2 \right], \quad (12)$$

where N_{kp} is the number of \mathbf{k} -points in the Monkhorst–Pack grid and $\bar{\mathbf{r}}_n$ is the centre of the n th Wannier function, given by

$$\bar{\mathbf{r}}_n = - \frac{1}{N_{\text{kp}}} \sum_{\mathbf{k},\mathbf{b}} \omega_b \mathbf{b} \text{Im} \ln M_{nn}^{(\mathbf{k},\mathbf{b})}. \quad (13)$$

Using these expressions, the gradient of the spread functional with respect to infinitesimal unitary rotations of the $\psi_{n\mathbf{k}}(\mathbf{r})$ may be calculated as a function of $M_{mn}^{(\mathbf{k},\mathbf{b})}$. It is then straightforward to evolve $U_{mn}^{(\mathbf{k})}$ (and consequently $M_{mn}^{(\mathbf{k},\mathbf{b})}$) towards the solution of maximum localisation using a steepest-descent or conjugate-gradient minimisation algorithm.

It is worth noting that $\tilde{\Omega}$ may be further decomposed into band-diagonal and band-off-diagonal parts

$$\tilde{\Omega} = \Omega_D + \Omega_{OD}, \quad (14)$$

where

$$\Omega_D = \sum_n \sum_{\mathbf{R} \neq \mathbf{0}} |\langle w_{n\mathbf{R}} | \mathbf{r} | w_{n\mathbf{0}} \rangle|^2 = \frac{1}{N_{\text{kp}}} \sum_{\mathbf{k}, \mathbf{b}} \omega_b \sum_{n=1}^N (-\text{Im} \ln M_{nn}^{(\mathbf{k}, \mathbf{b})} - \mathbf{b} \cdot \bar{\mathbf{r}}_n)^2, \quad (15)$$

$$\Omega_{OD} = \sum_{m \neq n} \sum_{\mathbf{R}} |\langle w_{m\mathbf{R}} | \mathbf{r} | w_{n\mathbf{0}} \rangle|^2 = \frac{1}{N_{\text{kp}}} \sum_{\mathbf{k}, \mathbf{b}} \omega_b \sum_{n=1}^N \sum_{m \neq n}^N |M_{mn}^{(\mathbf{k}, \mathbf{b})}|^2. \quad (16)$$

2.2. The case of entangled bands

The above description is sufficient to obtain MLWF for an isolated set of bands, such as the valence states in an insulator. In order to obtain MLWF for entangled energy bands, e.g., for metallic systems or the conduction bands of an insulator, we use the “disentanglement” procedure introduced by SMV [2].

The strategy proceeds as follows. An energy window (the “outer window”) is defined by the user such that at each \mathbf{k} -point there are $N_{\text{win}}^{(\mathbf{k})} \geq N$ states within the window. An orthonormal set of N Bloch states is obtained at each \mathbf{k} -point, defining an N -dimensional subspace $\mathcal{S}(\mathbf{k})$, by performing a unitary transformation amongst the $N_{\text{win}}^{(\mathbf{k})}$ states that fall within the energy window:

$$|u_{n\mathbf{k}}^{\text{opt}}\rangle = \sum_{m \in N_{\text{win}}^{(\mathbf{k})}} U_{mn}^{\text{dis}(\mathbf{k})} |u_{m\mathbf{k}}\rangle, \quad (17)$$

where $\mathbf{U}^{\text{dis}(\mathbf{k})}$ is a rectangular $N_{\text{win}}^{(\mathbf{k})} \times N$ matrix.¹

Recall that Ω_I is invariant under gauge transformations within a given subspace. Thus Ω_I may be considered as a functional of $\mathcal{S}(\mathbf{k})$. The subspace selection proceeds by minimising Ω_I with respect to the matrices $\mathbf{U}^{\text{dis}(\mathbf{k})}$ [2]. For a physical interpretation of this procedure, Eq. (11) for Ω_I may be rewritten as

$$\Omega_I = \frac{1}{N_{\text{kp}}} \sum_{\mathbf{k}, \mathbf{b}} \omega_b \text{tr} [\hat{P}_{\mathbf{k}} \hat{Q}_{\mathbf{k}+\mathbf{b}}], \quad (18)$$

where $\hat{P}_{\mathbf{k}} = \sum_{n=1}^N |u_{n\mathbf{k}}\rangle \langle u_{n\mathbf{k}}|$ is the projector onto $\mathcal{S}(\mathbf{k})$, and $\hat{Q}_{\mathbf{k}} = \mathbf{1} - \hat{P}_{\mathbf{k}}$. Now it can be seen that Ω_I is a measure of the degree of mismatch between the subspaces $\mathcal{S}(\mathbf{k})$ and $\mathcal{S}(\mathbf{k} + \mathbf{b})$. Minimising Ω_I corresponds to choosing self-consistently at every \mathbf{k} the subspace that has maximum overlap as \mathbf{k} is varied.

Once the projected N -dimensional subspace at each \mathbf{k} -point has been found, the MV localisation procedure described above may be used to minimise $\tilde{\Omega}$ within that subspace and hence obtain MLWF. As an alternative to this two step minimisation, a procedure to minimise the total spread function Ω has been proposed [23] and was found to give very similar MLWF to the present scheme.

It may be the case that the energy bands of the projected subspace do not correspond to any of the original energy bands due to mixing between states. In order to preserve exactly the properties of a system within a given energy range (e.g., around the Fermi level) a second *inner* energy window is introduced. States lying within this inner, or frozen, energy window are included unchanged in the projected subspace. The reader is referred to Ref. [2] for further details of the algorithm.

It is worth noting that the MLWF themselves are never explicitly constructed unless required for visualisation or other post-processing purposes. Minimisation of the spread functional results in finding the converged $\mathbf{U}^{\text{dis}(\mathbf{k})}$ (if disentanglement was used) and $\mathbf{U}^{(\mathbf{k})}$. Along with $\mathbf{M}^{(\mathbf{k}, \mathbf{b})}$, this is sufficient to define the centres and spreads of the MLWF. If the periodic parts of the Bloch wavefunctions are available, then the MLWF may be calculated on a real-space grid. For systems with time-reversal symmetry, we always find the MLWF corresponding to the minimum spread to be real, apart from a global complex phase factor, in empirical agreement with a recent claim of a mathematical proof [24].

2.3. Wannier interpolation

Having found $\mathbf{U}^{\text{dis}(\mathbf{k})}$ and $\mathbf{U}^{(\mathbf{k})}$ for the target system, it is straightforward to express the Hamiltonian in the basis of MLWF. We first obtain the Hamiltonian in the basis of rotated Bloch states

$$H^{(W)}(\mathbf{k}) = (\mathbf{U}^{(\mathbf{k})})^\dagger (\mathbf{U}^{\text{dis}(\mathbf{k})})^\dagger H(\mathbf{k}) \mathbf{U}^{\text{dis}(\mathbf{k})} \mathbf{U}^{(\mathbf{k})}, \quad (19)$$

where $H_{nm}(\mathbf{k}) = \varepsilon_{n\mathbf{k}} \delta_{nm}$. Next we find its Fourier sum

$$H_{nm}(\mathbf{R}) = \frac{1}{N_0} \sum_{\mathbf{k}} e^{-i\mathbf{k} \cdot \mathbf{R}} H_{nm}^{(W)}(\mathbf{k}). \quad (20)$$

¹ As $\mathbf{U}^{\text{dis}(\mathbf{k})}$ is rectangular, this is a unitary operation in the sense that $(\mathbf{U}^{\text{dis}(\mathbf{k})})^\dagger \mathbf{U}^{\text{dis}(\mathbf{k})} = \mathbb{I}$.

This operation is done once and for all for each of the N_0 lattice vectors \mathbf{R} lying in a supercell conjugate to the \mathbf{k} -mesh (in practice we choose a Wigner–Seitz supercell centred on the origin [25]). $H_{nm}(\mathbf{R})$ can be recognised as the matrix elements of the Hamiltonian between MLWF. Due to the strong localisation of the MLWF, the matrix elements $H_{nm}(\mathbf{R})$ decay rapidly with R . In the spirit of a Slater–Koster interpolation scheme [26] this allows us to compute the Hamiltonian on a much finer mesh of \mathbf{k} -points in the original Bloch space. We can perform the inverse Fourier transform

$$H_{nm}(\mathbf{k}') = \sum_{\mathbf{R}} e^{i\mathbf{k}' \cdot \mathbf{R}} H_{nm}(\mathbf{R}), \quad (21)$$

to yield the interpolation of Eq. (19) onto an arbitrary \mathbf{k} -point \mathbf{k}' . The final step is to diagonalise $H_{nm}(\mathbf{k}')$ to obtain the interpolated band energies.

By construction, the interpolated band energies coincide with the true bandstructure at the original \mathbf{k} -points of the Monkhorst–Pack mesh (if the SMV scheme is used, this is only guaranteed to occur within the inner energy window). At intermediate \mathbf{k} -points, the accuracy of the interpolation is dependent on the density of the original mesh [14,25].

Within `wannier90` this interpolation procedure can be used to obtain plots of band structure, density of states and the Fermi surface at modest computational cost. The Wannier interpolation formalism is rather general and can also be used to interpolate arbitrary periodic operators [25].

3. Some technical aspects

3.1. Initial guess for iterative minimisation

The iterative minimisation of Ω_{Γ} begins with an initial guess for the subspaces $\mathcal{S}(\mathbf{k})$. One possible choice is to start from the initial, random phases of the Bloch states provided by the *ab initio* code. Alternatively, we may define a set of N trial functions $g_n(\mathbf{r})$, $n \in [1, N]$, as an initial approximation to the N MLWF. These are projected onto the cell-periodic parts of the $N_{\text{win}}^{(\mathbf{k})}$ Bloch eigenstates inside the energy window:

$$|\phi_{n\mathbf{k}}\rangle = \sum_{m=1}^{N_{\text{win}}^{(\mathbf{k})}} A_{mn}^{(\mathbf{k})} |u_{m\mathbf{k}}\rangle, \quad (22)$$

where $A_{mn}^{(\mathbf{k})} = \langle u_{m\mathbf{k}} | g_n \rangle$ is an $N_{\text{win}}^{(\mathbf{k})} \times N$ matrix. Orthonormalising the resulting N functions $\{|\phi_{n\mathbf{k}}\rangle\}$ via a Löwdin transformation, we find

$$|u_{n\mathbf{k}}^{\text{opt}}\rangle = \sum_{m=1}^N (S^{-1/2})_{mn} |\phi_{m\mathbf{k}}\rangle = \sum_{m=1}^{N_{\text{win}}^{(\mathbf{k})}} (AS^{-1/2})_{mn} |u_{m\mathbf{k}}\rangle, \quad (23)$$

where $S_{mn} \equiv S_{mn}^{(\mathbf{k})} = \langle \phi_{m\mathbf{k}} | \phi_{n\mathbf{k}} \rangle = (A^\dagger A)_{mn}$, and $AS^{-1/2}$ is used as the initial guess for $\mathbf{U}^{\text{dis}(\mathbf{k})}$.

The same trial orbitals $\{g_n\}$ can also be used to initialise the minimisation of $\tilde{\Omega}$. Using a similar projection procedure to the one described above, an initial guess for the $N \times N$ unitary matrices $\mathbf{U}^{(\mathbf{k})}$ is obtained at each \mathbf{k} -point.

We now come to the choice of trial orbitals. As the minimisation scheme is quite robust, one option is to choose a set of spherically-symmetric Gaussian functions whose centres are chosen randomly; `wannier90` supports this option. On the other hand, a user may wish to apply chemical and physical insight in order to select a better starting point. An *ab initio* code that is interfaced to `wannier90` may use any localised functions desired to construct the projections $A_{mn}^{(\mathbf{k})}$. For convenience, we have defined a standard set of projection functions that should suffice for most situations. This is the set of atomic orbitals of the hydrogen atom, which is a convenient choice for two reasons: first, analytical mathematical forms exist in terms of the good quantum numbers n , l and m ; and second, hybrid orbitals (sp, sp², sp³, sp³d, etc.) may be constructed by simple linear combination $|\phi\rangle = \sum_{nlm} C_{nlm} |nlm\rangle$, for some coefficients C_{nlm} . The angular parts of our projection functions are not the canonical spherical harmonics Y_{lm} of the hydrogenic Schrödinger equation, but rather the *real* (in the sense of non-imaginary) states Θ_{lm_r} , $m_r \in [1, 2l + 1]$, obtained by unitary transformation of the Y_{lm} .

Each localised function is associated with a site and an angular momentum state. Optionally, one may define the spatial orientation, the radial form and the diffusivity for each function. `wannier90` is able to project onto functions with s, p, d and f symmetry, plus the hybrids sp, sp², sp³, sp³d, sp³d². The user is referred to the documentation in the `wannier90` distribution for mathematical definitions and details on how to specify projection functions in the input file.

In general, the spread functional Ω has local minima and, occasionally, the minimisation becomes trapped in one. In other words, the final solution may depend on the initial choice for $\mathcal{S}(\mathbf{k})$ and hence $\{g_n(\mathbf{r})\}$. In most cases, we find that these local minima give rise to MLWF that are complex, i.e., they have significant imaginary parts. In some cases, MLWF associated with local minima of Ω are found to be real and the reader is referred to Ref. [27] for more details.

3.2. Choosing the \mathbf{b} vectors

As discussed in Section 2.1, in order to compute the spread functional we require a finite-difference formula for $\nabla_{\mathbf{k}}$. We now describe an automatic procedure to choose, for cells of arbitrary symmetry, the simplest such formula.

It is assumed that the Brillouin zone is discretised on a uniform Monkhorst–Pack mesh [22]. The vectors $\{\mathbf{b}\}$ connect each mesh-point \mathbf{k} to its nearest neighbours. N_{sh} shells of neighbours are included in the finite-difference formula, with M_s vectors in the s th shell. For $\nabla_{\mathbf{k}}$ to be correct to linear order we require that (see Eq. (B1) of Ref. [1])

$$\sum_s \omega_s \sum_i b_{\alpha}^{i,s} b_{\beta}^{i,s} = \delta_{\alpha\beta}, \quad (24)$$

where $\mathbf{b}^{i,s}$, $i \in [1, M_s]$, is the i th vector belonging to the s th shell with associated weight ω_s , and α and β run over the three Cartesian indices. If $f(\mathbf{k})$ is a smooth function of \mathbf{k} , then its gradient may be expressed according to Eq. (8). For the case of a linear function $f(\mathbf{k}) = f_0 + \mathbf{g} \cdot \mathbf{k}$, this finite-difference formula gives the exact result $\nabla_{\mathbf{k},\alpha} f = \sum_{\mathbf{b}} \omega_{\mathbf{b}} b_{\alpha} g_{\beta} b_{\beta} = \delta_{\alpha\beta} g_{\beta} = g_{\alpha}$, where summation convention over repeated Greek indices is assumed.

In order to find the weights ω_s , we notice that Eq. (24) is symmetric in the Cartesian indices, therefore, there are six independent elements that may be expressed through a matrix equation

$$\mathbf{A}\mathbf{w} = \mathbf{q}, \quad (25)$$

where \mathbf{q} is a vector of length six containing the six elements of the lower triangular part of $\delta_{\alpha\beta}$, \mathbf{w} is the vector of weights with length N_{sh} , and \mathbf{A} is a $6 \times N_{\text{sh}}$ matrix given by $A_{j,s} = \sum_i b_{\alpha}^{i,s} b_{\beta}^{i,s}$, where there is a one-to-one correspondence between $j \in [1, 6]$ and the six independent pairings of α and β . \mathbf{A} may be factorised using a singular value decomposition [28],

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \quad (26)$$

which permits inversion of Eq. (25) and solution for the shell weights,

$$\mathbf{w} = \mathbf{V}\mathbf{D}^{-1}\mathbf{U}^T\mathbf{q}. \quad (27)$$

Our automatic procedure for choosing the shells is as follows: we add shells in order of increasing $|\mathbf{b}|$; with each additional shell we obtain the shell weights, and check to see if Eq. (24) is satisfied; if not, then we add a new shell and repeat the procedure. A new shell may be linearly dependent on the existing shells, in which case one or more of the singular values will be close to zero and we reject the new shell. When the Bravais lattice point group is cubic, the first shell of nearest neighbours is sufficient. The maximum number of shells required is six (for triclinic symmetry). For the case of a very elongated unit cell it may be necessary for the routine to search through a large number of shells in order to find the correct combination.

With the above finite-difference formalism, the quadratic spread converges only polynomially with the sampling of the Brillouin zone. This slow convergence is a property of the spread operator, whereas the underlying MLWF converge rapidly with the \mathbf{k} -point density [29]. It may be possible to achieve improved accuracy and \mathbf{k} -point convergence by using higher-order finite-difference formulas for $\nabla_{\mathbf{k}}$, but this has not been explored.

In some cases, the automatic procedure outlined above might not find the shells and weights that give the most spherically symmetric representation of the position operator. When this happens, special care may be required to ensure that the most symmetric choice of shells is made [6], and this can be done explicitly (i.e., by hand) in the input file.

3.3. Specific algorithms for Γ -point sampling

For isolated molecules or extended systems of large unit cell, single Γ -point sampling in the reciprocal space can provide an accurate description of the physical quantities of interest. When the Γ -point is sampled exclusively, the computational cost can be reduced by exploiting the symmetries of the overlap matrices $\mathbf{M}^{(\Gamma, \mathbf{b})}$.

First, at Γ , the \mathbf{b} -vectors are linear combinations of the primitive vectors of the reciprocal lattice, thus imposing the following conditions:

$$\psi_{n, \Gamma+\mathbf{b}}(\mathbf{r}) = \psi_{n\Gamma}(\mathbf{r}), \quad u_{n, \Gamma+\mathbf{b}}(\mathbf{r}) = e^{-i\mathbf{b} \cdot \mathbf{r}} u_{n\Gamma}(\mathbf{r}). \quad (28)$$

Then, it follows that $\mathbf{M}^{(\Gamma, -\mathbf{b})}$ is the Hermitian conjugate of $\mathbf{M}^{(\Gamma, \mathbf{b})}$,

$$\begin{aligned} M_{mn}^{(\Gamma, \mathbf{b})} &= \langle u_{m\Gamma} | u_{n, \Gamma+\mathbf{b}} \rangle = \langle u_{n, \Gamma+\mathbf{b}} | u_{m\Gamma} \rangle^* = \langle u_{n\Gamma} | u_{m, \Gamma-\mathbf{b}} \rangle^* \\ &= (M_{nm}^{(\Gamma, -\mathbf{b})})^*. \end{aligned} \quad (29)$$

Second, at Γ , the Bloch eigenfunctions may be chosen to be real. Then, the overlap matrices become symmetric, $M_{mn}^{(\Gamma, \mathbf{b})} = M_{nm}^{(\Gamma, \mathbf{b})}$, and only the upper or the lower half of $\mathbf{M}^{(\Gamma, \mathbf{b})}$ is independent.

Table 1

Timings (in seconds) for the valence states of crystalline silicon with (i) different densities of Monkhorst–Pack mesh

N_{kp}	T_{SCF}	$T_{M,A}$	T_{W90}	N_{iter}
64	7	5	< 1	5
512	52	44	4	14
1728	190	142	13	30
4096	462	329	56	96

T_{SCF} is the time taken by PWSCF to obtain the ground state wavefunctions at all N_{kp} k-points of the Monkhorst–Pack mesh, $T_{M,A}$ is the time taken by pw2wannier90 to calculate $M_{mn}^{(\mathbf{k},\mathbf{b})}$ and $A_{mn}^{(\mathbf{k})}$, and T_{W90} is the time taken by wannier90 to localise and plot the MLWF. N_{iter} is the number of iterations required to minimise the total spread.

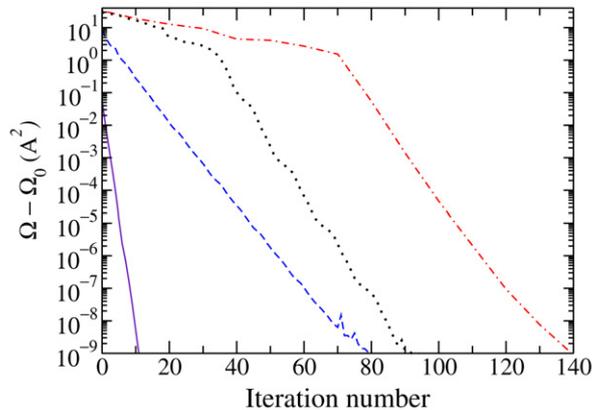


Fig. 1. Convergence of total spread Ω for crystalline silicon with an $8 \times 8 \times 8$ Monkhorst–Pack k-point grid. Solid curve: bond-centred s projection. Dashed curve: atom-centred sp^2 -like projection. Dotted curve: atom-centred s and p projection. Dot-dashed curve: randomly centred spherical s-type projection. Ω_0 is the converged total spread and is independent of the initial projection used.

wannier90 includes a Γ -only branch of algorithms that is able to exploit these symmetries. These algorithms may be activated by means of a logical keyword in the input file and they rely on the above relations being satisfied, i.e., the Bloch wavefunctions must be real.

There are several other advantages beside the significant decrease in the number of operations. In the disentanglement procedure [2], the diagonalisation of a complex Hermitian matrix at each iteration step is replaced by that of a real symmetric matrix, which ensures that $\mathbf{U}^{\text{dis}(\Gamma)}$ is real. The localisation procedure in the Γ -only branch adopts an efficient algorithm proposed by Gygi et al. [30]. This method minimises Ω_{OD} (not Ω) by simultaneously diagonalising a set of real symmetric matrices, $\{\text{Re}\mathbf{M}^{(\Gamma,\mathbf{b})}, \text{Im}\mathbf{M}^{(\Gamma,\mathbf{b})}\}$. The $\mathbf{U}^{(\Gamma)}$ from this method is inherently real, giving real MLWF as a final product.

It is worth mentioning that these MLWF are not necessarily identical to those obtained from the MV localisation procedure unless Ω_{D} is strictly zero due to symmetry of the system.

3.4. Representative timings and convergence

In this section, timings and convergence characteristics for representative calculations are presented. The initial band-structure calculations are performed using the PWSCF code [7]. The self-consistent ground state is obtained, then the required overlap matrices and projections are calculated using the post-processing routine pw2wannier90, supplied with the PWSCF distribution. wannier90 is then used to obtain the MLWF. The PWSCF and pw2wannier90 calculations were performed on a four-processor (dual dual-core) Intel Woodcrest 5130 2.0 GHz desktop computer and the wannier90 calculations on a single processor of the same machine. The gauge-invariant and gauge-dependent spreads were converged to a tolerance of 10^{-10} \AA^2 .

3.4.1. Crystalline silicon

First, we consider a two-atom unit cell of crystalline silicon. A kinetic-energy cutoff of 25 Ry is used for the plane-wave expansion of the valence wavefunctions. The core-valence interaction is described by means of norm-conserving pseudopotentials in separable Kleinman–Bylander form [31]. Table 1 gives timings for obtaining the valence MLWF for various densities of Monkhorst–Pack k-point grid. Bond-centred s-type functions were used for the initial projection. It can be seen that the most time consuming part of the procedure is the computation of the band structure, followed by construction of $M_{mn}^{(\mathbf{k},\mathbf{b})}$ and $A_{mn}^{(\mathbf{k})}$, with wannier90 taking only 5–7% of the total time to perform the localisation of the MLWF and to write files for their visualisation.

In Fig. 1 the effect of using different initial projections for the MLWF is demonstrated. It is worth noting that even when choosing randomly-centred s-type functions the correct MLWF are found in less than 10% of the total time.

Table 2
Timings (in seconds) for MLWF in fullerene from general k-point and Γ -specific schemes

Projection	Γ	N	$T_{M,A}$	T_{W90}	T_{dis}	$N_{\text{iter}}^{\text{dis}}$	T_{loc}	$N_{\text{iter}}^{\text{loc}}$
Randomly-centred s	F	120	901	33	n/a	n/a	33	226
	T	120	368	1	n/a	n/a	1	16
Bond-centred s	F	120	893	18	n/a	n/a	18	117
	T	120	372	<1	n/a	n/a	<1	11
Bond-centred s	F	150	1595	46	22	75	24	83
Atom-centred p_z	T	150	623	12	11	75	1	9

Different initial projections are tested. ‘‘T’’ in the second column (Γ) indicates that the Γ -only branch of algorithms is used. N is the number of MLWF obtained. $T_{M,A}$ is the time taken to calculate $M_{mn}^{(\mathbf{k},\mathbf{b})}$ and $A_{mn}^{(\mathbf{k})}$, and T_{W90} is the total time taken by `wannier90` to disentangle and localise MLWF. The individual time and the number of iterations for each of these operations are given as T_{oper} and $N_{\text{iter}}^{\text{oper}}$ (where oper = dis or loc), respectively.

3.4.2. Fullerene

We consider an isolated fullerene molecule (C_{60}) as a test case for the Γ -only algorithms. The molecule was placed in a cubic supercell of side-length $40 a_0$ (21.2 \AA), and the calculations were performed with Γ -point sampling, ultrasoft pseudopotentials [32], and a plane-wave cut-off of 30 Ry.

Fullerene has 120 valence states in total. For the localisation of the valence MLWF, two choices of initial projections are compared, one being 120 randomly-centred s-orbitals and the other being 120 s-orbitals placed on the MLWF centres generated by the former run. The disentanglement procedure is tested on the extended 150 states comprising 120 valence states and 30 states that cover the full π^* -manifold. Those 30 states are disentangled from the 100 unoccupied eigenfunctions spanning up to 7.5 eV above the HOMO level. The final set of MLWF are found to consist of 60 atom-centred p_z orbitals and 90 bond-centred σ -bonding orbitals, similar to the MLWF shown in Fig. 4.

Timings for these calculations are summarised in Table 2. $T_{M,A}$ decreases by 60% when the Γ -only branch is used, because only half the \mathbf{b} -vectors are needed and the post-processing routine `pw2wannier90` is optimised to take full advantage of real wavefunctions.

The minimisation method used in the Γ -only branch is apparently more efficient than the conjugate-gradient algorithm, especially when the initial projections are far from the final, converged MLWF. Ω_D vanishes in the case of a simple cubic lattice, and Ω from the two different minimisation methods converges to an identical value up to the given tolerance. T_{dis} decreases by 50% for the reason discussed in Section 3.3, but $N_{\text{iter}}^{\text{dis}}$ is the same, as it should be, and Ω_I is identical within machine precision.

4. Structure of the program

The schematic structure of the program is outlined in Fig. 2. Each box represents a Fortran90 module and the lines represent module dependencies. Modules only use data and subroutines from lower modules in the diagram. A description of the purpose of each module is given below.

- `constants`: definition of constants (e.g., π).
- `io`: error handling, timing, and input and output units.
- `utility`: commonly used operations such as conversion of Cartesian to fractional co-ordinates, string functions, matrix multiplication wrapper, etc.
- `parameters`: all physical parameters relevant to the calculation and subroutines for reading them from the input file at the start of a calculation. Subroutines for writing checkpoint files for restarting previous runs.
- `kmesh`: set up of the framework for reciprocal-space derivatives and, if in post-processing mode, writing a file which communicates to an ab initio code information on how to calculate $M_{mn}^{(\mathbf{k},\mathbf{b})}$ and $A_{mn}^{(\mathbf{k})}$.
- `overlap`: reading of the overlap matrices $M_{mn}^{(\mathbf{k},\mathbf{b})}$ and $A_{mn}^{(\mathbf{k})}$.
- `disentangle`: if using entangled energy bands, finding the optimal subspace within a specified energy window by minimising the gauge-invariant spread.
- `wannierise`: finding the unitary transformations $U_{mn}^{(\mathbf{k})}$ amongst the energy bands which give rise to MLWF.
- `plot`: routines for output of Wannier functions, Fermi surface and band structure in file formats suitable for visualisation.
- `wannier_prog`: the main program.
- `wannier_lib`: library routines. `wannier90` may be invoked directly from within an ab initio code as a set of library calls. The reader is referred to Section 6.1 for a brief description and to the documentation in the `wannier90` distribution for full details.

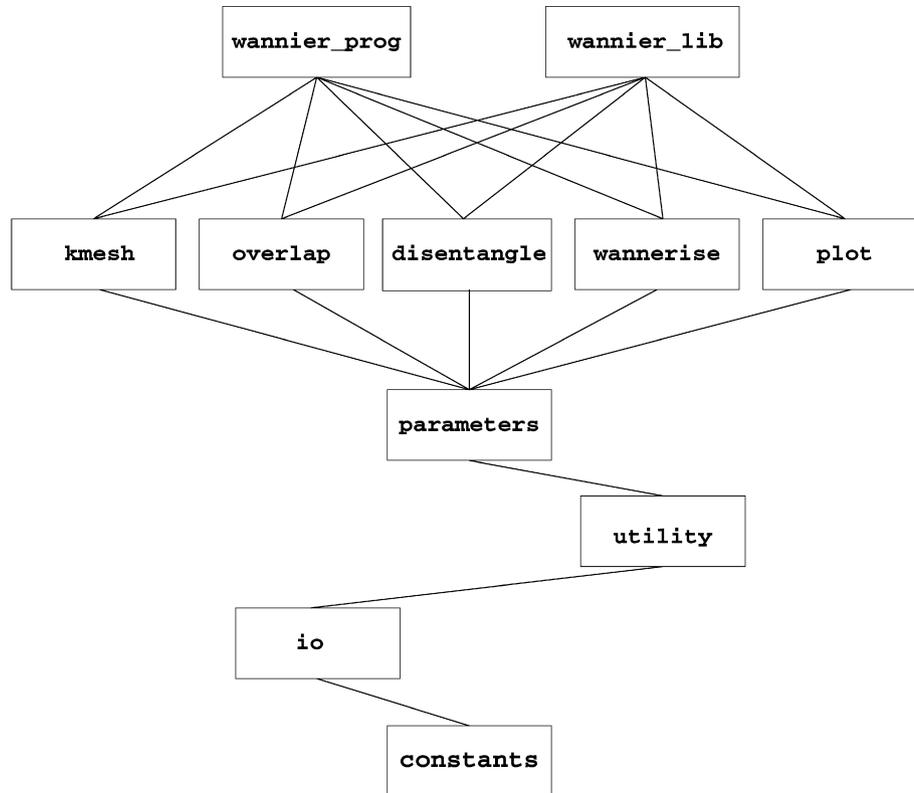


Fig. 2. Schematic structure of the program.

5. Installation

wannier90 is distributed as a gzipped tar file (<http://www.wannier.org/>). On Linux platforms, for example, it may be unpacked by typing

```
> tar zxvf wannier90.tar.gz
```

which creates a directory containing the source files, documentation, examples, etc.

Compilation is straightforward. From the `config` directory of the distribution the user should select the `make.sys.plat` file which corresponds most closely to the platform being used and copy it to the root directory of the distribution, renaming it `make.sys` in the process. The values of system-dependent parameters (e.g., the location of the BLAS and LAPACK libraries, the name of the Fortran compiler, the Fortran optimisation flags, etc.) that are defined therein should be modified to correspond to the user's particular system. Once this has been done, typing `make` from the root directory of the distribution will create an executable `wannier90.x`. Further details and make options may be found in the file `INSTALL.readme` in the root directory of the distribution.

6. Running wannier90

Before running `wannier90` the user must perform a self-consistent first-principles calculation on the system of interest in order to obtain a set of Bloch energy bands from which MLWF may then be constructed. Once the Bloch bands have been computed `wannier90` may be operated in a post-processing mode as described below.

The master input file for `wannier90` is called `seedname.win`, where `seedname` is the prefix of all of the input and output files. The input system is designed to be simple and user-friendly and is described comprehensively in the documentation that is contained within the `wannier90` distribution. An example input file is provided in [Appendix A](#).

`wannier90` must be run twice. On the first pass the command line option `-pp` must be used, as follows:

```
> wannier90.x -pp seedname
```

This causes the code to read the master input file `seedname.win` and generate the file `seedname.nnkp` that contains all the information necessary to construct the overlap matrices $M_{mn}^{(k,b)}$ and $A_{mn}^{(k)}$ from the Bloch bands already obtained from a first-principles calculation. In order to interface an ab initio code to `wannier90` one needs to write subroutines that read `seedname.nnkp`, compute the overlap matrix $M_{mn}^{(k,b)}$ and, optionally, $A_{mn}^{(k)}$ from the Bloch bands and write these matrices in the

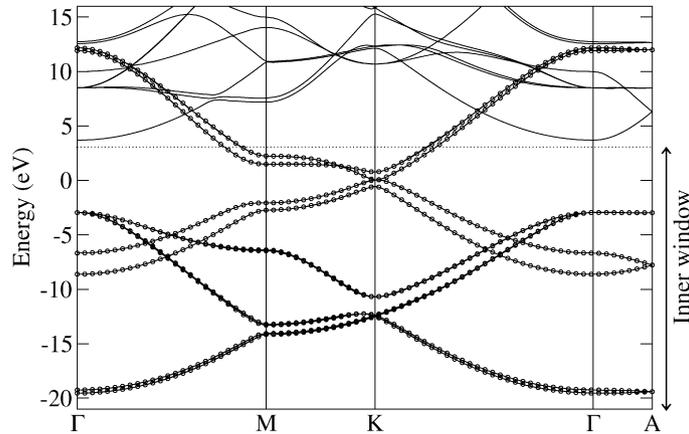


Fig. 3. Band structure of graphite. Solid lines: original band structure from a conventional first-principles calculation. Dotted lines: Wannier-interpolated band structure. The zero of energy is the Fermi level.

appropriate format to files called `seedname.mmn` and `seedname.amn`, respectively. If using the disentanglement procedure or plotting a band structure, density of states or Fermi surface, `wannier90` also requires the eigenvalues $\varepsilon_{n\mathbf{k}}$ corresponding to the Bloch states $\psi_{n\mathbf{k}}(\mathbf{r})$, which should be written to a file called `seedname.eig`. The reader is referred to the documentation that is contained within the `wannier90` distribution for complete details of these files.

Once the necessary files have been written, `wannier90` must be run again, this time without any command-line options:

```
> wannier90.x seedname
```

On this pass, the code reads the overlap matrices and eigenvalues (if required) from file and performs the maximal-localisation procedure as outlined in Section 2, writing the output to a file `seedname.wout`.

At the time of writing, the PWSCF code (a part of the QUANTUM-ESPRESSO package [7]), which is available under GNU General Public License, has a `wannier90` interface in the form of a post-processing program called `pw2wannier90`. It is the authors' hope that `wannier90` is a sufficiently useful tool for investigators to be motivated to write interfaces to other electronic structure codes.

6.1. Library mode

`wannier90` may also be compiled as a library and invoked with subroutine calls from within an ab initio code. The command

```
> make lib
```

creates a library `libwannier.a` in the root directory of the distribution. The library mode of `wannier90` works along exactly the same lines as the post-processing mode, described above. The formal difference is that in the latter, information is passed between the ab initio code and `wannier90` via *files*, such as `seedname.nnkp` and `seedname.mmn`, etc., whereas, in the former, it is done via direct *calls* to library subroutines. As before, there are two stages: first, a call to the library subroutine `wannier_setup`, which returns the information necessary to construct the overlap matrices $M_{mn}^{(\mathbf{k},\mathbf{b})}$ and $A_{mn}^{(\mathbf{k})}$ from the Bloch bands; second, a call to the subroutine `wannier_run`, which takes as input the $M_{mn}^{(\mathbf{k},\mathbf{b})}$ matrix (and $A_{mn}^{(\mathbf{k})}$ and the eigenvalues $\varepsilon_{n\mathbf{k}}$, if required) and performs the maximal-localisation procedure as outlined in Section 2. The reader is referred to the documentation in the `wannier90` distribution for further details.

7. Examples

7.1. Graphite

As an example we consider the generation of MLWF to describe the states around and below the Fermi level in Bernal (A-B-A) graphite. From the bandstructure (Fig. 3) we expect that the minimum number of Wannier functions needed to describe these states is 10 per unit cell (2.5 per atom). We shall see that this choice corresponds to an intuitive chemical description of the system.

We perform the initial band-structure calculations using the PWSCF code [7]. A kinetic-energy cutoff of 30 Ry is used for the plane-wave expansion of the valence wavefunctions. The core-valence interaction is described by means of norm-conserving pseudopotentials in separable Kleinman–Bylander form [31]. We obtain the self-consistent ground state using a $16 \times 16 \times 16$ Monkhorst–Pack mesh of k-points [22] and a fictitious Fermi smearing [3] of 0.02 Ry for the Brillouin-zone integration. Then, we freeze the self-consistent potential and perform a non-self-consistent calculation on a uniform $6 \times 6 \times 6$ grid of k-points. At each

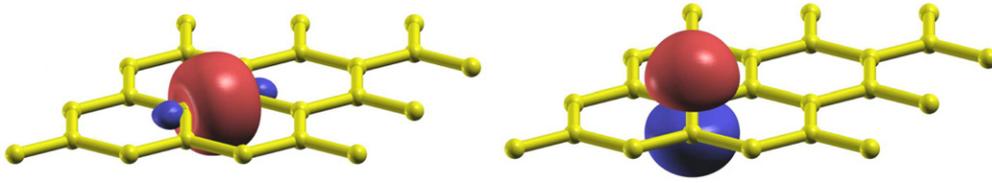


Fig. 4. Isosurface contours of MLWF in graphite (red for positive value and blue for negative). [left] σ -type MLWF. [right] p_z -type MLWF. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

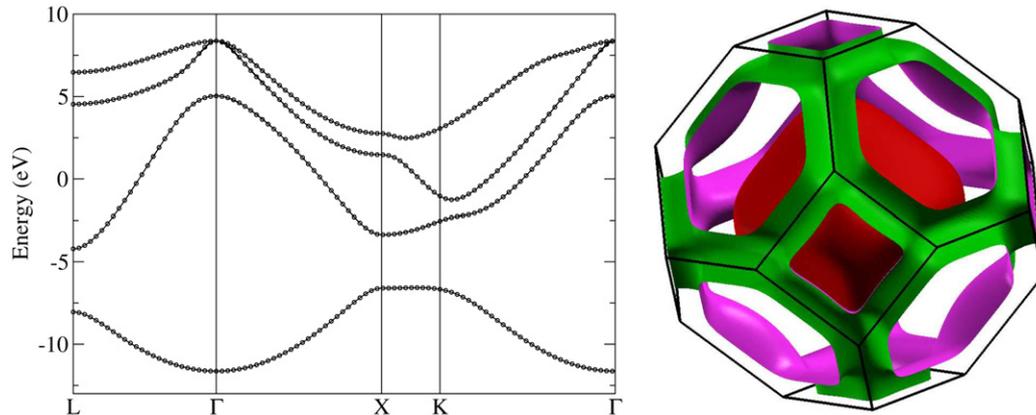


Fig. 5. [left] Band structure of lead. Solid lines: original band structure from a conventional first-principles calculation. Dotted lines: Wannier-interpolated band structure. The zero of energy is the Fermi level. [right] Fermi surface of lead.

k-point we calculate the first 20 bands. The required overlap matrices and projections are calculated using the post-processing routine `pw2wannier90`, supplied with the PWSCF distribution. Projections onto atom centred sp^2 and p_z functions are used to construct the initial guess, and `wannier90` is used to obtain the MLWF. The gauge-dependent and gauge-independent spreads converge to machine precision in 300 and 70 steps, respectively. The resulting MLWF are a set of six symmetry-related bond-centred σ -orbitals and four atom-centred p_z -orbitals, shown in Fig. 4 (as plotted with the XCrySDen package [33]). Similar MLWF were obtained for carbon nanotubes [14]. In Fig. 3 we show the band structure of graphite obtained using Wannier interpolation and compare it to the band structure obtained from a full first-principles calculation. Within the inner energy window the agreement is essentially perfect.

7.2. Lead

Let us consider the generation of MLWF to describe the states around and below the Fermi level in lead. As relativistic effects can be significant in heavy elements, we include spin–orbit coupling in our electronic structure calculation. In lead, the 6s and 6p states form an isolated set of bands around the Fermi level. The ground-state structure of lead has both time-reversal and inversion symmetries, so that each band is two-fold degenerate. We will obtain a set of eight MLWF to describe these states.

Once again we perform the initial band-structure calculations using the PWSCF code. A kinetic-energy cutoff of 45 Ry is used for the plane-wave expansion of the valence wavefunctions. The core–valence interaction is described by means of spin–orbit coupled norm-conserving pseudopotentials [34] in separable Kleinman–Bylander form. We obtain the self-consistent ground state using a $16 \times 16 \times 16$ Monkhorst–Pack mesh of k-points and a fictitious Fermi smearing [3] of 0.02 Ry for the Brillouin-zone integration. The self-consistent potential is frozen and we perform a non-self-consistent calculation on a uniform $12 \times 12 \times 12$ grid of k-points. Then we calculate the required overlap matrices and projections using `pw2wannier90`. For the initial projection we use orbitals with sp^3 symmetry, four projections onto spin-up states and four onto spin-down states.

`wannier90` is used to obtain the MLWF. The gauge-dependent spread is converged in 200 steps. In Fig. 5 we show both the band structure and Fermi surface obtained using Wannier interpolation. Although the spin–orbit induced splitting is large at certain points in the band structure (e.g., 3 eV at Γ), the Fermi surface is not significantly different from a scalar-relativistic calculation. The reader is referred to Ref. [12] for a detailed discussion of how to construct spinor Wannier functions for ferromagnetic systems with broken time-reversal symmetry.

8. Conclusions

In conclusion, we present a new code called `wannier90` for computing maximally-localised Wannier functions. `wannier90` is freely available under the GNU General Public Licence [35]. It is very user-friendly and is written in Fortran90, employing

modern programming techniques that enable the addition of further functionality, such as transport properties or interpolation of electron–phonon couplings, in an easy and modular fashion. `wannier90` has been seamlessly interfaced to the PWSCF plane-wave DFT code [7] and, at the time of writing, interfaces to other codes, e.g., ABINIT [36], CASTEP [37] and FLEUR [38], are in progress. We hope that the availability of `wannier90` will encourage the wider use of maximally-localised Wannier functions in the electronic structure community.

Acknowledgements

We would like to thank Stefano de Gironcoli, Malgorzata Wierzbowska and Maria Peressi for the interface to the PWSCF first-principles code [7]. This work was funded by DARPA-PROM and by the Laboratory Directed Research and Development Program of Lawrence Berkeley National Laboratory under the Department of Energy Contract No. DE-AC02-05CH11231.

Appendix A. Sample files

Input file:

```

num_bands      =    20
num_wann       =    10
dis_win_max    =   19.2
dis_froz_max   =    9.8
dis_num_iter   =   400
num_iter       =   100

Begin Atoms_Frac
  C1    0.0000000000    0.0000000000    0.7500000000
  C1    0.3333333333    0.6666666667    0.2500000000
  C2    0.0000000000    0.0000000000    0.2500000000
  C2   -0.3333333333   -0.6666666667    0.7500000000
End Atoms_Frac

Begin Unit_Cell_Cart
  2.1304215583   -1.2299994602    0.0000000000
  0.0000000000    2.4599989204    0.0000000000
  0.0000000000    0.0000000000    6.8000000000
End Unit_Cell_Cart

Begin Projections
  C1:sp2;pz
  C2:pz
End Projections

bands_plot = true

begin kpoint_path
  G 0.0000  0.0000  0.0000  M 0.5000  -0.5000  0.0000
  M 0.5000  -0.5000  0.0000  K 0.6667  -0.3333  0.0000
  K 0.6667  -0.3333  0.0000  G 0.0000  0.0000  0.0000
  G 0.0000  0.0000  0.0000  A 0.0000  0.0000  0.5000
end kpoint_path

mp_grid      = 6 6 6

begin kpoints
  0.00000000  0.00000000  0.00000000
  0.00000000  0.00000000  0.16666667
  0.00000000  0.00000000  0.33333333
  0.00000000  0.00000000  0.50000000
  0.00000000  0.00000000  0.66666667
  0.00000000  0.00000000  0.83333333
  0.00000000  0.16666667  0.00000000
.
```

```

.
.
0.83333333 0.83333333 0.66666667
0.83333333 0.83333333 0.83333333
End Kpoints

```

Output file (truncated):

```

-----
Final State
WF centre and spread 1 ( -0.354954, 0.614798, 5.100000 ) 0.59341185
WF centre and spread 2 ( -0.354954, -0.614798, 5.100000 ) 0.59341185
WF centre and spread 3 ( 0.709907, 0.000000, 5.100000 ) 0.59341190
WF centre and spread 4 ( 0.000000, 0.000000, 5.100000 ) 1.04236015
WF centre and spread 5 ( 0.354954, 1.845201, 1.700000 ) 0.59341120
WF centre and spread 6 ( 0.354954, 0.614798, 1.700000 ) 0.59341085
WF centre and spread 7 ( 1.420514, 1.230000, 1.700000 ) 0.59341100
WF centre and spread 8 ( 0.710140, 1.229999, 1.700000 ) 1.04371374
WF centre and spread 9 ( 0.000000, 0.000000, 1.700000 ) 1.04233128
WF centre and spread 10 ( -0.710140, -1.229999, 5.100000 ) 1.04368702
Sum of centres and spreads ( 2.130421, 3.689998, 33.999999 ) 7.73256084

      Spreads (Ang^2)      Omega I      =      6.121019675
      =====
      Omega D      =      0.032181361
      Omega OD     =      1.579353494
Final Spread (Ang^2)      Omega Total  =      7.732554530
-----

```

References

- [1] N. Marzari, D. Vanderbilt, Phys. Rev. B 56 (1997) 12847.
- [2] I. Souza, N. Marzari, D. Vanderbilt, Phys. Rev. B 65 (2001) 035109.
- [3] N. Marzari, D. Vanderbilt, M.C. Payne, Phys. Rev. Lett. 79 (1997) 1337.
- [4] See <http://www.cpmid.org/>.
- [5] P.L. Silvestrelli, Phys. Rev. B 59 (1999) 9703.
- [6] M. Posternak, A. Baldereschi, S. Massidda, N. Marzari, Phys. Rev. B 65 (2002) 184422.
- [7] S. Baroni, et al., <http://www.quantum-espresso.org/>.
- [8] A.J. Williamson, R.Q. Hood, J.C. Grossman, Phys. Rev. Lett. 87 (2001) 246406.
- [9] D.M. Whittaker, M.P. Croucher, Phys. Rev. B 67 (2003) 085204.
- [10] M. Schillinger, S.F. Mingaleev, D. Hermann, K. Busch, Highly localized Wannier functions for the efficient modeling of photonic crystal circuits, in: A. Adibi, Y. Lin, A. Scherer (Eds.), Photonic Crystal Materials and Devices III, Proceedings of SPIE, vol. 5733, SPIE, Bellingham, WA, 2005, pp. 324–335.
- [11] M. Stengel, N.A. Spaldin, Phys. Rev. B 75 (2007) 205121.
- [12] X. Wang, J.R. Yates, I. Souza, D. Vanderbilt, Phys. Rev. B 74 (2006) 195118.
- [13] F. Giustino, J.R. Yates, I. Souza, M.L. Cohen, S.G. Louie, Phys. Rev. Lett. 98 (2007) 047005.
- [14] Y.-S. Lee, M.B. Nardelli, N. Marzari, Phys. Rev. Lett. 95 (2005) 076804.
- [15] A.A. Mostofi, N. Marzari, in preparation.
- [16] W. Ku, H. Rosner, W.E. Pickett, R.T. Scalettar, Phys. Rev. Lett. 89 (2002) 167204.
- [17] V.I. Anisimov, A.V. Kozhevnikov, Phys. Rev. B 72 (2005) 075125.
- [18] F. Lechermann, et al., Phys. Rev. B 74 (2006) 125120.
- [19] O.K. Andersen, T. Saha-Dasgupta, Phys. Rev. B 62 (2000) R16219.
- [20] E. Zurek, O. Jepsen, O.K. Andersen, Chem. Phys. Chem. 6 (2005) 1934.
- [21] E.I. Blount, Formalisms of band theory, in: F. Seitz, D. Turnbull (Eds.), Solid State Physics, vol. 13, Academic Press, New York, 1962, pp. 305–373.
- [22] H.J. Monkhorst, J.D. Pack, Phys. Rev. B 13 (1976) 5188.
- [23] K.S. Thygesen, L.B. Hansen, K.W. Jacobsen, Phys. Rev. Lett. 94 (2005) 026405.
- [24] C. Brouder, G. Panati, M. Calandra, C. Mourougane, N. Marzari, Phys. Rev. Lett. 98 (2007) 046402.
- [25] J.R. Yates, X. Wang, I. Souza, D. Vanderbilt, Phys. Rev. B 75 (2007) 195121.
- [26] J.C. Slater, G.F. Koster, Phys. Rev. 94 (1954) 1498.
- [27] J.R. Yates, A.A. Mostofi, I. Souza, N. Marzari, in preparation.
- [28] G.H. Golub, C.F.V. Loan, Matrix Computations, third ed., John Hopkins, New York, 1996 (Chapter 2).
- [29] M. Stengel, N.A. Spaldin, Phys. Rev. B 73 (2006) 075121.
- [30] F. Gygi, J.-L. Fattebert, E. Schwegler, Comput. Phys. Comm. 155 (2003) 1.
- [31] L. Kleinman, D. Bylander, Phys. Rev. Lett. 48 (1982) 1425.
- [32] D. Vanderbilt, Phys. Rev. B 41 (1990) 7892.
- [33] A. Kokalj, Comp. Mater. Sci. 28 (2003) 155.

- [34] A.D. Corso, A.M. Conte, *Phys. Rev. B* 71 (2005) 115106.
- [35] See <http://www.gnu.org/> for details.
- [36] X. Gonze, et al., *Comp. Mater. Sci.* 25 (2002) 478.
- [37] S.J. Clark, et al., *Z. Kristallogr.* 220 (2005) 567.
- [38] See <http://www.flapw.de/>.