Wannier_tools User guide 2.0

QuanSheng Wu (wuquansheng@gmail.com)
ShengNan Zhang (springnancy8586@163.com)
29 Jun 2015
31 Mar 2016 @ Dubendorf
19 April 2016 @ Lausanne
08 May 2016 @Dubendorf
06 June 2016 @Jouvence

1. Installaion

(a) Check out the repository by " git clone https://github.com/quanshengwu/wannier_tools.git" or download the zip file.
(b) cd wannier_tools/soc directory, Edit Makefile, Change the blas library " libs= " to your mkl library, you can use your own blas and lapack library. Then type make. The executive binary is copied to directory wannier_tools/bin. PS, Now I have only tested it with intel compiler, not GFORTRAN.
(c) add the wannier_tools/bin path to your PATH by "export PATH=yourdirecory/wannier_tools/bin:$PATH"
(d) Look for more information and examples on wiki https://github.com/quanshengwu/wannier_tools/wiki

2. The introduction of input file

Before executing wann_tools, you should edit the input file in the directory wannier_tools/example by your own necessary. Here we take Bi2Se3 as an example.

In version 2.0, we updated the format of input.dat. The input file is structured in a number of NAMELIST and INPUT_CARDS. Look the table carefully. We will illustrate it later.

```
&TB_FILE
Hrfile = wannier90_hr.dat
/
```

```
&CONTROL
BulkBand_calc         = T       ! bulk band structure calculation flag
BulkFS_calc           = F
BulkGap_cube_calc     = F
BulkGap_plane_calc    = F
SlabBand_calc         = T
WireBand_calc         = F
SlabSS_calc           = T
SlabArc_calc          = T
SlabSpintexture_calc  = T
wanniercenter_calc    = F
BerryPhase_calc       = F
BerryCurvature_calc   = F
/

&SYSTEM
Nslab = 10
Nslab1= 6
Nslab2= 6
NumOccupied = 18        ! NumOccupied
SOC = 1                 ! soc
E_FERMI = 4.4195        ! e-fermi
Bx= 0, By= 0, Bz= 0     ! Bx By Bz
surf_onsite= 0.0        ! surf_onsite
/

&PARAMETERS
E_arc = 0.0         ! energy for calculate Fermi Arc
Eta_Arc = 0.001     ! infinite small value, like brodening
OmegaNum = 500      ! omega number
OmegaMin = -0.6     ! energy interval
OmegaMax =  0.5     ! energy interval
Nk1 = 200           ! number k points
Nk2 = 200           ! number k points
Nk3 = 50            ! number k points
NP = 2              ! number of principle layers
Gap_threshold = 1.0 ! threshold for GapCube output
```

```
/

LATTICE
Angstrom
-2.069  -3.583614  0.000000     ! crystal lattice information
 2.069  -3.583614  0.000000
 0.000   2.389075  9.546667

ATOM_POSITIONS
5                          ! number of atoms for projectors
Direct                       ! Direct or Cartisen coordinate
 Bi 0.3990    0.3990    0.6970
 Bi 0.6010    0.6010    0.3030
 Se 0      0      0.5
 Se 0.2060    0.2060    0.1180
 Se 0.7940    0.7940    0.8820

PROJECTORS
 3 3 3 3 3          ! number of projectors
Bi px py pz         ! projectors
Bi px py pz
Se px py pz
Se px py pz
Se px py pz

SURFACE            ! See doc for details
 1  0  0
 0  1  0
 0  0  1

KPATH_BULK            ! k point path
4              ! number of k line only for bulk band
G 0.00000 0.00000 0.0000 Z 0.00000 0.00000 0.5000
Z 0.00000 0.00000 0.5000 F 0.50000 0.50000 0.0000
F 0.50000 0.50000 0.0000 G 0.00000 0.00000 0.0000
G 0.00000 0.00000 0.0000 L 0.50000 0.00000 0.0000
```

```
KPATH_SLAB
2        ! numker of k line for 2D case
K 0.33 0.67 G 0.0 0.0  ! k path for 2D case
G 0.0 0.0 M 0.5 0.5

KPLANE_SLAB
-0.1 -0.1      ! Original point for 2D k plane
 0.2  0.0      ! The first vector to define 2D k plane
 0.0  0.2      ! The second vector to define 2D k plane  for arc plots

KPLANE_BULK
-0.50 -0.50  0.00   ! Original point for 3D k plane
 1.00  0.00  0.00   ! The first vector to define 3d k space plane
 0.00  1.00  0.00   ! The second vector to define 3d k space plane


KCUBE_BULK
-0.50 -0.50 -0.50   ! Original point for 3D k plane
 1.00  0.00  0.00   ! The first vector to define 3d k space plane
 0.00  1.00  0.00   ! The second vector to define 3d k space plane
 0.00  0.00  1.00   ! The third vector to define 3d k cube
```

NAMELISTS are a standard input construct in Fortran90. The use of NAMELISTS is relatively flexible. All the variables in the NAMELISTS have default values. You should set them only when it is needed. Variables can be inserted in any order. Besides, the comments started with the exclamation mark (!) can be inserted in it. Like this:

```
&NAMELIST
needed_variable2=XX, needed_variable1=YY,
character_variable1='a suitable string'
/
```

There are 4 NAMELISTS included in input.dat. They are TB_FILE, SYSTEM, CONTROL, PARAMETERS.

TB_FILE NAMELISTS

```
&TB_FILE
Hrfile = wannier90_hr.dat
/
```

gives the name of the tight-binding Hamiltonian. At present, we use the format of wannier90_hr.dat specified in Wannier90.

SYSTEM NAMELIST.

```
&SYSTEM
Nslab = 10
Nslab1= 6
Nslab2= 6
NumOccupied = 18        ! NumOccupied
SOC = 1                 ! soc
E_FERMI = 4.4195        ! e-fermi
Bx= 0, By= 0, Bz= 0     ! Bx By Bz
surf_onsite= 0.0        ! surf_onsite
/
```

In this namelists, we specify the system you need to compute. Including Nslab, Nslab1, Nslab2, NumOccupied, SOC, E_FERMI, Bx, By, Bz, surf_onsite.

NSlab: integer, Number of slabs for slab band, The default value is 10.
NSlab1, Nslab2: integers, The thickness of nano ribbon. If you don't want to calculate the band structure of it, then don't set it. The default values are Nslab1= 1, Nslab2= 1.
NumOccupied: integer, Number of occupied bands. No default value, you have to set it for Wilson loop and gap calculation.
SOC: integer, Flag for spin-orbital coupling. If SOC=0, it means there is no SOC included in your given tight binding model. if SOC=1 or >0, it means SOC is already included in the tight binding model.

E_FERMI: real-valued, Fermi level for the given tight binding model.
Bx, By, Bz: real-valued, magnetic field value. Ignore it in this version.
surf_onsite: Additional onsite energy on the surface, you can set this to see how surface state changes. But don't set it if you don't know what it is.

NAMELISTS CONTROL:

```
&CONTROL
BulkBand_calc        = T       ! bulk band structure calculation flag
BulkFS_calc          = F
BulkGap_cube_calc    = F
BulkGap_plane_calc   = F
SlabBand_calc        = T
WireBand_calc        = F
SlabSS_calc          = T
SlabArc_calc         = F
SlabSpintexture_calc = T
wanniercenter_calc   = F
BerryCurvature_calc  = F
/
```

In this name list, you can set the keywords to setup the tasks. All these tasks can be set to be true at the same time. We listed the features in the table below.

Table I Features

Flag option	**Function**	**Output**	**Tested**
BulkBand_calc	Band structure for bulk	bulkek.dat, bulkek.gnu	yes
BulkFS_calc	Fermi surface for bulk system	FS3D.dat, FS3D.bxsf	yes
	Energy gap for a		

BulkGap_cube_calc	given k cube for bulk system	GapCube.dat, GapCube.gnu	yes
BulkGap_plane_calc	Energy gap for a given k plane for bulk system	GapPlane.dat, GapPlane.gnu	yes
SlabBand_calc	Band structure for slab system	slabek.dat,slabek.gnu	yes
WireBand_calc	Band structure for ribbon system	ribbonek.dat,ribbonek.gnu	yes
SlabSS_calc	Surface spectrum A(k,E) along a k line and energy interval for slab system	dos.dat_l, dos.dat_r surfdos_l.gnu,surfdos_r.gnu	yes
SlabArc_calc	Surface spectrum A(k1,k2,E0) for a fixed energy E0 for slab system	arc.dat_l, arc.dat_r arc_l.gnu,arc_r.gnu	yes
SlabSpintexture_calc	Spin texture for slab system	spindos.dat, spintext.dat, spintext.gnu	yes
wanniercenter_calc	Wannier centers (Wilson loop) for slabs system	wcc.dat, wcc.gnu	yes
BerryCurvature_calc	Berry Curvature for a given k plane for bulk system	BerryCurvature.dat, BerryCurvature.gnu	yes

NAMELISTS PARAMETERS:

```
&PARAMETERS
E_arc = 0.0            ! energy for calculate Fermi Arc
Eta_Arc = 0.001     ! infinite small value, like broadening
OmegaNum = 200      ! omega number
OmegaMin = -0.6     ! energy interval
OmegaMax =  0.5     ! energy interval
Nk1 = 50            ! number k points
Nk2 = 50            ! number k points
Nk3 = 50            ! number k points
NP = 2             ! number of principle layers
Gap_threshold = 1.0    ! threshold for GapCube output
/
```

In this namelists, we listed some parameters necessary in the task you specified in namelists CONTROL.

E_arc : real-valued, energy for calculate Fermi arc, used if SlabArc_calc = T. The default value is 0.0.

Eta_Arc : real-valued, infinite same value for broadening used in Green's function calculation. used if SlabArc_calc = T. Default value is 0.001.

[**OmegaMin, OmegaMax**] : real-valued, energy interval for surface state calculation. used if SlabSS_calc= T. No default value.

OmegaNum : integer valued, Number of slices in the energy interval [**OmegaMin, OmegaMax**]. used if SlabSS_calc= T. The default value is 100.

Nk1, Nk2, Nk3 : integer valued, Number of k points for different purpose. I will explain that later. Default value is Nk1=20, Nk2=20, Nk3=20.

NP : integer valued, Number of principle layers, see details related to iterative green's function. Used if SlabSS_calc= T, SlabArc_calc=T, SlabSpintexture_calc=T. Default value is 2. You need to do a convergence test by setting Np= 1, Np=2, Np=3, and check the surface state spectrum. Basically, the value of Np depends on the spread of Wannier functions you constructed. One thing should be mentioned is that the computational time grows cubically of Np.

Gap_threshold : real valued. This value is used when you do energy gap calculation like **BulkGap_cube_calc=T**, **BulkGap_plane_calc=T**. The k points will be printed out in a file when the energy gap is smaller than Gap_threshold.

INPUT_CARD format.

The second important format in input.dat is the input_card format, which is relatively fixed format. First, we need a keyword like LATTICE, which is name of this card. After this keyword, the number of lines is fixed until it is done.

```
LATTICE
Angstrom
-2.069  -3.583614  0.000000     ! crystal lattice information
 2.069  -3.583614  0.000000
 0.000   2.389075  9.546667
```

There are 9 INPUT_CARDS in the input.dat. There is no order between the INPUT_CARDS. And any comments or blank lines could be added between the INPUT_CARDS. Lets introduce them one by one.

LATTICE CARDS: In this card, we set three lattice vectors coordinates. For the unit, you can use both Angstrom and Bohr. However, in the program, we use Angstrom. Bohr unit will be transformed to Angstrom automatically. No default values for the LATTICE CARD.

```
ATOM_POSITIONS
5                          ! number of atoms for projectors
Direct                       ! Direct or Cartisen coordinate
 Bi 0.3990    0.3990    0.6970
 Bi 0.6010    0.6010    0.3030
 Se 0     0     0.5
 Se 0.2060    0.2060    0.1180
 Se 0.7940    0.7940    0.8820
```

ATOM_POSITIONS CARDS: In this card, we set the atom's position. Notes: 1, Here the atom means that the atoms with projectors. not the whole atoms in the unit cell. 2, You can use "Direct" or "Cartesian" coordinates. "Direct" means the fractional coordinate based on the primitive lattice vector listed in LATTICE CARDS.

```
PROJECTORS
 3 3 3 3 3          ! number of projectors
Bi px py pz         ! projectors
Bi px py pz
Se px py pz
Se px py pz
Se px py pz
```

PROJECTORS CARDS : In this card, we set the Wannier projectors for the tight binding. Here we don't take into account the spin degeneracy, only consider the orbital part. The name of orbitals should be "s", "px", "py", "pz", "dxy", "dxz", "dyz", "dx2-y2", "dz2". I will add f orbitals latter. The order of the orbitals is very important if you want to analyze the symmetry properties.

```
SURFACE          ! See doc for details
 1  0  0         a11, a12, a13
 0  1  0         a21 a22 a23
 0  0  1         a31 a32 a33
```

SURFACE CARDS : This card is very important slabs and ribbon calculation. You need to read the following text carefully. In this card, we specify the surface you want to investigate. Basically, you should know of which surface you want to investigate, and of which direction you want to simulate the ribbon. So we need to define the new lattice vector system like this, choose two vectors on the surface we want to study, and choose another vector which is not on this plane. The volume of the new cell constructed by these three vectors should be the same of the previous one. And the orientation should be the same by checking $R_1'\cdot(R_2'\times R_3')= R_1\cdot(R_2\times

R_3)$. The relations between old and new basis are
\begin{eqnarray}
 R_1'= a_{11} R_1 + a_{12} R_2 + a_{13} R_3 \\
 R_2'= a_{21} R_1 + a_{22} R_2 + a_{23} R_3 \\
 R_3'= a_{31} R_1 + a_{32} R_2 + a_{33} R_3
\end{eqnarray}
The slab calculations are base on the surface constructed by vector R_1', R_2'. The ribbon is along R_3' direction, two surfaces are constructed by vector R_1', R_3' and R_2', R_3' respectively.

```
KPATH_BULK           ! k point path
4              ! number of k line only for bulk band
G 0.00000 0.00000 0.0000 Z 0.00000 0.00000 0.5000
Z 0.00000 0.00000 0.5000 F 0.50000 0.50000 0.0000
F 0.50000 0.50000 0.0000 G 0.00000 0.00000 0.0000
G 0.00000 0.00000 0.0000 L 0.50000 0.00000 0.0000
```

KPATH_BULK CARD: This is the k path for bulk band structure calculation. This k points are in unit of the reciprocal lattice built by the lattice vector LATTICE CARD. The number of k points is **Nk1**, which is set in NAMELISTS PARAMETERS. There are no default values for this CARD. It is necessary when **BulkBand_calc=T.**

```
KPATH_SLAB
2       ! numker of k line for 2D case
K 0.33 0.67 G 0.0 0.0  ! k path for 2D case
G 0.0 0.0 M 0.5 0.5
```

KPATH_SLAB CARD : This is the k path for slab system, including the band structure calculation and the surface state calculation. It is necessary when **SlabBand_calc=T or SlabSS_calc=T.** Number of k points along the line is **Nk1**.

```
KPLANE_SLAB
```

```
-0.1 -0.1      ! Original point for 2D k plane
 0.2  0.0      ! The first vector to define 2D k plane
 0.0  0.2      ! The second vector to define 2D k plane  for arc plots
```

KPLANE_SLAB CARD: Define a 2D k space plane for arc plots. Line 52 is the start point of the plane. Line 53 and Line 54 are the two vectors define the plane. The number of k points for the 1st and 2nd vector is **Nk1** and **Nk2** respectively. Set this CARD when **SlabArc_calc=T**, **SlabSpintexture_calc= T**. The default values are

```
KPLANE_SLAB
-0.5 -0.5      ! Original point for 2D k plane
 1.0  0.0      ! The first vector to define 2D k plane
 0.0  1.0      ! The second vector to define 2D k plane  for arc plots
```

```
KPLANE_BULK
-0.50 -0.50  0.00   ! Original point for 3D k plane
 1.00  0.00  0.00   ! The first vector to define 3d k space plane
 0.00  1.00  0.00   ! The second vector to define 3d k space plane
```

KPLANE_BULK CARD: The same set as KPLANE_SLAB CARD, but for 3D case. We can use these two vectors to calculate the band gap of a plane in 3D BZ, then we can check whether there are Weyl points or nodal line in that plane. Notice that these vectors is in unit of reciprocal vectors. Set this CARD when **BulkGap_plane_calc=T, BerryCurvature_calc=T, wanniercenter_calc=T.** Default values are

```
KPLANE_BULK
 0.00  0.00  0.00    ! Original point for 3D k plane
 1.00  0.00  0.00   ! The first vector to define 3d k space plane
 0.00  0.50  0.00   ! The second vector to define 3d k space plane
```

```
KCUBE_BULK
-0.50 -0.50 -0.50   ! Original point for 3D k plane
 1.00  0.00  0.00   ! The first vector to define 3d k space plane
 0.00  1.00  0.00   ! The second vector to define 3d k space plane
 0.00  0.00  1.00   ! The third vector to define 3d k cube
```

KCUBE_BULK CARD : The same set as KPLANE_BULK CARD. We add another k vector to construct a k cube. Set this for **BulkGap_cube_calc=T.** The values list above are default values.

3. Features

At present, We have 11 subroutines supported for your research. We listed them in Table I. Here we give some details one by one.

1. **Bulkband calculation**
   you should specify the number of k lines and the special lines **KPATH_BULK CARDS.** The outputs for this function are bulkek.dat and bulkek.gnu. You can run "gnuplot bulkek.gnu", or "xmgrace bulkek.dat" to get bandstucture plot.

2. **BulkFS calculation**
   Bulk Fermi surface calculation. you should specify the number of k points for each three reciprocal vectors **Nk1, Nk2, Nk3 in NAMELISTS PARAMETERS.** The outputs for this function are FS3D.bxsf. You can run “xcrysden —bxsf FS3D.bxsf” to get the plot. By the way, Bulk band and BulkFS calculations are already implemented in Wannier90 code.
3. **Energy gap** calculations
   We support two modes for energy gap calculations. A. We can plot the energy gap as a function of k in a k plane, which can be done by setting **BulkGap_plane_calc=T**. The k plane can be set in **KPLANE_BULK.** The number of k points for two vectors is **Nk1, Nk2,** set in **NAMELISTS PARAMETERS.** You can get gap

plot GapPlane.png with “gnuplot GapPlane.gnu”. B. We also can get energy gap in a 3D k cube defined by three vectors with a start point, which are set in CARD **KCUBE_BULK.** The number of k points for three vectors is **Nk1, Nk2, Nk3,** set in **NAMELISTS PARAMETERS.** This helps us to find Weyl points in the 3D BZ. For 3D GapCube calculation, the output is the k points at which the energy gap is small than Gap_threshold. You can list all the k points satisfied such criterion by “gnuplot GapCube.gnu”.

4. **Slab band calculation**
   Before using iterative green’s function to get the surface state spectrum for semi-infinite system. We also can just construct a finite thickness slab system and calculate the band structure for it. Set **SlabBand_calc=T** in **NAMELISTS CONTROL,** and set **Nk1** in **NAMELISTS PARAMETERS**, set k path in **KPATH_SLAB CARD**. And get plot slabek.png with “gnuplot slabek.gnu”.
5. **Nanowire/nanoribbon band calculation**
   Band calculation for wire system. Set **WireBand_calc=T** in **NAMELISTS CONTROL,** and set **Nk1** in **NAMELISTS PARAMETERS.** Get plot ribbonek.png with “gnuplot ribbonek.gnu"
6. **Surface state calculation**
   Surface state spectrum at each k point and each energy level. Set SlabSS_calc=T, and set **Nk1** in **NAMELISTS PARAMETERS**, set k path in **KPATH_SLAB CARD**. Get the plots with "gnuplot surfdos_l.gnu”, "gnuplot surfdos_r.gnu”.
7. **Fermi arc calculation**
   Surface state spectrum at fixed energy level **E_arc** set in **NAMELISTS PARAMETERS** . Set SlabArc_calc=T, and set **Nk1**, **Nk2**, in **NAMELISTS PARAMETERS**, set k plane in **KPLANE_SLAB CARD**. Get the plots with "gnuplot arc_l.gnu”, "gnuplot arc_r.gnu”.
8. **Spin texture calculation**
   Spin texture calculation at fixed energy level **E_arc** set in **NAMELISTS PARAMETERS** . Set SlabArc_calc=T, and set **Nk1**, **Nk2**, in **NAMELISTS PARAMETERS**, set k plane

in **KPLANE_SLAB CARD**. Get the plots with "gnuplot spintext_l.gnu", "gnuplot spintext_r.gnu".

9. **Berry curvature calculation**
   Calculate Berry curvature at a fixed k plane in 3D BZ. Set BerryCurvature_calc=T, and set **Nk1**, **Nk2**, in **NAMELISTS PARAMETERS**, set k plane in **KPLANE_BULK CARD**. Get the plot with "gnuplot Berrycurvature.gnu".
10. **Wannier charge center/Wilson loop calculation**
    Wannier charge center, which is sometimes called Wilson loop can be calculated by set **WannierCenter_calc=T** and set **KPLANE_BULK CARD,** set number of k points for two vectors is **Nk1, Nk2** in **NAMELISTS PARAMETERS.** Notice: You should notice that the first vector in **KPLANE_BULK CARD** is the integration direction, this vector should be equal to one primitive reciprocal lattice vector. If you want to calculate the Z2 number, Please set the second vector to be half of the reciprocal lattice vector. You can get the Wannier charge center along the second k line. See more details In the paper written by Alexey. Soluyanov (2011).

4. Usage and examples (Bi2Se3)

Please look for more information on the github wiki https://github.com/quanshengwu/wannier_tools/wiki. Here we give an example of Bi2Se3, which is a topological insulator predicted in 2009 (Haijun Zhang, 2009). The input.dat shown above is just for this example. First we need to construct Wannier based tight binding Hamiltonian use Wannier90. Here we constructed one.

The wannier90_hr.dat is compressed as Bi2Se3_hr.tar.gz in examples/Bi2Se3/ folder. You can decompressed by a command "tar xzvf Bi2Se3_hr.tar.gz". The input.dat file can also be found in the same folder.

Once you have wannier90_hr.dat and input.dat, copy them to the same directory. Then just run like this

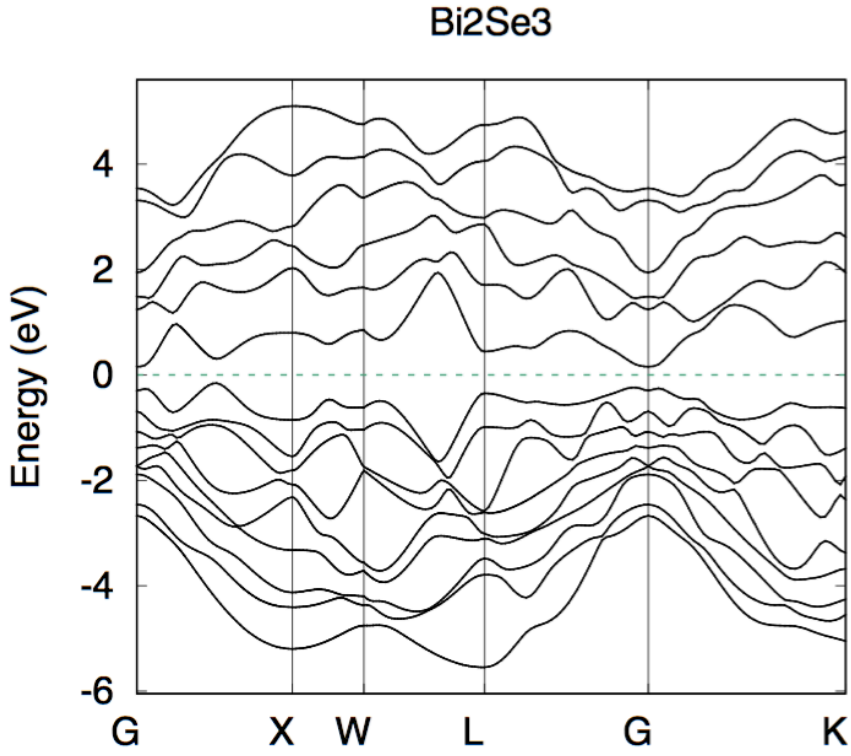
```
$ wann_tools &
```

Or run it in multi-cores

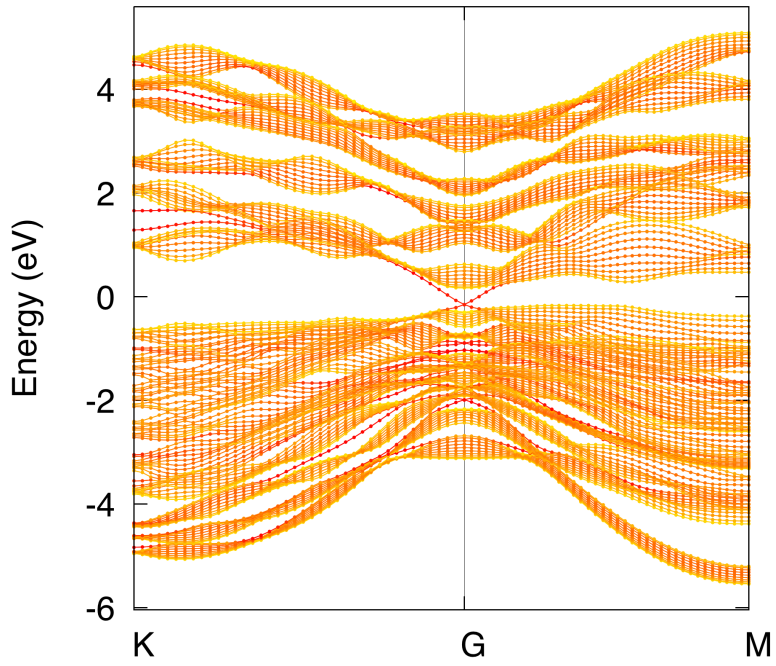
```
$ mpirun -np 4 wann_tools &
```

One important change, the information during the running is written into WT.out. You can check the content of it to get the status.

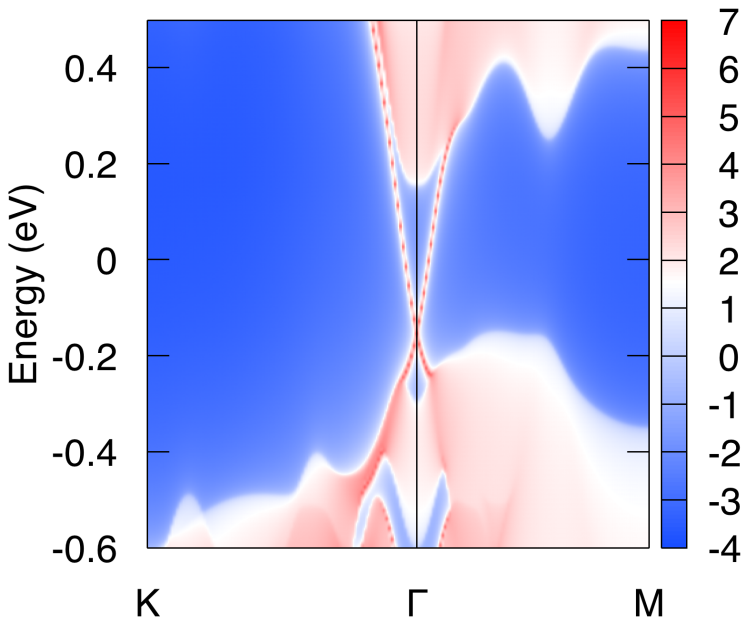
(1). Bulk band structure



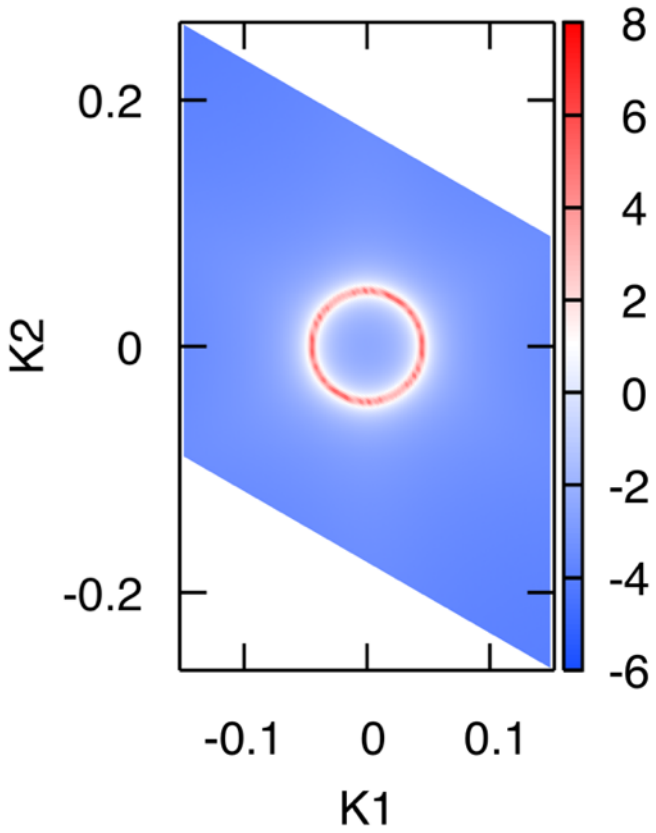
(2). (001) oriented slabs.



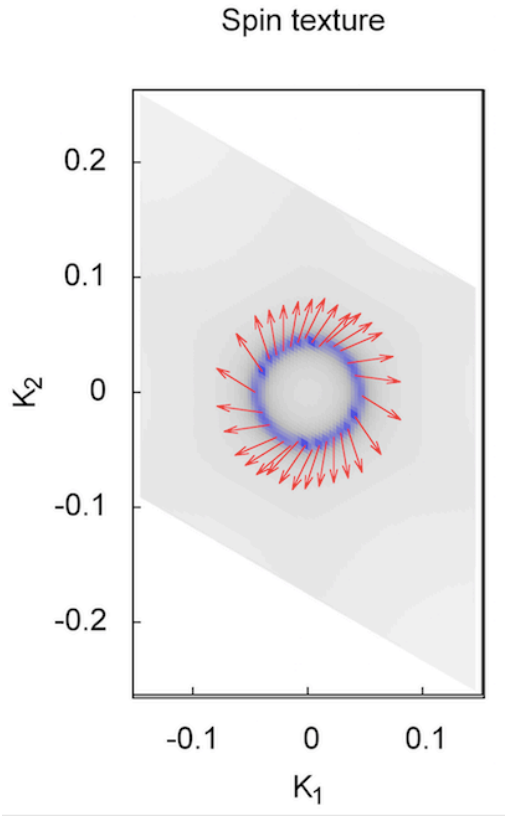
(3). (001) surface state



(4). (001) surface spectrum for a fixed energy E=0eV



(5). (001) spin texture for a fixed energy E=0eV



(6). The wannier charge center can be found in the wiki.

5. License and agreement.

If you use our code, please cite this website “Q.S Wu S.N Zhang, https://github.com/quanshengwu/wannier_tools" . If you have good ideas to improve this code, do not hesitate to contact me. Your contribution will be recorded.

6. Enjoy and good luck.