WannierTools Documentation

Release 2.2

QuanSheng Wu

Mar 24, 2017

CONTENTS

WannierTools: an open-source software package for novel topological materials Source on Github : https://github.com/quanshengwu/wannier_tools

Aiming to investigate topoligical properties of

- Electron systems (Tested)
- Phonon systems (Testing)
- Phonton systems (Developing)

CHAPTER

ONE

PRELIMINARIES

1.1 Installation of WannierTools (Linux or Mac)

1.1.1 Prerequisites

You need to install the following (mandatory) packages:

- Fortran compiler (Gfortran or ifort)
- MPICH version higher than 2.1.5
- Lapack and Blas library

1.1.2 Compilation

First Check out the repository by

```
git clone https://github.com/quanshengwu/wannier_tools.git
```

Or download the .zip file directly from https://github.com/quanshengwu/wannier_tools, then uncompress it

Then Go into wannier_tools/soc directory, Choose and Edit Makefile, Change the blas library " libs= " to your lapack+blas library

At present, we prepared 3 typical Makefiles, which are squential+gfotran, sequential+ifort and mpi+ifort.

For the mpi compiler, you should switch on the compile flag "-DMPI", see Makefile.intel-mpi

After the compliation, the binary 'wann_tools' is copied to wannier_tools/bin/, you can put this path to the system PATH with

```
export PATH=/where/you/downloaded/wannier_tools/bin:$PATH
```

to the `.bashrc` file in your home directory.

1.1.3 Usage

Now you can enjoy your exploration for topological materials with WannierTools.

There are two files you have to prepare,

1. wt.in. All the control and user specified parameters are inclued in this file.
2. wannier90_hr.dat. Tight binding model constructed by Wannier90 or written in the format as wannier90_hr.dat.

After the preparation of these two files, you can just run wann_tools in the same folder

```
wt.x &
```

or in multi-cores

```
mpirun -np 4 wt.x &
```

The output information during the running are written in WT.out.

1.1.4 Plotting tools

1. gnuplot
2. xmgrace
3. xcrysden
4. matlab

1.2 Introduction of input files

Attention: From WannierTools 2.2, the name of input file changes from 'input.dat' to **'wt.in'**. The excutable binary changes from 'wann_tools' to **'wt.x'**

There are two input files you should prepare *wt.in* and *wannier90_hr.dat*

1.2.1 wt.in

- *TB_FILE*
- *SYSTEM*
- *CONTROL*
- *PARAMETERS*
- *LATTICE*
- *ATOM_POSITIONS*
- *PROJECTORS*
- *SURFACE*
- *KPATH_BULK*
- *KPATH_SLAB*
- *KPLANE_SLAB*
- *KPLANE_BULK*
- *KCUBE_BULK*
- *EFFECTIVE_MASS*
- *WANNIER_CENTRES*

Before executing wann_tools, you should cp the wt.in file in the directory wannier_tools/example by your own necessary.

For version later than 2.0, we updated the format of wt.in. The input file is structured in a number of **NAMELIST** and **INPUT_CARDS**.

Here we introduce the wt.in for Bi2Se3 as an example

```
&TB_FILE
Hrfile = 'wannier90_hr.dat'
Package = 'VASP'             ! obtained from VASP, it could be 'VASP', 'QE', 'Wien2k',
↪ 'OpenMx'
/

&CONTROL
BulkBand_calc         = T
BulkFS_calc           = T
BulkGap_cube_calc     = T
BulkGap_plane_calc    = T
SlabBand_calc         = T
WireBand_calc         = T
SlabSS_calc           = T
SlabArc_calc          = T
SlabQPI_calc          = T
SlabSpintexture_calc  = T
Wanniercenter_calc    = T
BerryCurvature_calc   = T
EffectiveMass_calc    = T
/

&SYSTEM
NSLAB = 10               ! for thin film system
NSLAB1= 4                ! nanowire system
NSLAB2= 4                ! nanowire system
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
Eta_Arc = 0.001     ! infinite small value, like brodening
E_arc = 0.0         ! energy for calculate Fermi Arc
OmegaNum = 100      ! omega number
OmegaMin = -0.6     ! energy interval
OmegaMax =  0.5     ! energy interval
Nk1 = 21           ! number k points  odd number would be better
Nk2 = 21           ! number k points  odd number would be better
Nk3 = 21           ! number k points  odd number would be better
NP = 1              ! 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
-0.00 -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


EFFECTIVE_MASS      ! optional
2                   ! The i'th band to be calculated
0.01                ! k step in unit of (1/Angstrom)
0.0 0.0 0.0         ! k point where the effective mass calculated.
```

```
WANNIER_CENTRES     ! copy from wannier90.wout
Cartesian
 -0.000040  -1.194745   6.638646
  0.000038  -1.196699   6.640059
 -0.000032  -1.192363   6.640243
 -0.000086  -3.583414   2.908040
  0.000047  -3.581457   2.906587
 -0.000033  -3.585864   2.906443
 -0.000001   1.194527   4.773338
  0.000003   1.194538   4.773336
 -0.000037   1.194536   4.773327
  0.000006  -1.194384   1.130261
 -0.000018  -1.216986   1.140267
  0.000007  -1.172216   1.140684
  0.000011  -3.583770   8.416406
 -0.000002  -3.561169   8.406398
 -0.000007  -3.605960   8.405979
  0.000086  -1.194737   6.638626
 -0.000047  -1.196693   6.640080
  0.000033  -1.192286   6.640223
  0.000040  -3.583406   2.908021
 -0.000038  -3.581452   2.906608
  0.000032  -3.585788   2.906424
  0.000001   1.194548   4.773330
 -0.000003   1.194537   4.773332
  0.000037   1.194539   4.773340
 -0.000011  -1.194381   1.130260
  0.000002  -1.216981   1.140268
  0.000007  -1.172191   1.140687
 -0.000006  -3.583766   8.416405
  0.000018  -3.561165   8.406400
 -0.000007  -3.605935   8.405982
```

NAMELISTS

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. Such as

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

There are 4 **NAMELISTS** included in wt.in. They are *TB_FILE*, **SYSTEM**, **CONTROL**, **PARAMETERS**.

Note: If you want to comment one line, please use '!' instead of '#', because our codes were written in Fortran.

TB_FILE

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

```
&TB_FILE
Hrfile = 'wannier90_hr.dat'
Package = 'VASP'             ! obtained from VASP, it could be 'VASP', 'QE', 'Wien2k',
↪ 'OpenMx'
/
```

The default value for Hrfile is 'wannier90_hr.dat'. You could specify the first-principle package that used for obtaining wannier90_hr.dat. Default value for Package is 'VASP'. We support VASP, QE, Wien2k, OpenMx, Abinit at present. Please report new software package to me if you needed.

Note: Package is very important if you use QE to generate your tight binding model. Because the orbital order in QE is different from VASP, Wien2k et al.. And it will affect the results of spin texture. If you got strange spin texture, please carefully check this tag.

SYSTEM

In this namelists, we specify the system you need to compute.

```
&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
/
```

- 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 Wannier 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 : real-valued, 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.

CONTROL

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.

```
&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
/
```

Note: New features : :red: FindNodes_calc; WeylChirality_calc; Z2_3D_calc; Chern_3D_calc

We listed those features in the table below.

Flag options	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
BulkGap_cube_calc	Energy gap for a 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
FindNodes_calc	Find touching points between the Numoccpuied'th band and (Numoccpuied+1)'th band	Nodes.dat, Nodes.gnu	yes
SlabBand_calc	Band structure for 2D slab system	slabek.dat,slabek.gnu	yes
WireBand_calc	Band structure for 1D ribbon system	ribbonek.dat,ribbonek.gnu	yes
Dos_calc	Density of state for 3D bulk system	dos.dat	yes
JDos_calc	Joint Density of state for 3D bulk system	jdos.dat	yes
SlabSS_calc	Surface spectrum A(k,E) along a kline and energy interval for slab system	dos.dat_l, dos.dat_r, dos.dat_bulk,surfdos_l.gnu, surfdos_r.gnu, surfdos_l_only.gnu, surfdos_r_only.gnu, surfdos_bulk.gnu	yes
SlabArc_calc	Surface spectrum A(k,E0) for fixed energy E0 in 2D k-plane for slab system	arc.dat_l, arc.dat_r, arc_l.gnu, arc_r.gnu, arc_l_only.gnu, arc_l_only.gnu,	yes
SlabQPI_calc	Surface QPI for fixed energy E0 in 2D k-plane for slab system	arc.dat_l, arc.dat_r, arc_l.gnu, arc_r.gnu, arc_l_only.gnu, arc_l_only.gnu, arc.jdat_l, arc.jdat_r, arc.jsdat_l, arc.jsdat_r, arc_l_jdos.gnu, arc_l_jsdos.gnu, arc_r_jdos.gnu, arc_r_jsdos.gnu,	yes
SlabSpintexture_calc	Spin texture in 2D k-plane for slab system	spindos_l.dat spindos_r.dat spintext_l.gnu spintext_r.gnu spintext_l.dat spintext_r.dat	yes
wanniercenter_calc	Wilson loop of a given 3D k-plane for bulk system	wcc.dat, wcc.gnu	yes
Z2_3D_calc	Wilson loop in all 6 3D k-planes for bulk system Z2 number calculation	wanniercenter3D_Z2.gnu, wanniercenter3D_Z2_{1-6}.dat	yes
Chern_3D_calc	Wilson loop in all 6 3D k-planes for bulk system Chern number calculation	wanniercenter3D_Z2.gnu, wanniercenter3D_Z2_{1-6}.dat	yes
WeylChirality_calc	Weyl Chirality calculation for given k points	find chiralities in WT.out, wanniercenter3D_Weyl.dat, wanniercenter3D_Weyl_*.gnu	yes
BerryPhase_calc	Berry phase with a 3D k path for bulk system	find Berry phase in WT.out	Yes
BerryCurvature_calc	Berry Curvature in 3D k-plane for bulk system	BerryCurvature.dat, BerryCurvature.gnu	yes

PARAMETERS

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

```
&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
/
```

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

The second important format in wt.in 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. There are several INPUT_CARDS in the wt.in. 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

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.

```
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

In this card, we set the atom's position.

```
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
```

Note: 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

In this card, we set the Wannier projectors for the tight binding.

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

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. The default order in Wannier90 is "s", "pz", "px", "py", "dz2", "dxz", "dyz", "dx2-y2", "dxy". You can find the orbital order from wannier90.wout.

Note: If you don't care about the calculation related to symmetry like mirror chern number. The order or the name is not important. So for the f electrons, please write 7 random orbitals like px or dz2 or what else you want.

SURFACE

This card is very important for slabs and ribbon calculation. You need to read the following text carefully

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

In this card, we specify the surface you want to investigate. Basically, you should be aware of which surface you want to investigate, and of which direction you want to study the ribbon. So we need to define the new lattice vector system like this,

1. choose two vectors on the surface we want to study, and choose another vector which is not on this plane.

2. The volume of the new cell constructed by those 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

$R'_1 = a_{11}R_1 + a_{12}R_2 + a_{13}R_3$

$R'_2 = a_{21}R_1 + a_{22}R_2 + a_{23}R_3$

$R'_3 = a_{31}R_1 + a_{32}R_2 + a_{33}R_3$

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.

Note: a11, a12, a13 ..., a33 should be integers.

KPATH_BULK

This is the k path for bulk band structure calculation.

```
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
```

These k points are in unit of the reciprocal lattice constant 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. So you must set some value in the input file when choosing BulkBand_calc=T.

KPATH_SLAB

This is the k path for slab system.

```
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
```

including the band structure calculation and the surface state calculation. It is necessary to set it when SlabBand_calc=T or SlabSS_calc=T. Number of k points along the line is Nk1.

KPLANE_SLAB

Define a 2D k space plane for arc plots.

```
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
```

The first line is the start point of the plane. The second and third line are the two vectors defining 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

The same set as KPLANE_SLAB CARD, but for 3D case.

```
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
```

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

The same set as KPLANE_BULK CARD

```
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
```

We add another k vector to construct a k cube. Set this for BulkGap_cube_calc=T. The values list above are default values.

EFFECTIVE_MASS

This card is set for effective mass calculation

```
EFFECTIVE_MASS      ! optional
2                   ! The i'th band to be calculated
0.01                ! k step in unit of (1/Angstrom)
0.0 0.0 0.0         ! k point where the effective mass calculated.
```

WANNIER_CENTRES

This card will be usefull for Wilson loop calculations.

```
WANNIER_CENTRES     ! copy from wannier90.wout
Cartesian
 -0.000040  -1.194745   6.638646
  0.000038  -1.196699   6.640059
 -0.000032  -1.192363   6.640243
 -0.000086  -3.583414   2.908040
  0.000047  -3.581457   2.906587
 -0.000033  -3.585864   2.906443
 -0.000001   1.194527   4.773338
  0.000003   1.194538   4.773336
 -0.000037   1.194536   4.773327
  0.000006  -1.194384   1.130261
 -0.000018  -1.216986   1.140267
  0.000007  -1.172216   1.140684
  0.000011  -3.583770   8.416406
 -0.000002  -3.561169   8.406398
 -0.000007  -3.605960   8.405979
  0.000086  -1.194737   6.638626
 -0.000047  -1.196693   6.640080
  0.000033  -1.192286   6.640223
  0.000040  -3.583406   2.908021
 -0.000038  -3.581452   2.906608
  0.000032  -3.585788   2.906424
  0.000001   1.194548   4.773330
 -0.000003   1.194537   4.773332
  0.000037   1.194539   4.773340
 -0.000011  -1.194381   1.130260
  0.000002  -1.216981   1.140268
  0.000007  -1.172191   1.140687
 -0.000006  -3.583766   8.416405
  0.000018  -3.561165   8.406400
 -0.000007  -3.605935   8.405982
```

Those centres can be obtained from wannier90.wout by searching "Final state". The default values for this card are atomic positions.

1.2.2 wannier90_hr.dat

This file contains the TB parameters. Usually, it can be generated by Wannier90.

Of cource, you can generate it from the Slater-Koster method or discretize k.p model onto a cubic lattice. The format should like this

```
written on  8May2016 at 13:57:00
         30
        547
   2    2    1    1    1    1    1    1    1    1    2    2    2    2    2
   1    1    1    2    1    1    1    2    1    1    1    1    1    1    1
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    4
   2    2    2    2    2    2    2    4    1    1    1    1    1    1    1
```

```
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    2    1
  1    1    1    1    1    1    2    1    1    1    1    1    1    1    1
  2    2    2    2    2    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    2    1    1    1    1
  1    1    1    2    1    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    2    2    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    2    1    1    1
  1    1    1    1    2    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  1    2    1    1    1    2    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    2
  1    1    1    1    1    1    1    2    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  1    2    1    1    1    2    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  1    1    2    1    1    1    1    1    1    1    2    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    2    2    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  1    1    1    2    1    1    1    1    1    1    1    2    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  1    1    2    2    2    2    2    1    1    1    1    1    1    1    1
  2    1    1    1    1    1    1    1    2    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    4
  2    2    2    2    2    2    2    4    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1    1    1    2
  1    1    1    2    1    1    1    2    2    2    2    2    1    1    1
  1    1    1    1    1    2    2
 -6    2   -3    1    1   -0.000002    0.000003
 -6    2   -3    2    1    0.000002    0.000017
 -6    2   -3    3    1   -0.000053    0.000002
 -6    2   -3    4    1   -0.000031    0.000002
 -6    2   -3    5    1    0.000001   -0.000000
 -6    2   -3    6    1   -0.000003    0.000002
 -6    2   -3    7    1    0.000037   -0.000001
 -6    2   -3    8    1   -0.000001   -0.000003
 -6    2   -3    9    1   -0.000005   -0.000003
 -6    2   -3   10    1   -0.000062   -0.000001
 -6    2   -3   11    1   -0.000001    0.000001
 -6    2   -3   12    1   -0.000031    0.000002
 -6    2   -3   13    1    0.000011   -0.000000
 -6    2   -3   14    1   -0.000001    0.000001
 -6    2   -3   15    1    0.000003    0.000003
 -6    2   -3   16    1    0.000000   -0.000010
 -6    2   -3   17    1   -0.000010   -0.000001
 -6    2   -3   18    1   -0.000000   -0.000008
 -6    2   -3   19    1    0.000000    0.000000
 -6    2   -3   20    1    0.000012   -0.000002
......
```

1. The 1st line is a comment line with any content.
2. The 2nd line is the number of Wannier orbitals, in consideration of spin degeneracy. We call it NUM_WANNS

3. The 3rd line is the number of **R** lattice vectors, we call it NRPTS
4. This section is about the degeneracy of R points. If you generate wannier90_hr.dat by you self, please set it to 1. There are NRPTS number of 1.
5. This section gives the TB parameters. The first three integers are the coordinates or R vectors in unit of three lattice vectors. The 4th and 5th column are the band index (Row first). The 6th and 7th are complex entities of the Hamiltonian.

1.3 Capabilities of WannierTools

- *Bulk band calculation*
- *BulkFS calculation*
- *Find Nodes calculation*
- *Energy gap calculations*
- *Slab band calculation*
- *Nanowire/nanoribbon band calculation*
- *Surface state ARPES calculation*
- *Surface state QPI calculation*
- *Fermi arc calculation*
- *Spin texture calculation*
- *Berry phase calculation*
- *Berry curvature calculation*
- *Wannier charge center/Wilson loop calculation*
- *Z2 number for 3D bulk materials*
- *Weyl Chirality calculation*

1.3.1 Bulk band calculation

Calculate bulk energy band for a series k lines. This is the basic calculation after the construction of Wannier functions. You have to compare your Wannier interpolated bands with the DFT bands. Those two bands should match well around the Fermi level.

Input

Typical flags for bulk band calculation in the wt.in.

```
&CONTROL
BulkBand_calc = T
/
&PARAMETERS
Nk1 = 101   ! Number of k points for each k line
/

KPATH_BULK     ! k point path
```

```
4              ! number of k lines 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
```

See *CONTROL*, *PARAMETERS*, *KPATH_BULK*

Output

The outputs for bulk band calculation are **bulkek.dat** and **bulkek.gnu**. You can get the band plot by running

```
gnuplot bulkek.gnu
```

or

```
xmgrace bulkek.dat
```

to get bandstucture plot.

The data structure for **bulkek.dat**

```
0.000000000       -2.673821992  119   80   80  119   80   80  205  138  138   70   40␣
↪  40   70   40   40    0    0    0    0    0    0    0    0    0    0    0    0    ␣
↪0    0    0
0.016453872       -2.681536808  118   78   78  118   78   78  203  134  134   82   41␣
↪  41   82   41   41    0    0    0    0    0    0    0    0    0    0    0    0    ␣
↪0    0    0
```

1. The 1st column represents k points for the given kpath (KPATH_BULK)
2. The 2nd column is the energy level
3. From the 3rd to the n'th column are the projected weight of the wave function at each k point and each band onto each wannier orbitals. Those weights are normalized to 255 for the color plot convinence.

The subrotine for this feature is **:f:func:'ek_bulk'**

1.3.2 BulkFS calculation

Bulk Fermi surface calculation.

Input

You should specify the number of k points for each three reciprocal vectors Nk1, Nk2, Nk3 in NAMELISTS PARAMETERS

```
&CONTROL
BulkFS_calc = T
/
&PARAMETERS
Nk1 = 101    ! No. of slices for the 1st reciprocal vector
Nk2 = 101    ! No. of slices for the 2nd reciprocal vector
Nk3 = 101    ! No. of slices for the 3rd reciprocal vector
/
```

See *CONTROL*, *PARAMETERS*

Output

The outputs for this function are **FS3D.bxsf**. You can plot the FS with xcrysden run

```
xcrysden --bxsf FS3D.bxsf
```

to get the plot.

By the way, Bulk band and BulkFS calculations were already implemented in Wannier90 code.

1.3.3 Energy gap calculations

We support two modes for energy gap calculations.The formula is $gap(k) = E_{NumOccupied+1}(k) - E_{NumOccpuied}(k)$

Gap_plane mode

Calculate the energy gap for the k points in the KPLANE_BULK. This is useful to show Weyl points and nodal line structure.

Input

Typical input parameters for BulkGap_plane calculation

```
&CONTROL
BulkGap_Plane_calc = T
/
&PARAMETERS
Nk1 = 101   ! No. of slices for the 1st reciprocal vector
Nk2 = 101   ! No. of slices for the 2nd reciprocal vector
/

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
```

See *CONTROL*, *PARAMETERS*, *KPATH_BULK*

Output

The outputs for Gap_plane mode are **GapPlane.dat**, **GapPlane.gnu**. The **GapPlane.png** will get by

```
gnuplot GapPlane.gnu
```

The head of **GapPlane.dat**

```
kx              ky              kz              gap             Ev4             Ev3
↪         Ev2             Ev1             Ec1             Ec2             Ec3
↪     Ec4             k1              k2              k3
0.00000000      0.00000000      0.00000000      0.45569845     -0.69109275     -0.
↪69109055     -0.29654328     -0.29654073      0.15915772      0.15915871      1.
↪24348171      1.24348457      0.00000000      0.00000000      0.00000000
0.03796028     -0.02191637      0.00548462      0.43770730     -0.77636510     -0.
↪77598312     -0.26035113     -0.26027881      0.17742849      0.17771545      1.
↪29499437      1.29505298      0.00000000      0.02500000      0.00000000
```

- Column 1-3 are the Cartesian coordinates of the k points in the KPLANE_BULK, in unit of $\frac{1}{Angstrom}$
- Column 4 is the energy gap
- Column 5-12 are the energy value for valence and conduction bands close to the Fermi level. There are 4 conduction bands and 4 valence bands.
- Column 13-15 are the Direct coordinates of the k points in the KPLANE_BULK

Gap_Cube mode

This helps us to find Weyl points and nodal line structure in the 3D BZ.

Input

Typical input parameters for BulkGap_cube calculation

```
&CONTROL
BulkGap_Cube_calc = T
/
&PARAMETERS
Nk1 = 101    ! No. of slices for the 1st reciprocal vector
Nk2 = 101    ! No. of slices for the 2nd reciprocal vector
Nk3 = 101    ! No. of slices for the 3rd reciprocal vector
/

KCUBE_BULK
-0.50 -0.50 -0.50   ! Original point for 3D k plane
 1.00  0.00  0.00   ! The 1st vector to define 3d k cube
 0.00  1.00  0.00   ! The 2nd vector to define 3d k cube
 0.00  0.00  1.00   ! The 3rd vector to define 3d k cube
```

See *CONTROL*, *PARAMETERS*, kcubebulk

Output

The outputs for Gap_plane mode are **GapCube.dat**, **GapCube.gnu**. The **GapCube.png** will get by

```
gnuplot GapCube.gnu
```

The head of **GapCube.dat** are

```
kx (1/A)         ky (1/A)         kz (1/A)       Energy gap              Ev
↪Ec       k1 (2pi/a)       k2 (2pi/b)       k3 (2pi/c)
0.00000000       0.87665487      -0.54846229       0.79075142      -0.34827281       0.
↪44247861      -0.50000000      -0.50000000      -0.50000000
0.00000000       0.87665487      -0.51555455       0.86792416      -0.38635069       0.
↪48157346      -0.50000000      -0.50000000      -0.45000000
```

- Column 1-3 are the Cartesian coordinates of the k points where energy gap is small than Gap_threshold, in unit of $\frac{1}{Angstrom}$
- Column 4 is the energy gap. Those values are smaller than Gap_threshold, see *PARAMETERS*
- Column 5-6 are the energy value for valence and conduction bands close to the Fermi level. There are 4 conduction bands and 4 valence bands.
- Column 7-9 are the Direct coordinates of the k points.

1.3.4 Find Nodes calculation

Beside by using GapCube and GapPlane to find Weyl/Dirac nodes or node lines, we can directly using FindNodes function. $gap(k) = E_{NumOccupied+1}(k) - E_{NumOccpuied}(k)$

Input

Typical input parameters for FindNodes_cube calculation

```
&CONTROL
FindNodes_calc = T
/
&PARAMETERS
Nk1 = 8    ! No. of slices for the 1st reciprocal vector
Nk2 = 8    ! No. of slices for the 2nd reciprocal vector
Nk3 = 8    ! No. of slices for the 3rd reciprocal vector
Gap_threshold = 0.0001 ! a value to determine which point should be identified as a
↪node
/

KCUBE_BULK
-0.50 -0.50 -0.50   ! Original point for 3D k plane
 1.00  0.00  0.00   ! The 1st vector to define 3d k cube
 0.00  1.00  0.00   ! The 2nd vector to define 3d k cube
 0.00  0.00  1.00   ! The 3rd vector to define 3d k cube
```

Note: Please don't set Nk1, Nk2, Nk3 too large. Otherwise, it will become very time consuming. Usually, 15*15*15 is enough to get converged number of Weyl/Dirac points.

Output

Outputs are **Nodes.dat** and **Nodes.gnu**. Nodes.png will be obtained by

```
gnuplot Nodes.gnu
```

Here are heads of output for WTe2 **Nodes.dat**

```
# local minimal position and the related energy gap
#       kx          ky          kz          gap           E          k1          k2
↪      k3
    0.219436   -0.045611   -0.000001    0.000000    0.056688    0.121432   -0.045363
↪ -0.000003
   -0.219515   -0.045063   -0.000001    0.000000    0.056461   -0.121476   -0.044818
↪ -0.000002
    0.220195   -0.038682   -0.000002    0.000000    0.051264    0.121852   -0.038472
↪ -0.000003
   -0.220183   -0.038936   -0.000001    0.000000    0.051618   -0.121845   -0.038724
↪ -0.000003
    0.219514    0.045063    0.000001    0.000000    0.056459    0.121475    0.044818
↪  0.000003
   -0.219434    0.045620    0.000002    0.000000    0.056692   -0.121431    0.045371
↪  0.000004
   -0.220194    0.038678    0.000000    0.000000    0.051259   -0.121851    0.038468
↪  0.000001
    0.220181    0.038941    0.000000    0.000000    0.051620    0.121844    0.038729
↪  0.000001
```

You will find that there are 8 Weyl points in the BZ as expected.

1.3.5 Weyl Chirality calculation

After you identify the positions of Weyl points, you could use this function to calculate the chirality, which tells you whether a Weyl point is a sink or a source of the Berry Curvature.

Input

Typical input parameters for WeylChirality_calc calculation

```
&CONTROL
WeylChirality_calc = T
/
&PARAMETERS
Nk1 = 41   ! No. of slices for the 1st reciprocal vector, berry phase integration
 ↪direction
Nk2 = 21   ! No. of slices for the 2nd reciprocal vector
/

WEYL_CHIRALITY
8            ! Num_Weyls
Cartesian    ! Direct or Cartesian coordinate
0.004        ! Radius of the ball surround a Weyl point
 0.219436   -0.045611   -0.000000    ! Positions of Weyl points, No. of lines should
↪larger than Num_weyls
-0.219515   -0.045063   -0.000000
 0.220195   -0.038682   -0.000000
-0.220183   -0.038936   -0.000000
 0.219514    0.045063    0.000000
-0.219434    0.045620    0.000000
-0.220194    0.038678    0.000000
 0.220181    0.038941    0.000000
```

Output

Outputs are **wanniercenter3D_Weyl.dat** and **wanniercenter3D_Weyl_i.gnu**. wanniercenter3D_Weyl.png will be obtained by

```
gnuplot wanniercenter3D_Weyl_i.gnu

for ((i=1; i<9; i++)); do gnuplot wanniercenter3D_Weyl_$i.gnu;done
```

Note: i is an integer from 1 to Num_weyls

Here are heads of output for WTe2 **wanniercenter3D_Weyl.dat**

```
# Chirality              -1              1              1              -1
↪    1             -1             1             -1
      # k            phase          phase          phase          phase
↪phase           phase          phase          phase
 0.00000000      0.99970932      0.00005854      0.00004671      0.99975139      0.
↪00005851      0.99970861      0.00004736      0.99975087
 0.05000000      0.89229069      0.08696587      0.08941971      0.90855415      0.
↪08723118      0.89170870      0.09022452      0.90795187
 0.10000000      0.79659821      0.16589558      0.17112299      0.82248889      0.
↪16697194      0.79511289      0.17279423      0.82108022
```

The first line shows the chiralities of each Weyl point. The first column is k point. From the 2nd to the last column show the Wannier charge center phase. In total, there are Num_weyls columns.

1.3.6 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.

Note: For slab calculations, please read carefully the input card *SURFACE*

Input

```
&CONTROL
SlabBand_calc = T
/
&SYSTEM
NSLAB = 10
/
&PARAMETERS
Nk1 = 101   ! No. of slices for the 1st reciprocal vector
/
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
```

See *CONTROL*, *SYSTEM PARAMETERS*, *KPATH_SLAB*

Output

Outputs are **slabek.dat** and **slabek.gnu**

The heads of **slabek.dat** are

```
0.0000000     -4.9575466     240
0.0508687     -5.0110528     226
0.1017373     -5.0566963     221
0.1526060     -5.0671994     220
...
```

- The 1st column are k points in the KPATH_SLAB
- The 2nd column are energy values.
- The 3rd column represent the surface weight, which is normalized to 255.

The colorfull plot **slabek.png** of the slab energy bands can be obtained by

```
gnuplot slabek.gnu
```

1.3.7 Nanowire/nanoribbon band calculation

Band calculation for wire system. Only one direction is periodic, the other two directions are confined.

Input

You don't have to set the k path, because it only has one direction.

```
&CONTROL
WireBand_calc = T
/
&SYSTEM
NSLAB1 = 4
NSLAB2 = 4
/
&PARAMETERS
Nk1 = 101   ! No. of slices for the 1st reciprocal vector
/
```

See *CONTROL*, *SYSTEM PARAMETERS*

Output

Outputs are **ribbonek.dat** and **ribbonek.gnu**. The data format of **ribbonek.dat** is the same as **slabek.dat**. Get plot ribbonek.png with

```
gnuplot ribbonek.gnu
```

1.3.8 Surface state ARPES calculation

One important feature for topological materials is the surface state. The bulk-edge correspondence tells us, if the topological property of the bulk system is nontrivial, then there will be nontrivial states on the surface. Nowadays,

there are several method to detect the surface states. One direct way is the angle resolved photoemission spectroscopy (ARPES). Such spectrum can be obtained by the iterative Green's function.

Note: For slab calculations, please read carefully the input card *SURFACE*

Input

```
&CONTROL
SlabSS_calc = T
/
&PARAMETERS
OmegaNum = 101
OmegaMin = -1.0
OmegaMax =  1.0
Nk1 = 101   ! No. of slices for the 1st reciprocal vector
NP = 2      ! principle layer
/
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
```

See *CONTROL*, *PARAMETERS*, *KPATH_SLAB*

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.

1.3.9 Surface state QPI calculation

Settings for this feature are almost the same as *Fermi arc calculation*. Only difference is that you should set

```
&CONTROL
SlabQPI_calc          = T
/
```

Output

There are a lot of outputs for QPI calculation. including arc.dat_l, arc.dat_r, arc_l.gnu, arc_r.gnu, arc_l_only.gnu, arc_l_only.gnu, arc.jdat_l, arc.jdat_r, arc.jsdat_l, arc.jsdat_r, arc_l_jdos.gnu, arc_l_jsdos.gnu, arc_r_jdos.gnu, arc_r_jsdos.gnu.

The gnu script with 'only' means we only plot the spectrum with the surface contribution, exclude the bulk contribution.

1.3.10 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".

1.3.11 Spin texture calculation

Spin texture calculation at fixed energy level E_arc set in NAMELISTS PARAMETERS . Set Slabspintexture_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".

1.3.12 Berry phase calculation

Calculate Berry phase of a closed k path in 3D BZ. This is useful in a nodal line system. It is demonstrated that the Berry phase around a closed mirror symmetric k loop is either 0 or pi for a mirror protect nodal line system.

In WannierTools, you can specify a k path by a serials k points. Here we take the WC example, which has two nodal lines around K point.

Input

```
&CONTROL
BerryPhase_calc = T
/
&SYSTEM
NumOccupied = 10        ! Number of occupied Wannier orbitals
/
&PARAMETERS
Nk1 = 21    ! No. of slices for the 1st reciprocal vector
/

KPATH_BERRY
11
Direct
 0.3    0.333  -0.2
 0.3    0.333  -0.1
 0.3    0.333  -0.0
 0.3    0.333   0.1
 0.3    0.333   0.2
 0.33   0.333   0.2
 0.33   0.333   0.1
 0.33   0.333   0.0
 0.33   0.333  -0.1
 0.33   0.333  -0.2
 0.3    0.333  -0.2
```

Output

The value of Berry phase can be found in the **WT.out**.

Note: 1. In principlely, the Berry phase for around a nodal line should be interger. However, the MLWF-TB model usally is not symmetric. So the value of Berry phase is close to zero or one.

2. The first and the last kpoints in the KPATH_BERRY should be the same to form a loop.

1.3.13 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".

1.3.14 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).

1.3.15 Z2 number for 3D bulk materials

We can get Z2 topological index (v, v1v2v3) from the z2 calculations of six time invariant planes, i.e. k1=0.0; k1=0.5; k2=0.0; k2=0.5; k3=0.0; k3=0.5; Usually, you can call "Wannier charge center calculation for a plane" six times. Here we packed them up to get another function. You can set the input file like the following.

Input

The necessary tags that you should set in the wt.in

```
&CONTROL
Z2_3D_calc = T
/
&PARAMETERS
NumOccpuied = 18  ! No. of occupied wannier bands
Nk2 = 41    ! No. of slices of the k points for WCCs
/
```

Output

Outputs are **wanniercenter3D_Z2_1.dat, wanniercenter3D_Z2_2.dat, wanniercenter3D_Z2_3.dat, wanniercenter3D_Z2_4.dat, wanniercenter3D_Z2_5.dat, wanniercenter3D_Z2_6.dat** and **wanniercenter3D_Z2.gnu**. The z2 value can be found in the WT.out by searching "z2 number for 6 planes". The WCC (Wilson loop) plots **wanniercenter3D_Z2.eps** can be got with:

```
gnuplot wanniercenter3D_Z2.gnu
```

Note: Important: please set **NumOccpuied** correctly. It represents the occpuied wannier bands, not the total number of electrons.

CHAPTER

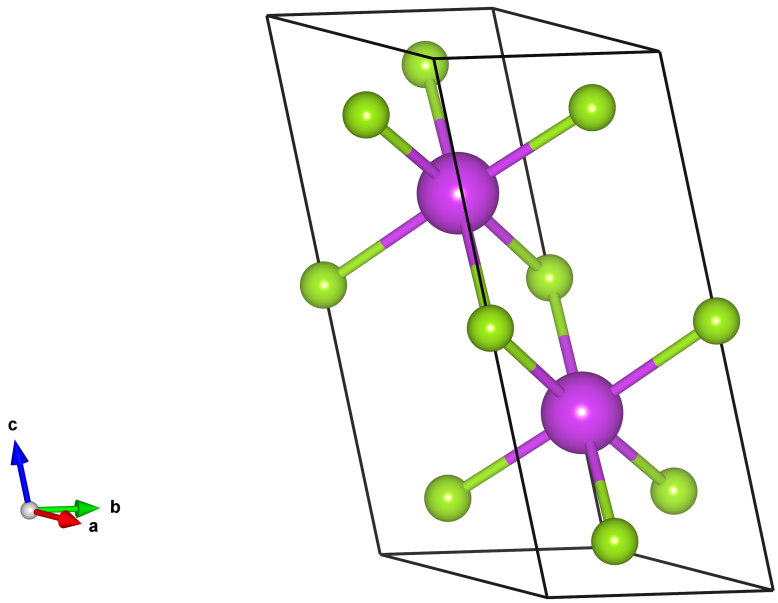
TWO

TUTORIAL: LEARNING WANNIERTOOLS THROUGH EXAMPLES

2.1 Bi2Se3 (3D strong TI)

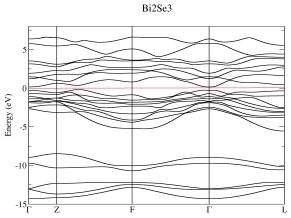
Bi2Se3 is a strong topological insulator. The Z2 topological index is (1, 000). Theoretical and experimental validation can be found in Nature Physics 5, 438-442 (2009) and Nature Physics 5, 398-402 (2009) respectively. Here we show you how to use WannierTools to study strong topological materials. The input file and some related files are included in each distribution.

Here is the primitive unit cell of Bi2Se3



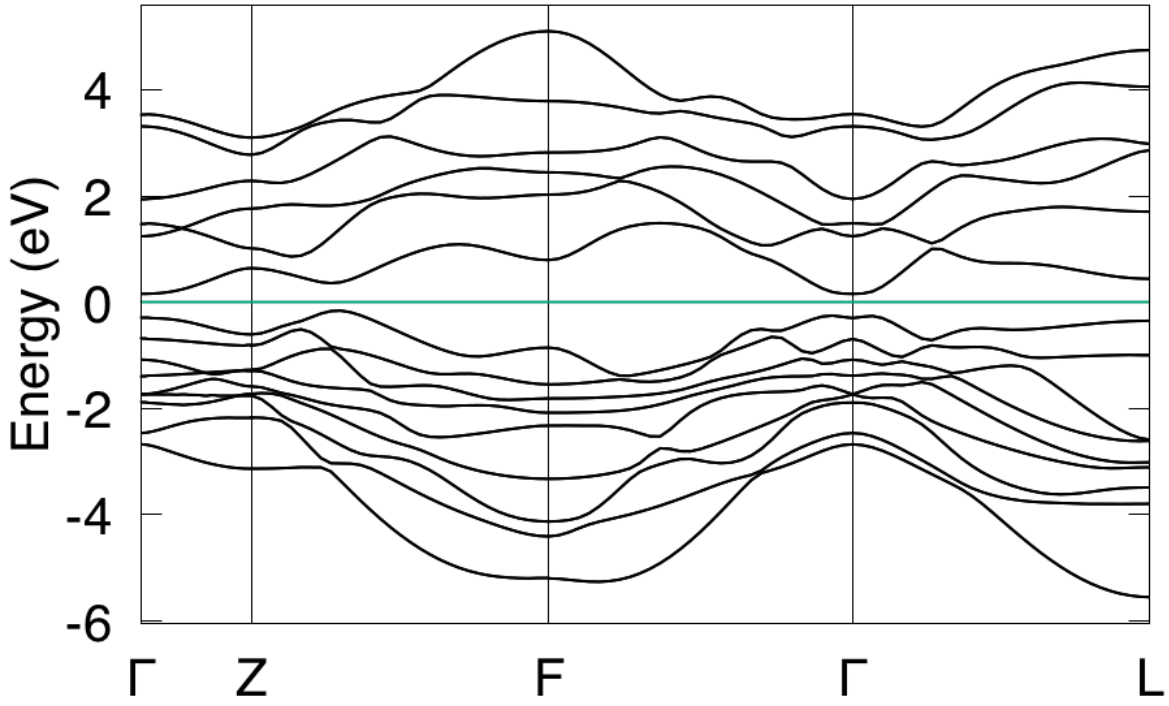
2.1.1 First principle calculation

Firstly, you need to study the electronic structure of Bi2Se3 with first-principle software package, like VASP, Wien2k, Abinit, Quantum-espresso et al. In this tutorial, we select VASP. Here is the calculated band structure.



2.1.2 Band structure

Then Wannier90 is applied to construct MLWF tight binding (TB) model (see more details from http://www.wannier.org). Here we only tell you that the p orbitals of Bi and Se are selected as the initial projectors for Wannier90. The band structure calculated from the MLWF-TB model is as follows



This band structure can be calculated directly from Wannier90. Also can be calculated from WannierTools. The settings in **WT.in** are

```
&CONTROL
BulkBand_calc         = T
/

&SYSTEM
SOC = 1                 ! soc
E_FERMI = 4.4195        ! e-fermi
/

&PARAMETERS
Nk1 = 41            ! number k points  odd number would be better
/
```

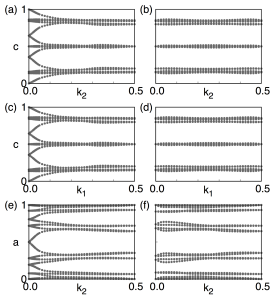
```
KPATH_BULK             ! k point path
4              ! number of k line only for bulk band
G 0.00000 0.00000 0.0000 Z 0.00000 0.00000 0.5000
Z 0.00000 0.00000 0.5000 F 0.50000 0.50000 0.0000
F 0.50000 0.50000 0.0000 G 0.00000 0.00000 0.0000
G 0.00000 0.00000 0.0000 L 0.50000 0.00000 0.0000
```

2.1.3 Z2 topological number

From the band structure above, it is clear that it is a insulator. This is also can be checked by calculating the density of state (DOS). In order to identify the topological properties, we have to calculate the Z2 topological number, which is valid for time-reversal invariant system with a continuous full gap in the Brilloin Zone. The Z2 topological number for 3D bulk system can be obtained from the calculation of the Wilson loop (Wannier charge center) for the six time-reversal invariant momentum plane. k1=0.0, k1=0.5; k2=0.0; k2=0.5; k3=0.0, k3=0.5. It can be done using WannierTools with setting in **WT.in**

```
&CONTROL
Z2_3D_calc             = T
/

&SYSTEM
SOC = 1                  ! soc
NumOccupied = 18         ! Number of occupied Wannier bands
/

&PARAMETERS
Nk1 = 41             ! number k points  odd number would be better
Nk2 = 41           ! number k points  odd number would be better
/
```

The resutls are

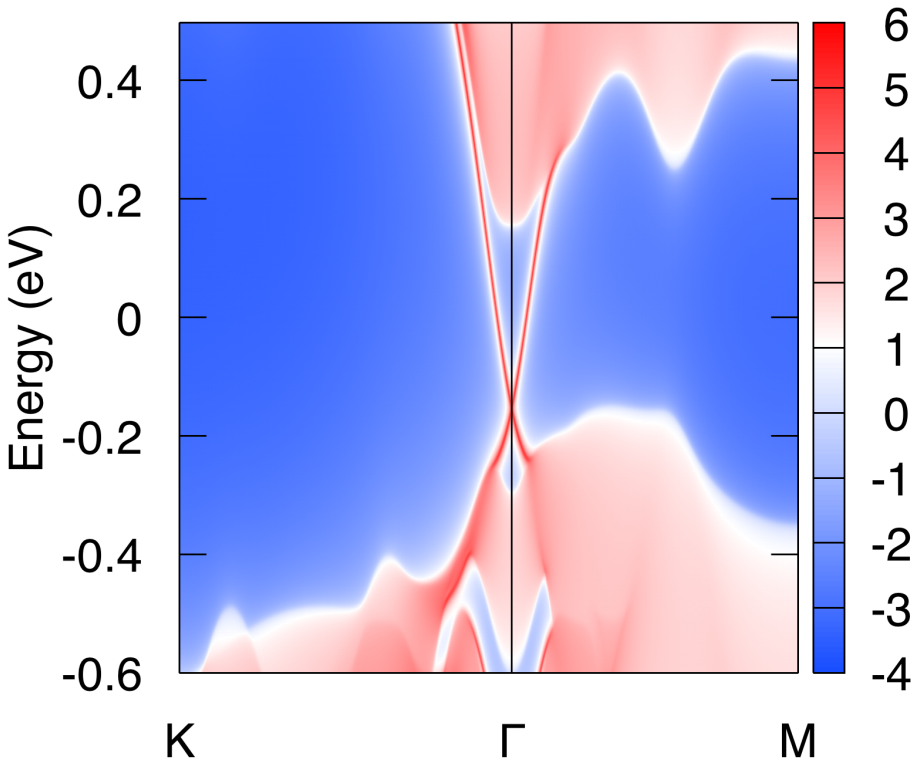


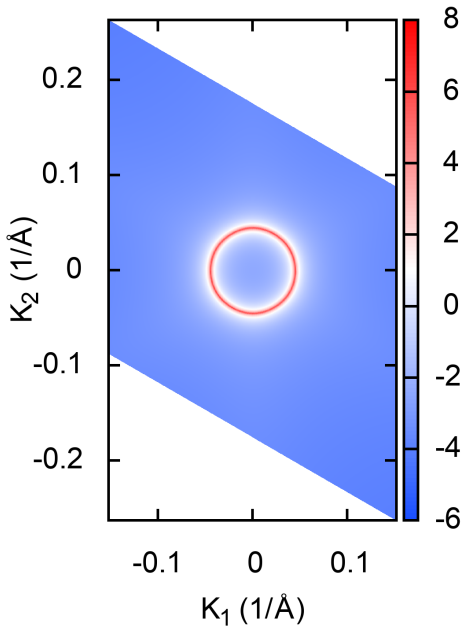
(a) k1=0.0, z2=1; (b) k1=0.5, z2=0; (c) k2=0.0, z2=1; (d) k2=0.5, z2=0; (e) k3=0.0, z2=1; (f) k3=0.5, z2=0;

So the bulk Z2 topological number is (1, 000), which means a strongly topological insulator.

2.1.4 Surface state

The surface states are the correspondence to the non-trivial bulk topology. They are detectable from ARPES experiments. The calculated surface states of Bi2Se3 on (0001) surface are


Energy (eV)
0.4
0.2
0
-0.2
-0.4
-0.6
K
Γ
M
6
5
4
3
2
1
0
-1
-2
-3
-4



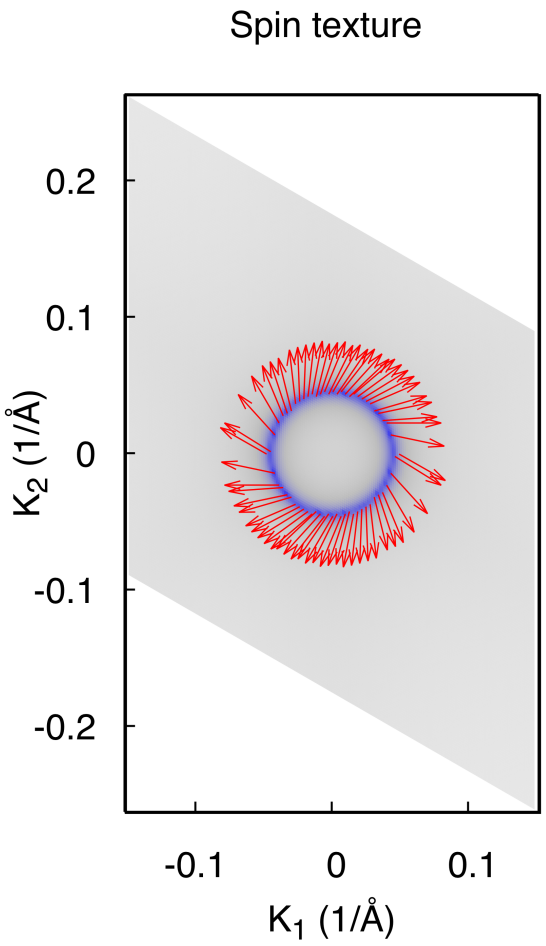
The settings in **WT.in** are

```
&CONTROL
SlabSS_calc            = T
SlabArc_calc           = T
/

&SYSTEM
SOC = 1                  ! soc
NumOccupied = 18         ! Number of occupied Wannier bands
E_FERMI = 4.4195         ! e-fermi
/

&PARAMETERS
Eta_Arc = 0.001      ! infinite small value, like brodening
E_arc = 0.0          ! energy for calculate Fermi Arc
OmegaMin = -0.6      ! energy interval
OmegaMax =  0.5      ! energy interval
OmegaNum = 401       ! omega number
Nk1 = 101            ! number k points  odd number would be better
Nk2 = 101            ! number k points  odd number would be better
/

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

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
```

2.1.5 Spin texture

Spin orbital coupling (SOC) is a very important to topological insulator. The spin texture of the surface states will form due to SOC. WannierTools can calculate spin texture like this



by setting

```
&CONTROL
SlabSpintexture_calc  = F
/

&SYSTEM
```

```
SOC = 1                 ! soc
E_FERMI = 4.4195        ! e-fermi
/

&PARAMETERS
Eta_Arc = 0.001     ! infinite small value, like brodening
E_arc = 0.0         ! energy for calculate Fermi Arc
Nk1 = 101           ! number k points  odd number would be better
Nk2 = 101           ! number k points  odd number would be better
/

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

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
```

2.1.6 Full settings in WT.in of Bi2Se3

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


&CONTROL
BulkBand_calc         = T
BulkFS_calc           = F
BulkGap_cube_calc     = F
BulkGap_plane_calc    = F
SlabBand_calc         = T
WireBand_calc         = F
SlabSS_calc           = T
SlabArc_calc          = T
SlabQPI_calc          = F
Z2_3D_calc            = T
Chern_3D_calc         = F
SlabSpintexture_calc  = F
Wanniercenter_calc    = F
BerryPhase_calc       = F
BerryCurvature_calc   = F
EffectiveMass_calc    = F
/

&SYSTEM
NSLAB = 10              ! for thin film system
NSLAB1= 4               ! nanowire system
NSLAB2= 4               ! nanowire system
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
Eta_Arc = 0.001     ! infinite small value, like brodening
E_arc = 0.0         ! energy for calculate Fermi Arc
OmegaNum = 401      ! omega number
OmegaMin = -0.6     ! energy interval
OmegaMax =  0.5     ! energy interval
Nk1 = 41            ! number k points  odd number would be better
Nk2 = 41           ! number k points  odd number would be better
Nk3 = 21           ! number k points  odd number would be better
NP = 1              ! 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
 0.00  0.00  0.00   ! Original point for 3D k plane  kz=0.5, bar{x}, along ky
 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

 0.00  0.00  0.00   ! Original point for 3D k plane  kz=0.0, bar{x}, along ky
 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

 0.00  0.50  0.00   ! Original point for 3D k plane  ky=0.5, bar{z}, along kx
 0.00  0.00  1.00   ! The first vector to define 3d k space plane
 0.50  0.00  0.00   ! The second vector to define 3d k space plane

 0.00  0.00  0.00   ! Original point for 3D k plane  ky=0, bar{z}, along kx
 0.00  0.00  1.00   ! The first vector to define 3d k space plane
 0.50  0.00  0.00   ! The second vector to define 3d k space plane

 0.50  0.00  0.00   ! Original point for 3D k plane  kx=0.5, bar{z}, along ky
 0.00  0.00  1.00   ! The first vector to define 3d k space plane
 0.00  0.50  0.00   ! The second vector to define 3d k space plane

 0.00  0.00  0.00   ! Original point for 3D k plane  kx=0, bar{z}, along ky
 0.00  0.00  1.00   ! 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


EFFECTIVE_MASS      ! optional
2                   ! The i'th band to be calculated
0.01                ! k step in unit of (1/Angstrom)
0.0 0.0 0.0         ! k point where the effective mass calculated.


WANNIER_CENTRES     ! copy from wannier90.wout
Cartesian
 -0.000040  -1.194745   6.638646
  0.000038  -1.196699   6.640059
 -0.000032  -1.192363   6.640243
 -0.000086  -3.583414   2.908040
  0.000047  -3.581457   2.906587
 -0.000033  -3.585864   2.906443
 -0.000001   1.194527   4.773338
  0.000003   1.194538   4.773336
 -0.000037   1.194536   4.773327
  0.000006  -1.194384   1.130261
 -0.000018  -1.216986   1.140267
  0.000007  -1.172216   1.140684
  0.000011  -3.583770   8.416406
 -0.000002  -3.561169   8.406398
```

```
-0.000007  -3.605960   8.405979
 0.000086  -1.194737   6.638626
-0.000047  -1.196693   6.640080
 0.000033  -1.192286   6.640223
 0.000040  -3.583406   2.908021
-0.000038  -3.581452   2.906608
 0.000032  -3.585788   2.906424
 0.000001   1.194548   4.773330
-0.000003   1.194537   4.773332
 0.000037   1.194539   4.773340
-0.000011  -1.194381   1.130260
 0.000002  -1.216981   1.140268
 0.000007  -1.172191   1.140687
-0.000006  -3.583766   8.416405
 0.000018  -3.561165   8.406400
-0.000007  -3.605935   8.405982
```

2.2 WTe2 (Type II Weyl semimetal)

to be continue

2.3 IrF4 (Nodal Chain metals)

to be continue