Topological properties with WannierTools

Tutorial III: Twisted bilayer graphene

Hands-on session

Wannier 2022 Summer School Trieste

Hands-on based on WannierTools v2.6.2

Useful information about WannierTools:
Documentation: www.wanniertools.com
Forum: www.wanniertools.org
Open source: https://github.com/quanshengwu/wannier_tools
Corresponding to: Dr. QuanSheng Wu, quansheng.wu@iphy.ac.cn

Tips: Open file.eps, file.pdf file in Linux, you can use command "evince". such as

```
$ evince file.eps
```

Open file.png file in Linux, you can use command "eog". such as

```
$ eog file.png
```

In this session we will study twisted bilayer graphene using WannierTools.

1 How to use tgtbgen to generate crystal structure and TB model for twisted graphene systems

A twisted bilayer graphene is made by put two graphene sheet on top of each other and make a twist. Recently, more complex stacking configuration of twisted graphene system are made, such as twisted mono-bilayer graphene, twisted double bilayer graphene, mirror symmetric twist trilayer graphene, twisted N+M generic stacking graphene. *tgtbgen* program will generate crystal structure and tight-binding model of generic stacking twisted graphene systems. *tgtbgen* is a subprogram of WannierTools.

https://github.com/quanshengwu/wannier_tools/tree/master/utility/twisted_graphene_system_tight_binding

The input file is *system.in*, output file is *wt.in* and *TG_hr.dat* which are the input files of *WannierTools*. *tgtbgen* can accept provided crystal structure stored as VASP format POSCAR or generate POSCAR with given stacking configuration. This is controled by a tag "use_poscar". *tgtbgen* can either generate dense-format stored or sparse-format stored tight-binding parameters "TG_hr.dat". This is controled by a tag "hr_generate".
Here is a typical input file *system.in* of *tgtbgen* to generate a sparse-format stored TB model of a mirror-symmetric twisted A-B-A trilayer graphene with twist angle 7.34°m=4).

```
&PARAMETERS
! nubmer of layers   trilayer
number_layers =      3

! twisted index m, theta= acos((3d0*m*m + 3d0*m + 0.5d0)/(3d0*m*m + 3d0*m + 1d0))
twisted_index_m =      4
```

```
! twisted angle array, unit is theta; number_layers numbers; mirror symmetrical twisted trilayer graphene
twisted_angle_array_input =   0 1 0

! stacking sequences only three values "A", "B", "C"; Number_layers numbers
stacking_sequences_input =    A  B A

! in plane lattice constant of graphene in unit of Angstrom
lattice_constant_a =     2.460000

! out of plane lattice constant of graphne in unit of Angstrom
lattice_constant_c =     3.360000

! vpppi (eV) p-p pi bond energy of Graphene, vpppi=-2.7 is a classical setting
vpppi =    -2.810000

! use POSCAR or not, use_poscar=F we will generate POSCAR, use_poscar=T use the given POSCAR
use_poscar =   F

! hr_generate = T, will generate hr.dat,  hr_generate = F, will not generate hr.dat
hr_generate =   T

! gen_sparse_hr = F, hr.dat is in wannier90 format,  hr_generate = T, hr.dat is in sparse format
gen_sparse_hr =   F

! hr_cutoff (eV) HmnR is set to zero if HmnR< hr_cutoff
hr_cutoff =     0.000100

! R is in [-iR_cut, -iR_cut+1, ..., iR_cut]
iR_cut =    1
/
```

We prepared several input files for mono-layer, bilayer, twisted bilayer graphene with different angles and a twisted trilayer graphene.

`system.in-monolayer,system.in-AB,system.in-TBG-m1,system.in-TBG-m4,system.in-TBG-m4-sparse,system.in-T-A-B-A-m4`

First, please copy the tutorial input files of the exercise:

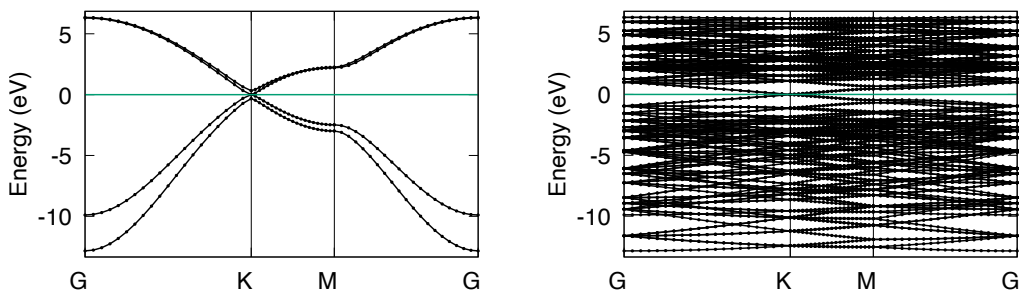
```
$ git clone https://github.com/wannier-developers/wannier-tutorials.git
$ cd wannier-tutorial/2022_05_Trieste/DAY4_AM_1_WannierTools/ex2/TBG-bands
```

▶ Obtain "wt.in" and "TG_hr.dat" for bilayer graphene and TBG with twist angle 7.34° (m=4), and run WannierTools to get band structures.

```
# scripts for bilayer graphene
$ cp system.in-AB system.in
$ tgtbgen
$ mpiexec -np 4 wt.x
$ gnuplot bulkek.gnu
$ evince bulkek.pdf

# scripts for TBG
$ cp system.in-TBG-m4 system.in
$ tgtbgen
$ mpiexec -np 4 wt.x
$ gnuplot bulkek.gnu
$ evince bulkek.pdf
```

The band structure looks like this



▶ Exercises

1. Generate sparse-format stored TB model of TBG with twist angle 7.34 °. The input file of tgtbgen is prepared as "system.in-TBG-m4-sparse".
2. Compare the difference of wt.in and "TG_hr.dat" between the sparse- and dense-format TB model generated by tgtbgen.
3. Run WannierTools to get the band structure and compare the time cost difference between two methods.
4. Get the band stucture of mirror-symmetric twisted trilayer graphene with input file "system.in-T-A-B-A-m4".
5. Play with tgtbgen by changing "system.in" to get band structures for different stacking configuration of twisted graphene systems. For example: AB-AB, A-AB-A, A-ABA et al. where "-" means there is a twist between two segments.

2 Wilson loop, band unfolding of a TBG

In the previous section, we learned how to construct TB model and get the band structures of twist graphene systems. In this section, we will use WannierTools to study the Wilson loop of the "flat bands" and perform band unfolding of a TBG with twist angle 7.34°.
The working directory is

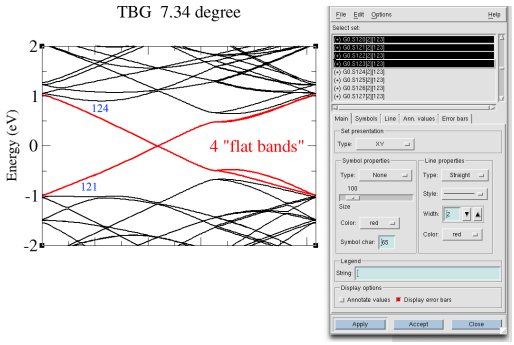
```
$ cd wannier-tutorial/2022_05_Trieste/DAY4_AM_1_WannierTools/ex2/TBG-7.34degree
$ tar xzvf TG_hr.dat-dense.tar.gz
$ tar xzvf TG_hr.dat-sparse.tar.gz
```

2.1 Wilson loop (Wannier charge center)

In order to obtain the Wilson loop, we need to figure out the band indices of the 4 "flat bands". The sparse version is more tricky for large twist angle since the "flat bands" are not "well" separate from the remote bands, so it's not easy to obtain their bands. However, for small angles, the size of the unit cell is very huge (e.g. num_wann=11164 for 1.08°) so that it's not suitable for tutorial. So in this tutorial we will take the dense format case as an example.
First, we need to obtain the band structure, and use software *xmgrace* to get the band indices.

```
$ cp wt.in-bands-dense wt.in
$ mpiexec -np 4 wt.x
$ xmgrace bulkek.dat
```

You can double clik the energy band lines in the xmgrace plot to get the band indices. You need to be careful that the starting index in xmgrace is 0. Please check the following figure.



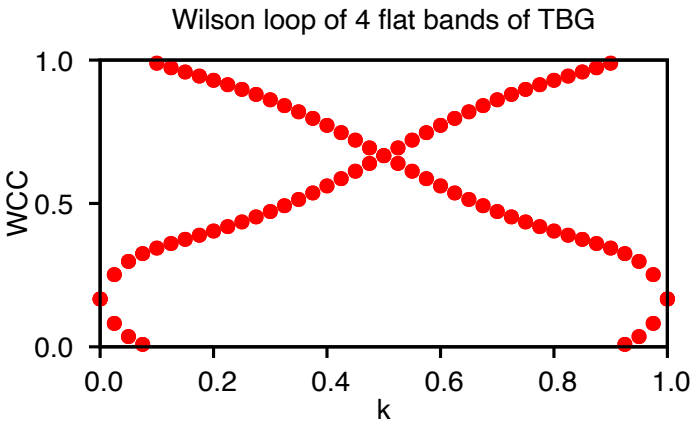
The indices of 4 "flat bands" start from 121 instead of 1. So we have to specify it in the "wt.in" by setting the card "SELECTED_OCCUPIED_BANDS" like this

```
SELECTED_OCCUPIED_BANDS
121-124
```

The full settings is prepared in file "wt.in-wilsonloop-dense".

```
$ cp wt.in-wilsonloop-dense wt.in
$ mpiexec -np 4 wt.x
$ gnuplot wcc.gnu0
$ evince wcc.eps
```

Then you will get the following non-trivial Wilson loop which is consistent with Fig.1b of reference PhysRevLett.123.036401.



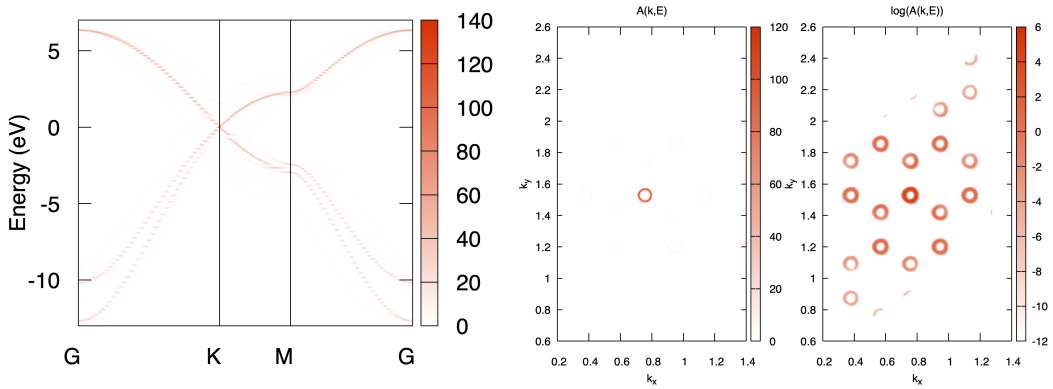
2.2 Band unfolding

When we look at the energy bands of a TBG system, the energy bands are very complex. It's not easy to tell something from it except the 4 flat bands close to the Fermi level. In order to tell how does the moire pattern affect the energy bands of Graphene, we need project the wave functions in the supercell onto the plane wave defined by the unit cell of Graphene. Such kind of projection is called the band unfolding procedure. The defination of the unfolded unit cell can be found on webpage https://www.wanniertools.org/examples/band-unfolding/.
WannierTools provides k-path mode and k-plane model when performing band unfolding.

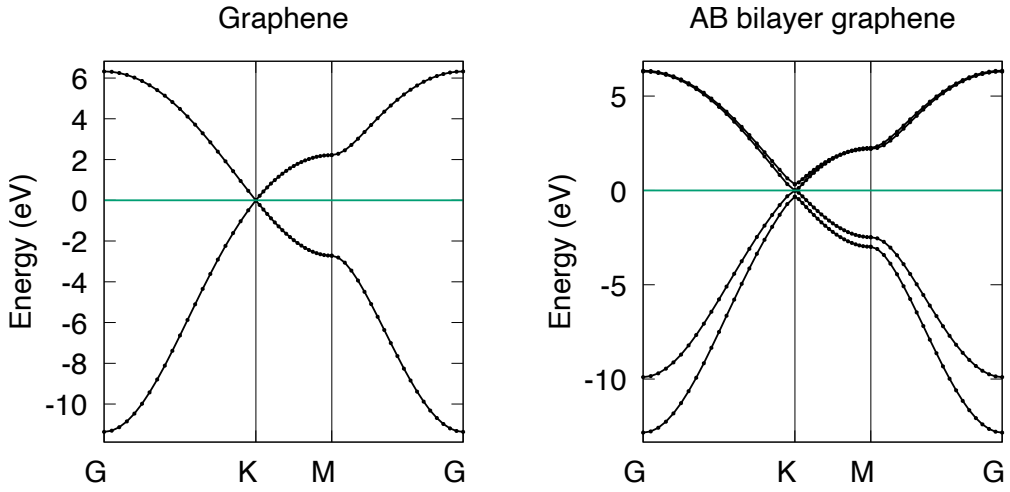
```
# k-path mode
$ cp wt.in-unfolding-kpath wt.in
$ mpiexec -np 4 wt.x
$ gnuplot spectrum_unfold_kpath.gnu
$ eog spectrum_unfold_kpath.png

# k-plane mode
$ cp wt.in-unfolding-kplane wt.in
$ mpiexec -np 4 wt.x
$ gnuplot spectrum_unfold_kplane.gnu0
$ eog spectrum_unfold_kplane.png
```

The results should like



The band structure of monolayer and bilayer graphene are shown as below



▶ Excercises

1. Please compare the unfolded energy bands with the band structure of monolayer and bilayer graphene.
2. Increase kmesh "Nk1" and "Nk2" in "wt.in-unfolding-kplane" to get high-resolution plot.
3. change "E_arc" in "wt.in-unfolding-kplane" to get different iso-energy spectrum plot.