KPROJ Tutorial

Mingxing Chen

School of Physics and Electronics,

Hunan Normal University,

Changsha, Hunan 410081, China

email: mxchen@hunnu.edu.cn

Introduction

- This manual describes using KPROJ to obtain unfolded electronic bands by the k-projection method for doped materials and interfaces modeled by supercells within the framework of first-principles method.
- The method has been given in Ref. [1] [Phys. Rev. B 98, 245421 (2018)]. This paper should be cited if you publish results using KPROJ. Some applications of the code can be found in Refs. [2-11].
- Essentially, it starts off with projections of the wavefunctions for a supercell onto the k-points of the BZ of the primitive cell. Then, it calculates the weights of the projected wavefunctions and plot the band structure with the weights. A layer-projection scheme is used for unfolding the electronic bands of interfaces, which significantly accelerates the calculations.
- Currently, the code has interfaces with VASP, Quantum Espresso, Ab-init, ABACUS, and Phonopy.

Programs come with the package

Program	Path	Use
kproj	~/kproj/src	Band unfolding
supercell	~/kproj/utils/supercell	Constructing supercell and generate the **pos_map.dat** file
kpts_path	~/kproj/src/utils	Genertaing *k*-points along the given path
ksplit	~/kproj/src/utils	Separating the k-points in **KPOINTS** into multiple segments
futils	~/kproj/src	Preparing the data prior to plotting
mergefile	~/kproj/src	Merge multiple **bs_projected.dat_i** files in to one
pos2stru	~/kproj/utils/supercell/pos2stru	Convert **POSCAR** to **STRU** for ABACUS and Phonopy calculations

Installation

- The distribution includes a **Makefile** for the intel fortran compiler. First, make sure that you have installed the FFTW and added its include directory into the **Makefile** of KPROJ.

```
cd /kproj/src
vim Makeile
```

Modify the FFTW_INCLUDES parameter ：

FFTW_INCLUDES = /your_fftw_directory/include

- Second, compile the XML library in **~/kproj/src**

```
make fox
```

- Finally, use the make command and once the program has been succesfully compiled, an executable named ***kproj*** will be found.

```
make
```

- Moreover, there is a program name ***futils*** for preparing files for plotting based on bs_projected.dat

```
cd ~/kproj/src
make futils
```

Installation

- ***kpts_path*** and ***ksplit*** are used to generate and separate the *k*-points, respectively , which can be compiled by :

```
cd ~/kproj/utils
ifort  -o kpts_path kpts_path.f90
ifort  -o ksplit ksplit.f90
```

- ***mergefile*** that is used to merge several **bs_projected.dat_i** files can be compiled by :

```
cd ~/kproj/src
make mergefile
```

- ***supercell*** can be compiled by :

```
cd ~/kproj/utils/supercell
make supercell
```

- ***pos2stru*** that is used to convert **POSCAR** to **STRU** can be compiled by:

```
cd ~/kproj/utils/supercell/pos2stru
make pos2stru
```

Brief decrition of doing band unfolding using KPROJ

Step 1: Band calculation

See **note 1** for setting the *k*-points for a supercell.
See **note 2** for generating supercells using program ***supercell***.
See **note 3** for building structures for heterostructures.

Step 2: Band unfolding

See **note 4** for files required by the program with different interfaces.
See **note 5** for description of input parameters.
See **note 6** for LCAO and Phonopy calculations.

Step 3: Plotting

See **note 7** for data processing using ***futils*** prior to plotting.
See **note 8** for plotting using ***xmgrace***, ***gnuplot*** and ***opendx***.
See **note 9** for splitting the band calculations for **large** systems.
See **note 10** for merge **bs_projected.dat** files for splitted calculations.

Procedures for doing band unfolding

Step 1 : Generate the supercell.

We strongly recommend to use ***supercell*** program for generating supercells. The reason is that it can automatically generate the **pos_map.dat** file, which is an important input file for unfolding the band structures obtained by ABACUS and Phonopy.

Step 2 : Band calculation.

Note that the *k*-points are different from those for the standard band calculations. The *k*-points are for the primitive cell, but have to be given in the reciprocal lattice vectors of the supercell. For instance, the *k*-point (1/3, 1/3, 0) for graphene in the primitive cell will be (5/3, 5/3, 0) for a 5 x 5 supercell. Detailed explanations will be given in **Note 1**.

If the system is large, you may split the band calculation into multiple jobs. In this case, you have to split the KPOINTS file containing the k-points list into several similar files named *KPOINTS_i*. Then one need to merge the corresponding **bs_projected.dat_i** into one for plotting (see ~/kproj/exampe/vasp_examples for Ge_Al_2O_3).

The file that stores the wavefunctions should be generated:

Package	Wavefunction File
VASP	WAVECAR
Quantum Espresso	wfc$num.dat
Ab-init	user-defined
Phonopy	band.yaml
ABACUS	LOWF_K_$num.dat

Procedures for doing band unfolding

Step 3 : KPROJ calculation.

The **wavefunction** file and **INKPROJ** are required.

For interfaces/heterostructures, users can set proper values for **LZLAYER, zlay1, zlay2** to perform layer projection (Currently, this feature is only available for the wavefunction in planewaves, i.e., generated by VASP\Quantum Espresso\Ab-init).

For band calculations using LCAO and Phonopy, the file **pos_map.dat** is required. For interfaces, the tag **proj_part** has to be specified for layer k-projection.

Step 4 : Preparation for plotting.

Set the right **E-fermi** in **bs_projected.dat**.

If you have multiple **bs_projected.dat** for a large system, please first merge them into one file. A program named ***mergefile*** under /kproj/src is provided for this operation. An example is given in Note 9.

Run ***futils*** to generate plotting files for different plotting tools.

futils can be obtained by "make futils" under /kproj/src.

Step 5 : Plotting

xmgrace/gnuplot/opendx

A detailed explanation will be provided in **Note 8**.

Note 1:
Set the *k*-points for band unfolding

- Using the **transformation matrix(M)** between the primitive cell and the supercell

lattice vectors for the **supercell**

$$\begin{pmatrix} A_1 \\ A_2 \\ A_3 \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

lattice vectors for the **primitive cell**

$$\vec{A} = M\vec{a}$$

- In reciprocal space

reciprocal vectors the **primitive cell**

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} M_{11} & M_{21} & M_{31} \\ M_{12} & M_{22} & M_{32} \\ M_{13} & M_{23} & M_{33} \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ B_3 \end{pmatrix}$$

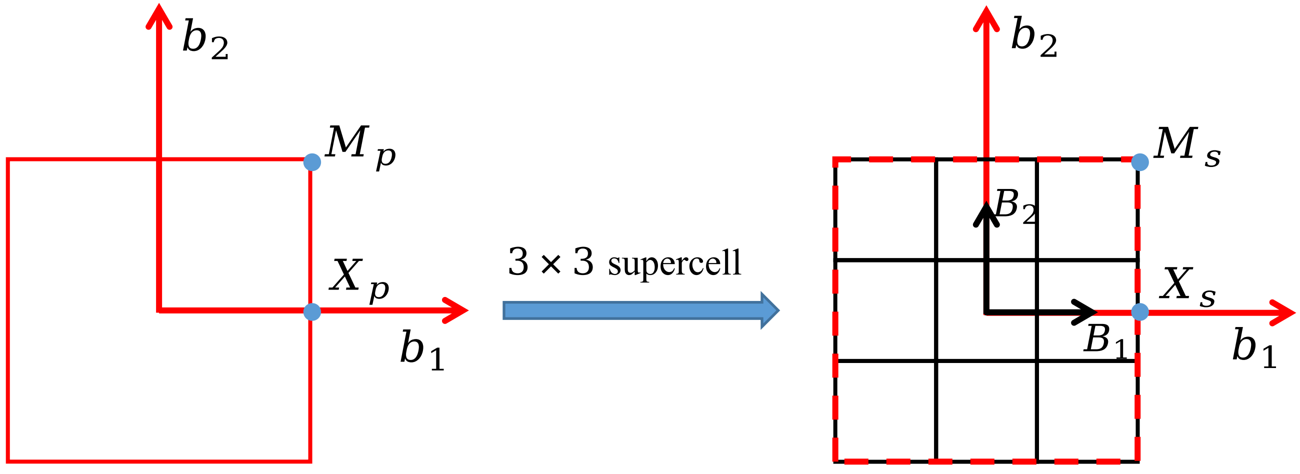
reciprocal vectors the **supercell**

$$\vec{b} = M^T \vec{B}$$

- Remember that we want the unfolded band structure for the primitive cell. So **the k-points used for the band calculation should be those in the BZ of the primitive cell.** If these k-points are given in the direct coordinates (in the basis $\boldsymbol{b_i}$). **We need to write them in the reciprocal lattice vectors of the supercell, i.e., $\boldsymbol{B_i}$.**

$$k_s = M k_p$$

Note 1:
Set the *k*-points for band unfolding



Primitive cell

Supercell

Brilloin zones

High-symmetry points of the **primitive cell** :

$$M_p = 0.5\vec{b}_1 + 0.5\vec{b}_2 + 0\vec{b}_3 = (0.5,0.5,0)$$

$$X_p = 0.5\vec{b}_1 + 0\vec{b}_2 + 0\vec{b}_3 = (0.5,0,0)$$

The high-symmetry points of the primitive cell are expressed using the reciprocal lattice vectors of the **supercell** :

$$M_s = 1.5\vec{B}_1 + 1.5\vec{B}_2 + 0\vec{B}_3 = (1.5,1.5,0)$$

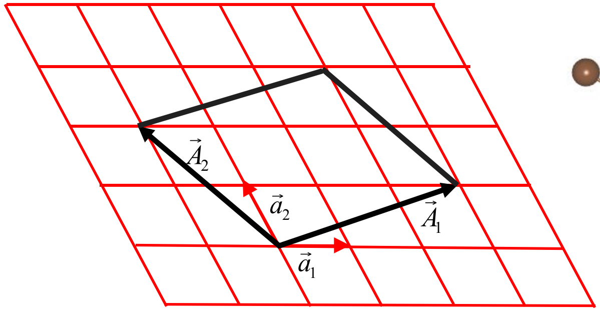
$$X_s = 1.5\vec{B}_1 + 0\vec{B}_2 + 0\vec{B}_3 = (1.5,0,0)$$

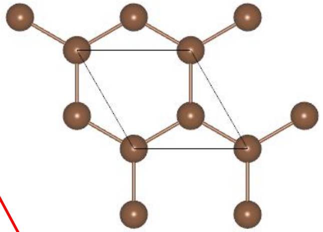
$$\begin{pmatrix}3&0&0\\0&3&0\\0&0&1\end{pmatrix}\begin{pmatrix}0.5\\0.5\\0\end{pmatrix}=\begin{pmatrix}1.5\\1.5\\0\end{pmatrix} \qquad \begin{pmatrix}3&0&0\\0&3&0\\0&0&1\end{pmatrix}\begin{pmatrix}0.5\\0\\0\end{pmatrix}=\begin{pmatrix}1.5\\0\\0\end{pmatrix}$$

Note 2:

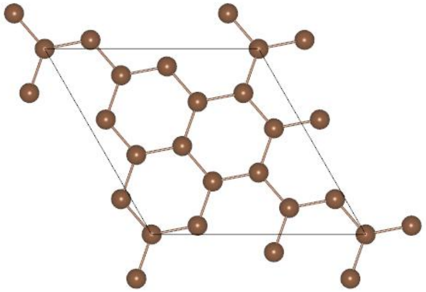
Generating supercells using program *supercell*.

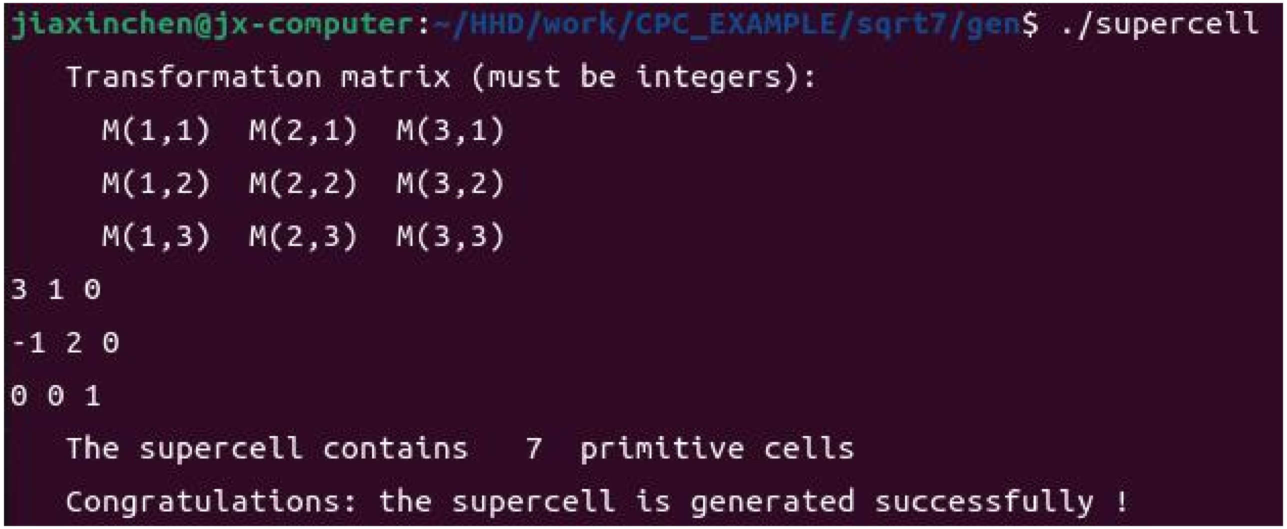
$\sqrt{7}\times\sqrt{7}$ **supercell of graphene**





$$\begin{pmatrix}\vec{A}_1\\ \vec{A}_2\\ \vec{A}_3\end{pmatrix}=\begin{pmatrix}3&1&0\\ -1&2&0\\ 0&0&1\end{pmatrix}\begin{pmatrix}\vec{a}_1\\ \vec{a}_2\\ \vec{a}_3\end{pmatrix}$$



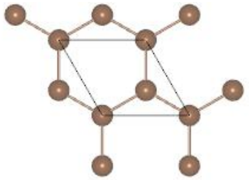


```
jiaxinchen@jx-computer:~/HHD/work/CPC_EXAMPLE/sqrt7/gen$ ./supercell
   Transformation matrix (must be integers):
     M(1,1)  M(2,1)  M(3,1)
     M(1,2)  M(2,2)  M(3,2)
     M(1,3)  M(2,3)  M(3,3)
3 1 0
-1 2 0
0 0 1
   The supercell contains    7  primitive cells
   Congratulations: the supercell is generated successfully !
```

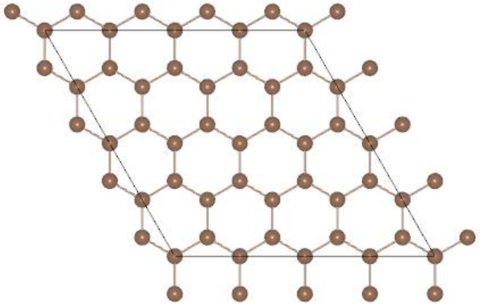
After running *supercell* with the transformation matrix, a file named **POSCAR_SUPER** and **pos_map.dat** will be generated.

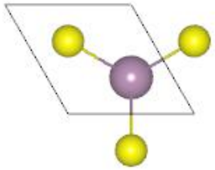
Note 3:
Building structures for heterostructures

4 × 4 graphene on 3 x 3 MoS_2

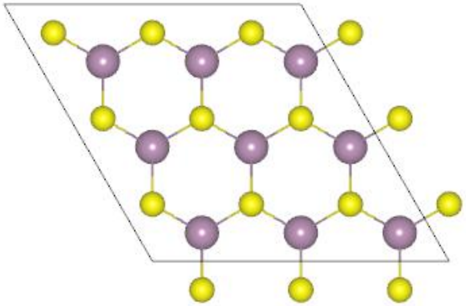


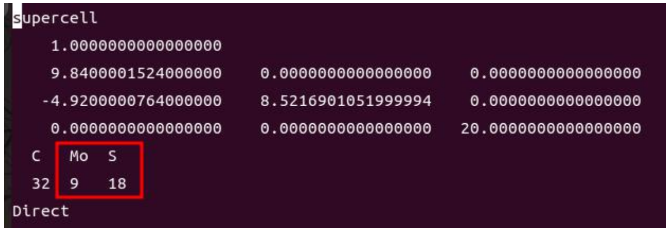
$$\xrightarrow{\begin{pmatrix}4 & 0 & 0\\ 0 & 4 & 0\\ 0 & 0 & 1\end{pmatrix}}$$



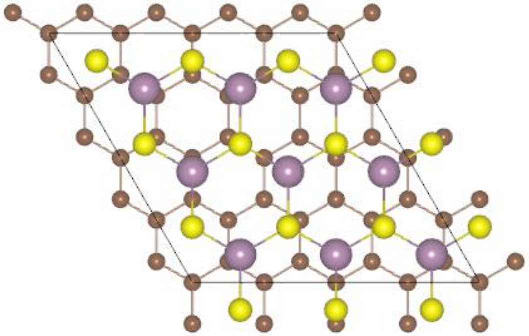


$$\xrightarrow{\begin{pmatrix}3 & 0 & 0\\ 0 & 3 & 0\\ 0 & 0 & 1\end{pmatrix}}$$





```
supercell
    1.0000000000000000
    9.8400001524000000    0.0000000000000000    0.0000000000000000
   -4.9200000764000000    8.5216901051999994    0.0000000000000000
    0.0000000000000000    0.0000000000000000   20.0000000000000000
  C   Mo   S
  32  9    18
Direct
```



Note 4:
Files Used by KPROJ

INKPROJ	in	input parameters
wavefunction file	in	eigenvector files output by different packages.
pos_map.dat	in	lattice and translation vectors (for interfaces with Phonopy and ABACUS)
OUTKPROJ	out	general output
bs_projected.dat	out	weights of the *k*-projected wavefunctions

Package	Wavefunction File
VASP	WAVECAR
Quantum Espresso	wfc$num.dat
Ab-init	user-defined
Phonopy	band.yaml
ABACUS	LOWF_K_$num.dat

Note 5: INKPROJ

- The **input parameters** are read by a user-friendly control. A sample of INPROJ for layer-projection along Z axis for a heterostructure:

```
LZLAYER = .TRUE.
zlay1 = 0.5
zlay2 = 0.62
MAT_P2S= 5 0 0 \
         0 5 0 \
         0 0 1
LSORBIT = .TRUE.
```

- **LXLAYER** (logical): If a layer-projection along the 1st axis will be done. Default: LXLAYER = .FALSE. When LXLAYER = .TRUE., **xlay1** and **xlay2** should be provided with suitable values

- **LYLAYER** (logical): If a layer-projection along the 2nd axis will be done. Default: LYLAYER = .FALSE. When LYLAYER = .TRUE., **ylay1** and **ylay2** should be provided with suitable values.

- **LZLAYER** (logical): If a layer-projection along the 3rd axis will be done. Default: LZLAYER = .FALSE. When LZLAYER = .TRUE., **zlay1** and **zlay2** should be provided with suitable values.

- When LXLAYER, LYLAYER, and LZLAYER are all .FALSE., the unfolding will be done for the whole space. This is useful for bulk calculations.

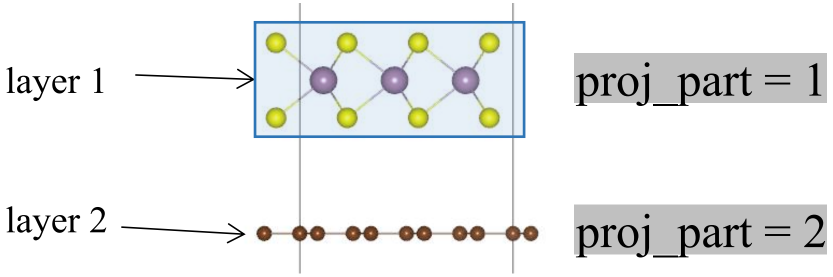
Note 5: INKPROJ

- **LSORBIT** (logical): If the WAVECAR was obtained from a spin-orbit coupling (noncollinear) calculation. Default: LSORBIT = .FALSE.

- **MAT_P2S** (integers): The transformation matrix from the primitive cell to the supercell. For instance, the MAT_P2S for a 5 × 5 supercell can be given either by :

```
MAT_P2S= 5 0 0 \
         0 5 0 \
         0 0 1
or
MAT_P2S= 5 0 0  0 5 0  0 0 1
```

- **proj_part** (integers): Project the wavefunction onto the specified layer (only for the interface with **ABACUS** and **Phonopy**). For a heterostructure contain two layers, the **proj_part** is given by **proj_part = nth** . It means projecting the wavefunction to the nth layers defined in file **pos_map.dat**. This tag is used for **LCAO (ABACUS)** and **Phonopy** calculations of interfaces in case you need layer *k*-projection.



layer 1

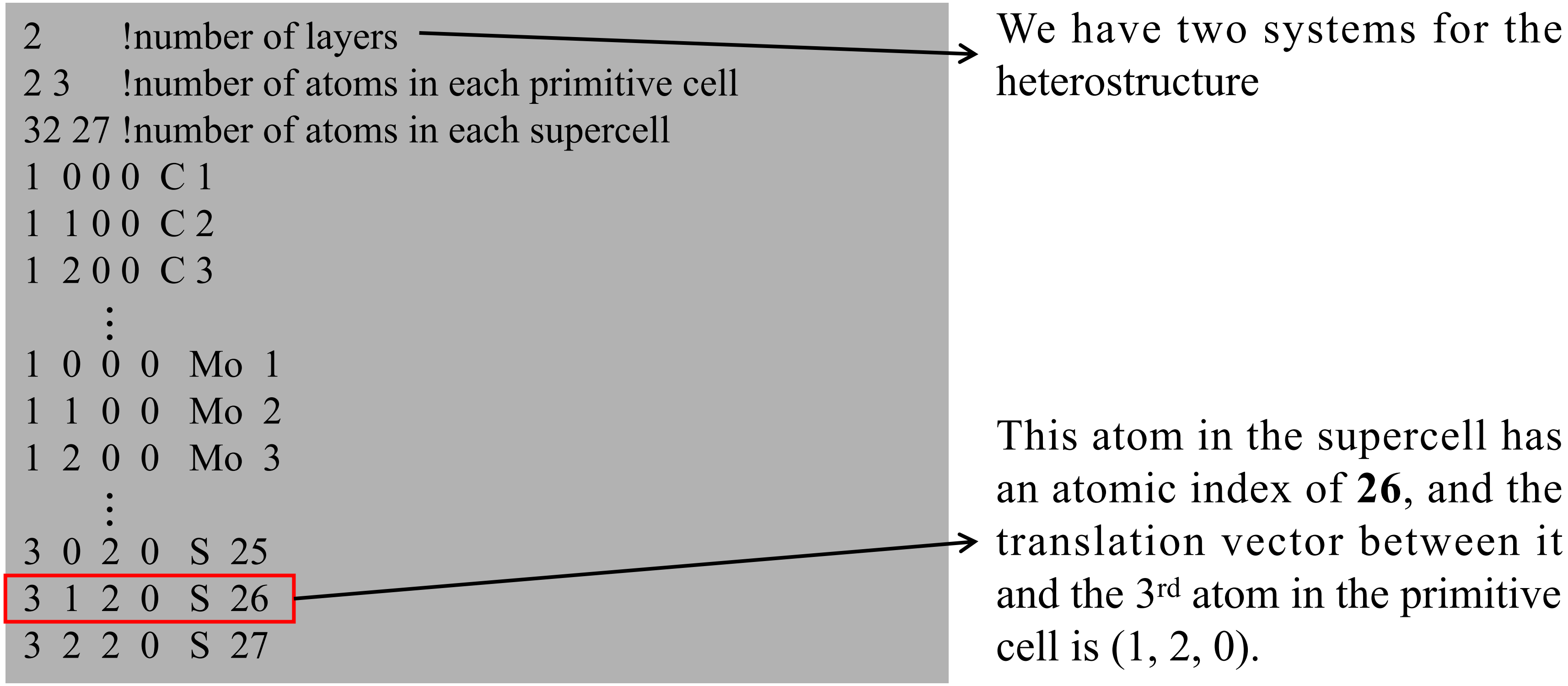
proj_part = 1

layer 2

proj_part = 2

pos_map.dat

- The file **pos_map.dat** is used for phonon and LCAO methods. It stores the number of atoms and translation vectors for the primitive cell and supercell. Below is an example of **pos_map.dat** for graphene/MoS_2 heterostructure :



```
2       !number of layers
2 3     !number of atoms in each primitive cell
32 27 !number of atoms in each supercell
1  0 0 0  C 1
1  1 0 0  C 2
1  2 0 0  C 3
       ⋮
1  0  0  0   Mo  1
1  1  0  0   Mo  2
1  2  0  0   Mo  3
       ⋮
3  0  2  0   S  25
3  1  2  0   S  26
3  2  2  0   S  27
```

We have two systems for the heterostructure

This atom in the supercell has an atomic index of **26**, and the translation vector between it and the 3rd atom in the primitive cell is (1, 2, 0).

- The program ***supercell*** under ~/**kproj/utils/supercell** can be used to generate supercells, which is accompanied by **pos_map.dat**.
- For heterostructures, users need to merge two **pos_map.dat** files into one. Details and an example is give in Note 6 .

Files used by KPROJ

VASP/QE/Ab-init interface

INKPROJ	**input**	**input parameters**
wavefunction	**input**	**the wavefunctions from DFT calculations**
OUTKPROJ	**output**	**general outputs**
bs_projected.dat	**output**	**origin data of unfolded band structure**

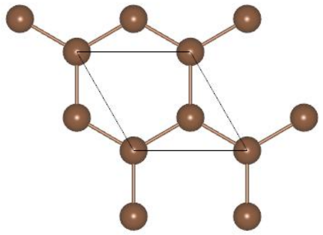
Package	**Wavefunction File**
VASP	WAVECAR
Quantum Espresso	wfc$num.dat
Ab-init	**user-defined**

It should be noted that the name of wavefunction output by Ab-init is user-defined. Therefore, we have added a parameter **WFCAB** in **INKPROJ**. When you specify the WFCAB (e.g. WFCAB = “wfcfile”), it will automatically switches to the interface with Ab-init.

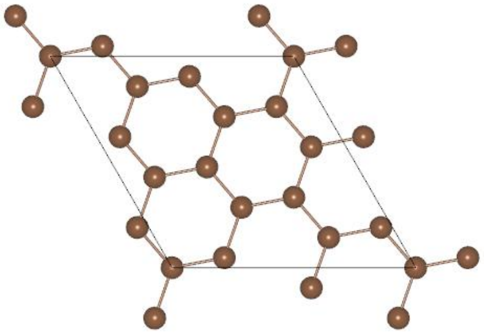
Example 1: $\sqrt{7}\times\sqrt{7}$ graphene

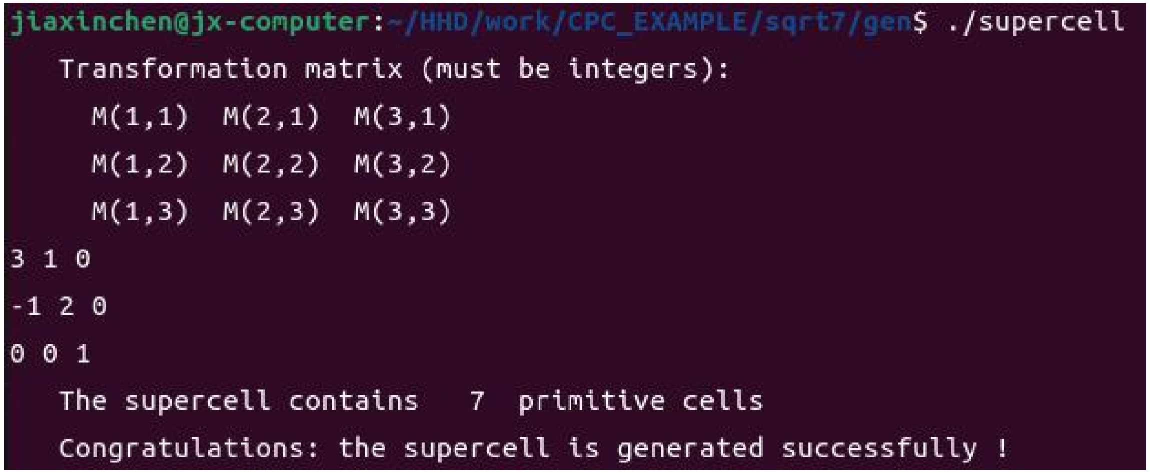
VASP/QE/Ab-init interface

Step 1 : Generate the supercell using program *supercell*.



$$\begin{pmatrix} 3 & 1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$





```
jiaxinchen@jx-computer:~/HHD/work/CPC_EXAMPLE/sqrt7/gen$ ./supercell
   Transformation matrix (must be integers):
     M(1,1)  M(2,1)  M(3,1)
     M(1,2)  M(2,2)  M(3,2)
     M(1,3)  M(2,3)  M(3,3)
3 1 0
-1 2 0
0 0 1
   The supercell contains    7  primitive cells
   Congratulations: the supercell is generated successfully !
```

The program *supercell* is under the ~/kproj/utils/supercell. Please make it before using.

Example 1: $\sqrt{7} \times \sqrt{7}$ graphene

VASP/QE/Ab-init interface

Step 1 : Generate the supercell using program *supercell*.

Step 2 : Do a band calculation to get the wavefunctions.

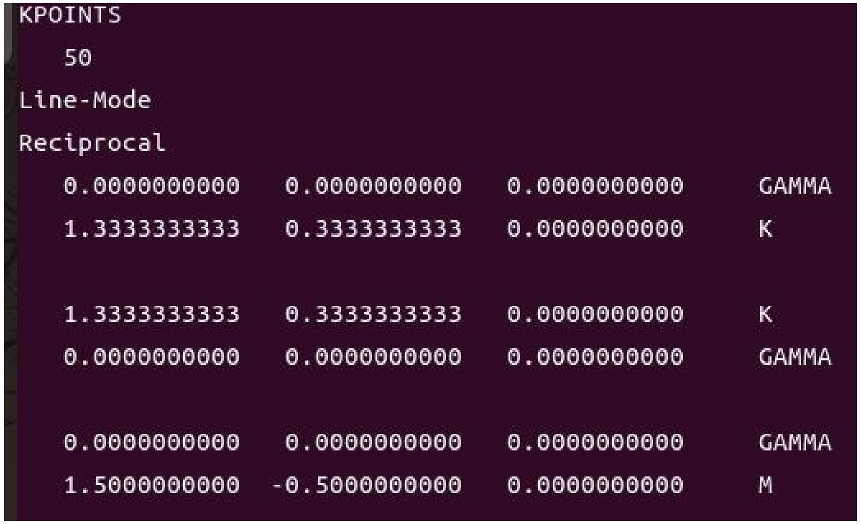
The high-symmetry points (in the unit of the reciprocal lattice vectors of the supercell) are:

$$M = \begin{pmatrix} 3 & 1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1.5 \\ -0.5 \\ 0 \end{pmatrix}$$

$$K = \begin{pmatrix} 3 & 1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1/3 \\ 1/3 \\ 0 \end{pmatrix} = \begin{pmatrix} 4/3 \\ 1/3 \\ 0 \end{pmatrix}$$

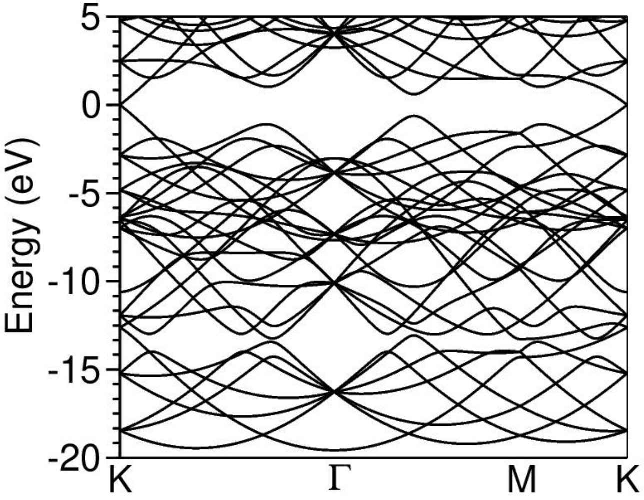
Set **LWAVE = .TRUE.** to output the wavefunction for VASP calculations.

The KPOINTS file used for the band calculations :



```
KPOINTS
   50
Line-Mode
Reciprocal
   0.0000000000    0.0000000000    0.0000000000     GAMMA
   1.3333333333    0.3333333333    0.0000000000     K

   1.3333333333    0.3333333333    0.0000000000     K
   0.0000000000    0.0000000000    0.0000000000     GAMMA

   0.0000000000    0.0000000000    0.0000000000     GAMMA
   1.5000000000   -0.5000000000    0.0000000000     M
```

Folded bands for $\sqrt{7} \times \sqrt{7}$ graphene :

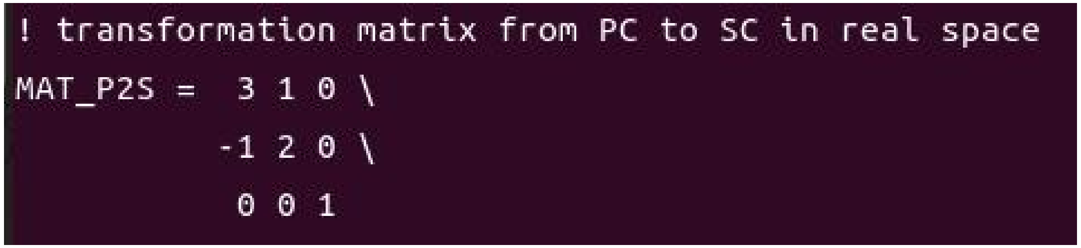


Example 1: $\sqrt{7} \times \sqrt{7}$ graphene

VASP/QE/Ab-init interface

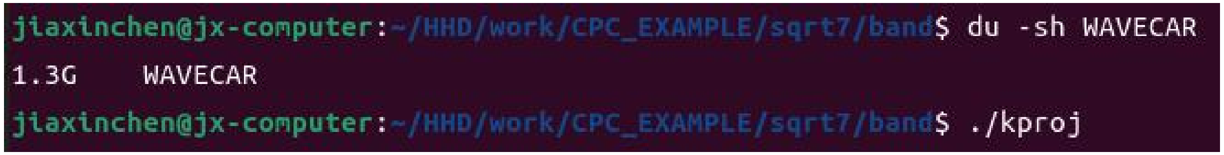
Step 3 : Do a KPROJ calculation.

The parameters used by the file INKPROJ are following :



```
! transformation matrix from PC to SC in real space
MAT_P2S =   3 1 0 \
           -1 2 0 \
            0 0 1
```

Check if the wavefunction was successfully generated, and then perform the *k*-projection calculation using **./kproj**



```
jiaxinchen@jx-computer:~/HHD/work/CPC_EXAMPLE/sqrt7/band$ du -sh WAVECAR
1.3G	WAVECAR
jiaxinchen@jx-computer:~/HHD/work/CPC_EXAMPLE/sqrt7/band$ ./kproj
```

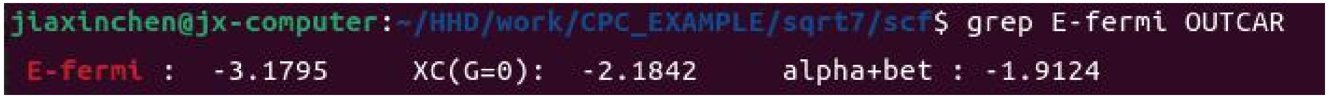
Reminder: the program ***kproj*** is under ~/kproj/src.

Example 1: $\sqrt{7} \times \sqrt{7}$ graphene

VASP/QE/Ab-init interface

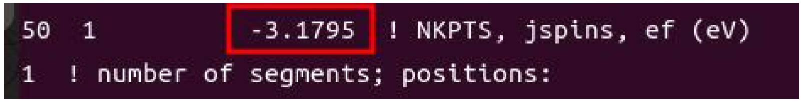
Step 4 : Preparing for plotting (Note 7)

Find the **Fermi level** from the SCF calculation and then fill it in the first line of the bs_projected.dat file.



```
jiaxinchen@jx-computer:~/HHD/work/CPC_EXAMPLE/sqrt7/scf$ grep E-fermi OUTCAR
 E-fermi :  -3.1795     XC(G=0):  -2.1842     alpha+bet : -1.9124
```

vim bs_projected.dat

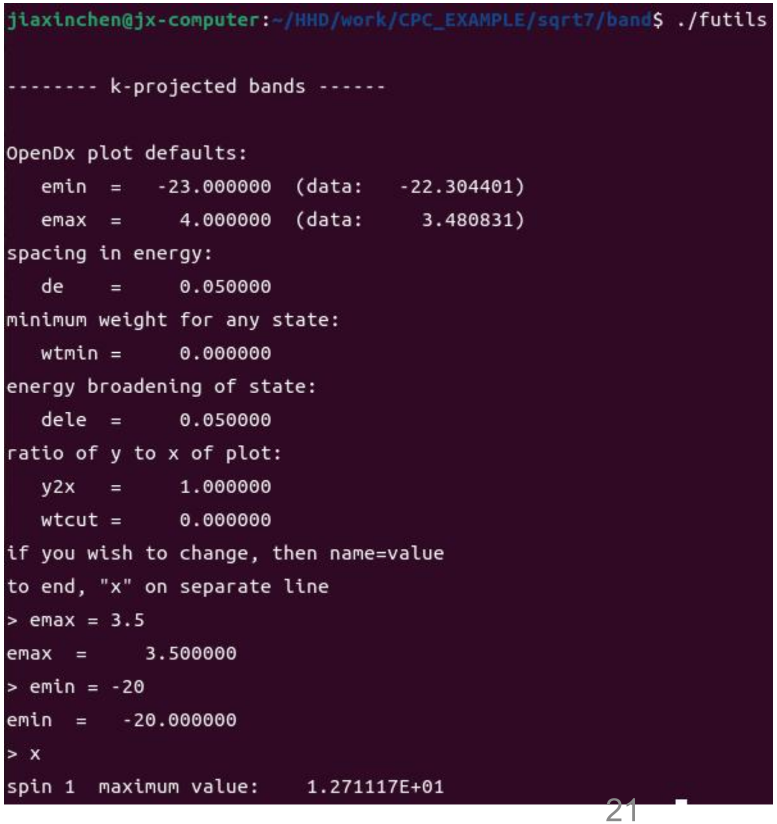


```
50  1           -3.1795  ! NKPTS, jspins, ef (eV)
1  ! number of segments; positions:
```

Run **./futils** to generate the plot files in different formats.

User needs to specify emin, emax, de and dele. Emin and Emax are used to define the energy window for plotting. de and dele are the energy grid and the smearing factor. **Please adjust de and dele for a nice plotting.** Finally, enter x and press Enter to finish the program.

Alternatively, user can use the script **run_futils** under ~/kproj/utils for the whole process (just run with **./run_futils** instead of the multiple steps for specifying the values of the parameters). Please provide the correct path to ***futuils*** in the file.



```
jiaxinchen@jx-computer:~/HHD/work/CPC_EXAMPLE/sqrt7/band$ ./futils

-------- k-projected bands ------

OpenDx plot defaults:
   emin  =   -23.000000  (data:   -22.304401)
   emax  =     4.000000  (data:     3.480831)
spacing in energy:
   de    =     0.050000
minimum weight for any state:
   wtmin =     0.000000
energy broadening of state:
   dele  =     0.050000
ratio of y to x of plot:
   y2x   =     1.000000
   wtcut =     0.000000
if you wish to change, then name=value
to end, "x" on separate line
> emax = 3.5
emax  =     3.500000
> emin = -20
emin  =   -20.000000
> x
spin 1  maximum value:    1.271117E+01
```

Outputs of futils

Nonmagnetic/SOC calculations:

band_kproj_1.dat
plotbnds.gnu

For gnuplot

plot_bands_1.dat
plot_bands_1.general

For opendx

bsplot.agr

For xmgrace

magnetic calculations:

band_kproj_1.dat	spin up	For gnuplot
band_kproj_2.dat	spin down	
band_kproj_tot.dat	total	
plotbnds.gnu		

plot_bands_1.dat plot_bands_1.general	spin up	For opendx
plot_bands_2.dat plot_bands_2.general	spin down	

Example 1: $\sqrt{7} \times \sqrt{7}$ graphene

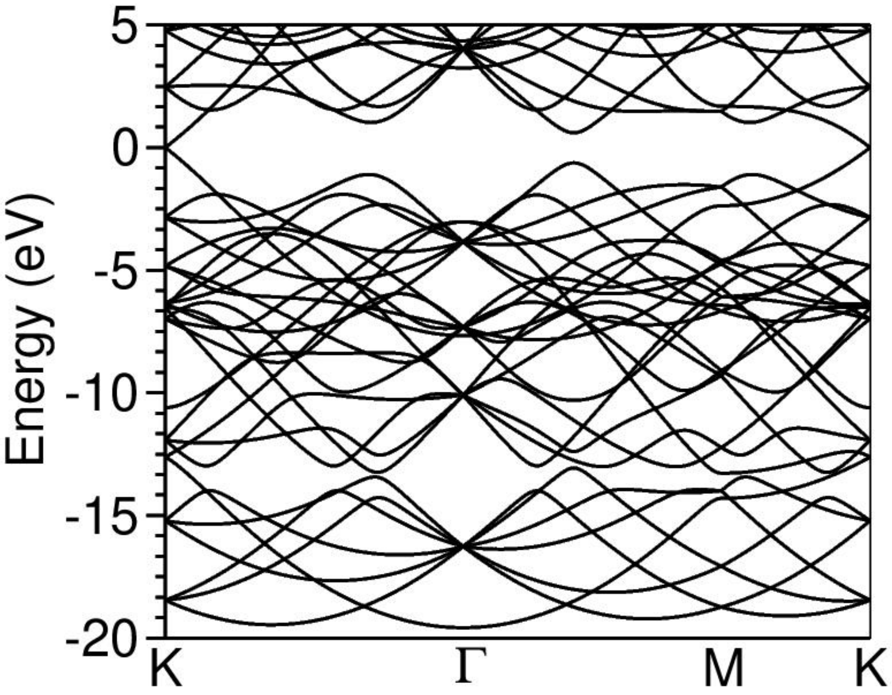
VASP/QE/Ab-init interface

Step 5 : Do the plotting (Note 8)

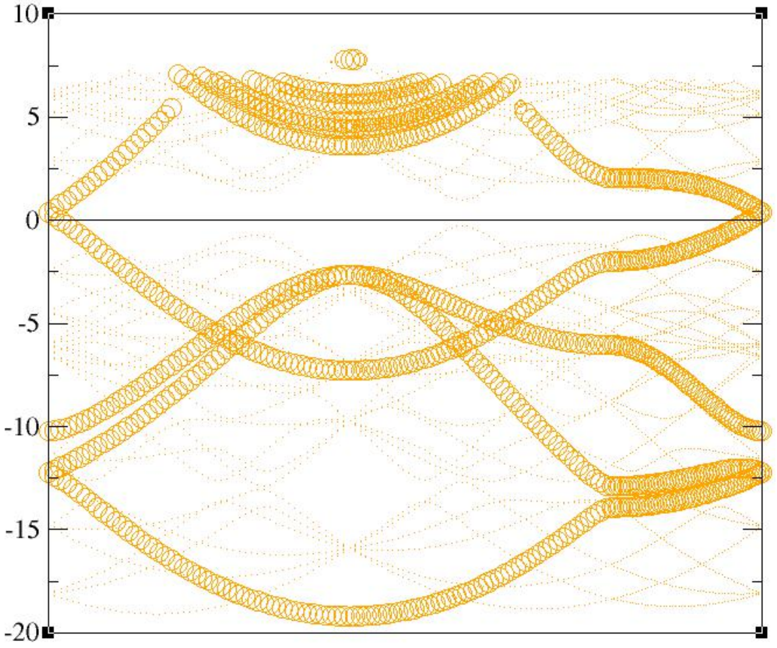
- Plotting use xmgrace

```
xmgrace bsplot.agr
```

This is the fastest method to view the unfolded bands.



folded bands



unfolded bands

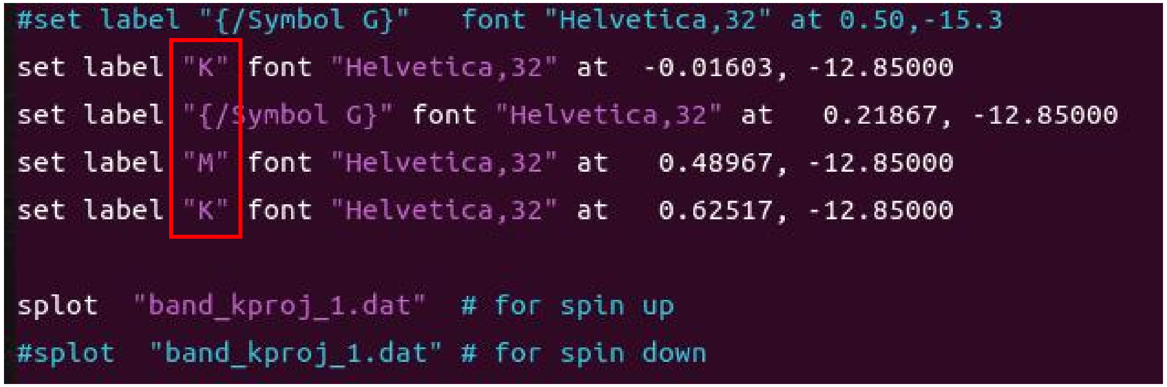
Example 1: $\sqrt{7} \times \sqrt{7}$ graphene

VASP/QE/Ab-init interface

Step 5 : Do the plotting

- Plotting use gnuplot

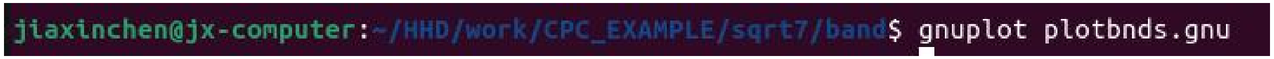
vim plotbnds.gnu



```
#set label "{/Symbol G}"   font "Helvetica,32" at 0.50,-15.3
set label "K" font "Helvetica,32" at  -0.01603, -12.85000
set label "{/Symbol G}" font "Helvetica,32" at   0.21867, -12.85000
set label "M" font "Helvetica,32" at   0.48967, -12.85000
set label "K" font "Helvetica,32" at   0.62517, -12.85000

splot  "band_kproj_1.dat"  # for spin up
#splot  "band_kproj_1.dat" # for spin down
```

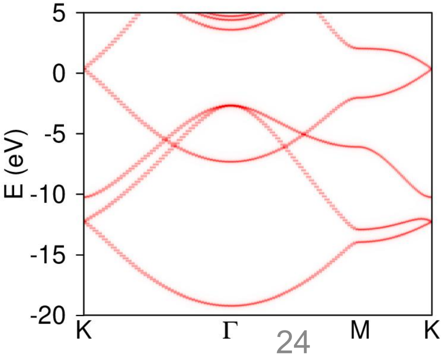
Give the name of the high symmetry k-points.

gnuplot plotbnds.gnu



```
jiaxinchen@jx-computer:~/HHD/work/CPC_EXAMPLE/sqrt7/band$ gnuplot plotbnds.gnu
```

Finally, generate the image file named **plotbnds_up(down).eps**.



Example 1: $\sqrt{7}\times\sqrt{7}$ graphene

VASP/QE/Ab-init interface

Step 5 : Do the plotting

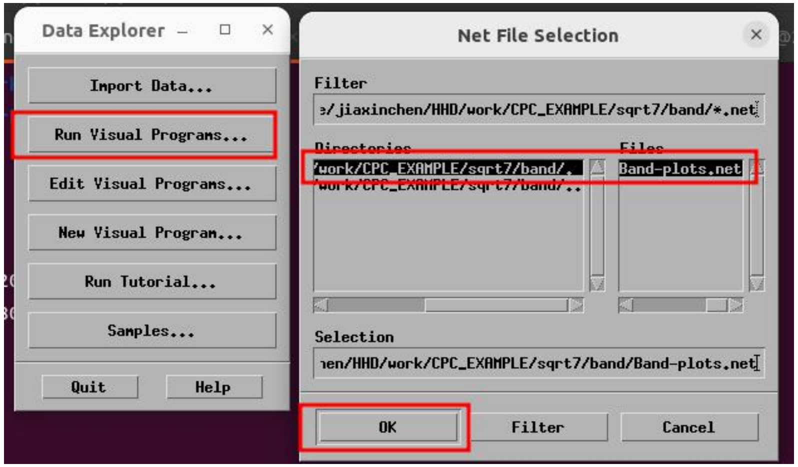
- Plotting use opendx

➢ First, copy **Band-plots.net** to the working directory.

```
cp /kproj/src/Band-plots.net    /work_directory
```

➢ run opendx

```
dx
```

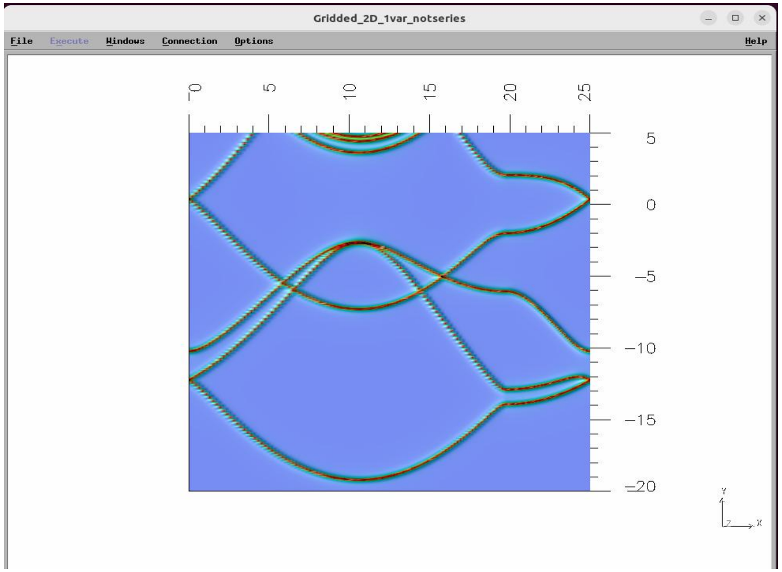


Filter: ɘ/jiaxinchen/HHD/work/CPC_EXAMPLE/sqrt7/band/*.net

Directories:
/work/CPC_EXAMPLE/sqrt7/band/.
/work/CPC_EXAMPLE/sqrt7/band/..

Files:
Band-plots.net

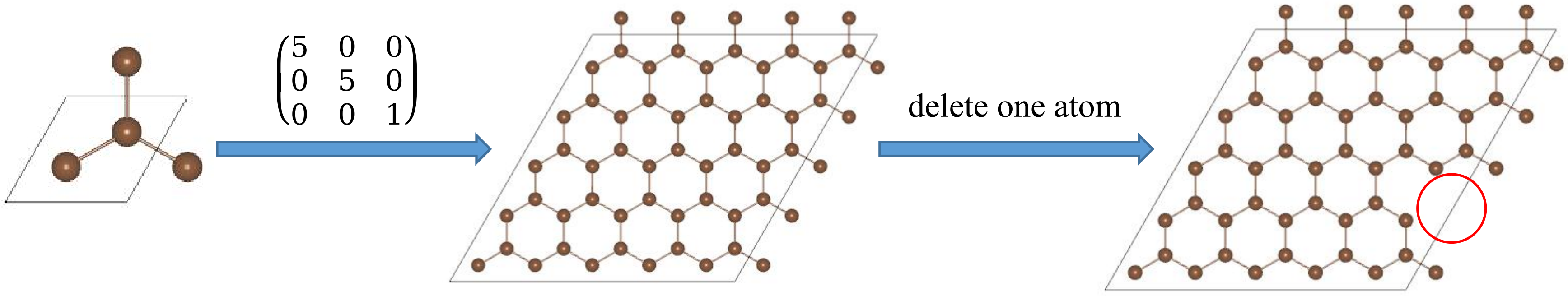
Selection: ıen/HHD/work/CPC_EXAMPLE/sqrt7/band/Band-plots.net



Example 2: 5 × 5 defected-Graphene

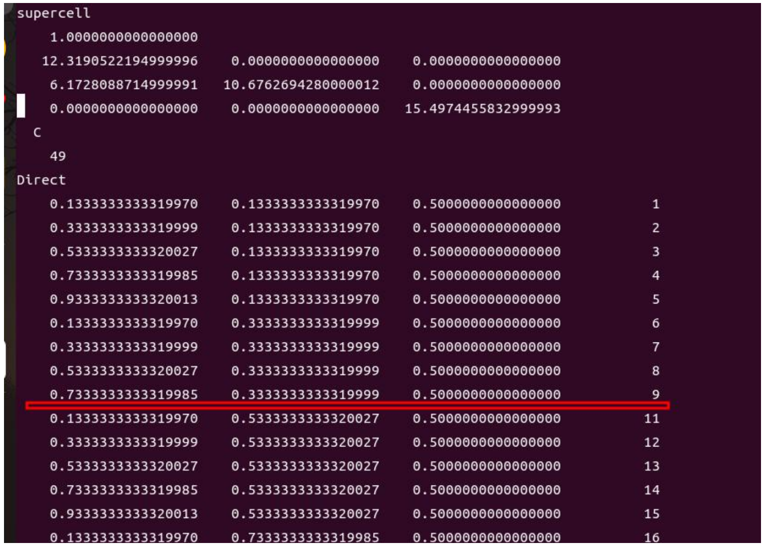
Phonopy and ABACUS interface

Step 1 : Generate the supercell.

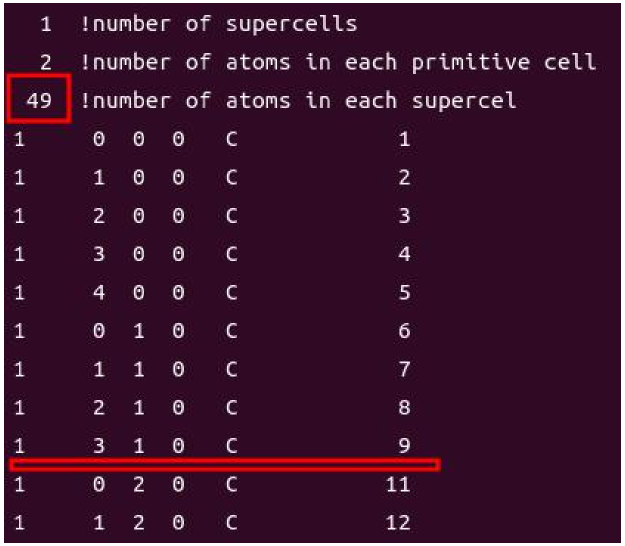


$$\begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

I deleted the 10-th atom in POSCAR_SUPER, so the corresponding translation vector for 10-th atom is also removed from the pos_map.dat file. Then, modify the total number of atoms of supercell to 49.



```
supercell
    1.0000000000000000
   12.3190522194999996    0.0000000000000000    0.0000000000000000
    6.1728088714999991   10.6762694280000012    0.0000000000000000
    0.0000000000000000    0.0000000000000000   15.4974455832999993
  C
    49
Direct
    0.1333333333319970    0.1333333333319970    0.5000000000000000          1
    0.3333333333319999    0.1333333333319970    0.5000000000000000          2
    0.5333333333320027    0.1333333333319970    0.5000000000000000          3
    0.7333333333319985    0.1333333333319970    0.5000000000000000          4
    0.9333333333320013    0.1333333333319970    0.5000000000000000          5
    0.1333333333319970    0.3333333333319999    0.5000000000000000          6
    0.3333333333319999    0.3333333333319999    0.5000000000000000          7
    0.5333333333320027    0.3333333333319999    0.5000000000000000          8
    0.7333333333319985    0.3333333333319999    0.5000000000000000          9
    0.1333333333319970    0.5333333333320027    0.5000000000000000         11
    0.3333333333319999    0.5333333333320027    0.5000000000000000         12
    0.5333333333320027    0.5333333333320027    0.5000000000000000         13
    0.7333333333319985    0.5333333333320027    0.5000000000000000         14
    0.9333333333320013    0.5333333333320027    0.5000000000000000         15
    0.1333333333319970    0.7333333333319985    0.5000000000000000         16
```



```
  1  !number of supercells
  2  !number of atoms in each primitive cell
 49  !number of atoms in each supercel
1      0  0  0   C            1
1      1  0  0   C            2
1      2  0  0   C            3
1      3  0  0   C            4
1      4  0  0   C            5
1      0  1  0   C            6
1      1  1  0   C            7
1      2  1  0   C            8
1      3  1  0   C            9
1      0  2  0   C           11
1      1  2  0   C           12
```

Example 2: 5 × 5 defected-Graphene

Phonopy and ABACUS interface

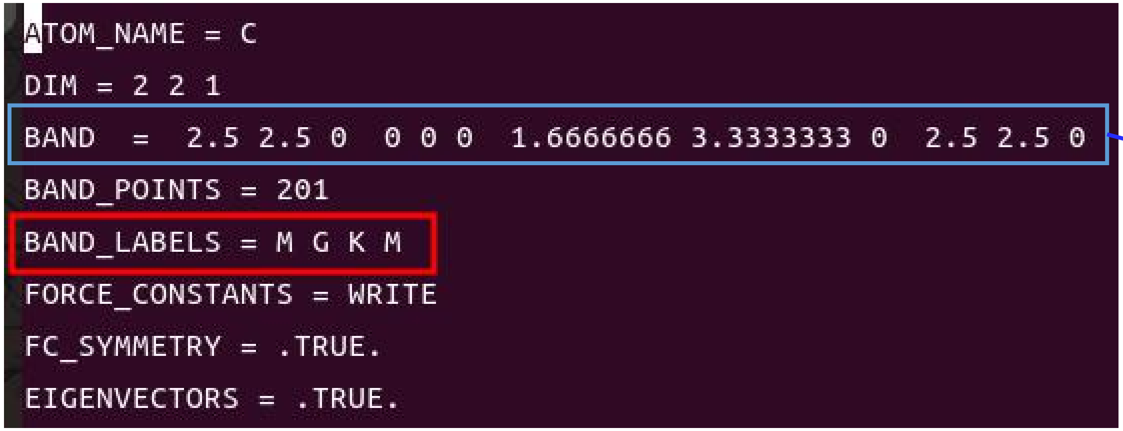
Step 2 : Do a band calculation to get the wavefunctions.

- For ABACUS calculation, we rename POSCAR_SUPER to POSCAR, and then run the pos2stru program to convert the structure to the ABACUS format. Then use it for band calculations.

  pos2stru is under the ~/kproj/utils/supercell/pos2stru folder.

- For Phonopy calculation, we rename POSCAR_SUPER to POSCAR. Then use it for phonon spectrum calculation.

  The **BAND_LABELS** must be provided during phonon calculations.



```
ATOM_NAME = C
DIM = 2 2 1
BAND  =  2.5 2.5 0  0 0 0  1.6666666 3.3333333 0  2.5 2.5 0
BAND_POINTS = 201
BAND_LABELS = M G K M
FORCE_CONSTANTS = WRITE
FC_SYMMETRY = .TRUE.
EIGENVECTORS = .TRUE.
```

kpoints used for band calculation

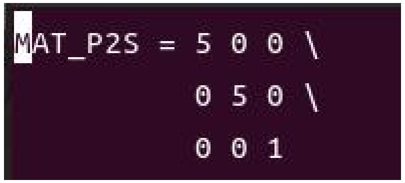
Example 2: 5 × 5 defected-Graphene

Phonopy and ABACUS interface

Step 3 : Do a KPROJ calculation.

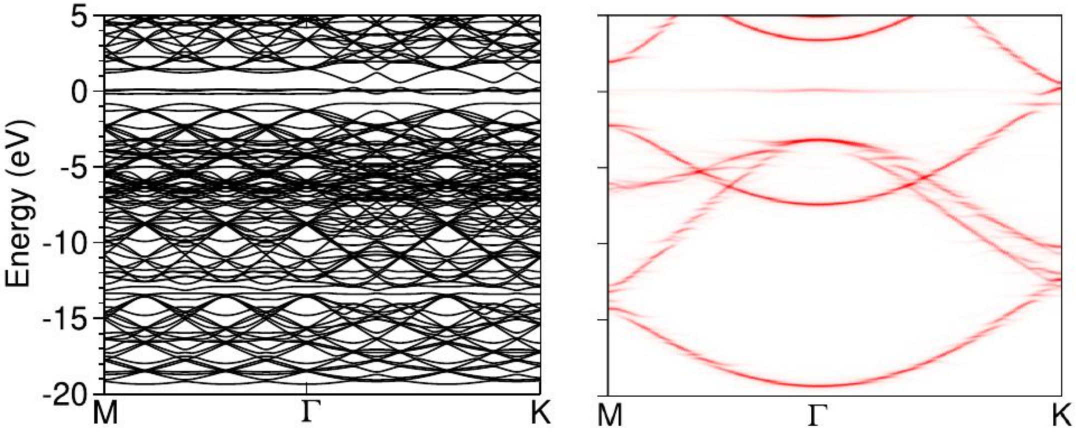
First, **copy pos_map.dat to the working directory**, prepare INKPROJ and then run ./kproj to perform the *k*-projection .

INKPROJ:

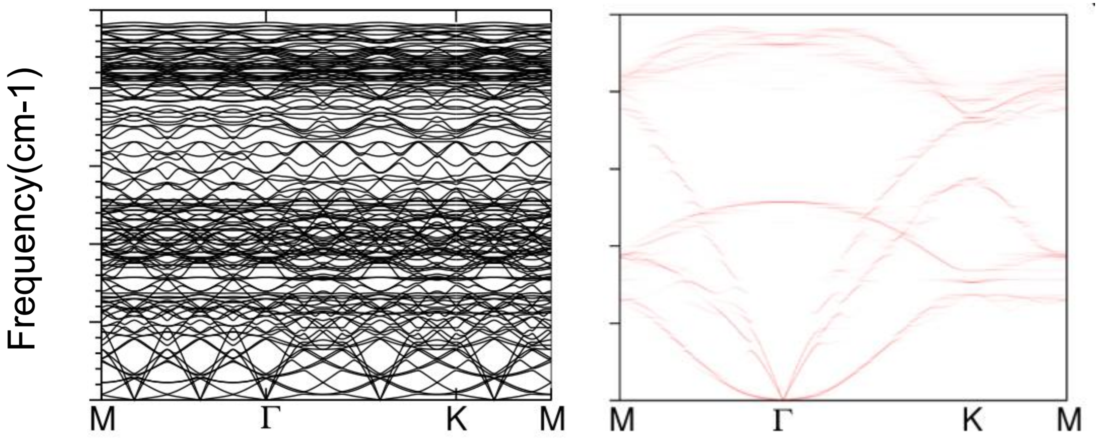


```
MAT_P2S = 5 0 0 \
          0 5 0 \
          0 0 1
```

ABACUS:



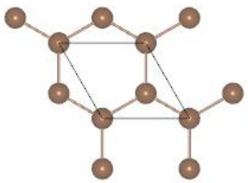
Phonopy:



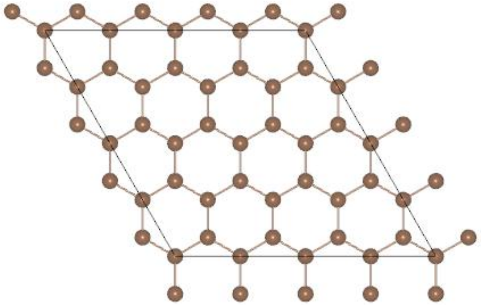
Example 3: 4 × 4 graphene/3 × 3 MoS_2 heterostructure

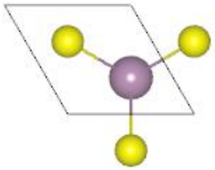
VAPS/QE/Ab-init interface

Step 1 : Build the structure for the heterostructure

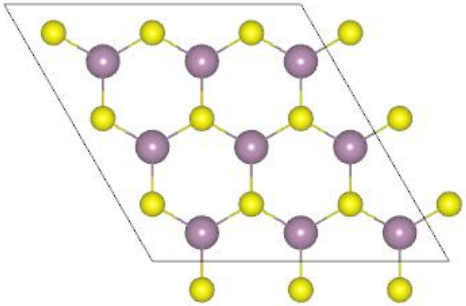


$$\xrightarrow{\begin{pmatrix}4 & 0 & 0\\ 0 & 4 & 0\\ 0 & 0 & 1\end{pmatrix}}$$

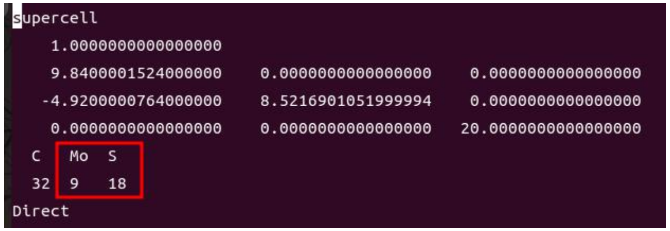




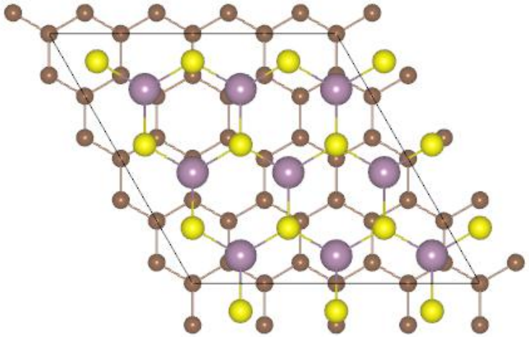
$$\xrightarrow{\begin{pmatrix}3 & 0 & 0\\ 0 & 3 & 0\\ 0 & 0 & 1\end{pmatrix}}$$



Copy the atomic coordinates of the MoS_2 supercell to the end of the graphene supercell, and add the corresponding atoms and their quantities at the head.



```
supercell
    1.0000000000000000
    9.8400001524000000    0.0000000000000000    0.0000000000000000
   -4.9200000764000000    8.5216901051999994    0.0000000000000000
    0.0000000000000000    0.0000000000000000   20.0000000000000000
  C   Mo   S
  32  9    18
Direct
```

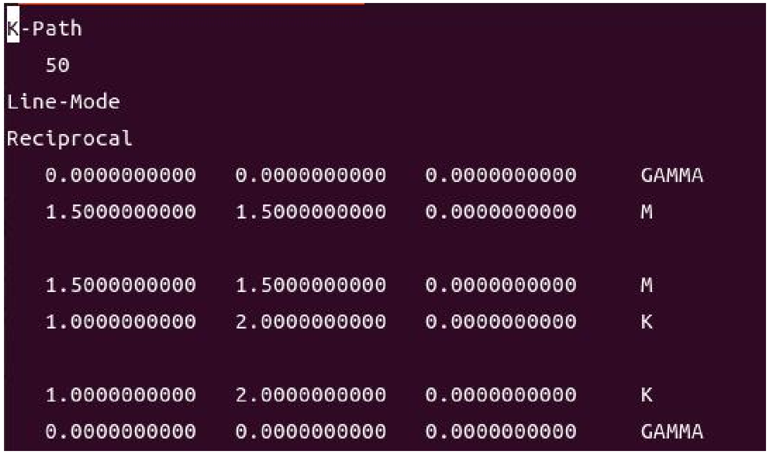


Example 3: 4 × 4 graphene/3 × 3 MoS$_2$ heterostructure

VASP/QE/Ab-init interface

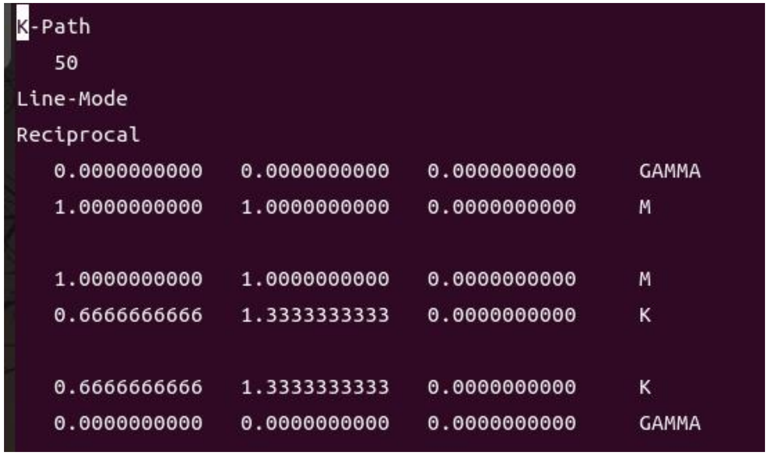
Step 2 : Band calculation

For projection onto the graphene, k-points specified in VASP are :



```
K-Path
   50
Line-Mode
Reciprocal
   0.0000000000   0.0000000000   0.0000000000     GAMMA
   1.5000000000   1.5000000000   0.0000000000     M

   1.5000000000   1.5000000000   0.0000000000     M
   1.0000000000   2.0000000000   0.0000000000     K

   1.0000000000   2.0000000000   0.0000000000     K
   0.0000000000   0.0000000000   0.0000000000     GAMMA
```

For projection onto MoS2, they are :

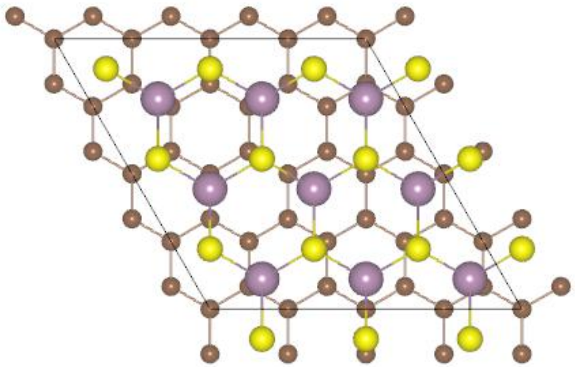


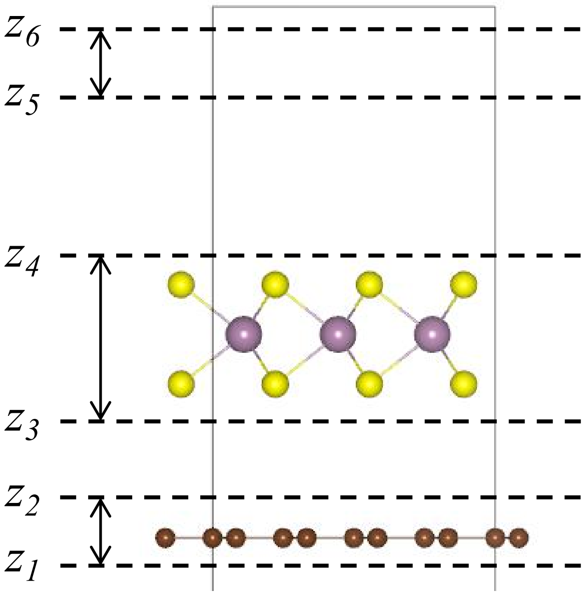
```
K-Path
   50
Line-Mode
Reciprocal
   0.0000000000   0.0000000000   0.0000000000     GAMMA
   1.0000000000   1.0000000000   0.0000000000     M

   1.0000000000   1.0000000000   0.0000000000     M
   0.6666666666   1.3333333333   0.0000000000     K

   0.6666666666   1.3333333333   0.0000000000     K
   0.0000000000   0.0000000000   0.0000000000     GAMMA
```

Example 3: 4 × 4 graphene/3 × 3 MoS_2 heterostructure

VASP/QE/Ab-init interface

Step 3 : Layer *k*-projection





INKPROJ for *k*-projection :

Projection onto graphene :

Mat_P2S = 4 0 0 \
0 4 0 \
0 0 1
LZLAYER = .TRUE.
zlay1 = z_1
zlay2 = z_2

Projection onto MoS_2 :

Mat_P2S = 3 0 0 \
0 3 0 \
0 0 1
LZLAYER = .TRUE.
zlay1 = z_3
zlay2 = z_4

Projection in the vacuum :

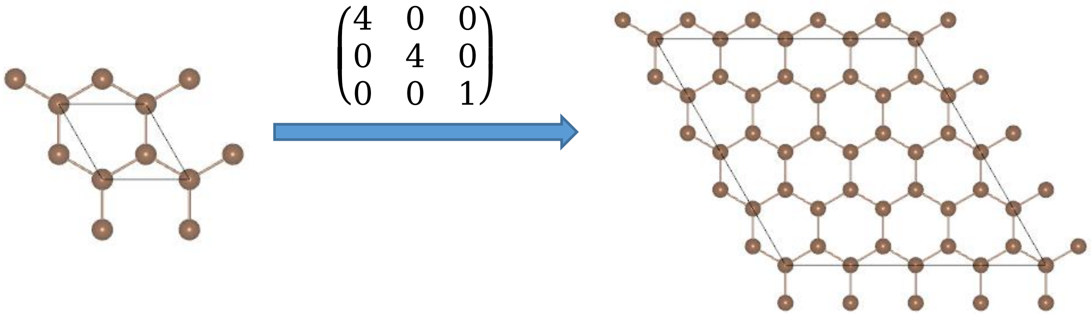
Mat_P2S = n 0 0 \
0 n 0 \
0 0 1
LZLAYER = .TRUE.
zlay1 = z_5
zlay2 = z_6

n=3 for MoS_2
n=4 for Graphene

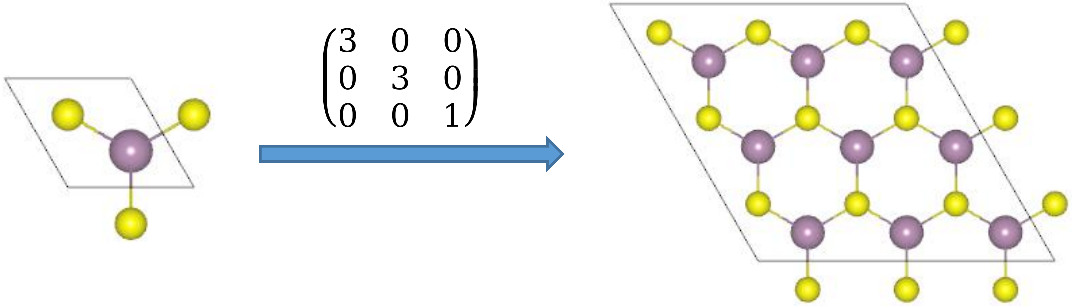
Example 4: 4 × 4 graphene/3 × 3 MoS_2 heterostructure

Note 6 : ABACUS interface

Step 1 : Generate the supercell.

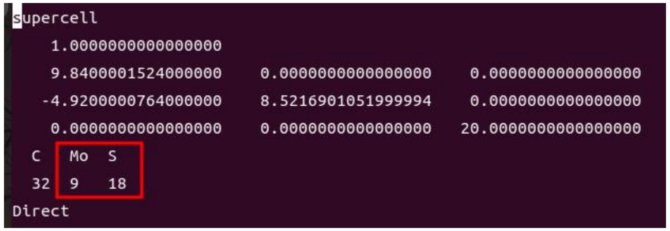


$$\xrightarrow{\begin{pmatrix}4 & 0 & 0\\ 0 & 4 & 0\\ 0 & 0 & 1\end{pmatrix}}$$

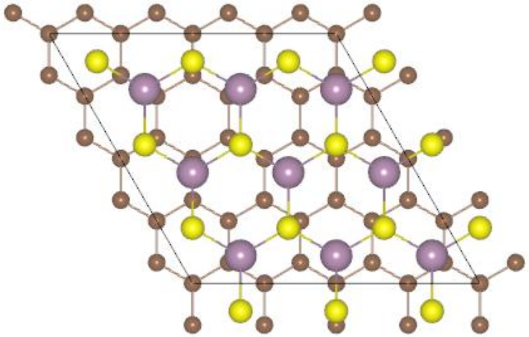


$$\xrightarrow{\begin{pmatrix}3 & 0 & 0\\ 0 & 3 & 0\\ 0 & 0 & 1\end{pmatrix}}$$

Copy the atomic coordinates of the MoS_2 supercell to the end of the graphene supercell, and add the corresponding atoms and their quantities at the head.



```
supercell
    1.0000000000000000
    9.8400001524000000    0.0000000000000000    0.0000000000000000
   -4.9200000764000000    8.5216901051999994    0.0000000000000000
    0.0000000000000000    0.0000000000000000   20.0000000000000000
  C   Mo   S
  32  9    18
Direct
```

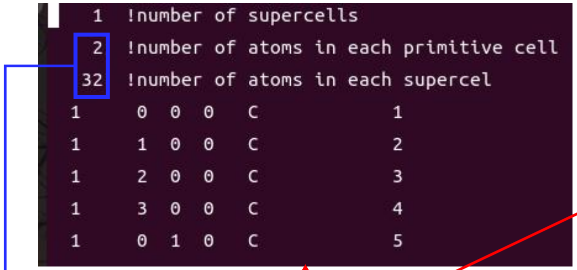


Example 4: 4 × 4 graphene/3 × 3 MoS_2 heterostructure

ABACUS interface

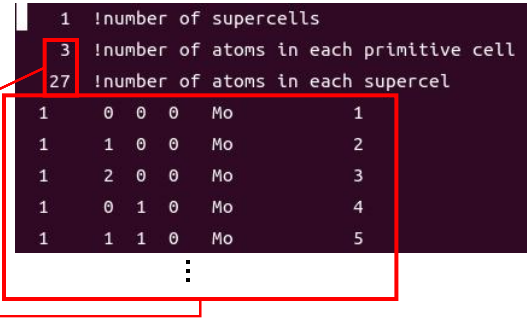
Step 2 : Merge the pos_map.dat files

pos_map.dat of graphene



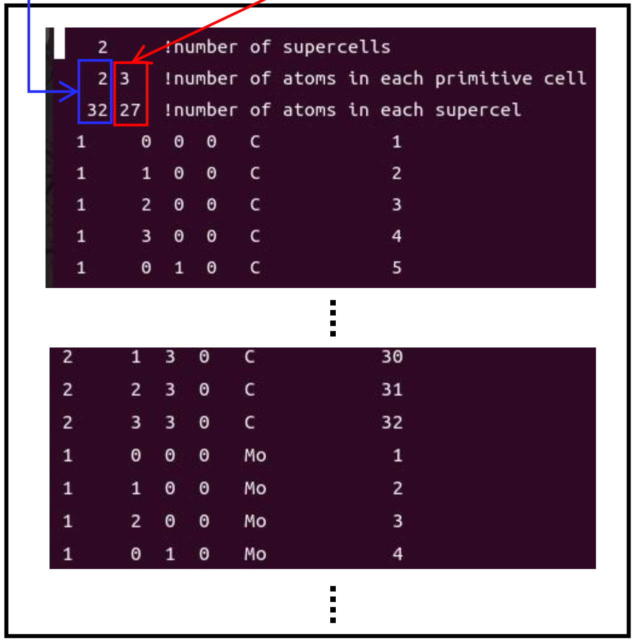
```
  1  !number of supercells
  2  !number of atoms in each primitive cell
 32  !number of atoms in each supercel
 1     0  0  0   C            1
 1     1  0  0   C            2
 1     2  0  0   C            3
 1     3  0  0   C            4
 1     0  1  0   C            5
```

pos_map.dat of MoS_2



```
  1  !number of supercells
  3  !number of atoms in each primitive cell
 27  !number of atoms in each supercel
 1     0  0  0   Mo           1
 1     1  0  0   Mo           2
 1     2  0  0   Mo           3
 1     0  1  0   Mo           4
 1     1  1  0   Mo           5
          ⋮
```

pos_map.dat for heterostructure



```
  2    !number of supercells
  2 3  !number of atoms in each primitive cell
 32 27 !number of atoms in each supercel
 1     0  0  0   C            1
 1     1  0  0   C            2
 1     2  0  0   C            3
 1     3  0  0   C            4
 1     0  1  0   C            5
               ⋮
2      1  3  0   C            30
2      2  3  0   C            31
2      3  3  0   C            32
1      0  0  0   Mo           1
1      1  0  0   Mo           2
1      2  0  0   Mo           3
1      0  1  0   Mo           4
               ⋮
```

Copy the translation vectors from the pos_map.dat of MoS_2 to the end of the graphene . Then, modify the head. This heterostructure consists of **2** supercells with **2** and **3** atoms in corresponding primitive cells, and **32 and 27** atoms in the supercells for graphene and MoS_2, respectively.

In this case, according to the writing order of **pos_map.dat**, graphene is defined as the 1th layer, and MoS_2 is named the 2th layer

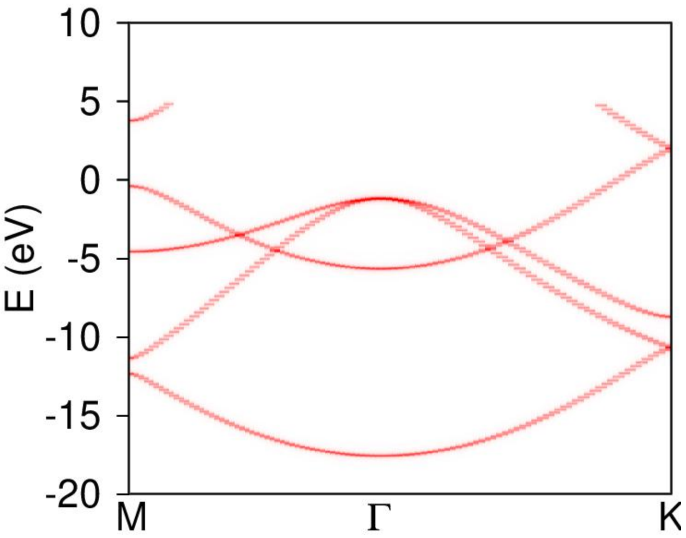
Example 4: 4 × 4 graphene/3 × 3 MoS_2 heterostructure

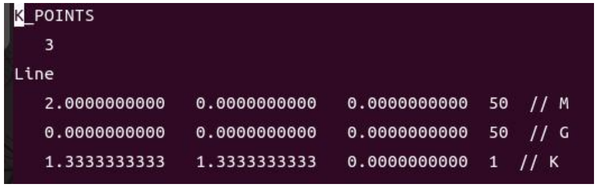
ABACUS interface

Step 3 : Do a KPROJ calculation.

Project to graphene :

```
Mat_P2S = 4 0 0 \
          0 4 0 \
          0 0 1
proj_part = 1
```

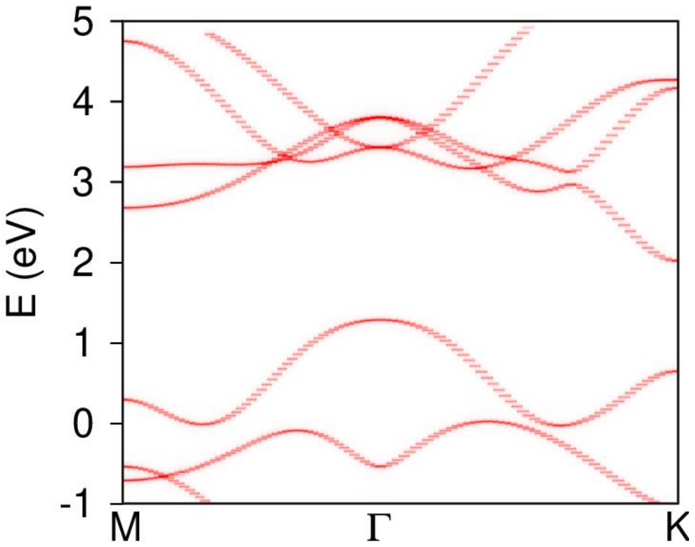


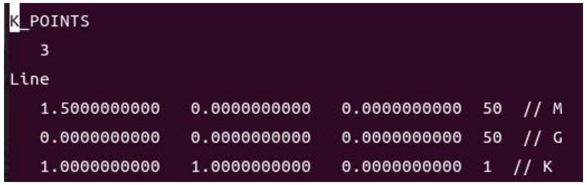


```
K_POINTS
   3
Line
   2.0000000000   0.0000000000   0.0000000000   50   // M
   0.0000000000   0.0000000000   0.0000000000   50   // G
   1.3333333333   1.3333333333   0.0000000000   1   // K
```

Project to MoS_2 :

```
Mat_P2S = 3 0 0 \
          0 3 0 \
          0 0 1
proj_part = 2
```





```
K_POINTS
   3
Line
   1.5000000000   0.0000000000   0.0000000000   50   // M
   0.0000000000   0.0000000000   0.0000000000   50   // G
   1.0000000000   1.0000000000   0.0000000000   1   // K
```

KPROJ for large systems

Step 1: Build the supercell use ***supercell*** program

Step 2: Band calculations (split the calculations into a few jobs)

split KPOINTS into several files using ***ksplit*** (~/kproj/utils/)

do the band calculations using KPOINTS_i

Step 3: Do kproj calculations to obtain **bs_projected.dat**. Rename the **bs_projected.dat** file into **bs_projected.dat_i**, which follows the order of KPOINTS_i.

Step 4: Merge multiple **bs_projected.dat** files (bs_projected.dat_i) into one **bs_projected.dat** file using progam ***mergefile***.

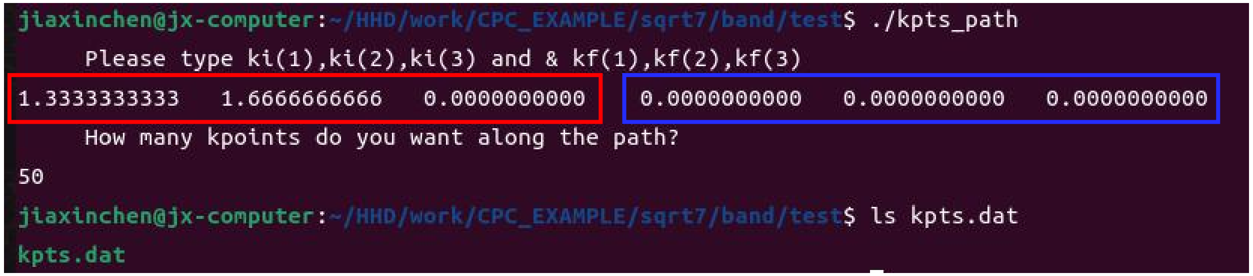
Step 5: Adjust the Fermi level in the first line of the merged **bs_projected.dat**.

Step 6: ***Run_futils*** to get the files for plotting.

Step 7: Plotting.

Note 9 : Split the K-path

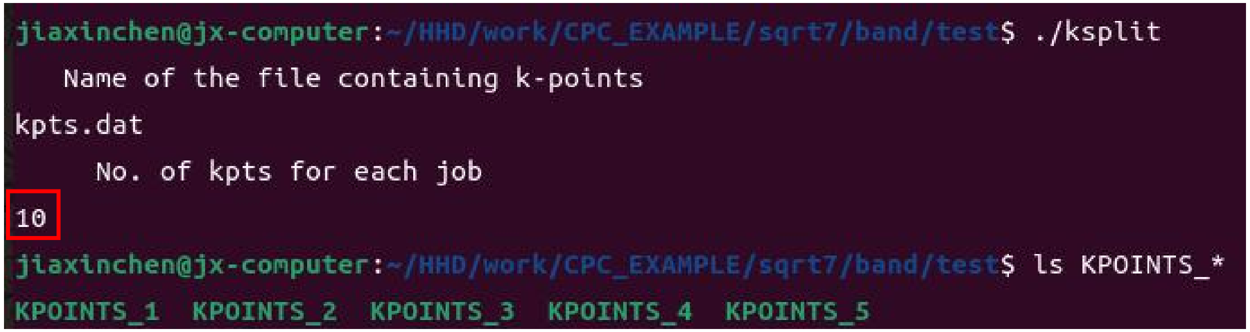
Step 1: Use ***kpt_path*** to generate the *k*-points.



```
jiaxinchen@jx-computer:~/HHD/work/CPC_EXAMPLE/sqrt7/band/test$ ./kpts_path
     Please type ki(1),ki(2),ki(3) and & kf(1),kf(2),kf(3)
1.3333333333   1.6666666666   0.0000000000   0.0000000000   0.0000000000   0.0000000000
     How many kpoints do you want along the path?
50
jiaxinchen@jx-computer:~/HHD/work/CPC_EXAMPLE/sqrt7/band/test$ ls kpts.dat
kpts.dat
```

Generate **50** *k*-points along the path from $(\frac{4}{3}, \frac{5}{3}, 0)$ to $(0, 0, 0)$ and store them in file **kpts.dat**.

Step 2: Use ***ksplit*** to split the path into several files.



```
jiaxinchen@jx-computer:~/HHD/work/CPC_EXAMPLE/sqrt7/band/test$ ./ksplit
   Name of the file containing k-points
kpts.dat
     No. of kpts for each job
10
jiaxinchen@jx-computer:~/HHD/work/CPC_EXAMPLE/sqrt7/band/test$ ls KPOINTS_*
KPOINTS_1  KPOINTS_2  KPOINTS_3  KPOINTS_4  KPOINTS_5
```

Split the **kpts.dat** into **5** KPOINTS files with **10** *k*-points in each file.

Note 10 : Merge bs_projected.dat

If you have multiple bs_projecte.dat, please merge them into one first.

- You need copy them to a new working directory and rename them like bs_projecte.dat_1, bs_projecte.dat_2, ...
- User program ***mergefile*** to do that. You can get it by "make mergefile" under src.
- An example is given under ~/kproj/example/vasp_examples/**Ge_Al2O3**.

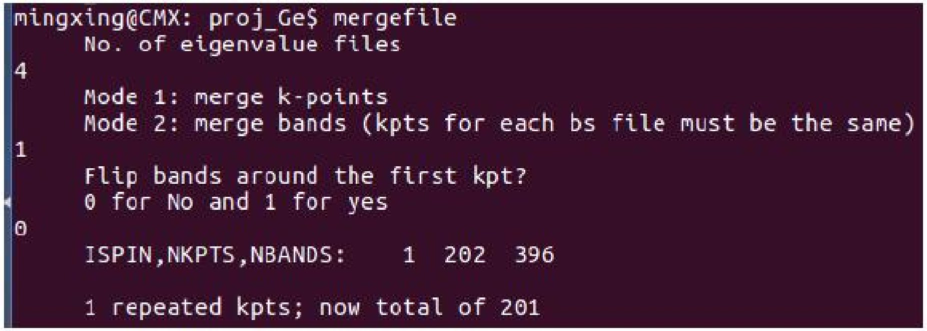
I split the calculation into four sub-calculations. Two are for the *k*-path along M-Γ and the others are for the *k*-path along Γ-K.

After KPROJ, each calculation gives me one bs_projected.dat.

Then, I make a new working directory **merge_bs**. After that, I copy them to **merge_bs:**

"cp MG/CAL1/bs_projected.dat merge_bs/bs_projected.dat_1" and
"cp MG/CAL2/bs_projected.dat merge_bs/bs_projected.dat_2", ...

Then, I merge them into one **bs_projected.dat**.



```
mingxing@CMX: proj_Ge$ mergefile
          No. of eigenvalue files
4
          Mode 1: merge k-points
          Mode 2: merge bands (kpts for each bs file must be the same)
1
          Flip bands around the first kpt?
          0 for No and 1 for yes
0
          ISPIN,NKPTS,NBANDS:    1  202  396

          1 repeated kpts; now total of 201
```

Note that E-fermi has to be set correctly (Using the one from the scf calculation).

Reference

[1] M. Chen and M. Weinert, Phys. Rev. B 98, 245421 (2018).

[2] M. X. Chen and M. Weinert, Nano Letters 14, 5189 (2014).

[3] M. X. Chen and M. Weinert, Phys. Rev. B 94, 035433 (2016).

[4] M. X. Chen, Z. Zhong, and M. Weinert, Phys. Rev. B 94, 075409 (2016).

[5] M. X. Chen, Z. Ge, Y. Y. Li, D. F. Agterberg, L. Li, and M. Weinert, Phys. Rev. B 94, 245139 (2016).

[6] M. X. Chen, W. Chen, Z. Zhang, and M. Weinert, Phys. Rev. B 96, 245111 (2017).

[7] S. Rajput, M. X. Chen, Y. Liu, Y. Y. Li, M. Weinert, and L. Li, Nature Communications 4, 2752 (2013).

[8] Y. Y. Li, M. X. Chen, M. Weinert, and L. Li, Nature Communications 5, 4311 (2014).

[9] W. Zhang, M. Chen, J. Dai, X. Wang, Z. Zhong, S. W. Cheong, and W. Wu, Nano Letters 18, 2677 (2018)

[10] Jiaxin Chen, Mingxing Chen*, Progress in Physics 25, 43(2023).

[11] Jiaxin Chen, Mingxing Chen*, arXiv:2410.10910.