1 Introduction

chn is a task of application level. It calculate the Chern number of the system. This task uses the method introduced by JPSJ 74.1674 (2005). Since this method only applies for type-A[1] Chern insulator, it only works to 2D system. Because the calculated results are too large, not all data are attached in this file. Also, since NiO is not a 2D structure, so the Haldane_chn.plb (can be found in examples) is used instead. This code cannot determine whether the band structure if an insulator or not, so one should confirm it before calculating Chern number.

2 Dictionary

2.1 Input

chn.Mesh This parameter tells PiLab how to mesh the k-space. The method of this code usually doesn't require very dense grids. So you can gradually increase this parameter form small values, say [5,5], to see if the the Chern number converges.

chn.DiffVal This parameter tells PiLab how to avoid divergent k-point. In the calculation method, it is possible to have lattice field divergent even if there is no degeneracy. To avoid this issue, one should input a small difference $\mathbf{k} \rightarrow \mathbf{k} + \delta \cdot \mathbf{k}^{'}$ to avoid the divergence,where δ is a small value and $\mathbf{k}^{'}$ is an arbitrary vector. Here, DiffVal gives δ and DiffVec gives $\mathbf{k}^{'}$.

chn.DiffVec See chn.DiffVal

2.2 Output

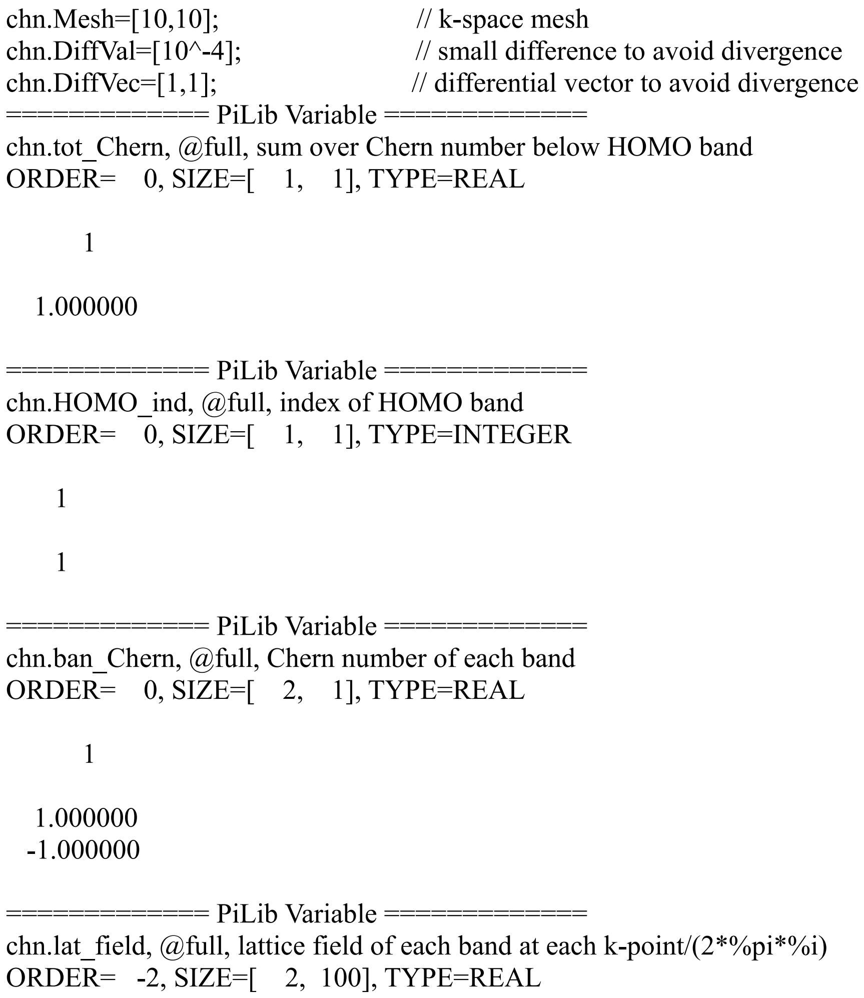
chn.tot_Chern This variable shows the total Chern number below Fermi level. It must be an integer.

chn.HOMO_ind This variable shows which band is the highest occupied band.

chn.ban_Chern This variable shows the Chern number of each band.

chn.lat_field This variable is the lattice field of each band at each k-point.If there are 10 bands with 25 k-points, this variable will be a 10x25 matrix. The lattice field is a quantity in this method. It can be regarded as the sum of the Chern number of a grid in the k-space contributed form a particular band.

[1] For definition, see the periodic table of topological insulators and superconductors shown in Rev. Mod. Phys. 82 3045



```
chn.Mesh=[10,10];                          // k-space mesh
chn.DiffVal=[10^-4];                       // small difference to avoid divergence
chn.DiffVec=[1,1];                         // differential vector to avoid divergence
============= PiLib Variable =============
chn.tot_Chern, @full, sum over Chern number below HOMO band
ORDER=    0, SIZE=[    1,    1], TYPE=REAL

          1

   1.000000

============= PiLib Variable =============
chn.HOMO_ind, @full, index of HOMO band
ORDER=    0, SIZE=[    1,    1], TYPE=INTEGER

       1

       1

============= PiLib Variable =============
chn.ban_Chern, @full, Chern number of each band
ORDER=    0, SIZE=[    2,    1], TYPE=REAL

          1

   1.000000
  -1.000000

============= PiLib Variable =============
chn.lat_field, @full, lattice field of each band at each k-point/(2*%pi*%i)
ORDER=   -2, SIZE=[    2,  100], TYPE=REAL
```

Figure 1: Haldane_chn.plb