1 Introduction

flq_ban is a task of extension level. It calculates the Floquet band structure. To use this task, you must first perform **flq** first. All of the keywords are the same with **ban**, so one should reference **ban** for details. Also, the attached output data are not complete because they are too large. Only the variable names are attached.

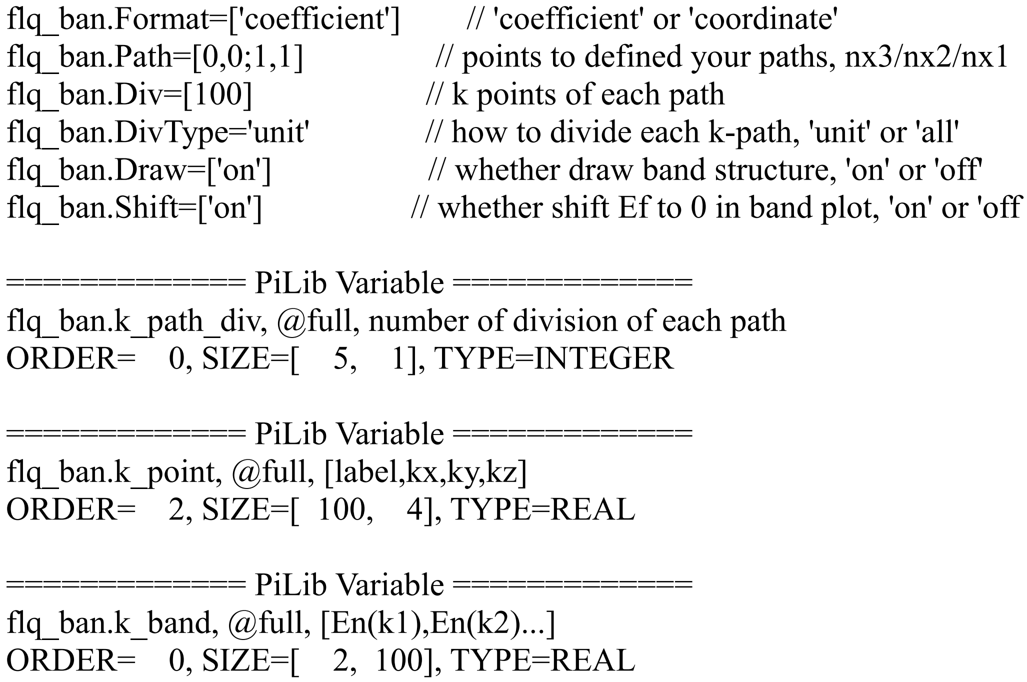
2 Dictionary

2.1 Input

flq_ban shares exactly the same keywords as **ban**. Therefore, one can directly reference the dictionary of **ban**

2.2 Output

flq_ban shares exactly the same keywords as **ban**. Therefore, one can directly reference the dictionary of **ban**



```
flq_ban.Format=['coefficient']        // 'coefficient' or 'coordinate'
flq_ban.Path=[0,0;1,1]                // points to defined your paths, nx3/nx2/nx1
flq_ban.Div=[100]                     // k points of each path
flq_ban.DivType='unit'                // how to divide each k-path, 'unit' or 'all'
flq_ban.Draw=['on']                   // whether draw band structure, 'on' or 'off'
flq_ban.Shift=['on']                  // whether shift Ef to 0 in band plot, 'on' or 'off

============= PiLib Variable =============
flq_ban.k_path_div, @full, number of division of each path
ORDER=    0, SIZE=[    5,    1], TYPE=INTEGER

============= PiLib Variable =============
flq_ban.k_point, @full, [label,kx,ky,kz]
ORDER=    2, SIZE=[  100,    4], TYPE=REAL

============= PiLib Variable =============
flq_ban.k_band, @full, [En(k1),En(k2)...]
ORDER=    0, SIZE=[    2,  100], TYPE=REAL
```

Figure 1: page 1 of Haldane_flq_ban.plb