Introduction of the band sorting code

Original paper:

Band sorting based on global continuity of eigenvalues and Berry curvature of phononic bands

Requirement:

1. A computer with Windows/Linux/MacOS operating system
2. The required Python packages have been installed. (Anaconda3 is recommended)

[Download Anaconda Distribution | Anaconda](https://www.anaconda.com/download/)

1. Jupyter notebook installed.

Files menu:

* Band processing codes

1. band sorting-eigenvalue\_derived\_quantities\_focused.ipynb
2. band sorting-eigenvector\_derived\_quantities\_focused.ipynb

* Input data

1. frequency-this\_example-entire\_FBZ.txt
2. displacement-this\_example-entire\_FBZ.txt
3. frequency-this\_example-IBZ.txt
4. basic\_info\_this\_example-IBZ.txt
5. kpoints\_x\_this\_example-IBZ.txt
6. kpoints\_y\_this\_example-IBZ.txt

* Auxiliary tools

1. 2D\_triangle\_k\_mesh\_generator.ipynb
2. bands\_visualization.nb

Tutorial:

1. **overview**

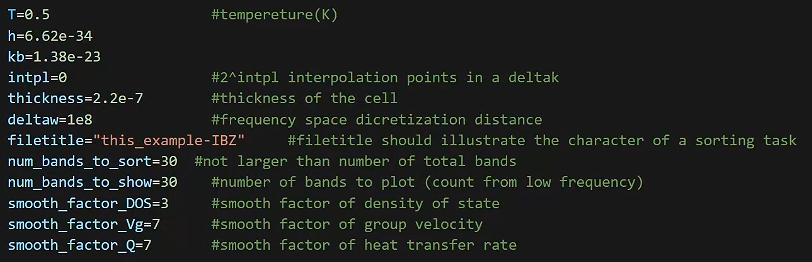
We provide a semiautomatic 2D Band sorting code based on the global continuity of eigenvalues with manual sweeping strategies and neighbor connection strategies. To reduce the difficulty of obtaining simulation results and the size of input data, the band sorting codes are divided into two parts: band\_sorting-eigenvalue\_derived\_quantities\_focused.ipynb and band\_sorting-eigenvector\_derived\_quantities\_focused.ipynb. The former inputs only eigenvalue data (such as frequency-this\_example-IBZ.txt), sorts the eigenvalue surfaces on 2D triangular irreducible Brillouin zone and derives density of states, group velocity and heat transfer rate (if the input data is phonon band), the latter input both eigenvalue data (such as frequency-this\_example-entire\_FBZ.txt) and eigenvector data (such as displacement-this\_example-entire\_FBZ.txt), sorts the eigenvalue surfaces on 2D rectangle first Brillouin zone, derives Berry connection and Berry curvature. Besides the band sorting codes, two auxiliary tools 2D\_\_triangle\_k\_mesh\_generator.ipynb and bands\_visualization.nb is provided for generating cartesian k-mesh in a triangular domain and flipping the eigenvalue surfaces.

1. **Input parameters and of**

**band\_sorting-eigenvalue\_derived\_quantities\_focused.ipynb**

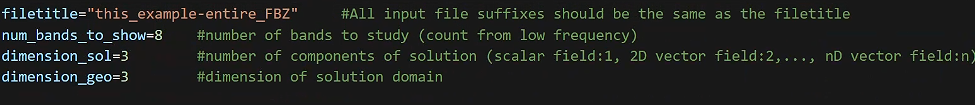
* filetitle: The mark of input data file. The input eigenfrequency data file names “frequency-filetitle”, while the input eigenvector data file names “displacement-filetitle”. This way of management is convenient for dealing with a series of band structures.
* num\_band\_to\_sort: Only this number of eigenvalues counting from low to high are processed by the band sorting code.
* num\_band\_to\_show: The code includes a preview of the result and an output function. This parameter specifies the number of eigenvalue surfaces that need to be displayed and output.
* smooth\_factor: Used for smoothing the curves of derived quantities such as group velocity.
* deltaw: Discretization accuracy of frequency, for deriving quantities such as group velocity.

Other parameters are constants related to the physical system, here is a 2D phononic crystal.



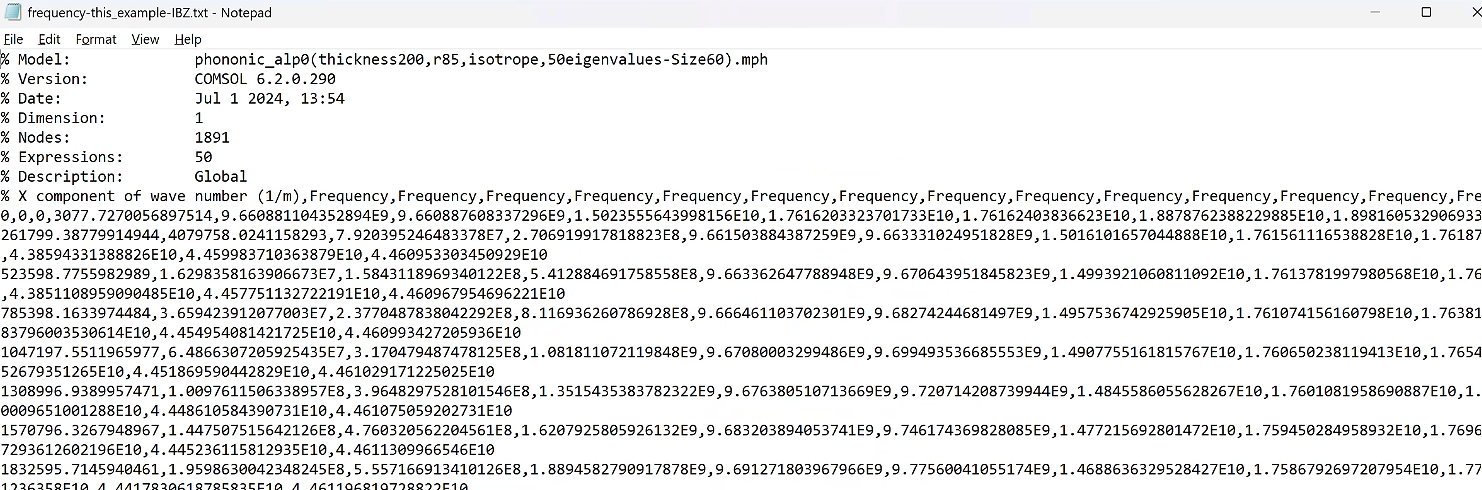
1. **Input parameters of band\_sorting-eigenvalue\_derived\_quantities\_focused.ipynb**

* filetitle: As above.
* num\_band\_to\_show: As above. Number of bands to sort is also set to this value.
* dimension\_sol: Number of dimensions of each component of the eigenvectors. For scalar field such as photonic intensity fields, acoustic pressure fields and electron wave function, dimension\_sol=1; For vector field such as 3D displacement field, dimension\_sol=3; for nD vector fields, dimension\_sol=n.
* dimension\_geo: Number of dimensions of the eigenvectors.



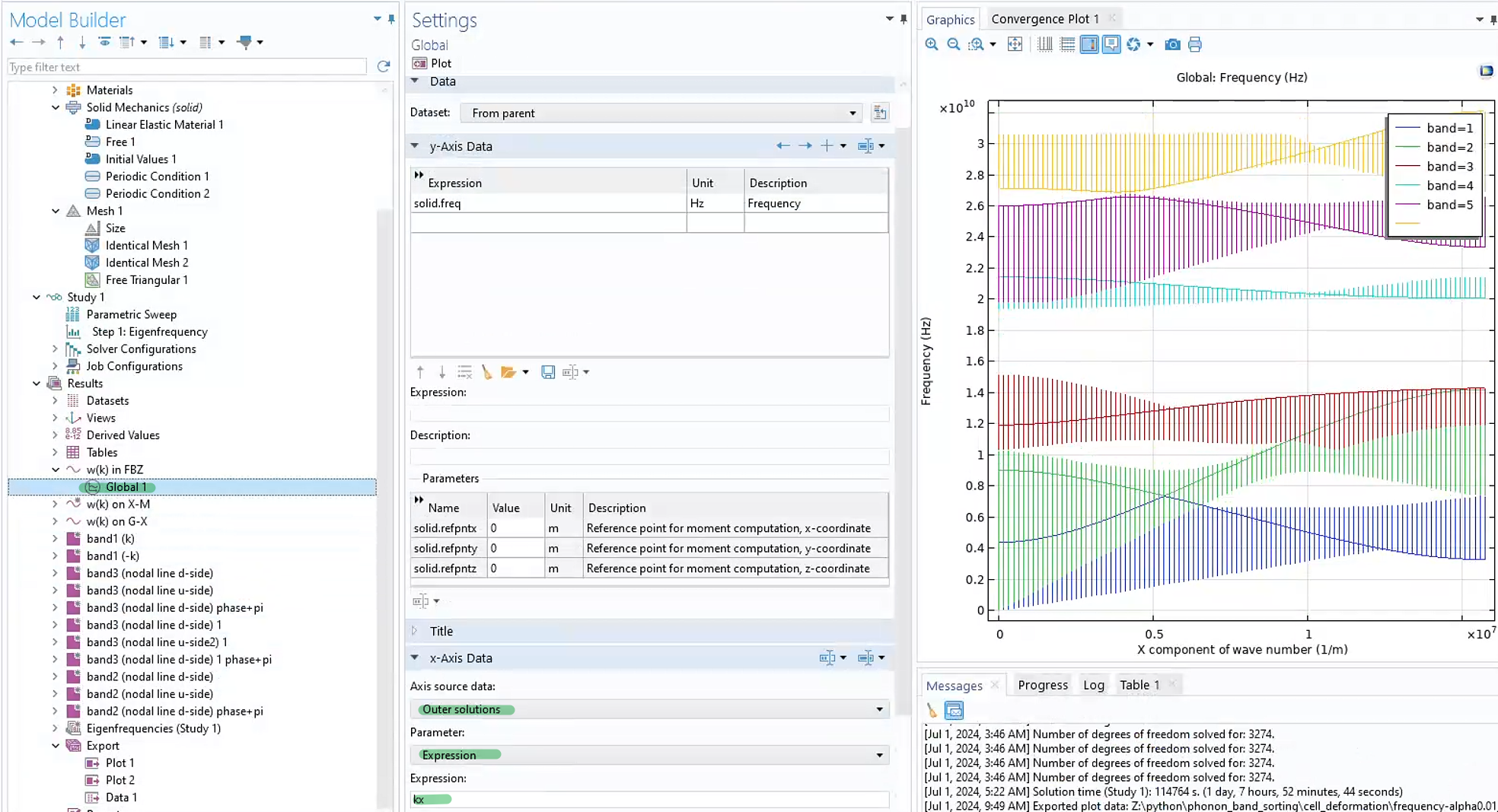
1. **Format of input eigenvalue data**

The format of frequency-this\_example-IBZ.txt is shown as follow:

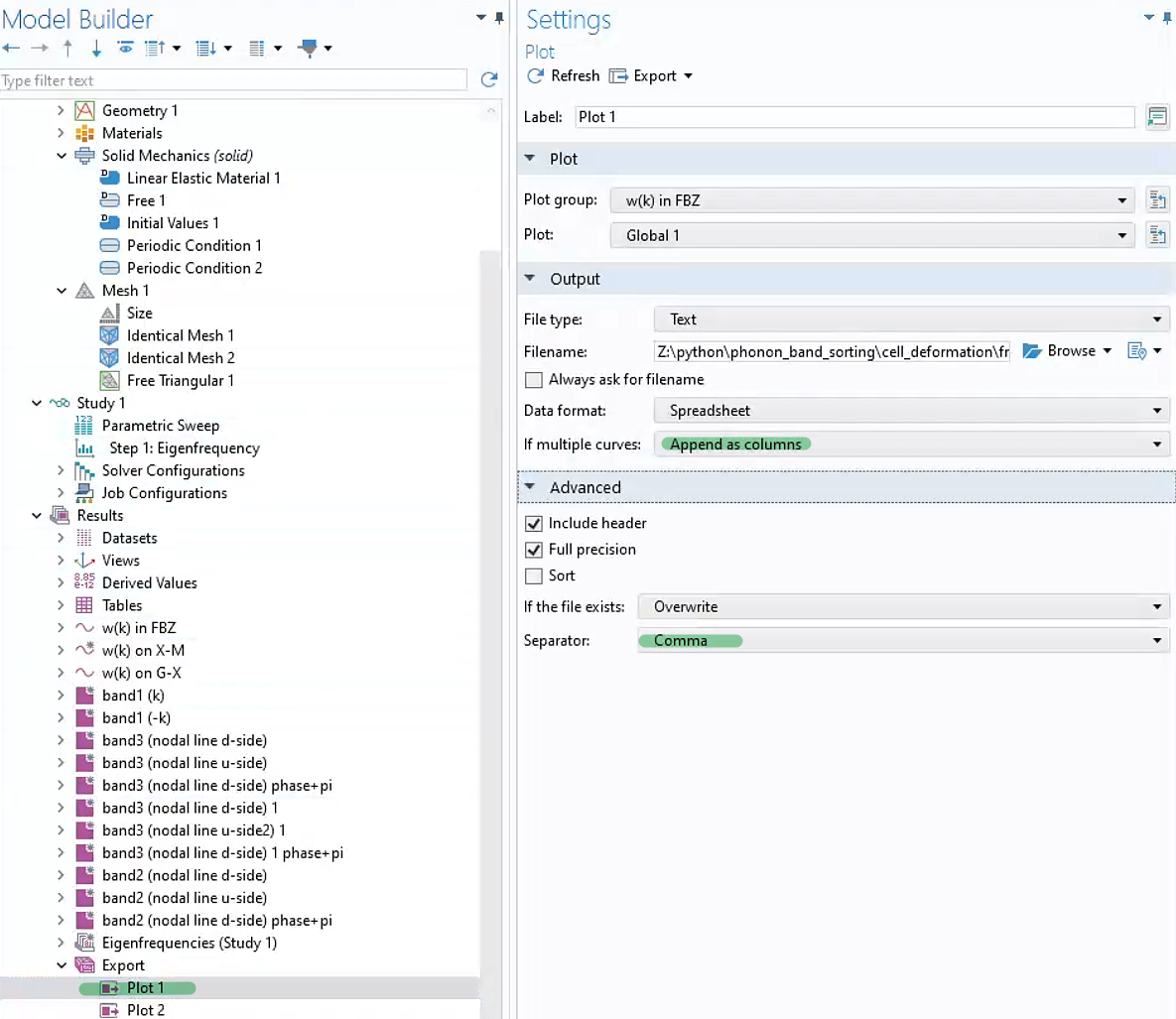


It’s output by Comsol Multiphysics software with the following steps:

* Plot the eigenvalues with x-axis data “outer solutions” and selecting one of the coordinate axis of the parameter space.



* Export the plot data to .txt file. Append multiple eigenvalue surfaces as columns, and separate as comma. Then the .txt file satisfies the input format.



If using other tools to obtain the eigenvalues, the input code should be modified. Anyway, the input eigenvalue data eventually has the form:

data[][][]

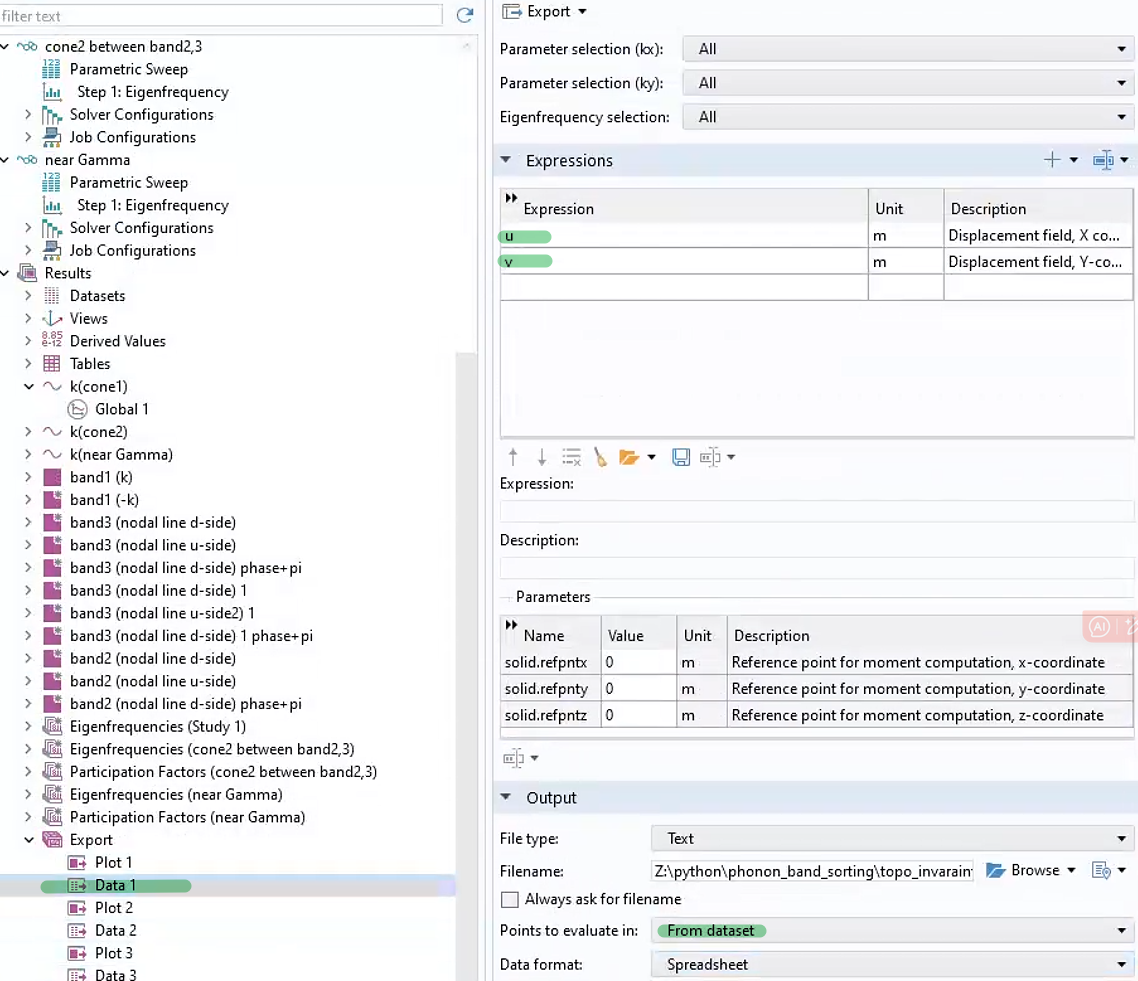
where and are the two dimensions of the parameter space, and they are arranged in cartesian grid, like kx, ky in this example. band\_sorting-eigenvector\_derived\_quantities\_focused.ipynb requires the parameter space is a rectangle, while band\_sorting-eigenvalue\_derived\_quantities\_focused.ipynb doesn’t require the parameter space is a rectangle, like the IBZ in this example is a triangle. But in such case, some vectorized packages like numpy are not recommended to use in the later development because data[band index] doesn’t have a uniform shape.

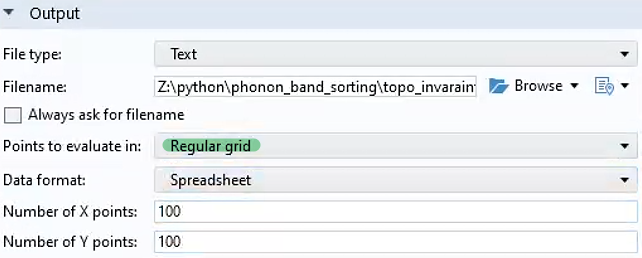
1. **Format of input eigenvector data**

The eigenvector input file displacement-this\_example-IBZ.txt is also output by dataset export function in Comsol Multiphysics software:

Select needed eigenvectors in the dataset to output. If the eigenfield is a vector field, list the components in “Expression” chart. If a uniform mesh is used to calculate the eigenvectors, the discretized eigenvectors can be directly extracted from the dataset (Mesh nodes or Gauss points). If the mesh is ununiform, the discretized eigenvectors should be extracted from a regular grid. For general cases, using an unstructured uniform mesh to calculate the eigenvectors and extract them form the dataset is recommended.

Exporting the eigenvectors usually needs several minutes or hours, and generates a .txt file of several GB to TB. The size of the file predicts the memory requirement of band sorting.





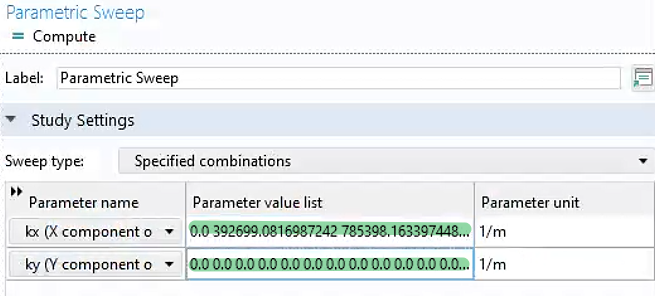
If using other tools to obtain the eigenvectors, the input code should be modified. Anyway, the input eigenvector data eventually has the form:

data[][][]

data[][][] is an array of complex numbers in Python, whose elements are like a+bj. If the eigenfield is a vector field, different components can be appended (direct summed) into data[][][], for example, . However, one of its component can also be the input eigenvector, it doesn’t impact the topology of .

1. **Usage of auxiliary tools**

2D\_triangle\_k\_mesh\_generator.ipynb is used to generate a cartesian grid in a 2D triangle domain. It outputs two coordinate list with equal length: kpoints\_x\_filename.txt and kpoints\_y\_filename.txt. It’s convenient to copy these points to “Parametric sweep” function in Comsol Multiphysics, with “Specified combinations” sweep type.



Jupyter notebook in VScode doesn’t has the function of flipping a 3D graph. bands\_visualization.nb is used to visualize the eigenvalue surface in different view angle to check whether the band sorting has mistake. Specify the sorted eigenvalue data output by the band sorting code, it shows a 3D graph that can be rotated with moving your mouse. Points inside the conventional path can be checked using this tool.

