
PYATB 教程

PYATB

2025 年 07 月 03 日

Contents:

1	教程说明	2
1.1	软件的安装	2
2	BAND_STRUCTURE、FAT_BAND、PDOS	4
2.1	介绍	4
2.2	计算步骤	5
2.3	结果展示	9
3	Fermi Surface	16
3.1	计算步骤	16
3.2	结果展示	19
4	BANDUNFOLDING	22
4.1	介绍	22
4.2	计算步骤	23
4.3	结果展示	27
5	SPIN_TEXTURE	29
5.1	介绍	29
5.2	结果展示	33
6	WILSON_LOOP	35
6.1	介绍	35
6.2	计算步骤	36
6.3	结果展示	37
7	OPTICAL_CONDUCTIVITY	39
7.1	介绍	39
7.2	计算步骤	40
7.3	结果展示	43
8	AHC、ANC	47
8.1	介绍	47
8.2	计算步骤	48
8.3	结果展示	51
9	SHIFT_CURRENT	52
9.1	计算步骤	53

9.2	结果展示	56
10	SHG	57
10.1	介绍	57
10.2	计算步骤	58
10.3	结果展示	61
11	Berry_Curvature_Dipole	63
11.1	介绍	63
11.2	计算步骤	64
11.3	结果展示	68

欢迎使用 PYATB 教程！

本教程围绕 PYATB 软件的三大核心模块——能带模块、几何模块与光学模块，结合实际案例进行详细讲解与操作演示，旨在帮助用户系统掌握 PYATB 的使用方法及其完整的计算流程。

教程内容包括：

1. 从 ABACUS 中提取第一性原理紧束缚哈密顿量，并开展能带结构分析及轨道投影等后处理工作；
2. 计算 Berry 相关物理量，包括 Berry 曲率、Wilson 环路、Chirality 等拓扑性质；
3. 计算线性与非线性光学响应，包括电子的介电函数、吸收谱、shift current、Berry curvature dipole 以及 second harmonic generation (SHG) 等光学性质。

1.1 软件的安装

1.1.1 ABACUS 的安装

本教程所使用的 ABACUS 版本为长期支持版 **LTS 3.10**，可通过以下命令获取源码：

```
git clone https://github.com/deepmodeling/abacus-develop.git -b LTS
```

关于 ABACUS 的编译与依赖环境配置，可参考以下官方安装教程：

- [ABACUS 安装教程 - Toolchain \(1-GNU\)](#)
- [ABACUS 安装教程 - Toolchain \(2-Intel\)](#)

ABACUS 运行所需的赝势与原子轨道可通过如下命令获取：

```
git clone https://github.com/abacusmodeling/ABACUS-orbitals.git
```

本教程中使用的是该仓库下 Dojo-NC-FR 目录中的 Dojo 赝势及对应的数值原子轨道。

1.1.2 PYATB 的安装

建议通过以下步骤安装 PYATB 及其依赖：

```
conda create -n pyatb python=3.12
conda activate pyatb
conda install -c conda-forge mpi4py
pip install pyatb
```

注 1，推荐优先使用 conda 安装 mpi4py，以避免 pip 在源码编译时可能引发的依赖问题；注 2，使用 conda-forge 作为软件源，可提升环境的兼容性与稳定性；

1.1.3 其他的说明

安装 PYATB 后，Python 环境中将自动包含一个辅助生成输入文件的命令行工具 `pyatb_input`。可以通过命令 `pyatb_input -h` 查看该工具的使用说明和可用选项。

例如，以下命令将在当前目录下基于 `abacus_scf_folder/` 中的自洽计算结果，生成包含能带计算模块的输入文件：

```
pyatb_input -i abacus_scf_folder/ --band
```

注：命令行参数中的 `-i` 用于指定 ABACUS 输出目录，`--band` 表示启用能带计算模块。

BAND_STRUCTURE、FAT_BAND、PDOS

2.1 介绍

材料电子性质的研究中，能带结构是最基本也是最关键的物理量之一。在周期性体系中，Kohn-Sham 方程为：

$$H|\Psi_{n\mathbf{k}}\rangle = E_{n\mathbf{k}}|\Psi_{n\mathbf{k}}\rangle,$$

其中 $|\Psi_{n\mathbf{k}}\rangle$ 为第 n 带的布洛赫波函数，可展开为：

$$|\Psi_{n\mathbf{k}}\rangle = \frac{1}{\sqrt{N}} \sum_{\mu} C_{n\mu}(\mathbf{k}) \sum_{\mathbf{R}} e^{i\mathbf{k}\cdot\mathbf{R}} |\mathbf{R}\mu\rangle,$$

其中 $|\mathbf{R}\mu\rangle \equiv \phi_{\mu}(\mathbf{r} - \tau_{\mu} - \mathbf{R})$ 是第 μ 个原子轨道，位于第 \mathbf{R} 个单位胞中， τ_{μ} 表示该轨道的中心位置。该方程可转化为广义特征值问题：

$$H(\mathbf{k})C_n(\mathbf{k}) = E_{n\mathbf{k}}S(\mathbf{k})C_n,$$

其中 $H(\mathbf{k})$ 和 $S(\mathbf{k})$ 分别是动量空间中的哈密顿量和重叠矩阵，通过对实空间矩阵进行傅里叶变换得到：

$$H_{\nu\mu}(\mathbf{k}) = \sum_{\mathbf{R}} e^{i\mathbf{k}\cdot\mathbf{R}} H_{\nu\mu}(\mathbf{R}),$$

$$S_{\nu\mu}(\mathbf{k}) = \sum_{\mathbf{R}} e^{i\mathbf{k}\cdot\mathbf{R}} S_{\nu\mu}(\mathbf{R}).$$

因此，在 PYATB 中，只需提供所需的 \mathbf{k} 点，即可通过上述公式求解特征值 $E_{n\mathbf{k}}$ ，从而绘制能带结构。

参考文献：

- Jin, G., Pang, H., Ji, Y., Dai, Z. & He, L. PYATB: An efficient Python package for electronic structure calculations using ab initio tight-binding model. Computer Physics Communications 291, 108844 (2023).

下面以三维拓扑绝缘体 Bi_2Se_3 为示例，演示如何使用 ABACUS 与 PYATB 联合进行能带结构、Fat band、PDOS 的计算。相关文件位于 tutorial/Bi2Se3_band 目录下，其中包含两个子文件夹：abacus 和 pyatb，分别提供了第一性原理计算与后续紧束缚模型计算所需的输入文件。

2.2 计算步骤

快速测试步骤：

```
$ cd tutorial/Bi2Se3_band/abacus
$ ./run_abacus.sh
$ cd ../pyatb
$ ./run_pyatb.sh
$ cd ./Out/Fat_Band
$ vi plot_fatband.py
$ python plot_fatband.py
$ cd ../../
$ python plot_spd_fatband.py
$ cd ./Out/PDOS
$ vi plot_dos.py
$ python plot_dos.py
$ cd ../../
$ python plot_spd_pdos.py
```

2.2.1 第一步，ABACUS 自洽计算产生紧束缚模型文件

在 tutorial/Bi2Se3_band/abacus 文件夹中，我们已经准备好了 ABACUS 自洽计算的输入文件，包括：

- 控制文件：INPUT
- 结构文件：STRU
- 赝势文件：*.upf
- 原子轨道基组文件：*.orb
- K 点采样文件：KPT

其中，INPUT 文件的核心参数如下，已对关键设置作出注释，其他细节可参考 ABACUS 官方文档：https://abacus.deepmodeling.com/en/latest/advanced/input_files/input-main.html

```
INPUT_PARAMETERS

# System variables
suffix          Bi2Se3
calculation      scf      # 指定计算类型为自洽计算
esolver_type     ksdfc
symmetry         1        # 设置为1时，开启对称性分析，为-1时，完全关闭对称性分析
init_chg         atomic

# Input Files
pseudo_dir       ./
orbital_dir      ./

# Plane Wave
ecutwfc          100      # 平面波截断值，可以选择轨道文件中所使用的截断能量

# Electronic structure
basis_type       lcao      # 使用原子轨道作为基组
ks_solver        genelpa
nspin            4         # 波函数的自旋分量数，开启自旋轨道耦合后自动为4
smearing_method   gauss    # 电子占据模糊方法
```

(续下页)

(接上页)

```

smearing_sigma      0.02
mixing_type          pulay    # 设置电荷密度混合方法
mixing_beta          0.7
scf_nmax             200      # 最大电子收敛步数
scf_thr              1e-8     # 电荷密度收敛精度
lspinorb             1        # 开启自旋轨道耦合
noncolin             0        # 关闭自旋非共线

# Output Variables
out_chg              1        # 输出自洽后的电荷密度
out_mat_hs2          1        # 输出 HR、SR 矩阵
out_mat_r            1        # 输出 rR 矩阵

```

执行 ABACUS 自洽计算后, 若设置了 `out_mat_hs2 = 1` 和 `out_mat_r = 1`, 将在 `OUT.*` 目录中生成所需紧束缚模型文件:

- **data-HR-sparse_SPIN0.csr**: 实空间哈密顿量
- **data-SR-sparse_SPIN0.csr**: 重叠矩阵
- **data-rR-sparse.csr**: 位置偶极矩阵

注 1: 若设置 `nspin = 2` (即自旋极化计算), 则还会生成 **data-HR-sparse_SPIN1.csr** 文件, 用于描述第二个自旋通道。注 2: ABACUS 生成紧束缚模型文件, 必须在 `lcao` 基组下, 即 `basis_type lcao`。注 3: 随着 ABACUS 版本的更新, 部分紧束缚模型文件的文件名称可能会发生变化。

在准备好了 ABACUS 各种输入文件之后, 我们可以运行目录下的 **run_abacus.sh** 脚本来进行 ABACUS 的自洽计算。

```

#!/bin/bash

# 加载ABACUS环境
source ~/softwares/abacus-develop-LTSv3.10.0/toolchain/abacus_env.sh

# 运行abacus
export OMP_NUM_THREADS=1 # 设置OpenMP线程数为1
num_mpi=8 # 设置MPI进程数
mpirun -n $num_mpi abacus > job.log 2> job.err

```

请根据具体的计算节点配置适当调整 `OMP_NUM_THREADS` 与 `num_mpi`, 以充分利用本地硬件资源, 获得更高的并行效率。

2.2.2 第二步, 使用紧束缚模型文件进行 PYATB 计算

在第一步中, ABACUS 完成自洽计算后, 会产生 HR、SR、rR 文件, 这些文件是 PYATB 必要的输入文件。然后我们进入 `tutorial/Bi2Se3_band/pyatb` 文件夹, 进行能带结构、Fat band、PDOS 的计算。接下来我们介绍 PYATB 的输入文件 **Input** 的内容:

```

INPUT_PARAMETERS
{
    nspin              4          # 与 ABACUS 的 nspin 保持一致
    package            ABACUS     # 指定数据来源为 ABACUS
    fermi_energy        9.5064566484
    fermi_energy_unit   eV
    HR_route            ../abacus/OUT.Bi2Se3/data-HR-sparse_SPIN0.csr #
↪ HR 矩阵

```

(续下页)

(接上页)

```

SR_route                ../abacus/OUT.Bi2Se3/data-SR-sparse_SPIN0.csr #_
↪SR 矩阵
rR_route                ../abacus/OUT.Bi2Se3/data-rR-sparse.csr      #_
↪rR 矩阵
HR_unit                 Ry                # HR 单位为 Rydberg
rR_unit                 Bohr              # rR 单位为 Bohr
max_kpoint_num          8000            # 单轮计算的最大k点数目
}

LATTICE
{
    lattice_constant      1.8897162
    lattice_constant_unit Bohr
    lattice_vector
    -2.069  -3.583614  0.000000
     2.069  -3.583614  0.000000
     0.000   2.389075  9.546667
}

BAND_STRUCTURE
{
    wf_collect            0                # 是否输出波函数的展开系数
    kpoint_mode           line            # 设置 k_
↪点采样模式为高对称路径, 有line, mp, direct三种
    kpoint_num            5                # 设置路径中的高对称点数
    high_symmetry_kpoint  # 设置高对称点坐标及插值数
    0.00000 0.00000 0.0000 100 # G
    0.00000 0.00000 0.5000 100 # Z
    0.50000 0.50000 0.0000 100 # F
    0.00000 0.00000 0.0000 100 # G
    0.50000 0.00000 0.0000 1  # L
    kpoint_label G, Z, F, G, L # 设置高对称点名称
}

FAT_BAND
{
    band_range           28 128           # 设置计算能带范围
    stru_file             STRU             #_
↪设置结构文件名, 与ABACUS的结构文件保持一致, 提取原子位置以及轨道文件
    kpoint_mode           line
    kpoint_num            5
    high_symmetry_kpoint
    0.00000 0.00000 0.0000 100 # G
    0.00000 0.00000 0.5000 100 # Z
    0.50000 0.50000 0.0000 100 # F
    0.00000 0.00000 0.0000 100 # G
    0.50000 0.00000 0.0000 1  # L
    kpoint_label G, Z, F, G, L
}

PDOS
{
    stru_file             STRU
    e_range               0.5064566484 19.5064566484 #_
↪设置PDOS能量范围, 绝对能量, 不是以费米能为中心的
    de                   0.01              # 设置能量间隔
    sigma                 0.07             # 设置gaussian smearing参数
}

```

(续下页)

(接上页)

```

kpoint_mode    mp                # 设置 k 点采样模式为均匀撒点
mp_grid        20 20 20          # 撒点网格
}

```

关于功能模块 BAND_STRUCTURE 完整设置参数为：

- **wf_collect**, bool 类型, 为 1 时输出波函数的展开系数, 默认为 0。
- **band_range**, int 类型, 有两个值, 设置计算的能带范围, 从 1 开始计数, 例如 1 20, 计算第 1 到第 20 共 21 条能带。不设置该参数, 会计算所有能带。
- **kpoint_mode**, str 类型, 指定 k 点采样模式, 存在此参数时会有额外的设置参数。

关于功能模块 FAT_BAND 完整设置参数为：

- **band_range**, int 类型, 有两个值, 设置计算的能带范围, 从 1 开始计数, 例如 1 20, 计算第 1 到第 20 共 21 条能带。
- **stru_file**, str 类型, 指定结构文件, 用于提取原子位置以及使用的轨道文件。
- **kpoint_mode**, str 类型, 指定 k 点采样模式, 存在此参数时会有额外的设置参数。

关于功能模块 PDOS 完整设置参数为：

- **stru_file**, str 类型, 指定结构文件, 用于提取原子位置以及使用的轨道文件。
- **e_range**, float 类型, 有两个值, 设置 PDOS 能量范围, 绝对能量, 不是以费米能为中心的。
- **de**, float 类型, 设置能量间隔
- **sigma**, float 类型, 设置 gaussian smearing 参数
- **kpoint_mode**, str 类型, 指定 k 点采样模式, 存在此参数时会有额外的设置参数。

如果某个功能模块中存在 **kpoint_mode** 参数时, 会添加额外的设置参数：

- 当 **kpoint_mode** 为 **mp** 时, 新增参数：
 - **k_start**, float 类型, 有三个值, 确定指定布里渊区范围的原点, 默认为 0.0 0.0 0.0。
 - **k_vect1**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 1, 默认为 1.0 0.0 0.0。
 - **k_vect2**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 2, 默认为 0.0 1.0 0.0。
 - **k_vect3**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 3, 默认为 0.0 0.0 1.0。
 - **mp_grid**, int 类型, 有三个值, 均匀划分指定布里渊区范围的网格数目, 没有默认值。
- 当 **kpoint_mode** 为 **line** 时, 新增参数：
 - **kpoint_num**, int 类型, 指定高对称点的数目。
 - **high_symmetry_kpoint**, float 类型, 每个高对称点有四个值, 设置高对称点坐标及插值数。
 - **kpoint_label**, str 类型, 数目与 **kpoint_num** 保持一致, 指定每一个高对称点的 label。
- 当 **kpoint_mode** 为 **direct** 时, 新增参数：
 - **kpoint_num**, int 类型, 指定 k 点的数目。
 - **kpoint_direct_coor**, float 类型, 每一个 k 点三个值, 指定 k 点的坐标。

注 1: FAT_BAND 和 PDOS 功能模块都需要结构文件, 格式与 ABACUS 的结构文件保持一致。

在准备好了 PYATB 各种输入文件之后, 我们可以运行目录下的 **run_pyatb.sh** 脚本来进行 PYATB 的计算。

```
#!/bin/bash

# 加载conda环境
source ~/software/miniconda3/bin/activate
conda activate pyatb

# 运行pyatb
export OMP_NUM_THREADS=1 # 设置OpenMP线程数为1
num_mpi=8 # 设置MPI进程数
mpirun -n $num_mpi pyatb > job.log 2> job.err
```

2.3 结果展示

运行 PYATB 后，所有计算结果将保存在 Out/ 文件夹中。每个功能模块对应一个子文件夹，便于结构化管理与后续分析

2.3.1 BAND_STRUCTURE

Out/Band_Structure 文件夹包含能带结构相关的输出，相应文件为：

- band_info.dat
- band.dat
- high_symmetry_kpoint.dat
- kpt.dat
- plot_band.py
- x_coor_array.dat

band_info.dat 如下：

```

Fermi Energy (eV):      9.5065
      Band gap (eV):      0.3197
Eigenvalue of VBM (eV):  9.3059
Eigenvalue of CBM (eV):  9.6256
VBM 1 (band index and k coor):  76  0.145000  0.145000  0.355000
VBM 2 (band index and k coor):  77  0.145000  0.145000  0.355000
CBM 1 (band index and k coor):  78  0.000000  0.000000  0.000000
CBM 2 (band index and k coor):  79  0.000000  0.000000  0.000000
CBM 3 (band index and k coor):  78  0.000000  0.000000  0.000000
CBM 4 (band index and k coor):  79  0.000000  0.000000  0.000000
```

该文件提供了基本的能带信息，包括费米能、带隙、价带顶（VBM）和导带底（CBM）的位置。其中能带指标是从 0 开始计数的。

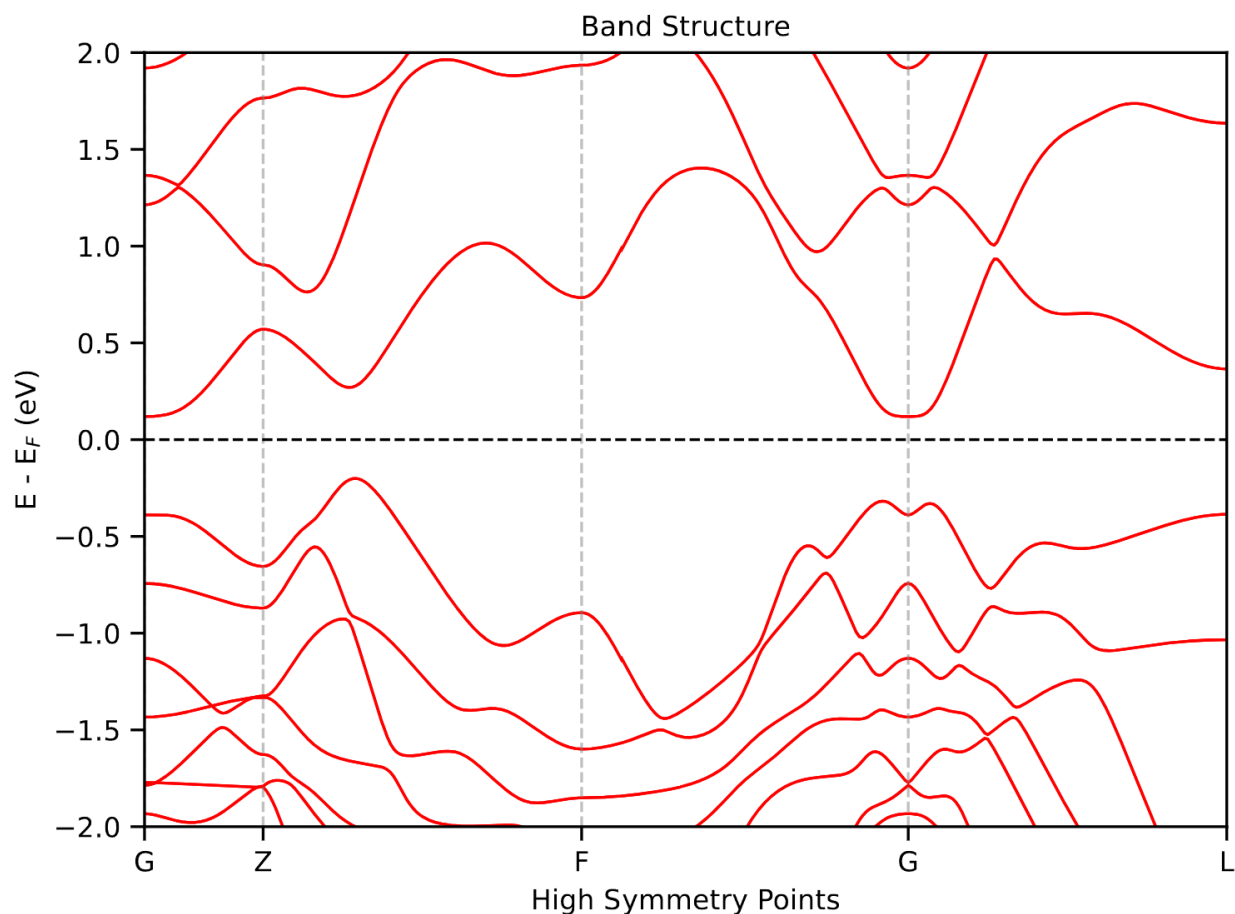
band.dat 文件包含所有 k 点的能带本征值，格式为二维矩阵：每行对应一个 k 点，每列表示一条能带，能量值未减去费米能。

kpt.dat 文件记录所有计算使用的 k 点坐标。

high_symmetry_kpoint.dat 和 **x_coor_array.dat** 是绘图脚本 **plot_band.py** 的必要输入文件。其中 **high_symmetry_kpoint.dat** 记录高对称点信息以及高对称点的 label、绘图坐标点。**x_coor_array.dat** 为绘

图的横坐标。在 `plot_band.py` 中，可以修改 `y_min` 和 `y_max` 变量来控制能带图的纵轴范围。该脚本默认绘制的是平移费米能后的能带图。

执行绘图脚本后，生成的能带图为：



2.3.2 FAT_BAND

在 `Out/Fat_Band` 文件夹下存放 Fat band 数据，包含以下文件：

- `band.dat`
- `fatband.xml`
- `high_symmetry_kpoint.dat`
- `pband.dat`
- `plot_fatband.py`

`band.dat` 文件存放所有 k 点、能带的本征值，为二维矩阵，每一行对应一个 k 点，每一列表示一条能带，能量是未平移费米能的。

`fatband.xml` 和 `pband.dat` 都是存放 fat band 数据的，只是格式不一样，前者提供了包含能带、不同轨道 fat band 的完整信息，而 `pband.dat` 是最原始数据。

`plot_fatband.py` 是绘制 fat band 的脚本，用于初步检查。你需要修改这个脚本的变量是指定绘制轨道的权重。例如我们需要绘制 Bi 和 Se 的 s、p、d 轨道的 fat band，可以修改如下的脚本参数：

```
from pyatb.tools.band import PBand
import matplotlib.pyplot as plt

efermi = 9.5064566484
pbandfile = './fatband.xml'
kptfile = './high_symmetry_kpoint.dat'

pband = PBand(pbandfile, kptfile)

# 使用species来绘制fat band, 指定Bi和Se的s、p、d轨道
species = {"Bi": [0, 1, 2], "Se": [0, 1, 2]}

# 设置能带显示范围
energy_range = [-10, 10]

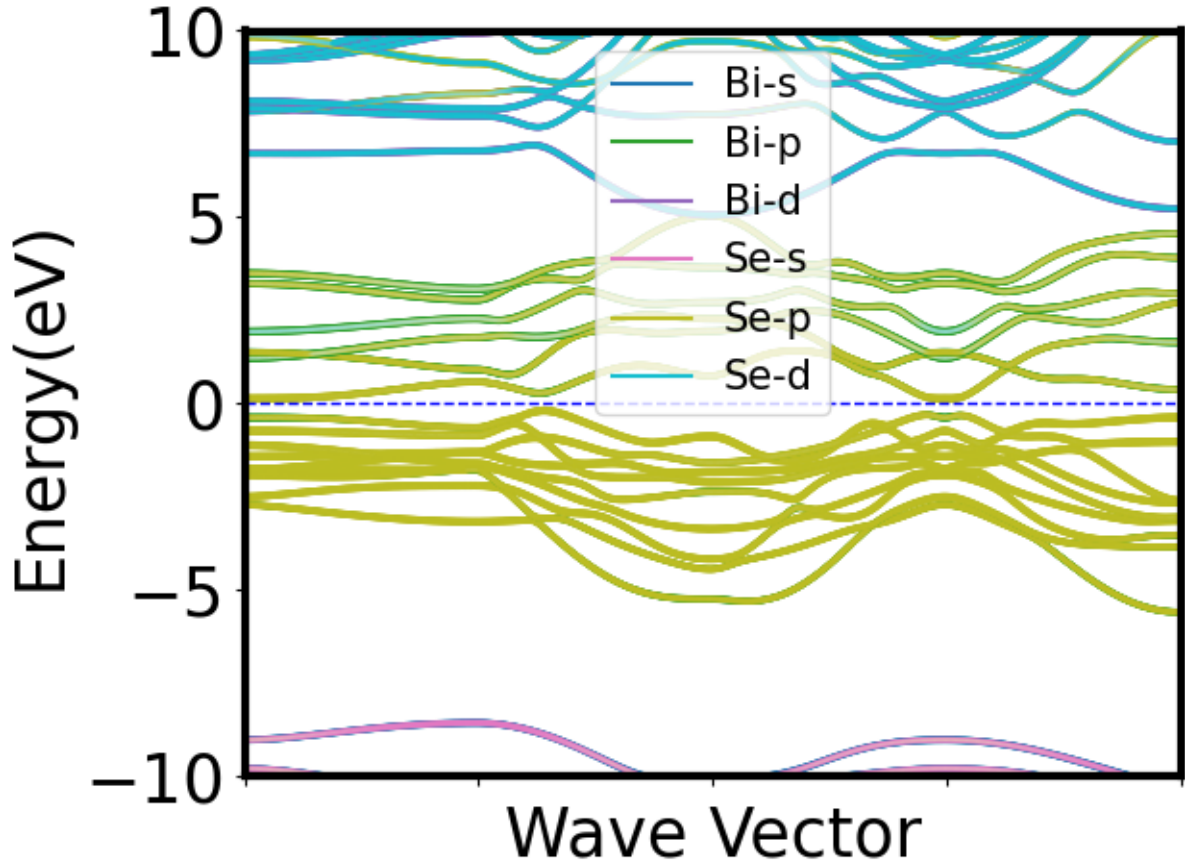
fig, ax = plt.subplots(sharex=True, figsize=(6.4, 4.8), tight_layout=True)

# 将绘图数据输出到文件中
pband.write(species=species)

# 绘制fat band草图
pband.plot_contributions(fig, ax, species=species, efermi=efermi, energy_range=energy_
    ↪range, colors=[])

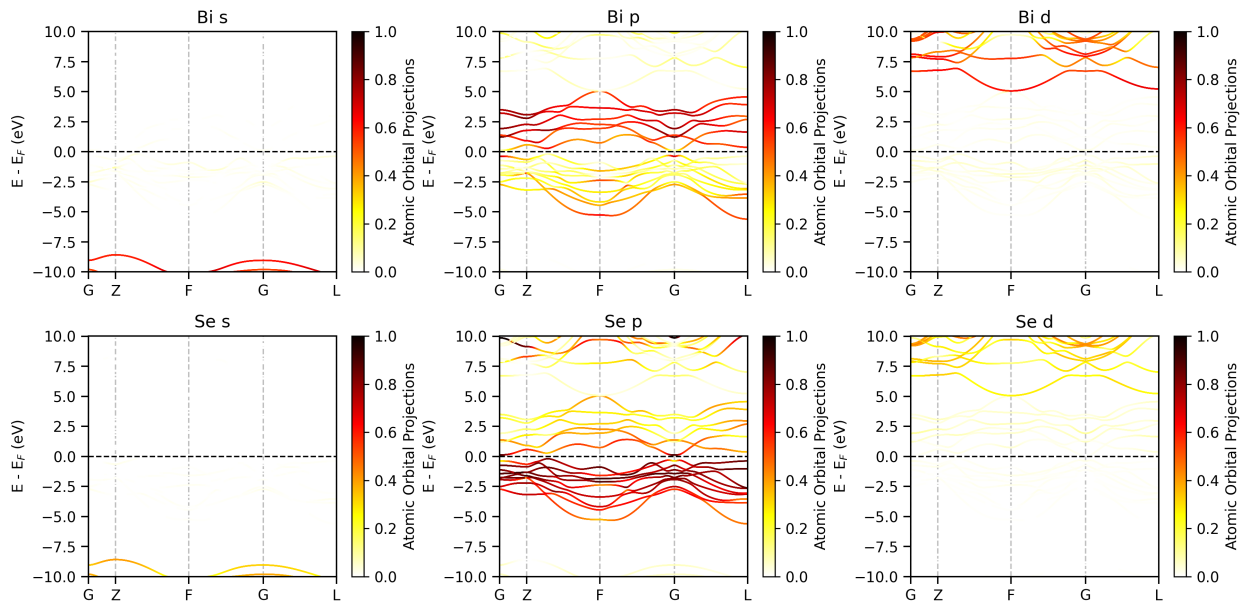
fig.savefig('fatband.png')
plt.close('all')
```

运行脚本后，会生成 **fatband.png**，绘制的 fat band 图如下：



同时绘图所用数据将保存至 `Out/Fat_Band/PBAND1_FILE` 目录中。

为获得更高质量的 fat band 图，我们在 `tutorial/Bi2Se3_band/pyatb` 文件夹下添加了新的绘图脚本 `plot_spd_fatband.py`，该脚本读取之前生成的绘图数据，进行自定义绘图，结果为：



从这个图中，我们可以看到 Bi_2Se_3 在费米能附近的能带主要是由 Bi 和 Se 的 p 轨道贡献的。

2.3.3 PDOS

在 Out/PDOS 文件夹下存放 PDOS 数据，包含：

- PDOS.dat
- PDOS.xml
- plot_dos.py
- TDOS.dat

PDOS.xml 和 PDOS.dat 都是存放 PDOS 数据的，只是格式不一样，前者提供了包含能量范围、不同轨道 PDOS 的完整信息，而 PDOS.dat 是最原始数据。TDOS.dat 存放总 DOS 的信息，第一列为能量点（未平移费米能），第二列为 TDOS 数据。

plot_dos.py 是绘制 PDOS 的脚本，用于初步检查。你需要修改这个脚本的变量是指定绘制轨道的权重。例如我们需要绘制 Bi 和 Se 的 s、p、d 轨道的 PDOS，可以修改如下的脚本参数：

```
from pyatb.tools.dosplot import TDOS, PDOS
import matplotlib.pyplot as plt

efermi = 9.5064566484
energy_range = [0.5064566484-efermi, 19.5064566484-efermi]

# ----- plot TDOS -----
tdosfile = './TDOS.dat'
tdos = TDOS(tdosfile)
dos_range = [0, 5]
fig, ax = plt.subplots(figsize=(8, 8))
dosplots = tdos.plot(fig, ax, efermi=efermi, shift=False, energy_range=energy_range,
    ↳ dos_range=dos_range)
fig.savefig('tdos.png')
plt.close()

# ----- plot PDOS -----
pdosfile = './PDOS.xml'
pdos = PDOS(pdosfile)

species = {"Bi": [0, 1, 2], "Se": [0, 1, 2]}
dos_range = [0, 5]

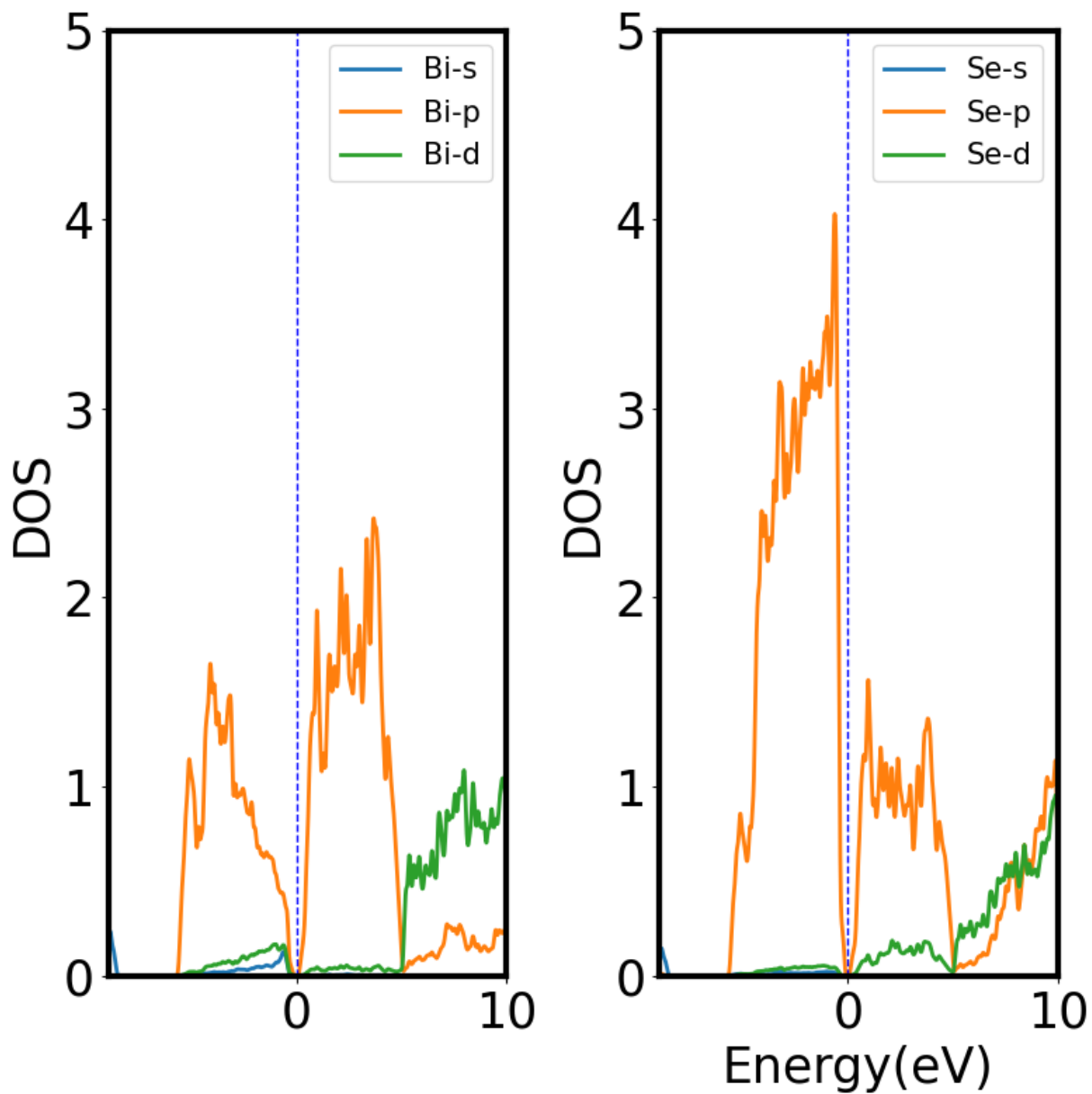
fig, ax = plt.subplots(1, 2, sharex=True, figsize=(8, 8), tight_layout=True)

pdos.write(species=species)

# 2. plot different contributions in single picture
dosplots = pdos.plot(fig, ax, species=species, efermi=efermi, shift=False, energy_
    ↳ range=energy_range, dos_range=dos_range)

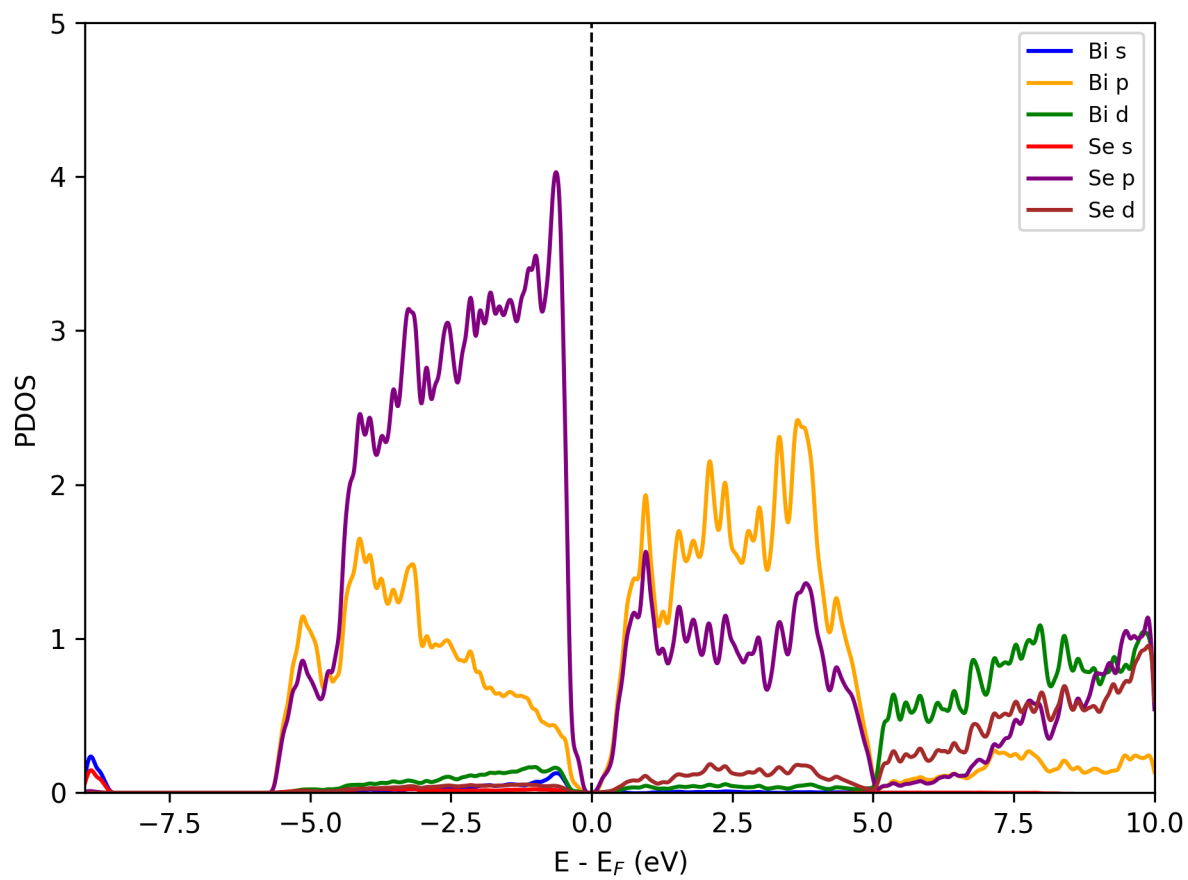
fig.savefig('pdos.png')
plt.close('all')
```

运行脚本后，会生成 pdos.png，绘制的 PDOS 图如下：



同时绘图数据也会保存在 `Out/Fat_Band/PDOS_FILE` 目录下。

为了获得更加好的 PDOS 图，我们在 `tutorial/Bi2Se3_band/pyatb` 文件夹下添加了新的绘图脚本 `plot_spd_pdos.py`，该脚本读取之前生成的绘图数据，进行自定义绘图，结果为：



下面以 Cu 为示例, 演示如何使用 ABACUS 与 PYATB 联合进行 Fermi Surface 计算。相关文件位于 `tutorial/Cu_fermi_surface` 目录下, 其中包含两个子文件夹: `abacus` 和 `pyatb`, 分别提供了第一性原理计算与后续紧束缚模型计算所需的输入文件。

3.1 计算步骤

快速测试步骤:

```
$ cd tutorial/Cu_fermi_surface/abacus
$ ./run_abacus.sh
$ cd ../pyatb
$ ./run_pyatb.sh
$ python dat_to_xsf.py
```

3.1.1 第一步, ABACUS 自洽计算产生紧束缚模型文件

在 `tutorial/Cu_fermi_surface/abacus` 文件夹中, 我们已经准备好了 ABACUS 自洽计算的输入文件, 包括:

- 控制文件: `INPUT`
- 结构文件: `STRU`
- 赝势文件: `*.upf`
- 原子轨道基组文件: `*.orb`
- K 点采样文件: `KPT`

其中, `INPUT` 文件的核心参数如下, 已对关键设置作出注释, 其他细节可参考 ABACUS 官方文档: https://abacus.deepmodeling.com/en/latest/advanced/input_files/input-main.html

```

INPUT_PARAMETERS

# System variables
suffix          Cu
calculation      scf      # 指定计算类型为自洽计算
esolver_type     ksdfc
symmetry         1        # 设置为1时，开启对称性分析，为-1时，完全关闭对称性分析
init_chg         atomic

# Input Files
pseudo_dir       ./
orbital_dir       ./

# Plane Wave
ecutwfc          100      # 平面波截断值，可以选择轨道文件中所使用的截断能量

# Electronic structure
basis_type       lcao      # 使用原子轨道作为基组
ks_solver         genelpa
nspin            1
smearing_method   gauss    # 电子占据模糊方法
smearing_sigma    0.02
mixing_type       pulay    # 设置电荷密度混合方法
mixing_beta       0.4
scf_nmax          200      # 最大电子收敛步数
scf_thr           1e-8     # 电荷密度收敛精度
lspinorb          0        # 关闭自旋轨道耦合
noncolin          0        # 关闭自旋非共线

# Output Variables
out_chg           1        # 输出自洽后的电荷密度
out_mat_hs2       1        # 输出HR、SR矩阵
out_mat_r         1        # 输出rR矩阵

```

执行 ABACUS 自洽计算后，若设置了 `out_mat_hs2 = 1` 和 `out_mat_r = 1`，将在 `OUT.*` 目录中生成所需紧束缚模型文件：

- **data-HR-sparse_SPIN0.csr**：实空间哈密顿量
- **data-SR-sparse_SPIN0.csr**：重叠矩阵
- **data-rR-sparse.csr**：位置偶极矩阵

注1：若设置 `nspin = 2`（即自旋极化计算），则还会生成 **data-HR-sparse_SPIN1.csr** 文件，用于描述第二个自旋通道。注2：ABACUS 生成紧束缚模型文件，必须在 `lcao` 基组下，即 `basis_type lcao`。注3：随着 ABACUS 版本的更新，部分紧束缚模型文件的文件名称可能会发生变化。

在准备好了 ABACUS 各种输入文件之后，我们可以运行目录下的 **run_abacus.sh** 脚本来进行 ABACUS 的自洽计算。

```

#!/bin/bash

# 加载ABACUS环境
source ~/softwares/abacus-develop-LTSv3.10.0/toolchain/abacus_env.sh

# 运行abacus
export OMP_NUM_THREADS=1 # 设置OpenMP线程数为1
num_mpi=8 # 设置MPI进程数
mpirun -n $num_mpi abacus > job.log 2> job.err

```

请根据具体的计算节点配置适当调整 OMP_NUM_THREADS 与 num_mpi，以充分利用本地硬件资源，获得更高的并行效率。

3.1.2 第二步，使用紧束缚模型文件进行 PYATB 计算

在第一步中，ABACUS 完成自洽计算后，会产生 HR、SR 文件，这些文件是 PYATB 必要的输入文件。然后我们进入 tutorial/Cu_fermi_surface/pyatb 文件夹，进行 fermi surface 的计算。接下来我们介绍 PYATB 的输入文件 **Input** 的内容：

```
INPUT_PARAMETERS
{
    nspin                1                # 与 ABACUS 的 nspin 保持一致
    package              ABACUS           # 指定数据来源为 ABACUS
    fermi_energy          17.62527821
    fermi_energy_unit     eV
    HR_route              ../abacus/OUT.Cu/data-HR-sparse_SPIN0.csr # HR 矩阵
    SR_route              ../abacus/OUT.Cu/data-SR-sparse_SPIN0.csr # SR 矩阵
    HR_unit               Ry               # HR 单位为 Rydberg
    max_kpoint_num        8000            # 单轮计算的最大 k 点数目
}

LATTICE
{
    lattice_constant      6.91640
    lattice_constant_unit Bohr
    lattice_vector
    0.50      0.50      0.00
    0.50      0.00      0.50
    0.00      0.50      0.50
}

BAND_STRUCTURE
{
    wf_collect            0                # 是否输出波函数的展开系数
    kpoint_mode           mp              # 设置 k_
    ↪点采样模式为均匀格点采样，有 line, mp, direct 三种
    mp_grid               50 50 50        # 划分布里渊区网格数目
}

FERMI_SURFACE
{
    bar                   1e-5            # 判断能量接近 fermi_energy_
    ↪的最大容许误差
    kpoint_mode           mp              # 设置 k_
    ↪点采样模式为均匀格点采样，有 line, mp, direct 三种
    mp_grid               50 50 50        # 划分布里渊区网格数目
}
```

关于功能模块 BAND_STRUCTURE 完整设置参数为：

- **wf_collect**, bool 类型，为 1 时输出波函数的展开系数，默认为 0。
- **band_range**, int 类型，有两个值，设置计算的能带范围，从 1 开始计数，例如 1 20，计算第 1 到第 20 共 21 条能带。不设置该参数，会计算所有能带。
- **kpoint_mode**, str 类型，指定 k 点采样模式，存在此参数时会有额外的设置参数。

关于功能模块 FERMI_SURFACE 完整设置参数为：

- **bar**, float 类型, 判断能量接近 `fermi_energy` 的最大容许误差。
- **nbands**, int 类型, 有两个值, 设置计算的能带范围, 用于加速寻找能量为 `fermi_energy` 的 k 点。默认值为 0 0, 意味着考虑所有能带。
- **kpoint_mode**, str 类型, 指定 k 点采样模式, 存在此参数时会有额外的设置参数。

如果某个功能模块中存在 **kpoint_mode** 参数时, 会添加额外的设置参数:

- 当 **kpoint_mode** 为 **mp** 时, 新增参数:
 - **k_start**, float 类型, 有三个值, 确定指定布里渊区范围的原点, 默认为 0.0 0.0 0.0。
 - **k_vect1**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 1, 默认为 1.0 0.0 0.0。
 - **k_vect2**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 2, 默认为 0.0 1.0 0.0。
 - **k_vect3**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 3, 默认为 0.0 0.0 1.0。
 - **mp_grid**, int 类型, 有三个值, 均匀划分指定布里渊区范围的网格数目, 没有默认值。
- 当 **kpoint_mode** 为 **line** 时, 新增参数:
 - **kpoint_num**, int 类型, 指定高对称点的数目。
 - **high_symmetry_kpoint**, float 类型, 每个高对称点有四个值, 设置高对称点坐标及插值数。
 - **kpoint_label**, str 类型, 数目与 **kpoint_num** 保持一致, 指定每一个高对称点的 label。
- 当 **kpoint_mode** 为 **direct** 时, 新增参数:
 - **kpoint_num**, int 类型, 指定 k 点的数目。
 - **kpoint_direct_coor**, float 类型, 每一个 k 点三个值, 指定 k 点的坐标。

在准备好了 PYATB 各种输入文件之后, 我们可以运行目录下的 **run_pyatb.sh** 脚本来进行 PYATB 的计算。

```
#!/bin/bash

# 加载 conda 环境
source ~/software/miniconda3/bin/activate
conda activate pyatb

# 运行 pyatb
export OMP_NUM_THREADS=1 # 设置 OpenMP 线程数为 1
num_mpi=8 # 设置 MPI 进程数
mpirun -n $num_mpi pyatb > job.log 2> job.err
```

3.2 结果展示

运行 PYATB 后, 所有计算结果将保存在 `Out/` 文件夹中。每个功能模块对应一个子文件夹, 便于结构化管理与后续分析

3.2.1 FERMI_SURFACE

Out/Fermi_Surface 文件夹包含费米能 k 点相关的输出，相应文件为：

- **fermi_surface_kpt.dat**：所有满足 Fermi 能的 k 点的分数坐标，格式为 (k_x, k_y, k_z) 。
- **plot_fermi_surface.py**：用于将 k 点绘制在布里渊区中的 Python 脚本。

注 1：当前 FERMI_SURFACE 模块的绘图较为粗略，但已可用于初步筛选满足条件的 k 点。

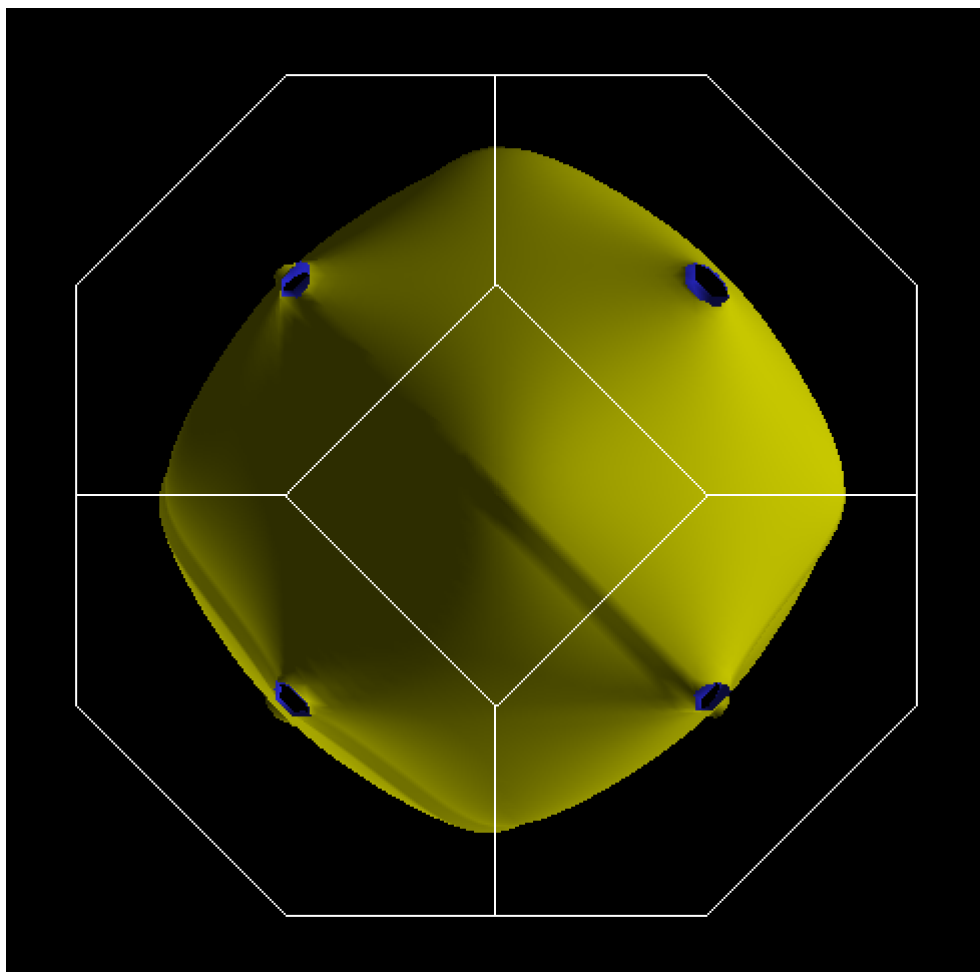
为获得更高质量的 fermi surface 可视化，我们推荐结合 BAND_STRUCTURE 模块与 XCrySDen 工具。

3.2.2 BAND_STRUCTURE 绘制 fermi surface

在 Out/Band_Structure 文件夹下存放 BAND_STRUCTURE 数据，包含以下文件：

- **band_info.dat**：包含费米能、带隙、价带顶（VBM）和导带底（CBM）信息。
- **band.dat**：包含所有 k 点的能带本征值，格式为二维矩阵：每行对应一个 k 点，每列表示一条能带，能量值未减去费米能。
- **kpt.dat**：记录所有计算使用的 k 点坐标。

我们运行 tutorial/Cu_fermi_surface/pyatb/dat_to_xsf.py 脚本，可将 **band.dat** 转换为 XCrySDen 支持的 **.xsf** 格式。转换完成后，即可使用 XCrySDen 查看 fermi surface 可视化结果：



注 1: 这个转换脚本在处理不同系统的时候, 需要自己修改脚本中的变量, 例如 `fermi_energy`、晶格参数、`band.dat` 文件路径、`kpt.dat` 文件路径。

4.1 介绍

Band unfolding 是一种用于将超胞（supercell）计算得到的波函数投影回原胞（primitive cell）倒易空间中对应 \mathbf{k} 点的技术，从而得到与实验可观测量（如 ARPES 光谱）一致的谱函数（spectral function）。

在第一性原理计算中，使用超胞往往是为了引入杂质、缺陷或模拟合金等非周期性扰动。但超胞计算结果的能带结构由于布里渊区缩小而难以直接与实验结果进行比较。通过 band unfolding，可以将超胞能带结构映射回原胞布里渊区，恢复更直观的动量空间信息。

在实际计算中，原胞的晶格矢量 $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ 与超胞的晶格矢量 $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$ 之间存在如下关系：

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

下面的图直观展示了超胞与原胞的实空间与倒空间对应关系。

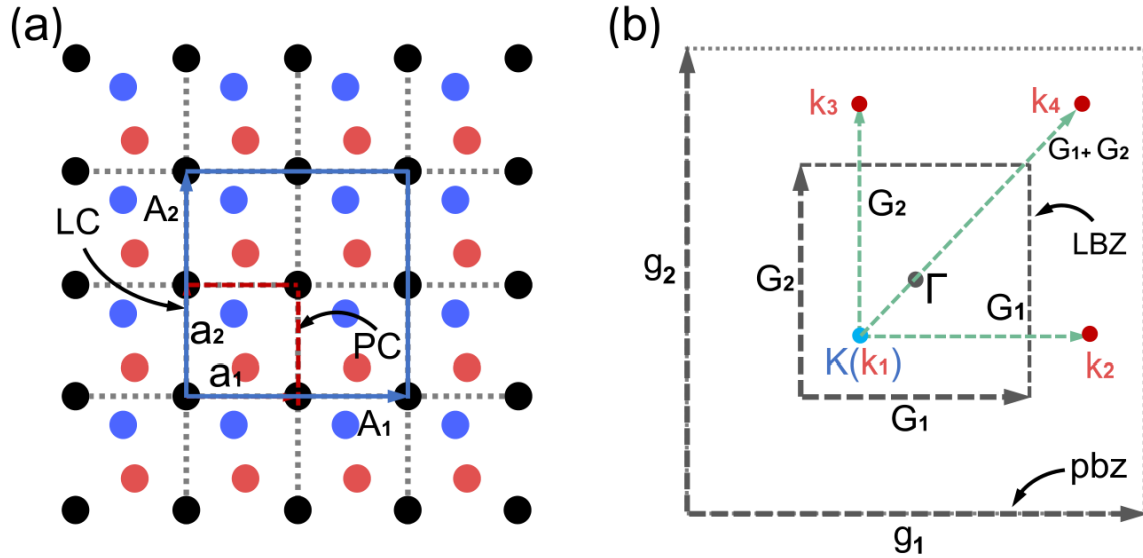


Fig. 1. Illustration of (a) the relation between the large cell (LC) and projected cell (PC), and (b) the corresponding BZs. $k_1 - k_4$ of the PC are folded to the K point of the LC. LBZ and pbz are the first BZs of the LC and PC, respectively.

Band unfolding 方法常用于含缺陷、杂质、合金或界面的体系分析，是连接第一性原理计算与实验观测的重要工具。更详细的理论背景与应用实例可参考以下文献：

- First-principles calculations of the surface states of doped and alloyed topological materials via band unfolding method.

下面以 NV 色心为示例，演示如何使用 ABACUS 与 PYATB 联合进行 BANDUNFOLDING 计算。相关文件位于 tutorial/NV_bandunfolding 目录下，其中包含两个子文件夹：abacus 和 pyatb，分别提供了第一性原理计算与后续紧束缚模型计算所需的输入文件。

4.2 计算步骤

快速测试步骤：

```
$ cd tutorial/NV_bandunfolding/abacus
$ ./run_abacus.sh
$ cd ../pyatb
$ ./run_pyatb.sh
$ cd ./Out/Bandunfolding
$ vi plot_unfold.py
$ python plot_unfold.py
```

4.2.1 第一步，ABACUS 自洽计算产生紧束缚模型文件

在 tutorial/NV_bandunfolding/abacus 文件夹中，我们已经准备好了 ABACUS 自洽计算的输入文件，包括：

- 控制文件：INPUT
- 结构文件：STRU
- 赝势文件：*.upf
- 原子轨道基组文件：*.orb
- K 点采样文件：KPT

其中，INPUT 文件的核心参数如下，已对关键设置作出注释，其他细节可参考 ABACUS 官方文档：https://abacus.deepmodeling.com/en/latest/advanced/input_files/input-main.html

```
INPUT_PARAMETERS

# System variables
suffix          NV
calculation      scf      # 指定计算类型为自洽计算
esolver_type     ksdfc
symmetry         1        # 设置为1时，开启对称性分析，为-1时，完全关闭对称性分析
init_chg         atomic

# Input Files
pseudo_dir       ./
orbital_dir       ./

# Plane Wave
ecutwfc          100      # 平面波截断值，可以选择轨道文件中所使用的截断能量

# Electronic structure
basis_type       lcao      # 使用原子轨道作为基组
ks_solver         genelpa
nelec            254      # 设置总电子数，多一个电子，体系为负一价电荷态
nspin            1
smearing_method   gauss    # 电子占据模糊方法
smearing_sigma    5e-4
mixing_type       broyden   # 设置电荷密度混合方法
mixing_beta       0.8
scf_nmax          300      # 最大电子收敛步数
scf_thr           1e-7     # 电荷密度收敛精度

# Output Variables
out_chg          1        # 输出自洽后的电荷密度
out_mat_hs2       1        # 输出 HR、SR 矩阵
out_mat_r         1        # 输出 rR 矩阵
```

执行 ABACUS 自洽计算后，若设置了 `out_mat_hs2 = 1` 和 `out_mat_r = 1`，将在 `OUT.*` 目录中生成所需紧束缚模型文件：

- **data-HR-sparse_SPIN0.csr**：实空间哈密顿量
- **data-SR-sparse_SPIN0.csr**：重叠矩阵
- **data-rR-sparse.csr**：位置偶极矩阵

注 1: 若设置 `nspin = 2` (即自旋极化计算), 则还会生成 **data-HR-sparse_SPIN1.csr** 文件, 用于描述第二个自旋通道。注 2: ABACUS 生成紧束缚模型文件, 必须在 `lcao` 基组下, 即 `basis_type lcao`。注 3: 随着 ABACUS 版本的更新, 部分紧束缚模型文件的文件名称可能会发生变化。

在准备好了 ABACUS 各种输入文件之后, 我们可以运行目录下的 **run_abacus.sh** 脚本来进行 ABACUS 的自洽计算。

```
#!/bin/bash

# 加载ABACUS环境
source ~/softwares/abacus-develop-LTSv3.10.0/toolchain/abacus_env.sh

# 运行abacus
export OMP_NUM_THREADS=1 # 设置OpenMP线程数为1
num_mpi=8 # 设置MPI进程数
mpirun -n $num_mpi abacus > job.log 2> job.err
```

请根据具体的计算节点配置适当调整 `OMP_NUM_THREADS` 与 `num_mpi`, 以充分利用本地硬件资源, 获得更高的并行效率。

4.2.2 第二步, 使用紧束缚模型文件进行 PYATB 计算

在第一步中, ABACUS 完成自洽计算后, 会产生 **HR**、**SR**、**rR** 文件, 这些文件是 PYATB 必要的输入文件。然后我们进入 `tutorial/NV_bandunfolding/pyatb` 文件夹, 进行 **BANDUNFOLDING** 的计算。接下来我们介绍 PYATB 的输入文件 **Input** 的内容:

```
INPUT_PARAMETERS
{
    nspin                1                # 与 ABACUS 的 nspin 保持一致
    package              ABACUS          # 指定数据来源为 ABACUS
    fermi_energy         15.984509735
    fermi_energy_unit    eV
    HR_route             ../abacus/OUT.NV/data-HR-sparse_SPIN0.csr # HR_
    ↪ 矩阵
    SR_route             ../abacus/OUT.NV/data-SR-sparse_SPIN0.csr # SR_
    ↪ 矩阵
    rR_route             ../abacus/OUT.NV/data-rR-sparse.csr      # rR_
    ↪ 矩阵
    HR_unit              Ry                # HR 单位为 Rydberg
    rR_unit              Bohr              # rR 单位为 Bohr
}

LATTICE
{
    lattice_constant     1.8897162
    lattice_constant_unit Bohr
    lattice_vector       7.13366 0 0
                        0 7.13366 0
                        0 0 7.13366
}

BANDUNFOLDING
{
    stru_file            STRU              #_
    ↪设置结构文件名, 与ABACUS的结构文件保持一致, 提取原子位置以及轨道文件
```

(续下页)

(接上页)

```

ecut                                10                                #_
→ 投影平面波截断能 (Ry), 数值远低于自洽计算截断能, 建议取值范围为 10~50 Ry
band_range                          10 250                        # 对指标范围内的能带进行反折叠处理
m_matrix                            -2 2 2 2 -2 2 2 2 -2 #_
→ 原胞到超胞的扩充矩阵, 按行输入矩阵的 9 个分量。A = M*a, A、a为超胞、原胞晶格矢量矩阵
kpoint_mode                         line                          # 设置 k_
→ 点采样模式为高对称路径, 有line, mp, direct三种
kpoint_num                          5                            # 设置路径中的高对称点数
high_symmetry_kpoint                #_
→ 设置高对称点坐标及插值数, 这里是原胞的高对称点
0.500000 0.000000 0.500000 300 # X
0.500000 0.250000 0.750000 300 # W
0.500000 0.500000 0.500000 300 # L
0.000000 0.000000 0.000000 300 # G
0.500000 0.000000 0.500000 1 # X
kpoint_label                        X, W, L, G, X # 设置高对称点名称
}

```

关于功能模块 BANDUNFOLDING 完整设置参数为:

- **stru_file**, str 类型, 指定结构文件, 用于提取原子位置以及使用的轨道文件。
- **ecut**, float 类型, 原子轨道投影到平面波基组时, 需要确定平面波数目。此参数为投影平面波截断能 (Ry), 数值远低于自洽计算截断能, 建议取值范围为 10~50 Ry
- **band_range**, int 类型, 有两个值, 设置反折叠的能带范围, 从 1 开始计数, 例如 1 20, 计算第 1 到第 20 共 21 条能带。不设置该参数, 会计算所有能带。
- **m_matrix**, float 类型, 有 9 个值, 原胞到超胞的扩充矩阵, 按行输入矩阵的 9 个分量。A = M*a, A、a 为超胞、原胞晶格矢量矩阵。
- **kpoint_mode**, str 类型, 指定 k 点采样模式, 存在此参数时会有额外的设置参数。

如果某个功能模块中存在 **kpoint_mode** 参数时, 会添加额外的设置参数:

- 当 **kpoint_mode** 为 **mp** 时, 新增参数:
 - **k_start**, float 类型, 有三个值, 确定指定布里渊区范围的原点, 默认为 0.0 0.0 0.0。
 - **k_vect1**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 1, 默认为 1.0 0.0 0.0。
 - **k_vect2**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 2, 默认为 0.0 1.0 0.0。
 - **k_vect3**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 3, 默认为 0.0 0.0 1.0。
 - **mp_grid**, int 类型, 有三个值, 均匀划分指定布里渊区范围的网格数目, 没有默认值。
- 当 **kpoint_mode** 为 **line** 时, 新增参数:
 - **kpoint_num**, int 类型, 指定高对称点的数目。
 - **high_symmetry_kpoint**, float 类型, 每个高对称点有四个值, 设置高对称点坐标及插值数。
 - **kpoint_label**, str 类型, 数目与 **kpoint_num** 保持一致, 指定每一个高对称点的 label。
- 当 **kpoint_mode** 为 **direct** 时, 新增参数:
 - **kpoint_num**, int 类型, 指定 k 点的数目。
 - **kpoint_direct_coor**, float 类型, 每一个 k 点三个值, 指定 k 点的坐标。

在准备好了 PYATB 各种输入文件之后, 我们可以运行目录下的 **run_pyatb.sh** 脚本来进行 PYATB 的计算。

```
#!/bin/bash

# 加载conda环境
source ~/software/miniconda3/bin/activate
conda activate pyatb

# 运行pyatb
export OMP_NUM_THREADS=1 # 设置OpenMP线程数为1
num_mpi=8 # 设置MPI进程数
mpirun -n $num_mpi pyatb > job.log 2> job.err
```

4.3 结果展示

运行 PYATB 后，所有计算结果将保存在 Out/ 文件夹中。每个功能模块对应一个子文件夹，便于结构化管理与后续分析

4.3.1 BANDUNFOLDING

Out/Bandunfolding 文件夹包含能带反折叠相关的输出，相应文件为：

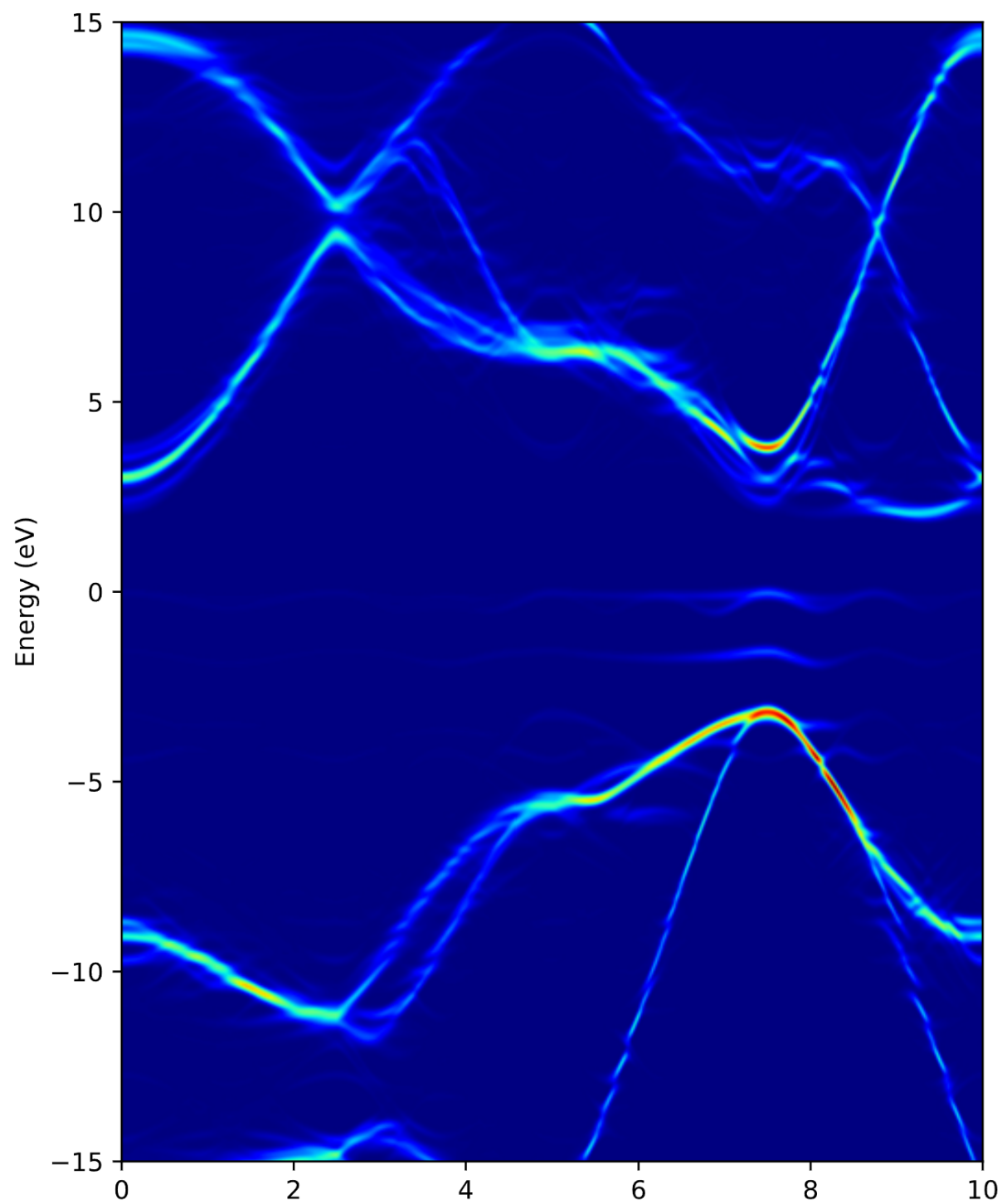
- **kpt.dat**：记录所有计算使用的 k 点坐标。
- **spectral_weight.dat**：能带反折叠谱函数权重数据。有两列数据，第一列为能量，第二列为谱函数。输出顺序：外循环 k 点，内循环能带。
- **plot_unfold.py**：绘图脚本。对该文件进行修改可以调整能带反折叠谱函数图。

我们进入 tutorial/NV_bandunfolding/pyatb/Out/Bandunfolding，对 **plot_unfold.py** 进行如下修改，来改变能带窗口范围、能量网格密度，色散大小：

```
# 纵轴能量范围
y_min = -15 # eV
y_max = 15 # eV

e_mesh = 4000 # 调整网格密度
sigma = 0.1 # 高斯展宽大小
```

执行该绘图脚本后，我们可以得到能带反折叠谱函数图：



5.1 介绍

自旋纹理 (Spin Texture) 描述的是在布里渊区中, 每个动量点 \mathbf{k} 对应的电子态的自旋期望值 $\langle \boldsymbol{\sigma}(\mathbf{k}) \rangle$ 的分布。它直观地揭示了能带中电子自旋与动量之间的耦合关系, 是理解拓扑物态、自旋霍尔效应、Rashba/Dresselhaus 效应以及自旋输运行为等关键物理现象的重要工具。

在含自旋的系统中, 考虑某一条能带 n , 其在 \mathbf{k} 点的自旋期望值定义为:

$$\langle \boldsymbol{\sigma} \rangle_n(\mathbf{k}) = \langle \Psi_{n\mathbf{k}} | \boldsymbol{\sigma} | \Psi_{n\mathbf{k}} \rangle$$

其中:

- $|\Psi_{n\mathbf{k}}\rangle$ 是该能带的布洛赫态;
- $\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$ 为泡利矩阵;
- $\langle \boldsymbol{\sigma} \rangle_n$ 即表示该态在 x 、 y 、 z 方向上的自旋分量。

典型的 spin texture 包括:

- **Rashba 型纹理**: 自旋呈现圆周形排列, 垂直于动量方向;
- **Dresselhaus 型纹理**: 与晶体对称性有关的交错型结构;

在 PYATB 中基于原子轨道基组, 实现方式如下:

$$\langle \Psi_{n\mathbf{k}} | \hat{\sigma}_i | \Psi_{n\mathbf{k}} \rangle = \sum_{\mu, \nu, s, s'} C_{n, \mu s}^*(\mathbf{k}) c(\mathbf{k}) \hat{\sigma}_{i, ss'} C_{n, \nu s'}(\mathbf{k}),$$

其中

- $s=\uparrow, \downarrow$, 是自旋指标;
- $C_{n, \mu s}(\mathbf{k})$ 是 $|\Psi_{n\mathbf{k}}\rangle$ 的轨道展开系数;
- $|\Psi_{n\mathbf{k}}\rangle$ 是原子轨道交叠矩阵。

参考文献:

- Cheng, Y. C., Zhu, Z. Y., Tahir, M. & Schwingenschlögl, U. Spin-orbit-induced spin splittings in polar transition metal dichalcogenide monolayers. EPL 102, 57001 (2013).

下面以二维 WSSe_2 为示例，演示如何使用 ABACUS 与 PYATB 联合进行 Spin texture 计算。相关文件位于 tutorial/WSSe_spin_texture 目录下，其中包含两个子文件夹：abacus 和 pyatb，分别提供了第一性原理计算与后续紧束缚模型计算所需的输入文件。

快速测试步骤：

```
$ cd tutorial/WSSe_spin_texture/abacus
$ ./run_abacus.sh
$ cd ../pyatb
$ ./run_pyatb.sh
```

5.1.1 第一步，ABACUS 自洽计算产生紧束缚模型文件

在 tutorial/WSSe_spin_texture/abacus 文件夹中，我们已经准备好了 ABACUS 自洽计算的输入文件，包括：

- 控制文件：INPUT
- 结构文件：STRU
- 赝势文件：*.upf
- 原子轨道基组文件：*.orb
- K 点采样文件：KPT

其中，INPUT 文件的核心参数如下，已对关键设置作出注释，其他细节可参考 ABACUS 官方文档：https://abacus.deepmodeling.com/en/latest/advanced/input_files/input-main.html

```
INPUT_PARAMETERS

# System variables
suffix          WSSe
calculation      scf          # 指定计算类型为自洽计算
esolver_type     ksdfc
symmetry         1           # 设置为1时，开启对称性分析，为-1时，完全关闭对称性分析
init_chg         atomic

# Input Files
pseudo_dir       ./
orbital_dir       ./

# Plane Wave
ecutwfc          100         # 平面波截断值，可以选择轨道文件中所使用的截断能量

# Electronic structure
basis_type       lcao        # 使用原子轨道作为基组
ks_solver         genelpa
nspin            4           # 波函数的自旋分量数，开启自旋轨道耦合后自动为4
smearing_method   gauss      # 电子占据模糊方法
smearing_sigma    0.02
mixing_type       pulay      # 设置电荷密度混合方法
mixing_beta       0.7
scf_nmax          200        # 最大电子收敛步数
scf_thr           1e-8       # 电荷密度收敛精度
lspinorb          1          # 开启自旋轨道耦合
```

(续下页)

(接上页)

```

noncolin          0          # 关闭自旋非共线

# Output Variables
out_chg            1          # 输出自洽后的电荷密度
out_mat_hs2        1          # 输出 HR、SR 矩阵
out_mat_r          1          # 输出 rR 矩阵

```

执行 ABACUS 自洽计算后，若设置了 `out_mat_hs2 = 1` 和 `out_mat_r = 1`，将在 `OUT.*` 目录中生成所需紧束缚模型文件：

- **data-HR-sparse_SPIN0.csr**：实空间哈密顿量
- **data-SR-sparse_SPIN0.csr**：重叠矩阵
- **data-rR-sparse.csr**：位置偶极矩阵

注 1：若设置 `nspin = 2`（即自旋极化计算），则还会生成 **data-HR-sparse_SPIN1.csr** 文件，用于描述第二个自旋通道。注 2：ABACUS 生成紧束缚模型文件，必须在 `lcao` 基组下，即 `basis_type lcao`。注 3：随着 ABACUS 版本的更新，部分紧束缚模型文件的文件名称可能会发生变化。

在准备好了 ABACUS 各种输入文件之后，我们可以运行目录下的 **run_abacus.sh** 脚本来进行 ABACUS 的自洽计算。

```

#!/bin/bash

# 加载 ABACUS 环境
source ~/softwares/abacus-develop-LTSv3.10.0/toolchain/abacus_env.sh

# 运行 abacus
export OMP_NUM_THREADS=1 # 设置 OpenMP 线程数为 1
num_mpi=8 # 设置 MPI 进程数
mpirun -n $num_mpi abacus > job.log 2> job.err

```

请根据具体的计算节点配置适当调整 `OMP_NUM_THREADS` 与 `num_mpi`，以充分利用本地硬件资源，获得更高的并行效率。

5.1.2 第二步，使用紧束缚模型文件进行 PYATB 计算

在第一步中，ABACUS 完成自洽计算后，会产生 HR、SR、rR 文件，这些文件是 PYATB 必要的输入文件。然后我们进入 `tutorial/WSSe_spin_texture/pyatb` 文件夹，进行能带结构、Fat band、PDOS 的计算。接下来我们介绍 PYATB 的输入文件 **Input** 的内容：

```

INPUT_PARAMETERS
{
    nspin                4
    package              ABACUS
    fermi_energy         1.4232523137
    fermi_energy_unit    eV
    HR_route             ../abacus/OUT.WSSe/data-HR-sparse_SPIN0.csr
    SR_route             ../abacus/OUT.WSSe/data-SR-sparse_SPIN0.csr
    rR_route             ../abacus/OUT.WSSe/data-rR-sparse.csr
    HR_unit              Ry
    rR_unit              Bohr
    max_kpoint_num       4000
}

```

(续下页)

(接上页)

```

LATTICE
{
    lattice_constant          1.8897162
    lattice_constant_unit     Bohr
    lattice_vector
    3.2521424294             0.0000000000             0.0000000000
    1.6260712147             2.8164379605             0.0000000000
    0.0000000000             0.0000000000             18.2324867249
}

BAND_STRUCTURE
{
    wf_collect                0
    kpoint_mode               line
    kpoint_num                7
    high_symmetry_kpoint
    0.0000000000             0.0000000000             0.0000000000             100             # G
    0.0000000000             0.5000000000             0.0000000000             100             # M'
    0.3333333333             0.6666666666             0.0000000000             100             # K'
    0.0000000000             0.0000000000             0.0000000000             100             # G
    0.5000000000             0.5000000000             0.0000000000             100             # M
    0.6666666666             0.3333333333             0.0000000000             100             # K
    0.0000000000             0.0000000000             0.0000000000             1             # G
    kpoint_label              G,M',K',G,M,K,G
}

SPIN_TEXTURE
{
    band_range                35 38
    kpoint_mode               line
    kpoint_num                7
    high_symmetry_kpoint
    0.0000000000             0.0000000000             0.0000000000             100             # G
    0.0000000000             0.5000000000             0.0000000000             100             # M'
    0.3333333333             0.6666666666             0.0000000000             100             # K'
    0.0000000000             0.0000000000             0.0000000000             100             # G
    0.5000000000             0.5000000000             0.0000000000             100             # M
    0.6666666666             0.3333333333             0.0000000000             100             # K
    0.0000000000             0.0000000000             0.0000000000             1             # G
    kpoint_label              G,M',K',G,M,K,G
}

```

关于功能模块 SPIN_TEXTURE 完整设置参数为：

- **band_range**, int 类型，有两个值，设置计算的能带范围，从 1 开始计数，例如 1 20，计算第 1 到第 20 共 21 条能带。
- **kpoint_mode**, str 类型，指定 k 点采样模式，存在此参数时会有额外的设置参数。

如果某个功能模块中存在 **kpoint_mode** 参数时，会添加额外的设置参数：

- 当 **kpoint_mode** 为 **mp** 时，新增参数：
 - **k_start**, float 类型，有三个值，确定指定布里渊区范围的原点，默认为 0.0 0.0 0.0。
 - **k_vect1**, float 类型，有三个值，确定指定布里渊区范围的展开矢量 1，默认为 1.0 0.0 0.0。
 - **k_vect2**, float 类型，有三个值，确定指定布里渊区范围的展开矢量 2，默认为 0.0 1.0 0.0。

- **k_vect3**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 3, 默认为 0.0 0.0 1.0。
- **mp_grid**, int 类型, 有三个值, 均匀划分指定布里渊区范围的网格数目, 没有默认值。
- 当 **kpoint_mode** 为 **line** 时, 新增参数:
 - **kpoint_num**, int 类型, 指定高对称点的数目。
 - **high_symmetry_kpoint**, float 类型, 每个高对称点有四个值, 设置高对称点坐标及插值数。
 - **kpoint_label**, str 类型, 数目与 **kpoint_num** 保持一致, 指定每一个高对称点的 label。
- 当 **kpoint_mode** 为 **direct** 时, 新增参数:
 - **kpoint_num**, int 类型, 指定 k 点的数目。
 - **kpoint_direct_coor**, float 类型, 每一个 k 点三个值, 指定 k 点的坐标。

在准备好了 PYATB 各种输入文件之后, 我们可以运行目录下的 **run_pyatb.sh** 脚本来进行 PYATB 的计算。

```
#!/bin/bash

# 加载 conda 环境
source ~/software/miniconda3/bin/activate
conda activate pyatb

# 运行 pyatb
export OMP_NUM_THREADS=1 # 设置 OpenMP 线程数为 1
num_mpi=8 # 设置 MPI 进程数
mpirun -n $num_mpi pyatb > job.log 2> job.err
```

5.2 结果展示

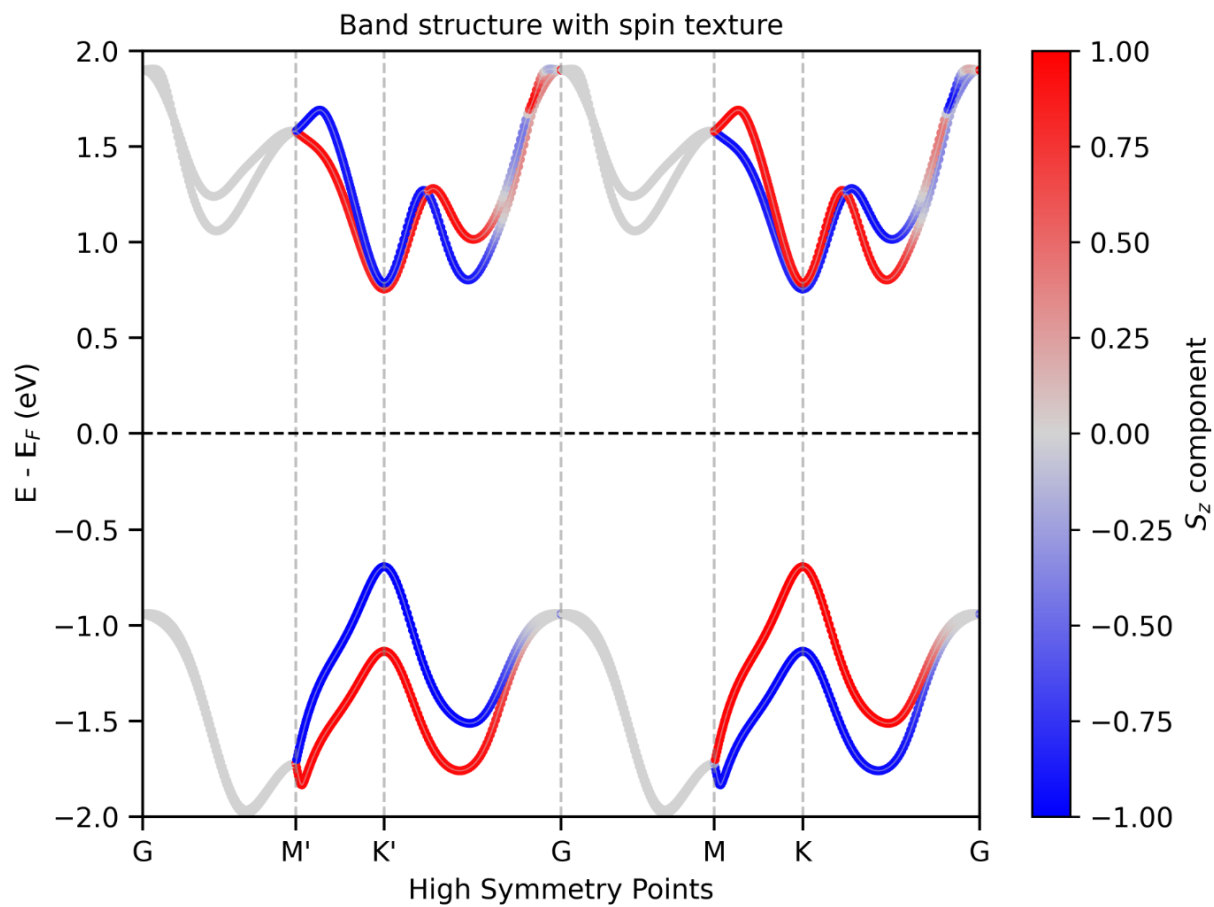
运行 PYATB 后, 所有计算结果将保存在 Out/ 文件夹中。每个功能模块对应一个子文件夹, 便于结构化管理与后续分析

5.2.1 SPIN_TEXTURE

Out/Spin_Texture 文件夹包含自旋纹理相关的输出, 相应文件为:

- **spin_texture_x.dat**, **spin_texture_y.dat**, **spin_texture_z.dat**: 记录不同 k 点, 不同能带的 Spin texture。每一行对应一个 k 点, 每一列为不同的能带。
- **kpt.dat**: 记录所有计算使用的 k 点坐标。
- **plot_spintexture_line.py**: 绘图脚本。

我们选择高对称线模式绘制能带的 Spin texture, 运行 **plot_spintexture_line.py** 后可以得到如下 $\langle S_z \rangle$ 的结果:



6.1 介绍

在具有时间反演对称性的体系中， Z_2 拓扑不变量可用于区分拓扑绝缘体与平庸绝缘体。Wilson loop 方法是一种基于占据态 Berry 相位演化的几何方法，广泛用于 Z_2 数的数值计算。

Wilson loop 的表达式为：

$$W_n(\mathbf{k}_2) = \frac{i}{2\pi} \int_0^{2\pi} d\mathbf{k}_1 \langle u_{n,\mathbf{k}_1,\mathbf{k}_2} | \partial_{\mathbf{k}_1} | u_{n,\mathbf{k}_1,\mathbf{k}_2} \rangle.$$

通过计算 Wilson loop，可以追踪 Wannier center 的演化，从而判断系统的拓扑性质。对于三维系统，我们在六个时间反演对称平面（TRIM planes）上进行计算，这些平面为：

- $k_i = 0.0$ 和 $k_i = 0.5$ ，其中 $i = x, y, z$ 。

在每个平面上，绘制 Wannier center 随 \mathbf{k} 点变化的轨迹。如果轨迹奇数次穿过参考线，则该平面 $Z_2 = 1$ ；否则为 0。

最终 Z_2 拓扑指标 ($\nu_0, \nu_1, \nu_2, \nu_3$) 由以下公式给出：

$$\begin{aligned} \nu_0 &= Z_2(k_i = 0) + Z_2(k_i = 0.5) \mod 2 \\ \nu_i &= Z_2(k_i = 0.5) \end{aligned}$$

注意，在选择平面时，因为时间反演对称的原因，我们只需选择平面的一半即可。

参考文献：

- Jin, G., Pang, H., Ji, Y., Dai, Z. & He, L. PYATB: An efficient Python package for electronic structure calculations using ab initio tight-binding model. Computer Physics Communications 291, 108844 (2023).

下面以三维拓扑绝缘体 Bi_2Se_3 为示例，演示如何使用 ABACUS 与 PYATB 联合进行 Wilson loop 计算。相关文件位于 tutorial/Bi2Se3_wilsonloop 目录下，其中包含 7 个子文件夹：abacus 和 pyatb_kx=0.0、pyatb_kx=0.5、pyatb_ky=0.0、pyatb_ky=0.5、pyatb_kz=0.0、pyatb_kz=0.5，分别提供了第一性原理计算与后续紧束缚模型计算所需的输入文件。

6.2 计算步骤

快速测试步骤：

```
$ cd tutorial/Bi2Se3_wilsonloop/abacus
$ ./run_abacus.sh
$ cd ../pyatb_kx\=0.0/
$ ./run_pyatb.sh
```

6.2.1 第一步，ABACUS 自洽计算产生紧束缚模型文件

我们已经在 Bi_2Se_3 的能带结构、Fat band、PDOS 中介绍了 ABACUS 自洽计算生成 HR、SR、rR 文件。在做 Wilson loop 计算的时候，这个前置任务是一样的。

6.2.2 第二步，使用紧束缚模型文件进行 PYATB 计算

在第一步中，ABACUS 完成自洽计算后，会产生 HR、SR、rR 文件，这些文件是 PYATB 必要的输入文件。然后我们进入 tutorial/Bi2Se3_wilsonloop/pyatb_kx=0.0 文件夹，进行 $k_x = 0$ 平面的 Wilson loop 的计算。其他平面的计算步骤是完全一样的。接下来我们介绍 PYATB 的输入文件 **Input** 的内容：

```
INPUT_PARAMETERS
{
    nspin                4                # 与 ABACUS 的 nspin 保持一致
    package              ABACUS          # 指定数据来源为 ABACUS
    fermi_energy          9.5064566484
    fermi_energy_unit    eV
    HR_route             ../abacus/OUT.Bi2Se3/data-HR-sparse_SPIN0.csr # HR_
    ↪ 矩阵
    SR_route             ../abacus/OUT.Bi2Se3/data-SR-sparse_SPIN0.csr # SR_
    ↪ 矩阵
    rR_route             ../abacus/OUT.Bi2Se3/data-rR-sparse.csr      # rR_
    ↪ 矩阵
    HR_unit              Ry              # HR 单位为 Rydberg
    rR_unit              Bohr            # rR 单位为 Bohr
    max_kpoint_num       8000           # 单轮计算的最大 k 点数目
}

LATTICE
{
    lattice_constant      1.8897162
    lattice_constant_unit Bohr
    lattice_vector
    -2.069  -3.583614  0.000000
    2.069   -3.583614  0.000000
    0.000   2.389075  9.546667
}

WILSON_LOOP
{
    occ_band              78              # 占据能带数目
    # kx = 0.0 half plane
    k_start               0.0  0.0  0.0    # 确定 kx=0.0 平面的原点
    k_vect1               0.0  0.0  1.0    # 确定 kx=0.0 平面的展开矢量1，此方向计算_
```

(续下页)

(接上页)

```

↪ Wannier center
    k_vect2          0.0  0.5  0.0      # 确定  $k_x=0.0$  平面的展开矢量2, 此方向为  $\perp$ 
↪ Wannier center 的演化路径
    nk1              101                # 展开矢量1的撒点数目
    nk2              101                # 展开矢量2的撒点数目
}

```

在准备好了 PYATB 各种输入文件之后, 我们可以运行目录下的 **run_pyatb.sh** 脚本来进行 PYATB 的计算。

```

#!/bin/bash

# 加载conda环境
source ~/software/miniconda3/bin/activate
conda activate pyatb

# 运行pyatb
export OMP_NUM_THREADS=1 # 设置OpenMP线程数为1
num_mpi=8 # 设置MPI进程数
mpirun -n $num_mpi pyatb > job.log 2> job.err

```

6.3 结果展示

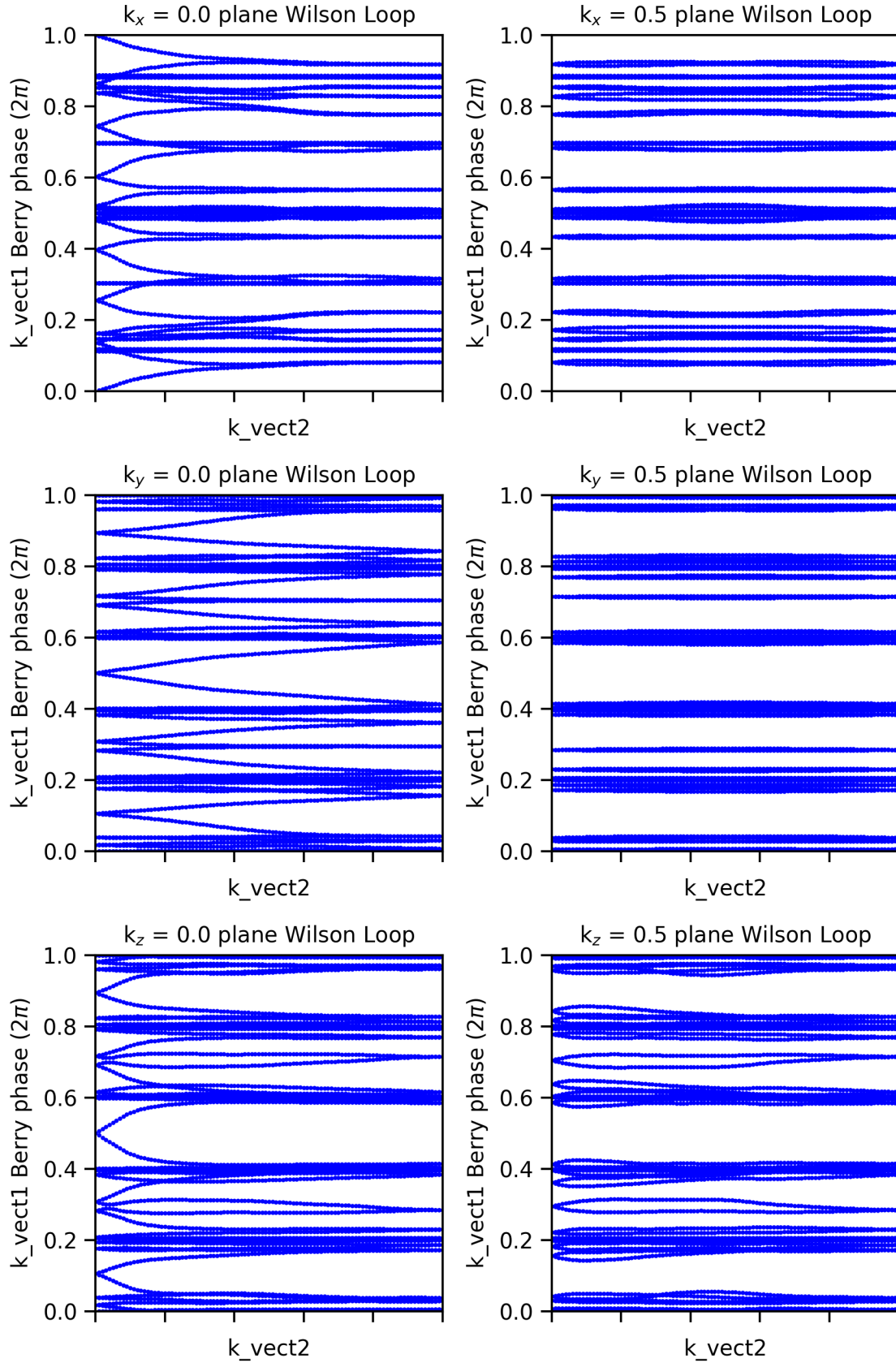
运行 PYATB 后, 所有计算结果将保存在 Out/ 文件夹中。每个功能模块对应一个子文件夹, 便于结构化管理与后续分析

6.3.1 WILSON_LOOP

Out/Wilson_Loop 文件夹包含 Wilson loop 相关的输出, 相应文件为:

- **wilson_loop.dat**: 记录 Wannier center 的演化轨迹。第一列为沿 **k_vect2** 方向的离散点编号, 其余各列分别对应每条占据能带 (共 **occ_band** 条) 的 Wannier center。
- **plot_wl.py**: 用于可视化 Wilson loop 结果的绘图脚本。

我们在 tutorial/Bi2Se3_wilsonloop 目录下提供了一个额外的绘图脚本 **plot_wl.py**, 用于读取 PYATB 对 6 个 TRIM 平面计算得到的 Wilson loop 数据, 并将它们统一绘制在同一张图中。运行该脚本后可得到如下图所示的结果:



7.1 介绍

在材料科学与凝聚态物理中，光学响应函数是研究材料与电磁波相互作用的重要工具。它不仅揭示了材料的电子结构特征，还能预测其在不同频率下的光学行为，如吸收、反射、透射和能量损耗等。

材料对外加电磁场的响应可通过两类核心物理量来描述：

- 光学电导率 (Optical Conductivity)：刻画电流响应与电场之间的关系；
- 介电函数 (Dielectric Function)：反映材料的极化性质，是计算折射率、吸收系数、反射率等线性光学性质的基础。

在微扰近似下，一阶光电导率可以由 Kubo-Greenwood 公式给出：

$$\sigma_{\alpha\beta}(\hbar\omega) = -\frac{ie^2\hbar}{NV_{\text{cell}}} \sum_{\mathbf{k}} \sum_{n,m} \left(\frac{f_{n\mathbf{k}} - f_{m\mathbf{k}}}{E_{n\mathbf{k}} - E_{m\mathbf{k}}} \right) \frac{\langle \psi_{n\mathbf{k}} | v_{\alpha} | \psi_{m\mathbf{k}} \rangle \langle \psi_{m\mathbf{k}} | v_{\beta} | \psi_{n\mathbf{k}} \rangle}{\hbar\omega + E_{n\mathbf{k}} - E_{m\mathbf{k}} + i\eta},$$

其中：

- $E_{n\mathbf{k}}$ 为能带能量， $f_{n\mathbf{k}}$ 为费米分布函数；
- $|\psi_{n\mathbf{k}}\rangle$ 为布洛赫态， v_{α} 为动量方向 α 上的速度算符；
- η 是能级展宽参数；

材料的介电函数 $\varepsilon(\omega)$ 是线性响应理论中的核心物理量，反映了材料极化对外加电场的响应特性。其虚部与实际电子跃迁过程直接相关，通常由如下公式计算：

$$\varepsilon_i^{\alpha\beta}(\omega) = -\frac{e^2\pi}{\varepsilon_0\hbar} \int \frac{d\mathbf{k}}{(2\pi)^3} \sum_{n,m} f_{nm} r_{nm}^{\alpha} r_{mn}^{\beta} \delta(\omega_{mn} - \omega),$$

其中：

- ε_0 为真空介电常数；
- $f_{nm} = f_n - f_m$ ；

- r_{nm}^α 是带间的贝里联络；
- $\omega_{mn} = (E_m - E_n)/\hbar$ ；

实部则可通过 Kramer-Kronig 变换获得：

$$\varepsilon_r^{\alpha\beta}(\omega) = \delta_{\alpha\beta} + \frac{2}{\pi} \mathcal{P} \int_0^\infty d\omega' \frac{\omega' \varepsilon_i^{\alpha\beta}(\omega')}{\omega'^2 - \omega^2},$$

其中 \mathcal{P} 表示主值积分。

一旦得到介电函数的实部 $\varepsilon_1 = \text{Re}(\varepsilon)$ 与虚部 $\varepsilon_2 = \text{Im}(\varepsilon)$ ，即可进一步计算其他重要的光学响应函数，包括：

- 折射率 $n(\omega)$
- 消光系数 $\kappa(\omega)$
- 吸收系数 $\alpha(\omega)$
- 能量损失函数 $L(\omega)$
- 反射率 $R(\omega)$

它们的表达式如下：

$$\begin{aligned} n(\omega) &= \left[\frac{\sqrt{\varepsilon_1^2 + \varepsilon_2^2} + \varepsilon_1}{2} \right]^{1/2} \\ \kappa(\omega) &= \left[\frac{\sqrt{\varepsilon_1^2 + \varepsilon_2^2} - \varepsilon_1}{2} \right]^{1/2} \\ \alpha(\omega) &= \frac{\sqrt{2}\omega}{c} \left[\sqrt{\varepsilon_1^2 + \varepsilon_2^2} - \varepsilon_1 \right]^{1/2} \\ L(\omega) &= \text{Im} \left(\frac{-1}{\varepsilon(\omega)} \right) = \frac{\varepsilon_2}{\varepsilon_1^2 + \varepsilon_2^2} \\ R(\omega) &= \frac{(n-1)^2 + \kappa^2}{(n+1)^2 + \kappa^2} \end{aligned}$$

参考文献：

- Jin, G., Pang, H., Ji, Y., Dai, Z. & He, L. PYATB: An efficient Python package for electronic structure calculations using ab initio tight-binding model. *Computer Physics Communications* 291, 108844 (2023).

下面以钙钛矿 CsPbI_3 为示例，演示如何使用 ABACUS 与 PYATB 联合进行 OPTICAL_CONDUCTIVITY 计算。相关文件位于 `tutorial/CsPbI3_optical` 目录下，其中包含两个子文件夹：`abacus` 和 `pyatb`，分别提供了第一性原理计算与后续紧束缚模型计算所需的输入文件。

7.2 计算步骤

快速测试步骤：

```
$ cd tutorial/CsPbI3_optical/abacus
$ ./run_abacus.sh
$ cd ../pyatb
$ ./run_pyatb.sh
$ python plot_absorption_lambda.py
```

7.2.1 第一步，ABACUS 自洽计算产生紧束缚模型文件

在 tutorial/CsPbI3_optical/abacus 文件夹中，我们已经准备好了 ABACUS 自洽计算的输入文件，包括：

- 控制文件：INPUT
- 结构文件：STRU
- 赝势文件：*.upf
- 原子轨道基组文件：*.orb
- K 点采样文件：KPT

其中，INPUT 文件的核心参数如下，已对关键设置作出注释，其他细节可参考 ABACUS 官方文档：https://abacus.deepmodeling.com/en/latest/advanced/input_files/input-main.html

```
INPUT_PARAMETERS

# System variables
suffix                CsPbI3
calculation            scf      # 指定计算类型为自洽计算
esolver_type          ksdfc
symmetry               -1      # 完全关闭对称性分析，如果发现不收敛，可以关闭对称性
init_chg               atomic

# Input Files
pseudo_dir            ./
orbital_dir            ./

# Plane Wave
ecutwfc                100     # 平面波截断值，可以选择轨道文件中所使用的截断能量

# Electronic structure
basis_type             lcao     # 使用原子轨道作为基组
ks_solver              genelpa
nspin                  1
smearing_method        gauss   # 电子占据模糊方法
smearing_sigma         0.01
mixing_type            pulay    # 设置电荷密度混合方法
mixing_beta            0.7
scf_nmax               200     # 最大电子收敛步数
scf_thr                1e-8    # 电荷密度收敛精度
lspinorb               0       # 关闭自旋轨道耦合
noncolin               0       # 关闭自旋非共线

# Output Variables
out_chg                1       # 输出自洽后的电荷密度
out_mat_hs2            1       # 输出 HR、SR 矩阵
out_mat_r              1       # 输出 rR 矩阵
```

执行 ABACUS 自洽计算后，若设置了 `out_mat_hs2 = 1` 和 `out_mat_r = 1`，将在 `OUT.*` 目录中生成所需紧束缚模型文件：

- **data-HR-sparse_SPIN0.csr**：实空间哈密顿量
- **data-SR-sparse_SPIN0.csr**：重叠矩阵
- **data-rR-sparse.csr**：位置偶极矩阵

注 1: 若设置 `nspin = 2` (即自旋极化计算), 则还会生成 **data-HR-sparse_SPIN1.csr** 文件, 用于描述第二个自旋通道。注 2: ABACUS 生成紧束缚模型文件, 必须在 `lcao` 基组下, 即 `basis_type lcao`。注 3: 随着 ABACUS 版本的更新, 部分紧束缚模型文件的文件名称可能会发生变化。

在准备好了 ABACUS 各种输入文件之后, 我们可以运行目录下的 **run_abacus.sh** 脚本来进行 ABACUS 的自洽计算。

```
#!/bin/bash

# 加载ABACUS环境
source ~/softwares/abacus-develop-LTSv3.10.0/toolchain/abacus_env.sh

# 运行abacus
export OMP_NUM_THREADS=1 # 设置OpenMP线程数为1
num_mpi=8 # 设置MPI进程数
mpirun -n $num_mpi abacus > job.log 2> job.err
```

请根据具体的计算节点配置适当调整 `OMP_NUM_THREADS` 与 `num_mpi`, 以充分利用本地硬件资源, 获得更高的并行效率。

7.2.2 第二步, 使用紧束缚模型文件进行 PYATB 计算

在第一步中, ABACUS 完成自洽计算后, 会产生 **HR**、**SR**、**rR** 文件, 这些文件是 PYATB 必要的输入文件。然后我们进入 `tutorial/CsPbI3_optical/pyatb` 文件夹, 进行 **Optical Conductivity** 的计算。接下来我们介绍 PYATB 的输入文件 **Input** 的内容:

```
INPUT_PARAMETERS
{
  nspin                1                # 与 ABACUS 的 nspin 保持一致
  package              ABACUS           # 指定数据来源为 ABACUS
  fermi_energy         4.2808350436
  fermi_energy_unit    eV
  HR_route             ../abacus/OUT.CsPbI3/data-HR-sparse_SPIN0.csr # HR 矩阵
  SR_route             ../abacus/OUT.CsPbI3/data-SR-sparse_SPIN0.csr # SR 矩阵
  rR_route             ../abacus/OUT.CsPbI3/data-rR-sparse.csr      # rR 矩阵
  HR_unit              Ry               # HR 单位为 Rydberg
  rR_unit              Bohr             # rR 单位为 Bohr
}

LATTICE
{
  lattice_constant      1.8897261258369282
  lattice_constant_unit Bohr
  lattice_vector
    6.2894000000      0.0000000000      0.0000000000
    0.0000000000      6.2894000000      0.0000000000
    0.0000000000      0.0000000000      6.2894000000
}

BAND_STRUCTURE
{
  wf_collect            0                # 是否输出波函数的展开系数
  kpoint_mode           line            # 设置 k_
  ↳点采样模式为高对称路径, 有 line, mp, direct 三种
  kpoint_num            6                #_
  ↳设置路径中的高对称点数
```

(续下页)

(接上页)

```

kpoint_label          G, X, M, G, R, X # 设置高对称点名称
high_symmetry_kpoint  # 设置高对称点坐标及插值数
0.00000 0.00000 0.0000 20 # G
0.00000 0.50000 0.0000 20 # X
0.50000 0.50000 0.0000 25 # M
0.00000 0.00000 0.0000 30 # G
0.50000 0.50000 0.5000 25 # R
0.00000 0.50000 0.0000 1  # X
}

OPTICAL_CONDUCTIVITY
{
  occ_band      22          # 总的占据能带数目
  omega         0.5  10     # 光子能量范围
  domega        0.01       # 光子能量间隔
  eta           0.2        # 能级展宽参数, 影响 smearing
  grid          30 30 30   # 划分布里渊区网格数
}

```

关于功能模块 SHIFT_CURRENT 完整设置参数为：

- **occ_band**, int 类型，半导体和绝缘体的占据能带数目。
- **omega**, float 类型，有两个值，设置光子能量范围。
- **domega**, float 类型，光子能量间隔。
- **eta**, float 类型，能级展宽参数，影响曲线的 smearing。
- **grid**, int 类型，三个值。划分布里渊区的均匀网格数。

在准备好了 PYATB 各种输入文件之后，我们可以运行目录下的 **run_pyatb.sh** 脚本来进行 PYATB 的计算。

```

#!/bin/bash

# 加载 conda 环境
source ~/software/miniconda3/bin/activate
conda activate pyatb

# 运行 pyatb
export OMP_NUM_THREADS=1 # 设置 OpenMP 线程数为 1
num_mpi=8 # 设置 MPI 进程数
mpirun -n $num_mpi pyatb > job.log 2> job.err

```

7.3 结果展示

运行 PYATB 后，所有计算结果将保存在 Out/ 文件夹中。每个功能模块对应一个子文件夹，便于结构化管理与后续分析

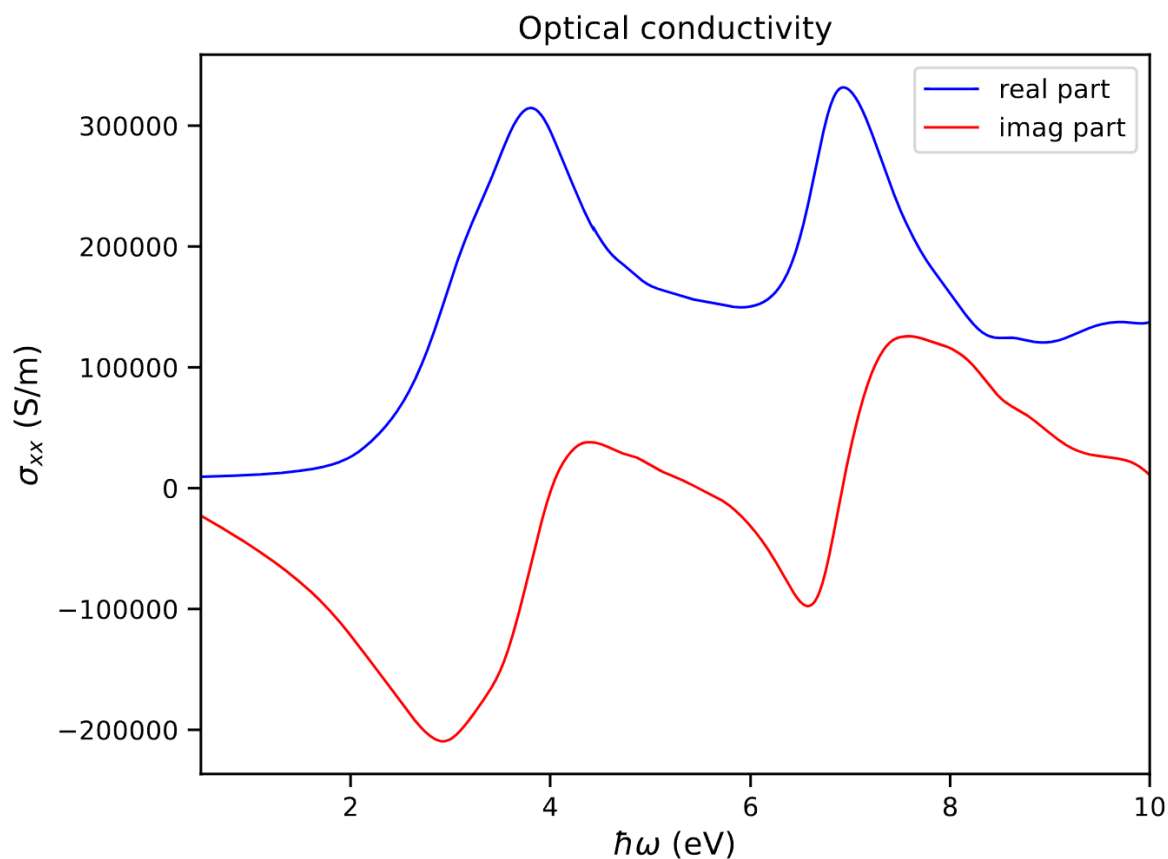
7.3.1 OPTICAL_CONDUCTIVITY

Out/Optical_Conductivity 文件夹包含光电导和介电函数相关的输出，相应文件为：

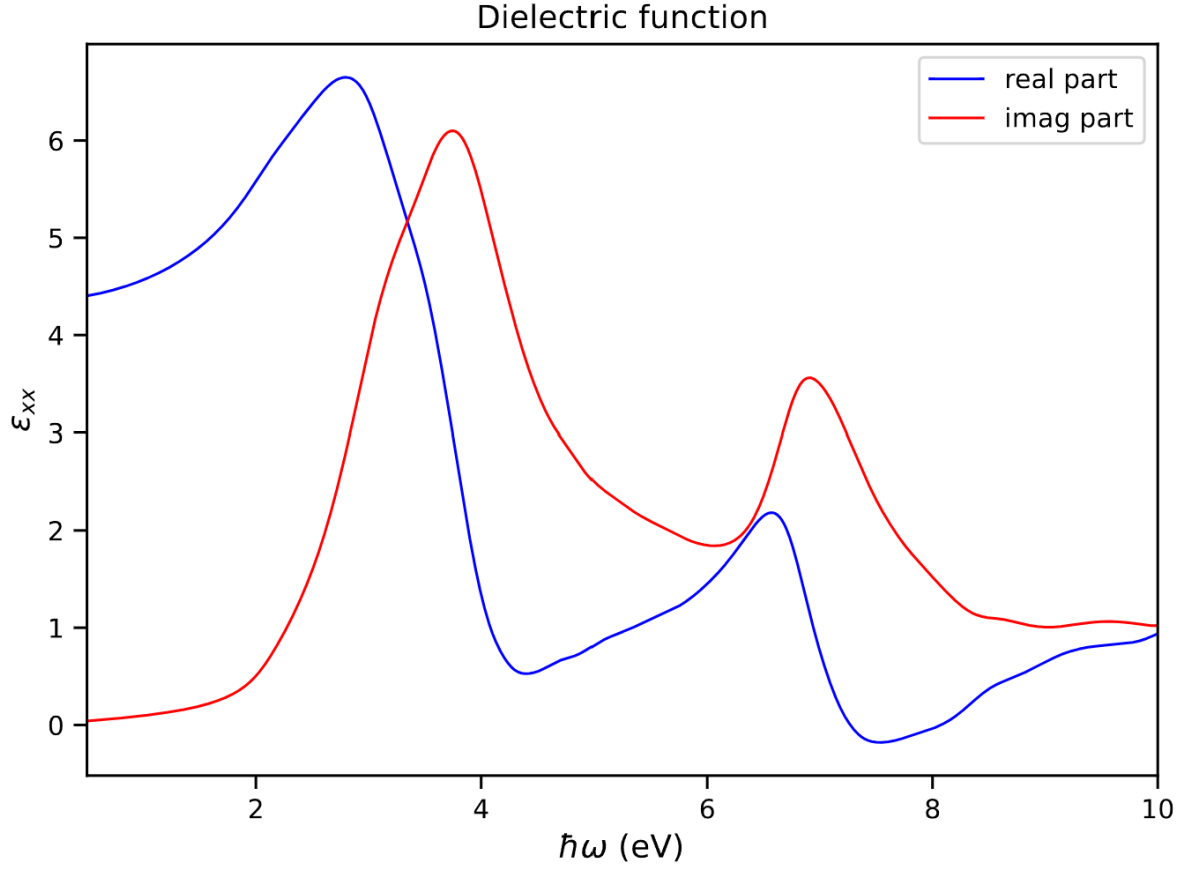
- **optical_conductivity_real_part.dat**, **optical_conductivity_imag_part.dat**: 记录光电导张量的实部与虚部。
- **dielectric_function_real_part.dat**, **dielectric_function_imag_part.dat**: 记录介电函数张量的实部与虚部。
- **plot_optical.py**: 绘制不同方向上的光电导和介电函数的脚本。
- **plot_absorption.py**: 根据介电函数计算吸收系数的绘图脚本。

在钙钛矿材料 CsPbI_3 中，由于晶体结构的对称性，其沿对角线方向的光学电导率与介电函数张量具有非零且等价的分量。下图展示了 PYATB 计算得到的 xx 方向上光学响应结果，包括：

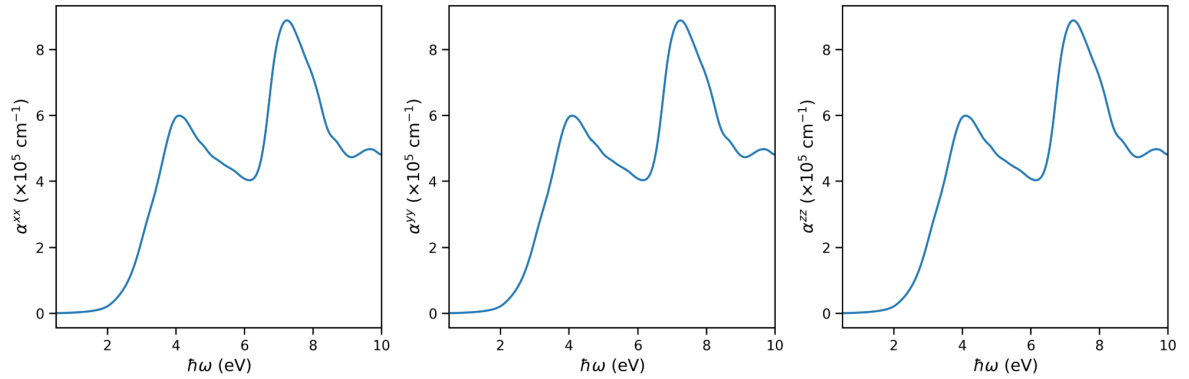
- 光学电导率实部与虚部：



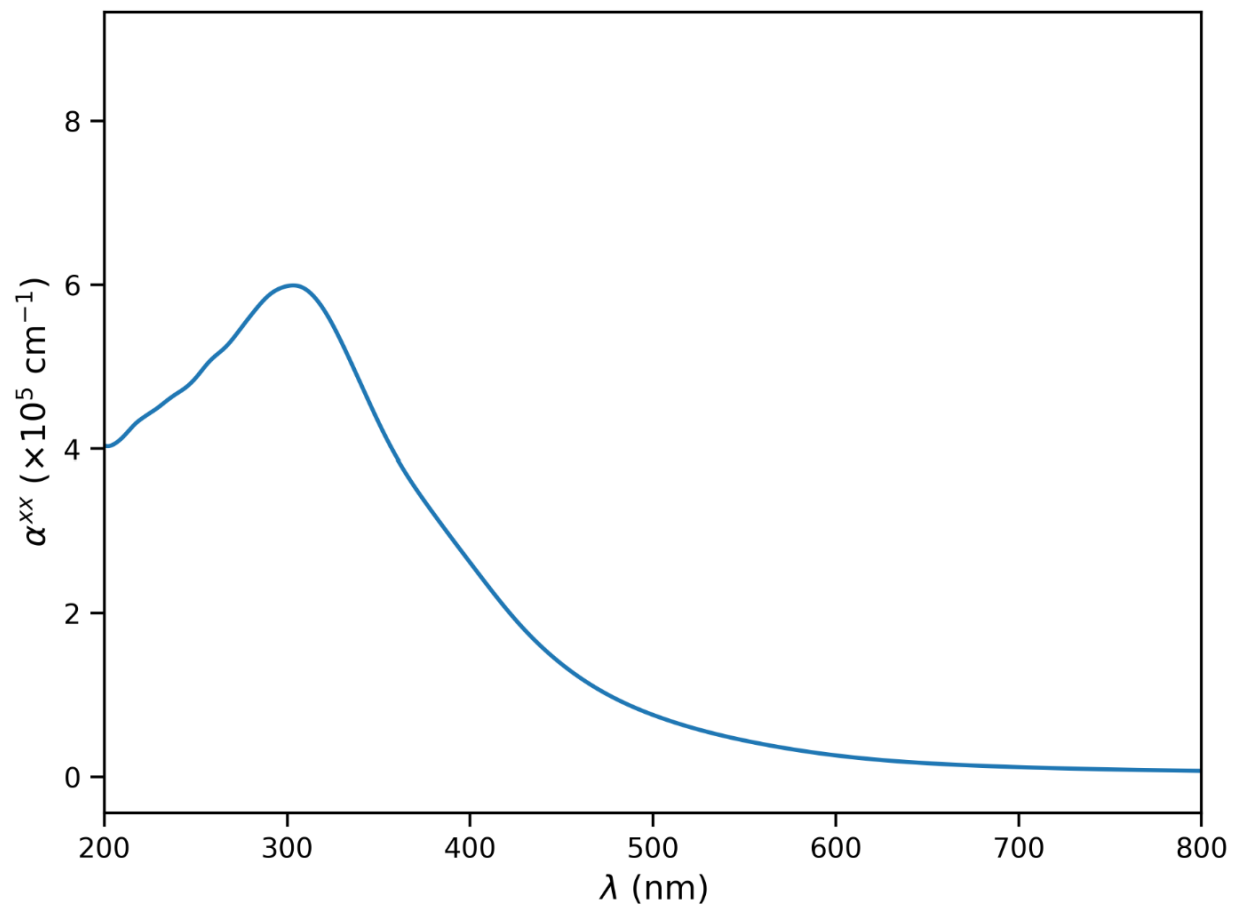
- 介电函数实部与虚部：



- 吸收系数 $\alpha(\hbar\omega)$ 随光子能量变化:



此外，我们在 `tutorial/CsPbI3_optical/pyatb` 目录下提供了绘图脚本 `plot_absorption_lambda.py`，可将横轴从光子能量（单位：eV）转换为波长（单位：nm），便于与实验吸收谱进行比较。对应结果如下所示：



8.1 介绍

反常霍尔电导率 (Anomalous Hall Conductivity, AHC) 和反常能斯特电导率 (Anomalous Nernst Conductivity, ANC) 是由电子波函数的几何性质 (特别是 Berry 曲率) 引起的横向输运响应。它们在拓扑材料、自旋轨道耦合系统、铁磁金属等中具有重要物理意义。

在零外加磁场下, 这两种输运现象分别描述如下:

- **AHC** 描述的是在外电场作用下, 系统在横向 (垂直于电场方向) 产生的反常霍尔电导:

$$\sigma_{xy}(\varepsilon) = -\frac{e^2}{\hbar} \sum_n^{\text{occ}} \int_{\text{BZ}} \frac{d^3k}{(2\pi)^3} f_{n\mathbf{k}}(\varepsilon) \Omega_{n,z}(\mathbf{k}),$$

其中 $\Omega_{n,z}(\mathbf{k})$ 为第 n 条能带在 \mathbf{k} 点的 Berry 曲率, $f_{n\mathbf{k}}$ 为 Fermi-Dirac 分布函数, ε 为费米能。

- **ANC** 则描述在温度梯度存在时, 系统产生的横向热电流响应, 反常能斯特系数为:

$$\alpha_{xy}(\mu, T) = \frac{1}{e} \int d\varepsilon \sigma_{xy}(\varepsilon) \Big|_{T=0} \frac{\varepsilon - \mu}{T} \left(-\frac{\partial f(\mu)}{\partial \varepsilon} \right),$$

其中 μ 是化学势, T 为温度。

参考文献:

- Sawahata, H., Yamaguchi, N., Minami, S. & Ishii, F. First-principles calculation of anomalous Hall and Nernst conductivity by local Berry phase. Phys. Rev. B 107, 024404 (2023).

下面以二维铁磁材料 FeCl_2 为示例, 演示如何使用 ABACUS 与 PYATB 联合进行 AHC 和 ANC 的计算。相关文件位于 `tutorial/FeCl2_AHC_ANC` 目录下, 其中包含两个子文件夹: `abacus` 和 `pyatb`, 分别提供了第一性原理计算与后续紧束缚模型计算所需的输入文件。

8.2 计算步骤

快速测试步骤：

```
$ cd tutorial/FeCl2_AHC_ANC/abacus
$ ./run_abacus.sh
$ cd ../pyatb
$ ./run_pyatb.sh
```

8.2.1 第一步，ABACUS 自洽计算产生紧束缚模型文件

在 tutorial/FeCl2_AHC_ANC/abacus 文件夹中，我们已经准备好了 ABACUS 自洽计算的输入文件，包括：

- 控制文件：INPUT
- 结构文件：STRU
- 赝势文件：*.upf
- 原子轨道基组文件：*.orb
- K 点采样文件：KPT

其中，INPUT 文件的核心参数如下，已对关键设置作出注释，其他细节可参考 ABACUS 官方文档：https://abacus.deepmodeling.com/en/latest/advanced/input_files/input-main.html

```
INPUT_PARAMETERS

# System variables
suffix          FeCl2
calculation      scf          # 指定计算类型为自洽计算
esolver_type     ksdftr
symmetry         -1          # 完全关闭对称性分析，如果发现不收敛，可以关闭对称性
init_chg        atomic

# Input Files
pseudo_dir       ./
orbital_dir      ./

# Plane Wave
ecutwfc          100          # 平面波截断值，可以选择轨道文件中所使用的截断能量

# Electronic structure
basis_type       lcao          # 使用原子轨道作为基组
ks_solver        genelpa
nspin            4
smearing_method  gauss        # 电子占据模糊方法
smearing_sigma   0.001
mixing_type       pulay        # 设置电荷密度混合方法
mixing_beta      0.1
mixing_gg0       3.0
scf_nmax         200          # 最大电子收敛步数
scf_thr          1e-8         # 电荷密度收敛精度
lspinorb         1           # 开启自旋轨道耦合
noncolin         0           # 关闭自旋非共线
```

(续下页)

(接上页)

```
# Output Variables
out_chg          1          # 输出自洽后的电荷密度
out_mat_hs2      1          # 输出 HR、SR 矩阵
out_mat_r        1          # 输出 rR 矩阵
```

执行 ABACUS 自洽计算后，若设置了 `out_mat_hs2 = 1` 和 `out_mat_r = 1`，将在 `OUT.*` 目录中生成所需紧束缚模型文件：

- **data-HR-sparse_SPIN0.csr**：实空间哈密顿量
- **data-SR-sparse_SPIN0.csr**：重叠矩阵
- **data-rR-sparse.csr**：位置偶极矩阵

注 1：若设置 `nspin = 2`（即自旋极化计算），则还会生成 **data-HR-sparse_SPIN1.csr** 文件，用于描述第二个自旋通道。注 2：ABACUS 生成紧束缚模型文件，必须在 `lcao` 基组下，即 `basis_type lcao`。注 3：随着 ABACUS 版本的更新，部分紧束缚模型文件的文件名称可能会发生变化。

在准备好了 ABACUS 各种输入文件之后，我们可以运行目录下的 **run_abacus.sh** 脚本来进行 ABACUS 的自洽计算。

```
#!/bin/bash

# 加载 ABACUS 环境
source ~/softwares/abacus-develop-LTSv3.10.0/toolchain/abacus_env.sh

# 运行 abacus
export OMP_NUM_THREADS=1 # 设置 OpenMP 线程数为 1
num_mpi=8 # 设置 MPI 进程数
mpirun -n $num_mpi abacus > job.log 2> job.err
```

请根据具体的计算节点配置适当调整 `OMP_NUM_THREADS` 与 `num_mpi`，以充分利用本地硬件资源，获得更高的并行效率。

8.2.2 第二步，使用紧束缚模型文件进行 PYATB 计算

在第一步中，ABACUS 完成自洽计算后，会产生 `HR`、`SR`、`rR` 文件，这些文件是 PYATB 必要的输入文件。然后我们进入 `tutorial/FeCl2_AHC_ANC/pyatb` 文件夹，进行 AHC 和 ANC 的计算。接下来我们介绍 PYATB 的输入文件 **Input** 的内容：

```
INPUT_PARAMETERS
{
    nspin          4          # 与 ABACUS 的 nspin 保持一致
    package        ABACUS    # 指定数据来源为 ABACUS
    fermi_energy    -0.82497406093
    fermi_energy_unit eV
    HR_route        ../abacus/OUT.FeCl2/data-HR-sparse_SPIN0.csr # HR_
    ↪ 矩阵
    SR_route        ../abacus/OUT.FeCl2/data-SR-sparse_SPIN0.csr # SR_
    ↪ 矩阵
    rR_route        ../abacus/OUT.FeCl2/data-rR-sparse.csr      # rR_
    ↪ 矩阵
    HR_unit         Ry          # HR 单位为 Rydberg
    rR_unit         Bohr        # rR 单位为 Bohr
}
```

(续下页)

(接上页)

```

LATTICE
{
    lattice_constant          1.8897162
    lattice_constant_unit     Bohr
    lattice_vector
    3.4750    0.0000    0.00000
    -1.7375    3.0094    0.00000
    0.0000    0.0000    17.53999
}

BAND_STRUCTURE
{
    wf_collect                0          # 是否输出波函数的展开系数
    kpoint_mode               line        # 设置  $k_{\parallel}$ 
    ↪点采样模式为高对称路径，有 line, mp, direct 三种
    kpoint_num                4          # 设置路径中的高对称点数
    high_symmetry_kpoint
    0.5          0          0          200 # M
    0            0          0          200 # G
    0.3333       0.3333    0.3333    200 # K
    0.5          0          0          1   # M
    kpoint_label             M, G, K, M  # 设置高对称点名称
}

ANC
{
    fermi_range               -1.0 1.0    # 以 fermi_energy
    ↪为中心，设置费米能范围 (eV)
    de                       0.01        # 费米能范围能量间隔
    eta                      0.01        # 处理简并时的展宽 (eV)
    integrate_grid            200 200 1   # 布里渊区均匀网格数
}

```

关于功能模块 ANC 完整设置参数为：

- **fermi_range**, float 类型，有两个值，以 *fermi_energy* 为中心，设置费米能范围 (eV)。
- **de**, float 类型，费米能范围能量间隔 (eV)。
- **eta**, float 类型。使用 Kubo 公式计算 Berry curvature 的时候，分母的能量差 $E_{n\mathbf{k}} - E_{m\mathbf{k}} \rightarrow E_{n\mathbf{k}} - E_{m\mathbf{k}} + i\eta$ ，避免简并导致的数值问题。
- **integrate_grid**, int 类型，三个值。划分布里渊区的均匀网格数。

在准备好了 PYATB 各种输入文件之后，我们可以运行目录下的 **run_pyatb.sh** 脚本来进行 PYATB 的计算。

```

#!/bin/bash

# 加载 conda 环境
source ~/software/miniconda3/bin/activate
conda activate pyatb

# 运行 pyatb
export OMP_NUM_THREADS=1 # 设置 OpenMP 线程数为 1
num_mpi=8 # 设置 MPI 进程数
mpirun -n $num_mpi pyatb > job.log 2> job.err

```

8.3 结果展示

运行 PYATB 后，所有计算结果将保存在 Out/ 文件夹中。每个功能模块对应一个子文件夹，便于结构化管理与后续分析。

8.3.1 ANC

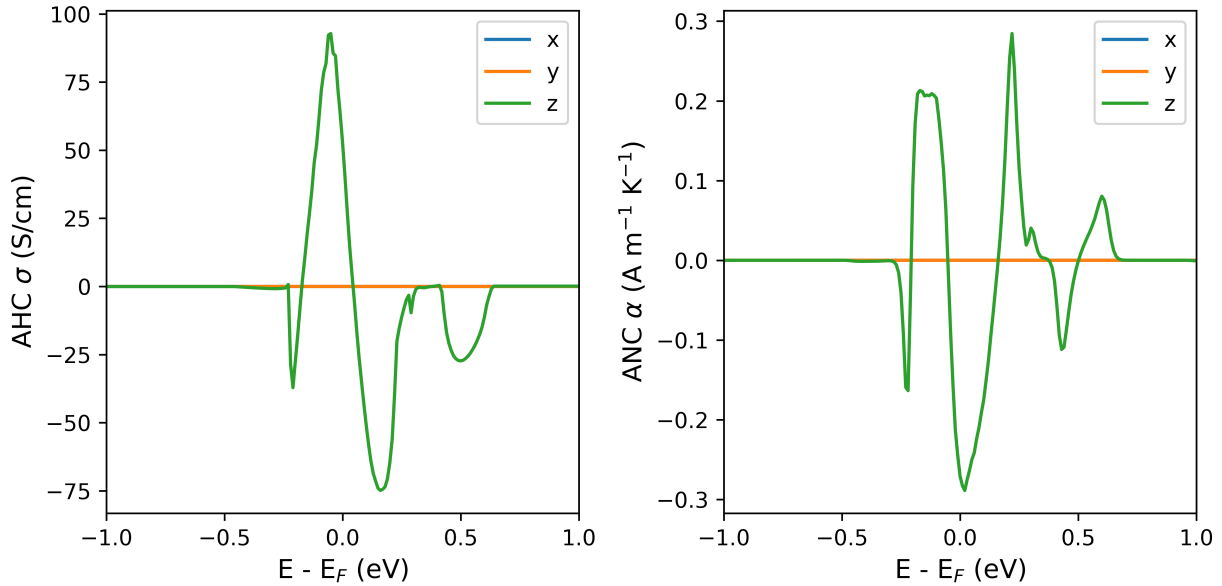
Out/ANC 文件夹包含 AHC 和 ANC 相关的输出，相应文件为：

- **sigma_mu.dat**：不同费米能下的反常霍尔电导。单位是 S/cm。
- **plot_anc.py**：绘制 AHC 和 ANC 的脚本。通过修改脚本变量，计算不同温度下的 ANC。

在 tutorial/FeCl2_AHC_ANC/pyatb/Out/ANC/plot_anc.py 绘图脚本中：

- **temperature** 变量，定义温度。
- **is_plot_2D** 变量，是否绘制二维的 AHC 和 ANC，单位会进行转换。并假设材料厚度方向为 z 方向。

不修改脚本，执行后可以得到下面的三维条件下的 AHC 和 ANC 结果：



SHIFT_CURRENT

Shift current (位移电流) 是非线性光电响应中的一个重要现象, 属于体光伏效应 (Bulk Photovoltaic Effect, BPVE) 的一种本征贡献。它仅出现在**缺乏中心反演对称性**的材料中, 其产生机制源于光照引起的电子跃迁过程中, 电子波函数的几何相位结构 (如 Berry connection 与其导数) 导致的实空间电荷中心偏移, 从而形成净电流。

在单色光照射下, 电场形式为 $\mathbf{E}(t) = \mathbf{E}(\omega)e^{i\omega t} + \mathbf{E}(-\omega)e^{-i\omega t}$, 无反演对称性的材料中会产生如下的光生电流密度:

$$J^a = 2 \sigma^{abc}(0; \omega, -\omega) E_b(\omega) E_c(-\omega),$$

其中 $a, b, c \in x, y, z$, $\sigma^{abc}(0; \omega, -\omega)$ 为三阶的非线性光电导张量, 描述了电场作用下第 a 方向上的位移电流响应。

该张量的表达式为:

$$\sigma^{abc}(0; \omega, -\omega) = \frac{\pi e^3}{\hbar^2} \int \frac{d\mathbf{k}}{8\pi^3} \sum_{n,m} f_{nm} \text{Im} [I_{mn}^{abc} + I_{mn}^{acb}] \delta(\omega_{mn} - \omega),$$

其中 $f_{nm} = f_n - f_m$ 是带间的费米分布函数差, $\omega_{mn} = (E_m - E_n)/\hbar$ 是能量差对应的跃迁频率。 $I_{mn}^{abc} = r_{mn}^b r_{nm;a}^c$ 被称为 Hermitian connection 项, 其中 r_{mn}^b 是价带到导带的带间偶极跃迁元 (Berry connection), $r_{nm;a}^c$ 是其关于波矢的协变导数, 物理上对应跃迁过程中电子在实空间中的位移 (即 shift vector)。

Shift current 的存在不依赖于外加电场或结结构, 因此在非中心对称材料中, 即使在体内部也能实现零偏压发电, 是设计新型光伏器件 (如铁电光伏、拓扑光电材料等) 的一种关键机制。

更多理论细节可参考以下文献:

- Sipe & Shkrebtii, Second-order optical response in semiconductors, Phys. Rev. B 61, 5337 (2000)

下面以二维 WS_2 为示例, 演示如何使用 ABACUS 与 PYATB 联合进行 Shift Current 计算。相关文件位于 tutorial/WS2_shift_current 目录下, 其中包含两个子文件夹: abacus 和 pyatb, 分别提供了第一性原理计算与后续紧束缚模型计算所需的输入文件。

注 1: 目前 PYATB 版本中暂时不支持金属的计算, 且 Shift Current 计算针对于存在时间反演的体系。下个版本中将支持这些计算。

9.1 计算步骤

快速测试步骤：

```
$ cd tutorial/WS2_shift_current/abacus
$ ./run_abacus.sh
$ cd ../pyatb
$ ./run_pyatb.sh
```

9.1.1 第一步，ABACUS 自洽计算产生紧束缚模型文件

在 tutorial/WS2_shift_current/abacus 文件夹中，我们已经准备好了 ABACUS 自洽计算的输入文件，包括：

- 控制文件：INPUT
- 结构文件：STRU
- 赝势文件：*.upf
- 原子轨道基组文件：*.orb
- K 点采样文件：KPT

其中，INPUT 文件的核心参数如下，已对关键设置作出注释，其他细节可参考 ABACUS 官方文档：https://abacus.deepmodeling.com/en/latest/advanced/input_files/input-main.html

```
INPUT_PARAMETERS

# System variables
suffix          WS2
calculation      scf      # 指定计算类型为自洽计算
esolver_type     ksdfc
symmetry         -1      # 完全关闭对称性分析，如果发现不收敛，可以关闭对称性
init_chg        atomic

# Input Files
pseudo_dir      ./
orbital_dir      ./

# Plane Wave
ecutwfc          100      # 平面波截断值，可以选择轨道文件中所使用的截断能量

# Electronic structure
basis_type       lcao      # 使用原子轨道作为基组
ks_solver        genelpa
nspin            1
smearing_method  gauss     # 电子占据模糊方法
smearing_sigma   0.01
mixing_type       broyden  # 设置电荷密度混合方法
mixing_beta      0.8
scf_nmax         200      # 最大电子收敛步数
scf_thr          1e-7     # 电荷密度收敛精度
lspinorb         0        # 关闭自旋轨道耦合
noncolin         0        # 关闭自旋非共线

# Output Variables
```

(续下页)

(接上页)

```

out_chg          1          # 输出自洽后的电荷密度
out_mat_hs2      1          # 输出 HR、SR 矩阵
out_mat_r        1          # 输出 rR 矩阵

```

执行 ABACUS 自洽计算后，若设置了 `out_mat_hs2 = 1` 和 `out_mat_r = 1`，将在 `OUT.*` 目录中生成所需紧束缚模型文件：

- **data-HR-sparse_SPIN0.csr**：实空间哈密顿量
- **data-SR-sparse_SPIN0.csr**：重叠矩阵
- **data-rR-sparse.csr**：位置偶极矩阵

注 1：若设置 `nspin = 2`（即自旋极化计算），则还会生成 **data-HR-sparse_SPIN1.csr** 文件，用于描述第二个自旋通道。注 2：ABACUS 生成紧束缚模型文件，必须在 `lcao` 基组下，即 `basis_type lcao`。注 3：随着 ABACUS 版本的更新，部分紧束缚模型文件的文件名称可能会发生变化。

在准备好了 ABACUS 各种输入文件之后，我们可以运行目录下的 **run_abacus.sh** 脚本来进行 ABACUS 的自洽计算。

```

#!/bin/bash

# 加载 ABACUS 环境
source ~/softwares/abacus-develop-LTSv3.10.0/toolchain/abacus_env.sh

# 运行 abacus
export OMP_NUM_THREADS=1 # 设置 OpenMP 线程数为 1
num_mpi=8 # 设置 MPI 进程数
mpirun -n $num_mpi abacus > job.log 2> job.err

```

请根据具体的计算节点配置适当调整 `OMP_NUM_THREADS` 与 `num_mpi`，以充分利用本地硬件资源，获得更高的并行效率。

9.1.2 第二步，使用紧束缚模型文件进行 PYATB 计算

在第一步中，ABACUS 完成自洽计算后，会产生 `HR`、`SR`、`rR` 文件，这些文件是 PYATB 必要的输入文件。然后我们进入 `tutorial/WS2_shift_current/pyatb` 文件夹，进行 Shift Current 的计算。接下来我们介绍 PYATB 的输入文件 **Input** 的内容：

```

INPUT_PARAMETERS
{
    nspin          1          # 与 ABACUS 的 nspin 保持一致
    package        ABACUS    # 指定数据来源为 ABACUS
    fermi_energy    0.93900603892
    fermi_energy_unit eV
    HR_route        ../abacus/OUT.WS2/data-HR-sparse_SPIN0.csr # HR 矩阵
    SR_route        ../abacus/OUT.WS2/data-SR-sparse_SPIN0.csr # SR 矩阵
    rR_route        ../abacus/OUT.WS2/data-rR-sparse.csr      # rR 矩阵
    HR_unit         Ry      # HR 单位为 Rydberg
    rR_unit         Bohr    # rR 单位为 Bohr
}

LATTICE
{
    lattice_constant 1.8897162
    lattice_constant_unit Bohr
}

```

(续下页)

(接上页)

```

lattice_vector
3.183820900165    0.0    0.0
-1.591910450082    2.757269780643    0.0
0.0    0.0    20.086904001384
}

BAND_STRUCTURE
{
    kpoint_mode      line      # 设置 k_
    ↪点采样模式为高对称路径，有line, mp, direct三种
    kpoint_num        4        # 设置路径中的高对称点数
    high_symmetry_kpoint      # 设置高对称点坐标及插值数
    0.00000000000    0.0000000000    0.0000000000 200 # G
    0.50000000000    0.0000000000    0.0000000000 200 # M
    0.33333333333    0.3333333333    0.0000000000 200 # K
    0.00000000000    0.0000000000    0.0000000000 1   # G
    kpoint_label      G,M,K,G      # 设置高对称点名称
}

SHIFT_CURRENT
{
    occ_band          13          # 总的占据能带数目
    omega              0 4        # 光子能量范围
    domega             0.01       # 光子能量间隔
    smearing_method    1          # 使用 Gauss smearing
    eta                0.02       # Gauss smearing 参数
    grid               200 200 1  # 划分布里渊区网格数
}

```

关于功能模块 SHIFT_CURRENT 完整设置参数为：

- **occ_band**, int 类型，半导体和绝缘体的占据能带数目。
- **omega**, float 类型，有两个值，设置光子能量范围。
- **domega**, float 类型，光子能量间隔。
- **smearing_method**, int 类型，指定 smearing 方法。0: no smearing, 1: Gauss smearing, 2: adaptive smearing。
- **eta**, float 类型，Gauss smearing 参数。
- **grid**, int 类型，三个值。划分布里渊区的均匀网格数。

在准备好了 PYATB 各种输入文件之后，我们可以运行目录下的 **run_pyatb.sh** 脚本来进行 PYATB 的计算。

```

#!/bin/bash

# 加载conda环境
source ~/software/miniconda3/bin/activate
conda activate pyatb

# 运行pyatb
export OMP_NUM_THREADS=1 # 设置OpenMP线程数为1
num_mpi=8 # 设置MPI进程数
mpirun -n $num_mpi pyatb > job.log 2> job.err

```

9.2 结果展示

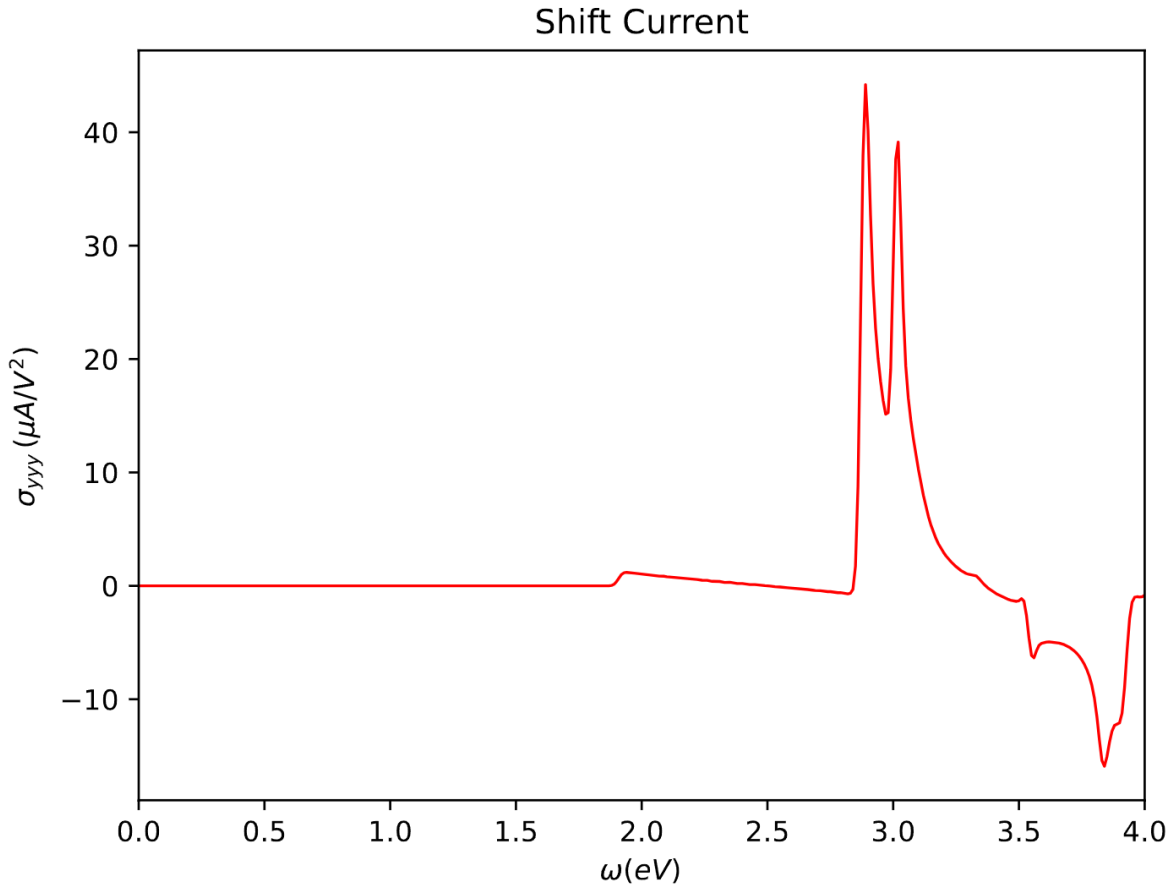
运行 PYATB 后，所有计算结果将保存在 Out/ 文件夹中。每个功能模块对应一个子文件夹，便于结构化管理与后续分析

9.2.1 SHIFT_CURRENT

Out/Shift_Current 文件夹包含位移电流相关的输出，相应文件为：

- **shift_current.dat**: 记录位移电流电导的 18 个分量数据。第一列为光子能量，后面每一列为三阶张量 18 个不等价的分量。
- **plot_shift_current.py**: 绘制不同方向位移电流电导的脚本。

由于这里的 WS_2 存在 D_{6h} 点群，因此只有 1 个非零独立分量，即 σ_{yyy} 。在 tutorial/WS2_shift_current/pyatb/Out/Shift_Current 目录下执行 **plot_shift_current.py** 脚本后，我们可以得到该分量的位移电流电导：



10.1 介绍

二次谐波生成 (Second Harmonic Generation, SHG) 是一种典型的二阶非线性光学效应, 指材料在光照下吸收两个频率为 ω 的光子并辐射出一个频率为 2ω 的光子。该过程体现了材料对强光场的非线性响应特性, 广泛应用于频率变换、激光器、非线性显微成像以及对材料对称性的探测等领域。

SHG 只能发生在缺乏中心反演对称性的材料中。由于反演对称会强制二阶极化张量 $\chi^{(2)}$ 为零, 因此 SHG 也常被用作判断材料是否破缺反演对称性的实验手段, 特别是在铁电体、拓扑材料、以及层状范德瓦耳斯异质结构中具有重要应用价值。

SHG 的极化响应可表示为:

$$P^a(2\omega) = \chi^{abc}(-2\omega; \omega, \omega) E^b(\omega) E^c(\omega),$$

其中:

- $P^a(2\omega)$ 表示沿 a 方向的非线性极化强度;
- $E^b(\omega)$ 和 $E^c(\omega)$ 是频率为 ω 的电场;
- $\chi^{abc}(-2\omega; \omega, \omega)$ 是二阶非线性极化张量, 描述频率倍频过程的强度与方向性。

在 PYATB 中, SHG 的计算支持两种不同规范的实现方式:

1. 长度规范 (length gauge)

在长度规范下, SHG 电导率可拆分为带间项和带内项两部分, 分别对应三重能带循环与二重能带循环。其表达式如下:

带间项:

$$\chi_{interband}^{abc}(-2\omega, \omega, \omega) = \frac{e^3}{\hbar^2 \Omega} \sum_{nml, \mathbf{k}} \frac{r_{nm}^a \{r_{ml}^b r_{ln}^c\}}{(\omega_{ln} - \omega_{ml})} \left[\frac{2f_{nm}}{\omega_{mn} - 2\omega} + \frac{f_{ln}}{\omega_{ln} - \omega} + \frac{f_{ml}}{\omega_{ml} - \omega} \right]$$

带内项:

$$\begin{aligned} \chi_{inband}^{abc}(-2\omega, \omega, \omega) = & \frac{i}{2\hbar^2\Omega} \sum_{nm, \mathbf{k}} f_{nm} \left[\frac{2}{\omega_{mn}(\omega_{mn} - 2\omega)} r_{nm}^a (r_{nm;c}^b + r_{mn;b}^c) + \frac{1}{\omega_{mn}(\omega_{mn} - \omega)} (r_{nm;c}^a r_{mn}^b + r_{nm;b}^a r_{mn}^c) \right. \\ & + \frac{1}{\omega_{mn}^2} \left(\frac{1}{\omega_{mn} - \omega} - \frac{4}{\omega_{mn} - 2\omega} \right) r_{nm}^a (r_{mn}^b \Delta_{mn}^c + r_{mn}^c \Delta_{mn}^b) \\ & \left. - \frac{1}{2\omega_{mn}(\omega_{mn} - \omega)} (r_{nm;a}^b r_{mn}^c + r_{nm;a}^c r_{mn}^b) \right] \end{aligned}$$

其中 r_{nm}^a 是带间偶极跃迁元, $r_{nm;b}^c$ 为其协变导数, Δ_{mn}^a 为能带间速度差。

2. 速度规范 (velocity gauge)

在速度规范下, SHG 的表达更为简洁, 写作:

$$\chi^{abc}(-2\omega; \omega, \omega) = - \sum_{nml, \mathbf{k}} \frac{i}{2\omega^3(2\omega - \omega_{mn})} \left(\frac{f_{nl}}{\omega - \omega_{ln}} + \frac{f_{ml}}{\omega - \omega_{ml}} \right) v_{nm}^c v_{ml}^a v_{ln}^b$$

其中 v_{nm}^a 是带间速度矩阵元。

两种方法在物理上是等价的, 但在数值实现上可能有不同的稳定性与适用范围。长度规范方法能够明确区分不同物理过程贡献, 且耗时比速度规范要少; 速度规范形式更紧凑, 便于快速评估频率响应。

参考文献:

- Hughes, J. L. P. & Sipe, J. E. Calculation of second-order optical response in semiconductors. Phys. Rev. B 53, 10751–10763 (1996).

下面以 GaAs 为示例, 演示如何使用 ABACUS 与 PYATB 联合进行 SHG 计算。相关文件位于 tutorial/GaAs_SHG 目录下, 其中包含两个子文件夹: abacus 和 pyatb, 分别提供了第一性原理计算与后续紧束缚模型计算所需的输入文件。

10.2 计算步骤

快速测试步骤:

```
$ cd tutorial/GaAs_SHG/abacus
$ ./run_abacus.sh
$ cd ../pyatb
$ ./run_pyatb.sh
```

10.2.1 第一步, ABACUS 自洽计算产生紧束缚模型文件

在 tutorial/GaAs_SHG/abacus 文件夹中, 我们已经准备好了 ABACUS 自洽计算的输入文件, 包括:

- 控制文件: INPUT
- 结构文件: STRU
- 赝势文件: *.upf
- 原子轨道基组文件: *.orb
- K 点采样文件: KPT

其中, INPUT 文件的核心参数如下, 已对关键设置作出注释, 其他细节可参考 ABACUS 官方文档: https://abacus.deepmodeling.com/en/latest/advanced/input_files/input-main.html

```

INPUT_PARAMETERS

# System variables
suffix          GaAs
calculation     scf          # 指定计算类型为自洽计算
esolver_type    ksdfc
symmetry        -1          # 完全关闭对称性分析，如果发现不收敛，可以关闭对称性
init_chg        atomic

# Input Files
pseudo_dir      ./
orbital_dir     ./

# Plane Wave
ecutwfc         100          # 平面波截断值，可以选择轨道文件中所使用的截断能量

# Electronic structure
basis_type      lcao          # 使用原子轨道作为基组
ks_solver        genelpa
dft_functional   hse          # 使用杂化泛函修正带隙
nspin            1
smearing_method  gauss        # 电子占据模糊方法
smearing_sigma   0.02
mixing_type      pulay        # 设置电荷密度混合方法
mixing_beta      0.7
scf_nmax         200          # 最大电子收敛步数
scf_thr          1e-8         # 电荷密度收敛精度
lspinorb         0           # 关闭自旋轨道耦合
noncolin         0           # 关闭自旋非共线

# Output Variables
out_chg          1           # 输出自洽后的电荷密度
out_mat_hs2       1           # 输出HR、SR矩阵
out_mat_r         1           # 输出rR矩阵

# Exact Exchange
# HSE 相关参数
exx_separate_loop 0
exx_pca_threshold 1e-4
exx_c_threshold    1e-4
exx_dm_threshold   1e-4
exx_ccp_rmesh_times 1.5

```

执行 ABACUS 自洽计算后，若设置了 `out_mat_hs2 = 1` 和 `out_mat_r = 1`，将在 `OUT.*` 目录中生成所需紧束缚模型文件：

- **data-HR-sparse_SPIN0.csr**：实空间哈密顿量
- **data-SR-sparse_SPIN0.csr**：重叠矩阵
- **data-rR-sparse.csr**：位置偶极矩阵

注 1：若设置 `nspin = 2`（即自旋极化计算），则还会生成 **data-HR-sparse_SPIN1.csr** 文件，用于描述第二个自旋通道。注 2：ABACUS 生成紧束缚模型文件，必须在 `lcao` 基组下，即 `basis_type lcao`。注 3：随着 ABACUS 版本的更新，部分紧束缚模型文件的文件名称可能会发生变化。

在准备好了 ABACUS 各种输入文件之后，我们可以运行目录下的 `run_abacus.sh` 脚本来进行 ABACUS 的自洽计算。

```
#!/bin/bash

# 加载ABACUS环境
source ~/softwares/abacus-develop-LTSv3.10.0/toolchain/abacus_env.sh

# 运行abacus
export OMP_NUM_THREADS=1 # 设置OpenMP线程数为1
num_mpi=8 # 设置MPI进程数
mpirun -n $num_mpi abacus > job.log 2> job.err
```

请根据具体的计算节点配置适当调整 OMP_NUM_THREADS 与 num_mpi，以充分利用本地硬件资源，获得更高的并行效率。

10.2.2 第二步，使用紧束缚模型文件进行 PYATB 计算

在第一步中，ABACUS 完成自洽计算后，会产生 HR、SR 文件，这些文件是 PYATB 必要的输入文件。然后我们进入 tutorial/GaAs_SHG/pyatb 文件夹，进行 SHG 的计算。接下来我们介绍 PYATB 的输入文件 **Input** 的内容：

```
INPUT_PARAMETERS
{
    nspin                1                # 与 ABACUS 的 nspin 保持一致
    package              ABACUS           # 指定数据来源为 ABACUS
    fermi_energy          10.171348972
    fermi_energy_unit     eV
    HR_route              ../abacus/OUT.GaAs/data-HR-sparse_SPIN0.csr # HR 矩阵
    SR_route              ../abacus/OUT.GaAs/data-SR-sparse_SPIN0.csr # SR 矩阵
    rR_route              ../abacus/OUT.GaAs/data-rR-sparse.csr      # rR 矩阵
    HR_unit               Ry               # HR 单位为 Rydberg
    rR_unit               Bohr             # rR 单位为 Bohr
    max_kpoint_num        100000          # 单轮计算的最大k点数目
}

LATTICE
{
    lattice_constant      1.889727
    lattice_constant_unit Bohr
    lattice_vector
    0.0000000000000000    2.7650000000000001    2.7650000000000001
    2.7650000000000001    0.0000000000000000    2.7650000000000001
    2.7650000000000001    2.7650000000000001    0.0000000000000000
}

BAND_STRUCTURE
{
    kpoint_mode           line            # 设置 k_
    ↪点采样模式为高对称路径，有line, mp, direct三种
    kpoint_num            5               # 设置路径中的高对称点数
    high_symmetry_kpoint   # 设置高对称点坐标及插值数
    0.00000 0.00000 0.0000 200 # G
    0.50000 0.00000 0.0000 200 # X
    0.50000 0.50000 0.0000 200 # T
    0.00000 0.50000 0.0000 200 # Y
    0.00000 0.00000 0.0000 1  # G
    kpoint_label          G,X,T,Y,G      # 设置高对称点名称
}
```

(续下页)

(接上页)

```

SHG
{
    omega          0.01  4          # 光子能量范围 (eV)
    domega         0.01           # 光子能量间隔 (eV)
    eta            0.05           # 能量展宽 (eV)
    grid           50 50 50       # 划分布里渊区网格数
}

```

关于功能模块 SHG 完整设置参数为：

- **method**, int 类型, 0 表示使用长度规范, 1 表示使用速度规范, 默认值为 0。
- **omega**, float 类型, 有两个值, 设置光子能量范围。
- **domega**, float 类型, 光子能量间隔。
- **eta**, float 类型, $\hbar\omega \rightarrow \hbar\omega + i\eta$, 用于避免分母为零引起的数值发散问题。
- **grid**, int 类型, 三个值。划分布里渊区的均匀网格数。

在准备好了 PYATB 各种输入文件之后, 我们可以运行目录下的 **run_pyatb.sh** 脚本来进行 PYATB 的计算。

```

#!/bin/bash

# 加载conda环境
source ~/software/miniconda3/bin/activate
conda activate pyatb

# 运行pyatb
export OMP_NUM_THREADS=1 # 设置OpenMP线程数为1
num_mpi=8 # 设置MPI进程数
mpirun -n $num_mpi pyatb > job.log 2> job.err

```

10.3 结果展示

运行 PYATB 后, 所有计算结果将保存在 Out/ 文件夹中。每个功能模块对应一个子文件夹, 便于结构化管理与后续分析

10.3.1 SHG

Out/Second_Harmonic_Generation 文件夹包含 SHG 相关的输出, 相应文件为:

- **shg_real.dat**: 极化张量实部的 27 个分量数据。第一列为光子能量, 后续列为 χ^{abc} 的所有 27 个分量 (按 abc 排列顺序展开)。
- **shg_imag.dat**: 极化张量虚部的 27 个分量数据, 格式与 **shg_real.dat** 相同。
- **plot_shg.py**: 用于绘制特定方向的 SHG 响应的绘图脚本。

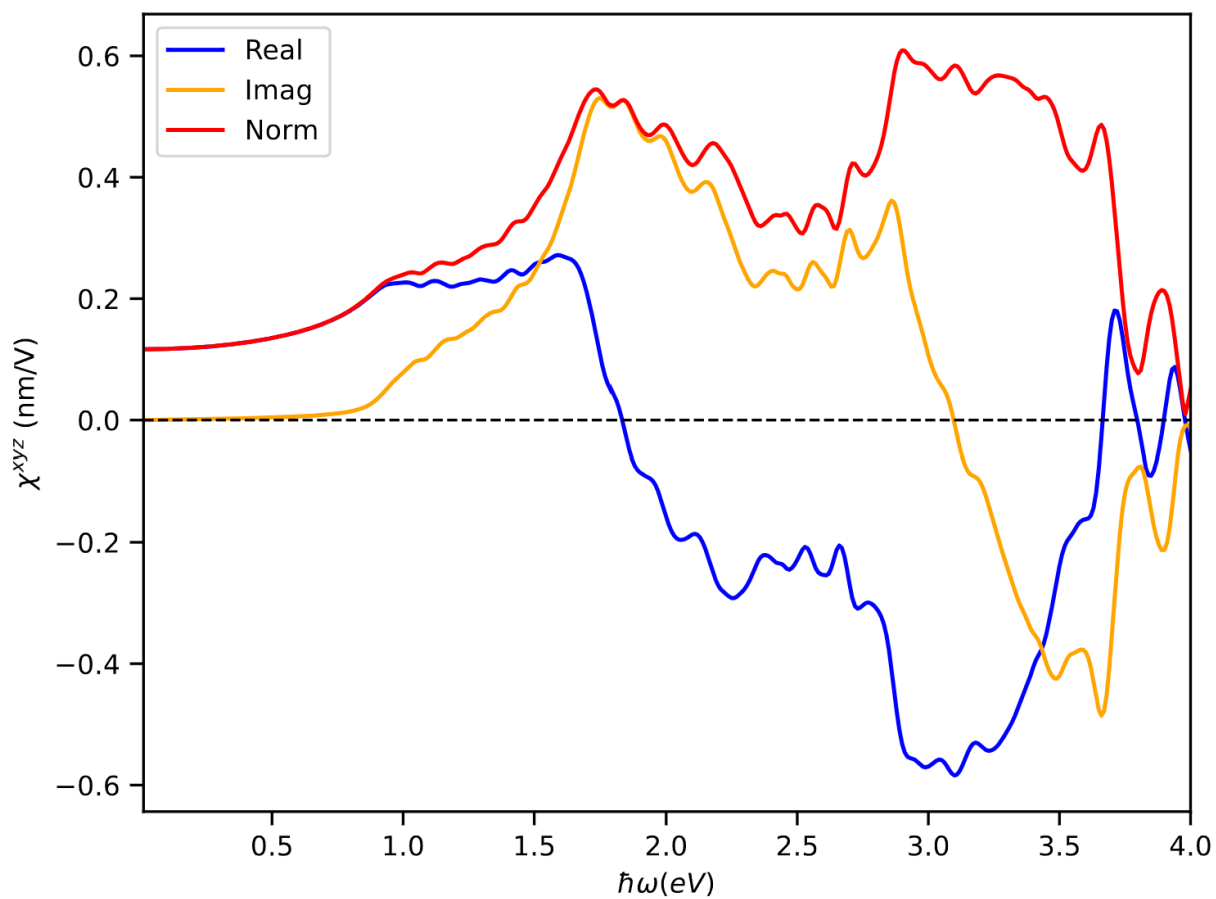
可通过修改绘图脚本 **plot_shg.py** 中的方向设置来选择绘制的极化张量分量:

```

a = 'x'
b = 'y'
c = 'z'

```


例如，上述设置将绘制 χ^{xyz} 分量随频率变化的实部与虚部。运行后绘图结果如下所示：



11.1 介绍

Berry curvature 是现代凝聚态物理中描述电子波函数几何性质的核心量之一，是多种非平庸输运和光学效应（如反常霍尔效应、自旋霍尔效应、非线性响应等）的物理源头。

对于能带 n 和动量 \mathbf{k} ，Berry curvature 定义为：

$$\Omega_n^{ab}(\mathbf{k}) = -2 \text{Im} \sum_{m \neq n} \frac{\langle u_{n\mathbf{k}} | \partial_{k_a} H | u_{m\mathbf{k}} \rangle \langle u_{m\mathbf{k}} | \partial_{k_b} H | u_{n\mathbf{k}} \rangle}{(E_{n\mathbf{k}} - E_{m\mathbf{k}})^2},$$

其中 H 是哈密顿量， $u_{n\mathbf{k}}$ 是布洛赫波函数， $E_{n\mathbf{k}}$ 是能带能量。

在具有时间反演对称性的系统中，Berry curvature 满足奇函数关系：

$$\Omega_a(\mathbf{k}) = -\Omega_a(-\mathbf{k}),$$

这意味着在整个布里渊区内对 Berry curvature 的积分严格为零，因此不会出现反常霍尔电导率（AHC）的线性响应项。

然而，如果系统破缺中心反演对称性，尽管 Berry curvature 本身在 BZ 中的积分仍为零，其在动量空间中的分布梯度可能不为零，从而诱导出高阶非线性输运效应。具体而言：

- 当系统处于交流光场激发下，响应电流可以写为

$$j_a^0 = \chi_{abc} E_b(\omega) E_c(-\omega), \quad j_a^{2\omega} = \chi_{abc} E_b(\omega) E_c(\omega),$$

分别对应整流电流（直流响应）与二次谐波电流（ 2ω 响应），其中 ω 为外加光场的频率。

- 非线性响应张量 χ_{abc} 的表达式为：

$$\chi_{abc} = -\varepsilon_{adc} \cdot \frac{e^3 \tau}{2(1 + i\omega\tau)} D_{bd},$$

其中 ε_{adc} 是 Levi-Civita 反对称张量， τ 是电子弛豫时间， D_{bd} 是 Berry curvature dipole，定义如下：

$$D_{ab}(T) = \int [dk] \sum_n \frac{\partial E_n}{\partial k_a} \Omega_{n,b} \left(-\frac{\partial f_0}{\partial E} \right)_{E=E_n}.$$

其中 T 是温度, f_0 是费米-狄拉克分布函数。该张量描述了 Berry curvature 在动量空间中的不均匀性, 代表了非线性响应的几何起源。

在 PYATB 中, 可以在给定温度 T 和化学势 μ 下, 通过对 Berry curvature 偶极在能量空间的加权积分, 得到对应的 BCD 值, 其表达式为:

$$D_{ab}(\mu, T) = - \int dE \frac{\partial f_0(E, \mu, T)}{\partial E} \cdot D_{ab}(E),$$

其中 $D_{ab}(E)$ 是能量分辨的 Berry curvature 偶极, $\frac{\partial f_0}{\partial E}$ 用于考虑温度效应下的费米面加权。

Berry curvature 偶极是非线性霍尔效应 (Nonlinear Hall Effect) 研究的核心几何量, 在二维材料、Weyl 半金属、非中心对称拓扑系统中具有广泛的物理意义和潜在应用价值。

参考文献:

- Zeng, C., Nandy, S. & Tewari, S. Nonlinear transport in Weyl semimetals induced by Berry curvature dipole. Phys. Rev. B 103, 245119 (2021).

下面以 Weyl 半金属模型为例, 演示如何使用 PYATB 进行 Berry curvature 和 Berry curvature dipole 的计算。相关文件位于 `tutorial/weyl_model_BCD` 目录下。

11.2 计算步骤

快速测试步骤:

```
$ cd tutorial/weyl_model_BCD
$ python generate_weyl_model.py
$ ./run_pyatb.sh
$ cd ./Out/BERRY_CURVATURE_DIPOLE
$ python plot_bcd.py
```

11.2.1 第一步, 生成 PYATB 需要的格点紧束缚模型

我们在 `tutorial/weyl_model_BCD` 目录下提供了一个脚本 `generate_weyl_model.py`, 用于从 Weyl 半金属的动量空间哈密顿量 $H(\mathbf{k})$ 生成格点空间哈密顿量 $H(\mathbf{R})$ 。该模型的形式如下:

$$\begin{aligned} H_1(\mathbf{k}) &= \mathbf{N}(\mathbf{k})\boldsymbol{\sigma} + N_0(\mathbf{k})\sigma_0 \\ N_0(\mathbf{k}) &= \gamma (\cos(2k_x) - \cos k_0) (\cos k_z - \cos k_0) \\ N_x(\mathbf{k}) &= [m (1 - \cos^2 k_z - \cos k_y) + 2t_x (\cos k_x - \cos k_0)] \\ N_y(\mathbf{k}) &= -2t \sin k_y, \quad N_z(\mathbf{k}) = -2t \cos k_z \end{aligned}$$

该模型具有时间反演对称性但破缺空间反演对称性, 属于典型的 Weyl 半金属系统。模型中存在四个 Weyl 点, 分布在 $(\pm k_0, 0, \pm 0.25)$ (单位为 2π), 并具有 M_x 和 M_y 镜面对称性。采用的参数设置如下:

```
t = 1
tx = 0.5*t
m = 2*t
k0 = np.pi/2
gamma = t*2.5
```

运行 `generate_weyl_model.py` 后, 将生成三个束缚模型文件 `HR`、`SR`、`rR`:

- `data-HR-sparse_SPIN0.csr`: `HR`
- `data-SR-sparse_SPIN0.csr`: `SR`

- data-rR-sparse_SPIN0.csr: rR

11.2.2 第二步，使用紧束缚模型文件进行 PYATB 计算

继续在 tutorial/weyl_model_BCD 目录下，我们将基于上述生成的模型文件，使用 PYATB 计算能带结构、Berry curvature、Berry curvature dipole 等物理量。

我们介绍 PYATB 的输入文件 **Input** 的内容：

```
INPUT_PARAMETERS
{
    nspin                4                # HR、SR文件为复数。nspin=1 时读取实数的HR和SR。
    package              ABACUS
    fermi_energy         0.0
    fermi_energy_unit    eV
    HR_route             data-HR-sparse_SPIN0.csr # HR 矩阵
    SR_route             data-SR-sparse_SPIN0.csr # SR 矩阵
    rR_route             data-rR-sparse_SPIN0.csr # rR 矩阵
    HR_unit              eV                # HR 单位为 eV
    rR_unit              Angstrom          # rR 单位为 Å
    ↪ Bohr, 这个模型中 rR 为零矩阵
    max_kpoint_num       1000000          # 单轮计算的最大 k 点数目
}

LATTICE
{
    lattice_constant     1
    lattice_constant_unit Angstrom
    lattice_vector
    1 0 0
    0 1 0
    0 0 1
}

BAND_STRUCTURE
{
    kpoint_mode          line              # 设置 k_
    ↪ 点采样模式为高对称路径，有 line, mp, direct 三种
    kpoint_num           6                # 设置路径中的高对称点数
    high_symmetry_kpoint # 设置高对称点坐标及插值数
    0.50000 0.00000 0.5000 200 # A1
    0.00000 0.00000 0.0000 200 # G
    -0.50000 0.00000 -0.5000 200 # A2
    -0.50000 0.00000 0.5000 200 # A3
    0.00000 0.00000 0.0000 200 # G
    0.50000 0.00000 -0.5000 1 # A4
    kpoint_label         A1,G,A2,A3,G,A4 # 设置高对称点名称
}

FIND_NODES
{
    energy_range         -0.1 0.1        # 设置寻找简并点的能量范围
    initial_grid          100 100 100    # 初筛时，划分布里渊区的均匀网格
    initial_threshold     0.01           # 判断初筛时能量简并的阈值
    adaptive_grid         10 10 10       # 第二次筛选，对初筛的网格加密。
    adaptive_threshold    0.001          # 第二次能量简并的阈值
}
```

(续下页)

(接上页)

```

BERRY_CURVATURE
{
    kpoint_mode      line
    kpoint_num       2
    high_symmetry_kpoint
-0.50000 0.00000 0.2500 200 # -X
0.50000 0.00000 0.2500 1   #  X
    kpoint_label     -X, X
}

CHIRALITY
{
    k_vect            0.25000000 0.00000000 0.25000000 # Weyl点坐标
    radius            0.9                               # 以Weyl点为中心的球面半径
    point_num         10000                             # 球面撒点
    method            1                                 # 使用Kubo公式计算Berry curvature
}

BERRY_CURVATURE_DIPOLE
{
    omega            -3 3                               # 设置费米能级范围 (eV) , 不是以 fermi_energy_
    ↪为中心的
    domega           0.001                             # 能量间隔 (eV)
    grid             200 200 200                       # 划分布里渊区的均匀网格
}

```

关于功能模块 BAND_STRUCTURE 完整设置参数为：

- **wf_collect**, bool 类型, 为 1 时输出波函数的展开系数, 默认为 0。
- **band_range**, int 类型, 有两个值, 设置计算的能带范围, 从 1 开始计数, 例如 1 20, 计算第 1 到第 20 共 21 条能带。不设置该参数, 会计算所有能带。
- **kpoint_mode**, str 类型, 指定 k 点采样模式, 存在此参数时会有额外的设置参数。

关于功能模块 FIND_NODES 完整设置参数为：

- **energy_range**, float 类型, 有两个值, 寻找简并点的能量范围, 不是以 `fermi_energy` 为中心的。
- **initial_grid**, int 类型, 有三个值。初筛时, 划分布里渊区的均匀网格, 以粗网格确定可能存在简并的 k 点和能带的范围。
- **initial_threshold**, float 类型, 判断初筛时能量简并的阈值。
- **adaptive_grid**, int 类型, 有三个值。第二次筛选, 对初筛可能有简并的 k 点再进行网格加密。
- **adaptive_threshold**, float 类型, 第二次筛选能量简并的阈值。
- **k_start**, float 类型, 有三个值, 确定指定布里渊区范围的原点, 默认为 0.0 0.0 0.0。
- **k_vect1**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 1, 默认为 1.0 0.0 0.0。
- **k_vect2**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 2, 默认为 0.0 1.0 0.0。
- **k_vect3**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 3, 默认为 0.0 0.0 1.0。

关于功能模块 BERRY_CURVATURE 完整设置参数为：

- **method**, int 类型, 计算 Berry curvature 的方法。为 0 时, 直接偏导数求解。为 1 时, 使用 Kubo 公式求解。默认为 0。
- **occ_band**, int 类型, 设置占据能带数目。不设此值时, 以 **fermi_energy** 来判断占据情况。
- **kpoint_mode**, str 类型, 指定 k 点采样模式, 存在此参数时会有额外的设置参数。

关于功能模块 CHIRALITY 完整设置参数为:

- **method**, int 类型, 计算 Berry curvature 的方法。为 0 时, 直接偏导数求解。为 1 时, 使用 Kubo 公式求解。默认为 0。
- **k_vect**, float 类型, 指定某个具体 Weyl 点的分数坐标。
- **radius**, float 类型, 以 Weyl 点为中心创建一个球面, 此参数为球面半径。对应 k 点的 Cartesian 坐标。
- **point_num**, int 类型, 指定球面上撒点的数目。

关于功能模块 BERRY_CURVATURE_DIPOLE 完整设置参数为:

- **omega**, float 类型, 有两个值。设置费米能级范围 (eV), 不是以 **fermi_energy** 为中心的。
- **domega**, float 类型, 能量间隔 (eV)。
- **grid**, int 类型, 有三个值, 划分布里渊区的均匀网格。

如果某个功能模块中存在 **kpoint_mode** 参数时, 会添加额外的设置参数:

- 当 **kpoint_mode** 为 **mp** 时, 新增参数:
 - **k_start**, float 类型, 有三个值, 确定指定布里渊区范围的原点, 默认为 0.0 0.0 0.0。
 - **k_vect1**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 1, 默认为 1.0 0.0 0.0。
 - **k_vect2**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 2, 默认为 0.0 1.0 0.0。
 - **k_vect3**, float 类型, 有三个值, 确定指定布里渊区范围的展开矢量 3, 默认为 0.0 0.0 1.0。
 - **mp_grid**, int 类型, 有三个值, 均匀划分指定布里渊区范围的网格数目, 没有默认值。
- 当 **kpoint_mode** 为 **line** 时, 新增参数:
 - **kpoint_num**, int 类型, 指定高对称点的数目。
 - **high_symmetry_kpoint**, float 类型, 每个高对称点有四个值, 设置高对称点坐标及插值数。
 - **kpoint_label**, str 类型, 数目与 **kpoint_num** 保持一致, 指定每一个高对称点的 label。
- 当 **kpoint_mode** 为 **direct** 时, 新增参数:
 - **kpoint_num**, int 类型, 指定 k 点的数目。
 - **kpoint_direct_coor**, float 类型, 每一个 k 点三个值, 指定 k 点的坐标。

在准备好了 PYATB 各种输入文件之后, 我们可以运行目录下的 **run_pyatb.sh** 脚本来进行 PYATB 的计算。

```
#!/bin/bash

# 加载conda环境
source ~/software/miniconda3/bin/activate
conda activate pyatb

# 运行pyatb
export OMP_NUM_THREADS=1 # 设置OpenMP线程数为1
num_mpi=8 # 设置MPI进程数
mpirun -n $num_mpi pyatb > job.log 2> job.err
```

11.3 结果展示

运行 PYATB 后，所有计算结果将保存在 Out/ 文件夹中。每个功能模块对应一个子文件夹，便于结构化管理与后续分析。

11.3.1 BAND_STRUCTURE

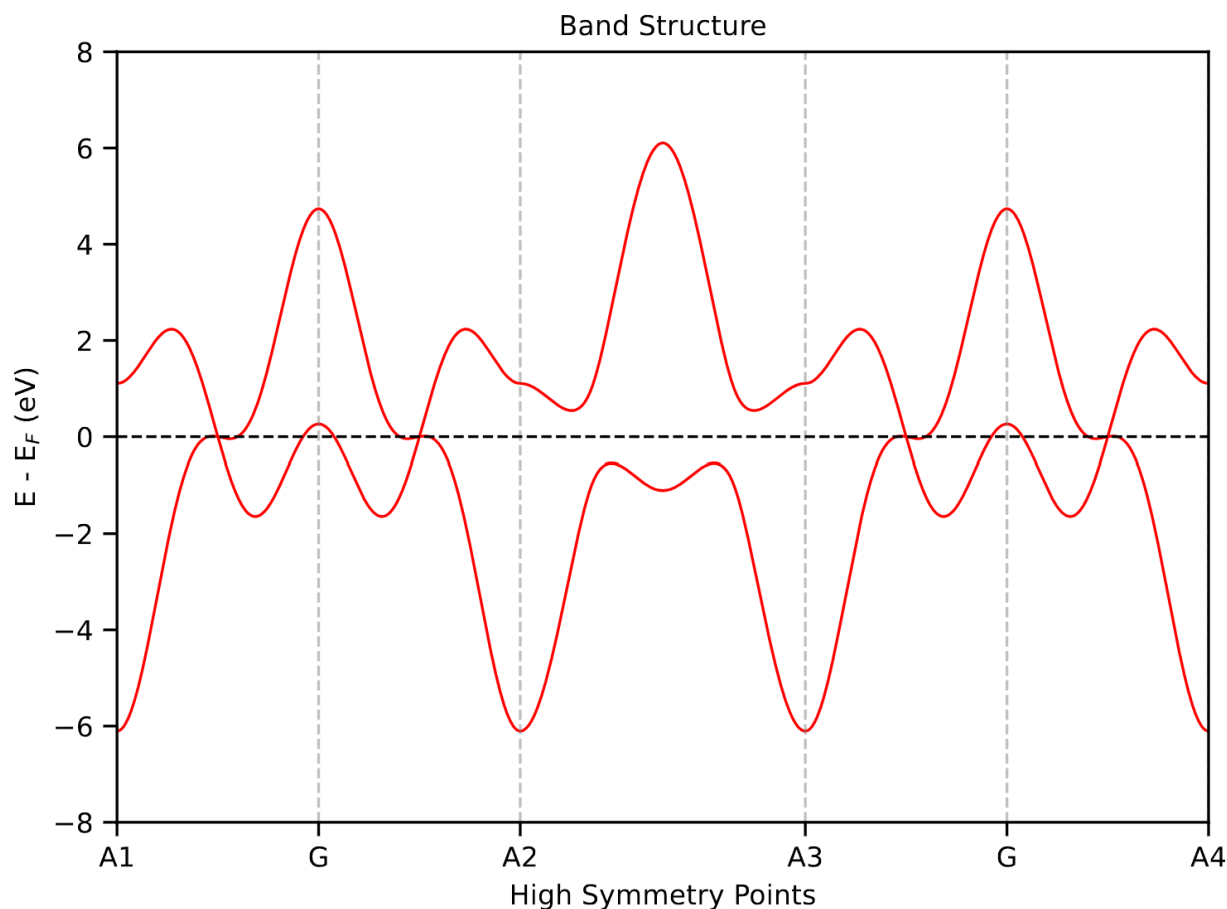
Out/Band_Structure 文件夹包含能带结构相关的输出，相应文件为：

- **band_info.dat**：提供了基本的能带信息，包括费米能、带隙、价带顶（VBM）和导带底（CBM）的位置。其中能带指标是从 0 开始计数的。
- **band.dat**：包含所有 k 点的能带本征值，格式为二维矩阵：每行对应一个 k 点，每列表示一条能带，能量值未减去费米能。
- **kpt.dat**：记录所有计算使用的 k 点坐标。
- **high_symmetry_kpoint.dat**：记录高对称点信息，用于绘图。
- **x_coor_array.dat**：能带绘图时使用的 X 轴坐标。
- **plot_band.py**：能带绘图脚本

在 **plot_band.py** 中，可以修改 `y_min` 和 `y_max` 变量来控制能带图的纵轴范围。该脚本默认绘制的是平移费米能后的能带图。我们修改纵轴范围：

```
y_min = -8 # eV
y_max = 8 # eV
```

执行绘图脚本后，生成的能带图为：



从能带图中，我们可以看到在费米能处存在 4 个简并点，对应着 Weyl 点。

11.3.2 FIND_NODES

Out/Find_Nodes 文件夹包含简并点相关的输出，相应文件为：

- **nodes.dat**: 提供了简并点信息。每一行为一个简并点。前三个数值为 **k** 点的分数坐标，之后的数值表示简并的能带指标（从 1 计算）。
- **plot_nodes.py**: 在三维空间中绘制简并点的脚本。

我们执行 FIND_NODES 功能可以在指定布里渊区、只能能量范围内寻找简并点。我们设置在费米能附近来寻找 Weyl 点。计算符合模型解析结果，对应文件是 **nodes.dat**：

```
0.25000000 0.00000000 0.25000000 1.00000000 2.00000000
0.25000000 0.00000000 0.75000000 1.00000000 2.00000000
0.75000000 0.00000000 0.25000000 1.00000000 2.00000000
0.75000000 0.00000000 0.75000000 1.00000000 2.00000000
```

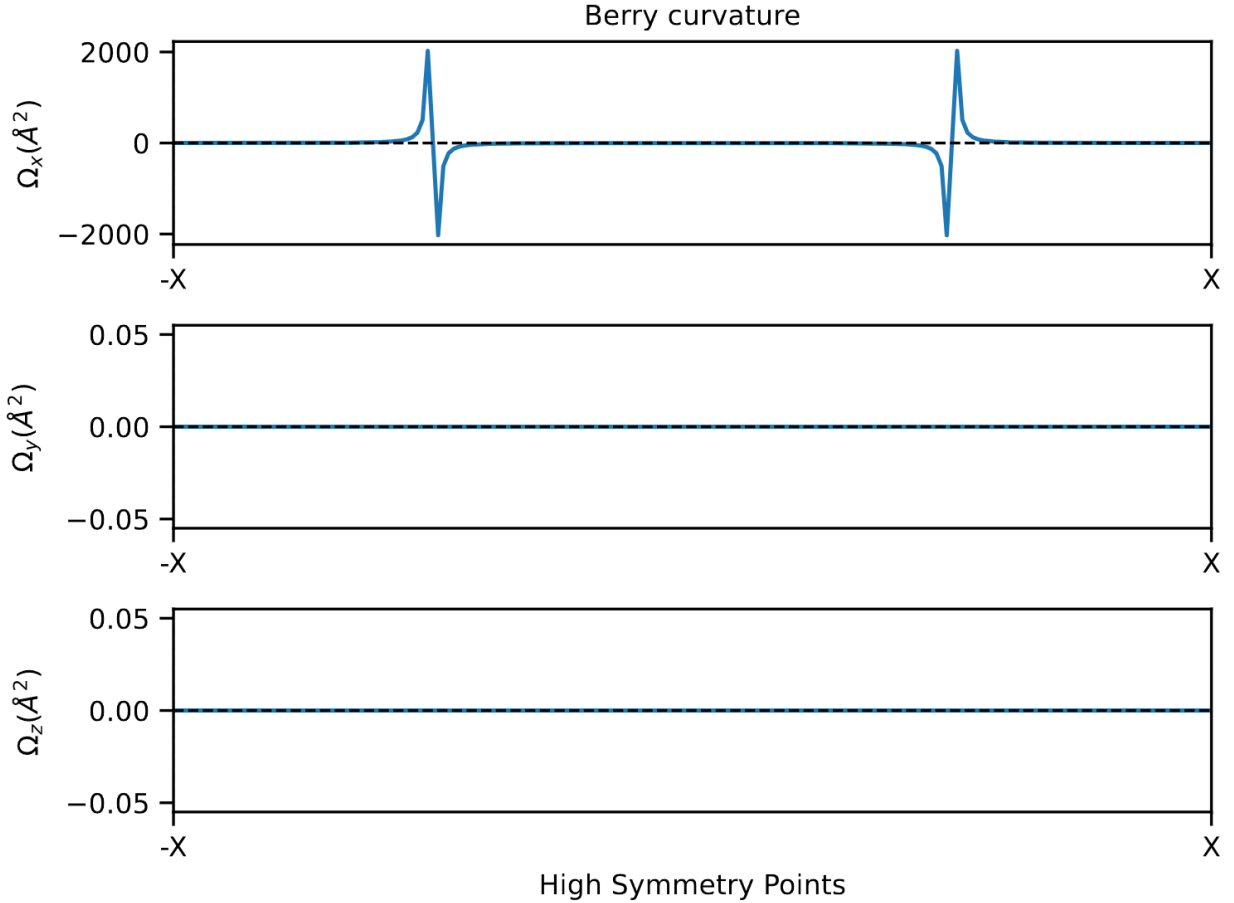
然后我们可以进一步使用 CHIRALITY 功能模块来检查这些 Weyl 点的手性。例如我们计算了 (0.25, 0.00, 0.25) 的手性，结果在 Out/Chirality/chirality.dat 文件中，结果显示为 +1。

11.3.3 BERRY_CURVATURE

Out/Berry_Curvature 文件夹包含 Berry curvature 相关的输出，相应文件为：

- **berry_curvature.dat**: 记录 Berry curvature 数据，每一行为某一 k 点的 x, y, z 三个方向的数值。
- **kpt.dat**: 记录所有计算使用的 k 点坐标。
- **plot_berry_curvature_line.py**: 绘图脚本。

我们可以得到 Berry curvature 沿着高对称点的结果：



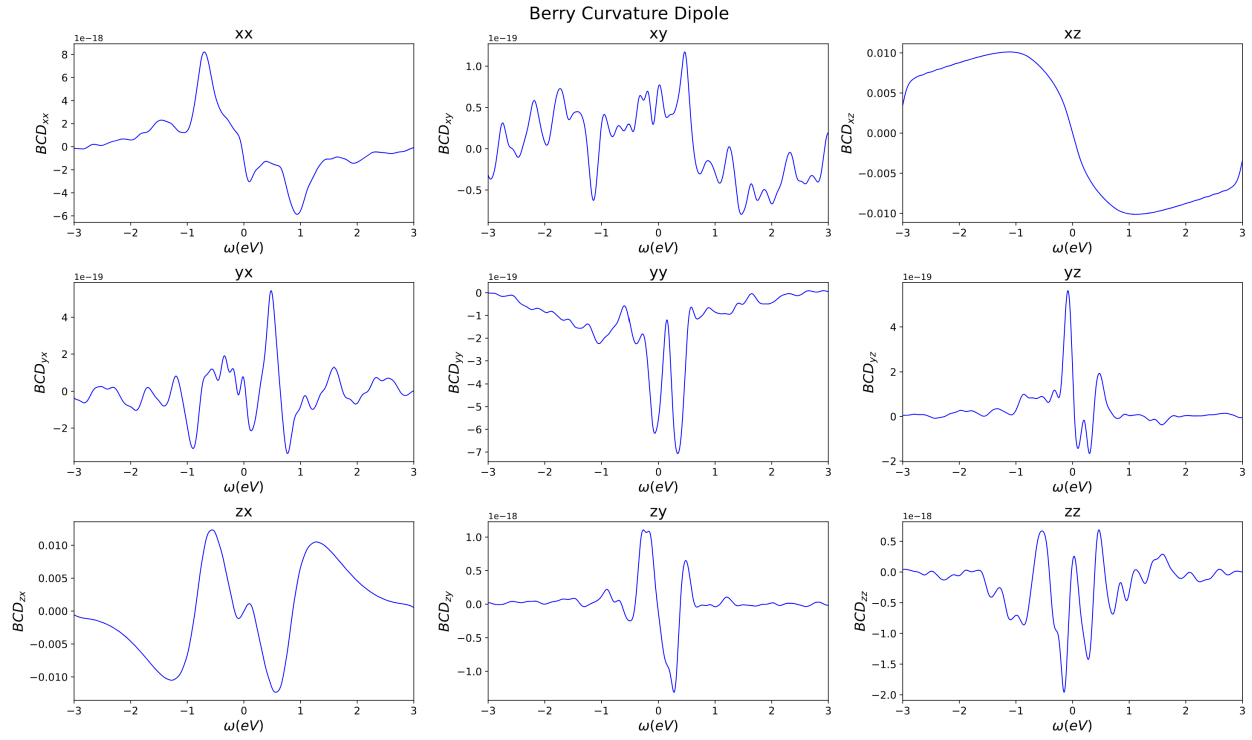
一对手性相反的 Weyl 点附近会出现巨大的 Berry curvature，同时 Weyl 点处的 Berry curvature 是发散的。

11.3.4 BERRY_CURVATURE_DIPOLE

Out/BERRY_CURVATURE_DIPOLE 文件夹包含 Berry curvature dipole 相关的输出，相应文件为：

- **bcd.dat**, 记录不同能量点，不同方向上的能量分辨的 Berry curvature dipole 数值。一共有 9 行，对于 ab 方向。每一列对应能量点。
- **plot_bcd.py**, 绘制不同温度下、不同方向上的 Berry curvature dipole。对能量和费米分布函数导数进行积分得到最终的 Berry curvature dipole。

我们执行 **plot_bcd.py** 脚本，可以得到以下结果：



注 1: 与文献结果出现符号相反是因为 BCD 定义上存在符号差异。另外, 猜测文献图中的 zx 和 xz 可能画反了。