

RESCU应用教程

RESCU2021B



前言

鸿之微科技(上海)股份有限公司(简称:鸿之微,英文: hzwtech, 股票代码: 300110)更是一家致力于开发材料设计软件、集成电路工艺参数提取软件及集成电路器件设计软件的科技公司。鸿之微科技由多名海内外知名专家和学者联合发起,以打造自主知识产权的软件为愿景,以成为材料设计、模拟计算软件领域全球引领者为使命的高科技公司。经过三年的高速发展,公司拥有多个完全自主知识产权的材料及器件设计软件,其中包括 1 个全球专利, 10 项软件著作权, 31 个国内专利。产品包括纳米器件和量子输运计算软件 Nanodcal、紧束缚程序 Nanoskif 和 Nanoskim、大体系 DFT 计算软件 RESCU、OLED 设计平台 MOMAP 等。

本书针对广大用户对 RESCU 软件的需求,鸿之微公司特推出 RESCU 应用教程供用户学习和交流。

手册共四章,其编写遵循由易到难的原则,较为全面地介绍了 RESCU 软件的使用方法,详细地讨论和分析了 RESCU 的计算结果,列举了 RESCU 安装和使用过程中一些常见问题。主要内容如下:

第一章为基本应用,介绍了 RESCU 基本计算功能案例。

第二章,以已发表的科研论文为题材,目前已有一个案例——双层扭转黑磷摩尔条纹,详细地介绍了 RESCU 从模型搭建到结果分析的使用过程。帮助用户系统的了解整套计算过程。

第三章为常见问题,其中收集了自 RESCU 发布以来,安装和使用过程中遇到的一些经典问题,并在不断地更新中。用户可以先浏览一遍对其有个大概的了解,以便在遇到类似问题时,可以通过检索关键词的方法找到解决途径,快速解决自己的问题。

以上涉及到的算例结构,均可以通过公司的另外一款图形界面软件 Device Studio 直接进行建模,生成对应的输入文件。

此外,在学习本手册的同时,用户可以不时地翻阅 RESCU 的英文参数说明,以便学习更多的计算技能。

为了大家快速和便捷的找到鸿之微的对应服务,大家可以按照下面的方式和鸿之微联系:



目录

前 言.....	2
目录.....	4
第一章 基本使用.....	6
1.1 自洽.....	6
1.1.1 REAL 的自洽计算.....	6
1.1.2 LCAO 的自洽计算.....	10
1.2 能带.....	13
1.2.1 Si 体系(无自旋体系 degenerate).....	13
1.2.2 Ni 体系(线性自旋体系 collinear).....	16
1.2.3 MoS ₂ 体系(一般自旋+自旋轨道耦合体系 non-collinear+SOC).....	17
1.2.4 Si 体系(能带投影计算).....	20
1.3 态密度.....	25
1.3.1 Si 总态密度计算.....	25
1.3.2 graphene 投影态密度计算.....	27
1.3.3 graphene 局域态密度计算.....	30
1.4 Mulliken 电荷及实空间电荷处理.....	34
1.4.1 Mulliken 电荷及数据处理.....	34
1.4.2 实空间电荷密度数据处理.....	36
1.5 HSE06.....	38
1.5.1 HSE06 自洽计算.....	38
1.5.2 HSE06 能带计算.....	40
1.5.3 HSE06 态密度计算.....	41
1.6 DFT+U 计算.....	42
1.6.1 DFT+U 自洽计算.....	42
1.6.2 DFT+U 能带计算.....	46
1.6.3 DFT+U 态密度计算.....	47
1.7 结构弛豫计算.....	48
1.7.1 Si 单胞的结构弛豫计算.....	48
1.8 密度泛函微扰理论.....	50
1.8.1 准备基态的电荷密度和波函数.....	50
1.8.2 计算离子错位介电张量.....	52
1.8.3 计算动力学矩阵和波恩有效电荷.....	53
1.8.4 计算声子能带.....	53
1.8.5 计算声子态密度.....	55
1.8.6 计算低频长波的光学性质.....	56
1.8.7 计算拉曼光谱.....	57
1.9 有限位移法.....	60
1.10 Wannier 函数计算.....	62
1.10.1 自洽计算.....	62
1.10.2 wannier 函数计算.....	63
1.11 STM 仿真.....	67
1.11.1 自洽计算.....	68



1.11.2 STM 计算.....	70
1.12 贝利曲率计算.....	71
1.12.1 自洽计算.....	72
1.12.2 贝利曲率计算.....	73
1.13 加电场功能计算.....	74
1.13.1 自洽计算.....	74
1.13.2 能带计算.....	76
第二章 专题应用.....	78
第一部分：双层扭转黑磷摩尔条纹.....	78
2.1 Device Studio 建模.....	78
2.1.1 结构搭建.....	78
2.1.2 输入文件生成.....	82
2.2 RESCU 计算过程.....	84
2.2.1 自洽计算.....	84
2.2.2 能带计算.....	86
2.2.3 实空间波函数计算.....	89
2.3 原文解读.....	92
第三章 并行版安装教程.....	93
3.1 系统及环境要求.....	93
3.1.1 系统要求.....	93
3.1.2 软件环境要求.....	93
3.2 软件安装过程.....	93
3.3 软件测试及脚本编写.....	99
3.4 安装常见问题.....	100
第四章 常见问题 (FAQ)	101
4.1 计算 RESCU 安装及资料问题.....	101
4.2 RESCU 使用及计算参数.....	101
4.3 RESCU 并行版计算相关问题.....	103

第一章 基本使用

导读：

本章将介绍 RESCU 的各种功能的基本使用，具体包括：自洽计算、能带计算、态密度计算、Mulliken 电荷计算、HSE06 计算；由于 RESCU 的很多参数都有着默认值，且这些参数基本不会影响计算结果；因此在介绍功能时，本章只设置了一些特别重要的参数，同时对这些参数进行简单的讲解。

1.1 自洽

何为自洽计算？自洽计算能够得到特定晶体的电荷密度和有效哈密顿量，有了电荷密度和哈密顿量之后才能有计算该体系的能带、态密度等电子结构性质。因此特别需要注意的就是：自洽与能带、态密度等电子结构性质计算是有先后顺序的，能带、态密度等电子结构性质计算的前提条件就是需要电荷密度或者有效哈密顿量。

RESCU 软件在中内置了两种不同计算方法来进行电子结构计算，一种采用的是实空间（REAL），这种方法使用的是实空间格点基矢，其计算精度相当于使用平面波的程序，例如 VASP 等；另一种为原子轨道线性组合（LCAO），这种方法采用的是 LCAO 的基组，其计算精度相当于 Nanodcal、SIESTA 等软件。本节将使用 RESCU 中两种不同的方法（LCAO 和 REAL）进行自洽计算。

1.1.1 REAL 的自洽计算

以 CO 分子为例：

（1）准备输入文件 scf.input，结构文件 CO.xyz 和基组文件 C_DZP.mat，O_DZP.mat。scf.input 文件参考如下：

```
info.calculationType = 'self-consistent'  
domain.latvec        = [[10 0 0]; [0 10 0]; [0 0 10]]  
element(1).species   = 'C'  
element(1).path      = './C_DZP.mat'  
element(2).species   = 'O'  
element(2).path      = './O_DZP.mat'  
atom.xyz             = './CO.xyz'  
kpoint.gridn         = [1,1,1]  
domain.lowres        = 0.4  
units.length         = 'Angstrom'
```

CO.xyz 文件参考如下:

```
2  
AtomType X      Y      Z  
C         0      0      0  
O         0      0      1.128
```

Notes: 在 scf.input 文件中'**info.calcualtionType**'控制需要计算的内容, 本例为进行自洽计算; '**domain.latvec**'为晶体的晶格常数; '**element(?).species**'为该计算的晶体结构中包含了那些元素; '**element(?).path**'为指定该元素所使用的基组的路径, 该路径可以是绝对路径也可以是相对路径; '**atom.xyz**'为指定该晶体结构坐标文件的位置, 同样的, 该路径可以是绝对路径也可以是相对路径; '**kpoint.gridn**'为倒空间格点数目, '**domain.lowres**'为实空间格点数目, 这两个参数及其重要, 不仅影响计算所需时间同时也会影响计算结果, 将在之后的章节中详细介绍; '**units.length**'能够控制所有与长度量的单位, 将控制所有的长度单位。RESCU 在计算过程中默认使用的方法为 REAL。CO.xyz 中包含了原子的结构信息, 该文件的格式为: 第一行为体系包含的原子个数; 第二行为提示行, 可以书写是任何东西, 但不能省略; 从第三行开始代表了每个原子的信息, 第 1 列为原子类型, 2-4 列为原子的坐标。需要注意的是, 原子个数与第三行开始的每个原子的信息总个数一致。

(2) 自洽计算

打开 Matlab 软件, 在 matlab 中进入 CO 体系的文件目录下, 进入之后可以在 matlab 的 Current Folder 中看到为上述计算准备的 4 个输入文件 scf.input,

CO.xyz, C_DZP.mat 和 O_DZP.mat。在 Matlab 的 Command Window 中输入命令:

```
>> rescu -i scf.input
```

自洽计算开始, matlab 的 Command Window 中开始出现 RESCU 自洽的过程, 此时需要注意的是, 不要去操作这个正在运行 RESCU 的 matlab, 静待自洽计算完成。

自洽结束后, 在 CO/real 文件夹下可以得到: **self-consistent.mat**、**self-consistent.h5** 和 **resculog.out**。RESCU 将自洽过程中在 Command Window 显示的过程保存在了 **resculog.out** 中, 通过这个文件可以读取到很多有用的数据,

具体输出信息 **resculog.out** 的部分内容如下:

```

## INFO ##   % 参数显示
RESCU version (path)           = 2.0.3 (E:\rescu 更新\2.0.3\rescu-2.0.3)
Calculation typ                = self-consistent
## PSEUDOPOTENTIAL ##
Pseudopotential for  C         = E:\rescu 教程\素材\co\real\C_DZP.mat
Pseudopotential for  O         = E:\rescu 教程\素材\co\real\O_DZP.mat
## BASIS ##
Calculation basis              = REAL   % REAL 方法
## DOMAIN ##
Lattice vectors (Bohr)        = [1.890e+01,0.000e+00,0.000e+00]
                               [0.000e+00,1.890e+01,0.000e+00]
                               [0.000e+00,0.000e+00,1.890e+01]
Low resolution N               = [48,48,48]
## ATOMS ##
Number of (elements,atoms)    = (2,2)
Number of electrons            = 10
## SPIN ##
Spin type                      = degenerate
## KPOINT ##
Monkhorst-Pack grid           = [1,1,1]
K-point grid shift             = [0.00,0.00,0.00]
Reduced k-grid size            = 1
Sampling method (smearing)     = fermi-dirac (2.721e-02 eV)
.....
## MIXING ##
Mixing algorithm                = pulay
.....
Mixing tolerance (Rho,Etot)    = (1.000e-05,1.000e-05)
## OPTION ##
Maximal number of SCF iterations = 100
-----
Entering self-consistent loop at 2018:03:16::11:34:18   % 计算开始
-----
#iter | Etot      dRho      dE      dt      time
#   1 | -2.281e+01 0.000e+00 1.000e+00 1.618e+01 11:34:34
#   2 | -2.221e+01 2.665e-02 5.971e-02 1.827e+01 11:34:52
.....
#  10 | -2.171e+01 2.516e-06 3.114e-06 1.921e+01 11:37:25
-----
Total time                      187.859   % 自洽用时
It is now                       2018:03:16::11:37:26

```

输出文件的部分信息如下:

## INFO ##	版本号及所计算性质的名称;
## PSEUDOPOTENTIAL ##	基组路径基组名称等相关信息
## BASIS ##	计算方法
## DOMAIN ##	晶格常数及实空间格点信息
## ATOMS ##	元素种类, 原子数目, 电子数等信息
## SPIN ##	自旋信息
## KPOINT ##	K 点及 K 空间相关信息
## FUNCTIONAL ##	泛函相关信息
## EIGENSOLVER ##	求解本征值的相关信息及方法
## MIXING ##	迭代中混合算法及收敛精度控制
## OPTION ##	RESCU 计算过程中的控制参数

1.1.2 LCAO 的自洽计算

同样以 CO 分子为例:

(1) 准备输入文件 scf.input, 结构文件 CO.xyz 和基组文件 C_DZP.mat,

O_DZP.mat

scf.input 文件参考如下:

```
info.calculationType = 'self-consistent'  
info.savepath        = './results/CO_scf'  
domain.latvec        = [[10 0 0]; [0 10 0]; [0 0 10]]  
element(1).species   = 'C'  
element(1).path      = './C_DZP.mat'  
element(2).species   = 'O'  
element(2).path      = './O_DZP.mat'  
atom.xyz              = './CO.xyz'  
kpoint.gridn         = [1,1,1]  
domain.lowres        = 0.4  
units.length         = 'Angstrom'
```

CO.xyz 文件参考如下:

```
2
AtomType X Y Z
C 0 0 0
O 0 0 1.128
```

(2) 自洽计算

打开 Matlab 软件，进入 CO 体系的文件目录下，该目录需包含上面准备的 4 个输入文件 `scf.input`，`CO.xyz`，`C_DZP.mat` 和 `O_DZP.mat`。在 Matlab 中输入命令：

```
>> rescu -i scf.input
```

Notes: 在计算中不仅打开了 LCAO 的开关 '`LCAO.status`'，还设置了 '`info.savepath`'，这个参数表示输出文件的保存路径和命名。计算结束后，在 CO/LCAO 文件夹下可以得到输出日志文件 `resculog.out`，同时在 CO/LCAO/results 文件夹下可以得到重要输出数据文件 `CO_scf.mat` 和 `CO_scf.h5`。`mat` 文件保存了大部分重要数据，一些特别大的三维数据将保存在 `h5` 文件中，例如：电荷密度 `rho`、波函数 `psi`、密立根布居 Mulliken population、局域态密度 LDOS。

输出信息 `resculog.out` 的部分内容如下：

```

## INFO ##   % 参数显示
RESCU version (path)           = 2.0.3 (E:\rescu 更新\2.0.3\rescu-2.0.3)
Calculation type                = self-consistent
## PSEUDOPOTENTIAL ##
Pseudopotential for  C         = E:\rescu 教程\素材\co\C_DZP.mat
Pseudopotential for  O         = E:\rescu 教程\素材\co\O_DZP.mat
## BASIS ##
Calculation basis               = LCAO           % LCAO 方法
Number of basis functions       = 26
## DOMAIN ##
Lattice vectors (Bohr)         = [1.890e+01,0.000e+00,0.000e+00]
                                [0.000e+00,1.890e+01,0.000e+00]
                                [0.000e+00,0.000e+00,1.890e+01]
Low resolution N                = [48,48,48]
## ATOMS ##
Number of (elements,atoms)     = (2,2)
Number of electrons             = 10
## SPIN ##
Spin type                       = degenerate
.....
Mixing tolerance (Rho,Etot)    = (1.000e-05,1.000e-05)
## OPTION ##
Maximal number of SCF iterations = 100
-----
Entering self-consistent loop at 2018:03:16::13:43:05   % 计算开始
.....

```

参数常用设置及经验分享:

参数	含义	设置及建议
info.calculationType	计算内容	'self-consistent', 'dos' 等
info.savepath	输出文件保存位置	自定义
domain.latvec	晶格基矢	行向量形式
element(1).species	元素名	括号数字代表不同元素
element(1).path	基组的路径	绝对/相对路径均可
atom.xyz	原子结构文件	一般使用 xyz 文件形式
kpoint.gridn	倒空间格点数目	
domain.lowres	实空间格点数目	精度参数, 数越小精度越高

units.length	长度单位控制	默认 Bohr
LCAO.status	LCAO 开关	默认 REAL

1.2 能带

本节将从自洽出发介绍如何使用 RESCU 计算不同自旋状态下 Nospin, CollinearSpin, GeneralSpin+SOC 能带结构。在完成了不同自旋状态下的能带计算之后, 我们还将要学习如何计算能带去折叠, 能带投影等功能。再次强调能带计算之前必须要做自洽计算。

本章节将以下面这些常见的体系为例:

- A. 无自旋 (degenerate) 体系
 - **Si 体系**
- B. 线性自旋 (collinear) 体系
 - **Ni 体系**
- C. 一般自旋+自旋轨道耦合 (non-collinear+SOC) 体系
 - **MoS₂ 体系**
- D. 能带去折叠计算
 - **Cu₃Au 体系**
- E. 能带投影计算
 - **Si 体系**

注: 在这里我们先不使用结构驰豫来优化晶体结构, 结构驰豫将在后面的章节中进行介绍。

1.2.1 Si 体系(无自旋体系 degenerate)

自洽计算

(1) 准备输入文件 scf.input, Si.xyz 和基组文件 Si_DZP.mat

scf.input 文件参考如下:

```
info.calculationType = 'self-consistent'  
info.savepath        = './results/Si_scf'  
domain.latvec       = [[0 2.715 2.715]; [2.715 0 2.715]; [2.715 2.715 0]]  
element(1).species  = 'Si'  
element(1).path     = './Si_DZP.mat'  
atom.fracxyz        = './Si.xyz'  
kpoint.gridn        = [7,7,7]  
domain.lowres       = 0.6  
units.latvec        = 'Angstrom'
```

Si.xyz 文件参考如下:

```
2  
AtomType X Y Z  
Si 0 0 0  
Si 0.25 0.25 0.25
```

Notes: (1) 在这个例子中 Si 原子坐标采用了分数坐标的形式, 因此在计算中需要使用'**atom.fracxyz**' 参数。(2) RESCU 中有三个控制长度单位的参数'**units.latvec**'是控制晶格常数单位; '**units.xyz**'控制 xyz 坐标的单位, '**units.length**'控制所有长度单位。

(2) 自洽计算

打开 Matlab 软件, 进入 Si 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i scf.input
```

自洽结束后, 会产生主要的输出文件: **Si_scf.mat** 和 **Si_scf.h5**

能带计算

(1) 准备输入文件 band.input

band.input 文件参考如下:

```
info.calculationType = 'band-structure'  
info.savepath        = './results/Si_bs'  
rho.in               = 'results/Si_scf'
```

Notes: 对于晶体结构类型: 'CUB'、'FCC'、'BCC'、'TET'、'BCT1'、'BCT2'、'ORC'、'ORCF1'、'ORCF2'、'ORCF3'、'ORCI'、'ORCC'、'HEX'、'RHL1'、'RHL2', RESCU 程序本身会自动识别晶格类型和高对称点, 故可以在不设置'**kpoint.sympoints**'的情况下进行能带计算。

(2) 能带计算

打开 Matlab 软件，进入 Si 体系的文件目录下，在 Matlab 中输入命令：

```
>> rescu -i band.input
```

能带计算结束后，会产生主要的输出文件：**Si_bs.mat**、**Si_bs.h5** 和 **BandStructure.txt**；在 Matlab 中输入命令：

```
>> rescu -p ./results/Si_bs.mat
```

得到能带结构图如下：

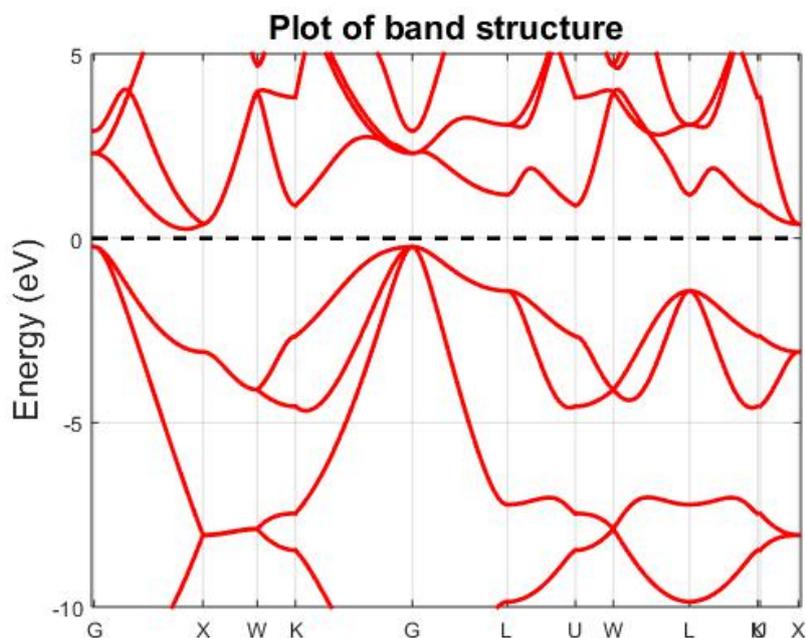


图 1-1 Si 的能带图

打开 Si 文件夹下的 **BandStructure.txt** 文件，可以看到 **BandStructure.txt** 有 524 行数据，第 1 行只有一个数字 12，代表高对称点数目；第 2 行是高对称点的字母表示；第 3 行是高对称点对应的序号；第 4 行有三个数字 260、8、1，其分别表示 K 点数目能带数目、体系是否有自旋，1 表示体系没有自旋；第 5-264 行是 260 个 K 点的坐标；第 265-524 行是 8 条能带的的数据。

同时 RESCU2020A 用户还可以通过 DeviceStudio2020A 中的 Anaylsis plot 显示 **BandStructure.txt**，具体操作如下：**Simulator**→**RESCU**→**Anaylsis plot**。

1.2.2 Ni 体系(线性自旋体系 collinear)

自洽计算

(1) 准备输入文件 `scf.input`, `Ni.xyz` 和基组文件 `Ni_DZP.mat`

`scf.input` 文件参考如下:

```
info.calculationType = 'self-consistent'  
info.savepath        = './results/Ni_scf'  
domain.latvec        = [[0 1.762 1.762]; [1.762 0 1.762]; [1.762 1.762 0]]  
element(1).species   = 'Ni'  
element(1).path      = './Ni_DZP.mat'  
atom.xyz             = './Ni.xyz';  
spin.type            = 'collinear';  
atom.magmom          = './Ni.xyz';  
kpoint.gridn         = [9,9,9]  
domain.lowres        = 0.35  
units.length         = 'Angstrom'  
LCAO.status          = 1
```

`Ni.xyz` 文件参考如下:

```
1  
AtomType X Y Z Magmom  
Ni 0 0 0 2
```

Notes: 自旋类型'`spin.type`' 分为'`degenerate`', '`collinear`', '`non-collinear`', 其默认值为'`degenerate`'. 本例 Ni 采用线性自旋, 即'`non-collinear`'; 在 RESCU 中可以通过 '`atom.magmom`', '`atom.magmomd`', '`atom.magmomcart`' 三个参数来设置磁矩, 常用的是 '`atom.magmom`'. 此时, 我们需要在 `xyz` 文件的第 5 行定义线性自旋磁矩的大小。

(2) 自洽计算

打开 Matlab 软件, 进入 Ni 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i scf.input
```

能带计算

(1) 准备输入文件 `band.input`

band.input 文件参考如下:

```
info.calculationType = 'band-structure'  
info.savepath        = './results/Ni_bs'  
rho.in               = 'results/Ni_scf'  
kpoint.sympoints     = {'W', 'L', 'G', 'X', 'W', 'K'}  
kpoint.gridn         = 250
```

Notes: RESCU 程序本身会自动识别晶格类型和高对称点, 用户可以使用字母来表示对应高对称点坐标, 本例中 `kpoint.sympoints={'W', 'L', 'G', 'X', 'W', 'K'}`。在能带计算中'`kpoint.gridn`'为 1*1 的数字, 表示的是在高对称路径上总的 K 点数; 而在自洽中该参数为 1*3 的矩阵表示为 K 网格的大小。

(2) 能带计算

打开 Matlab 软件, 进入 Ni 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i band.input
```

能带计算结束后, 在 Matlab 中输入命令:

```
>> rescu -p ./results/Ni_bs.mat
```

这里./results 只是表示 Ni_bs.mat 存在的路径, 得到能带结构图如下:

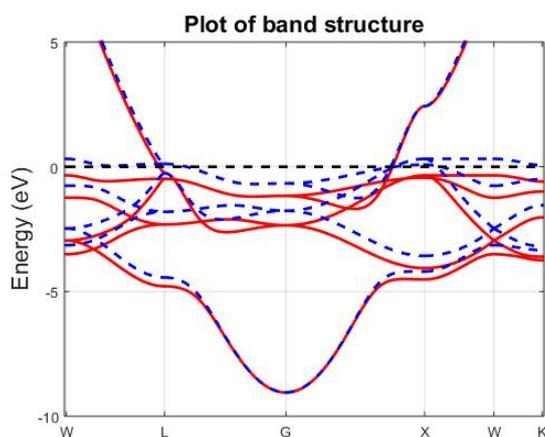


图 1-2 Ni 的能带图

1.2.3 MoS₂ 体系(一般自旋+自旋轨道耦合体系 non-collinear+SOC)

自洽计算

(1) 准备输入文件 scf.input, MoS2.xyz 和基组文件 Mo_DZP.mat, S_DZP.mat

scf.input 文件参考如下:

```
info.calculationType = 'self-consistent'  
info.savepath       = './results/MoS2_scf'  
domain.latvec      = [[2.753 -1.589 0]; [2.753 1.589 0]; [0 0 20]]  
element(1).species = 'Mo'  
element(1).path    = './Mo_DZP.mat'  
element(2).species = 'S'  
element(2).path    = './S_DZP.mat'  
atom.fracxyz       = './MoS2.xyz';  
spin.type          = 'non-collinear';  
atom.magmom        = './MoS2.xyz';  
spin.SOI           = 1  
kpoint.gridn       = [5,5,1]  
domain.lowres      = 0.35  
units.length       = 'Angstrom'  
LCAO.status        = 1
```

MoS2.xyz 文件参考如下:

```
3  
AtomType      X      Y      Z      Magmom  Phi  Theta  
Mo      1.70937290      0.06791215      11.89143508      0.1  1  1  
S      3.54500964      0.06789605      10.32652746      0.1  1  1  
S      3.54500969      0.06789596      13.45661181      0.1  1  1
```

Notes: 当考虑自旋轨道耦合 SOI 时,自旋类型'spin.type'必须设置为'non-collinear'。如果只想设置本例中通过设置了'non-collinear',可以将 spin.SOI=1 参数注释掉或设置成 0; 本例中设置了 spin.SOI=1 打开自旋轨道耦合。同时需要注意的是,由于设置的是'non-collinear'的自旋,此时自旋是带有方向的,因此在 xyz 文件中的第 5-7 行都是用来描述磁矩的。

(2) 自洽计算

打开 Matlab 软件,进入 MoS₂ 体系的文件目录下,在 Matlab 中输入命令:

```
>> rescu -i scf.input
```

能带计算

(1) 准备输入文件 band.input

band.input 文件参考如下:

```
info.calculationType = 'band-structure'  
info.savepath        = './results/MoS2_bs'  
rho.in               = 'results/MoS2_scf'  
kpoint.symptoms     = {[0 0 0];[1/2 0 0];[2/3 1/3 0];[0 0 0]}  
kpoint.gridn        = [51,51,51]  
eigensolver.emptyBand = 25
```

Notes:(1)本例子中通过直接写坐标的方法 $kpoint.symptoms = \{[0\ 0\ 0]; [1/2\ 0\ 0]; [2/3\ 1/3\ 0]; [0\ 0\ 0]\}$ 来定义高对称点, 在本例中 $[0\ 0\ 0]; [1/2\ 0\ 0]; [2/3\ 1/3\ 0]$ 分别代表高对称点 'G', 'M', 'K'; (2)同时, 本例通过 $kpoint.gridn = [51, 51, 51]$ 定义每两个高对称点之间的 K 点数目; 而在 Ni 的案例中我们只设置了一个总的 K 点数目; (3)在本例中还用到一个参数 'eigensolver.emptyBand', 这个参数是用来定义空带的数目。对于半导体来说导带没有电子占据, 因此在计算过程中导带数目的多少不会影响整个体系的收敛和最后的哈密顿量参数影响, 而这 'eigensolver.emptyBand' 这个参数就是增加没有电子占据空带数目的参数。

(2) 能带计算

打开 Matlab 软件, 进入 MoS₂ 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i band.input
```

能带计算结束后, 在 Matlab 中输入命令:

```
>> rescu -p ./results/MoS2_bs.mat
```

得到能带结构图如下

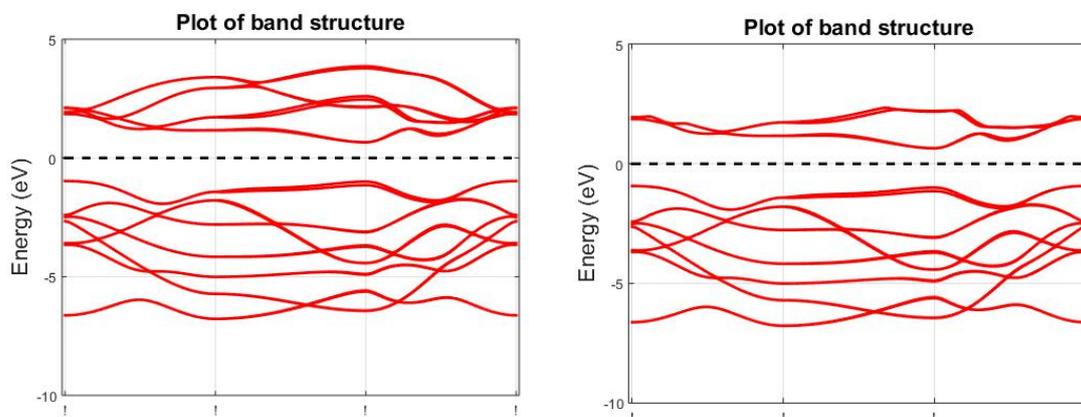


图 1-3 MoS₂ 的能带图; (左) emptyBand=25 (右) emptyBand=16

可以看到通过直接写坐标的方法画出来的能带图的高对称点都是'!', 因此需要手动加上高对称点。在 Matlab 中输入命令:

```
>> rescu -p ./results/MoS2_bs.mat  
>> set(gca,'XTickLabel',{'G','M','K','G'});
```

重新得到能带结构图如下:

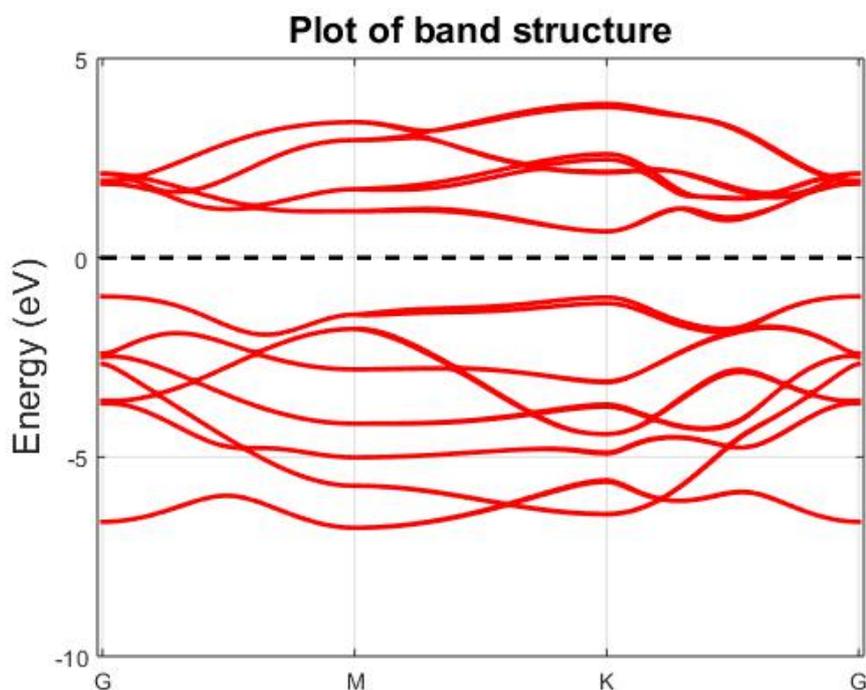


图 1-4 修改后的 MoS₂ 能带图

1.2.4 Si 体系(能带投影计算)

所谓能带投影就是在原有的能带基础上去细分每条能带上轨道的贡献, 通过能带投影计算我们可以更加直观的得到能带中轨道的贡献。

(1) 以 Si 体系为例, 进行自洽计算 (见 1.2.1 Si 体系), 自洽完成之后准备准备能带投影计算。

(2) 准备能带投影输入文件 band_pw.input

band_pw.input 文件参考如下：

```
info.calculationType = 'band-structure'  
info.savepath        = './results/Si_bs'  
rho.in               = 'results/Si_scf'  
kpoint.gridn = 150;  
kpoint.symptoms = {'L','G','X','W','K'};  
option.savePartialWeights = true
```

Notes: 计算能带投影的时候只要打开能带投影的开关 `option.savePartialWeights` 即可，其他参数和普通能带计算的参数一样。

(3) 能带投影计算

打开 Matlab 软件，进入 Si 体系的文件目录下，在 Matlab 中输入命令：

```
>> rescu -i band_pw.input
```

(4) 画能带投影图

由于能带投影计算是将能带投影到每个原子轨道上面得到每个 K 点下每个原子轨道对本征值的贡献，因此能带投影图可以有很多种花哨的玩法，比如最基本的研究 S, P, D 轨道的贡献；同时也可以更详细的去区分 P 轨道或者 D 轨道；也可以去查看是不同元素或者原子的轨道贡献。我们这边介绍最基本的 S, P, D 轨道的贡献图如何用脚本实现，其余的玩法还需用户自己体会。

打开 Matlab 软件，进入 Si 体系的文件目录下，运行 band_pw.m 脚本：

```
>> run band_pw
```

得到 S, P, D 能带投影图：

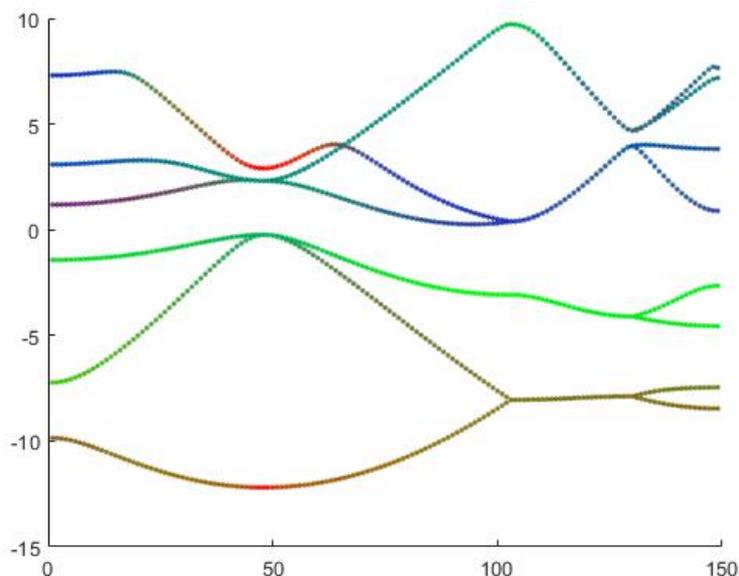


图 1-6 Si 体系 S, P, D 能带投影图

其中红, 绿, 蓝分别对应着 S, P, D 轨道的贡献。

Band.pw.m 脚本具体内容如下:

```

1. clear; close all
2. % load
3. matpath = './results/Si_bs.mat';
4. h5path = './results/Si_bs.h5';
5. mrc = load(matpath);
6. % parse results
7. ksnrg = mrc.band.ksnrg;
8. kdir = mrc.band.kdirect; nk = size(kdir,1);
9. ikdir = mrc.kpoint.ikdirect; nik = size(ikdir,1);
10. [~,permk] = ismember(mrc.band.kdirect,ikdir,'rows');
11. tspin = mrc.spin.type;
12. if strcmp(tspin,'degenerate')
13. ns = 1;
14. else
15. ns = 2;
16. end
17. % parse orbital info
18. Aorb = mrc.LCAO.orbInfo.Aorb;
19. Lorb = mrc.LCAO.orbInfo.Lorb;
20. Morb = mrc.LCAO.orbInfo.Morb;
21. Oorb = mrc.LCAO.orbInfo.Oorb;
22. Rorb = mrc.LCAO.orbInfo.Rorb;
  
```

```

23. Sorb = mrc.LCAO.orbInfo.Sorb;
24. Zorb = mrc.LCAO.orbInfo.Zorb;
25. % load weights and plot
26. figure(); hold on; marker = {'!','--'};
27. for kk = 1:nk
28. kk
29. [~,irk] = ismember(kdir(kk,:),ikdir,'rows');
30. for ss = 1:ns
31. ii = sub2ind([nik,ns],irk,ss);
32. pw = loadDistArray(h5path,['/LCAO/partialWeight',num2str(ii)]);
33. nb = pw.n; pw = pw.data;
34. %%%%%%%%%%%
35. % PARTITION %
36. %%%%%%%%%%%
37. pwtmp = 0*pw(1:3,:);
38. ind = Lorb == 0 ; % s
39. pwtmp(1,:) = sum(pw(ind,:));
40. ind = Lorb == 1 ; % p
41. pwtmp(2,:) = sum(pw(ind,:));
42. ind = Lorb == 2; % d
43. pwtmp(3,:) = sum(pw(ind,:));
44. pw = abs(pwtmp);
45. %%%%%%%%%%%
46. %%%%%%%%%%%
47. for bb = 1:nb
48. color = [pw(1,bb) pw(2,bb) pw(3,bb)]/sum(pw(:,bb));
49. plot(kk,ksnrg(kk,bb),marker{ss},'color',color);
50. end
51. end
52. end

```

Notes: (1) RESCU 的能带数据和 K 点的信息都保存在 mat 文件中，每个原子轨道在每个 K 点每个能带的贡献保存在 h5 文件中；(2) 脚本的前 33 行都是在读取 RESCU 计算完的数据，34 行-44 行为处理能带贡献，将能带贡献以 S, P, D 轨道区分开来，45 行-52 行是进行画图处理。(3) 如果需要计算能带投影，用户则需要了解以下知识，RESCU 将每个轨道按照原子，主量子数，磁量子数，原子线性轨道，轨道截断，元素种类这些判据进行了分类。这些分类信息保存在 mat 文件中 LCAO.orbInfo 标签下。其对应关系如图所示。

Field	Index
Aorb	atom
Lorb	L
Morb	L_z
Oorb	basis
Rorb	cutoff radius
Sorb	species
Zorb	zeta

图 1-7 LCAO.orbInfo 标签的对应关系

在 RESCU2020A 版本中，能带投影计算完成之后将会在 results 文件夹下生成 PartialWeightsBandStructure.txt 文件，使用 DeviceStudio2020A 可以直接显示能带投影计算。具体操作如下：Simulator → RESCU → Analysis Plot → 选择 PartialWeightsBandStructure.txt 文件导入。

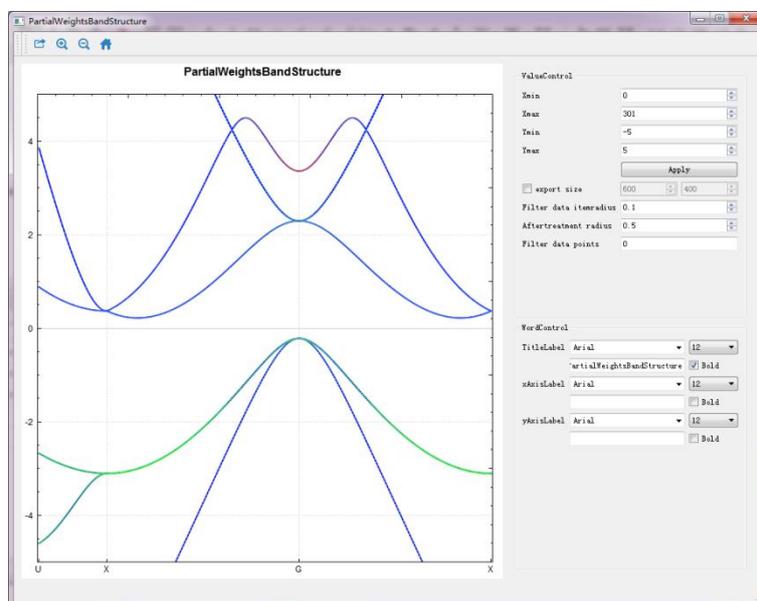


图 1-8 DeviceStudio 导入 Si 体系 S, P, D 能带投影图

其中右边工具栏的 Filter data itemradius、Aftertreatment readius、Filter data points 等设置可以对图进行修改。

参数常用设置及经验分享:

参数	含义	设置及建议
atom.fracxyz	原子结构文件	必须为分数坐标
units.latvec	晶格常数单位控制	默认 Bohr
units.xyz	xyz 文件单位控制	默认 Bohr
rho.in	引入之前计算的结果	本章中调用 scf 结果计算 band
spin.type	自旋类型	默认 degenerate
atom.magmom	磁矩	xyz 文件中的 5-7 列代表磁矩
kpoint.symptoms	高对称点	可以使用字母或坐标
spin.SOI	自旋轨道耦合	默认关闭
eigensolver.emptyBand	空带数目	体系越大空带数目越多
functional.list	泛函选取	固定格式, 见常见问题
domain.primLatvec	原胞的晶格常数	请确认原胞的晶格常数
option.savePartialWeights	能带投影开关	能带投影开关

1.3 态密度

本节将介绍电子结构分析中的重要组成——态密度 (DOS) 计算。态密度 (DOS) 需要在自洽完成的基础上进行计算。DOS 计算常见的三种情况是: (1) 总态密度 (Total DOS); (2) 投影态密度 (Projected DOS); (3) 局域态密度 (Local DOS)。

1.3.1 Si 总态密度计算

- (1) 以 Si 体系为例, 进行自洽计算 (见 1.2.1 Si 体系)
- (2) 准备输入文件 dos.input

```
info.calculationType = 'dos'  
info.savepath        = './results/Si_dos'  
rho.in               = './results/Si_scf'  
kpoint.sampling      = 'tetrahedron+blochl'  
kpoint.gridn         = [15,15,15]  
smearing.sigma       = 0.01;  
dos.range             = [-5 , 5]  
dos.resolution       = 0.01  
units.energy         = 'eV'
```

Notes: (1) 态密度计算时需要加密 K 点, 如果自洽计算时 `kpoint.gridn` 设置大于 1 时, 态密度计算时 `kpoint.gridn` 应是自洽值的两倍或两倍以上。(2) 不同的材料可以选择不同的 `kpoint.sampling` 方法, RESCU 支持 'fermi-dirac', 'gauss', 'methfessel-paxton', 'tetrahedron', 'tetrahedron+blochl' 这五种 K 空间取样方法。(3) `smearing.sigma` 表示的展宽, `sigma` 越大, 态密度越平滑。选择 'tetrahedron', 'tetrahedron+blochl' 方法是 `sigma` 为 0。(4) 态密度计算中涉及到多个能量单位的控制。例如, 'units.sigma' 用来控制 'smearing.sigma' 的单位; 'units.dos.range' 用来控制 'dos.range' 的单位; 'units.dos.resolution' 用来控制 'dos.resolution' 的单位; 在本例中通过使用 'units.energy' 来控制所有能量单位。

(3) 总态密度计算

打开 Matlab 软件, 进入 Si 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i dos.input
```

态密度计算结束后, 会产生主要的输出文件:

Si_dos.mat、**Si_dos.h5** 和 **DensityOfStates.txt**

在 Matlab 中输入命令:

```
>> rescu -p ./results/Si_dos.mat
```

得到总的态密度图如下:

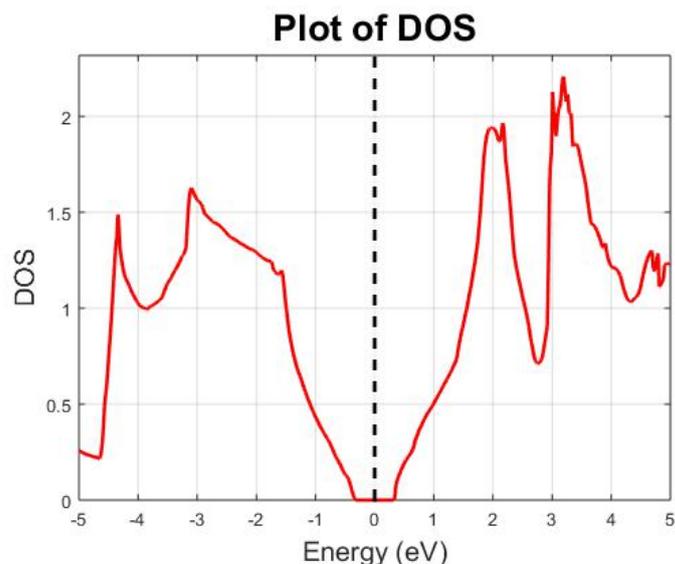


图 1-9 Si 的总态密度图

打开 DensityOfStates.txt 文件，可以看到 DensityOfStates.txt 有 1001 行数据，第 1 行有三个数字 1000 1 1，1000 代表指定能量范围内的能量点个数、1 代表总态密度、1 代表 degenerate；第 2-1001 行表示每一个能量点对应的态密度值。

同时 RESCU2020A 用户还可以通过 DeviceStudio2020A 中的 Analysis plot 显示 DensityOfStates.txt，具体操作如下：Simulator→RESCU→Analysis plot。

1.3.2 graphene 投影态密度计算

(1) 准备输入文件 scf.input，graphene.xyz 和基组文件 C_PBE_DZP.mat

scf.input 文件参考如下：

```

info.calculationType = 'self-consistent'
info.savepath        = './results/graphene_scf'
domain.latvec        = [[2.4612 0 0]; [-1.2306 2.13146172 0]; [0 0 15]]
element(1).species   = 'C'
element(1).path      = './C_PBE_DZP.mat'
functional.list       = {'XC_GGA_X_PBE','XC_GGA_C_PBE'}
atom.xyz             = './graphene.xyz';
kpoint.gridn         = [9,9,1]
domain.lowres        = 0.3
units.length         = 'Angstrom'
LCAO.status          = 1
  
```

graphene.xyz 文件参考如下：

```
2
AtomType X Y Z
C 0.6153000000 0.3552436206 3.3545000000
C 0.6153000000 1.7762181032 3.3545000000
```

Notes: RESCU 默认使用 LDA 的泛函，如果需要使用 GGA, HSE06 等其他泛函则需要开启泛函设置参数'functional.list', 并行版则还需要开启 libxc 环境调用参数'functional.libxc', 目前由于 windows 环境下不能按照 libxc 环境, 因此单机试用版 RESCU 不需要开启'functional.libxc'参数, 并且此版本只支持 LDA, GGA 等少量泛函。

(2) 自洽计算

打开 Matlab 软件, 进入 graphene 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i scf.input
```

自洽结束后, 会产生主要的输出文件: **graphene_scf.mat** 和 **graphene_scf.h5**

(2) 准备输入文件 pdos.input

```
info.calculationType = 'dos'
info.savepath        = './results/graphene_pdos'
rho.in              = 'results/graphene_scf'
kpoint.sampling     = 'tetrahedron '
kpoint.gridn        = [21,21,1]
smearing.sigma      = 0.1
dos.range            = [-5 , 5]
dos.resolution       = 0.01
dos.projL            = [0,1]
dos.projM            = [-1,0,1]
units.energy         = 'eV'
```

Notes: 在 RESCU 可以投影到: 角量子数'dos.projL', 磁量子数'dos.projM', 原子轨道的 Zeta 值'dos.projZ ', 不同原子'dos.projAtom', 不同元素种类'dos.projSpecies'。这五个参数可以组合使用, 例如想要计算一个体系第 10-20 个原子的 5 个 d 轨道分别的贡献, 此时需要设置 dos.projL = [2], dos.projM = [-2:2], dos.projAtom = [10:20]。

这里简单介绍下[-2:2], 在 matlab 中[-2:2]等价于[-2:1:2], 意思是生成一个从-2 到 2 的数组, 其步长为 1。即[-2:2] = [-2,-1,0,1,2]。

实系数球谐函数如下图所示:

l	m	Y_l^m	Y_l^m
0	0	Y_0^0	$= \frac{1}{2}\sqrt{\frac{1}{\pi}}$
1	-1	$i\sqrt{\frac{1}{2}}(Y_1^{-1} + Y_1^1)$	$= \sqrt{\frac{3}{4\pi}} \cdot \frac{y}{r}$
1	0	Y_1^0	$= \sqrt{\frac{3}{4\pi}} \cdot \frac{z}{r}$
1	1	$\sqrt{\frac{1}{2}}(Y_1^{-1} - Y_1^1)$	$= \sqrt{\frac{3}{4\pi}} \cdot \frac{x}{r}$
2	-2	$i\sqrt{\frac{1}{2}}(Y_2^{-2} - Y_2^2)$	$= \frac{1}{2}\sqrt{\frac{15}{\pi}} \cdot \frac{xy}{r^2}$
2	-1	$i\sqrt{\frac{1}{2}}(Y_2^{-1} + Y_2^1)$	$= \frac{1}{2}\sqrt{\frac{15}{\pi}} \cdot \frac{yz}{r^2}$
2	0	Y_2^0	$= \frac{1}{4}\sqrt{\frac{5}{\pi}} \cdot \frac{-x^2 - y^2 + 2z^2}{r^2}$
2	1	$\sqrt{\frac{1}{2}}(Y_2^{-1} - Y_2^1)$	$= \frac{1}{2}\sqrt{\frac{15}{\pi}} \cdot \frac{zx}{r^2}$
2	2	$\sqrt{\frac{1}{2}}(Y_2^{-2} + Y_2^2)$	$= \frac{1}{4}\sqrt{\frac{15}{\pi}} \cdot \frac{x^2 - y^2}{r^2}$
3	-3	$i\sqrt{\frac{1}{2}}(Y_3^{-3} + Y_3^3)$	$= \frac{1}{4}\sqrt{\frac{35}{2\pi}} \cdot \frac{(3x^2 - y^2)y}{r^3}$
3	-2	$i\sqrt{\frac{1}{2}}(Y_3^{-2} - Y_3^2)$	$= \frac{1}{2}\sqrt{\frac{105}{\pi}} \cdot \frac{xyz}{r^3}$
3	-1	$i\sqrt{\frac{1}{2}}(Y_3^{-1} + Y_3^1)$	$= \frac{1}{4}\sqrt{\frac{21}{2\pi}} \cdot \frac{y(4z^2 - x^2 - y^2)}{r^3}$
3	0	Y_3^0	$= \frac{1}{4}\sqrt{\frac{7}{\pi}} \cdot \frac{z(2z^2 - 3x^2 - 3y^2)}{r^3}$
3	1	$\sqrt{\frac{1}{2}}(Y_3^{-1} - Y_3^1)$	$= \frac{1}{4}\sqrt{\frac{21}{2\pi}} \cdot \frac{x(4z^2 - x^2 - y^2)}{r^3}$
3	2	$\sqrt{\frac{1}{2}}(Y_3^{-2} + Y_3^2)$	$= \frac{1}{4}\sqrt{\frac{105}{\pi}} \cdot \frac{(x^2 - y^2)z}{r^3}$
3	3	$\sqrt{\frac{1}{2}}(Y_3^{-3} - Y_3^3)$	$= \frac{1}{4}\sqrt{\frac{35}{2\pi}} \cdot \frac{(x^2 - 3y^2)x}{r^3}$

图 1-10 实系数球谐函数对应关系图

(3) 投影态密度计算

打开 Matlab 软件, 进入 graphene 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i pdos.input
```

投影态密度计算结束后, 会产生主要的输出文件:

graphene_pdos.mat、**graphene_pdos.h5** 和 **DensityOfStates.txt**

在 Matlab 中输入命令:

```
>> rescu -p ./results/ graphene_pdos.mat
```

此时可以分别得到总态密度和投影态密度如下:

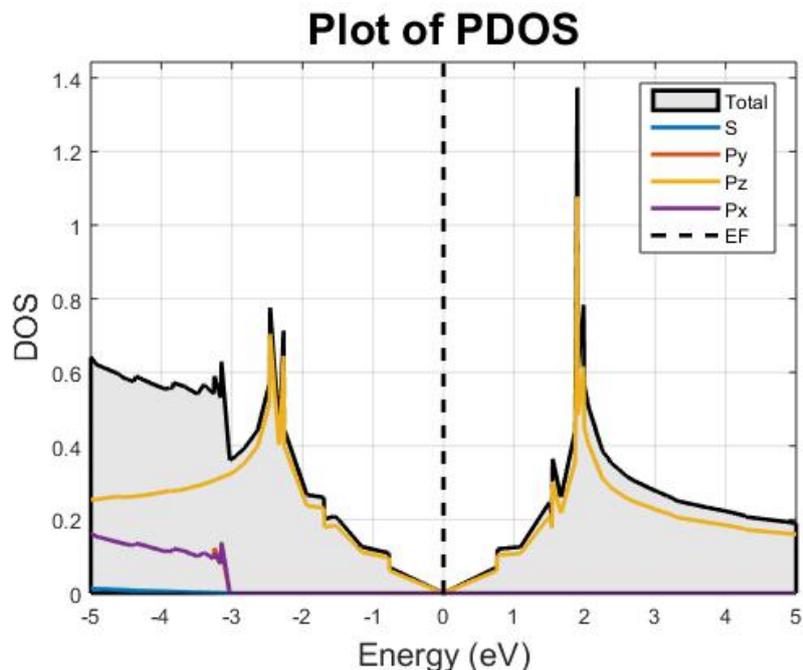


图 1-11 graphene 的总态密度图和投影态密度

可以看到石墨烯狄拉克锥附近的主要贡献来自于 p_z 轨道。

打开 DensityOfStates.txt 文件，可以看到 DensityOfStates.txt 有 2001 行数据，第 1 行有三个数字 1000 5 1，1000 代表指定能量范围内的能量点个数、5 分别代表总态密度和 s, p_y , p_z , p_x 四个轨道的态密度、1 代表没开自旋；第 2-6 行表示的每列态密度的命名：首先，数据总共有 5 列表示 5 条态密度，对于 Atom, Species, Zeta -1 表示 false，而 L 和 M -4 表示 false，所以第 1 列所有都是 false 表示总的态密度，第二列 L 和 M 都是 0，找出对应的球谐函数，即为 s 轨道的态密度；第 7-1006 行表示每一个能量点对应的态密度值，从第二行开始，第 1-6 列分别代表能量点、总态密度、s 轨道、 p_y 轨道、 p_z 轨道、 p_x 轨道。

1.3.3 graphene 局域态密度计算

本小节将计算 graphene 费米能级处的局域态密度。

- (1) 同样以 graphene 体系为例，进行自洽计算（见 1.3.2 graphene 体系）
- (2) 提取费米能级，在 Matlab 中输入命令：

```

>>load ./results/graphene_scf.mat
>>energy.EFermi
  
```

可以得到石墨烯的费米能级的能量为 $E_f = -0.0914$ Hartree。Hartree 换算成 eV 需要乘以 27.211。

(2) 准备输入文件 ldos.input

```
info.calculationType = 'dos'  
info.savepath       = './results/graphene_ldos'  
rho.in              = 'results/graphene_scf'  
kpoint.sampling     = 'tetrahedron'  
kpoint.gridn        = [21,21,1]  
smearing.sigma      = 0.01;  
dos.ldosEnergy       = -0.0914*27.211  
units.energy         = 'eV'  
dos.ldos             = true
```

Notes: (1) 计算局域态密度时, 必须开启'dos.ldos'参数; (2) 当 'dos.ldosEnergy' 参数缺失时, 默认计算费米能级处的局域态密度, 因此在这个案例中去掉 'dos.ldosEnergy' 得到的结果是相同的; (3) 当某些情况下不知道想要计算的能量的具体值时, 可以组合使用'dos.ldosEnergyi', 'dos.ldosEnergyk'这两个参数, 其意义是计算某条能带某个 K 点对应的能量下的局域态密度。

(3) 局域态密度计算

打开 Matlab 软件, 进入 graphene 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i ldos.input
```

态密度计算结束后, 会产生主要的输出文件:

graphene_dos.mat、**graphene_dos.h5**

(4) 局域态密度数据处理

由于局域态密度的值保存在每个实空间格点中, 数据会比较大, 因此 RESCU 将其保存在 **graphene_dos.h5** 文件中基本信息仍然保存在 **graphene_dos.mat** 文件中; 为了可视化局域态密度的结果, 可通过 ldos.m 脚本将 RESCU 的计算结果保存 vasp 后缀形式, 将 vasp 后缀拖入可视化软件 VESTA 中, 具体如下:

打开 Matlab 软件, 进入 graphene 体系的文件目录下, 运行 ldos.m 脚本:

```
>> run ldos
```

得到局域态密度如图所示:

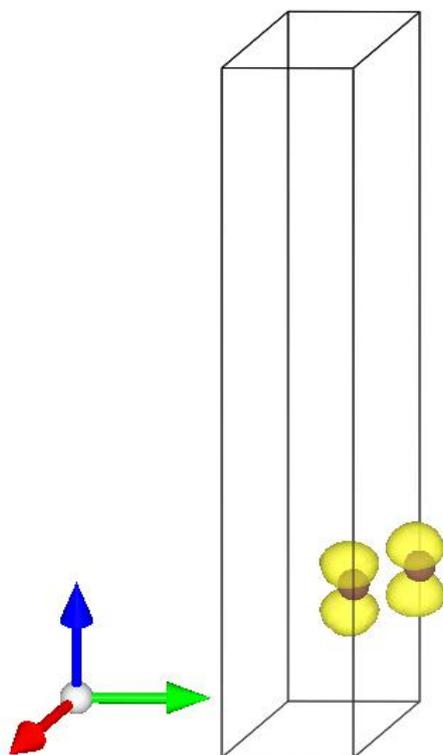


图 1-8 graphene 费米能级处的局域态密度态密度图

从费米能级处的局域态密度可以看到，费米能级处的主要贡献来自于 p_z 这和投影态密度中的到的结果吻合。

ldos.m 脚本如下：

```
1. clear all;
2. %%%%%%%%%%%PARAMETER REGION%%%%%%%%%
3. output_name = './results/graphene_ldos'; %RESCU 输出文件名,不带后缀名
4. %%%%%%%%%%%
5. matfile = [output_name, '.mat'];
6. load (matfile);
7. h5file = [output_name, '.h5'];
8. dos_data = loadDistArray(h5file, '/dos/ldosVal');
9. filename = {'ldos.vasp', 'ldos2.vasp'};
10. for j = 1:size(dos_data, 2);
11. %%%%%%%%%%%下面部分 VASP CONTCAR 形式
12. fid = fopen(filename{j}, 'w');
13. fprintf(fid, 'LDOS \n');
14. fprintf(fid, ' %6.8f\n', 1.0000000);
15. for i = 1:3
16. a = domain.latvec(i, :) * 0.529177;
17. fprintf(fid, ' %6.10f %6.10f %6.10f\n', a);
18. end
```

```

19. atom_species = size(element,2);
20. number_of_atoms = size(atom.xyz,1);
21. each_atom = hist(atom.element,unique(atom.element));
22. for i = 1:atom_species
23. fprintf(fid, '%s ', element(i).species);
24. end
25. fprintf(fid, '\n');
26. for i = 1:atom_species
27. fprintf(fid, '%6.0f ', each_atom(i));
28. end
29. fprintf(fid, '\n');
30. fprintf(fid, 'Cartesian\n');
31. for i = 1:number_of_atoms
32. b = atom.xyz(i,:)*0.529177;
33. fprintf(fid, ' %6.9f %6.9f %6.9f\n', b);
34. end
35. fprintf(fid, '\n');
36. %%%%%%%%%%写入实空间格点数
37. realgrid = domain.fgridn;
38. fprintf(fid, '%6.0f %6.0f %6.0f\n', realgrid);
39. %%%%%%%%%%ldos 写入, 10 个数据一行
40. dosxyz = dos_data.data(:,j);dosxyz = dosxyz.*conj(dosxyz);
41. fprintf(fid, '%f %f %f %f %f %f %f %f %f %f\n',dosxyz);
42. fclose(fid);
43. end

```

参数常用设置及经验分享:

参数	含义	设置及建议
kpoint.sampling	K 空间取样方法	不同的体系设置不同
smearing.sigma	展宽	按体系设置一般在 0.01--0.1eV
dos.range	态密度计算范围	按体系设置一般在-5--5eV
dos.resolution	态密度精度	每两个能量点之间的间隔
functional.list	泛函选择	常用的 LDA 和 GGA
functional.libxc	Libxc 泛函库	调用 Libxc 泛函库中的泛函
dos.projL	投影到角量子数	dos.projL=[0,1,2]

dos.projM	投影到磁量子数	dos.projM=[-2,-1,0,1,2]
dos.projAtom	投影到原子	dos.projAtom=[10:20]
dos.projSpecies	投影到元素	序号与 element 参数中的数相同
dos.ldos	LDOS 计算	LDOS 控制开关
dos.ldosEnergy	某个能量下的 LDOS	此能量须为未减去 E_f 前的能量
dos.ldosEnergyi	某条能带下的 LDOS	与 dos.ldosEnergyk 组合使用
dos.ldosEnergyk	某个 K 点下的 LDOS	与 dos.ldosEnergyi 组合使用

1.4 Mulliken 电荷及实空间电荷处理

Mulliken 电荷计算需要在自洽完成的基础上进行计算，Mulliken 电荷计算可以将电荷投影到原子轨道上。本节中将以 GaAs 体系为例介绍：（1）Mulliken 电荷计算及数据处理；（2）实空间电荷密度数据处理。

1.4.1 Mulliken 电荷及数据处理

自洽计算

（1）准备输入文件 scf.input， GaAs.xyz 和基组文件 As_DZP.mat， Ga_DZP.mat
scf.input 文件参考如下：

```
info.calculationType = 'self-consistent'
info.savepath       = './results/GaAs_scf'
domain.latvec       = [[0 2.8266 2.8266]; [2.8266 0 2.8266]; [2.8266 2.8266 0]]
element(1).species  = 'Ga'
element(1).path     = './Ga_DZP.mat'
element(2).species  = 'As'
element(2).path     = './As_DZP.mat'
atom.xyz            = './GaAs.xyz'
kpoint.gridn        = [9,9,9]
domain.lowres       = 0.3
LCAO.status         = 1
units.length        = 'Angstrom'
```

GaAs.xyz 文件参考如下：

```
2
AtomType  X Y Z
As         2.11995000    2.11995000    2.11995000
Ga         0.70665000    0.70665000    0.70665000
```

(2) 自洽计算

打开 Matlab 软件，进入 GaAs 体系的文件目录下，在 Matlab 中输入命令：

```
>> rescu -i scf.input
```

Mulliken 电荷计算

(1) 准备输入文件 mulliken.input

mulliken.input 文件参考如下：

```
info.calculationType    = 'mulliken'
info.savepath           = './results/mulliken'
rho.in                  = './results/GaAs_scf.mat'
```

Notes: mulliken 计算是将电荷密度分布到每个原子轨道上，与能带投影一样，可以对原子轨道进行重新排列组合就能得到很多种电荷分布，例如，每个原子的电荷分布，S，P，D 轨道的电荷分布等。

(2) Mulliken 电荷计算

打开 Matlab 软件，进入 GaAs 体系的文件目录下，在 Matlab 中输入命令：

```
>> rescu -i mulliken.input
```

Mulliken 电荷计算结束后，可以得到每个原子轨道上的电荷，通过读取 h5 文件中的/LCAO/mullikenPop1 数据，我们可以知道该矩阵的大小为 31*31；下面我们举例怎么得到 Ga 和 As 的电荷密度：

打开 Matlab 软件，进入 GaAs 体系的文件目录下，运行 chargetest.m 脚本：

```
>> run chargetest
```

得到 Ga 的电荷为 6.1346，As 的电荷为 11.8654。

chargetest.m 脚本如下：

```
1. clear
2. h5path = './results/mulliken.h5';
```

```
3. load './results/mulliken.mat';  
4. mul = loadDistArray(h5path,'LCAO/mullikenPop1');  
5. mul = mul.data;  
6. Aorb = LCAO.orbInfo.Aorb;  
7. Ga = sum(sum(mul(:,Aorb == 1)))  
8. As = sum(sum(mul(:,Aorb == 2)))
```

1.4.2 实空间电荷密度数据处理

(1) 以 GaAs 体系为例，进行自洽计算（见 1.4.1 GaAs 体系）

(2) 实空间电荷密度数据处理

自洽计算后，实空间电荷密度的数据将保存在 **h5** 输出文件中，我通过对数据进行处理可以得到可视化软件 VESTA 所支持的格式。

在 Matlab 界面输入：

```
>> h5disp('./results/GaAs_scf.h5')
```

可以得到与实空间电荷密度相关的信息：

```
Group '/rho'  
  Dataset 'rhoOut'  
    Size: 17576x1  
   MaxSize: 17576x1  
  Datatype: H5T_IEEE_F64LE (double)  
 ChunkSize: []  
  Filters: none  
 FillValue: 0.000000  
 Attributes:  
   'real': 1.000000
```

其中'/rho/rhoOut'为数据的保存的位置，**Size: 17576x1** 这表示实空间电荷密度数据的大小。

在 GaAs 体系的文件目录下存放了处理实空间电荷密度的脚本 **realcharge.m**，**realcharge.m** 脚本参考如下：

```
1. clear all;  
2. %%%%%%%%%%%PARAMETER REGION%%%%%%%%  
3. output_name = './results/GaAs_scf'; %RESCU 输出文件名，不带后缀名  
4. %%%%%%%%%%%
```



```
5. matfile = [output_name, '.mat']; load (matfile);
6. h5file = [output_name, '.h5'];
7. chg = loadDistArray(h5file, '/rho/rhoOut');
8. filename = {'CHG.vasp', 'CHG_down.vasp'};
9. for j = 1:size(chg.data, 2);
10. %%%%%%%%%%下面部分 VASP CONTCAR 形式
11. fid = fopen(filename{j}, 'w');
12. fprintf(fid, 'CHG \n'); fprintf(fid, ' %6.8f\n', 1.0000000);
13. for i = 1:3
14. a = domain.latvec(i, :) * 0.529177;
15. fprintf(fid, ' %6.10f %6.10f %6.10f\n', a);
16. end
17. atom_species = size(element, 2);
18. number_of_atoms = size(atom.xyz, 1);
19. each_atom = hist(atom.element, unique(atom.element));
20. for i = 1:atom_species
21. fprintf(fid, '%s ', element(i).species);
22. end
23. fprintf(fid, '\n');
24. for i = 1:atom_species
25. fprintf(fid, '%6.0f ', each_atom(i));
26. end
27. fprintf(fid, '\n'); fprintf(fid, 'Cartesian\n');
28. for i = 1:number_of_atoms
29. b = atom.xyz(i, :) * 0.529177;
30. fprintf(fid, ' %6.9f %6.9f %6.9f\n', b);
31. end
32. fprintf(fid, '\n');
33. %%%%%%%%%%写入实空间格点数
34. realgrid = domain.fgridn;
35. fprintf(fid, '%6.0f %6.0f %6.0f\n', realgrid);
36. %%%%%%%%%%将实空间电荷密度写入，10 个数据一行
37. chgxyz = chg.data(:, j);
38. fprintf(fid, '%f %f %f %f %f %f %f %f %f %f\n', chgxyz);
39. fclose(fid);
40. end
```

(3) 运行 realcharge.m 脚本，在 Matlab 中输入命令：

```
>>run realcharge.m
```

运行结束后，可以得到 **CHG.vasp** 文件，如果体系打开了自旋则可以得到 **CHG.vasp** 和 **CHG_down.vasp**。将 **CHG.vasp** 导入到 VESTA 中，可以得到 GaAs 体系实空间电荷密度分布：

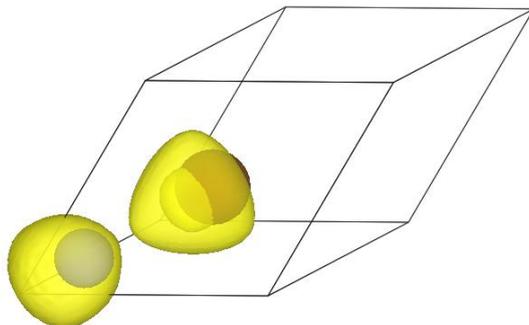


图 1-12 GaAs 体系实空间电荷密度分布

同时 RESCU2020A 用户还可以通过 DeviceStudio2020A 中的 Analysis plot 显示 Density，具体操作如下：Simulator→RESCU→Analysis plot。

1.5 HSE06

在 DFT 计算中，使用 LDA/GGA 泛函往往会低估半导体的带隙，为了能够更精确的计算半导体的带隙，本节中将介绍如何使用 HSE06 杂化泛函进行自洽、能带及态密度计算。

1.5.1 HSE06 自洽计算

(1) 准备输入文件 **scf.input**，**GaAs.xyz** 和基组文件 **As_sp_oncv_PBE_SZP.mat**，**Ga_sp_oncv_PBE_SZP.mat**
scf.input 文件参考如下：

```

info.calculationType = 'self-consistent'
info.savepath        = './results/HSE_scf'
domain.latvec        = [[0 2.8266 2.8266]; [2.8266 0 2.8266]; [2.8266 2.8266 0]]
element(1).species   = 'Ga'
element(1).path      = './Ga_sp_oncv_PBE_SZP.mat'
element(2).species   = 'As'
element(2).path      = './As_sp_oncv_PBE_SZP.mat'
atom.xyz             = './GaAs.xyz'
kpoint.gridn         = [5,5,5]
domain.lowres        = 0.3
LCAO.status          = 1
functional.list       = {'XC_HYB_GGA_XC_HSE06'};
functional.libxc      = 1
Exx.Gcutoff          = 4.0
Exx.qgridn           = [5 5 5]
Exx.dmDistCut        = 30
Exx.highGcorr        = 1
mixing.type           = 'density-matrix'
units.length         = 'Angstrom'
mpi.status            = 1
smi.status            = 1

```

GaAs.xyz 文件参考如下:

```

2
AtomType  X Y Z
As        2.11995000    2.11995000    2.11995000
Ga        0.70665000    0.70665000    0.70665000

```

Notes: (1) 为了更准确的计算 GaAs 体系的带隙, 需将最外层和次外层的电子都包含进了新的基组中。(2) HSE06 计算需要 Libxc 泛函库中的 'XC_HYB_GGA_XC_HSE06' 泛函, 因此需要打开 'functional.libxc' 的开关。(3) 'Exx.Gcutoff' 表示用波函数展开库伦相互项的截断能, 'Exx.Gcutoff' 不应该小于 3Hartree, 'Exx.Gcutoff' 越大越精确, 并且计算量越大。(4) 'Exx.highGcorr' 对能量大于截断能的电子互斥积分进行修正, 该修正可以减少计算量。(5) 'Exx.dmDistCut' 密度矩阵的截断半径, 默认值为 25, 根据自己的体系可以将该参数设置为 5-15。(6) 'Exx.qgridn' 引入一套 Q 网格用来处理 Hartree-Fock 交换项, 设置较小的 Q 网格可以减少计算量, 默认值与 K 网格一致, 实际案例中可以设置比 K 网格小的 Q 网格来减少计算量。(7) 由于 HSE06 计算需要 Libxc 泛函库,

且 windows 安装泛函库很麻烦, 因此本例在服务器上进行 HSE06 计算, 并在输入文件中开启并行开关 **mpi.status= 1**, **smi.status= 1**。

(2) 自洽计算

在服务器上进入 GaAs 体系的文件目录下, 使用任务提交脚本进行并行计算。

1.5.2 HSE06 能带计算

(1) 准备输入文件 band.input

band.input 文件参考如下:

```
info.calculationType = 'band-structure'  
info.savepath        = './results/HSE_bs'  
rho.in               = './results/HSE_scf'  
Exx.VFinput         = './results/HSE_scf'  
kpoint.symptoms     = {'W', 'L', 'G', 'X', 'W', 'K'}  
kpoint.gridn        = 151  
mpi.status           = 1  
smi.status           = 1
```

Notes: 在 HSE06 能带态密度计算时需要 exact exchange 矩阵的信息, 因此需要设置'Exx.VFinput' 这个参数。

(2) 能带计算

在服务器上进入 GaAs 体系的文件目录下, 使用任务提交脚本进行并行计算。

计算结束后, 将数据传到本地进行画图, 如图所示:

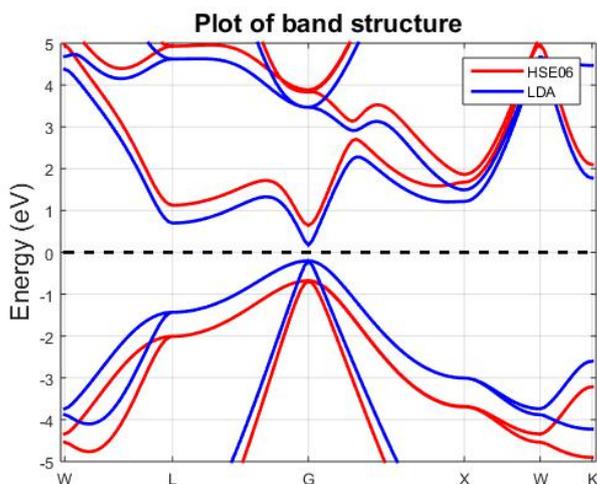


图 1-13 GaAs 体系 LDA 和 HSE06 能带对比图

可以看到 LDA 计算的带隙约为 0.4eV 左右, 使用 HSE06 计算得到的带隙扩大到 1.32eV。

1.5.3 HSE06 态密度计算

(1) 准备输入文件 dos.input

dos.input 文件参考如下:

```
info.calculationType = 'dos'  
info.savepath        = './results/HSE_dos'  
rho.in               = './results/HSE_scf'  
Exx.VFinput         = './results/HSE_scf'  
kpoint.sampling      = 'gauss'  
kpoint.gridn         = [20,20,20]  
smearing.sigma       = 0.05;  
dos.range             = [-3 , 3]  
dos.resolution        = 0.01  
units.energy          = 'eV'  
mpi.status            = 1  
smi.status            = 1
```

Notes: 在 HSE06 态密度计算时同样能够计算体系投影态密度和局域态密度。

(2) 能带计算

在服务器上进入 GaAs 体系的文件目录下, 使用任务提交脚本进行并行计算。计算结束后, 将数据传到本地进行画图, 如图所示:

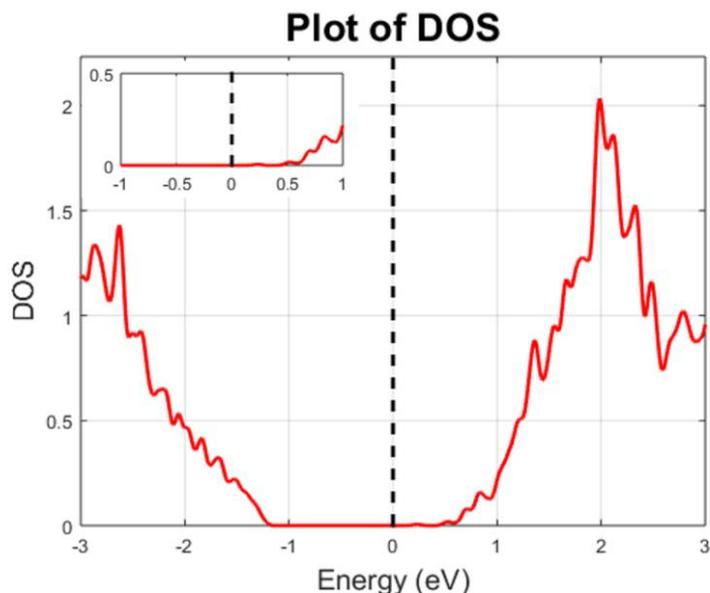


图 1-14 GaAs 体系 HSE06 态密度图

参数常用设置及经验分享:

参数	含义	设置及建议
Exx.Gcutoff	波函数展开库伦项的截断能	≥ 3 Hartree
Exx.highGcorr	能量大于截断能的电子互斥积分进行修正	默认开, 建议打开
Exx.dmDistCut	密度矩阵的截断半径	根据自己的体系设置为 5-15
Exx.qgridn	处理 Hartree-Fock 交换项的网格	可以设置的比 K 网格小, 减小计算量
mixing.type	控制迭代的参数	HSE 需使用 density-matrix
mpi.status	mpi 并行开关	默认关, 并行时需打开
smi.status	smi 并行开关	默认关, 并行时需打开
Exx.VFinput	读取 exact exchange 矩阵	HSE 能带, 态密度计算开启

1.6 DFT+U 计算

本节将介绍如何使用 DFT+U 的方法处理具有局域电子的强关联体系。

1.6.1 DFT+U 自洽计算

在使用 DFT+U 的方法处理强关联体系时往往会使得体系收敛比较困难, 这里将以反铁磁材料 NiO 为例介绍两种 DFT+U 自洽的方法。

方法一: 直接自洽法

(1) 准备输入文件 scf_u.input, NiO.xyz 和基组文件 Ni_DZP.mat, O_DZP.mat
 scf.input 文件参考如下:

```
info.calculationType = 'self-consistent'  
info.savepath        = 'results/NiO_scf_u'  
domain.latvec       = [4.168 2.084 2.084;2.084 4.168 2.084;2.084 2.084 4.168];  
domain.lowres       = 0.3  
kpoint.gridn        = [7,7,7];  
atom.xyz            = 'NiO.xyz'  
functional.includeU = 1  
element(1).species  = 'Ni'  
element(1).path     = './Ni_DZP.mat'  
element(1).CoulombEnergyUd = 4  
element(2).species  = 'O'  
element(2).path     = './O_DZP.mat'  
spin.type           = 'collinear'  
atom.magmom         = './NiO.xyz';  
LCAO.status         = 1  
units.length        = 'Angstrom'  
units.energy        = 'eV'
```

NiO.xyz 文件参考如下:

```
4  
AtomType X Y Z mag  
Ni 1.04210000 1.04210000 1.04210000 1  
Ni 5.21050000 5.21050000 5.21050000 -1  
O 3.12630000 3.12630000 3.12630000 0.0  
O 7.29470000 7.29470000 7.29470000 0.0
```

Notes: (1) 进行 DFT+U 计算时需要开启控制开关'functional.includeU'。(2) U 值大小通过'element(?).CoulombEnergyUs' , 'element(?).CoulombEnergyUp', 'element(?).CoulombEnergyUd'控制,在本例中需要在 Ni 原子的 d 轨道上加上 4eV 的 U 值。因此我们设置 element(?).CoulombEnergyUd=4(eV 由'units.energy'控制);

(3) 因为 NiO 是反铁磁材料, 所以需要一个至少含有两个 Ni 原子的单胞来设置 Ni 的±磁矩。

(2) 自洽计算

打开 Matlab 软件, 进入 NiO 体系的 way1 文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i scf_u.input
```

方法二: 调用普通自洽结果计算法

在某些情况下，使用直接自洽法进行自洽计算会出现体系较难收敛的情况。因此这里介绍另外一种更加稳定的计算方法：首先，进行普通自洽计算，并将密度矩阵保存在每个 **K** 点下；之后，调用普通自洽的结果进行 DFT+U 自洽计算。这样计算往往会比直接计算容易收敛。

(1) 准备普通自洽计算输入文件 `scf_1.input`, `NiO.xyz` 和基组文件 `Ni_DZP.mat`, `O_DZP.mat`。

`scf_1.input` 文件参考如下：

```
info.calculationType = 'self-consistent'  
info.savepath        = 'results/NiO_scf_1'  
domain.latvec        = [4.168 2.084 2.084;2.084 4.168 2.084;2.084 2.084 4.168];  
domain.lowres        = 0.3  
kpoint.gridn         = [7,7,7];  
atom.xyz              = 'NiO.xyz'  
element(1).species   = 'Ni'  
element(1).path      = './Ni_DZP.mat'  
element(2).species   = 'O'  
element(2).path      = './O_DZP.mat'  
spin.type             = 'collinear'  
atom.magmom           = './NiO.xyz';  
LCAO.status           = 1  
units.length          = 'Angstrom'  
units.energy          = 'eV'  
symmetry.spacesymmetry = 0  
symmetry.pointsymmetry = 0  
symmetry.timereversal = 0  
option.saveDensityMatrixK = 1
```

`NiO.xyz` 文件参考如下：

```
4  
AtomType X Y Z mag  
Ni 1.04210000 1.04210000 1.04210000 1  
Ni 5.21050000 5.21050000 5.21050000 -1  
O 3.12630000 3.12630000 3.12630000 0.0  
O 7.29470000 7.29470000 7.29470000 0.0
```

Notes: (1) 在不同自洽下，需要将密度矩阵保存在每个 **K** 点下，因此要开启 'option.saveDensityMatrixK' (2) 为了将密度矩阵保存在每个 **K** 点下，必须将所

有对称性关闭。即关闭 'symmetry.spacesymmetry', 'symmetry.pointsymmetry', 'symmetry.timereversal'。

(2) 普通自洽计算

打开 Matlab 软件, 进入 NiO 体系的 way2 文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i scf_1.input
```

计算结束后, 调用普通自洽结果进行 DFT+U 自洽计算。

(3) DFT+U 自洽计算

scf_u.input 文件参考如下:

```
info.calculationType = 'self-consistent'  
info.savepath = 'results/NiO_scf_u'  
domain.latvec = [4.168 2.084 2.084;2.084 4.168 2.084;2.084 2.084 4.168];  
domain.lowres = 0.3  
kpoint.gridn = [7,7,7];  
atom.xyz = 'NiO.xyz'  
functional.includeU = 1  
element(1).species = 'Ni'  
element(1).path = './Ni_DZP.mat'  
element(1).CoulombEnergyUd = 4  
element(2).species = 'O'  
element(2).path = './O_DZP.mat'  
spin.type = 'collinear'  
atom.magmom = './NiO.xyz';  
LCAO.status = 1  
units.length = 'Angstrom'  
units.energy = 'eV'  
rho.in = 'results/NiO_scf_1'  
LCAO.DMkcell = rho.in
```

Notes: (1) 读取上次计算的密度矩阵需要设置以下两个参数: 'rho.in'和 'LCAO.DMkcell'。 (2) 在第二次计算中打开+U 的开关, 同时在 Ni 的 d 轨道上设置 4eV 的 U 值。

打开 Matlab 软件, 进入 NiO 体系的 way2 文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i scf_u.input
```

1.6.2 DFT+U 能带计算

(1) 准备输入文件 band.input

band.input 文件参考如下:

```
info.savepath = 'results/NiO_bs_u'  
info.calculationType = 'band-structure'  
rho.in = 'results/NiO_scf_u'  
kpoint.gridn =250
```

(2) 能带计算

打开 Matlab 软件, 进入 NiO 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i band_u.input
```

NiO+U 的能带如图所示:

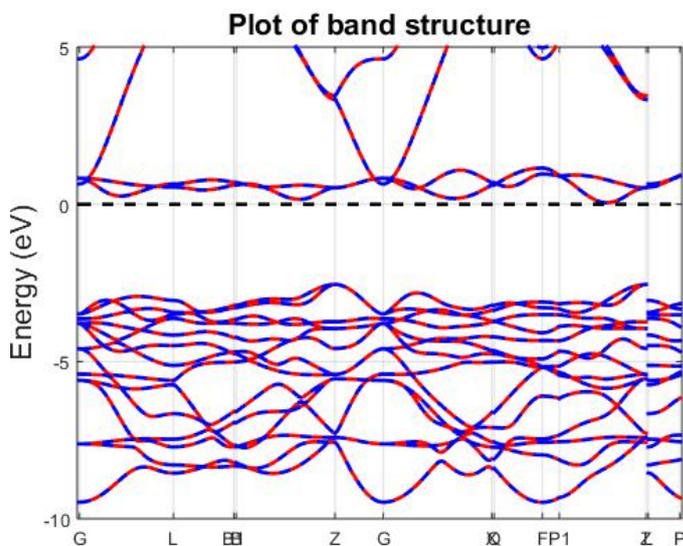


图 1-15 NiO 体系 DFT+U 能带图

对于 NiO 体系来说, 不加 U 值的带隙为 0.4eV 左右, 可以看到 U=4eV 时带隙变大了很多; 为了得到带隙的具体值, 可从 NiO_bs_u.mat 中读取带隙。

在 Matlab 中输入命令:

```
>> load NiO_bs_u.mat  
>> band.bandGap
```

在 Matlab 界面得到如下信息: **bandGap = 2.60eV**

1.6.3 DFT+U 态密度计算

(1) 准备输入文件 dos_u.input

dos_u.input 文件参考如下:

```

info.savepath = 'results/NiO_dos_u'
info.calculationType = 'dos'
kpoint.gridn = [17,17,17];
kpoint.sampling = 'tetrahedron+blochl'
rho.in = 'results/NiO_scf_u'
dos.range = [-8 , 5]
dos.resolution = 0.03
dos.projL = [1,2]
dos.projM = [-2,-1,0,1,2]
dos.projAtom = [1,3]
units.energy='eV'
  
```

(2) 态密度计算

打开 Matlab 软件, 进入 NiO 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i dos_u.input
```

态密度计算结束后, 对数据进行简单的处理可以得到 Ni 的 d 轨道和 O 的 p 轨道的数据, 如图所示:

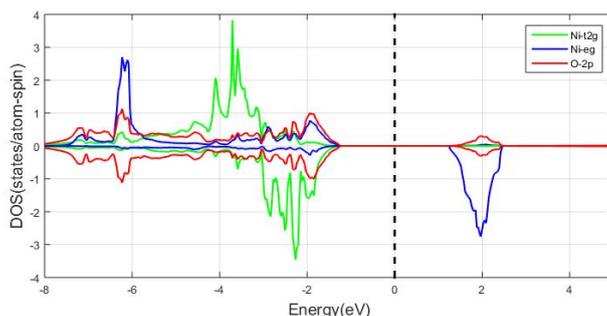


图 1-16 NiO 体系 DFT+U 态密度图

参数常用设置及经验分享:

参数	含义	设置及建议
functional.includeU	DFT+U 开关	/
element(?).CoulombEnergyU	U 值设置	按经验和文献设置

LCAO.DMkcell	K 点密度矩阵的路径	默认和 rho.in 相同
symmetry.spacesymmetry	空间群对称性	默认关
symmetry.pointsymmetry	点群对称性	默认开
symmetry.timereversal	时间反演对称性	默认开
option.saveDensityMatrixK	保存每个 K 点密度矩阵	/

1.7 结构弛豫计算

本节将介绍如何使用 RESCU 进行结构弛豫计算，目前 RESCU 只能进行原子位置弛豫，还无法进行晶格常数弛豫。因此可以通过以下两种方法来得到合适的晶格常数：1.状态方程拟合晶格常数；2.直接从相关文献中读取。

1.7.1 Si 单胞的结构弛豫计算

以 Si 体系为例：

(1) 准备输入文件 relax.input, Si.xyz 和基组文件 Si_DZP.mat

relax.input 文件参考如下：

```

info.calculationType = 'relaxation'
info.savepath        = './results/Si_relax'
domain.latvec        = [[0 2.71535 2.71535]; [2.71535 0 2.71535];
[2.71535 2.71535 0]]
element(1).species   = 'Si'
element(1).path      = './Si_DZP.mat'
atom.xyz             = './Si.xyz'
kpoint.gridn         = [5,5,5]
domain.lowres        = 0.5
units.length         = 'Angstrom'
sr.tol               = 1e-3
sr.method            = 'cg-fr'
    
```

Notes: sr.tol 表示原子受力的收敛精度，sr.method 表示 RESCU 中结构弛豫的方法，主要有三类：'qn'准牛顿方法，'cg'共轭梯度法和'fire'阻尼分子动力学法。

Si.xyz 文件参考如下：

```
2
AtomType X Y Z
Si 0.67883750 0.67883750 0.67883750
Si 2.03631139 2.00000000 2.03631139
```

Notes: 我们将第二个 Si 原子的 Y 坐标从 2.03600964 改为了 2.00000000。

(2) 结构弛豫计算

打开 Matlab 软件, 进入 Si 体系的文件目录下, 该目录需包含上面准备的 3 个输入文件 relax.input, Si.xyz 和 Si_DZP.mat。在 Matlab 中输入命令:

```
>> rescu -i relax.input
```

计算结束, 可以得到输出日志文件 **resculog.out**, 弛豫后的原子结构文件 **relaxedstructure.xyz** 和弛豫过程日志文件 **relaxationprocess.txt**。

(3) 结构弛豫原子位置处理

结构弛豫完成之后, 直接把 relaxedstructure.xyz 拖动到建模软件 DeviceStudio2020A 中, 即可显示 relaxedstructure.xyz。观察 relaxedstructure.hzw 文件可以发现, 第二个原子的 Y 坐标已经变回了移动前的坐标了。

(4) 约束结构弛豫设置

对于上述两个 Si 原子的单胞, 如果只需要弛豫第二个原子的 Y 方向, 此时需要怎么设置呢?

这里需要引入一个新的参数'sr.moveableAtomList', 其意义为设置可动的原子, 将其设置为 sr.moveableAtomList = './Si.xyz', 意为从 Si.xyz 中去读取可动原子信息。

同时 Si.xyz 文件参考如下:

```
2
AtomType X Y Z
Si 0.67883750 0.67883750 0.67883750 0 0 0 0 0 0
Si 2.03651250 2.00000000 2.03651250 0 0 0 0 1 0
```

上述 xyz 文件比之前的 xyz 文件多了 6 列 (第 5-10 列) 数据, xyz 文件中的第 8-10 列为弛豫时三个方向的控制参数, 其中 0 为固定 1 为弛豫; 对于没有磁性的材料, 我们需要将 5-7 列补为 0。设置好 Si.xyz 文件和'sr.moveableAtomList'之后即可进行约束结构弛豫了。

参数常用设置及经验分享:

参数	含义	设置及建议
sr.tol	原子受力的收敛精度	一般设置为 1e-2
sr.method	结构驰豫的方法	/
sr.moveableAtomList	驰豫某些原子设置	/

1.8 密度泛函微扰理论

本节将介绍如何使用 RESCU 中的 DFPT 模块进行计算，目前 RESCU 能够使用 DFPT 模块进行介电张量、波恩有效电荷、动力学矩阵、声子能带、声子态密度及低频长波的光学性质。其中声子和光学性质计算需要用到离子钳位介电张量、波恩有效电荷、动力学矩阵的信息，因此计算声子和光学性质的时候需要先计算离子钳位介电张量、波恩有效电荷、动力学矩阵这三个物理量。

1.8.1 准备基态的电荷密度和波函数

使用 RESCU 的 DFPT 进行计算需要先计算基态的电荷密度和波函数，下面我们以 BN 体系为例：

- (1) 准备输入文件 scf.input, BN.xyz 和基组文件 N_DZP.mat, B_DZP.mat
scf.input 文件参考如下：



```
info.calculationType      = 'self-consistent'
info.savepath             = './results/BN_scf'

%symmetry.spacesymmetry  = 0
domain.latvec            = [[0 1.8075 1.8075]; [1.8075 0 1.8075]; [1.8075
1.8075 0]]

%eigensolver.emptyBand   = 16
eigensolver.maxit        = 15
eigensolver.algo         = 'cfsi'
LCAO.status              = 0

smearing.sigma           = 0    % advised for dfpt
units.energy             = 'eV'
kpoint.gridn             = [3,3,3]

domain.lowres            = 0.3
%domain.highres          = 0.3
functional.libxc         = 1
functional.list           = {'XC_LDA_X','XC_LDA_C_PW'}

option.maxSCFiteration    = 200
mixing.type               = 'density'
mixing.method             = 'pulay'
mixing.tol                = [1e-07,1e-07] % advised for dfpt
mixing.maxhistory        = 20

spin.type                 = 'degenerate'

element(1).species       = 'N'
element(1).path          = './N_DZP.mat'
element(2).species       = 'B'
element(2).path          = './B_DZP.mat'
atom.xyz                  = 'Atom.xyz'
units.length             = 'Angstrom'

option.saveWavefunction  = 1      % required for dfpt
diffop.method            = 'fft'  % required for dfpt
domain.fourierInit       = false  % required for dfpt
symmetry.spacesymmetry   = 1      % required for dfpt
symmetry.pointsymmetry   = 1      % required for dfpt
symmetry.timereversal    = 1      % required for dfpt
```

BN.xyz 文件参考如下:

2				
AtomType	X	Y	Z	
N	1.35562500	1.35562500	1.35562500	
B	0.45187500	0.45187500	0.45187500	

Notes: (1) 由于 DFPT 继续需要使用 Libxc 泛函库, 且 windows 安装泛函库很麻烦, 本例和 HSE06 的例子一样在服务器上进行计算; (2) kpoint.grin 将用于计算动力学矩阵, 必须使用基数的网格; (3) sigma 必须设置为 0; (4) 设置足够小的收敛精度; (5) diffop.method 必须设置为 'fft'; (6) domain.fourierInit 设置为 'false'; (7) 保存波函数 option.saveWavefunction=1。

(2) 自洽计算

在服务器上进入 BN 体系的文件目录下, 使用任务提交脚本计算 scf.input 文件。计算完成之后电荷密度和波函数将保存在 BN_scf.mat 和 BN_scf.h5 文件中。

1.8.2 计算离子钳位介电张量

在 DFPT 的框架下, 离子钳位介电张量是电子对长波外部电场的响应。因此需要给定一个微扰电场。离子钳位介电张量的计算需要使用到普通自洽的波函数和电荷密度, 因此需要使用 1.8.1 的计算结果。

(1) 准备输入文件 dielectric.input

dielectric.input 文件参考如下:

```

info.calculationType = 'dfpt-dielectric';
info.savepath        = './results/BN_dielectric'
rho.in               = './results/BN_scf.mat'
psi                  = './results/BN_scf.h5'
mixing.tol           = [1e-07,1e-07]; % advised for dfpt
dfpt.method          = 'sternheimer';
%option.saveWavefunction = 1
    
```

Notes: (1) psi 表示波函数导入的参数; (2) 这里的收敛精度微扰自洽中的电荷密度和势函数, 建议尽量取小。(3) 当'dfpt.method'设置为 cfsi 时, 将采用切比雪夫滤波技术, 目前版本中只支持 K=[1 1 1]的大体系计算, 同时用户还需要设置 'eigensolver.maxit=15', 'eigensolver.maxRestart = 1'和' eigensolver.tol= 1e-4' 。

(2) 离子钳位介电张量计算

在服务器上进入 BN 体系的文件目录下，使用任务提交脚本计算 dielectric.input 文件。计算完成之后在 results 下得到 BN_dielectric.mat。离子钳位介电张量保存在 dfpt.dielectricTensor 中。

1.8.3 计算动力学矩阵和波恩有效电荷

本节我们将介绍如何计算动力学矩阵和波恩有效电荷，动力学矩阵可用于计算声子谱及光学性质，对于声子谱计算我们需要计算倒空间中不同 K 点下的动力学矩阵，而光学性质的计算只需要 G 点的动力学矩阵。这里以声子计算所需的动力学为例：

(1) 准备输入文件 phonon.input

phonon.input 文件参考如下：

```
info.calculationType = 'dfpt-phonon';
info.savepath        = './results/BN_phonon'
rho.in              = './results/BN_scf.mat'
psi                 = './results/BN_scf.h5'
mixing.tol          = [1e-07,1e-07]; % advised for dfpt
%dfpt.qpointGridn   = [1 1 1];
dfpt.method         = 'sternheimer';
%option.saveWavefunction = 1
```

Notes: (1) 对于声子谱计算我们需要计算倒空间中不同 K 点下的动力学矩阵且此时的 K 点必须和基态自洽计算的 K 点一致因此我们这边省略了对 K 点的设置。(2) 这里'dfpt.method'需要与介电张量计算的设置一致。

(2) 动力学矩阵和波恩有效电荷计算

在服务器上进入 BN 体系的文件目录下，使用任务提交脚本计算 phonon.input 文件。计算完成之后在 results 下得到 BN_phonon.mat。动力学矩阵保存在 dfpt.dynMat 中其大小是 $3nat*3nat*nq$ 的矩阵，其中 nat 是计算原胞内的原子数目，nq 是倒空间的 K 点数目；波恩有效电荷保存在 dfpt.BornEC 中其大小是 $3*3*nat$ 。

1.8.4 计算声子能带

计算完了离子钳位介电张量、动力学矩阵、波恩有效电荷之后我们就可以通过以上三个参数来进行 DFPT 的方法计算声子能带及态密度了。

(1) 准备输入文件 phonon_bs.input

phonon_bs.input 文件参考如下:

```
info.calculationType = 'dfpt-phonon-bs'  
info.savepath        = './results/BN_phonon_bs'  
dfpt.phononData     = './results/BN_phonon.mat'  
dfpt.dielectricData = './results/BN_dielectric.mat'  
dfpt.qpointGridn    = 96  
dfpt.qpointSymptoms = {'L','G','X','W'}
```

Notes: (1) dfpt.dielectricData 为导入离子钳位介电张量; (2) dfpt.phononData 为导入动力学矩阵和波恩有效电荷; (3) dfpt.qpointSymptoms 为声子能带的高对称点输入, 程序自动识别字母表示和笛卡尔坐标; (4) dfpt.qpointGridn 为声子能带路径, 可以是一个数或者两个高对称点之间的间隔。

(2) 声子能带计算

在服务器上进入 BN 体系的文件目录下, 使用任务提交脚本计算 phonon_bs.input 文件。计算完成之后在 results 下得到 BN_phonon_bs.mat。将该文件导入到本地使用 Matlab, 输入以下命令:

```
>> rescu -p ./results/BN_phonon_bs.mat
```

得到声子能带结构图如下:

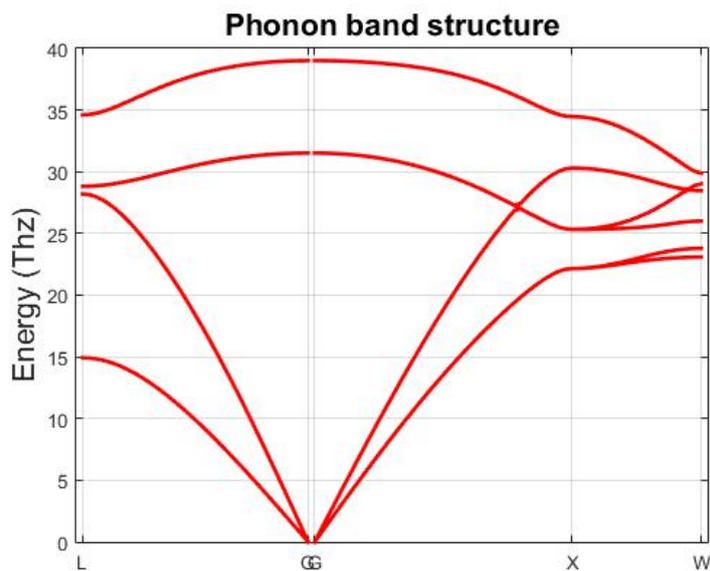


图 1-17 BN 的声子能带图

1.8.5 计算声子态密度

计算完了离子赝位介电张量、动力学矩阵、波恩有效电荷之后我们就可以通过以上三个参数来进行 DFPT 的方法计算声子能带及态密度了。

(1) 准备输入文件 `phonon_dos.input`

`phonon_dos.input` 文件参考如下:

```
info.calculationType      = 'dfpt-phonon-dos'  
info.savepath             = './results/BN_phonon_dos'  
dfpt.phononData          = './results/BN_phonon.mat'  
dfpt.dielectricData      = './results/BN_dielectric.mat'  
dfpt.qpointGridn        = [9 9 9]
```

Notes: (1) DFPT 计算声子态密度和声子能带一样都需要设置 `dfpt.dielectricData` 和 `dfpt.phononData` 参数; (2) DFPT 声子态密度计算和普通态密度计算一样都需要设置较密的 K 空间网格即 `dfpt.qpointGridn` 参数。

(2) 声子态密度计算

在服务器上进入 BN 体系的文件目录下,使用任务提交脚本计算 `phonon_dos.input` 文件。计算完成之后在 `results` 下得到 `BN_phonon_dos.mat`。将该文件导入到本地使用 Matlab,输入以下命令:

```
>> rescu -p ./results/BN_phonon_dos.mat
```

得到声子态密度结构图如下:

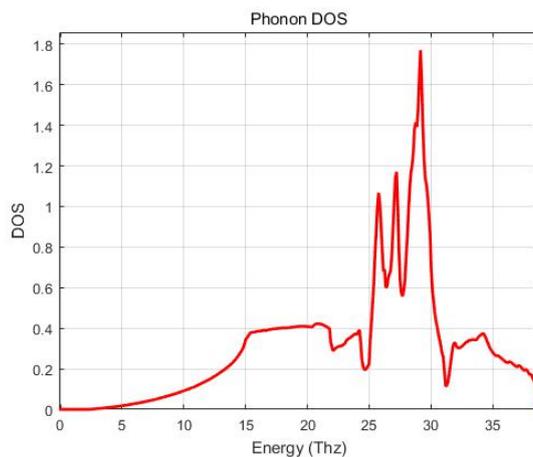


图 1-18 BN 的声子态密度图

1.8.6 计算低频长波的光学性质

目前 RESCU 在 DFPT 框架下能够计算的光学性质包括介电常数函数的实部和虚部、折射率的实部和虚部、能量损失函数、红外强度，并且这些性质的计算都能在一个文件完成。下面是光学性质计算的细节：

(1) 准备输入文件 optic.input

optic.input 文件参考如下：

```
info.calculationType = 'dfpt-optic'  
info.savepath        = './results/BN_optic'  
rho.in               = './results/BN_scf.mat'  
psi                  = './results/BN_scf.h5'  
dfpt.phononData      = './results/BN_phonon.mat'  
dfpt.dielectricData  = './results/BN_dielectric.mat'  
dfpt.qpointKdirect   = [0 0 1]  
dfpt.opticDamping    = 1e-4
```

Notes: (1) 同样的 DFPT 计算光学性质也都需要设置 `dfpt.dielectricData` 和 `dfpt.phononData` 参数； (2) `dfpt.qpointKdirect` 表示计算光学性质所需给定的入射电场的方向，默认为笛卡尔坐标；方向可以时多个，例如 `dfpt.qpointKdirect = [1 0 1;1 1 1;1 0 0]` (3) `dfpt.opticDamping` 当高于零温是，原子将做阻尼耦合谐振子振动，其系数称之为非零阻尼系数，该阻尼因子能够改变光学性质的展宽。(4) 光学性质计算时只需要 G 点的信息，因此做独立计算时可以将介电张量和波恩有效电荷计算时打开 `dfpt.qpointGridn=[1 1 1]` 的开关。

(2) 光学性质计算

在服务器上进入 BN 体系的文件目录下，使用任务提交脚本计算 `optic.input` 文件。计算完成之后在 `results` 下得到 `BN_optic.mat`。将该文件导入到本地使用 Matlab，输入以下命令：

```
>> rescu -p ./results/BN_optic.mat
```

此时将四幅图片分别为

`BN_plot_diefunc.fig`、介电常数函数的实部和虚部

`BN_plot_refindx.fig`、折射率的实部和虚部

`BN_plot_nrgloss.fig`、能量损失函数

BN_plot_rintns.fig、红外强度四幅图如下：

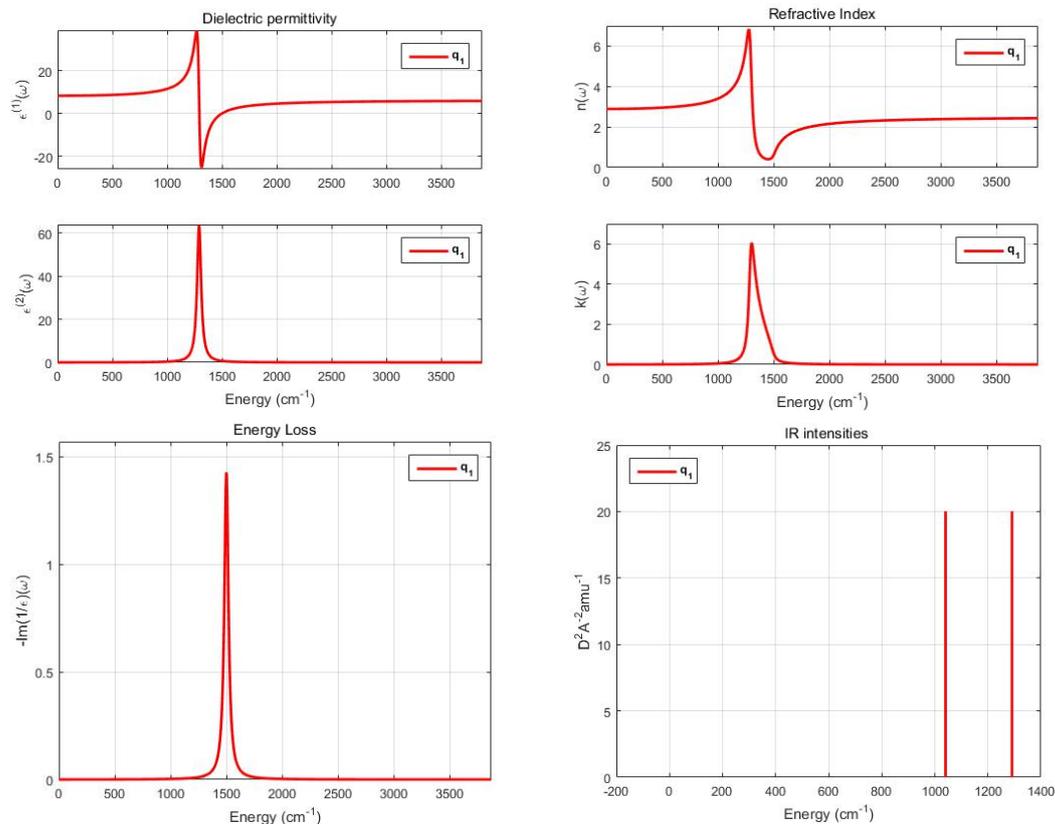


图 1-19 BN 介电常数函数图、折射率图、能量损失函数图、红外强度图

1.8.7 计算拉曼光谱

计算完了离子锚位介电张量、动力学矩阵、波恩有效电荷之后我们就可以通过以上三个参数来进行 DFPT 的方法计算拉曼光谱了。

(1) 准备输入文件 `raman.input`

`raman.input` 文件参考如下：

```
info.calculationType      = 'dfpt-raman'  
info.savepath             = './results/BN_raman'  
rho.in                   = './results/BN_scf.mat'  
psi                       = './results/BN_scf.h5'  
dfpt.phononData          = './results/BN_phonon.mat'  
dfpt.dielectricData      = './results/BN_dielectric.mat'  
dfpt.qpointCartesian     = [1,0,0;1,0,0]  
dfpt.ramanIncPhotonPol   = [0 1 0;0 0 1]  
dfpt.ramanOutPhotonPol   = [0 1 0;0 1 0]  
dfpt.ramanIncPhotonWav   = 2660  
units.length             = 'Angstrom'  
dfpt.ramanTemperature    = 300  
dfpt.opticDamping        = 1e-4
```

Notes: (1) 同样的 DFPT 计算拉曼光谱也都需要设置 `dfpt.dielectricData` 和 `dfpt.phononData` 参数; (2) `dfpt.qpointCartesian` 表示计算拉曼性质所需给定的入射电场的方向, 默认为笛卡尔坐标; (3) `dfpt.ramanIncPhotonPol` 表示笛卡尔坐标下拉曼测量中入射光子的偏振方向; (4) `dfpt.ramanOutcPhotonPol` 表示笛卡尔坐标下拉曼测量中出射光子的偏振方向; (5) `dfpt.ramanIncPhotonWav` 表示拉曼测量的入射光子的波长。如果与一起使用, 可以设置 `units.length` 为 `Angstrom`。**RESCU** 假定波长为 4500 埃 (450 nm) 作为默认值; (6) `dfpt.ramanTemperature` 表示拉曼测量的温度, 以开尔文为单位, 用于根据玻色爱因斯坦统计分布, 默认值为 300 开尔文。

(2) 拉曼光谱性质计算

在服务器上进入 BN 体系的文件目录下, 使用任务提交脚本计算 `raman.input` 文件。计算完成之后在 `results` 下得到 `BN_raman.mat`。将该文件导入到本地使用 Matlab, 输入以下命令:

```
>> rescu -p ./results/BN_raman.mat
```

此时将两幅图片分别为 `BN__Polarized.fig`、`BN_plot_Average.fig`、两幅图如下:

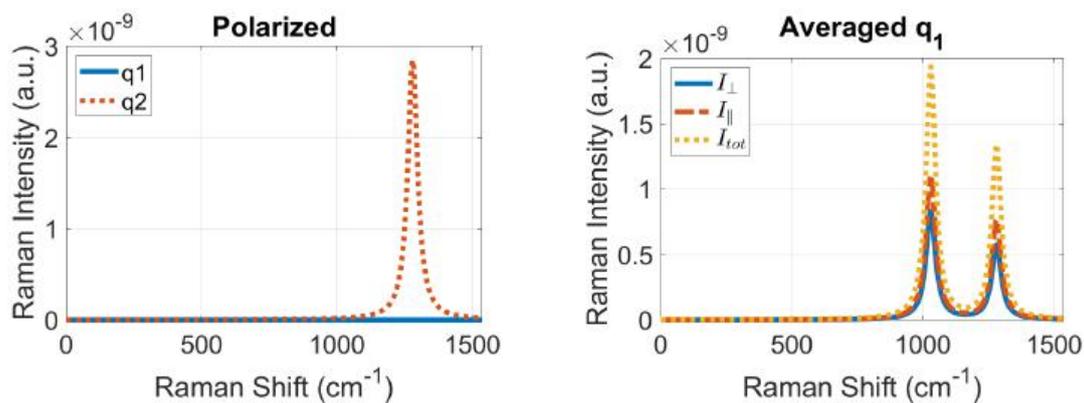


图 1-20 BN 拉曼 Polarized 图、拉曼 Average 图

参数常用设置及经验分享:

参数	含义	设置及建议
mixing.tol	电荷密度和势的精度	
option.saveWavefunction	保存波函数	开关 0/1
diffop.method	结构弛豫的方法	/
domain.fourierInit	弛豫某些原子设置	/
psi	导入波函数	
dfpt.eField	微扰电场	
dfpt.qpointGridn	Q 空间格点数目	
dfpt.phononData	导入 dfpt-phonon 的计算结果	
dfpt.dielectricData	导入 dfpt-dielectric 的计算结果	
dfpt.qpointSympoints	Q 空间的高对称点	
dfpt.qpointKdirect	电场方向	
dfpt.opticDamping	光学性质展宽	
dfpt.qpointCartesian	入射电场的方向	默认为笛卡尔坐标
dfpt.ramanIncPhotonPol	入射光子的偏振方向	笛卡尔坐标下
dfpt.ramanOutcPhotonPol	出射光子的偏振方向	笛卡尔坐标下
dfpt.ramanIncPhotonWav	入射光子的波长。	450 nm

dfpt.ramanTemperature	拉曼测量的温度,	300 开尔文
-----------------------	----------	---------

1.9 有限位移法

本节将介绍如何使用 RESCU 中的 finite displacement 模块进行声子谱计算。声子计算的工作流程如下。首先, RESCU 将该结构扩展到指定的超胞结构。然后确定其对称性, 产生非等效位移并对每个原子结构进行自洽计算。为每个非等效位移生成输入/输出文件, 以便可以并行执行自洽计算。完成后, 输出文件包含每一个原子。最后, RESCU 计算力常数、动力学矩阵和声子能带结构等相关量。在这一节中, 我们以 Au 体系为例:

(1) 准备输入文件 phonon.input, Au.xyz 和基组文件 Au_DZP.mat.

phonon_Au.input 文件参考如下:

```

info.calculationType = 'phonon'
info.savepath        = './results/au_ph'
domain.latvec        = [[0 2.0391 2.0391]; [2.0391 0 2.0391]; [2.0391 2.0391 0]]
element(1).species   = 'Au'
element(1).path      = './Au_DZP.mat'
functional.list       = {'XC_LDA_X','XC_LDA_C_PW'}
atom.xyz              = './Atom.xyz'
kpoint.gridn         = [2,2,2]
domain.lowres        = 0.4
units.length         = 'Angstrom'
ph.mode               = 'phonon'
ph.qpointSympoints   = {{[0 0 0] [0.5 0.5 0]} {[1 0.5 0.5] [3/4 3/8 3/8]} {[3/4 3/8 3/8] [0 0 0]} {[0 0 0] [0.5 0.5 0.5]}}
ph.labels             = {'G','X','X1','K','G','G','L'};
ph.supercellDimension = [2 2 2]
ph.ASR                = 1
ph.dr                 = 0.1
    
```

Au.xyz 参考文件如下:

```

1
AtomType X Y Z
Au 0 0 0
    
```

Notes: (1) ph.mode 声子谱计算方法为 phonon, 当使用 phonon 方法时, RESCU 将自动进行完整的计算, 如果需要分步计算请参考英文手册; (2) ph.labels 为

声子能带的高对称点的标签（字母表示）；（3）`ph.qpointSympoints` 为声子能带的 Q 空间高对称点；（4）`ph.supercellDimension` 决定用于计算原子间力常数的超胞的大小，即原胞的扩胞倍数；（5）`ph.ASR` 为 Acoustic-Sum-Rule (ASR) 校正方案，当 `ph.ASR = 1`，执行 ASR 校正方案。理论上，三维块体系统的声子谱在高对称点 G 有三个零频率，对应声子谱声学支，ASR 校正方案通常可以提高在高对称点 G 的精确度。（6）`ph.dr = 0.1` 决定原子位移大小（波尔单位），经典取值大约 0.1。

（2）声子能带计算

在服务器上进入 Au 体系的文件目录下，使用任务提交脚本计算 `phonon_Au.input` 文件。计算完成之后在 `results` 下得到 `au_ph.mat`，`au_ph.h5`。

打开 Matlab 软件，进入 Au 体系的文件目录下，在 Matlab 中输入命令：

```
>> rescu -p results/au_ph.mat
```

得到声子能带结构图如下：

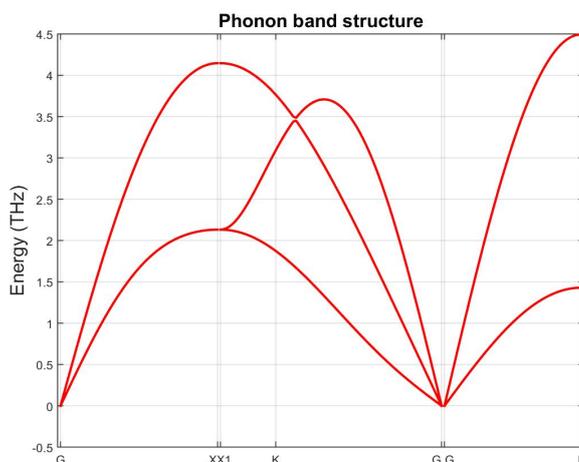


图 1-21 Au 的声子能带图

参数常用设置及经验分享：

参数	含义	设置及建议
<code>info.calculationType</code>	计算内容	<code>phonon</code>
<code>ph.mode</code>	计算声子谱的方法	<code>phonon</code>
<code>ph.dr</code>	决定原子位移大小（波	经典之值 0.1

	尔单位)	
ph.supercellDimension	决定用于计算原子间力常数的超胞的大小	
ph.ASR	切换为 Acoustic-Sum-Rule (ASR) 校正方案	1 或 true
ph.qpointSympoints	Q 空间的高对称点	
ph.labels	高对称点的标签	
ph.qpointGridn	Q 空间格点数目	

1.10 Wannier 函数计算

本节将介绍如何使用 RESCU 中的 MLWF 模块进行 Wannier 函数计算。在自洽计算之后，我们得到基态密度，由此可以计算 Kohn-Sham (KS) 态。KS 状态基通常在实空间是非定域的，在倒空间是非光滑的，这在某些应用中是不实际的。可以从标准 KS 态（或其子集）导出最大局部化基，这是一个称为规范优化的过程，它产生（最大局部 Wannier 函数）MLWF。请注意，MLWF 仅适用于 LCAO 计算(LCAO.statue=1) wannier 函数计算方法有两种，在这一节中，我们以 Si 体系为例：

1.10.1 自洽计算

(1) 准备输入文件 scf.input, Si.xyz 和基组文件 Si_DZP.mat

scf.input 文件参考如下：

```
info.savepath      = './results/si_lcao_scf'  
info.calculationType = 'self-consistent'  
atom.fracxyz      = 'Atom.xyz'  
domain.latvec     = '[[0 2.715 2.715]; [2.715 0 2.715]; [2.715 2.715 0]]'  
domain.lowres     = 0.3  
element(1).species = 'Si'  
element(1).path   = './Si_AtomicData.mat'  
functional.list   = {'XC_LDA_X','XC_LDA_C_PW'}  
mixing.tol        = [1e-7 1e-7]  
kpoint.gridn      = [9 9 9]  
units.length      = 'Angstrom'  
LCAO.status       = 1
```

Notes: (1) MLWF 仅适用于 LCAO 计算(LCAO.statue=1); (2) mixing.tol 这里设置为[1e-7 1e-7], 建议提高收敛标准, 高于普通自洽收敛计算; (3) atom.fracxyz 表示结构原子位置采用分数坐标形式。

Si.xyz 文件参考如下:

```
2  
AtomType  X      Y      Z  
Si        0.3750 0.3750 0.3750  
Si        0.6250 0.6250 0.6250
```

(2) 自洽计算

打开 Matlab 软件, 进入 Si 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i scf.input
```

自洽结束后, 会产生主要的输出文件: Si_scf.mat 和 Si_scf.h5。

1.10.2 wannier 函数计算

wannier 函数的计算需要使用到普通自洽的波函数和电荷密度, 因此需要使用 1.9.1 的计算结果。下面介绍两种 wannier 函数计算的方法。

方法一: wannier 函数计算

(1) 准备输入文件 si_lcao_wan_wino.input.

si_lcao_wan_wino.inputt 文件参考如下:

```
info.savepath      = './results/si_lcao_wan_wino'  
info.calculationType = 'wannier'  
rho.in            = './results/si_lcao_scf'  
kpoint.gridn     = [9 9 9]  
units.energy      = 'ev'  
wannier.shiftEf   = true  
wannier.outwin    = [-15 15]  
wannier.innwin    = [-15 0]  
wannier.orbi      = [1 3:5 14 16:18]  
wannier.maxit     = 250
```

Notes: (1) wannier.shiftE 设置为 true, 则 **wannier.outwin** 能量窗口是相对费米能级的; (2) wannier.outwin 指定外部能量窗口; (3) wannier.innwin 指定内部能量窗口; (4) wannier.maxit 设置最大迭代次数 (在子空间平滑过程和扩展函数优化过程使用); (5) 能量单位默认是 Hartree. 这里能量单位指定为 eV, 即 units.energy = 'ev'; (6) wannier.orbi 是一个轨道指数向量, 它决定使用哪个轨道作为投影基。它是 Wannier 函数计算中最重要的关键字。首先, 投影基决定了 MLWF 基的维数: 每个投影轨道都有一个 MLWF。因此, Wannier 哈密顿量和运算符也有这个维度。其次, 投影基为 Wannier 函数提供了一个初始的排序猜测。因此, 它应该根据目标能量窗中的带的特性来选择, 例如可以通过对称性、物理直觉来确定或根据 LCAO.orbInfo 给的信息确定。

(2) wannier 函数计算

打开 Matlab 软件, 进入 Si 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i scf.input
```

会产生主要的输出文件: si_lcao_wan_wino.input.mat 和 si_lcao_wan_wino.input.h5。

方法一: wannier 函数能带计算

(1) 准备输入文件 si_lcao_wbs_wino.input.

si_lcao_wbs_wino.input 文件参考如下:

```
rho.in            = 'results/si_lcao_wan_wino'  
info.savepath     = './results/si_lcao_wbs_wino'  
info.calculationType = 'wannier-bs'
```

Notes: (1) info.calculationType 设置为 wannier-bs; (2) 调用 wannier 函数计算方法一的计算结果。

(2) wannier 函数能带计算

打开 Matlab 软件，进入 Si 体系的文件目录下，在 Matlab 中输入命令：

```
>> rescu -i si_lcao_wbs_wino.input
```

会产生主要的输出文件：si_lcao_wbs_wino.input.mat 和 si_lcao_wbs_wino.input.h5。

```
>> rescu -p results/si_lcao_wbs_wino.input.mat
```

得到 wannier 函数能带结构图如下：

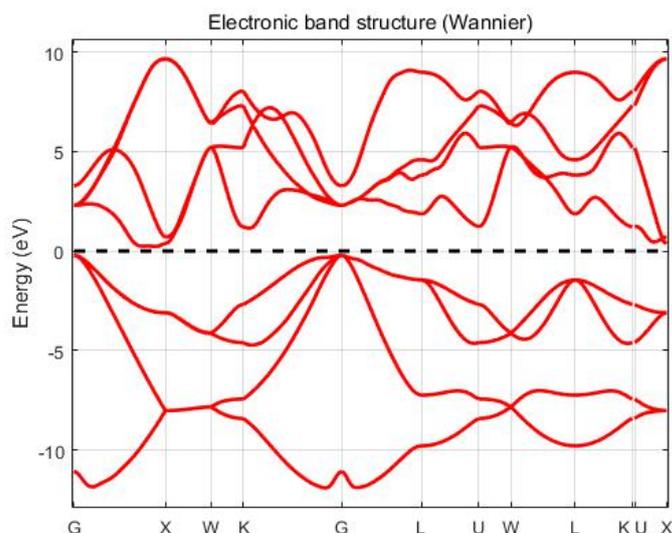


图 1-19 Si 的 wannier 能带图

方法二：wannier 函数计算

(1) 准备输入文件 si_lcao_wan_bndi.input.

si_lcao_wan_bndi.inpuutt 文件参考如下：

```
info.savepath      = './results/si_lcao_wan_bndi'  
info.calculationType = 'wannier'  
rho.in            = './results/si_lcao_scf'  
kpoint.gridn      = [9 9 9]  
wannier.band1     = [1,4]  
wannier.orbi      = [1 3 4 5]  
wannier.maxit     = 250
```

Notes: (1) wannier.band1 指定计算那条能带； (2) wannier.orbi 指定那些轨道作为投影基。

(2) wannier 函数计算

打开 Matlab 软件，进入 Si 体系的文件目录下，在 Matlab 中输入命令：

```
>> rescu -i si_lcao_wan_bndi.input
```

会产生主要的输出文件: si_lcao_wan_bndi.input.mat 和 si_lcao_wan_bndi.input.h5。

方法二: wannier 函数能带计算

(1) 准备输入文件 si_lcao_wbs_bndi.input

si_lcao_wbs_bndi.input 文件参考如下:

```
rho.in = 'results/si_lcao_wan_wino'  
info.savepath = './results/si_lcao_wbs_bndi'  
info.calculationType = 'wannier-bs'
```

Notes: (1) info.calculationType 设置为 wannier-bs; (2) 调用 wannier 函数计算方法二的计算结果。

(2) wannier 函数能带计算

打开 Matlab 软件, 进入 Si 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i si_lcao_wbs_bndi.input
```

会产生主要的输出文件: si_lcao_wbs_bndi.input.mat 和 si_lcao_wbs_bndi.input.h5。

```
>> rescu -p results/si_lcao_wbs_bndi.input.mat
```

得到 wannier 能带结构图如下:

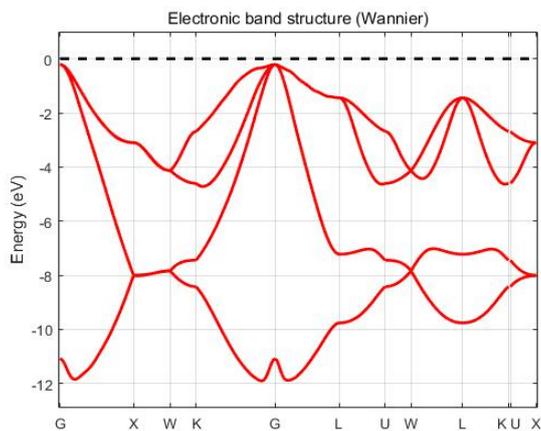


图 1-20 Si 的 wannier 能带图

参数常用设置及经验分享:

参数	含义	设置及建议
----	----	-------

info.calculationType	计算内容	wannier
units.energy	控制能量单位	能量单位默认 Hartree
wannier.outwin	指定外部能量窗口	能量单位默认 Hartree
wannier.innwin	指定内部能量窗口	
wannier.shiftEf	wannier.outwin 能量窗口设置是相对于费米能级的;	true
wannier.orbi	决定使用那些轨道作为投影基	

1.11 STM 仿真

在本节中,我们将介绍使用巴丁顿形式来获得从样品流向针尖的电流的步骤。我们对巴丁顿的理论进行基本回顾。该理论的主要假设如下:

1. 隧穿电流足够微弱采用一阶含时微扰理论;
2. 针尖和样品几乎是正交的;
3. 电子-电子相互作用可以忽略;
4. 针尖和样品的占据位置是相互独立的;
5. 针尖和样品均处于电化学平衡状态。

利用上述假设,推导出如下电流方程

$$I = \frac{2\pi e}{\hbar} \sum_{\mu\nu} [f_{DF}(E_{\mu}^S - E_F^S) - f_{DF}(E_{\mu}^T - E_F^T)] |M_{\mu\nu}|^2 \delta(E_{\mu}^T - E_{\nu}^T - E - eV) \quad (1.11.1)$$

e 是电子的电荷; \hbar 是约化普朗克常数; μ, ν 是能带指数; f_{DF} 是费米-狄拉克统计分布; E_{μ}^S, E_{μ}^T 是样品针尖的本征值; E_F^S, E_F^T 是样品或针尖的费米能级; V 是样品和针尖之间施加的偏压,最后, $M_{\mu,\nu}$ 是隧穿矩阵元, 定义为

$$M_{\mu\nu} = \frac{\hbar}{2m} \int_{z=z_0} dx dy [\psi_{\mu}^S \frac{\partial \psi_{\nu}^T}{\partial z} - \frac{\partial \psi_{\mu}^S}{\partial z} \psi_{\nu}^T] \quad (1.11.2)$$

在 1.10.2 公式中, 面积分的区域仅限于单个原胞。

RESCU 评估了公式 1.10.1, 1.10.3 的连续版本

$$I = I = \frac{2\pi e}{\hbar} \int d\varepsilon [f_{FD}(\varepsilon - eV - E_F^S) - f_{FD}(\varepsilon - E_F^T)] n^S(\varepsilon + eV) n^T(\varepsilon) M(\varepsilon + eV, \varepsilon) \quad (1.11.3)$$

这是在 **stm** 电流计算中完成的, 在 RESCU 中计算公式 1.11.3 如下:

1. 求解样品加针尖系统 (作为一个整体系统) 的 **KS** 方程。这提供了样品和针尖具有相同的本征值 $E_\mu^{S,T}$ 和本征态 ψ_μ , 因为整个系统是计算出来的。

2. 计算样品 $n^S(\varepsilon)$ 和针尖 $n^T(\varepsilon)$ 的态密度;

3. 通过计算将整个系统分解为样品波函数和针尖波函数 $\psi = a^S \psi^S + a^T \psi^T$,

$\psi^{S,T} = \sum a_n^{S,T} \phi_n^{S,T}$ 分别是以样品和针尖原子为中心的原子轨道波函数,

$$a_n^{S,T} = \langle \phi_n | \psi \rangle;$$

4. 使用公式 1.10.2 计算隧穿矩阵元 $M_{\mu,\nu}$;

5. 计算公式 1.10.3, 插值矩阵元 $M_{\mu,\nu}$ 获得 $M(\varepsilon + eV, \varepsilon)$ 。

RESCU 的这项 **STM** 新功能适用条件如下:

1. 仅适用原子轨道计算 (LCAO.status=1)
2. 针尖位于样片正上方而且沿+z 方向;
3. 在 $z=z_0$ 处的平面把样品和针尖分离;
4. **k**-抽样目前仅限于伽马点 Γ , 因为该功能是为了研究真实的大样本和针尖而开发的。

本节将介绍如何使用 RESCU 进行 **STM** 仿真计算, 我们以 **CW** 体系为例:

1.11.1 自洽计算

(1) 准备输入文件 **scf.input**, **CW.xyz** 和基组文件 **C_DZP.mat**, **W_DZP.mat**

scf.input 文件参考如下:

```

info.calculationType      = 'self-consistent'
info.savepath             = './results/CW_scf'
%symmetry.spacesymmetry  = 0
domain.latvec             = [[6.15 10.652 0]; [6.15 -10.652 0]; [0 0 14.2878]]
%eigensolver.emptyBand   = 16
eigensolver.maxit        = 15
eigensolver.algo         = 'cfsi'
LCAO.status              = 1
smearing.sigma           = 0.01
units.energy             = 'eV'
kpoint.gridn             = [1,1,1]
domain.lowres            = 0.3
domain.boundary          = [1 1 2]
%domain.highres         = 0.3
%functional.libxc        = 1
functional.list          = {'XC_LDA_X','XC_LDA_C_PW'}
option.maxSCFiteration    = 200
mixing.type              = 'density'
mixing.method            = 'pulay'
mixing.tol               = [1e-05,1e-05]
mixing.maxhistory        = 20
spin.type                = 'degenerate'
element(1).species       = 'C'
element(1).path          = './C_DZP.mat'
element(2).species       = 'W'
element(2).path          = './W_DZP.mat'
atom.xyz                 = './CW.xyz'
units.length             = 'Angstrom'
gpu.status               = 0

```

CW.xyz 参考文件如下:

```

51
AtomType  X Y Z
C          1.22999989    0.71014074    5.53387061
C          1.22999989   -0.71014074    5.53387061
.....
C          9.83999929    1.42028163    5.53387061
C          11.06999920    0.71014071    5.53387061
C          11.06999920   -0.71014071    5.53387061
W          0.00000000    0.00000000    8.75391408

```

Notes: (1) `domain.boundary`=[1,1,2]。如果 `domain.boundary` 设为 1 时，用周期边界条件求解 Poisson 方程。**RESCU** 默认是周期性边界条件；如 `domain.boundary` 设为 2 时，采用 Neumann 边界条件求解 Poisson 方程。

(2) 自洽计算

打开 Matlab 软件，进入 CW 体系的文件目录下，在 Matlab 中输入命令：

```
>> rescu -i scf.input
```

自洽结束后，会产生主要的输出文件：`CW_scf.mat` 和 `CW_scf.h5`

1.11.2 STM 计算

(1) 准备输入文件 `stm.input`

`stm.input` 文件参考如下：

```
info.calculationType    = 'stm-current'  
info.savepath           = './results/CW_stm'  
  
rho.in                  = './results/CW_scf'  
stm.z0                  = 14  
stm.voltage              = 1.2  
units.energy            = 'eV'
```

Notes: (1) `stm.z0` 表示隧穿矩阵元在平面 $z=z_0$ 上计算，单位是 Bohr。系统分解为 z_0 平面下方（样品）和上方（针尖）轨道，因此，样品 (n^S) 和针尖 (n^T) 的 PDOS 以及样品 (ψ^S) 和针尖 (ψ^T) 的波函数都依赖于 `stm.z0` 值； (2) `stm.voltage` 指定样品和针尖之间的偏压大小。

(2) `stm` 计算

打开 Matlab 软件，进入 CW 体系的文件目录下，在 Matlab 中输入命令：

```
>> rescu -i stm.input
```

会产生主要的输出文件：`stm.mat` 和 `stm.h5`。

```
>> load('CW_stm.mat')
>> stm.I
ans =
    -2.3085e-18    #计算得到的电流值单位 (amp)
```

参数常用设置及经验分享:

参数	含义	设置及建议
info.calculationType	计算方法	stm-current
rho.in	调用自洽计算结果的路径	
stm.z0	隧穿矩阵元计算在平面 z=z0 上	单位 Bohr
stm.voltage	指定样品和针尖之间的偏压大小	

1.12 贝利曲率计算

贝利曲率定义如下

$$\Omega_n(k) = \nabla_k \times A_n(k) \quad (1.12.1)$$

$$A_n(k) = i \langle u_{nk}(r) | \nabla_k | u_{nk}(r) \rangle \quad (1.12.2)$$

由于 k 空间梯度，公式 1.11.2 很难用数值计算。因此，RESCU 采用这个等价的 sum-over-eigenstates 公式

$$\Omega_{n,xy}(k) = -2 \operatorname{Im} \sum_{m \neq n} \frac{\langle \psi_{nk} | \hat{v}_x | \psi_{mk} \rangle \langle \psi_{mk} | \hat{v}_y | \psi_{nk} \rangle}{(\varepsilon_{mk} - \varepsilon_{nk})^2} \quad (1.12.3)$$

其中速度矩阵元定义如下

$$v_{nm,\alpha}(k) = \langle \psi_{nk} | \hat{v}_\alpha | \psi_{mk} \rangle = \left\langle \psi_{nk} \left| \frac{\partial \hat{H}(k)}{\partial k_\alpha} \right| \psi_{mk} \right\rangle \quad (1.12.4)$$

$$\hat{H}(k) = e^{-ik \cdot r} H e^{ik \cdot r} \quad (1.12.5)$$

Berry 曲率计算功能是为了研究特定的二维异质结构（例如沉积在氮化硼上的石墨烯）而开发的，因此，它现在受到以下条件限制：

1. 系统必须是二维体系，体系面内方向垂直 z 方向；

2. 贝利曲率在 $k_z = 0$ 平面计算;
3. 系统求解必须通过数值原子轨道求解 (LCAO.statutes=true);
4. 仅针对 degenerate spin 系统进行了测试, 但是支持 collinear spin 系统。

本节将介绍如何使用 RESCU 进行贝利曲率计算。请注意, 仅适用于 LCAO 计算 (LCAO.statue=1)。在这一节中, 我们以 graphene-boron-nitride heterostructure 体系为例:

1.12.1 自洽计算

(1) 准备输入文件 gbn_lcao_scf.input, Atom.xyz 和基组文件 C_ONCV_LDA.mat, B_ONCV_LDA.mat, N_ONCV_LDA.mat

gbn_lcao_scf.input 文件参考如下:

```
LCAO.status           = 1
info.calculationType  = 'self-consistent'
info.savepath         = './results/gbn_lcao_scf'
domain.latvec         = [[2.32436 4.02591 0]; [2.32436 -4.02591 0]; [0 0 30]]
eigensolver.algo     = 'cfsi'
kpoint.gridn         = [15,15,1]
domain.lowres         = 0.3
%functional.libxc     = 1
%functional.list      = {'XC_LDA_X','XC_LDA_C_PW'}
option.maxSCFiteration = 200
mixing.type           = 'density'
mixing.method         = 'pulay'
mixing.tol            = [1e-05,1e-05]
mixing.maxhistory     = 20
spin.type             = 'degenerate'
atom.xyz              = './gbn.xyz'
units.length         = 'Angstrom'
element(1).species    = 'C'
element(1).path       = './C_OV_LDA.mat'
element(2).species    = 'B'
element(2).path       = './B_OV_LDA.mat'
element(3).species    = 'N'
element(3).path       = './N_OV_LDA.mat'
gpu.status            = 0
```

ggn.xyz 文件如下:

4	AtomType	X	Y	Z
	C	2.32436297	1.34197158	18.04250000
	C	2.32436297	-1.34197158	18.04250000
	B	2.32436297	1.34197158	11.95750000
	N	2.32436297	-1.34197158	11.95750000

(2) 自洽计算

打开 Matlab 软件, 进入 graphene-boron-nitride heterostructure 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i gbn_lcao_scf.input
```

自洽结束后, 会产生主要的输出文件: **gbn_lcao_scf.mat** 和 **gbn_lcao_scf.h5**

1.12.2 贝利曲率计算

(1) 准备输入文件 gbn_lcao_ber.input

gbn_lcao_ber.input 文件参考如下:

```
info.calculationType = 'berry-curvature'  
info.savepath        = './results/gbn_lcao_ber'  
rho.in               = './results/gbn_lcao_scf'  
kpoint.gridn         = [102,102,1]  
berry.bandIndex      = 9
```

Notes: (1) kpoint.gridn 要求 k 空间格点数目取值很密集; (2) berry.bandIndex 确定计算 Berry 曲率的能带索引。

(2) berry-curvature 计算

进入 graphene-boron-nitride heterostructure 体系的文件目录下, 在 Matlab 中输入命令:

```
>> rescu -i gbn_lcao_ber.input
```

或在服务器上进入 graphene-boron-nitride heterostructure 体系的文件目录下, 使用任务提交脚本计算 rescu -i gbn_lcao_ber.input 文件。计算完成之后在 results 下得到 gbn_lcao_ber.mat, 其中贝利曲率保存在 berry.Omega 中。将该文件导入到本地使用 Matlab, 输入以下命令:

```
>> rescu -p results/gbn_lcao_ber.mat
```

得到贝利曲率图如下：

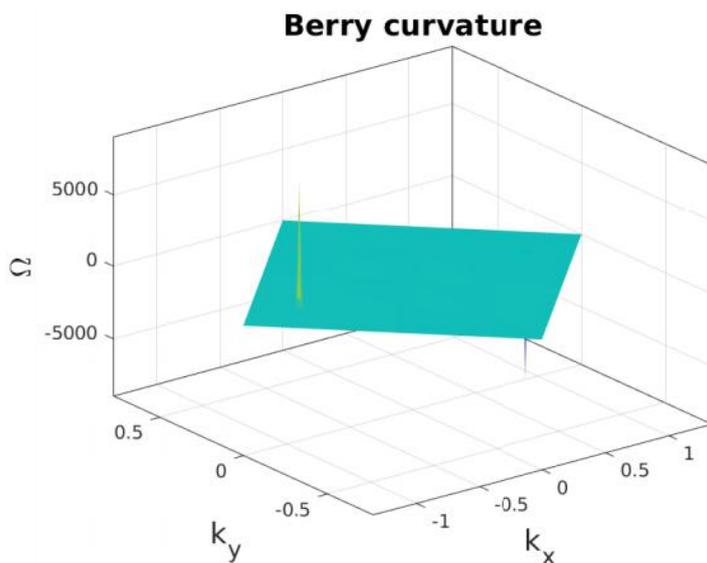


图 1-21. 贝利曲率图

参数常用设置及经验分享：

参数	含义	设置及建议
info.calculationType	计算内容	berry-curvature
berry.bandIndex	决定计算 Berry 曲率的频带指数	

1.13 加电场功能计算

本节将介绍如何使用 RESCU 中的施加电场模块进行计算，在 RESCU 中施加的电场被视为一定的边界条件，在此基础上求解 Poisson 方程。在这一节中，我们以 silicene 体系为例：

1.13.1 自洽计算

(1) 准备输入文件 scf.input, silicene.xyz 和基组文件 Si_DZP.mat.

scf.input 文件参考如下：

```

info.calculationType      = 'self-consistent'
info.savepath             = './results/silicene_scf'

%symmetry.spacesymmetry  = 0
domain.latvec             = [[3.86 0 0]; [-1.93 3.34286 0]; [0 0 20]]

%eigensolver.emptyBand   = 16
eigensolver.maxit        = 15
eigensolver.algo         = 'cfsi'
LCAO.status               = 1

smearing.sigma           = 0.01
units.energy              = 'eV'
kpoint.gridn              = [21,21,1]

domain.lowres             = 0.3
%domain.highres           = 0.3
functional.libxc          = 1
functional.list           = {'XC_LDA_X','XC_LDA_C_PW'}

domain.boundary           = [1,1,4];
domain.bvalz              = {0,0.734996876263276}

option.maxSCFiteration    = 200
mixing.type               = 'density'
mixing.method             = 'pulay'
mixing.tol                = [1e-05,1e-05]
mixing.maxhistory         = 20

spin.type                 = 'degenerate'

element(1).species        = 'Si'
element(1).path           = './Si_DZP.mat'
atom.xyz                  = './silicene.xyz'
units.length              = 'Angstrom'
gpu.status                 = 0

```

silicene.xyz 文件参考如下:

```

2
AtomType  X Y Z
Si        0.96500000    0.55714301    10.50000000
Si        0.96526134    2.78587216    10.03405467

```

Note: (1) 如果需要在开放边界上增加了一个外部电势, 即指定外部场情况下, domain.boundary 必须设置为 4, 求解 Poisson 方程用 Dirichlet 边界条件。施加的外部电场可以使用 domain.bvalx, domain.bvaly, domain.bvalz 关键字分别确定沿 x,y,z 方向的边界值。该值定义了边界上电势值的标量, 只有在 domain.boundary=4 时, 该值才有意义, (2) 电场的单位为 hartree, 不受 energy.units 控制。本案例中 $0.734996876263276 * 27.211 = 20\text{eV}$ 。

(2) 自洽计算

在服务器上进入 silicene 体系的文件目录下, 使用任务提交脚本计算 silicene.input 文件。计算完成之后在 results 下得到 silicene_scf.mat 和 silicene_scf.h5

1.13.2 能带计算

(1) 准备输入文件 band.input

band.input 文件参考如下:

```
info.calculationType      = 'band-structure'  
info.savepath             = './results/silicene_bs'  
rho.in                   = './results/silicene_scf'  
LCAO.status              = 1  
kpoint.gridn             = 500  
kpoint.symptoms          = {'G','M','K','G'}  
%kpoint.symptoms         = {[0 0 0],[0.5 0 0],[0.333333 0.333333 0],[0 0 0]}  
  
%eigensolver.emptyBand = 16
```

(2) 自洽计算

在服务器上进入 silicene 体系的文件目录下, 使用任务提交脚本计算 band.input 文件。计算完成之后在 results 下得到 silicene_bs.mat 和 silicene_bs.h5。

将该文件导入到本地使用 Matlab, 输入以下命令:

```
>> rescu -p ./results/silicene_bs.mat
```

得到施加电场 silicene 能带结构图如下:

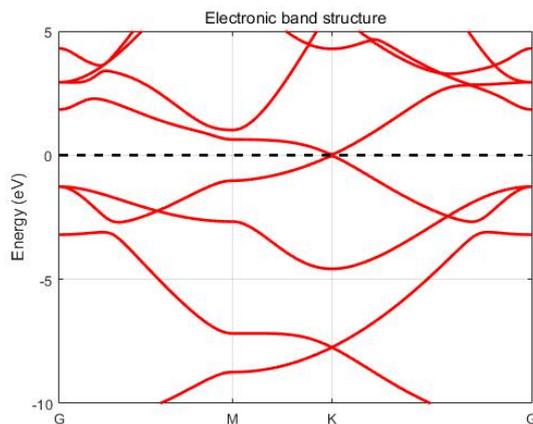


图 1-22 加电场 silicene 能带结构图

加电场 silicene 能带结构图在高对称点 K 附近局部放大，可以发现能带结构打开了一定的带隙。如图 1-23 所示。通过尝试不同的电压值会发现带隙基本上是随电场线性增加。

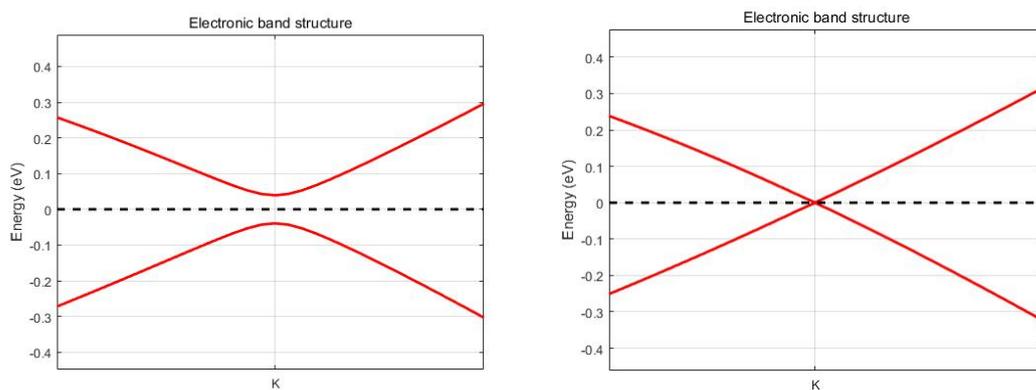


图 1-23 左施加电场和右未施加电场 silicene 能带结构局部放大图

未施加电场 silicene 能带结构带隙为零，如图 1-24 所示。

参数常用设置及经验分享:

参数	含义	设置及建议
domain.boundary	控制边界条件类型	施加电场方向必须设置为 4
domain.bvalx	指定施加电场沿 x 方向	
domain.bvaly	指定施加电场沿 y 方向	
domain.bvalz	指定施加电场沿 z 方向	

第二章 专题应用

本章目前只有第一部分，之后将使用真实的案例不断的扩充：

(1) 双层扭转黑磷摩尔条纹

本章以已发表的科研论文为题材，从 Device Studio 建模过程出发，到计算、分析数据，重述了 RESCU 科研文章的整个过程。涉及到的计算性质有：能带计算、总态密度计算、局域态密度计算、波函数分析。

用户可以根据自己的研究方向，选择性的阅读相应的部分。

第一部分：双层扭转黑磷摩尔条纹

由二维材料堆叠而成的范德瓦尔斯异质结大大扩展了具有独特性质的二维材料相空间。但是在实验上制备范德瓦尔斯异质结的过程中，由于存在于两层之间的较小的晶格失配和相对转动，会在范德瓦尔斯异质结中引入摩尔条纹。并且摩尔条纹也会对范德瓦尔斯异质结的性质造成影响，如，诱导一些新奇的电子性质。因此本章将以 900 个原子的双层扭转黑磷结构为例，重复作者的部分工作。

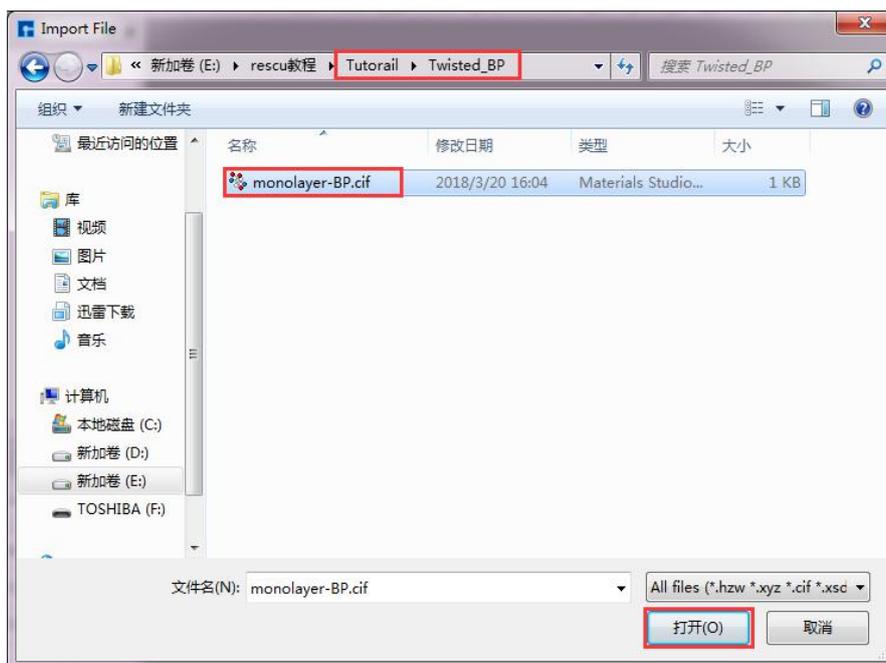
Physical Review B, 2017, 96(19): 195406.

2.1 Device Studio 建模

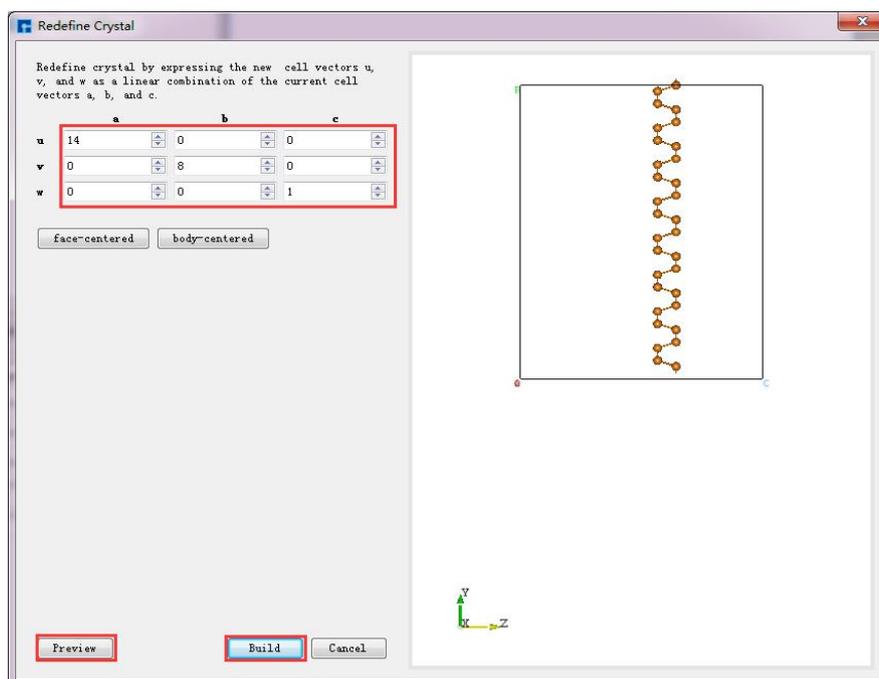
以 900 个原子的双层扭转黑磷结构为例，简单的介绍下使用 Device Studio 搭建该结构的过程。

2.1.1 结构搭建

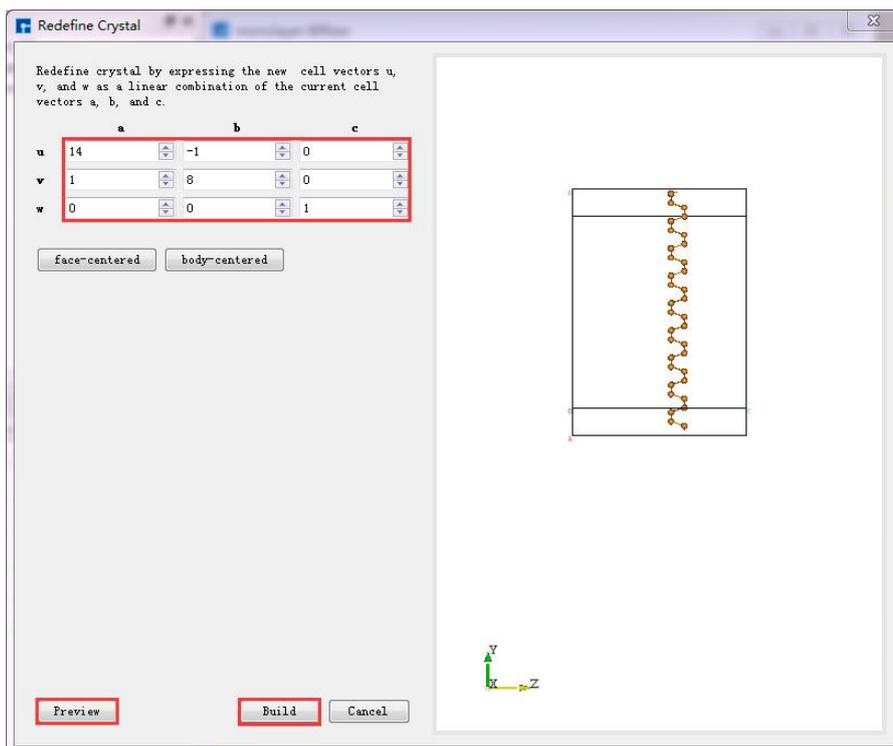
- (1) 双击图标“[DeviceStudio 快捷方式](#)”打开软件；
- (2) [Create a new Project-OK](#)—文件名：[Twisted-BP](#)，保存类型：[ProjectFiles \(*.hpf\)](#)-保存即可；
- (3) 从 [Tutorail\Twisted_BP](#) 文件夹中导入单层黑磷结构，[File-Import...](#)，找到 [monolayer-BP.cif](#) 并导入；



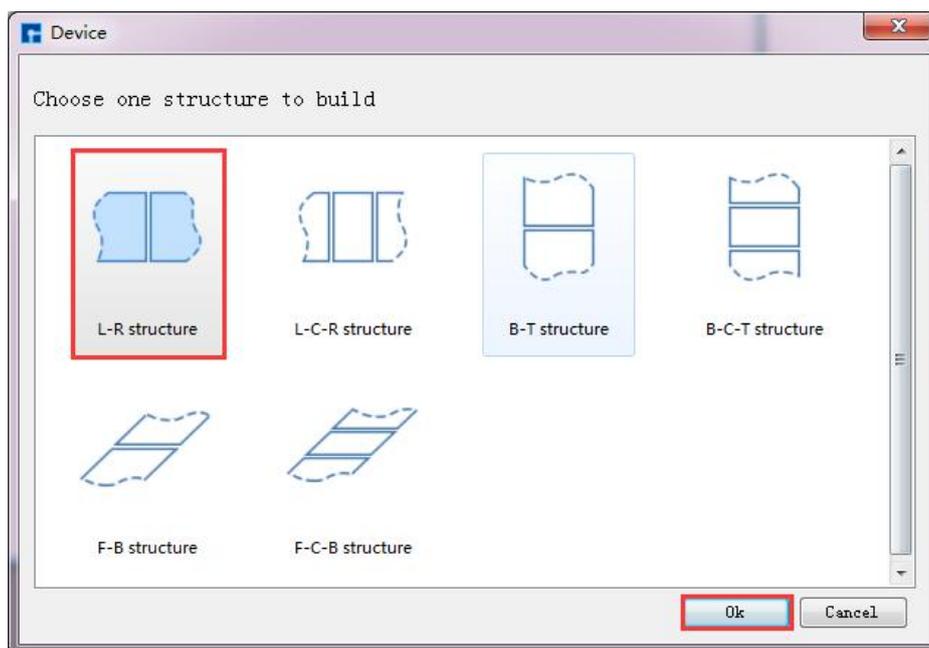
(4) **Build-Redefine Crystal**, 按如下操作将对黑磷单胞进行 $14*8*1$ 周期的扩胞操作得到 `monolayer-BP_Rede.hzw` 文件;



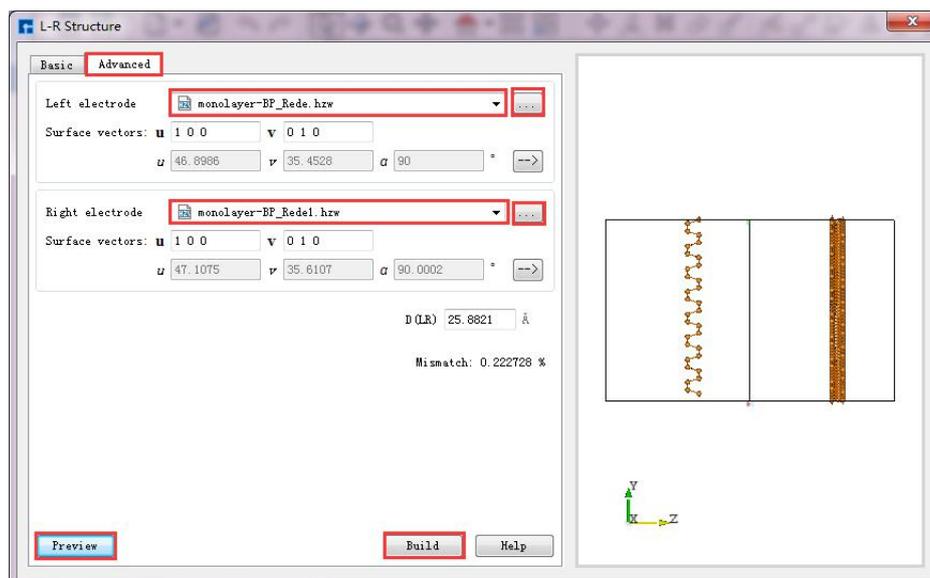
(5) 使用原始单胞 `monolayer-BP.hzw` 构造出一个倾斜角度的扩胞, 形成双层扭转黑磷结构的扭转层。这里, 通过对晶胞的基矢方向重新定义以获得与未扭转层之间的扭转角度, 双击 `monolayer-BP.hzw`, **Build-Redefine Crystal** 进行扭转角度, 得到 `monolayer-BP_Rede1.hzw` 文件。



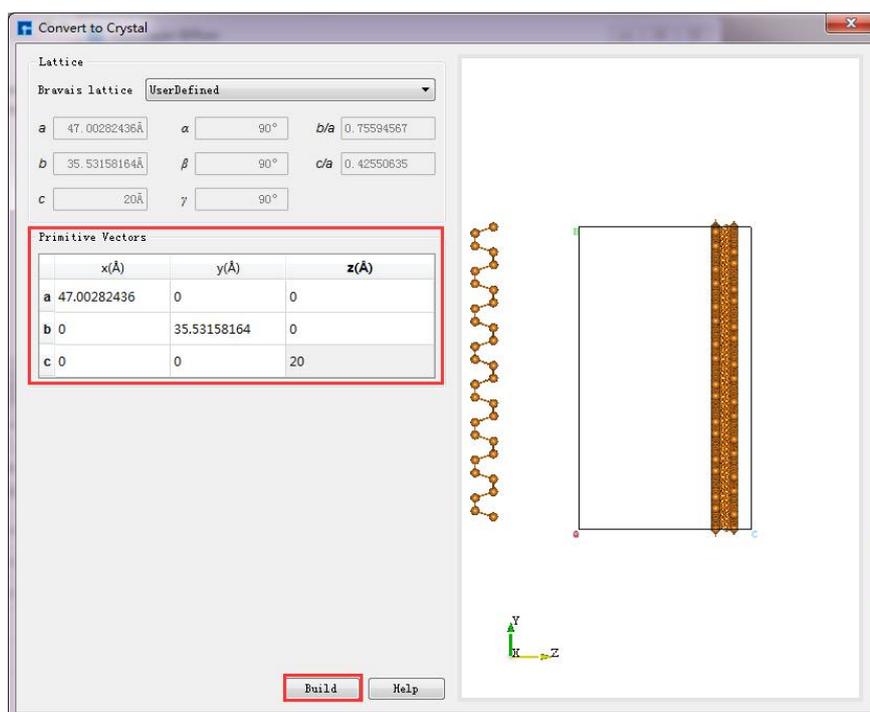
(6) **Build-Device**, 选择 L-R structure;



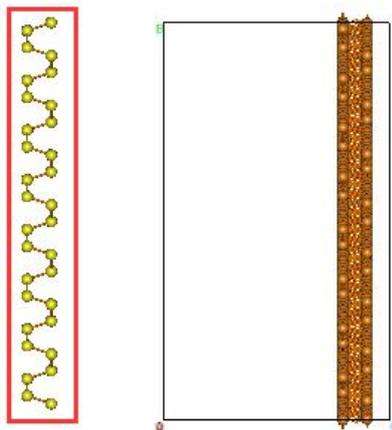
(7) 选择 **Advanced**, 点击 Left electrode 后面的 选中 monolayer-BP_Rede.hzw 文件, 点击 Right electrode 后面的 选中 monolayer-BP_Rede1.hzw 文件, 点击 **Build**, 可以得到 Device.hzw;



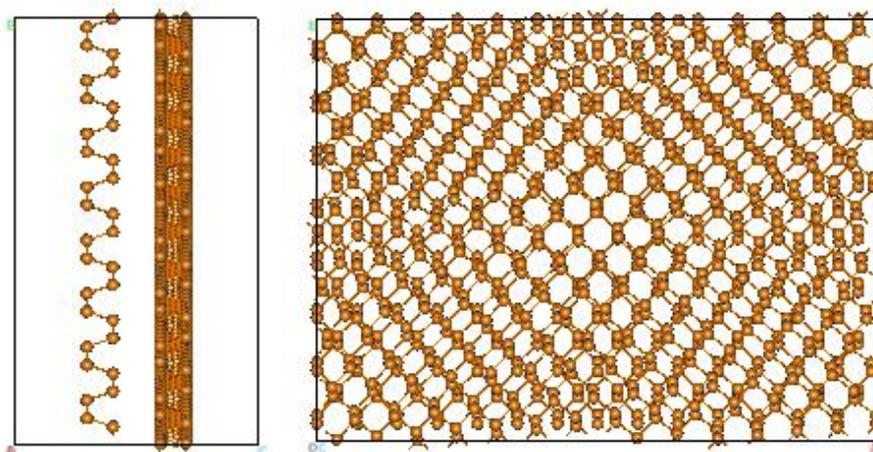
(8) 双击 **Device.hzw**，点击 **Convert to Crystal** ，按下图中的 **Primitive Vectors** 进行输入，将器件转变成晶体；



(9) 选中下图中的原子，可以在 **Properties** 栏中查看到该原子的位置信息，再点击 **Move Atom** ，调整两层之间的位置形成一个合理的双层扭转黑磷结构，本例中对选中的原子沿着+Z 方向平移 22 Å；



(10) 点击 **Center** ，将原子移至中心，点击 **3D Viewer**  下的 **xy View** 得到如下图所示结构。



注：该部分只是给用户演示双层扭转黑磷结构如何搭建，之后的计算过程直接使用已经优化好的结构进行。

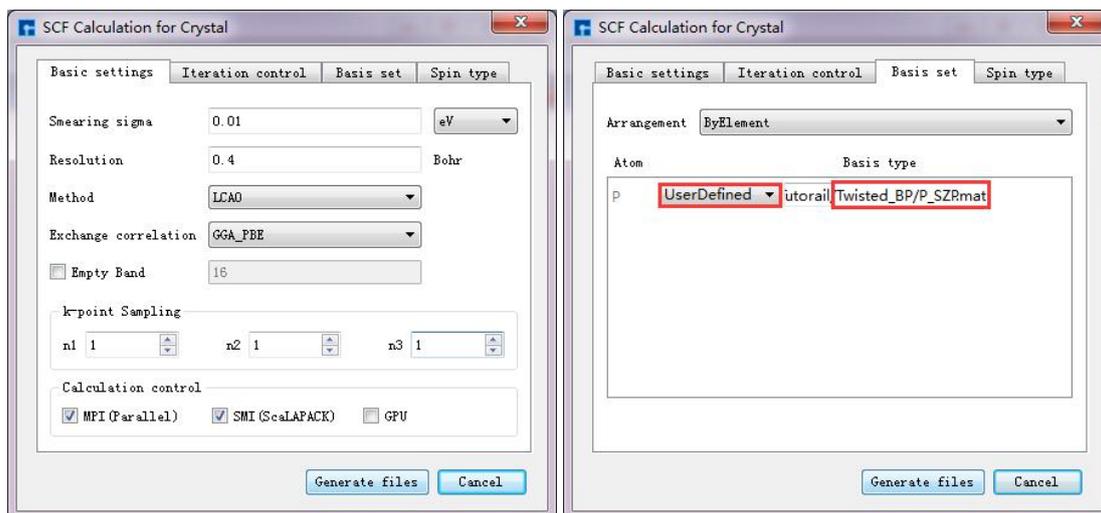
2.1.2 输入文件生成

- (1) 从 Tutorial\Twisted_BP 文件夹中导入已经搭建好的双层扭转黑磷结构，**File-Import...**，找到 BP_900.hzw 并导入；
- (2) 生成 RESCU 的自洽的输入文件：**Simulator-RESCU-SCF Calculation**；

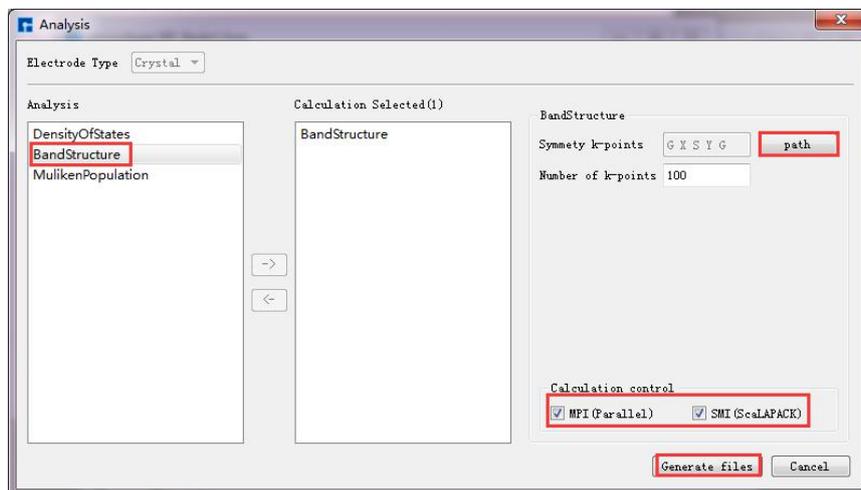


- (3) 根据自己需要选择参数设置，**Basic settings** 中需要修改这些参数：

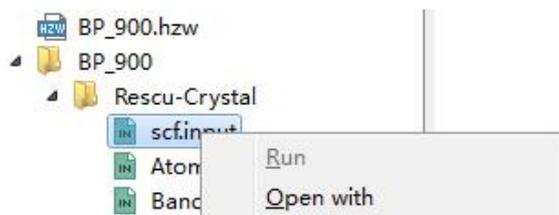
Resolution=0.4, Method=LCAO, Exchange correlation=GGA_PBE, k-point Sampling 选择 [1, 1, 1], 如果进行并行计算必须要勾选 MPI 和 SMI MPI (Parallel) SMI (ScaLAPACK); **Basis set** 中选择 UserDefined 并选择 Tutorial\Twisted_BP 中的 P_SZP.mat 基组文件, 点击 [Generate files](#) 生成相应文件夹;



(4) 生成 RESCU 的能带计算的输入文件: [Simulator-RESCU-Analysis](#), 双击 [BandStructure](#), 之后点击 [path](#) 设置高对称点为 G-X-S-Y-G, 如果需要并行则选中并行控制参数 MPI 和 SMI, 点击 [Generate files](#) 生成相应文件;



(5) 在 [Project](#) 栏下, 右键 [Rescu-Crystal](#) 下的 scf.input, 点击 [Open Containing Folder](#), 可以找到 RESCU 输入文件在电脑中的位置。



2.2 RESCU 计算过程

2.2.1 自洽计算

(1) 准备自洽所需的输入文件 `scf.input`，坐标文件 `Atom.xyz` 和基组文件 `P_SZP.mat`

`scf.input` 文件参考如下：

```
info.calculationType = 'self-consistent'
info.savepath        = './results/ BP_900_scf'

%symmetry.spacesymmetry = 0
units.latvec         = 'Angstrom'
domain.latvec        = [[47.0045 0 0]; [0 35.3518 0]; [0 0 15]]

%eigensolver.emptyBand = 16
eigensolver.algo       = 'cfsi'
LCAO.status           = 1

smearing.sigma        = 0.01
units.sigma           = 'eV'
kpoint.gridn          = [1,1,1]
domain.lowres         = 0.4
%domain.highres       = 0.3
%functional.libxc     = 1
functional.list        = {'XC_GGA_X_PBE','XC_GGA_C_PBE'}

option.maxSCFiteration = 200
mixing.type            = 'density'
mixing.method          = 'pulay'
mixing.tol             = [1e-05,1e-05]
mixing.maxhistory      = 20

spin.type              = 'degenerate'

element(1).species     = 'P'
element(1).path        = './P_SZP.mat'
atom.xyz               = 'Atom.xyz'
units.xyz              = 'Angstrom'
mpi.status             = 1
smi.status             = 1
gpu.status             = 0
domain.boundary        = [1 1 2];
```

Notes : 蓝色部分的参数由 Device Studio 设置生成，在最后加参数 **domain.boundary= [1 1 2]**，'domain.boundary'是设置体系三个方向的边界条件，1 表示周期性边界条件，2 表示冯诺依曼边界条件，在这个案例中为了减少计算量，通过设置对 Z 方向设置冯诺依曼边界条件来减少真空层的厚度，因此在 Z 方向上该体系总共的长度只有 15 Å。

Atom.xyz 文件参考如下:

```
900
AtomType X Y Z
P 45.34465939 0.14557856 3.70047105
P 45.76070673 4.54848753 3.70047105
P 46.17675407 8.95102846 3.70047105
P 46.59280141 13.35361939 3.70047105
P 47.00864872 17.75662038 3.70047105
P 42.01668073 0.46019036 3.70047105
P 42.43272807 4.86307333 3.70047105
P 42.84877541 9.26561426 3.70047105
P 43.26482274 13.66825520 3.70047105
P 43.68057005 18.07095615 3.70047105
P 44.09661739 22.47345707 3.70047105
P 44.51266473 26.87645805 3.70047105
P 44.92861205 31.27885896 3.70047105
P 38.68830202 0.77468314 3.70047105
P 39.10434936 5.17757911 3.70047105
P 39.52039670 9.58011004 3.70047105
.....
```

(2) 自洽计算

由于该案例原子数较多,因此在输入文件中开启并行开关 **mpi.status= 1**, **smi.status= 1**, 建议用户在服务器上使用任务提交脚本进行并行计算;若电脑性能比较好能够进行单机计算,请将'**mpi.status**', '**smi.status**'这两个参数设置为 0 之后,打开 Matlab 软件,进入体系的文件目录下,在 Matlab 中输入命令:

```
>> rescu -i scf.input
```

并行或单机自洽结束后,会产生输出文件:BP_900_scf.mat 和 BP_900_scf.h5;所有的计算过程将保存在 rescu.log.out 文件夹下。

2.2.2 能带计算

(1) 准备输入文件 BandStructure.input

BandStructure.input 文件参考如下:

```
info.calculationType = 'band-structure'  
info.savepath        = './results/BP_900_bs'  
rho.in               = './results/BP_900_scf.mat'  
LCAO.status          = 1  
element(1).species   = 'P'  
element(1).path      = './P_SZP.mat'  
kpoint.type          = 'line'  
kpoint.gridn         = 100  
kpoint.symptoms      = {'G','X','S','Y','G'}  
%kpoint.symptoms     = {[0 0 0],[0.5 0 0],[0.5 0.5 0],[0 0.5 0],[0 0 0]}  
  
%eigensolver.emptyBand = 16  
mpi.status            = 1  
smi.status            = 1  
eigensolver.emptyBand = 400  
domain.bravaisLattice = 'ORC'
```

Notes: 蓝色部分是 Device Studio 生成的,最后加了 **eigensolver.emptyBand = 400**, **domain.bravaisLattice= 'ORC'**, 在自洽计算的输出文件 rescu.log.out 的开头存在着一个 Warning: Warning: RESCU recognized the lattice to be an orthorhombic lattice. However, $a_1 > a_2$ (norms). Please use a basis in which $a_1 < a_2$. Otherwise, the lattice will be treated as a triclinic lattice. 这说明自洽计算时 RESCU 将'ORC'布拉菲晶格当做了'TRI'布拉菲晶格进行的计算,为了能够让 RESCU 自动识别高对称点,要将布拉菲晶格重新设置为'ORC';因 RESCU 计算能带时只会计算少量的空带,所以需要根据体系大小自己设置空带数目,该体系中设置空带为 400。

(2) 能带计算

服务器上使用任务提交脚本进行并行计算,单机版计算请将'mpi.status', 'smi.status'这两个参数设置为 0 之后,打开 Matlab 软件,进入体系的文件目录下,在 Matlab 中输入命令:

```
>> rescu -i BandStructure.input
```

计算结束后,会产生输出文件: BP_900_bs.mat 和 BP_900_bs.h5

在 Matlab 中输入命令:

```
>> rescu -p ./results/ BP_900_bs.mat
```

得到能带结构图如下：

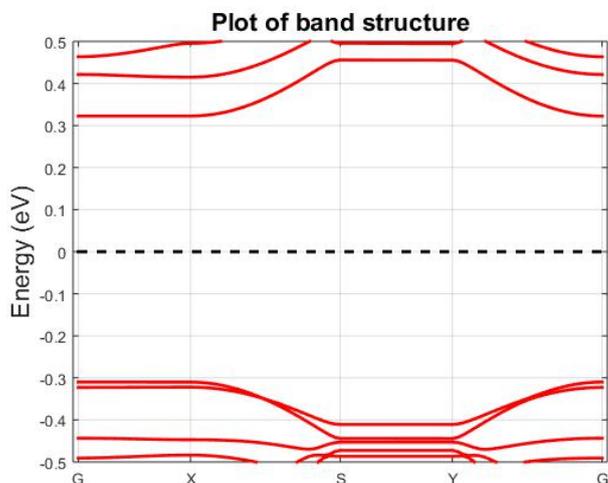


图 2-1 900 原子扭转黑磷能带结构图

(3) 能带数据分析

进入 results 文件夹，在 Matlab 中输入命令：

```
>> clear
>> load BP_900_bs.mat
```

在 Matlab 的 workspace 双击 band，如图二左所示，

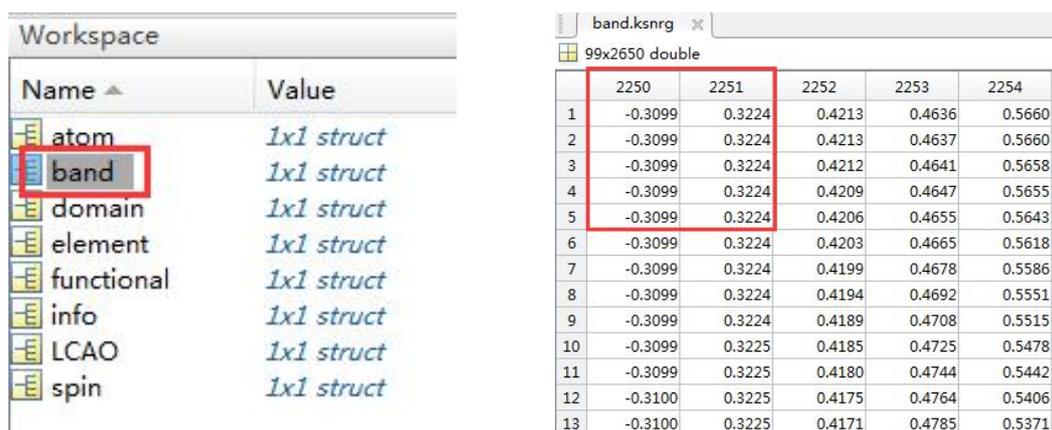


图 2-2 Matlab 中读取数据操作示意图

双击 ksnrg，（ksnrg 是一个 99*2650 的矩阵，99 代表高对称点数目，2650 代表能带数目），可以在所有的能带数据中价带顶与导带底分为位于第 2250 及 2251 两条带上。

2.2.3 实空间波函数计算

(1) 准备输入文件 BandStructure_decom.input

BandStructure_decom.input 文件参考如下：

```
info.calculationType = 'band-structure'  
info.savepath       = './results/BP_900_bs_decom'  
rho.in              = './results/BP_900_scf.mat'  
LCAO.status         = 1  
element(1).species = 'P'  
element(1).path     = './P_SZP.mat'  
kpoint.type         = 'line'  
kpoint.gridn        = 100  
kpoint.symptoms     = {'G','X','S','Y','G'}  
%kpoint.symptoms   = {[0 0 0],[0.5 0 0],[0.5 0.5 0],[0 0.5 0],[0 0 0]}  
%eigensolver.emptyBand = 16  
mpi.status           = 1  
smi.status           = 1  
eigensolver.emptyBand = 400  
domain.bravaisLattice = 'ORC'  
eigensolver.bandi = [2250,2251];  
option.saveWavefunction = true;  
kpoint.kdirect = [0 0 0]
```

Notes: 将 Bandstructure.input 的所有内容复制 BandStructure_decom.input 中，然后做了如下修改：**info.savepath= './results/BP_900_bs_decom'**，表示定义输出结果路径和名字 **eigensolver.bandi = [2250,2251]**，表示只计算序号为 2250 和 2251 的能带，**option.saveWavefunction = true**，表示保存 K 点波函数，**kpoint.kdirect = [0 0 0]**表示只计算 G 点的能带。

(2) 实空间波函数计算

服务器上使用任务提交脚本进行并行计算，单机版计算请将'mpi.status'，'smi.status'这两个参数设置为 0 之后，打开 Matlab 软件，进入体系的文件目录下，在 Matlab 中输入命令：

```
>> rescu -i BandStructure_decom.input
```

计算结束后,会产生输出文件: BP_900_bs_decom.mat 和 BP_900_bs_com.h5

(3) 实空间波函数数据分析

计算完存有特定能量点的波函数之后,在 Matlab 中使用语句 h5disp('文件名.h5')查看保存的数据结构。在 Group/waveFunction 中含有名为 psi1 的波函数文件。然后使用 loadDistArray('文件名.h5', '/waveFunction/psi1')将数据存储下来。进入 results 文件夹,并将 newrealplot.m 脚本复制到该文件夹下,双击查看:

```
1. clear all;
2. %%%%%%%%%%%PARAMETER REGION%%%%%%%%%
3. output_name = 'BP_900_bs_decom'; %RESCU 输出文件名,不带后缀名
4. %%%%%%%%%%%
5. matfile = [output_name, '.mat'];
6. load (matfile);
7. filename = {'psivbm.vasp', 'psicbm.vasp'};
8. for j = 1:2;
9. %%%%%%%%%%%下面部分 VASP CONTCAR 形式
10. fid = fopen(filename{j}, 'w');
11. fprintf(fid, 'psi \n');
12. fprintf(fid, ' %6.8f\n', 1.0000000);
13. for i = 1:3
14. a = domain.latvec(i,:)*0.529177;
15. fprintf(fid, ' %6.10f %6.10f %6.10f\n', a);
16. end
17. atom_species = size(element,2);
18. number_of_atoms = size(atom.xyz,1);
19. each_atom = hist(atom.element,unique(atom.element));
20. for i = 1:atom_species
21. fprintf(fid, '%s ', element(i).species);
22. end
23. fprintf(fid, '\n');
24. for i = 1:atom_species
25. fprintf(fid, '%6.0f ', each_atom(i));
26. end
27. fprintf(fid, '\n');
28. fprintf(fid, 'Cartesian\n');
29. for i = 1:number_of_atoms
30. b = atom.xyz(i,:)*0.529177;
```

```
31. fprintf(fid, '%6.9f %6.9f %6.9f\n', b);
32. end
33. fprintf(fid, '\n');
34. %%%%%%%%%%写入实空间格点数
35. realgrid = domain.fgridn;
36. fprintf(fid, '%6.0f %6.0f %6.0f\n', realgrid);
37. %%%%%%%%%%将 G 点出的 VBM,CBM 的波函数写入，10 个数据一行
38. h5file = [output_name, '.h5'];
39. psi = loadDistArray(h5file, '/waveFunction/psi1');
40. psixyz = psi.data(:,j);
41. psixyz = psixyz.*conj(psixyz);
42. %fprintf(fid, '%f %f %f %f %f %f %f %f %f %f\n',psixyz);
43. fprintf(fid, '%15.10f %15.10f %15.10f %15.10f %15.10f %15.10f %15.10f %15.10f %15.10f %15.10f\n',psixyz);
44. fclose(fid);
45. end
```

在 Matlab 中输入命令，运行脚本：

```
>> run newrealplot.m
```

运行结束后在当前文件夹中将产生 **psivbm.vasp** 和 **psicbm.vasp** 文件。

Notes: newrealplot.m 只适合用于计算某一 K 点下的导带价带两个点的实空间波函数的情况，可将上述脚本中红色部分替换成自己体系输出结果的名称即可；如果计算了多个 K 点下的多条能带的实空间波函数，此时需要对脚本 **newrealplot.m** 进行适当的修改才能正常运行。

(4) 实空间波函数图像化

上一步已经得到 G 点处导带价带的实空间波函数模平方的数据 **psivbm.vasp** 和 **psicbm.vasp**，为了将这两个数据图像化，要用到一个软件 VESTA，双击打开 VESTA 软件，将 **psivbm.vasp** 和 **psicbm.vasp** 分别拖入 VESTA 软件中。可以得到 CBM，VBM 处实空间波函数的图像，如下图：

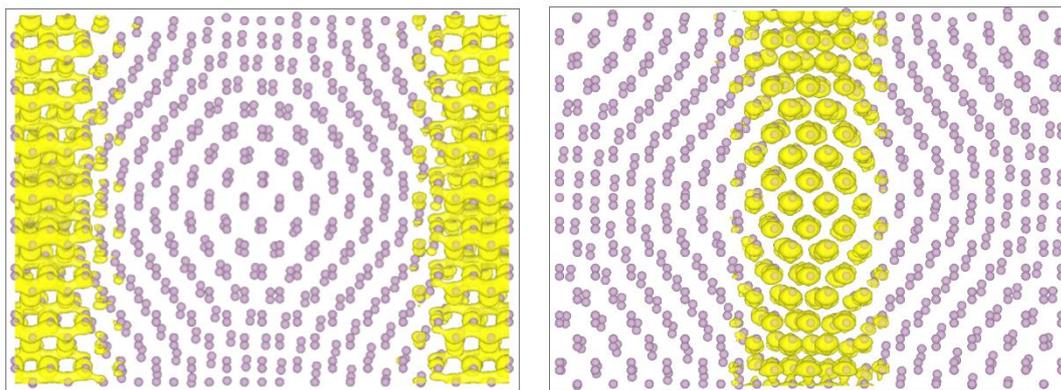


图 2-3 90 度原子扭转黑磷实空间波函数图（左：CBM 右：VBM）

2.3 原文解读

通过对双层扭转黑磷电子结构的研究发现，摩尔杂质（同时存在于双层扭转黑磷结构中的四种不同的局部的高对称堆叠方式可以看成周期性分布于二维双层结构中的杂质，简称摩尔杂质）会引发其价带顶与导带低处实空间波函数的局域化现象，从而在能带结构中形成平能带。当扭转角度从 1.8 度增大到 5.4 度时，导带低处实空间波函数在 X 方向一直保持着局域，而在 Y 方向的局域性随着角度的增加逐渐降低；对于价带顶处实空间波函数，X 方向则始终保持着局域，Y 方向始终处于不局域的状态，因此随着波函数的局域化的降低，作者猜测体系的载流子迁移率将会逐渐增大。为了验证此猜测作者进行了载流子迁移率的计算，计算结果表明：当扭转角度从 1.8 度增大到 5.4 度时，电子迁移率随着扭转角度增大而增大，这个结果与实空间波函数局域化的趋势相同。

第三章 并行版安装教程

3.1 系统及环境要求

3.1.1 系统要求

并行版 RESCU 软件需要在 linux 环境下运行，目前 CENTOS 6.X 的系统能够正常安装并行版 RESCU，建议用户使用该系统，其他系统尚处于测试中。

3.1.2 软件环境要求

并行版 RESCU 软件使用 MPI 进行并行计算，且同时可以使用 scalapack 数学库进行矩阵加速计算。

目前 openmpi 对并行版 RESCU 的加速效果较好，建议用户安装 openmpi。以下是具体的软件版本要求：

- MPI 并行 RESCU 需要：
 1. GCC-4.7.4 或 GCC-4.9.2;
 2. openmpi-1.8.5;
 3. libxc-3.0.0 或 libxc-4.0.0
- Scalapack 加速 RESCU 需要：
 1. GCC-4.7.4 或 GCC-4.9.2;
 2. openmpi-1.8.5;
 3. libxc-3.0.0 或 libxc-4.0.0
 4. openblas-0.2.14/openblas-0.2.15 或openblas-0.2.20
 5. lapack-3.5.0
 6. scalapack-2.0.2

注：在安装并行版 RESCU 过程中我们还需要 matlab2017 及以上的 matlab 对 RESCU 的程序进行编译。

3.2 软件安装过程

在安装并行版 RESCU 的过程中，安装过程需要完全按照以下顺序及操作来，具体安装过程如下：

1. GCC-4.7.4 安装

- 1) 将RESCUPackage.tar.gz 文件上传到服务器上。
- 2) 使用 `tar -zxf RESCUPackage.tar.gz` 解压 RESCUPackage.tar.gz 文件。
- 3) 进入 `cd RESCUPackage` 进入文件夹。
- 4) 使用 `gcc -v` 查看 GCC 版本，如果版本为上述提到的两个版本，则跳过 GCC 安装过程。
- 5) 使用 `tar -zxvf gcc-4.7.4.tar.gz` 解压 gcc-4.7.4.tar.gz 文件。
- 6) 使用 `mkdir GCC` 新建 GCC 安装目录。
- 7) 使用 `cd gcc-4.7.4` 进入 gcc-4.7.4 文件夹并新建 BUILD 目录 `mkdir BUILD`。
- 8) 在 BUILD 目录下进行 GCC 编译，并将 GCC 安装到 RESCUPackage 目录下的GCC 目录下。

GCC 安装的具体命令如下：

```
1. tar -zxf RESCUPackage.tar.gz
2. cd RESCUPackage
3. tar -zxvf gcc-4.7.4.tar.gz
4. mkdir GCC
5. cd ../gcc-4.7.4/
6. mkdir BUILD
7. cd BUILD
8. ../configure --prefix=/your/own/GCC/folder --enable-checking=release
  --enable-languages=c,c++,fortran --disable-multilib
9. make -j8
10. make install -j8
```

注：其中/`your/own/GCC/folder` 即为为第 6 步新建的 GCC 的安装目录，用户安装时填写自己的目录。

- 9) 安装完成之后，使用 `vi GCCPATH` 命令在 RESCUPackage 目录下新建将 GCC 的环境变量文件 GCCPATH，GCCPATH 的内容如下：

```
1. export PATH=/your/own/GCC/folder/bin:$PATH
2. export LD_LIBRARY_PATH=/your/own/GCC/folder/lib64:$LD_LIBRARY_PATH
```

2. openmpi-1.8.5 安装

安装完成 GCC 之后就可以进行 openmpi 的安装了。需要注意的是：openmpi 需要使用 GCC 编译才能正常运行，因此必须先要使得 GCCPATH 文

件生效。

- 1) 在RESCUPackage 目录下 source GCCPATH 使得 GCC 的环境生效。
- 2) 使用 which gcc 查看 GCC 的路径是否为自己安装的 GCC 路径。如果不是请重复过程 1 的 source 命令。
- 3) 之后正式进行 openmpi 安装，我们将 openmpi 安装在新建的 OpenMPI 文件夹下，具体安装过程如下：

```
1. cd RESCUPackage
2. source GCCPATH
3. which gcc
4. mkdir OpenMPI
5. tar -zxvf openmpi-1.8.5.tar.gz
6. cd openmpi-1.8.5
7. mkdir BUILD
8. cd BUILD
9. ../configure --prefix=/your/own/OpenMPI/folder --disable-dlopen
10. Make
11. make install
```

注：其中/your/own/OpenMPI/folder 即为新建的 OpenMPI 的安装目录，用户安装时填写自己的目录。

- 4) 与 GCCPATH 一样，安装完 openmpi 之后我们也需要将 openmpi 的环境变量写进 mpipath 中；使用 vi mpipath 命令在 RESCUPackage 目录下新建将 openmpi 的环境变量文件 mpipath，mpipath 的内容如下：

```
1. export PATH=/your/own/OpenMPI/folder/bin:$PATH
2. export
   LD_LIBRARY_PATH=/your/own/OpenMPI/folder/lib:$LD_LIBRARY_PATH
```

3. lapack-3.5.0 安装

安装完成 GCC 和 openmpi 之后就可以进行其他软件的的安装了。需要注意的是：其他软件都需要使用 GCC 和 openmpi 编译才能正常运行，因此必须先要使得 GCCPATH 和mpipath 文件生效。

- 1) 在RESCUPackage 目录下 source GCCPATH 使得 GCC 的环境生效。
- 2) 在RESCUPackage 目录下 source mpipath 使得 openmpi 的环境生效。
- 3) 使用 which gcc 查看 GCC 的路径是否为自己安装的 GCC 路径。如果不是请



重复过程 1 的 source 命令。

4) 使用 `which mpirun` 查看 `openmpi` 的路径是否为自己安装的 `openmpi` 路径。

如果不是请重复过程 2 的source 命令。

5) 之后正式进行 `lapack-3.5.0` 安装, 我们将 `lapack` 直接安装在 `lapack-3.5.0` 文件夹下, 具体安装过程如下:

```
1. cd RESCUPackage
2. source GCCPATH
3. which gcc
4. source mpipath
5. which mpirun
6. tar -zxvf lapack-3.5.0.tar.gz
7. cd lapack-3.5.0
8. make blaslib
9. make lapacklib
```

4. `openblas-0.2.14` 安装

1) 在 `RESCUPackage` 目录下 `source GCCPATH` 使得 `GCC` 的环境生效。

2) 在 `RESCUPackage` 目录下 `source mpipath` 使得 `openmpi` 的环境生效。

3) 使用 `which gcc` 查看 `GCC` 的路径是否为自己安装的 `GCC` 路径。如果不是请重复过程 1 的source 命令。

4) 使用 `which mpirun` 查看 `openmpi` 的路径是否为自己安装的 `openmpi` 路径。如果不是请重复过程 2 的source 命令。

5) 之后正式进行 `openblas-0.2.14` 安装, 我们直接安装在 `openblas-0.2.14` 文件夹下, 具体安装过程如下:

```
1. cd RESCUPackage
2. source GCCPATH
3. which gcc
4. source mpipath
5. which mpirun
6. tar -zxvf OpenBLAS-0.2.14.tar.gz
7. cd OpenBLAS-0.2.14
8. make CC=gcc FC=gfortran NO_AVX2=1
```

5. `libxc-3.0.0` 安装

1) 在 `RESCUPackage` 目录下 `source GCCPATH` 使得 `GCC` 的环境生效。



- 2) 在RESCUPackage 目录下 source mpipath 使得 openmpi 的环境生效。
- 3) 使用 which gcc 查看 GCC 的路径是否为自己安装的 GCC 路径。如果不是请重复过程 1 的source 命令。
- 4) 使用 which mpirun 查看 openmpi 的路径是否为自己安装的 openmpi 路径。如果不是请重复过程 2 的source 命令。
- 5) 之后正式进行 libxc-3.0.0 安装,我们将libxc 安装在新建的 LIBXC 文件夹下,具体安装过程如下:

```
1. cd RESCUPackage
2. source GCCPATH
3. which gcc
4. source mpipath
5. which mpirun
6. tar -zxvf libxc-3.0.0.tar.gz
7. mkdir LIBXC
8. cd libxc-3.0.0
9. mkdir BUILD
10. cd BUILD
11. ../configure --prefix=/your/own/LIBXC/folder CFLAGS="-O2 -fPIC"
12. Make
13. make install
```

6. scalapack-2.0.2 安装

- 1) 在RESCUPackage 目录下 source GCCPATH 使得 GCC 的环境生效。
- 2) 在RESCUPackage 目录下 source mpipath 使得 openmpi 的环境生效。
- 3) 使用 which gcc 查看 GCC 的路径是否为自己安装的 GCC 路径。如果不是请重复过程 1 的source 命令。
- 4) 使用 which mpirun 查看 openmpi 的路径是否为自己安装的 openmpi 路径。如果不是请重复过程 2 的source 命令。
- 5) 之后正式进行 scalapack-2.0.2 安装,我们直接安装在 scalapack-2.0.2 文件夹下,具体安装过程如下:

```
1. cd RESCUPackage
2. source GCCPATH
3. which gcc
4. source mpipath
5. which mpirun
```

6. `tar -zxvf scalapack-2.0.2.tar.gz`
7. `cd scalapack-2.0.2`
8. `make lib`

7. MATLAB 安装

由于在安装过程中需要 MATLAB 对 RESCU 进行编译，因此我们在 RESCUPackage 安装包放了一个免安装的 MATLAB。使用 tar 命令解压就能用。

1. `tar -zxvf matlab-2017gpu-hpc2.tar.gz`

8. RESCU 安装或更新方法

1) 所有软件安装完成之后,我们还需要在 RESCUPackage 文件夹下新建两个文件: `barc` 文件及 `smilibxc.m` 文件。`barc` 文件包含 `openmpi` 和 `MATLAB` 环境变量以及 `openblas` 的进程数。`smilibxc.m` 文件包含了 `lapack-3.5.0`、`OpenBLAS-0.2.14`、`scalapack-2.0.2` 及 `LIBXC` 编译后的文件的路径。

`barc` 文件的内容如下:

1. `export PATH=/your/own/OpenMPI/folder/bin:$PATH`
2. `export`
`LD_LIBRARY_PATH=/your/own/OpenMPI/folder/lib:$LD_LIBRARY_PATH`
3. `export MATLABROOT=/your/own/Matlab/folder`
4. `export PATH=$MATLABROOT/bin:$PATH`
5. `export OPENBLAS_NUM_THREADS=1`

`smilibxc` 文件的内容如下:

1. `lapack='/your/own/lapack-3.5.0/folder/liblapack.a';`
2. `blas='/your/own/OpenBLAS-0.2.14/folder/libopenblas.a -lz -lgfortran';`
3. `scalapack='/your/own/scalapack-2.0.2/folder/libscalapack.a';`
4. `xc='/your/own/LIBXC/folder';`

2) 之后 `source GCCPATH` 和 `source barc` 两个文件。

3) 解压 `rescu` 的源文件包, 进行安装, 过程如下:

1. `cd RESCUPackage`
2. `source GCCPATH`
3. `source barc`
4. `tar -zxf rescu-2020B-2.4.1.tar.gz`
5. `cd rescu-2020B-2.4.1`
6. `cd src`



```
7. matlab -nodisplay -r "makeRESCU -agreeToLicense y -licenseType hzw -smi  
-libxc -mpi ../../smilibxc.m;exit"
```

4) 安装过程中, 如果最后出现以下字样则说明 RESCU 安装成功。

```
1. Modifying MATLAB path to add RESCU  
2. Generating license number file.  
3. Building with 'gcc'.  
4. MEX completed successfully.  
5. Building with 'gcc'.  
6. MEX completed successfully.  
7. Building with 'gcc'.  
8. MEX completed successfully.  
9. Building with 'gcc'.  
10. MEX completed successfully.  
11. Building libxc wrapper: success  
12. Building smi: success
```

5) RESCU 用户将 rescu-2020B-2.4.1/license 文件夹下的 licenseNumber.txt 发送给对应地区的销售人员, 得到 license.dat 和 .uas 文件之后将其放入 rescu-2020B-2.4.1/license 中。

6) 如果用户需要多节点并行, 则需要在 rescu-2020B-2.4.1/license 文件夹新建 ip.dat 文件。

- a. 查询 ip: 使用命令 /sbin/ifconfig
- b. 将 inet addr 后面的 ip 添加 ip.dat 文件 (lo 开头的一段中的 ip 不需要添加)

7) 至此 RESCU 软件安装完成。

3.3 软件测试及脚本编写

在 RESCUPackage 文件下有个 blas_test.tar.gz 文件即为 RESCU 的测试文件。

使用 tar 命令解压文件并进入到该文件夹中。

下面将介绍如何编写 RESCU 任务提交脚本:

如果用户的机器上没有排队系统, 那我们则使用 bash 脚本进行 RESCU 任务提交; 如果用户机器上有 PBS 排队系统, 则我们用 PBS 排队系统提交任务。

1) bash 脚本示例:

```
1. #!/bin/sh
2. source /your/own/RESCUPackage/folder/barc
3. mpirun --map-by node:PE=1 matlab -nojvm -nosplash -nodisplay -singleCompThread -r "addpath(genpath('/your/own/rescu-2020B-2.4.1/src/folder'));rescu --smi -i$*"
```

将上述 bash 脚本命名为 rescu.sh, 并使用 `chmod +x rescu.sh` 赋予 rescu.sh 可执行权限。

使用 `rescu.sh +文件名的命令` 进行 RESCU 任务提交。例:

```
1. rescu.sh scf.input
```

2) PBS 脚本示例:

```
1. #!/bin/sh
2. #PBS -N rescu
3. #PBS -l nodes=1:ppn=16
4. #PBS -l walltime=400:00:00
5. #PBS -o ./std.log
6. #PBS -e ./err.log
7. cd $PBS_O_WORKDIR
8. source /your/own/RESCUPackage/barc
9. mpirun --map-by node:PE=1 matlab -nojvm -nosplash -nodisplay -singleCompThread -r "addpath(genpath('/your/own/rescu-2020B-2.4.1/src/folder'));rescu --smi -i scf.input"
```

将上述 PBS 脚本命名为 rescu.pbs, 使用 `qsub` 命令进行提交。例:

```
1. qsub rescu.pbs
```

测试成功后, RESCU 正式安装完成。

3.4 安装常见问题

1. 如果用户自己的 GCC 版本符合要求, 则可以跳过所有的 `source GCCPATH` 及 `which gcc` 的命令。
2. 如果 OpenBLAS-0.2.14 安装不成功, 用户可以选择 OpenBLAS-0.2.15 或更高的 OpenBLAS 版本进行尝试。

第四章 常见问题 (FAQ)

4.1 计算 RESCU 安装及资料问题

1. **问题:** 关闭 matlab 后重新打开之后无法使用 RESCU, 提示 **Undefined function or variable 'rescu'** ?

解答: RESCU 安装之后并没有在 matlab 中保存其路径, 因此在使用 RESCU 之前, 请用户进入到 rescuPackage 目录下——选择 src 文件夹——鼠标右击——选择 Add to Path——Selected Folders and Subfolders, 这样就可以了; 为了方便用户使用试用版的 RESCU, 我们将在下个版本中自动保存 RESCU 的路径。

2. **问题:** 用户初次使用 RESCU, 但是无法在 <http://www.nanoacademic.ca/> 网站生成试用版 RESCU license.dat 文件?

解答: 用户可以将 licensenumber.txt 到 support@hzwtech.com, 并清楚的描述该问题, 我们的技术人员会发送给您一个试用版的 license.dat 文件。

3. **问题:** 如何下载最新版的 RESCU 安装包及 LDA 和 PBE 的赝势基组文件?

解答: 用户可以访问 <http://www.nanoacademic.ca/> 网站, 点击 Download——RESCU, 下载自己所需要的资料。

4.2 RESCU 使用及计算参数

4. **问题:** RESCU 如何使用 LCAO 的方法进行计算?

解答: RESCU 默认使用的是 real space 实空间格点的方法进行计算的, 如果用户需要使用 LCAO 的方法进行计算, 则需要开启 LCAO.status=1 的开关。

5. **问题:** 使用 real space 实空间格点方法时 RESCU 中几种算法, 那个计算效果好?

解答: 使用 real space 实空间格点方法时 RESCU 中有两种算法, 分别是 'cfsi' 和 'lobpcg' 在体系小于 64 个原子时 RESCU 默认使用 'lobpcg' 的方法, 当体

系大于 64 个原子时则会使用 ‘cfsi’ 的方法，一般情况下 ‘cfsi’ 的方法效果比 ‘lobpcg’ 好。

6. 问题：如何使用 libxc 泛函库里的泛函？

解答：首先，单机版中无法使用 libxc 泛函库，只能在并行版中使用。如果用户需要使用 libxc 泛函库则需要开启 `functional.libxc = 1` 的开关，然后选择合适的泛函。

7. 问题：RESCU 能够考虑范德瓦尔斯力吗？

解答：libxc 泛函库中包含着相应的泛函，用户可以在并行版中进行考虑范德瓦尔斯力的计算。

8. 问题：RESCU 精度控制参数如何设置？

解答：RESCU 中精度由 `domain.lowres` 控制，`domain.lowres` 越小越准确但计算量也越大，对于一般体系 0.35 是比较合适的值，部分体系例如 Si, `domain.lowres` 能够取到 0.55 左右。

9. 问题：RESCU 计算能带或态密度时，为什么导带只有部分能带或占据态？

解答：RESCU 在自洽中使用了切比雪夫滤波技术滤掉了没有电子占据的态，因此用户可以在能带或态密度计算时设置 `eigensolver.emptyBand` 参数，通过加大 `eigensolver.emptyBand` 的数值来增加没有占据的能带或态。

10. 问题：RESCU 支持哪些泛函？

解答：RESCU 单机版支持普通的 LDA 和 GGA，即下表中的前两个；并行版支持 Libxc 中的大部分泛函。

functional name	functional.list
LDA_PW	{'XC_LDA_X','XC_LDA_C_PW'}
GGA_PBE	{'XC_GGA_X_PBE','XC_GGA_C_PBE'}
LDA_PZ	{'XC_LDA_X','XC_LDA_C_PZ'}

LDA_VWN	{'XC_LDA_X','XC_LDA_C_VWN'}
GGA_XPBE	{'XC_GGA_X_XPBE','GGA_C_XPBE'}
GGA_PW91	{'XC_GGA_X_PW91','XC_GGA_C_PW91'}
GGA_PBE_SOL	{'XC_GGA_X_PBE_SOL','GGA_C_PBE_SOL'}
GGA_B86	{'XC_GGA_X_B86','XC_GGA_C_PBE'}
GGA_B88	{'XC_GGA_X_B88','XC_GGA_C_PBE'}
GGA_OPTB88_VDW	{'XC_GGA_X_OPTB88_VDW','XC_GGA_C_PBE'}
GGA_OPTPBE_VDW	{'XC_GGA_X_OPTPBE_VDW','XC_GGA_C_PBE'}
HYB_GGA_B3LYP	{'XC_HYB_GGA_XC_B3LYP'}
HYB_GGA_HSE03	{'XC_HYB_GGA_XC_HSE03'}
HYB_GGA_HSE06	{'XC_HYB_GGA_XC_HSE06'}
HYB_GGA_PBEH	{'XC_HYB_GGA_XC_PBEH'}
MGGA_TPSS	{'XC_MGGA_X_TPSS','XC_MGGA_C_TPSS'}
MGGA_SCAN	{'XC_MGGA_X_SCAN','XC_MGGA_C_SCAN'}
MGGA_BR89	{'MGGA_X_BR89','XC_LDA_C_PW'}
MGGA_BJ06	{'MGGA_X_BJ06','XC_LDA_C_PW'}
MGGA_TB09	{'MGGA_X_TB09','XC_LDA_C_PW'}

11. 问题：计算能带时出现 **Error using syn2kpt Undefined K-point** 错误？

解答：这是因为这个结构的晶格 RESCU 无法识别或者您给出的晶格常数 RESCU 无法识别，这种情况下可以使用高对称点坐标的方法来定义 kpoint.symoints 参数，详见 1.2.3 MoS₂ 体系能带计算章节。

4.3 RESCU 并行版计算相关问题

12. 问题：在服务器上使用 RESCU 进行并行计算和单机版有区别吗？

解答：使用 RESCU 进行并行计算时需要在输入文件中开启 smi.status=1 和 mpi.status=1 两个参数，或者在脚本中使用 rescu --smi -i scf.input 去执行。下面是直接用 mpirun 并行和 pbs 并行的例子：

13. 问题：如何在服务器上通过 mpirun 使用 RESCU 进行并行计算？

解答：RESCU 在服务器上分两种，一种为没编译版，该版在 MATLAB 环境下运行；另一种为编译版，该版本脱离可直接在 linux 环境下运行。这两个版本的运行命令有所区别。

下面是直接用 mpirun 并行和 pbs 并行的例子：

1) 直接用 mpirun 并行，首先需要确保 RESCU 的环境变量是有效的。

没编译版：

```
#!/bin/sh  
  
source /home/lfruan/software/rescu/RESCUPackage/barc  
  
mpirun -np 6 matlab -nojvm -nosplash -nodisplay -singleCompThread -r  
"addpath(genpath('/home /rescuPackage/src'));rescu --smi -i scf.input"
```

其中 source 命令就是 RESCU 的环境变量，mpirun -np 6 表示用 6 个进程去运行，matlab -nojvm -nosplash -nodisplay -singleCompThread -r 是 matlab 运行并行版 RESCU 的固定格式，addpath(genpath('/home /rescuPackage/src'))表示添加 RESCU 安装路径，rescu --smi -i scf.input 表示使用并行计算 scf.input 文件。

编译版：

```
#!/bin/sh  
  
source /home/lfruan/software/rescu/RESCUPackage/barc  
  
mpirun -np 6 rescu --smi -i scf.input
```

2) 通过 pbs 进行任务提交的例子：（LSF 及 slurm 可以做相应修改自行设置）

没编译版：

```
#!/bin/sh  
  
#PBS -N rescu  
  
#PBS -q blade  
  
#PBS -l nodes=1:ppn=16  
  
#PBS -l walltime=400:00:00  
  
NCPU=`wc -l < $PBS_NODEFILE`  
  
WORKDIR=$PBS_O_WORKDIR  
  
cd ${WORKDIR}  
  
source /public/software/apps/RESCU/barc
```

```
mpirun -hostfile $PBS_NODEFILE -np 16 matlab -nojvm -nosplash -nodisplay  
-singleCompThread -r "addpath(genpath('/public/rescu-2.0.4/src'));rescu --smi -i  
scf.input"
```

编译版:

```
#!/bin/sh
```

```
#PBS -N rescu
```

```
#PBS -q blade
```

```
#PBS -l nodes=1:ppn=16
```

```
#PBS -l walltime=400:00:00
```

```
NCPU=`wc -l < $PBS_NODEFILE`
```

```
WORKDIR=$PBS_O_WORKDIR
```

```
cd ${WORKDIR}
```

```
source /public/software/apps/RESCU/barc
```

```
mpirun -hostfile $PBS_NODEFILE -np 16 rescu --smi -i scf.input
```