

数据处理工具包
Data Processing Toolkit
Version 0.8.3

梁汉普

2020/10/28

目录

1	声明	2
1.1	Licence	2
2	安装	2
2.1	依赖环境	2
2.2	获取软件安装包	2
2.3	安装软件	2
3	命令概述	4
3.1	DPT -b处理能带数据	4
3.2	DPT -d处理能态密度数据	5
3.3	DPT -f得到投影能带数据	6
3.4	DPT --bader得到Bader电荷数据	7
3.5	DPT --diele得到Born电荷与介电矩阵	7
3.6	DPT --elastic得到弹性模量及其相关参数	8
3.7	DPT -s对结构依次施加一定大小应变	8
3.7.1	生成应变的结构文件	8
3.7.2	处理计算好的应变数据	10
3.8	DPT -v生成VASP计算用文件	11
3.9	DPT --vdW在INCAR中添加van der Waals修正	11
4	DPT文件夹内包含内容	12

1 声明

Data Processing Toolkit (DPT)是一个用来处理计算结束后数据的一个工具包。目前版本为测试版0.8.3, 尚未成为正式版, 因此如有出现bug、或者想要添加某些功能, 可以联系邮箱 hanpu-liang@cumt.edu.cn进行反馈。

1.1 Licence

MIT License

Copyright (c) 2019 Hanpu Liang

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

2 安装

2.1 依赖环境

为了能够让本软件顺利的运行下去, 你需要拥有以下必要的软件/库:

- Shell的版本为bash。
- python 2.7及以上。并且拥有numpy库, 后续版本中如果需要可视化作图, 则还会需要matplotlib库。

2.2 获取软件安装包

2.3 安装软件

在获取了安装包之后, 在终端命令行中输入解压命令, 将压缩包解压

```
$ tar -zxvf DPT-0.8.3.tar.gz
```

压缩包解压后会生成一个文件夹 DPT-0.8.3/, 这个文件夹中包含了所有的DPT的安装文件与运行文件。然后进入文件夹并修改文件install.sh中的路径参数。

```
$ cd DPT-0.8.3
$ vim install.sh
```

在文件中存在以下路径需要修改:

```
#====You can set this parameters by yourself====#
usr='whoami'
BIN=/home/$usr/bin           # the path you want create the DPT link
MPI_PATH=mpirun              # the path of mpirun
VASP_STD_PATH=vasp_std        # the path of vasp_std
VASP_RELAX_AB_PATH=vasp_relax_ab # the path of vasp_std for 2D materials
VASP_GAM_PATH=vasp_gam        # the path of vasp_gam
POTCAR_PATH=/home/$usr/POTCAR/PBE # the path of POTCAR files
#====End of parameters setting area====#
```

修改完成后, 就可以开始安装, 运行install.sh文件

```
$ sh install.sh
```

如果没有报错的成功执行完了脚本的内容, 重新开一个终端窗口, 就可以尝试运行本软件了。

```
*-----*
| Data      | D |
| Processing| P |
| Toolkit   | T |
*-----*
Version: 0.8.1
Author: Liang Hanpu

=====
[DPT] - Installing DPT ...
[DPT] - Python found at : /usr/bin/python
=====

=====
[DPT] - DPT is installed successfully!
=====

[DPT] - You can check DPT by "DPT -h"
=====

[DPT] - Enjoy DPT!
```

可以通过输入

```
$ DPT -h
```

来得到DPT的指令表。

```
Usage: DPT [options]

Options:
  -h, --help      show this help message and exit
  -b, --band       calculate the energy band and save it into DPT.BAND.dat
```

```

-d, --dos      calculate the density of states and save it into ATOM_dos.out
-f, --fat      calculate the projected band and save it into all_orbits/
-c, --column   save all files into band_data.out or fat_band_ATOM.out and one
               data one column
--bader        interface of VASP and bader analysis scripts
--diele        contract the born effective charges and dielectric tensor into
               files
--elastic      contract elastic moduli matrix from OUTCAR and calculate each
               elastic modulus
-s, --strain   Strain the lattice vectors in POSCAR, direction: x, y, z, xy,
               xyz, a, b, c
--te           Create the jobs for the calculation of thermal electric
               properties. 'files' for preperation files, 'rel' for
               relaxation, 'band' for electronic structure and 'cal' for data
               processing.
-v, --vasp     Create the input files for VASP. Enter 'DPT -v' to know the
               detail parameters.
--vdW          Add the vdW correction into the INCARs.

```

3 命令概述

3.1 DPT -b处理能带数据

处理能带数据需要文件

EIGENVAL, OUTCAR

所使用的命令为

```
$ DPT -b
```

运行该命令后, 当前目录下会生成文件

DPT.BAND.dat

如果计算的能带考虑了电子自旋, 即在INCAR中加了参数ISPIN=2, 那么会生成两个文件

DPT.BAND.UP.dat, DPT.BAND.DOWN.dat

其中DPT.BAND.UP.dat与DPT.BAND.DOWN.dat分别代表自旋向上与自旋向下的能带。

生成的这三个文件格式相同, 如下面的示例所示, 第一列为布里渊区路径, 往后所有列为能带, 每一列代表一条能带。

```

# DPT.BAND.dat
2.3057055647 -20.6961260000 -20.6888790000 -19.6399690000 .....
2.3238657920 -20.6961180000 -20.6888920000 -19.6399890000 .....
2.3420260194 -20.6960970000 -20.6889570000 -19.6399720000 .....
2.3601862467 -20.6960660000 -20.6890350000 -19.6399190000 .....
2.3783464741 -20.6959950000 -20.6891380000 -19.6398660000 .....
.....
.....

```

在生成文件的同时, 屏幕上会输出下列内容, 即本次能带处理时的一些基础参数。

```
[DPT] - Reading EIGENVAL...
[DPT] - Reading OUTCAR...
[DPT] - Basic electronic structure parameters
[ Electrons number :      144 ]
[   K points number :      80 ]
[       Band number :     112 ]

[DPT] - VBM, CBM and Gap (unit eV)
[       E-fermi : -4.222200 ]
[       CBM : -1.627646 ]
[       VBM : -4.369111 ]
[       Gap : 2.741465 ]

[DPT] - Fermi Energy Be Set to VBM!
[DPT] - DPT.BAND.dat saved!
```

其中, Electrons number是结构中的电子数目, K points number是布里渊区路径的采样点数量, Band number是能带数目。所以实际上, DPT.BAND.dat文件里面的矩阵大小可以表示为K points number \times (Band number+1)。

E-fermi为费米能级对应的能量值, CBM与VBM分别对应导带底和价带顶。Gap就是导带底与价带顶之间的差值, 即带隙。值得注意的是, 费米能级的定义是“价电子填充的最后一个能级”, 通常意义上指的就是VBM。但是VASP算出来的VBM和费米能级很少会完全对上, 这是因为计算出来的费米能级是通过Fermi-Dirac分布在概率等于1/2时取得值, 因此与VBM对应不上。所以在输出的DPT.BAND.dat文件中, 已经将所有能带减去了VBM的能量大小, 考虑在能带图中将VBM设为0, 而不是费米能级, 但是意义上仍然是以0为费米能级。

3.2 DPT -d处理能态密度数据

处理能态密度数据需要的文件有

POSCAR, DOSCAR

所使用的命令为

\$ DPT -d

该命令会生成文件所有原子的态密度的数据文件ATOM_dos.dat。如果一共有3个原子, 则有3个文件。以GaN-ZnO合金为例, 输入命令后产生的信息为

```
[DPT] - Reading POSCAR...
[DPT] - Atom [ Zn] number: [   4]
[DPT] - Atom [ N] number: [   4]
[DPT] - Atom [ Ga] number: [   4]
[DPT] - Atom [ O] number: [   4]
[DPT] - Reading DOSCAR...
[DPT] - DPT.TOTAL_dos.dat saved!
[DPT] - DPT.Zn_dos.dat saved!
[DPT] - DPT.O_dos.dat saved!
[DPT] - DPT.Ga_dos.dat saved!
[DPT] - DPT.N_dos.dat saved!
```

然后在目录中多出了文件

DPT.TOTAL_dos.dat, DPT.Zn_dos.dat, DPT.O_dos.dat, DPT.Ga_dos.dat
 , DPT.N_dos.dat

首先是DPT.TOTAL_dos.dat文件, 其第一列为能量, 第二列为总态密度。

```
# DPT.TOTAL_dos.dat
      E      TOTAL_DOS
-23.663800000  0.000000000
-23.555800000  0.000000000
-23.446800000  0.000000000
-23.338800000  0.000000000
.....
```

然后是对于DPT.N_dos.dat文件, 是N原子每个轨道的态密度。如, 第一列为能量, 第二列为N原子的总态密度, 第三列为N原子的s轨道的态密度。

```
# DPT.N_dos.dat
E  N_tot  N_s  N_p  N_d  N_py  N_pz  N_px  N_dxy  N_dyz  N_dz2  N_dxz  N_dx2
-23.66380  0.00000  0.00000  0.00000  0.00000  0.00000  .....
-23.55580  0.00000  0.00000  0.00000  0.00000  0.00000  .....
-23.44680  0.00000  0.00000  0.00000  0.00000  0.00000  .....
-23.33880  0.00000  0.00000  0.00000  0.00000  0.00000  .....
.....
```

3.3 DPT -f得到投影能带数据

处理能态密度数据需要的文件有

POSCAR, PROCAR

所使用的命令为

```
$ DPT -f
```

该命令会生成一个目录

all_orbits/

这个目录中存放着所有原子的所有轨道对能带的贡献大小。以CN结构举例, 输入命令后会输出以下信息。

```
[DPT] - Reading EIGENVAL...
[DPT] - Reading OUTCAR...
[DPT] - Reading PROCAR...
[DPT] - all_orbits/ created!
[DPT] - Reading POSCAR...
[DPT] - Atom [ C ] number: [ 12]
[DPT] - Atom [ N ] number: [ 2]
[DPT] - Atom C's_orbits_file_saved!
[DPT] - Atom N's_orbits_file_saved!
```

在all_orbits/中有以下文件

C_dx2.dat	C_dxz.dat	C_dz2.dat	C_py.dat	C_s.dat
C_dxy.dat	C_dyz.dat	C_px.dat	C_pz.dat	C_tot.dat
N_dx2.dat	N_dxz.dat	N_dz2.dat	N_py.dat	N_s.dat
N_dxy.dat	N_dyz.dat	N_px.dat	N_pz.dat	N_tot.dat

其中C_s.dat代表着C原子的s轨道对能带的贡献。其中文件大小与能带文件DPT.BAND.dat中能带部分相同。假如DPT.BAND.dat大小为60行51列(即 60×51)，那么能带部分大小为 60×50 ，因此C_s.dat大小也为 60×50 。文件中每个点代表着C_s轨道对“当前能带位置”的贡献，用来控制作图时每个点的大小。

3.4 DPT --bader得到Bader电荷数据

处理能态密度数据需要的文件有

```
AECCAR0, AECCAR1, AECCAR2
```

这三个文件通过在INCAR中加入参数

```
LARCHG = .TRUE.
```

来得到。

在已经计算好Bader电荷的目录中输入

```
$ DPT --bader
```

就可以得到Bader电荷的处理数据。

注：1. 计算Bader电荷需要在才会生成AECCAR等文件。2. Bader电荷分析调用的是Wenjie Tang等人脚本chgsum.pl，如需详细信息，可以去Bader Charge Analysis网站: <http://theory.cm.utexas.edu/henkelman/bader/>进行进一步了解。

3.5 DPT --diele得到Born电荷与介电矩阵

Born有效电荷数据与Delectric矩阵在OUTCAR中，DPT的这个命令可以将这两个数据从OUTCAR中提取出来。需要文件

```
OUTCAR
```

然后输入指令

```
$ DPT --diele
```

就可以得到输出数据

```
[DPT] - Dielectric tensor save into dielectric.dat successful!
[DPT] - Born effective charges save into born_chagres.dat successful!
```

也就是生成了两个文件

```
dielectric.dat, born_chagres.dat
```

以GaN计算出来的文件举例，两个文件内容分别为


```
# born_charges.dat
BORN EFFECTIVE CHARGES (in e, cumulative output)
-----
ion    1
1      3.07023    -0.00118    0.00000
2      0.00000     3.07104    0.00000
3      0.00000     0.00000    0.33540
ion    2
1     -3.06602     0.01227    0.00000
2      0.00000    -3.07319    0.00000
3     -0.00000     0.00000   -0.33549
# dielectric.dat
1.849004    0.000000    0.000000
-0.000000    1.849004    0.000000
-0.000000    0.000000    1.146479
```

3.6 DPT --elastic得到弹性模量及其相关参数

弹性模量矩阵存在OUTCAR中，因此DPT的作用是将矩阵提取出来，并进行处理，得到杨氏模量(Young's modulus)，泊松率(Poisson's ratios)和剪切模量(Shear moduli)。通过输入

```
$ DPT --elastic
```

可以得到输出

```
[DPT] - This structure is mechanically stable!
[DPT] - Elastic Constant C_ij (unit N/m)
58.350860    23.220090    0.207650   -0.000000    0.000000    0.000000
23.220090    82.593840    0.200830   -0.000000   -0.000000   -0.000000
0.207650     0.200830    0.837670   -0.000000    0.000000    0.000000
-0.000000   -0.000000   -0.000000    1.884750    0.000000   -0.000000
0.000000   -0.000000    0.000000   -0.000000    0.211900   -0.000000
0.000000   -0.000000    0.000000   -0.000000   -0.000000    7.810810
[DPT] - Young's modulus E, Poisson's ratios v, and shear moduli G (unit N/m)
E_x      E_y      v_xy      v_yx      G_xy
51.822860  73.353658    0.281136    0.397939    7.810810
[DPT] Files DPT.elastic_constant.dat and DPT.modulous.dat have been created!
```

这两项输出结果被存放在DPT.elastic_constant.dat与DPT.modulous.dat这两个文件中。

3.7 DPT -s对结构依次施加一定大小应变

3.7.1 生成应变的结构文件

首先需要文件

```
POSCAR
```

然后应变部分需要指定参数：

```
$ DPT -s [direction] [begin] [end] [step] [type]
```

其中[direction]参数可以选择: x, y, z, xy, xyz, a, b, c共7种。指的是可以对着7个方向依次进行施加应力。

begin, end和step三个参数是用来控制生成应力的大小与间隔。begin指的是应变起始值, end是应变最大值, step是间隔值。也就是从begin开始, 每隔step大小就取一个点算应变, 直到end结束。

type是施加应变的类型, 可以取pos或者neg。应变存在“拉伸”和“压缩”两种情况。如果取“拉伸”, 即type取pos, 那么就会将晶胞拉长, 则begin这三个参数为正值。如果取“压缩”, 即type为neg, 那么就会将晶胞压缩, 则begin这三个参数在处理过程中为负值。【注意】: 输入的begin, end和step应始终为正, pos和neg影响到的只是处理过程中的正负。

例, 我们要对结构的x轴施加从0.0(也就是0%)开始, 到0.2(也就是20%)结束, 每隔0.01(即1%)取一个点的应变, 类型为拉伸。那么输入的参数为

```
$ DPT -s x 0.0 0.2 0.01 pos
```

此时屏幕上会输出

```
[DPT] - Reading POSCAR...
[DPT] - Straining Parameters:
  Strain direction      : x
  Initial strain length : 0.000
  End strain length     : 0.200
  Strain step           : 0.010
  Total strain number   :    21
[DPT] - Strain Finish.
```

上面显示的就是我们输入的参数。其中Total strain number就是生成的结构总数量, 从0.0到0.2, 间隔为0.01, 总共会生成21个结构。生成的文件以DPT-strain.POSCAR.XXXX的形式放在当前目录中。

如果输入

```
$ DPT -s x 0.0 0.2 0.01 neg
```

那么就会输出

```
[DPT] - Reading POSCAR...
[DPT] - Straining Parameters:
  Strain direction      : x
  Initial strain length : -0.000
  End strain length     : -0.200
  Strain step           : -0.010
  Total strain number   :    21
[DPT] - Strain Finish.
```

可以看到应变的值从正变成了负数, 也就是压缩结构。

准备好计算必须的INCAR等文件后, 就可以提交任务进行计算。这里推荐使用提交任务的shell脚本为:

```
# job.pbs
for i in {0000..0020}
do
  mkdir job-$i
```

```

cp INCAR.* KPOINTS POTCAR job-$i/
cp DPT.strain.POSCAR.$i job-$i/POSCAR
cd job-$i/
cp INCAR_rel INCAR
mpirun -machinefile $PBS_NODEFILE -np $NP vasp_std >>out1.vasp 2>>err.vasp
cp POSCAR POSCAR_ori
cp CONTCAR POSCAR
cp OUTCAR OUTCAR1
cp INCAR_scf INCAR
mpirun -machinefile $PBS_NODEFILE -np $NP vasp_std >>out2.vasp 2>>err.vasp
cp OUTCAR OUTCAR2
cd ..
done

```

3.7.2 处理计算好的应变数据

DPT会读取计算好后的所有job-XXXX/目录中的OUTCAR文件。因此如果缺少目录，或者缺少文件，DPT就会报错。

输入

```
$ DPT -s plot
```

DPT就会从system.log文件中读取之前生成的信息，从而读取每个目录中的结果。屏幕上会输出

```
[DPT] - File DPT.stress-strain.dat created!
```

在文件DPT.stress-strain.dat中，存放的是应变大小与及其所有方向(共6个方向)的应力。第一列为应变大小，往后六列为应力。

```

# DPT.stress-strain.dat
Strain    F_xx    F_yy    F_zz    F_xy    F_yz    F_xz
0.000     0.544   3.058   -0.209   0.000   0.000   0.000
0.010    -11.352  0.495   -0.206   0.000   0.000   0.000
0.020    -22.411 -1.857   -0.203   0.000   0.000   0.000
0.030    -33.002 -47757   -0.756   0.000   0.000   0.000
... ..
... ..

```

一般来说，对着哪个轴施加应变，就应该取对应方向的应力来作图(该方向上的应力一定最大)。

【注意1】：这里输出的应力单位为kB，因此我们需要换算成为N/m。对于二维材料，可以通过

$$1\text{N/m} = 1\text{kB} \times 0.01 \times \text{厚度}$$

来换算。其中厚度就是z轴高度。

【注意2】：当施加的应变为拉伸时，那么输出的应力为负值，所以需要再加一个将应力大小变为正。如果应变为压缩，那么则不需要加负号。

3.8 DPT -v生成VASP计算用文件

DPT内部存储了一些常用的VASP计算用的INCAR, KPOINTS与任务脚本job.pbs文件。可以通过-v参数来复制到当前目录下。通过输入

```
$ DPT -v
```

就会在屏幕上打印出所有的可生成内容：

```
$ DPT -v [type] [method]
type:
  rel      : Relaxation
  band     : Electronic structure
  diele    : Born effective charge and dielectric matrix
  absorp   : Absorption coefficient
  elastic  : Elastic constants
  ELF      : Electron localization function
  PCD      : Partial charge density
  force    : Force Constants (Phonon Spectrum)
  POTCAR   : Get POTCAR corresponding atom type of POSCAR
method:
  energy:  only for 'rel' type. This INCAR only calculate energy
  2d      : only for 'rel' type. Modify vasp_std to vasp_relax_ab
  gam     : only for 'rel' type. Modify vasp_std to vasp_gam
  PBE     : only for 'band' type. Calculate the electronic structure by PBE method.
  HSE     : only for 'band' type. Calculate the electronic structure by HSE06 method.
```

【例1】生成利用HSE06方法计算能带的文件。

```
$ DPT -v band HSE
```

输入结束后，当前目录下会多出6个文件，即

```
INCAR_1, INCAR_2, INCAR_3, KPOINTS_1, KPOINTS_3, job-hse.pbs
```

其中KPOINTS_3为布里渊区路径文件，不同晶系的布里渊区路径不同，可以通过计算PBE能带来获取。相同晶系的布里渊区相同，则KPOINTS_3可以互通。文件中默认存放的是二维材料六方晶系的KPOINTS_3文件。PBE方法得到的KPOINTS_2默认存放的同样是二维材料六方晶系布里渊区路径。

【例2】生成光吸收系数的文件。

```
$ DPT -v absorp
```

输入结束后，当前目录下会多出5个文件，即

```
INCAR_1, INCAR_2, INCAR_3, KPOINTS, job-hse.pbs
```

【注意】job文件中mpirun与vasp的路径需要在安装DPT时在install.sh文件中修改。如果没有修改的，可以修改后重新sh install.sh。

3.9 DPT --vdW在INCAR中添加van der Waals修正

有时候二维材料需要添加vdW修正，但是手动的一个个往INCAR里面添加参数很麻烦。所以--vdW参数可以自动在INCAR中添加对应的vdW修正的参数。

一共可以添加5种vdW修正，分别是：optB88, optB86b, optPBE, vdW-DF和vdW-DF2。如果在-- vdW后不加参数，则默认为optB88修正。

【例1】-- vdW后不加参数

```
$ DPT --vdW
```

如果当前目录下有INCAR文件的话，就会输出

```
[DPT] - INCAR_1 is corrected by vdW. Method: optB88
[DPT] - INCAR_3 is corrected by vdW. Method: optB88
[DPT] - INCAR_2 is corrected by vdW. Method: optB88
```

【例2】添加optB86b修正

```
$ DPT --vdW optB86b
```

如果当前目录下有INCAR文件的话，就会输出

```
[DPT] - INCAR_1 is corrected by vdW. Method: optB86b
[DPT] - INCAR_3 is corrected by vdW. Method: optB86b
[DPT] - INCAR_2 is corrected by vdW. Method: optB86b
```

【注意】如果INCAR中已经存在某种修正的参数了，那么就会跳过这个INCAR。为了防止所有INCAR内vdW修正不同的情况出现，请尽量避免添加修正时，INCAR里面已经存在修正这种情况。

4 DPT文件夹内包含内容

- src (源代码)
- doc (用户使用手册)
- lib (引用已有的脚本库)
- example (运行样例-介电常数/电子结构/应力)
- LICENCE.txt (版权声明)
- install.sh (安装脚本)
- DPT (启动指令)