**Vasp流程**

林子越

## 一、准备文件

准备赝势文件：POTCAR，一般在当前服务器的vasp安装文件夹下有，如6001-ddf6/vasppp下即有vasp赝势包，注意一般选择PBE赝势，不然后面步骤可能做不出来（注：以F为例，F即是普通赝势，F\_s是软赝势，F\_h是硬赝势，一般选普通赝势即可，另外sv和pv是考虑不同价电子的赝势，一般考虑的越多越准，具体选择依照vasp中文手册）合成赝势命令，cat POTCAR-1 POTCAR-2 POTCAR-3 >> POTCAR

例如物质W4S8，含有12个原子，但合成赝势的时候只需将W的赝势和S的赝势这两个赝势合成为一个，即cat POTCAR-W POTCAR-S >> POTCAR，

需要注意最终的赝势文件名只能是POTCAR.

准备POSCAR：将cell文件导入Vesta转换为vasp文件并重命名为POSCAR

(一般在这一步就先用MS把结构加上对称性，因为可以化成更小的胞，便于计算。

如果加对称性变成大胞就先加对称性再转成原胞Build-Symmetry-Primitive Cell)

## 二、产生K点

在打包好的输入文件里拷贝a.out，createKpoints，prodKpts.f90

并将上一步准备的POSCAR放入同文件夹内

执行：chmod +x a.out

再执行：./a.out

即产生KPOINTS文件

## 三、结构优化

### （1）准备vasp.pbs脚本，POSCAR，POTCAR文件

### （2）提交vasp.pbs（提交命令见脚本及命令文档）

vasp.pbs

#!/bin/bash

#PBS -q CT1

#PBS -l nodes=1:ppn=12

#PBS -j oe

#PBS -V

#PBS -N VASP-opt

cd $PBS\_O\_WORKDIR

for i in 30000 #压力点，单位为kbar，1GPa=10kbar，可以

do #继续输入如1000 2000 3000即计算多压力点

cat > INCAR << ! #此处将INCAR整合到脚本内，如不整合，

PSTRESS = $i #可单独建一个INCAR文件，将下方蓝色内

ISTART = 0 #容输入

ICHARG = 2

IBRION = 2

NSW= 300

POTIM = 0.1

NELM = 60

EDIFF = 1.0E-7

EDIFFG = -0.005

ISMEAR = 1

SIGMA = 0.2

ISIF = 3

ENCUT = 550

PREC = Accurate

LWAVE = .FALSE.

LCHARG = .TURE.

NPAR = 4

LORBIT=11

#ISPIN= 2

GGA=PE

!

echo "a= $i" ;

KPOINT=/public/home/YHY/lzy/Vasp/Gen\_kpoints/a.out #此处有两种方案，第一种是像此脚本里使用路径直接生成K点，这样操作便不用执行上述第二步产生K点的过程，直接给出a.out的位置即可

$KPOINT #另一种就是在第二步产生K点，然后将KPOINTS文件拷贝至此文件夹内，这样操作可以把脚本里的这两行删掉

mpirun -np 12 /share/apps/compiler/VASP-5.3/vasp > log 2>&1 #此处取决于vasp在本机器上的安 装位置，可以看一下同机器上别人的脚本

E=`tail -1 OSZICAR` ; echo $i $E >>SUMMARY.dia

F=`grep ' enthalpy is TOTEN' OUTCAR | tail -1` ; echo $i $F >>zToten.dia

V=`grep 'volume of cell' OUTCAR | tail -1` ; echo $i $V >>zVolum.dia

W=`grep 'reached required accuracy - stopping' OUTCAR` ; echo $i $W >>zWork.dia

cp OUTCAR outcar\_$i

cp OSZICAR oszicar\_$i

cp CONTCAR CONTCAR\_$i

cp KPOINTS KPOINTS\_$i

cp CONTCAR POSCAR #注意此处已经将原POSCAR覆盖

done #如不想覆盖可删掉此行

cat SUMMARY.dia

cat zToten.dia

cat zVolum.dia

cat zWork.dia

### （3）最后得到的POSCAR即为优化出来的结构，用于之后的计算

## 四、静态自洽计算

### （1）准备 INCAR：

即定义 PREC，EDIFF，ENCUT，ISTART = 0（默认）,ISMEAR(对于半导体和绝缘体：ISMEAR= 0，SIGMA = 0.05； 对于金属：ISMEAR=1， SIGMA = 0.2)

SYSTEM =CoH

PREC=Accurate

EDIFF = 1e-5

ENCUT = 700

ISTART = 0 ； ISMEAR= 1 ; SIGMA = 0.2

### （2）准备好 POSCAR 文件，上一步优化得到的POSCAR即为优化过的

### （3）准备POTCAR 文件

### （4）KPOINTS 在脚本中给出产生路径或者将生成好的KPOINTS拷贝入文件夹内

### （5）提交脚本

### （6）记下 OUTCAR 中 E-fermi 的数值，下一步处理结果时会用到。

grep “E-fermi” OUTCAR

脚本文件：（此处使用已整合INCAR的脚本）

vasp.pbs

#!/bin/bash

#PBS -q CT1

#PBS -l nodes=1:ppn=12

#PBS -j oe

#PBS -V

#PBS -N vasp-scf

cd $PBS\_O\_WORKDIR

for i in 30000 #压力点，单位kbar

do

cat > INCAR << !

SYSTEM =F #INCAR内容

PREC=Accurate

EDIFF = 1e-5

ENCUT = 550

ISTART = 0

ISMEAR= 1

SIGMA = 0.2

!

echo "a= $i" ;

KPOINT=/public/home/YHY/lzy/Vasp/Gen\_kpoints/a.out

$KPOINT

mpirun -np 12 /share/apps/compiler/VASP-5.3/vasp > log 2>&1

E=`tail -1 OSZICAR` ; echo $i $E >>SUMMARY.dia

F=`grep ' enthalpy is TOTEN' OUTCAR | tail -1` ; echo $i $F >>zToten.dia

V=`grep 'volume of cell' OUTCAR | tail -1` ; echo $i $V >>zVolum.dia

W=`grep 'reached required accuracy - stopping' OUTCAR` ; echo $i $W >>zWork.dia

cp OUTCAR outcar\_$i

cp OSZICAR oszicar\_$i

cp CONTCAR CONTCAR\_$i

cp KPOINTS KPOINTS\_$i

cp CONTCAR POSCAR

done

cat SUMMARY.dia

cat zToten.dia

cat zVolum.dia

cat zWork.dia

## 五、非自洽计算（能带图）

### （1）准备 INCAR：

即定义 ENCUT， PREC，ISTART = 1，ICHARG = 11（表示从 CHGCAR 中读入电荷分布，并且在计算中保持不变），ISMEAR = 1，SIGMA = 0.2， LWAVE (对于半导体和绝缘体：ISMEAR= 0，SIGMA = 0.05； 对于金属：ISMEAR=1， SIGMA = 0.2) ，并增加 NBANDS

（默认值为 NELECT/2+NIONS/2，能带太少可以设的大一点，默认值x2或者x4，NELECT 和 NIONS 分别为电子数和离子数，可以在 OUTCAR文件中找到这两个参数 grep ''NIONS'' OUTCAR && grep ''NELECT'' OUTCAR）

SYSTEM=CoH

ENCUT=700

PREC=Accurate

LWAVE=.FALSE.

ISTART=1

ICHARG=11

ISMEAR=1

SIGMA=0.2

NBANDS=8

NELECT=10

### （2）POSCAR、POTCAR 同上一步自洽计算。

### （3）将上一步自洽计算得到的 CHG、CHGCAR、WAVECAR 拷贝至同一目录下

### （4）建立syml文件

5 # 沿高对称线的K点，MS→tools→Brillouin Zone Path。就是指定 VASP

20 20 20 20 # 计算沿着 X 到R点再到M、G回到R点，每个方向上各取 20 个 K 点。

X 0.500 0.000 0.000

R 0.500 0.500 0.500

M 0.500 0.500 0.000

G 0.000 0.000 0.000

R 0.500 0.500 0.500 #上一步自洽中的outcar中正、倒格子基矢

0.000000000 1.840208159 1.840208159 0.271708392 0.271708392 0.271708392

1.840208159 0.000000000 1.840208159 0.271708392 -0.271708392 0.271708392

1.840208159 1.840208159 0.000000000 0.271708392 0.271708392 -0.271708392

-20 50 #能量范围

5.9036 #费米能grep ''E-fermi'' 自洽OUTCAR中得到

### （5）建立另一文件夹Genq\_points(产生KPOINS)，同QE流程

#### i）需要准备的文件：

~

a.out （执行文件），

band

elph

GenKpaths.f90

inp.kpt

nestVects

phonon

qpoints

syml

（文件具体内容见群里发的例子）

#### ii) 首先将优化后加完对称性的POSCAR导入MS查看布里渊区高对称点

（具体操作见 CASTEP操作流程中相关操作）

将以下内容输入syml文件

5 #高对称点个数

20 20 20 20 #在5个高对称点的4个间隔中每个插入20个点

X 0.500 0.000 0.000 #高对称点路径

R 0.500 0.500 0.500

M 0.500 0.500 0.000

G 0.000 0.000 0.000

R 0.500 0.500 0.500

#### iii) chmod +x a.out

**./a.out**

执行完毕后会在inp.kpt文件中输出具体k/q点

下面建立KPOINTS文件:

k-points along high symmetry lines #这里一定要输入K点个数，输入多少就会取下方的多

141 少K点

Reciprocal

0.000000 0.000000 0.000000 1.00 #这里粘贴inp.kpt中生成的K点

0.000000 0.000000 0.025000 1.00

0.000000 0.000000 0.050000 1.00

0.000000 0.000000 0.075000 1.00

……

-0.283050 0.641950 0.075000 1.00

-0.299700 0.650300 0.050000 1.00

-0.316350 0.658650 0.025000 1.00

-0.333000 0.667000 -0.000000 1.00

### （6）提交运行

vasp.pbs

#!/bin/bash

#PBS -q CT1

#PBS -l nodes=1:ppn=12

#PBS -j oe

#PBS -V

#PBS -N vasp-band

# go to work dir

cd $PBS\_O\_WORKDIR #不同服务器路径不同

export PW\_ROOT=/share/apps/compiler/VASP-5.3/

mpirun -n 12 $PW\_ROOT/vasp > log

（7）./a.out 产生bnd.dat导入Origin作图。

***Notes:***

WARNING: small aliasing (wrap around) errors must be expected

能带态密度里报这个错误需要

INCAR里面提高截断能，到700-800一般即可

高对称点在横坐标位置highk.dat文件中

## 六、非自洽计算（态密度图）

### （1）准备 INCAR：

即定义 ENCUT， PREC，ISTART = 1，ICHARG = 11（表示从 CHGCAR 中读入电荷分布，并且在计算中保持不变），ISMEAR = -5，LORBIT = 11，（或LORBIT=10，后续脚本使用split.sh） EDIFE, LWAVE, ALGO。

SYSTEM=CoH

ISTART=1

ICHARG=11

EDIFE=1.0E-5

ISMEAR=-5

LORBIT=11

LWAVE=.TRUE.

ALGO=Fast

ENCUT=700

PREC=Accurate

NEDOS=3001 （默认值是301，但如果不够平滑，建议设大一点）

### （2）KPOINS可使用上面能带图中的结果POSCAR、POTCAR

同自洽计算，不要用带具体K点的，除了能带图都不需要具体K点，phonopy做声子也不需要

### （3）将上一步自洽计算得到的 CHG、CHGCAR、WAVECAR 拷贝至同一目录下。

### （4）提交运行（脚本与之前能带图相同）

### （5）./a.out 或./split.sh (产生DOS\_\*)导入Origin作图

注意：Dos\_total.dat 的第一列数据是能量值，单位为 eV；第二列数据是总态密度的值，单位 State/eV.unit cell；第三列数据是总态密度的积分值，也就是电子数，单位为 electrons (用 origin 等软件绘图时把dos\_total.dat 的最后一列删除即可) 。Dos\_.atom\* 是相应原子的分波态密度值，其中的第一列数据是能量值，单位为 eV；第二、

三、四列数据分别对应于 s、p、d 态的分波态密度值，单位为 State/eV.atom。

## 七、计算bader电荷

### （1）准备INCAR文件

SYSTEM=CoH

ISTART=0

ICHARG=2

EDIFF=1.0E-5

ISMEAR=1

SIGMA=0.2

ENCUT=700

PREC=Accurate

LWAVE=.FALSE.

NGXF=192 #在对应压力下的outcar\_\*中的值×4

NGYF=192

NGZF=192

LAECHG=.TRUE.

### （2）准备好 POSCAR 文件（结构优化产生）

### （3）POTCAR 直接从赝势库中得到

### （4）准备好bader、chgsum.pl、vasp.pbs文件

### （5）提交脚本vasp.pbs，脚本同上

chmod +x bader chgsum.pl

./chgsum.pl AECCAR0 AECCAR2

./bader CHGCAR -ref CHGCAR\_sum

vi ACF.dat （分析得失电子）观察Charge变化

## 八、计算电子局域函数ELF

### （1）准备INCAR文件

SYSTEM = CoH

PREC = Accurate

ENCUT = 700

EDIFF = 1e-6

ISMEAR = 1

SIGMA = 0.2

LWAVE = FALSE

LCHARG = TRUE

LELF = TRUE

NGX=48 #在对应压力下的outcar\_\*中

NGY=48

NGZ=48

NPAR = 4

### （2）准备POSCAR文件，注意这里POSCAR里的元素名称要删掉。

### （3）准备POTCAR和vasp.pbs文件（脚本同上）

### （4）提交vasp.pbs运行

### （5）计算完后的ELFCAR文件加上元素名称导出，用VESTA打开→Utilities→2D Data Display（Max：1 Min：0）→Slice

注意：在进行扩胞计算的时候只需要把POSCAR替换成扩胞后的POSCAR文件（把要扩胞的POSCAR导入MS→Build→Symmetry→Supercell→设置a，b，c（a，b，c的定义在Symmetry→Redefine Lattice））

1 1 1面扩展方向的时候，要向负方向扩展即a,b,c均设为-1到1

## 九、结合phonopy计算声子谱及其态密度

### （1）准备INPHON-in

ATOM\_NAME= F #原子名称

DIM=1 1 1 #扩胞每方向倍数，每个方向都扩2

PM=.TRUE. #倍胞即为2 2 2

DIAG=.FALSE.

#LSUPER = .TRUE. #超胞方法，取消#生效

### （2）准备INPHON-band

ATOM\_NAME=F

DIM=1 1 1

PM=.TRUE.

DIAG=.FALSE.

ND=4

FORCE\_CONSTANTS=WRITE

NPOINTS=51 #画图K点个数，按需给即可

BAND= 0.500 0.000 0.000 0.500 0.500 0.500 0.500 0.500 0.000 0.000 0.000 0.000 0.500 0.500 0.500 #高对称点按此格式依次输入即可

### （3）准备INPHON-dos

ATOM\_NAME= F

DIM= 1 1 1

PM=.TRUE.

DIAG=.FALSE.

DOS\_RANGE=-20 90 0.1 #DOS图范围-20到90，精度0.1，

FORCE\_CONSTANTS=WRITE #范围取决于声子谱频率

MP = 9 9 9

SIGMA=0.5

#PDOS= 1 2 3 4, 5 #画分立态密度图去掉#号

#TPROP=.TRUE

### （4）准备POSCAR（优化得到）POTCAR，KPOINTS（均与之前相同）

phonopy --symmetry --tolerance=0.01|head；确定结构对称性，tolerance代表精度

将加了对称性的PPOSCAR改为POSCAR并备份

phonopy -d INPHON-in

输入完这一步后会产生一系列POSCAR文件，需要看POSCAR的后缀最大是多少

从而设置INCAR里面

### （5）提交vasp.pbs

vasp.pbs

#!/bin/bash

#PBS -q CT1

#PBS -l nodes=1:ppn=12

#PBS -j oe

#PBS -V

#PBS -N vasp-phonon

cd $PBS\_O\_WORKDIR

for i in 001 002 003 004 005 006 #这里的个数取决于POSCAR的最大后缀，不一致之后产生力文件会报错

do

cp POSCAR-$i POSCAR

cat > INCAR << !

SYSTEM = Phon

ISTART = 0

ICHARG = 2

INIWAV = 1

NELMIN = 2

EDIFF =1.0E-5

EDIFFG = -0.005

NSW = 0

IBRION = 8

ISMEAR = 0

SIGMA = 0.05

ADDGRID = .TRUE.

ENCUT =550

PREC = Accurate

IALGO = 38

LREAL = .FALSE.

LWAVE = .FALSE.

LCHARG = .FALSE.

NPAR = 4

!

echo "a= $i" ;

export PW\_ROOT=/share/apps/compiler/VASP-5.3/

mpirun -n 12 $PW\_ROOT/vasp > log

cp OUTCAR outcar\_$i

cp OSZICAR oszicar\_$i

cp vasprun.xml vasprun.xml-$i

done

phonopy -f vasprun.xml-00\* 生成力文件，(注意如果十个以上最后是0\*)

### （6）将备份的POSCAR拷贝覆盖本文件夹内的POSCAR

phonopy INPHON-band，生成矩阵文件

phonopy-bandplot --gnuplot band.yaml > band.dat 得到声子谱文件band.dat

phonopy --dos INPHON-dos 得到声子态密度 total\_dos.dat

***Notes:***

log文件中产生

WARNING: small aliasing (wrap around) errors must be expected

提高截断能到700-800即可

扩胞之后如果算不出来oszicar记得减小k点网格

## 十、计算COHP

### （1）准备POSCAR、POTCAR、KPOINTS

### （2）准备INCAR

PREC = Accurate

LREAL = .FALSE.

ISMEAR = -5

NSW = 0 ; ISIF = 0

EDIFF = 1.0d-7

EDIFFG = -1.0d-6

ENCUT = 900

ISYM=-1

NBANDS =58 （可以设的大一点，lobster会自动忽略不需要的能带）

NPAR = 4

NEDOS = 801 （默认值301，可加大提高精度）

LORBIT = 12

### （3）准备lobsterin、lobster执行文件，lobsterin：

COHPstartEnergy -30

COHPendEnergy 30

basisSet Bunge

includeOrbitals spd

cohpGenerator from 1.886 to 1.887 type F type Rb

#可以去Vesta确定需要哪两个原子之间成键，点化学键即可在下方显示键长

#其他产生COHP方式见手册lobsterin部分，注意如果使用cohpbetween产生cohp的方式注意键长可能会出现错误

# cohpGenerator方式产生cohp时，COHPCAR第一列为横坐标能量值，第二列为所有原子相互作用的平均COHP，第三列为所有原子相互作用的ICOHP(COHP积分)，第四列为最近的相互作用的COHP，第五列为最近的相互作用的ICOHP(COHP积分), 第六列为第二近的相互作用的COHP，第七列为最近的相互作用的ICOHP(COHP积分)，以此类推。画图一般用第四列。

### （4）准备vasp.pbs

同dos、band的简单脚本

### （5）提交vasp.pbs

执行chmod +x lobster&&./lobster-x.x.x

得到的结果中COHPCAR.lobster里第一列是横轴，第二列是COHP

ICOHPLIST.lobster里给出ICOHP值