**PWscf操作流程**

**林子越**

## 一、动能（Encut）测试

### **（1）** **准备赝势文件**

\*.pbe-n-rrkjus\_psl.1.0.0.UPF #类似格式的.UPF文件，具体名称不同

PF（网址http://www.quantum-espresso.org/ 找到PSEUDOPOTENTIALS点击→下拉在元素周期表中点中所要的元素→在显示的页面里找到\*.pbe-n-rrkjus\_psl.1.0.0.UPF右键→连接另存为（选择桌面上传方便））

NCPP：Norm-Conserving Pseudopotentials 模守恒赝势

USPP：Ultrasoft pseudopotentials 超软赝势

PAW：Plane-waves 平面波赝势

### （2）准备脚本文件run.pbs

#注释用**蓝色**标出，需修改内容用**红色**标出，粘贴时删掉蓝色部分即可

#脚本生成可以用linux系统中的脚本直接修改，不然可能提示

qsub: script is written in DOS/Windows text format

或者修改文件format为unix：

使用vi/vim修改文件format（vi xxx.pbs）

命令：:set ff=unix

或者：:set fileformat=unix

然后:wq保存

#!/bin/bash

#

#PBS -q CT6

#PBS -N encut #任务名

#PBS -l nodes=1:ppn=36 #本项及-q根据具体机器的核心数修改

#PBS -j oe

#PBS -V

cd $PBS\_O\_WORKDIR

ulimit -s 40000 #非特定机器不需要加这两行

ulimit -n 4096

export PW\_ROOT=/home/ddf/program/espresso-5.4.0/bin #此项需要根据软件安装的具体路径给出，找到路径的办法是，去其他人工作区下的PW结果内找到相同或类似脚本的该字段

mkdir -p ./tmp

rm -rf ./tmp/\*

for ec in 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120

do

cat > F.scf.in\_$ec << EOF

&control

calculation='scf',

restart\_mode='from\_scratch',

prefix='F' #可不用改，不影响计算，如若改把脚本里的所有F替换掉

pseudo\_dir = '.',

outdir='./tmp'

tstress=.t.,

tprnfor=.t.

/

&system

ibrav = 0,

celldm(1)= 1.88972688,

nat= 8,  #结构的原子个数, 依照晶格参数第二项ATOMIC\_POSITIONS有几行来决定

ntyp= 1, #结构原子种类的个数

occupations='smearing',

smearing='methfessel-paxton',

degauss=0.02

ecutwfc=$ec,

/

&electrons

mixing\_beta = 0.7

conv\_thr = 1.0d-8

/

ATOMIC\_SPECIES

F 18.9984 F.pbe-n-rrkjus\_psl.1.0.0.UPF #元素 相对原子质量 赝势

如果有别的元素则再写一行

CELL\_PARAMETERS #加了对称性的结构

2.605373874519238 0.000000000000000 0.000000000000000

0.000000000000000 2.605373874519239 0.000000000000000

0.000000000000000 0.000000000000000 2.605373874519239

ATOMIC\_POSITIONS {crystal}

F 0.0000000000000000 0.5000000000000000 0.7500000000000000

F 0.0000000000000000 0.5000000000000000 -0.7500000000000000

F 0.7500000000000000 0.0000000000000000 0.5000000000000001

F -0.7500000000000000 0.0000000000000000 0.5000000000000000

F 0.5000000000000000 0.7500000000000000 0.0000000000000000

F 0.5000000000000000 -0.7500000000000001 0.0000000000000000

F 0.0000000000000000 0.0000000000000000 0.0000000000000000

F 0.5000000000000000 0.4999999999999999 0.5000000000000000

#以上两项从cell文件的晶格参数获得，名称可能不同，但可以直接粘贴，E指数化为小数

K\_POINTS {automatic}

12 12 12 0 0 0

EOF

mpirun -n 36 $PW\_ROOT/pw.x <F.scf.in\_$ec >F.scf.out\_$ec

done

rm -rf ./tmp\*

### （3）准备getE.sh文件

#!/bin/sh

echo "happy everyday....."

for a in 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120

do

pre=$(awk '/kinetic-energy cutoff/{ print $4 }' F.scf.out\_$a |tail -n 1)

ent=$(awk '/! total energy/{print $5}' F.scf.out\_$a |tail -n 1)

echo $pre $ent >> pe.txt

done

echo "end"

### **（4）qsub run.pbs提交运行**

### （5）得到结果，out文件未报错进行下一步

### （6）chmod a＋x getE.sh 使该文件获得管理员权限，文件名变为绿色，./getE.sh提交

**得到pe.txt文件**

40.0000 -376.71234415

45.0000 -376.72445036

50.0000 -376.72957826

55.0000 -376.73705113

60.0000 -376.74526493

65.0000 -376.75168878

70.0000 -376.75550688

75.0000 -376.75729302

80.0000 -376.75786603

85.0000 -376.75799146

90.0000 -376.75812468

95.0000 -376.75837947

100.0000 -376.75875647

105.0000 -376.75914959

110.0000 -376.75949136

115.0000 -376.75973600

120.0000 -376.75987688

通过上下间隔大概小于0.001即可选取截断能 这里选encut = 80 Ry

## 二、展宽（delph）测试

### （1）准备scf.in文件

&control

calculation='scf'

restart\_mode='from\_scratch',

prefix='F', #同上

pseudo\_dir = '.',

outdir='./tmp'

/

&system

ibrav = 0,

celldm(1)=1.88972688,

nat= 8, #同上

ntyp= 1, #同上

ecutwfc=80, #上一步动能Encut测试得到的值

ecutrho=800, #一般设置为ecutwfc的4倍 超软赝势设置为10倍

occupations='smearing',

smearing='methfessel-paxton',

degauss=0.01

la2F = .true.,

/

&electrons

conv\_thr = 1.0d-8

mixing\_beta = 0.7

/

ATOMIC\_SPECIES

F 18.9984 F.pbe-n-rrkjus\_psl.1.0.0.UPF

CELL\_PARAMETERS #结构一定之前要加对称性，不然报错

2.60537300000000 0.00000000000000 0.00000000000000

0.00000000000000 2.60537300000000 0.00000000000000

0.00000000000000 0.00000000000000 2.60537300000000

ATOMIC\_POSITIONS {crystal}

F 0.000000000117762 0.499999999338523 0.749986558395997

F 0.000000000339613 0.500000000256769 -0.749986558605385

F 0.749987711283413 0.000000000252767 0.499999999798142

F -0.749987712380392 0.000000000217478 0.500000000505681

F 0.500000000854287 0.749988023424662 0.000000002496860

F 0.500000000234474 -0.749988023462086 -0.000000002382903

F -0.000000000905482 -0.000000000090146 -0.000000000183659

F 0.500000000456325 0.500000000062032 0.499999999975266

K\_POINTS {automatic} #设太大算不完，也容易删不完，考虑机器性能

10 10 10 0 0 0

EOF

### （2）准备ph.in文件

Electron-phonon coefficients for F

&inputph

prefix='F',

fildvscf='Fdv',

amass(1)= 18.9984, #F原子的质量，如果有其他原子，则逗号后继续输入amass(2) = xxxx

outdir='./tmp',

fildyn='F.dyn',

electron\_phonon='interpolated'

el\_ph\_sigma=0.005,

el\_ph\_nsigma=10,

trans=.true.,

ldisp=.false.

tr2\_ph = 1.0d-12 /

0.000000000000000E+00 0.000000000000000E+00 0.000000000000000E+00

### （3）准备赝势文件（与encut相同）

### （4）准备getdegauss\_1.sh文件

#!/bin/bash

#by authors mayanbin 20150924

date=`date +%Y%m%d`

echo -e "\033[032m$date\033[0m"

awk '/Gaussian Broadening:/{print $3}' elph.\* > degauss.dat

sed -e "s/( /a/g" elph.\* | awk '/lambda/{print $2}' > lambda.dat

wait

awk '{print $1}' lambda.dat | head -n 24 > 0.005.dat #第一个数是原子个数的三倍

awk '{print $1}' lambda.dat | sed -n '25,48p' > 0.010.dat 之后往下依次每组三个，比如最开始是24，

awk '{print $1}' lambda.dat | sed -n '49,72p' > 0.015.dat 下面就是25,48，然后49,72，遍及范围内的所有

awk '{print $1}' lambda.dat | sed -n '73,96p' > 0.020.dat 数字以此类推

awk '{print $1}' lambda.dat | sed -n '97,120p' > 0.025.dat

awk '{print $1}' lambda.dat | sed -n '121,144p' > 0.030.dat

awk '{print $1}' lambda.dat | sed -n '145,168p' > 0.035.dat

awk '{print $1}' lambda.dat | sed -n '169,192p' > 0.040.dat

awk '{print $1}' lambda.dat | sed -n '193,216p' > 0.045.dat

awk '{print $1}' lambda.dat | tail -n 24 > 0.050.dat

wait

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.005.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.010.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.015.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.020.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.025.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.030.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.035.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.040.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.045.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.050.dat >> log

echo -e "\033[034mend\033[0m"

cp log ../log\_$i

cp degauss.dat ../

rm log\* \*.dat job.pbs\*

cd ../

done

paste degauss.dat log\_0.005 log\_0.01 log\_0.015 log\_0.02 log\_0.025 log\_0.03 log\_0.035 log\_0.04 log\_0.045 log\_0.05 >> pe.txt

rm log\_\* degauss.dat

### （5）准备rmdat.sh文件

#!/bin/bash

#by mayanbin

rm pe.txt

for i in 0.005 0.01 0.015 0.02 0.025 0.03 0.035 0.04 0.045 0.05

do

cd $i

rm log\*

rm \*.dat

rm -r tmp

cd ../

done

### （6）准备getdate\_2.sh文件

#!/bin/bash

#by mayanbin

for i in 0.005 0.01 0.015 0.02 0.025 0.03 0.035 0.04 0.045 0.05

do

cp getdegauss\_1.sh getlambda\_2.sh $i

cd $i

awk '/Gaussian Broadening:/{print $3}' elph.\* > degauss.dat

sed -e "s/( /a/g" elph.\* | awk '/lambda/{print $3}' > lambda.dat

wait

awk '{print $1}' lambda.dat | head -n 24 > 0.005.dat #第一个数是原子个数的三倍

awk '{print $1}' lambda.dat | sed -n '25,48p' > 0.010.dat 之后往下依次每组三个，比如最开始是24，

awk '{print $1}' lambda.dat | sed -n '49,72p' > 0.015.dat 下面就是25,48，然后49,72，遍及范围内的所有

awk '{print $1}' lambda.dat | sed -n '73,96p' > 0.020.dat 数字以此类推，总之同上

awk '{print $1}' lambda.dat | sed -n '97,120p' > 0.025.dat

awk '{print $1}' lambda.dat | sed -n '121,144p' > 0.030.dat

awk '{print $1}' lambda.dat | sed -n '145,168p' > 0.035.dat

awk '{print $1}' lambda.dat | sed -n '169,192p' > 0.040.dat

awk '{print $1}' lambda.dat | sed -n '193,216p' > 0.045.dat

awk '{print $1}' lambda.dat | tail -n 24 > 0.050.dat

wait

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.005.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.010.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.015.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.020.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.025.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.030.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.035.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.040.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.045.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.050.dat >> log

cp log ../log\_$i

cp degauss.dat ../

rm log\* \*.dat job.pbs\*

cd ../

done

paste degauss.dat log\_0.005 log\_0.01 log\_0.015 log\_0.02 log\_0.025 log\_0.03 log\_0.035 log\_0.04 log\_0.045 log\_0.05 >> pe.txt

rm log\_\* degauss.dat

### （7）准备getlambda\_2.sh文件

#!/bin/bash

#by mayanbin 20150924

awk '{print $1}' lambda.dat | head -n 24 > 0.005.dat

awk '{print $1}' lambda.dat | sed -n '25,48p' > 0.010.dat

awk '{print $1}' lambda.dat | sed -n '49,72p' > 0.015.dat

awk '{print $1}' lambda.dat | sed -n '73,96p' > 0.020.dat

awk '{print $1}' lambda.dat | sed -n '97,120p' > 0.025.dat

awk '{print $1}' lambda.dat | sed -n '121,144p' > 0.030.dat

awk '{print $1}' lambda.dat | sed -n '145,168p' > 0.035.dat

awk '{print $1}' lambda.dat | sed -n '169,192p' > 0.040.dat

awk '{print $1}' lambda.dat | sed -n '193,216p' > 0.045.dat

awk '{print $1}' lambda.dat | tail -n 24 > 0.050.dat

wait

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.005.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.010.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.015.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.020.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.025.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.030.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.035.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.040.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.045.dat >> log

awk 'BEGIN{sum=0}{sum+=$1}END{print sum}' 0.050.dat >> log

### （8）准备脚本job.pbs

#!/bin/bash

#

#PBS -q CT6 #需修改内容已用红色标出，修改内容与encut基本相同，仅节点、核心数及路径

#PBS -N Delph

#PBS -l nodes=1:ppn=36

#PBS -j oe

#PBS -V

cd $PBS\_O\_WORKDIR

export PW\_ROOT=/share/apps/compiler/QE/espresso-5.4.0/bin/

mkdir -p ./tmp

rm -rf ./tmp/\*

# running program

#SCF at dense k-mesh, good enough for electronic DOS

mpirun -n 36 $PW\_ROOT/pw.x <scf.in >scf.out

mpirun -n 36 $PW\_ROOT/ph.x <ph.in >ph.out

# clean

rm -rf ./tmp/\*

### **（9）准备cpelph.sh**

**（该步骤目的是拷贝出lambda文件使得getdate\_2.sh能够运行）**

cp 0.005/elph\_dir/elph.inp\_lambda.1 0.005/

cp 0.01/elph\_dir/elph.inp\_lambda.1 0.01/

cp 0.015/elph\_dir/elph.inp\_lambda.1 0.015/

cp 0.02/elph\_dir/elph.inp\_lambda.1 0.02/

cp 0.025/elph\_dir/elph.inp\_lambda.1 0.025/

cp 0.03/elph\_dir/elph.inp\_lambda.1 0.03/

cp 0.035/elph\_dir/elph.inp\_lambda.1 0.035/

cp 0.04/elph\_dir/elph.inp\_lambda.1 0.04/

cp 0.045/elph\_dir/elph.inp\_lambda.1 0.045/

cp 0.005/elph\_dir/elph.inp\_lambda.1 0.05/

### （10）准备degauss\_1.sh

#!/bin/bash

#by authors mayanbin

for d in 0.005 0.01 0.015 0.02 0.025 0.03 0.035 0.04 0.045 0.05

do

mkdir $d

cp \*.UPF ph.in job.pbs $d

cd $d

mkdir tmp

cd ../

cat ./scf.in | sed -e "s/degauss=0.02/degauss=$d/g" \

> $d/scf.in

cd $d

qsub job.pbs

cd ../

done

### （11）qsub job.pbs提交脚本运行（物理学院为qsub，超算bsub）

提交后输出\*.dyn, Delph.oxxxxx, degauss\_1.sh, ph.out, scf.out, tmp文件夹, elph\_dir文件夹

检查out文件里是否有报错内容，若无报错，进行下一步。

### （12）提交顺序

**chmod +x** **degauss\_1.sh**

**./degauss\_1.sh**

**获取管理员权限并提交并自动产生一系列任务**

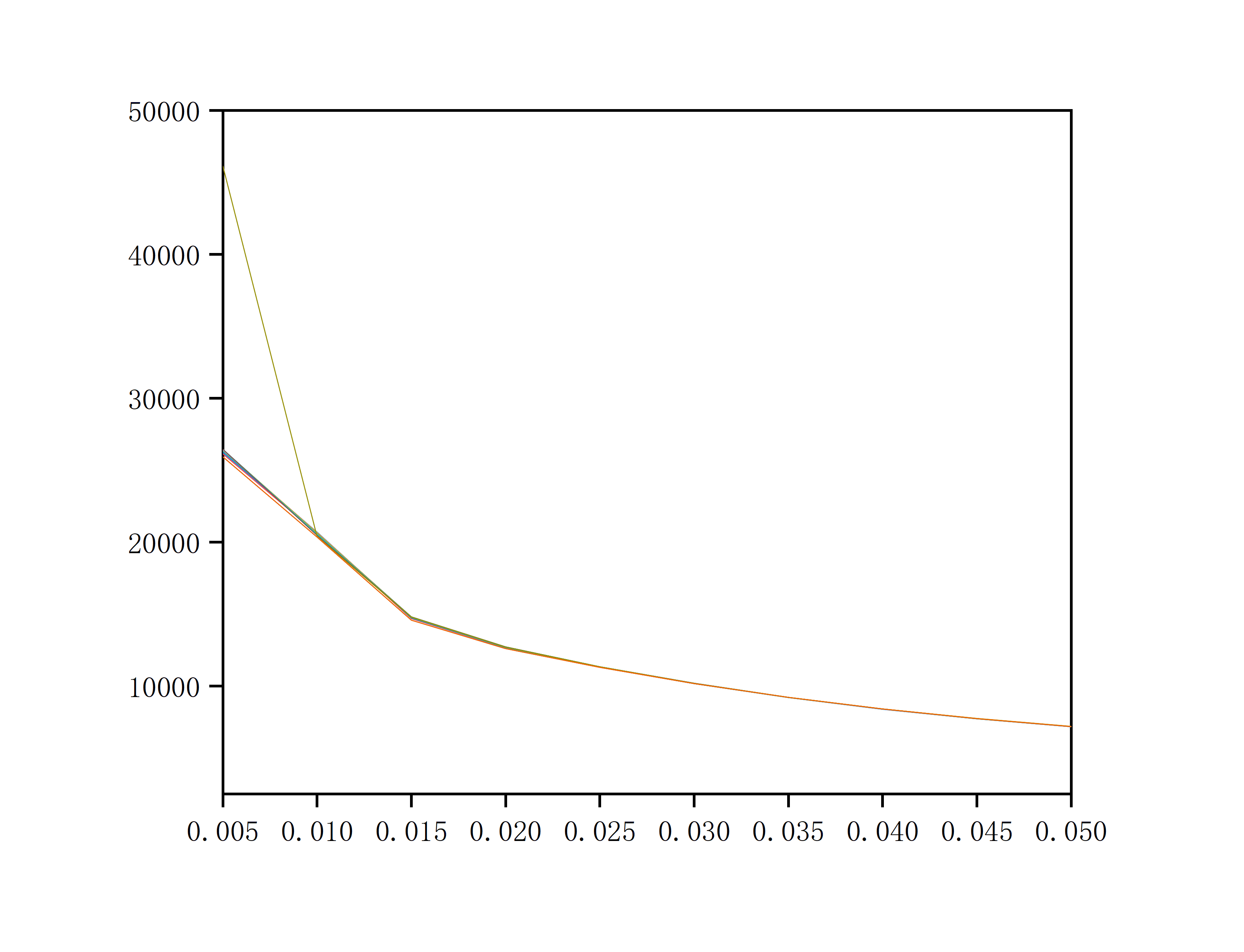
**chmod +x cpelph.sh**

**./cpelph.sh**

**chmod +x getdate\_2.sh**

**./getdate\_2.sh**

**产生pe.txt将其导入Origin，看图定展宽**

****

如图展宽取0.03（即取几条线重合的位置）

## 三、结构优化

### （1）准备赝势文件（与之前相同）

### （2）利用phonopy命令加对称性

#### i) 首先将cell格式的晶格参数手动改为如下的POSCAR格式

**CELL:**

CELL\_PARAMETERS (alat= 1.88972688)

2.6086092816666668 0.0000000000000000 0.0000000000000000

0.0000000000000000 2.6086092816666668 0.0000000000000000

0.0000000000000000 0.0000000000000000 2.6086092816666668

ATOMIC\_POSITIONS {crystal}

F 0.0000000000000000 0.5000000000000000 0.7500000000000000

F 0.0000000000000000 0.5000000000000000 0.2500000000000000

F 0.7500000000000000 0.0000000000000000 0.5000000000000000

F 0.2500000000000000 0.0000000000000000 0.5000000000000000

F 0.5000000000000000 0.7500000000000000 0.0000000000000000

F 0.5000000000000000 0.2500000000000000 0.0000000000000000

F 0.0000000000000000 0.0000000000000000 0.0000000000000000

F 0.5000000000000000 0.5000000000000000 0.5000000000000000F

**POSCAR：**

1.0

2.605373874519238 0.000000000000000 0.000000000000000

0.000000000000000 2.605373874519239 0.000000000000000

0.000000000000000 0.000000000000000 2.605373874519239

F

8

Direct

0.0000000000000000 0.5000000000000000 0.7500000000000000

0.0000000000000000 0.5000000000000000 -0.7500000000000000

0.7500000000000000 0.0000000000000000 0.5000000000000001

-0.7500000000000000 0.0000000000000000 0.5000000000000000

0.5000000000000000 0.7500000000000000 0.0000000000000000

0.5000000000000000 -0.7500000000000001 0.0000000000000000

0.0000000000000000 0.0000000000000000 0.0000000000000000

0.5000000000000000 0.4999999999999999 0.5000000000000000

#### ii) 然后将POSCAR保存为单独文件

#### iii) 执行：phonopy --symmetry POSCAR

#### iv) 得到PPOSCAR文件和BPOSCAR文件

#### v) 将得到的PPOSCAR文件中的参数以cell文件的格式输入到下方脚本

### （3）准备脚本pwelph.pbs

#!/bin/bash

#

#PBS -q CT6

#PBS -N pw-opt

#PBS -l nodes=1:ppn=36

#PBS -j oe

#PBS -V

cd $PBS\_O\_WORKDIR

export PW\_ROOT=/share/apps/compiler/QE/espresso-5.4.0/bin/

#ulimit -s 40000 #个别服务器添加

#ulimit -n 4096

# running program

cat > F.scf.in << EOF

&control

calculation='vc-relax',

restart\_mode='from\_scratch',

dt=30

prefix='F.scf'

pseudo\_dir = '.',

outdir='./tmp'

tstress=.t.,

tprnfor=.t.

nstep=10000

/

&system

ibrav = 0,

celldm(1)=1.88972688

nat= 8, #这四项与展宽测试scf.in相同

ntyp= 1,

ecutwfc=80,

ecutrho=800.

occupations='smearing',

smearing='mp',

degauss=0.03 #展宽测试结果

nosym = .f. #此处设为f以加上对称性操作

/

&electrons

conv\_thr = 1.0d-8

mixing\_beta = 0.7

/

&IONS

ion\_dynamics = 'damp'

upscale = 20

/

&CELL

cell\_dynamics = 'damp-pr'

press = 30000 #设置压力 (单位kbar，1kbar=0.01GPa)

/

ATOMIC\_SPECIES #以下均与之前相同

F 18.9984 F.pbe-n-rrkjus\_psl.1.0.0.UPF

CELL\_PARAMETERS

2.60537300000000 0.00000000000000 0.00000000000000

0.00000000000000 2.60537300000000 0.00000000000000

0.00000000000000 0.00000000000000 2.60537300000000

ATOMIC\_POSITIONS {crystal}

F 0.000000000117762 0.499999999338523 0.749986558395997

F 0.000000000339613 0.500000000256769 -0.749986558605385

F 0.749987711283413 0.000000000252767 0.499999999798142

F -0.749987712380392 0.000000000217478 0.500000000505681

F 0.500000000854287 0.749988023424662 0.000000002496860

F 0.500000000234474 -0.749988023462086 -0.000000002382903

F -0.000000000905482 -0.000000000090146 -0.000000000183659

F 0.500000000456325 0.500000000062032 0.499999999975266

K\_POINTS {automatic}

10 10 10 0 0 0

EOF

mpiexec -n 36 $PW\_ROOT/pw.x <F.scf.in >F.scf.out

done

### （4）提交脚本 qsub pwelph.pbs

### （5）格式转换

**在scf.out中得到的晶格参数还需要转换为POSCAR格式，用phonopy命令再次加对称性（不加对称性scf计算可能报错；可以通过设置nosym = .f.解决），然后将最终的结果以cell文件晶格参数的格式输入之后的自洽计算中。**

## 四、自洽计算

**#**第一步自洽目的有三个：

1. 确定对称操作个数，判断对称性是否正确 在scf.out 中Sym. Ops.字段可以找到

点群-对称操作个数对应网站：http://img.chem.ucl.ac.uk/sgp/large/sgp.htm

1. 确定q点个数，亦即.dyn文件的数目 cart. coord. 字段
2. 考察作用力，是否为所需压强 total stress 字段

**新建另一文件夹 可命名为3-scf**

### （1）准备赝势文件

### （2）scf.in （q点与ph.in一致）

&control

calculation='scf'

restart\_mode='from\_scratch',

prefix='F',

pseudo\_dir = './',

outdir='./tmp'

tstress=.t.,

verbosity='high'

/

&system

ibrav = 0,

celldm(1)=1.88972688,

nat= 8, #结构的原子个数, 依照晶格参数第二项ATOMIC\_POSITIONS有几行来决定

ntyp= 1, #结构原子种类的个数

ecutwfc=80, #截断能测试得到的值

ecutrho=800, #一般为ecutwfc的四倍，超软赝势给10倍

occupations='smearing',

smearing='methfessel-paxton',

degauss=0.03 #展宽测试的结果

la2F = .true.,

/

&electrons

conv\_thr = 1.0d-8

mixing\_beta = 0.7

/

ATOMIC\_SPECIES

F 18.9984 F.pbe-n-rrkjus\_psl.1.0.0.UPF #原子质量、赝势及晶格参数等

CELL\_PARAMETERS (alat= 1.88972688)

2.6086092816666668 0.0000000000000000 0.0000000000000000

0.0000000000000000 2.6086092816666668 0.0000000000000000

0.0000000000000000 0.0000000000000000 2.6086092816666668

ATOMIC\_POSITIONS {crystal}

F 0.0000000000000000 0.5000000000000000 0.7500000000000000

F 0.0000000000000000 0.5000000000000000 0.2500000000000000

F 0.7500000000000000 0.0000000000000000 0.5000000000000000

F 0.2500000000000000 0.0000000000000000 0.5000000000000000

F 0.5000000000000000 0.7500000000000000 0.0000000000000000

F 0.5000000000000000 0.2500000000000000 0.0000000000000000

F 0.0000000000000000 0.0000000000000000 0.0000000000000000

F 0.5000000000000000 0.5000000000000000 0.5000000000000000

K\_POINTS {automatic}

4 4 4 0 0 0 #设为ph.in中给出的q点（ph.in给的是4 4 4）

### （3）准备脚本

#!/bin/bash

#PBS -q CT1

#PBS -l nodes=1:ppn=12

#PBS -j oe

#PBS -V

#PBS -N scf-3

cd $PBS\_O\_WORKDIR

mpirun -np 12 /share/apps/compiler/QE/espresso-5.4.0/bin/pw.x <scf.in> scf.out

#clean

rm -rf tmp

### （4）提交脚本

## 五、第二次自洽计算

**新建另一个文件夹可命名为4-scf**

### （1）准备赝势文件

### （2）产生具体q点

**（例子在Genq\_qpoints文件夹内）**

**首先新建文件夹Genq\_qpoints**

#### i）需要准备一系列文件

**（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**

**将这一部分输入到下一步的k点位置处**

### （3）scf.in （q/k点为具体q/k点）

&control

calculation='scf'

restart\_mode='from\_scratch',

prefix='F',

pseudo\_dir = './',

outdir='./tmp'

tstress=.t.,

verbosity='high'

/

&system

ibrav = 0,

celldm(1)=1.88972688,

nat= 8, #结构的原子个数, 依照晶格参数第二项ATOMIC\_POSITIONS有几行来决定

ntyp= 1, #结构原子种类的个数

ecutwfc=80, #截断能测试得到的值

ecutrho=800, #一般为ecutwfc的四倍，超软赝势给8倍

occupations='smearing',

smearing='methfessel-paxton',

degauss=0.03 #展宽测试的结果

la2F = .true.,

/

&electrons

conv\_thr = 1.0d-8

mixing\_beta = 0.7

/

ATOMIC\_SPECIES

F 18.9984 F.pbe-n-rrkjus\_psl.1.0.0.UPF #原子质量、赝势及晶格参数等

CELL\_PARAMETERS (alat= 1.88972688)

2.6086092816666668 0.0000000000000000 0.0000000000000000

0.0000000000000000 2.6086092816666668 0.0000000000000000

0.0000000000000000 0.0000000000000000 2.6086092816666668

ATOMIC\_POSITIONS {crystal}

F 0.0000000000000000 0.5000000000000000 0.7500000000000000

F 0.0000000000000000 0.5000000000000000 0.2500000000000000

F 0.7500000000000000 0.0000000000000000 0.5000000000000000

F 0.2500000000000000 0.0000000000000000 0.5000000000000000

F 0.5000000000000000 0.7500000000000000 0.0000000000000000

F 0.5000000000000000 0.2500000000000000 0.0000000000000000

F 0.0000000000000000 0.0000000000000000 0.0000000000000000

F 0.5000000000000000 0.5000000000000000 0.5000000000000000

K\_POINTS {crystal} #注意此处为crystal  
81

0.500000 0.000000 0.000000 1.00 #将inp.kpt文件中的值粘贴到此处，并查看行数，例如此处共给出了81个q点

0.500000 0.025000 0.025000 1.00

……

0.500000 0.050000 0.050000 1.00

### （4）准备脚本

#!/bin/bash #脚本与第一步自洽相同

#PBS -q CT1

#PBS -l nodes=1:ppn=12

#PBS -j oe

#PBS -V

#PBS -N scf-4

cd $PBS\_O\_WORKDIR

mpirun -np 12 /share/apps/compiler/QE/espresso-5.4.0/bin/pw.x <scf.in> scf.out

#clean

rm -rf tmp

### （5）提交脚本

## 六、电声耦合及自洽计算

### （1）准备赝势文件（与之前一致）

### （2）准备文件ph.in

Electron-phonon coefficients for F

&inputph

prefix = 'F'

fildvscf = 'dvscf'

fildyn = 'F.dyn'

amass(1)= 18.9984,

outdir='./tmp',

electron\_phonon='interpolated',

el\_ph\_sigma=0.05,

el\_ph\_nsigma=10,

alpha\_mix=0.5

recover=.true.

trans=.true.,

ldisp=.true.

nq1=4, #此处给的q/k点为4 4 4，一般给一个3-5之间的值，会影响计算速度，给的越大计算越慢；但给的较大可能会消除虚频

nq2=4, 把上一步最后输出的的PPOSCAR用Vesta转换成cif格式再导入MS

先加上对称性再换成原胞primitive cell，点三个波浪线图标，点calculation

具体操作可以参照CASTEP流程里相关操作

nq3=4,

tr2\_ph = 1.0d-12

/

### （3）准备文件scf.in

&control

calculation='scf'

restart\_mode='from\_scratch',

prefix='F',

pseudo\_dir = './',

outdir='./tmp'

tstress=.t.,

verbosity='high'

/

&system

ibrav = 0,

celldm(1)=1.88972688,

nat= 8, #结构的原子个数, 依照晶格参数第二项ATOMIC\_POSITIONS有几行来决定

ntyp= 1, #结构原子种类的个数

ecutwfc=80, #截断能测试得到的值

ecutrho=800, #一般为ecutwfc的四倍，超软赝势给8倍

occupations='smearing',

smearing='methfessel-paxton',

degauss=0.03 #展宽测试的结果

la2F = .true.,

/

&electrons

conv\_thr = 1.0d-8

mixing\_beta = 0.7

/

ATOMIC\_SPECIES

F 18.9984 F.pbe-n-rrkjus\_psl.1.0.0.UPF #原子质量、赝势及晶格参数等

CELL\_PARAMETERS (alat= 1.88972688)

2.6086092816666668 0.0000000000000000 0.0000000000000000

0.0000000000000000 2.6086092816666668 0.0000000000000000

0.0000000000000000 0.0000000000000000 2.6086092816666668

ATOMIC\_POSITIONS {crystal}

F 0.0000000000000000 0.5000000000000000 0.7500000000000000

F 0.0000000000000000 0.5000000000000000 0.2500000000000000

F 0.7500000000000000 0.0000000000000000 0.5000000000000000

F 0.2500000000000000 0.0000000000000000 0.5000000000000000

F 0.5000000000000000 0.7500000000000000 0.0000000000000000

F 0.5000000000000000 0.2500000000000000 0.0000000000000000

F 0.0000000000000000 0.0000000000000000 0.0000000000000000

F 0.5000000000000000 0.5000000000000000 0.5000000000000000

K\_POINTS {automatic}

8 8 8 0 0 0 #设为ph.in中给出q点的二倍（ph.in给的是4 4 4）

### （4）准备文件fit.scf.in （其它与scf相同，k点网格再x2）

&control

calculation='scf'

restart\_mode='from\_scratch',

prefix='F',

pseudo\_dir = './',

outdir='./tmp'

tstress=.t.,

verbosity='high'

/

&system

ibrav = 0,

celldm(1)=1.88972688,

nat= 8, #结构的原子个数, 依照晶格参数第二项ATOMIC\_POSITIONS有几行来决定

ntyp= 1, #结构原子种类的个数

ecutwfc=80, #截断能测试得到的值

ecutrho=800, #一般为ecutwfc的四倍，超软赝势给8倍

occupations='smearing',

smearing='methfessel-paxton',

degauss=0.03 #展宽测试的结果

la2F = .true.,

/

&electrons

conv\_thr = 1.0d-8

mixing\_beta = 0.7

/

ATOMIC\_SPECIES

F 18.9984 F.pbe-n-rrkjus\_psl.1.0.0.UPF #原子质量、赝势及晶格参数等

CELL\_PARAMETERS (alat= 1.88972688)

2.6086092816666668 0.0000000000000000 0.0000000000000000

0.0000000000000000 2.6086092816666668 0.0000000000000000

0.0000000000000000 0.0000000000000000 2.6086092816666668

ATOMIC\_POSITIONS {crystal}

F 0.0000000000000000 0.5000000000000000 0.7500000000000000

F 0.0000000000000000 0.5000000000000000 0.2500000000000000

F 0.7500000000000000 0.0000000000000000 0.5000000000000000

F 0.2500000000000000 0.0000000000000000 0.5000000000000000

F 0.5000000000000000 0.7500000000000000 0.0000000000000000

F 0.5000000000000000 0.2500000000000000 0.0000000000000000

F 0.0000000000000000 0.0000000000000000 0.0000000000000000

F 0.5000000000000000 0.5000000000000000 0.5000000000000000

K\_POINTS {automatic}

16 16 16 0 0 0 #设为ph.in中给出q点的4倍（ph.in给的是4 4 4）

**（5）准备脚本**

#!/bin/bash

#PBS -q CT1

#PBS -l nodes=1:ppn=12

#PBS -j oe

#PBS -V

#PBS -N QE-E.P.

cd $PBS\_O\_WORKDIR

mpirun -np 12 /share/apps/compiler/QE/espresso-5.4.0/bin/pw.x <fit.scf.in> fit.scf.out

mpirun -np 12 /share/apps/compiler/QE/espresso-5.4.0/bin/pw.x <scf.in> scf.out

mpirun -np 12 /share/apps/compiler/QE/espresso-5.4.0/bin/ph.x <ph.in> ph.out

#clean

rm -rf tmp

#以上按具体路径及节点修改即可

***最后提交脚本***

## 七、声子谱及其态密度

**现在开始后处理，首先准备脚本：**

### （1）run\_q2r\_5.4.0

#!/bin/sh

cat > q2r.in << EOF

&input

fildyn='F.dyn', zasr='crystal', la2F = .true., flfrc='F.fc'

/

EOF

/share/apps/compiler/QE/espresso-5.4.0/bin/q2r.x < q2r.in > q2r.out

**然后执行脚本**

**chmod +x run\_q2r\_5.4.0**

**./ run\_q2r\_5.4.0**

### （2）run\_dos-5.4.0

#!/bin/sh

#PBS -q CT1

#PBS -l nodes=1:ppn=12

#PBS -j oe

#PBS -V

#PBS -N phonon-dos

cd $PBS\_O\_WORKDIR

####################################################################

cat > matdyn.in.dos << EOF

&input

asr='simple',

amass(1)= 18.9984,

flfrc='F.fc',

flfrq='F.freq',

dos=.true.

fldos='phdos.dat',

nk1=30, nk2=30, nk3=30,

ndos=5000

/

EOF

mpirun -np 12 /share/apps/compiler/QE/espresso-5.4.0/PHonon/PH/matdyn.x < matdyn.in.dos > matdyn.out.dos

#/share/apps/compiler/QE/espresso-5.4.0/matdyn.x < matdyn.in.dos > matdyn.out.dos

**然后提交脚本**

**qsub run\_dos\_5.4.0 （qsub为具体提交命令）**

### （3）run\_frequency

#!/bin/sh

#PBS -q CT1

#PBS -l nodes=1:ppn=12

#PBS -j oe

#PBS -V

#PBS -N phonon-frq

cd $PBS\_O\_WORKDIR

cat > matdyn.in.freq << EOF

&input

asr='simple', amass(1)= 18.9984,

flfrc='F.fc', flfrq='F.freq',la2F = .true., dos=.false.,

/

81

0.1916730 0.0000000 0.0000000 0.0246914

0.1916730 0.0095836 0.0095836 0.0246914

0.1916730 0.0191673 0.0191673 0.0246914

0.1150038 0.1150038 0.1150038 0.0246914

……

0.1916730 0.1916730 0.1916730 0.0246914

EOF

mpirun -np 12 /share/apps/compiler/QE/espresso-5.4.0/bin/matdyn.x < matdyn.in.freq > matdyn.out

#这一部分需要说明一下，上面的k点内容找第二步自洽结果scf.out里面的

number of k points= 81 Methfessel-Paxton smearing, width (Ry)= 0.0300

cart. coord. in units 2pi/alat

这一字段，注意一定是cart. coord.而不是crystal.coord. 前者为笛卡尔坐标，后者是晶体坐标

粘贴进来的过程中 可能涉及删除列的操作

改列方法：

1. linux : 不按insert 直接输入ctrl+v 用上下左右键选中需要修改的部分
2. windows: 粘贴到word内按住 alt在用左键圈选即可圈选矩形

**最后执行脚本**

**chmod +x run\_frequency**

**./run\_frequency**

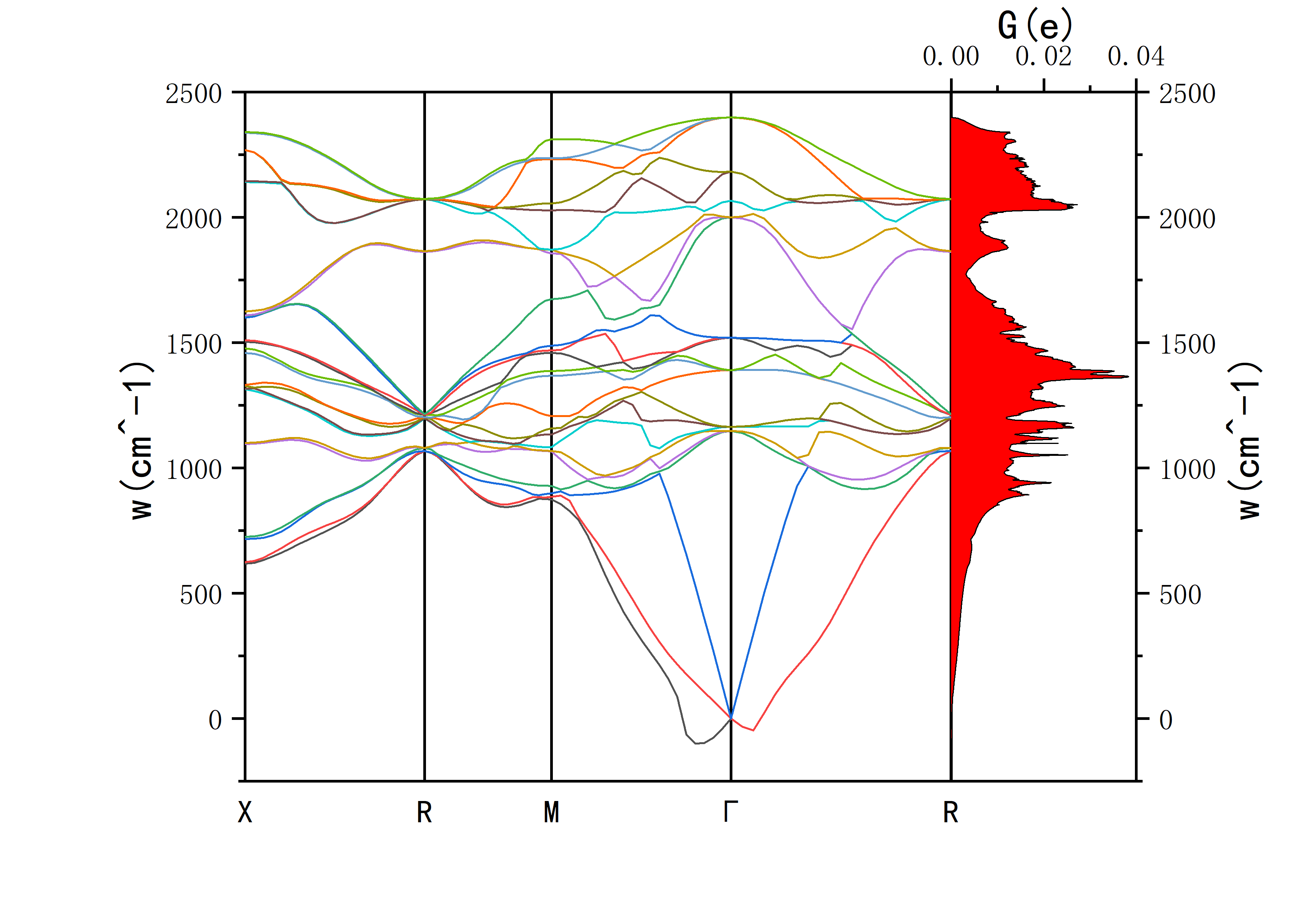
**即输出声子谱及其态密度**

**其中F.freq.gp用于画声子谱**

**phdos.dat用于做态密度**

**具体作图方法参见Origin作图流程**

**然后我们得到声子谱-态密度**



## 八、超导电性的计算

**首先打开elph\_dir文件夹**

### （1）准备脚本run\_lambda

#!/bin/sh

cat > lambda.in << EOF

80 0.52 1 #这里第一行第一个数据是需要大于声子的最高频率，单位是THz，第二个数据是展宽，给 0.5 几个。第三个数据是展宽方法，给 1 就可以。

11 #下方坐标及权重找第一步自恰(3-scf)的scf.out中的笛卡尔坐标的K点cart. coord.（类似声子谱的获取 k点步骤）

0.0000000 0.0000000 0.0000000 0.0160000

0.0000000 0.0000000 0.0766692 0.0960000

0.0000000 0.0000000 0.1533384 0.0960000

0.0000000 0.0766692 0.0766692 0.1920000

0.0000000 0.0766692 0.1533384 0.1920000

0.0000000 0.1533384 0.1533384 0.1920000

0.0766692 0.0766692 0.0766692 0.1280000

0.0766692 0.0766692 0.1533384 0.3840000

0.0766692 0.1533384 0.1533384 0.3840000

0.1533384 0.1533384 0.1533384 0.1280000

0.0000000 0.1533384 0.0766692 0.1920000

elph.inp\_lambda.1

elph.inp\_lambda.2

elph.inp\_lambda.3

elph.inp\_lambda.4

elph.inp\_lambda.5

elph.inp\_lambda.6

elph.inp\_lambda.7

elph.inp\_lambda.8

elph.inp\_lambda.9

elph.inp\_lambda.10

elph.inp\_lambda.11

0.13 ! \mu the Coloumb coefficient in the modified #mu star， 电声耦合公式的一个参数，一 般给 0.1－0.13

! Allen-Dynes formula for T\_c (via \omega\_log)

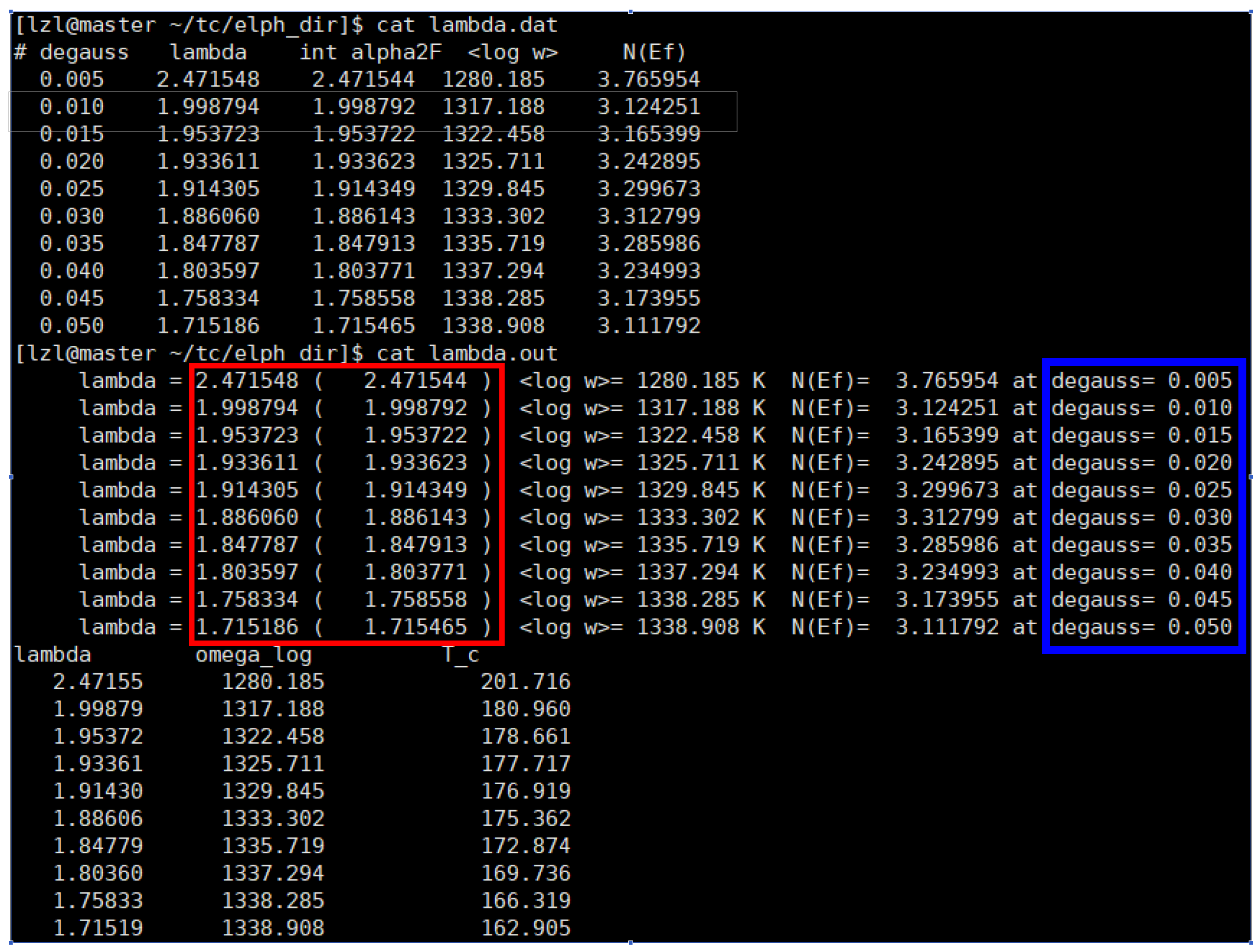
EOF

/share/apps/compiler/QE/espresso-5.4.0/bin/lambda.x < lambda.in > lambda.out

### （2）chmod +x run\_lambda

**./run\_lambda**

### （3）结果分析：计算结束产生 lambda.out 和 lambda.dat 两个文件



**以上红色矩形框中括号外和括号内的数值差别越小越好，一般到倒数第三个数字之前的都是相同比较理想。这里主要是调节 lambda.in的参数，特别是第一个高于所有声子频率的值，对括号内外数值差影响巨大，一般高于所有声子频率 1~2KHz 即可。第二个展宽调节也会有影响，至于超导转变温度 Tc，建议通过 origin 作图看一下收敛情况，收敛后的 Tc 值即为所计算结构的超导转变温度。其中作图以高斯展宽（蓝色框部分）为横坐标， Tc 为纵坐标。**

## 九、单算声子谱及其态密度

**只算一步自洽，scf.in和ph.in文件需要修改其他与电声耦合相同（删除部分参数）**

**scf.in:**

**&system字段**

&system

ibrav = 0,

celldm(1)=1.88972688,

nat= 8,

ntyp= 1,

ecutwfc=80,

ecutrho=800,

occupations='smearing',

smearing='methfessel-paxton',

degauss=0.03

/

**ph.in**

Electron-phonon coefficients for F

&inputph

tr2\_ph=1.0d-12,

prefix='F',

ldisp=.true.,

nq1=5, nq2=5, nq3=5

amass(1)= 18.9984,

outdir='./tmp',

fildyn='F.dyn',

alpha\_mix=0.5

/

## 十、超导电性的计算-四面体方法

一切文件均在压缩包Tc.tar.gz内

### （1）结构优化（同第三步）

新建文件夹opt进行结构优化，输出scf.out文件

### （2）自洽计算

准备文件：\*.scf.in，job.pbs，赝势文件。

#### \*.scf.in

&CONTROL

calculation = 'scf' ,

prefix='RbF',

pseudo\_dir = './',

outdir='./tmp'

/

&SYSTEM

ibrav = 0,

celldm(1) = 1.88972688,

nat = 12,

ntyp = 2,

ecutwfc = 80 ,

ecutrho = 320 ,

occupations = 'tetrahedra\_opt' ,

/

&ELECTRONS

/

ATOMIC\_SPECIES

F 18.9984 F.pbe-n-kjpaw\_psl.1.0.0.UPF

Rb 85.4678 Rb.pbe-spn-kjpaw\_psl.1.0.0.UPF

CELL\_PARAMETERS

2.96999643307421 0.00000000000000 0.00000000000000

-1.48499821653710 2.57209236019143 0.00000000000000

0.00000000000000 0.00000000000000 4.82953068500014

ATOMIC\_POSITIONS (crystal)

F 0.00000000000000 0.00000000000000 0.50000000000000

F 0.00000000000000 0.00000000000000 0.00000000000000

F 0.66197346800000 0.83098673400000 0.25000000000000

F 0.33802653200000 0.16901326600000 0.75000000000000

F 0.16901326600000 0.33802653200000 0.25000000000000

F 0.83098673400000 0.66197346800000 0.75000000000000

F 0.83098673400000 0.16901326600000 0.75000000000000

F 0.16901326600000 0.83098673400000 0.25000000000000

Rb 0.33333333333333 0.66666666666667 0.56280298000000

Rb 0.66666666666667 0.33333333333333 0.06280298000000

Rb 0.33333333333333 0.66666666666667 0.93719702000000

Rb 0.66666666666667 0.33333333333333 0.43719702000000

K\_POINTS automatic

8 8 8 0 0 0

#改法同前面的scf.in

#### job.pbs

#!/bin/bash

#

#PBS -q CT3

#PBS -N RbF2\_scf

#PBS -l nodes=1:ppn=28

#PBS -j oe

#PBS -V

cd $PBS\_O\_WORKDIR

mpirun -n 28 /share/apps/compiler/QE/espresso-6.4/bin/pw.x <RbF.scf.in >RbF.scf.out

#### 提交脚本 qsub job.pbs

会输出tmp文件夹。

### （3）phonon

新建文件夹phonon-2进行第二步计算；拷贝scf的输出文件夹tmp到此文件夹内。

准备文件：\*.ph.in，job.pbs，赝势文件

#### \*.ph.in

RbF phonon

&INPUTPH

prefix = 'RbF',

outdir = './tmp',

fildyn = 'RbF.dyn'

fildvscf = 'dv',

fildrho = 'drho',

ldisp = .true.,

lshift\_q = .true.,

nq1 = 4,

nq2 = 4,

nq3 = 4,

/

#酌情设置nq1,nq2,nq3；一般为4 4 4；与之前超导做法设置原则类似

#### job.pbs

#!/bin/bash

#PBS -q CT3

#PBS -l nodes=1:ppn=28

#PBS -j oe

#PBS -V

#PBS -N RbF7\_ph

cd $PBS\_O\_WORKDIR

mpirun -n 28 /share/apps/compiler/QE/espresso-6.4/bin/ph.x < RbF.ph.in >RbF.ph.out

#### 提交脚本

得到新的tmp文件夹，及一系列.dyn文件

### （4）elph

新建文件夹elph-3进行第三步计算；拷贝phonon-2中的tmp文件夹及一系列.dyn文件到elph-3文件夹内。

准备文件：\*.elph.in，job.pbs，赝势文件，alpha2x.pbs

#### \*.elph.in

&INPUTPH

prefix = 'RbF',

outdir = './tmp',

fildyn = 'RbF.dyn'

fildvscf = 'dv',

fildrho = 'drho',

ldisp = .true.,

lshift\_q = .true.,

nq1 = 4,

nq2 = 4,

nq3 = 4,

electron\_phonon = "lambda\_tetra"

nk1 = 4,

nk2 = 4,

nk3 = 4,

/

&INPUTa2F

nfreq = 800

/

#### job.pbs

#!/bin/bash

#PBS -q CT3

#PBS -l nodes=1:ppn=28

#PBS -j oe

#PBS -V

#PBS -N RbF6\_elph

cd $PBS\_O\_WORKDIR

mpirun -np 28 /share/apps/compiler/QE/espresso-6.4/bin/ph.x < RbF.elph.in >RbF.elph.out

#### alpha2x.pbs

#!/bin/bash

#PBS -q CT3

#PBS -l nodes=1:ppn=28

#PBS -j oe

#PBS -V

#PBS -N RbF6\_elph

cd $PBS\_O\_WORKDIR

mpirun -np 28 /share/apps/compiler/QE/espresso-6.4/bin/alpha2f.x < RbF.elph.in > alpha2f.out

#### 首先提交job.pbs，待任务完成，产生了所有\*.dyn\*文件对应的\*.dyn\*.elph.\*后，提交alpha2x.pbs

#### gnuplot \*.McMillan.gp 画图，观察曲线在哪里收敛，得到Tc. 其余参数输出在alpha2f.out内。