

VASP 手册  
——学习记录

From goodogo.github.io

辞暮尔尔，烟火年年，朝朝暮暮，岁岁平安。

# 目录

<b>1 写在前面</b>	<b>7</b>
<b>2 操作软件进行计算</b>	<b>7</b>
2.1 用什么软件? . . . . .	7
2.1.1 VASP . . . . .	7
2.1.2 VESTA . . . . .	7
2.1.3 Material Studio . . . . .	7
2.1.4 Matlab . . . . .	8
2.1.5 Gunplot . . . . .	8
2.1.6 Origin . . . . .	8
2.1.7 Vasplit . . . . .	8
2.1.8 EndNote . . . . .	8
2.1.9 MathType . . . . .	8
2.2 如何输入数据? . . . . .	8
2.2.1 四个输入文件 . . . . .	8
2.2.2 INCAR . . . . .	9
2.2.3 POSCAR . . . . .	12
2.2.4 POTCAR . . . . .	13
2.2.5 KPOINTS . . . . .	13
2.2.6 任务提交 . . . . .	14
2.2.7 说明 . . . . .	14
2.3 数据提取 . . . . .	15
2.3.1 OUTCAR . . . . .	15
2.3.2 grep 语句 . . . . .	15
2.3.3 说明 . . . . .	15
<b>3 基本实例计算</b>	<b>16</b>
3.1 单点能计算 . . . . .	16
3.1.1 准备 POSCAR . . . . .	16
3.1.2 准备 INCAR . . . . .	17
3.1.3 准备 KPOINTS . . . . .	18
3.1.4 准备 POTCAR . . . . .	18
3.1.5 进行计算 . . . . .	18
3.1.6 数据提取与收集 . . . . .	18
3.2 结构优化 . . . . .	19
3.2.1 准备四种基本输入 . . . . .	19
3.2.2 进行计算 . . . . .	19
3.2.3 数据收集以及 CONTCAR . . . . .	19
3.2.4 说明 . . . . .	20
3.3 拟合方程 . . . . .	20

3.3.1	什么是 BM 方程? *	20
3.3.2	什么是二项式?	21
3.3.3	什么是拟合?	21
3.3.4	如何得到许多晶格参数 (a) 与能量 (E)?	21
3.4	批量操作	22
3.4.1	for 循环	22
3.4.2	sed 语句	22
3.4.3	混合使用 (1) 之 for+sed	23
3.4.4	路径	23
3.4.5	pwd 语句	24
3.4.6	混合使用 (2) 之 for+sed	24
3.4.7	混合使用 (3) 之 for+mkdir	24
3.4.8	cd 语句	25
3.4.9	混合使用 (4) 之 for+vasp 运行	25
3.4.10	echo 语句	25
3.4.11	混合使用 (5) 之 for+grep+echo	26
3.4.12	说明	26
3.5	拟合的代码实现	27
3.5.1	计算流程	27
3.5.2	代码实现之 python	27
3.5.3	代码解释	27
3.5.4	数据输出	28
3.5.5	代码实现之使用 python 可视化拟合曲线	28
3.5.6	代码实现之 matlab	29
3.6	掺杂	29
3.6.1	扩胞	30
3.6.2	掺杂	31
3.6.3	掺杂后结构优化	32
3.6.4	空位缺陷	32
3.7	切面	32
3.7.1	切表面	32
3.7.2	表面模型优化	33
3.7.3	构建界面模型	33
3.7.4	界面模型优化	33
3.7.5	原子固定	33
4	进阶实例计算	34
4.1	态密度 (DOS)	34
4.1.1	DOS 计算流程	34
4.1.2	以 mapbi3 钙钛矿为例	34
4.2	电荷密度图	36
4.2.1	电荷密度计算流程	36

4.2.2 以 CsPbI <sub>3</sub> 钙钛矿为例 . . . . .	37
4.3 差分电荷密度图 (电荷密度差) . . . . .	39
4.3.1 计算流程 . . . . .	39
4.4 能带计算 . . . . .	39
4.4.1 计算流程 . . . . .	39
4.4.2 用 p4vasp 查看能带与 DOS . . . . .	40
4.4.3 利用 pymatgen 绘制能带与 DOS . . . . .	41
4.5 声子谱计算 . . . . .	41
4.5.1 有限位移法 . . . . .	41
4.5.2 有限位移法以单层 MoS <sub>2</sub> 为例 . . . . .	42
4.5.3 密度泛函微扰法 . . . . .	44
4.6 晶体轨道重叠布局 (COOP)、晶体轨道哈密顿布局 (COHP) . . . . .	45
4.6.1 LOBSTER 程序之 lobsterin . . . . .	45
4.6.2 计算流程 . . . . .	45
4.7 力学性质 . . . . .	46
4.7.1 弹性常数定义 . . . . .	46
4.7.2 弹性常数: 利用 vaspy 计算 . . . . .	49
4.7.3 弹性常数: 脚本计算 . . . . .	51
4.7.4 脚本文件注释 . . . . .	52
4.8 热学性质 . . . . .	55
4.9 热电性质 . . . . .	55
4.9.1 基本参数介绍 . . . . .	55
4.9.2 基本流程介绍 . . . . .	55
4.9.3 Thirdorder . . . . .	56
4.9.4 ShengBTE . . . . .	56
4.9.5 BoltzTrap . . . . .	57
4.9.6 vaspy-68 . . . . .	58
4.9.7 形变势理论 . . . . .	59
4.9.8 电子热导率 . . . . .	60
4.9.9 Seebeck 系数与电导率 . . . . .	60
4.9.10 晶格热导率 . . . . .	60
4.9.11 热电优值 . . . . .	60
4.10 高熵合金 . . . . .	60
4.10.1 ATAT-mcsqs 生成准随机结构 . . . . .	60
4.10.2 生成指定晶系的无序结构 . . . . .	61
<b>5 软件安装以及配置</b>	<b>62</b>
5.1 VASP . . . . .	62
5.1.1 版本说明 . . . . .	62
5.1.2 安装环境准备 . . . . .	62
5.1.3 安装 OneAPI . . . . .	62
5.1.4 安装 VASP . . . . .	63

5.1.5	vaspkit 安装	63
5.1.6	参考文献	64
5.2	BoltzTraP	64
5.2.1	环境以及软件安装	64
5.2.2	结合 vaspkit 尝试运行	65
5.2.3	参考文献	65
5.3	ShengBTE	65
5.3.1	环境安装	65
5.3.2	安装	65
5.3.3	参考文献	67
5.4	MS	67
5.4.1	安装（无网关版）	68
5.4.2	文件上传	68
5.4.3	模块运行（以 DMOL3 为例）	68
5.4.4	下载查看	69
5.4.5	参考文献	69
5.5	关于大模型的使用以及本地部署-以 Deepseek 为例	69
5.5.1	DeepSeek 本地部署	69
5.6	关于 pymatgen 中 BSDOSPlotter 模块的使用	70
5.6.1	一个例子	70
5.6.2	结果输出	71
6	后记	71

## 1 写在前面

“共情”是一件难事，所以把之前写的“前言”删掉了。总之，这是一些关于 VASP 操作的手册。从某种角度来看，这是一些没有什么用处的东西，只需费些功夫，基本上涉及的所有内容都可以在网络上查到。从另一角度来看，这里面所有的文字多少都是实际经过验证，一字一句的敲出来的，或许能当作参考。

希望能为你提供一些帮助，尽早完成像是毕业论文之类的这些无法避免的事情，去干一些其他的有意义、感兴趣的事。

第二章介绍了各种软件、VASP 基本的输入输出文件及其重要参数。第三章介绍了最基本的计算流程，包括单点能计算、结构优化、拟合、批量操作、掺杂、切面等。第四章介绍了计算态密度、能带、声子谱、电荷密度及其差分电荷密度、晶体轨道布局，还有基本的力学性质、热学性质、热电性质和高熵合金建模。第五章介绍了各种软件的安装及其配置、DeepSeek 的本地部署、pymatgen 绘图、Material Studio 基于 Linux-centos7 系统的模块计算简介等。

能力有限，难免出现纰漏甚至错误，若计算过程中出现手册中未提及的情况请以实际查询到的资料为准，对因此而浪费的时间致以歉意。

## 2 操作软件进行计算

### 2.1 用什么软件？

#### 2.1.1 VASP

环境：Linux 系统

功能：基于密度泛函理论计算晶体

手册官网：[https://www.vasp.at/wiki/index.php/The\\_VASP\\_Manual](https://www.vasp.at/wiki/index.php/The_VASP_Manual)

如何使用：在 LINUX 系统的 Terminal 窗口中输入：mpirun -np 72 vasp\_std(在实验室的机器中是这样的，可以当作固定搭配)

mpirun：并行命令行，简而言之用这个能加快计算速度。-np：参数指定并行的任务数量，简而言之固定搭配。72：所用机器核数，随便 32、48 都可以，据说是 6 的倍数效果都很好。vasp\_std：执行文件的名称，在 407 机器中 VASP 的执行文件名就叫这个。

#### 2.1.2 VESTA

外貌：Windows 或 Linux，可视化界面

功能：数据可视化，简而言之能让你看到计算的东西长什么样子。

如何使用：双击图标

官网：<http://jp-minerals.org/vesta/en/>

#### 2.1.3 Material Studio

这个软件是在个人电脑上安装的，主要用来对晶体模型进行一系列操作用的，例如，切面，扩胞等，其中的一些计算模块如 CASTEP 也能进行第一性原理计算。

一般使用的是盗版软件，在网上很多此类的下载、破解教程。

### 2.1.4 Matlab

这个软件有强大的矩阵运算功能，平时用于对于一般的重复性套公式计算，需要一定的代码基础。

**重点：**在计算过程中涉及到数据的二项式拟合的步骤，可以用 matlab 做的，但是这个拟合不仅仅可以用 matlab 做，选择方便以及自己喜欢的方式为好。

### 2.1.5 Gunplot

数据绘图软件，在 Linux 系统和 Windows 都可以使用，利用类似于 matlab 的代码段进行数据输入绘制，使用较为繁琐。

推荐哔哩哔哩 up 主：朱老师讲 vasp，对 Gunplot 有一系列的讲解。

### 2.1.6 Origin

科研绘图必学的强大的绘图软件，简单直接，比 Gunplot 更易上手。

### 2.1.7 Vaskit

强大的 VASP 软件的辅助软件包，从基本的输出文件的生成到力学性质、载运性质的计算一应俱全，其中的功能能节省大量的时间。

### 2.1.8 EndNote

常用的文献管理软件，常使用在文献插入引用方面。

### 2.1.9 MathType

适用于 Word 的公式编辑工具软件，由于在 Word 自带的公式插入选项卡中无法更改公式为 Times New Roman，MathType 不但能更改为相应公式字体并且能够很方便的进行公式自动后缀编号。

在官网进行下载，首次使用提供三十天的免费试用时间。（<https://mathtype.org/download/>）

## 2.2 如何输入数据？

### 2.2.1 四个输入文件

上文所述，若将 VASP 软件比作计算器，那肯定是少不了两个必要的过程：输入与输出过程。如下给出 VASP 中的输入、输出文件可以识别的名称（注意全为大写且不可写错），其中 **INCAR**、**POTCAR**、**KPOINTS**、**POSCAR** 为必要输入文件。（且全部为文本文件格式）

当在文件夹中准备好这四个输入文件后，在此目录下空白处右击，进入 Terminal，输入 `mpirun -np 72 vasp_std` 即可进行运算。

	文件名称	输入 / 输出
1	INCAR	in **
2	STOFCAR	in
3	stout	out
4	POTCAR	in **
5		

```
6          KPOINTS    in    **
7          IBZKPT     out
8          POSCAR     in    **
9          CONTCAR    out
10         CHGCAR     in/out
11         CHG        out
12         WAVECAR    in/out
13         TMPCAR     in/out
14         EIGENVAL   out
15         DOSCAR     out
16         PROCAR     out
17         OSZICAR    out
18         PCDAT      out
19         XDATCAR    out
20         LOCPOT     out
21         ELFCAR     out
22         PROOUT     out
```

### 2.2.2 INCAR

INCAR 是 VASP 的核心输入文件，所涉及的参数截止到 2024/12 共 522 个（据 vaspwiki），但是这么多的参数仅有少部分是需要经常用到的，且更少的一小部分是需要一知半解其具体含义的。

```
1          EDIFF = 0.0001
2          ENCUT = 520
3          IBRION = 2
4          ISIF = 3
5          ISMEAR = -5
6          NELM = 100
7          NSW = 99
8          SIGMA = 0.05
```

**1、ISIF:** 规定晶胞形状、体积以及原子位置是否变化，是否受应力作用。

**注意：**体积改变时推荐 ENCUT=ENMAX\*1.3 以及 PREC=HIGH

**常用：**ISIF=2：仅改变原子位置不改变晶胞形状与体积。ISIF=3：原子位置，晶胞形状、体积都改变。

ISIF	应力	应力张量	位置	晶胞形状	晶胞体积
0	yes	no	yes	no	no
1	yes	trace only	yes	no	no
2	yes	yes	yes	no	no
3	yes	yes	yes	yes	yes
4	yes	yes	yes	yes	no
5	yes	yes	no	yes	no
6	yes	yes	no	yes	yes
7	yes	yes	no	no	yes
8	yes	yes	yes	no	yes

2、ICHARG: 规定 VASP 的初始电荷密度。

3、ISTART: 是否对于 WAVECAR 文件进行读取。默认为零，即从头开始不进行任何读取。

4、ENCUT: 用于规定平面波的能量截断能，默认为 POTCAR 文件中最大的 ENMAX。

5、EDIFF: 是用于规定电子自洽循环中的收敛条件，即在循环中达到这个范围可以认定符合收敛条件可以结束计算。单位为 eV, 默认为  $10^{-4}$ .

6、NELM: 是用于规定最小电子步，即达到规定的最小电子步可认定为计算结束。默认为 60。

7、EDIFFG: 是用于规定离子弛豫循环中的收敛条件，默认为 EDIFF 的十倍。但常设置为负数，当力的标准小于其绝对值后认为达到收敛。(注意不能应用于分子动力学模拟)

7、NSW: 是用于规定最小离子步，即达到规定的最小离子步可认定为计算结束。默认为零，即不进行离子步。

8、IBRION: 是用于描述在计算过程中晶体结构的改变。

```

1 # 结构不变
2 IBRION=-1 (避免同时设置 NES>0)
3 # 分子动力学
4 IBRION=0
5 # 结构优化
6 IBRION=1 RMM-DIIS
7 IBRION=2 共轭梯度
8 IBRION=3 damped molecular dynamics
9 # 声子计算
10 IBRION=5 无对称有限差分
11 IBRION=6 对称有限差分
12 IBRION=7 无对称微扰
13 IBRION=8 对称微扰
14 # 过度态分析
15 IBRION=40 intrinsic-reaction-coordinate
16 # 计算
17 IBRION=44 improved dimer method
18 # 用户交互
19 IBRION=11 from standard input
20 IBRION=12 from Python plugin

```

**9、ISMEAR:** 这是一个比较难以理解的参数，只需知道在计算半导体、金属与绝缘体下，这个参数影响很大。需要与 SIGMA 参数配合使用

```

1 # 未知材料
2 ISMEAR=0 且 SIGMA 取 0.03-0.05
3 # 半导体和绝缘体
4 ISMEAR=-5，在晶胞过大时推荐按未知材料设置
5 # 金属弛豫
6 ISMEAR=1 或 ISMEAR=2 且 SIGMA 通常取 0.2
7 注意：不要在已知材料为半导体或绝缘体下取 ISMEAR 大于 0
8 # 计算态密度（DOS）或金属静态
9 ISMEAR=-5

```

**10、SIGMA:** 默认为 0.2，与 ISMEAR 配合使用。规定 smearing 的宽度，单位 eV。

**11、SYSTEM:** 对计算并无影响，可输入任何非中文字符，主要作用用来标识或说明计算的体系。

说明：以上参数是计算中经常涉及到的核心参数，包括但不限于以上列出的参数，参数说明皆来自vaspwiki，更多详细查看中的 INCAR tags。([https://www.vasp.at/wiki/index.php/The\\_VASP\\_Manual](https://www.vasp.at/wiki/index.php/The_VASP_Manual))

以下是来自大师兄网站中的几点比较重要的说明：

(<https://www.bigbrosci.com/zh-cn/latest/LVASPTHW/Ex2.2/>)

#### 关于 INCAR 的说明：

1、INCAR 主要是告诉 VASP 要算什么以及计算的细节参数。

2、INCAR 准备的原则是：越简单越好，不知道的，不理解的就不往里面放。

3、INCAR 中有很多参数，改变参数来控制计算细节，从而得到我们期望的结果。所以，记住：Rubbish in, Rubbish out。自己算什么，有什么需要注意的事项，心里要有数。

4、网上很多人发 INCAR 的教程，列了一堆参数进去，甚至长达两页，有些连默认值也放在里面了，恨不得把所有参数全部放进去。不过，请记住：那些令人眼花缭乱的只是唬人装逼用的，在实际计算中一旦你熟悉了自己的体系，花里胡哨的描述都是垃圾，你根本不会去看。

5、熟知各个参数的数值意义，以及在不同计算任务中的设置至关重要。

6、有人会说，那么多参数，怎么能记得过来？放心，完全可以的。

7、一方面，对于特定的体系，所需要的参数不同。如果你的课题比较稳定，那么你应该不会接触太多的参数。比如你专注于体系的能带结构，那么算过渡态的参数你就不会需要。

8、另一方面，这些计算参数又不是一天两天能记住的，是一个长期积累的过程。随着时间的增长和个人经验的增加，更重要的是长时间地浏览官网内容，以及与其他人讨论，自然而然地就记住了。

#### 关于 INCAR 的注意事项：

1、# 后面写的是汉语，是给大家展示用的。大家在写 INCAR 的时候，最好不要把汉语写进去，有可能会出错！

2、# 号后面的内容对计算不会造成影响，主要作为注释用，解释这个参数干嘛用的，可以不用输入；

3、SYSTEM, ISMEAR, SIGMA 就是 INCAR 中的参数，每个参数后面跟着等号 = 以及它们的取值（不一定是数字）；

4、等号 (=) 前后可以有空格，也可以没有。ISMEAR=0 和 ISMEAR = 0, 以及 ISMEAR =0

是一样的，但是大师兄强烈建议：在写 INCAR 的时候 = 前后各空一个格，一方面比较美观，另一方面我们把这种格式从一开始就定死，后面修改就很方便。

5、写 INCAR 的时候，不要使用 tab，用空格替换 tab；

6、官网有些旧的文件可能用的不是 # 号，而是!，！可能会出错。最好的办法就是记住参数是干嘛用的，# 和！都不写。

7、O 和 0 不要弄混了。很多人把数字 0 误写成字母 O，或者反过来，导致计算出错。

```

1      SYSTEM = O atom      #
2      ISMEAR = 0          #
3      SIGMA = 0.01        #

```

**最后：做事不要刻板**，以上提及的参数不必要都要记住，但是学会去 Vasp 手册 (<https://www.vasp.at/wiki/index.php>) 查询更为重要，牢记不要加一些自己都不清楚的参数进去，当准备好 INCAR 文件后，整个 Vasp 的学习其实也完成了 50% 了。

### 2.2.3 POSCAR

主要用于解决两个问题：1、指定计算哪种原子。2、原子坐标以及晶格大小。

如何形象理解 POSCAR：如果见过魔方，可以把晶胞想象成它的样子，其中每一个魔方块就可以看作一个原子位点，这样，以一个魔方块为原点 (0, 0) 作笛卡尔坐标轴，我们就可以数学抽象出每个原子位点以及整个魔方块的大小数字，而 POSCAR 中存储的即为这种信息。

文本文件内基本格式如下：

```

1      Ti1 Al1
2
3      1.0
4      3.179    0.00    0.00
5      0.00     3.17    0.00
6      0.00     0.00    3.17
7
8      Ti Al
9      1 1
10     direct
11     0.00    0.00    0.00
12     0.50    0.50    0.50

```

下面对于各行进行说明：

第 1 行：随意书写，可备注为何种体系。

第 2 行：缩放系数

第 3-5 行：分别为晶胞 xyz 的大小数值，可看作组成三维空间的三个基（基准向量），表中所表示三个向量，分别为 (3.17, 0, 0), (0, 3.17, 0), (0, 0, 3.17)。

第 6 行：所算原子符号，上表中为 Ti、Al，注意前后位序且不可写错。

第 7 行：第 6 行分别的原子个数，注意一一对应。

第 8 行：原子的坐标表示方法，分为笛卡尔 (Cartesian) 坐标轴和分数 (direct) 坐标轴，笛卡尔坐标轴范围为 0 到晶胞 3-5 行中规定的晶胞最大值单位为埃，分数坐标轴范围为 0-1 无量纲为百分率。

第 9 行：第 6 行标识的第一个原子 Ti 在晶胞中的位置。

第 10 行：第 6 行标识的第二个原子 Al 在晶胞中的位置。

大师兄网站：<https://www.bigbrosci.com/zh-cn/latest/LVASPTHW/Ex2.4/>

### 2.2.4 POTCAR

上文所述，POSCAR 文件的第 6 行，我们对所需要计算的原子是什么进行了标识，但是所用来描述原子内部各种性质的信息远不仅如此，POTCAR 就是用来书写原子详细信息的文件。以下我们称作其为赝势。鉴于赝势的复杂性我们只需知道：

1、已经在电脑中以文本文件的形式给出，即所需数据已经给出，直接“复制粘贴”进去就可以使用。

2、所给出的赝势以单种原子的形式存放在一个文本文件中，需要我们根据 **POSCAR 第 6 行的原子先后顺序** 进行“复制粘贴”到一个文件中（即 POTCAR）。

3、因为赝势文本量比较大，在 Linux 操作系统中我们有一套便捷的方式进行“复制粘贴”。

根据第 3 条，我们仅需要学习 Linux 中的一个小指令：cat

假设赝势文件存放在 PBE 文件夹下，在 Terminal 中输入基本格式如下：

```
1 cat PBE\Ti PBE\Al > POTCAR
```

**注意：** 1、要用 cat 命令形成 POTCAR 文件，且严格按照 POSCAR 第 6 行顺序进行。

2、以上只给出 cat 命令的一个形式，具体学习望发挥主观能动性，善用各种搜索引擎，多次尝试。

3、常用 PBE 蕴势，严格按 vasp 官网给出的 **List of PBE potentials** 加粗的赝势选择使用。

（见 [https://www.vasp.at/wiki/index.php/Available\\_pseudopotentials](https://www.vasp.at/wiki/index.php/Available_pseudopotentials)）（大师兄网站：<https://www.bigbrosci.com/zh-cn/latest/LVASPTHW/Ex2.5/>）

4、cat 后的“Ti Al”为路径，到时候 PBE 蕴势库在哪个路径下就要换成机器中的路径！

### 2.2.5 KPOINTS

KPOINTS 这个文件格式如下：

```
1 KPOINTS
2
3 0
4 Gamma
5 7 7 7
```

第 1 行：类似于 POSCAR 第一行随便写，非中文即可。

第 2 行：0 代表自动生成 k 点，至于 k 点是什么没有学固体物理前不必了解。

第 3 行：一种生成网格的方法，有 Monkhorst-Pack 和 Gamma 两种，可固定使用 Gamma。不必深入理解。

第 4 行：生成网格的密度，分别标识 xyz 方向的各密度，一般数字越大越数据越精确，计算时间越长，尽量保持精度和时间的平衡。说明：第 4 行数据是我们最需要关注的数据，也是可能进行修改的数据，其余行一般不进行变化，由于 K 点的具体物理含义需要有一定的理论基础，故不用细致理解。

### 2.2.6 任务提交

恭喜你！如果已经将上述四个文件大致了解，且已经在一个单独的文件夹中准备好了，那只需在此文件目录中的空白处右击进入 Terminal 输入：

```
1 mpirun -np 72 vasp_std
```

就可以等待完成了。

### 2.2.7 说明

1、大师兄网站以及其他教程的学习，需要有一定基础，至少是从熟练运用 Linux 操作系统开始，对于 vasp 的编译、安装开始，逐步到了解运行 vasp 的所必要的四个文件，过程中不免要牵扯到包括不限于 bash 等语言，这就提出了一定的编程能力与语法阅读能力。但是，**如果不会这些东西，就真的没办法用 vasp 进行计算吗？**

事实上，不会编程依然能够去进行 vasp 的一些简单计算，得益于 ubantu 可视化操作窗口，我们完全可以套用 Windows 操作系统的习惯。最后能正确的准备好四个输入文件：INCAR、POSCAR、POTCAR、KPOINTS，学会 2.2.6 提交任务的命令以及 cat 语句的基本使用方法即可。

**2、为什么别的教程很注重编程的学习，而上述所有东西只涉及到 cat 语句以及任务提交命令？**

工欲善其事必先利其器，同样是砍一棵树，不一定非要用伐木机砍树，在有限时间内，我们完全可以用手头的工具也能砍下一棵树来，区别就是工作时间与效率问题。万事开头难，编程是进行模拟计算的利器，对于本科论文而言，我更推荐用好自己已经会的知识，在用中学，学中用，若有空闲时间以及感兴趣自然可以根据大师兄网站的流程，对于代码进行复现。如果不感兴趣也不必强求，此后遇到问题再去专门补齐用到的知识即可。

**3、我已经明白了上述文件的具体含义，但是这四种文件具体怎么准备？**

POTCAR：用 cat 命令在老师已给的 PBE 质势库中按 POSCAR 第 6 行的前后顺序导入一个文件中，命名为 POTCAR。**切记！顺序一定要正确，例如，POSCAR 第 6 行为：TiAl，那么 cat 命令为 cat Ti Al > POTCAR。**

INCAR 以及 KPOINTS：因为所包含内容很少，推荐第一次用键盘敲出来即可。**切记！INCAR 不要胡乱在网上复制，尽量只往里面放入了解的参数。**

POSCAR：这里提供两种方法：(1)、对于面心立方及体心立方等简单体系，完全可以根据实验值（文献查询），自己键盘输入。(2)、也可以在晶体学网站中，下载自己所需体系进行计算，但是注意晶体学网站中可能不包含一些金属间化合物。

(晶体学网站：<https://next-gen.materialsproject.org/>)

**4、什么是弛豫、静态计算？**

在上述描述中，不免提到弛豫、静态等词。这与 ISIF 这个参数息息相关，在 2.2.2 中我们可以发现在 ISIF=2 是只进行原子位置的变化，而不进行晶胞形状体积的变化。这就是静态计算。而 ISIF=3，无论原子位置还是晶胞形状体积皆发生变化，这就是弛豫。例如，如果将晶胞看作一个魔方，ISIF=2 下，仅魔方块位点进行移动找到合适的位点，整个魔方形状大小不会发生变化。

**5、什么是离子步、电子步？**

这是我们在 INCAR 中控制收敛的两个标准，当离子步与电子步同时达到设定的收敛标准时才认定为系统完全收敛，计算结束。

**6、什么是切面、扩胞？**

切面：即使得整块材料的模型构建表现为某一指定方向（晶向）的表面裸露，其他方向均为周期排列状态无裸露在外的表面。这给我们研究表界面性质提供方便。

扩胞：即使得原本是原胞的晶体按所需要的整数倍在空间中进行单纯的个数变多，不改变本来原胞的晶格参数，原子间距，仅改变空间中可视存在的原子数目。这给我们研究掺杂等提供方便。

对于切面、扩胞等操作都可以在 2.1.3 中提到的 Material Studio 中进行操作，具体流程自行查看哔哩哔哩及百度。

### 7、什么是 bulk、slab?

bulk：即体系所存在的三维空间中，任意方向都是无限周期循环的状态，并无某方向存在表面等空缺。例如，原胞可称作 bulk 或块体。

slab：通常在研究界表面性质时要将某一方向暴露在空间中，这就需要对 bulk 进行切面，切面后的模型就被称为 slab。

### 8、四个必要文件的名称可以随意改变吗？

vasp 只识别以 INCAR、POSCAR、KPOINTS、POTCAR 命名的文本文档，运行时候缺一不可，名称一个字都不能变！

## 2.3 数据提取

### 2.3.1 OUTCAR

经过一段时间，当我们运行的计算结束后，就会发现原本目录下多了许多其他文本文件，无非是 2.2.1 中列举的，除四个输入文件外，其余文件皆为输出文件，而其中最常用的就是 OUTCAR 文件。

OUTCAR 包括如下内容：输入参数摘要、电子步等信息、应力张量与原子受力、局域电荷以及磁矩、介电性能等。这么多信息在对于简单体系的计算后实际上常常关注体系能量。

### 2.3.2 grep 语句

我们如何在杂乱的 OUTCAR 输出数据中提取到我们想要的信息？那么就要学习第二个 Linux 操作系统语句：grep。以提取体系能量为例，只需在已经运行完成的目录窗口下右击空白处，调出 Terminal 输入：

```
1      grep ' without ' OUTCAR
```

grep：语句标识。在”中即为想要提取的内容。OUTCAR：提取的对象名称。格式为：

```
1      grep '想提取的内容' 提取目标的名称
```

注意：“不可省略。

### 2.3.3 说明

1、其实，完成从 2.1 到 2.3 的流程，就已经实现的对于计算进行一个输入-运行-输出的过程，像是单纯获得一个模型的总能此类的工作已经可以宣告结束。

2、从上述中，我们已经看出，并不需要我们需要有多大的对于编程、晶体学的理解，很多时候我们都是将其作为黑箱来使用的。其实这种“黑箱操作”在我们现实中的例子比比皆是，是一种我们没有意识到，但每时每刻都在发生的过程。例如，我们使用电脑并不需要熟知底层逻辑；用微

波炉不用知道微波工作机理；思考问题不必熟知神经突触介质的传导……诚然，底层逻辑可以指导工作，给予巨大的帮助，但是，只是工具的使用者依然可以很好的利用工具。千百万年前，我们的祖先，也没傻到先去推理为什么更尖锐的物品能够更轻易的划开野兽的皮毛，再去使用这一特性。所以，有时候保持不求甚解也是行得通的（笑）。

3、在计算前后过程中，难免遇到很多问题，无论是粗心大意还是知识性问题，最好是寻求自我独立解决的方法。如今，无论是百度等搜索引擎，还是 ChatGpt 等 AI，都有常人无所匹敌的知识含量，请善用此类工具。实在解决不了的办法再去找老师进行讨论。

4、时间一长，所学知识都会逐渐遗忘，而留下的是看待问题的高度、对待问题的态度和解决问题的能力，将是一生的财富。

5、注意具体问题具体分析，所有的经验在未成为真理前都具有其限制以及时效性，使用时多次尝试，不要刻板。

### 3 基本实例计算

#### 3.1 单点能计算

以面心立方的 Al 为例计算其静态下的总能量。

##### 3.1.1 准备 POSCAR

方法一：

1、进入晶体学网站 (<https://next-gen.materialsproject.org/>)，如果是第一次使用，需要点击 Login 进行登录，站点非国内网站载入速度较慢。

2、登陆后，点击 Start Exploring Materials，在跳出的输入框，输入 Al 或在下方元素周期表中选择元素，点击 search。

3、选择第一顺位的那一项，标识 Crystal System(晶系)为 Cubic(立方)，Space Group Symbol(空间群)为 Fm3m。

4、接下来跳出的界面中，居中存在一个晶体可视化结构模型，其左上角五个图标中最后一个图标可将此晶体下载到本地电脑。点击后会让我们选择以 CIF、POSCAR、JSON、VASP Input Set 等形式下载，为方便演示这里以 VASP Input Set 下载，如果只需要模型下载 POSCAR 一项即可。

5、下载完成后会以 Al MPRelaxSet.zip 压缩包形式存在本地（若是 POSCAR，则以 Al.poscar 文本文档形式存储），压缩包解压后，即为我们所需的四种输入文件。

6、将 POSCAR 文件用文本软件打开（推荐 Notepad++），如下：

```
1          Al4
2
3          1.0
4          4.038    0.00    0.00
5          0.00     4.038    0.00
6          0.00     0.00     4.038
7
8          Al
9          4
10         direct
11         0.00     0.00    0.00 Al0+
12         0.00     0.50    0.50 Al0+
```

```

11      0.50    0.00    0.50 Al0+
12      0.50    0.50    0.00 Al0+

```

(格式原因演示只保留两位小数) 7、至此，POSCAR 准备完毕。

方法二：

由于面心立方 Bulk 所含信息较少，完全可以查询文献实验值后，进行手动输入即可。

方法三 (来自大师兄网站)：

1、MS 导出 Al 原胞的模型 (Material Studio):

A) 左上角 file—> newproject，输入 Al，点 OK;

B) file—> import—> Structures—>metals—>pure-metals—> 找到 Al，选择打开即可；

C) file—>export—> 选择 cif 格式 (此时导出的是 Al 的单胞，conventional cell)

2、MS 导出 Al 原胞的模型：

A) 和 B) 步骤与前面一样，

C) Build —> Symmetry—> 选择 primitive cell

D) file—>export—> 选择 cif 格式 (此时导出的是 Al 的原胞，primitive cell)

3、VESTA 转换成 VASP 的格式 (注：将 CIF 转化为 POSCAR 这种操作常用)

a) 打开 VESTA 软件，file—>open—> 选择之前保存的 cif 文件

b) file—> export data —> 保存类型选择 VASP

c) 将保存的文件重新命名为 POSCAR 即可

### 3.1.2 准备 INCAR

在 3.1.1 中晶体学网站中下载的 VASP Input Set 包含 INCAR 文件，但是我们要进行的是单点运算，所以需要根据 2.2.2 中参数含义对以下内容进行修改：

```

1      ALGO = Fast
2      EDIFF = 0.0002
3      ENCUT = 520.0
4      IBRION = 2
5      ISIF = 3
6      ISMEAR = -5
7      ISPIN = 2
8      LASPH = True
9      LORBIT = 11
10     LREAL = Auto
11     LWAVE = False
12     MAGMOM = 4*0.6
13     NELM = 100
14     NSW = 99
15     PREC = Accurate
16     SIGMA = 0.05

```

删去我们不熟悉的参数：ALGO、ISPIN、LASPH、LORBIT、LREAL、LWAVE、MAGMOM、PREC，注意不存在的必要参数 vasp 以默认值运行。

```

1      SYSTEM = Al #添加标识，可随意输入
2      EDIFF = 0.0001 #精度变大
3      ENCUT = 520.0
4      #IBRION = 2 #静态计算注释
5      ISIF = 2 #因为单点计算，由3改为2
6      ISMEAR = -5 #金属静态计算不变
7      NELM = 100 #最小电子步不变
8      #NSW = 99 #静态离子步注释
9      SIGMA = 0.05 #不变

```

文件名重新命名为 INCAR，准备完毕。

### 3.1.3 准备 KPOINTS

由 2.2.5 可知，这部分内容较少完全可以由自己手动输入：

```

1      Al
2      0
3      Gamma
4      12 12 12

```

第 4 行  $12 \times 12 \times 12$ ，因为本次原子数较少可以适当增大，当考虑原子量较大体系时，需要平衡精度与计算时间。

### 3.1.4 准备 POTCAR

1、见 POSCAR 第 6 行仅 Al 一种原子，查询 vaspwiki 选择 List of PBE potentials 展开。加粗字体为合适赝势，Al 为 **Al**。

([https://www.vasp.at/wiki/index.php/Available\\_pseudopotentials](https://www.vasp.at/wiki/index.php/Available_pseudopotentials))

2、见 POSCAR 第 6 行仅 Al 一种原子，在计算文件夹下使用 2.2.4 中的 cat 命令：

```
1      cat PBE\Al > POTCAR
```

注意：cat 后面一项内容为你此时的赝势库路径，这里仅为演示。

### 3.1.5 进行计算

将以上备好的 POSCAR、KPOINTS、POTCAR、INCAR 放入同一文件夹下，注意此文件夹下仅此四种文件，且只能如此命名。进入此路径的 Terminal 窗口输入 2.2.6 命令：

```
1      mpirun -np 72 vasp_std
```

即可等待完成。

### 3.1.6 数据提取与收集

观察运行数据并无报错（主要是过程中出现的 WARNING），Terminal 窗口出现可输入光标时，运行结束。

使用 2.3.2 中 grep 语句：

```
1      grep ' without' OUTCAR
```

即为单点能，将能量记录，流程结束。

## 3.2 结构优化

以面心立方的 Al 为例计算对其进行结构优化，所谓结构优化就是寻求此体系下，最稳定的原子位置与晶胞大小形状。在计算模拟中，一般表现为收敛到设定的收敛标准，体系总能量为最小值。

### 3.2.1 准备四种基本输入

对于结构优化，我们只需在静态计算的文件基础上对于 INCAR 参数进行调整即可，POTCAR、POSCAR、KPOINTS 可沿用静态计算中已经准备好的。根据 2.2.2 中参数含义对 INCAR 进行修改：

```
1      SYSTEM = Al #标识，可随意输入
2      EDIFF = 0.0001 #精度变大
3      ENCUT = 520.0
4      IBRION = 2 #结构优化，选共轭梯度
5      ISIF = 3 #因为结构优化，修改为3
6      ISMEAR = 2 #金属结构优化计算
7      SIGMA = 0.2#配合ISMEAR
8      NELM = 100 #最小电子步不变
9      NSW = 99 #最小离子步
10     EDIFFG = -0.01 #离子步收敛标准
```

### 3.2.2 进行计算

运行命令见 3.1.5，不同于静态计算，当结构优化结束后，最后一行数据为：

```
1      reached required accuracy
2      - stopping structural energy
3      minimisation
4      writing wavefunctions
```

### 3.2.3 数据收集以及 CONTCAR

当结构优化完毕后，同 3.1.6 可将能量进行提取。然而，既然已经进行了结构优化，那么，优化后的结构数据在哪里？结构的输入文件即 POSCAR，而结构的输出文件为 CONTCAR，可将其看作输出形式的 POSCAR，这也是我们要涉及到的第二个输出文件。

CONTCAR 的格式与 POSCAR 相同：

```
1      这里加以下结构优化后的 CONTCAR
```

### 3.2.4 说明

1、对于结构优化我们不仅要关注收敛情况、体系能量，也要对结构优化的目的：“获得可稳定存在的结构。”有一定认识。CONTCAR 即为计算结束后的稳定存在的结构的数据。

2、CONTCAR 格式与 POSCAR 几乎一模一样，这也就是说我们将 CONTCAR 重命名为 POSCAR 就可以作为无论是静态计算、扩胞、切面等接续操作的基础。

3、结构优化后的结构为以后计算的基础，这就对精度有比较高的要求，超高的可信度是今后所有以此为基础的计算正确性的保证，一般我们对此会使用 BM 方程二项式拟合的方式获得精准的晶格参数。

## 3.3 拟合方程

对于结构优化得到平衡的体系我们一般有两种方法得到：

第一种即 3.2，在算力足够的前提下，我们选择适当的初始体系（POSCAR），晶格参数一般为实验值，在 INCAR 中选择 ISIF=3，提高电子步以及离子步收敛标准，进行结构优化即可得到平衡的体系。

第二种即二项式拟合，这需要涉及到另一个语句：for，需要一定的编程能力，且要对 BM 方程的有初步的理解。

### 3.3.1 什么是 BM 方程？\*

简而言之，BM 方程是 Birch 和 Murnaghan 在 1947 年提出的一个等温情况下的体积与压力关系的一个方程。三阶形式如下：

$$P(V) = \frac{3B_0}{A} \left[ \left( \frac{V_0}{V} \right)^{\frac{7}{3}} - \left( \frac{V_0}{V} \right)^{\frac{5}{3}} \right] \left\{ 1 + \frac{3}{4}(B'_0 - 4) \left[ \left( \frac{V_0}{V} \right)^{\frac{2}{3}} \right] - 1 \right\}. \quad (1)$$

上式中： $P$  为压力， $V_0$  为初始的体积， $V$  为形变后的体积， $B_0$  为体积模量， $B'_0$  为体积模量的导数。 $B'_0$ 、 $B_0$  皆为经验常数：

$$B_0 = -V \left( \frac{\partial P}{\partial V} \right)_{P=0} \quad (2)$$

$$B'_0 = \left( \frac{\partial B}{\partial P} \right)_{P=0} \quad (3)$$

(1) 式对  $V$  积分：

$$E(V) = \int P dV \quad (4)$$

得：

$$E(V) = E_0 + \frac{9B_0V_0}{16} \left\{ \left[ \left( \frac{V_0}{V} \right)^2 - 1 \right]^3 B'_0 + \left[ \left( \frac{V_0}{V} \right)^2 - 1 \right]^2 \left[ 6 - 4 \left( \frac{V_0}{V} \right)^2 \right] \right\} \quad (5)$$

实际上，当我们看到如上推导还是会很懵的，其实我们只需关注式 (5)，除  $E$  外其余为常数，即表示能量 ( $E$ ) 与体积 ( $V$ ) 的关系，其中  $V_0$  为初始体积。

### 3.3.2 什么是二项式？

设晶格常数为  $a$ , 修改式 (5) 得:

$$E(V) = E_0 + \frac{9B_0V_0}{16} \left\{ \left[ \left( \frac{a_0}{a} \right)^2 - 1 \right]^3 B'_0 + \left[ \left( \frac{a_0}{a} \right)^2 - 1 \right]^2 \left[ 6 - 4 \left( \frac{a_0}{a} \right)^2 \right] \right\} \quad (6)$$

若令  $x = (1/a)^2$ , 式 (6) 可即为如下形式:

$$E(x) = c_0 + c_1x + c_2x^2 + c_3x^3 \quad (7)$$

称其为: 二项式。根据微积分基本的求极值方法: 对式 (7) 求导:

$$\frac{dE}{dx} = c_1 + 2c_2x + 3c_3x^2 \quad (8)$$

令  $\frac{dE}{dx} = 0$ , 根据求根公式:

$$x = \frac{-2c_2 \pm \sqrt{(2c_2)^2 - 4 \times 3c_3 \times C_1}}{2 \times 3C_3} \quad (9)$$

舍去负值即可得曲线极小点, 在本例中  $x = (1/a)^2$ , 反解得极小点处的  $a$ 。

然而, 为什么要求极小点处的  $a$ ?

我们已知总能量最小时体系最稳定, 且能量 (E) 为 Y 轴, 二次曲线的极小值即为曲线 Y 轴最小值, 也即能量最小值, 此时的  $a$  即为总能最小时的晶格参数。

### 3.3.3 什么是拟合?

百度百科对拟合是这样解释的: 形象的说, 拟合就是把平面上一系列的点, 用一条光滑的曲线连接起来。因为这条曲线有无数种可能, 从而有各种拟合方法。拟合的曲线一般可以用函数表示, 根据这个函数的不同有不同的拟合名字。

通俗来讲就是: 用一系列的点去寻找最接近的曲线, 而曲线就可以用方程形式表示出来。也可以进一步理解为通过已知  $xy$  两解求方程参数, 进而得出方程表达式。

故而要将 3.3.2 步骤进行拟合就要找很多很多  $x$  和  $y$ , 即上例中的能量 (E) 与晶格参数 ( $a$ ), 通过点拟合出方程, 这即为拟合的形象理解。

### 3.3.4 如何得到许多晶格参数 ( $a$ ) 与能量 (E)?

在 3.1 中, 我们进行了单点能的计算, 输入参数中有晶格参数 ( $a$ ), 即 POSCAR 的第 3-5 行。计算结果文件中, 根据 2.3.2 可提取能量 (E)。那么是不是进行许多个静态计算就可以得出许多晶格参数 ( $a$ ) 与能量 (E)。

答案是肯定的, 那么我们要以什么原则得到这一组数据呢? 即通过改变 **POSCAR** 中的晶格参数来得到不同的能量作为要拟合成曲线的各个点。

当我们取点时, 要遵循均匀的原则, 即在初始晶格参数的上下各取同样的点。在 POSCAR 第 2 行为缩放系数, 其作用就是对第 3-5 行的晶格参数进行调控, 缩放系数为 1.0 就表示此 POSCAR 最后的晶格参数为  $a*1.0$ , 由此就可以为我们取点提供思路, 仅去修改每个 POSCAR 缩放系数的大小, 例如取 0.95-1.05, 间隔为 0.01, 就可以均匀的取好 11 个点了。

注意: 除 **POSCAR** 的缩放系数外其余三个输入文件均保持不变。

### 3.4 批量操作

在 3.3.4 中，我们或许会发现如果手动去进行 POSCAR 中缩放系数的修改以及取 11 个点的静态计算，不仅要创建 11 个文件夹，还要对 11 个文件夹中的每个 POSCAR 进行类似的操作，在计算时当一个文件夹中的静态计算完成时也要手动切换到另一个文件夹中，这就使得我们的精力大量的浪费在了无意义的重复劳动之中。

那么如何去改变这种无意义的重复工作呢？曾记得大师兄网站的教程中有一句话令人印象深刻：如果想不去动手，那便多去思考。而 Linux 中的 bash 就给我们提供了一个平台，去进行一些简单的语句命令从重复劳动的苦海中脱离。

#### 3.4.1 for 循环

在 2.3.2 中提到的 grep 语句即为 bash 的应用。如果在之前多少接触过 C 语言或者其他任何语言，一定会知道程序中存在顺序、循环和选择三种结构，而循环结构中常用的语句便是 for。当然，如果没有接触过任何计算机语言，那么也不要担心，通过如下的文字便可很轻松的理解何为 for 循环：

1、在 Linux 系统中创建一个新文件夹，进入新文件夹，打开此目录下的 Terminal，输入：

```
1     for i in 1 2 3;do mkdir $i;done
```

按回车。

2、如果不意外，我们将会在本来什么都不存在的目录下发现三个分别名为 1、2、3 的文件夹。

3、这便是 for 循环可以实现的功能，接下来将给出 1 中代码块的伪代码表达式：

```
1     for 变量 in 值1 值2 值3;do mkdir $变量;
2         done
```

4、for 标示着 for 循环开始；其中变量可以是任何英文文字；in 后的值可以是任何英文文字，且不限制个数，亦可理解为有几个值便循环几次；do 后便为循环的操作描述；done 为 for 循环结束。

5、如果还不理解，请看下面例子：你手边有三个苹果，请用 for 循环将三个苹果放进篮子里，伪代码如下：

```
1     for 苹果 in 1号 2号 3号;
2         do 将$苹果放进篮子中;done
```

6、恭喜你，已经学会 for 循环了，可以随意练习一下！

7、上述出现的 mkdir 表示创建文件，是与 grep 一样，皆为 bash 的用法，创建一个新文件在 Terminal 中可输入：

```
1     mkdir 文件名称
```

具有与用鼠标新建文件夹的相同功能。

#### 3.4.2 sed 语句

在 3.3.4 中，我们涉及到要对 POSCAR 的缩放系数进行修改，在一般操作中，我们用鼠标打开 POSCAR，光标移到 POSCAR 第 2 行，即可完成修改。然而，当我们用 for 循环时，如何用 bash 实现相同的修改行为呢？

请在已经准备好 POSCAR 文件的文件夹中进入 Terminal，并在 Terminal 中输入：

```
1      sed -i '2s/1.0/0.95/g' POSCAR
```

如果不出意外，完成上述操作后，打开 POSCAR 会发现第 2 行的 1.0 已然被替换成 0.95。然而，上述 sed 语句的含义是什么？

请看以下解释：

- 1、你面前有一个很多抽屉的柜子，请用 sed 语句将第 2 个抽屉中的苹果换成梨。
- 2、于是你输入以下伪代码，并按回车：

```
1      sed -i '第2个抽屉/苹果/梨/g' 柜子
```

3、打开柜子第 2 个抽屉，发现苹果果然换成了梨，这真要比手动打开柜子，手动替换要方便且神奇（笑）。

### 3.4.3 混合使用 (1) 之 for+sed

因为动过了脑子尝到了甜头，所以更不想进行纯体力劳动，所以立志要用 bash 把所有柜子的第 2 个抽屉中的苹果全换成梨，于是：

```
1      for 柜子 in 1号 2号 3号;
2          do sed -i '第2个抽屉/苹果/梨/g' $柜子;
3              done
```

你可以很方便的对同一文件夹中的文件进行批量替换了，例如本文件夹下有以 1、2、3 命名的 POSCAR 文件，我想将他们的第 2 行的 1.0 都改成 0.95，于是在 Terminal 中输入：

```
1      for i in 1 2 3;
2          do sed -i '2s/1.0/0.95/g' $i;
3              done
```

### 3.4.4 路径

然而在实际应用中，我们发现我们需要更改的 POSCAR 各属于不同的文件夹。例如，我们的本目录下存在着名称分别为 0.95-1.05 各间隔 0.01 命名的文件夹，共 11 个文件夹，每个文件夹下存在着一个 POSCAR。

那么，(1)、我们如何将各个文件夹下的 POSCAR 的第 2 行的 1.0 也进行替换成 0.95 呢？

在 2.2.4 中注意：4 中，说：cat 后的“Ti Al”为路径，到时候 PBE 质量库在哪个路径下就要换成机器中的路径！这里所说路径例如：

```
1      C:\texworks\POSCAR
```

就表示 C 盘中 texworks 文件夹里面 POSCAR 的路径，若是上述存在 11 个文件夹的情境中。

请问，(2)、我们如何将名为 0.95 的文件夹下的 POSCAR 的第 2 行的 1.0 进行替换成 0.95 呢？

那就需要我们在此目录下的 Terminal 中输入：

```
1      sed -i '2s/1.0/0.95/g' 0.95\POSCAR
```

所谓 0.95/POSCAR 便为此时的路径。

于是请问，(3)、我们如何解决(1)中的问题呢？

就需要我们对 3.4.3 混合使用中的代码块进行改写：

```
1     for i in *;
2         do sed -i '2s/1.0/0.95/g' $i/POSCAR;
3             done
```

观察以上得知与 3.4.3 有两点差异：(1)、sed 语句的最后一项被写成了路径（其实本来的也是文件路径）；(2)、for 的 in 后面的赋值操作修改为 \*。(1) 中问题通过以上说明我们已经能够理解，而(2) 中的 \* 表达什么？

这是因为我们要对 11 个文件夹进行其中的文件内容修改，难道我们要一个一个的打出它们的名称吗？如果要修改的文件是 100 个、1000 个呢？针对这个问题，我们的 bash 可以用 \* 来表示本目录下所有的文件夹及文件。

#### 3.4.5 pwd 语句

再 3.3.4 中我们明白了什么是路径，那么如何用 bash 在 Terminal 中获取路径呢？我们只需输入：

```
1     pwd
```

即可得到本目录的路径！

#### 3.4.6 混合使用(2)之 for+sed

我们使用如下代码块，完成了将本目录下所有文件夹里面的 POSCAR 的第 2 行的 1.0 修改为 0.95 的操作。然而，我们要求将 POSCAR 的第 2 行的 1.0 分别修改为 0.95-1.05 各自不相同的数字这样怎么办呢？

上述曾提到：本目录下存在着名称分别为 0.95-1.05 各间隔 0.01 命名的文件夹，共 11 个文件。也即，本目录下的文件夹名称即为我们想要修改成的内容，那么我们是否可以进行如下操作：

```
1     for i in *;
2         do sed -i '2s/1.0/$i/g' $i/POSCAR;
3             done
```

观察以上代码块，我们仅将 0.95 修改为 \$i，标志着每次循环取 i 的值代替 1.0，而我们 i 的值正好为想要修改的数据，完美！

#### 3.4.7 混合使用(3)之 for+mkdir

上述提到 for 循环以及 mkdir 语句，我们已经了解其含义，请问如果创建名为 0.95-1.05 间隔为 0.01 命名的文件夹，如何用 bash 实现？

我们可以：

```
1     for i in {95..99};
2         do mkdir 0."$i";
3             done
```

与：

```
1      for i in {00..05};
2          do mkdir 1."$i";
3      done
```

其中的 00..05 代表 00、01、02、03、04、05；同样如果是 000..100 就代表从 000 到 100 共 101 个数字。

#### 3.4.8 cd 语句

随着进步深入，我们当逐渐习惯使用 bash 代替鼠标操作后，发现很多操作要在进入一个文件夹后才可以对文件进行操作，那么在 bash 中什么语句可以起到进入一个文件夹以及退出的操作呢？

我们可以在 Terminal 中输入：

```
1      cd 文件夹名称
```

于是我们在 Terminal 窗口中便进入了“文件夹名称”的文件夹中。但是如何退出这个文件夹进入上一级呢，同样输入：

```
1      cd ..
```

注意：cd 后输入一个空格后再输入.. 再进而，我们想要进入路径为/home/1.00 的文件夹中呢？输入：

```
1      cd /home/1.00
```

#### 3.4.9 混合使用 (4) 之 for+vasp 运行

现在我们以及了解 for 循环、cd 语句、vasp 运行命令等，请将本目录下准备好的 0.95-1.05 共 11 个静态计算用 bash 一次性运行完毕。

伪代码形式：

```
1      for i in *;
2          do 进入文件夹i; 进行计算 ;
3              退出文件夹i;
4      done
```

在 Terminal 中输入：

```
1      for i in *;
2          do cd $i;mpirun -np 72 vasp_std;
3              cd ..;
4          done
```

#### 3.4.10 echo 语句

如果学过 C 语言（没学过也没关系），那么所学的第一个语法结构必然是 Hello, world！那么如何将 Hello, world！显示到屏幕上呢？C 语言中只需输入：

```
1     printf(hello,world!);
```

而在 Linux 中，用 echo 语句可以实现同样的功能：

```
1     echo hello,world!
```

echo 语句在我们数据提取中有重要作用！

### 3.4.11 混合使用 (5) 之 for+grep+echo

恭喜！通过以上操作，我们成功的进行了缩放系数不同的 11 个静态计算，那么请问如何使用已知的命令将能量以及缩放系数数据提出到一个文件中？这就需要 for 循环、grep 和 echo 的配合使用。伪代码为：

```
1         for i in *;
2             do 提取 i 中的 OUTCAR 能量到 data 文件 ;
3                 done
```

在本目录下进入 Terminal，输入：

```
1         for i in *;
2             do echo -e $i "\t"
3                 $(grep 'without' $i/OUTCAR);
4                 done
```

我们会发现每一个文件夹中 OUTCAR 的能量都显示在屏幕中了，那么如何使其输入到一个空白文件中呢，只需在：

```
1         for i in *;
2             do echo -e $i "\t"
3                 $(grep 'without' $i/OUTCAR);
4                 done >> data;
```

以上各自的含义：

```
1         for i in *;
2             do echo -e i 的取值 换行符
3                 $(grep '要取的内容' 目标文件路径);
4                 done 输入到 随意文件名称;
```

这样，我们就发现目录下出现一个名为 data 的文本文件其中包含着我们所需的数据。

### 3.4.12 说明

1、以上所有代码块在 Terminal 中输入时，都不输入任何回车键，上述代码块在描述时，为方便起见，有时会另起一行。

2、上述 3.4 所有内容围绕着如何拟合出一个可以置信的晶格参数做出一系列不同晶格参数（通过改变缩放系数）下的静态计算，同时串联了在 VASP 计算中常用到的许多语句与些许编程，在熟悉语句的同时，尽量理解，起初陌生，大量练习后自然会熟络。

3、一个小技巧，在 Terminal 中按↑，即可得到上次所输入的数据内容。

## 3.5 拟合的代码实现

### 3.5.1 计算流程

在 3.4 中我们已经得出了拟合所需的基本的输入参数，根据 3.3 中的计算流程，我们已经知道只需输入以晶格参数与能量为 xy 轴的取的点，便可得知以下三次二项式：

$$E(x) = c_0 + c_1x + c_2x^2 + c_3x^3 \quad (10)$$

中的系数  $c_0, c_1, c_2, c_3$ 。观察：

$$\frac{dE}{dx} = c_1 + 2c_2x + 3c_3x^2 \quad (11)$$

中的系数  $c_1, c_2, c_3$  不变。也即当  $\frac{dE}{dx} = 0$  时，我们可以用系数  $c_1, c_2, c_3$  根据求根公式解出 x，在 3.3.2 中我们定义  $x = (1/a)^2$  反解即可求出 a。

### 3.5.2 代码实现之 python

然而，如何用点求出方程的系数呢？

使用 python 的 math 与 numpy 计算库。

1、在 data(即 3.4.11 已经算好的数据) 所在的目录中新建文本文件。

2、打开后输入：

```

1 import math
2 import numpy as np
3
4 b,E = np.loadtxt('data', usecols =
5 (0,1), delimiter='\t', unpack = True)
6
7 x = (b*4.03)**(-2)#注意
8
9 p = np.polyfit(x,E,3)
10
11 c0 = p[3]
12 c1 = p[2]
13 c2 = p[1]
14 c3 = p[0]
15
16 x = (math.sqrt(4*c2**2 - 12*c1*c3)
17 - 2*c2)/(6*c3)
18
19 a = 1/math.sqrt(x)
20
21 print 'a = ' %(a)

```

### 3.5.3 代码解释

1-2 行声明用 math 和 numpy 库，且把 numpy 命名为 np 方便使用。

4-5 行使用 numpy 库在名为 data 的文件中读取数据，即：

```

1 b,E = np.loadtxt('要读取的文件名称' , 使用0和1两列 ,
2 加载分隔符 , 把每一列当成一个向量输出不合并)

```

7 行即令:  $x = (b * 4.03)^{-2}$  注意这里的 4.03 代表缩放系数 1.0 时候的晶格参数, 按自己静态计算下缩放系数 1.0 下的 POSCAR 第 3-5 行的数据进行修改。b 代表数据记录的缩放系数, 见 3.3.2。

第 9 行根据输入的 data 中的数据进行使用 numpy 库中的 polyfit 函数进行拟合, 得出:

```

1 p = np.polyfit(数据 1, 数据 2, 3 次方程)

```

第 11-14 行将拟合方程的系数分别取为  $c_0, c_1, c_2, c_3$  以便使用。

第 16-17 行, 即为求根公式:

$$x = \frac{-2c_2 + \sqrt{(2c_2)^2 - 4 \times 3C_3 \times C_1}}{2 \times 3C_3} \quad (12)$$

其中 math.sqrt 为开方, \*\*2 为平方。

第 19 行将 a 反解, 得出晶格参数 (a)。

第 21 行将 a 的值打印到屏幕上。

### 3.5.4 数据输出

我们将以上包含 python 代码的文本文件命名为 bm.py 与 data 文件放在一个文件夹中, 在 Terminal 中输入:

```

1 python bm.py

```

在 python, math, numpy 等语言以及运行库完备的情况下, 屏幕就能得到输出:

```

1 a = 数字

```

有任何报错和差异, 请: 1、检查语言与运行库是否都已经安装在机器中。2、检查输入数据以及程序是否有错误, 特别是 7 行的晶格参数是否已修改为自己的晶格参数。

### 3.5.5 代码实现之使用 python 可视化拟合曲线

简单的代码实现:

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 #####以数组形式读取数据#####
5 x = np.array([1, 2, 3, 4, 5])
6 y = np.array([9.5, 10.3, 11.7, 12, 13.6])
7 #####以 csv 文件形式读取数据#####
8 data = pd.read_csv('1.txt', sep = '\t')
9 #1.txt 为要读取的文件名称, sep 为数据分隔方法
10 x = data.iloc[:,1]
11 #读取所有行的第一列的所有数据

```

```

12 y = data.iloc[:,2]
13 # 读取所有行的第二列的所有数据
14
15 ##### 拟合 #####
16 a , b ,c = polyfit(x , y , 2)
17 ##### 绘制 #####
18 plt.scatter(x,y) # 散点图
19 plt.plot( x , a*x**2 + b*x + c , color = 'red' )
20 # x 为横轴坐标 第二项为以y为自变量的曲线公式
21 # 第三项为曲线颜色
22 plt.show()# 显示

```

### 3.5.6 代码实现之 matlab

对于 3.5.2 中的流程我们同样可以用 matlab 实现, 见 2.1.4, matlab 是 windows 系统的计算软件, 因为安装较为容易便不多赘述, 主要对用 matlab 实现拟合进行说明。

打开 matlab, 点击左上角主页——> 新建脚本, 输入:

```

1      syms x
2      A = load('D:\桌面\答辩\TiAl\data');
3      a = power((A(:, 1)*4.03), -2);
4      P = polyfit(a,A(:,2),3);
5      digits(11);
6      P = vpa(P);
7      c0 = P(1);
8      c1 = P(2);
9      c2 = P(3);
10     c3 = P(4);
11     f = c0*x^3 + c1*x^2 +c2*x +c3;
12     df = diff(f,x);
13     f1 = matlabFunction(f);
14     eqn = df;
15     B = solve(eqn == 0);
16     para = 1./sqrt(B);

```

注意: 第 2 行路径修改为自己本机的 data 存放的位置, 第 3 行,4.03 的晶格参数同样为 1.0 时 POSCAR 的晶格参数。

点击编辑器——> 运行, 运行完毕后在命令行窗口输入: para 查看结果。

## 3.6 掺杂

掺杂是某种材料中掺入少量的其他的化合物或元素, 能够起到改善性能的作用。掺杂需要加入少量的其他元素或化合物, 这就需要我们在建立掺杂模型时, 第一步构建超胞, 所谓超胞就是扩大原胞, 所谓原胞就是材料的最小重复单元, 如 3.1.1 所示 POSCAR 即为一个原胞。

### 3.6.1 扩胞

要进行扩胞那就要使用我们之前提到的两个软件：2.1.3 Material Studio 与 2.1.2 VESTA。主要扩胞操作是在 MS 中完成的。

在晶体学网站，或手动输入的输入的 POSCAR 文件是文本文件，不能直接导入 MS 中，这时候就需要将文本文件转化为 MS 能够识别的 CIF 文件。

将文本文件转化为 CIF 文件需要 VESTA 软件的帮助：

1、打开 VESTA 软件，直接将备好的 POSCAR 文件拖入窗口。(注意导入的名称一定要为 POSCAR，要不然 VESTA 会不能识别。)

2、点击左上角 File —> Export Data，跳出窗口的第二项保存类型中选择 CIF file(\*.cif)，选择合适路径(注意要为英文路径)点击保存即可。

如此，我们就将文本文件转化为 CIF 文件了。

接下来就是要利用 MS 软件对 CIF 原胞进行扩胞：

1、同样是将已准备好的 CIF 文件新建的项目中。

2、点击上方操作栏中的 Build —> Symmtry —> Supercell，完成后跳出对话框输入 A, B, C 即 xyz 三轴分别扩胞多少倍。

3、假设我们需要 2\*2\*2 的超胞，那么输入 A: 2, B:2, C:2。点击 Create Supercell，便扩胞完成。

4、导出扩胞数据，点击左上角 File —> Export...，选择保存类型为 Crystallographic Information Files (\*.cif;\*.cmf)，选择合适路径，点击保存。

5、将扩胞后的 CIF 拖入 VESTA 将 CIF 转化为文本文件(见如上操作)，只需多选择一步 Fractional coordinates(分数坐标) 或 Cartesian coordinates(笛卡尔坐标)，并重命名为 POSCAR。

6、以下为我扩胞后的 POSCAR(方便演示只取两位有效数字)：

```
1      POSCAR\(2)
2      1.0
3      8.07      0.00      0.00
4      0.00      8.07      0.00
5      0.00      0.00      8.07
6      A1
7      32
8      Direct
9      0.00      0.00      0.00
10     0.00      0.25      0.25
11     0.25      0.00      0.25
12     0.25      0.25      0.00
13     0.50      0.00      0.00
14     0.50      0.25      0.25
15     0.75      0.00      0.25
16     0.75      0.25      0.00
17     0.00      0.50      0.00
18     0.00      0.75      0.25
19     0.25      0.50      0.25
20     0.25      0.75      0.00
```

```

21      0.50    0.50    0.00
22      0.50    0.75    0.25
23      0.75    0.50    0.25
24      0.75    0.75    0.00
25      0.00    0.00    0.50
26      0.00    0.25    0.75
27      0.25    0.00    0.75
28      0.25    0.25    0.50
29      0.50    0.00    0.50
30      0.50    0.25    0.75
31      0.75    0.00    0.75
32      0.75    0.25    0.50
33      0.00    0.50    0.50
34      0.00    0.75    0.75
35      0.25    0.50    0.75
36      0.25    0.75    0.50
37      0.50    0.50    0.50
38      0.50    0.75    0.75
39      0.75    0.50    0.75
40      0.75    0.75    0.50

```

扩胞前:

```

1      Al4
2      1.0
3      4.03    0.00    0.00
4      0.00    4.03    0.00
5      0.00    0.00    4.03
6      Al
7      4
8      direct
9      0.00    0.00    0.00
10     0.00    0.50    0.50
11     0.50    0.00    0.50
12     0.50    0.50    0.00

```

### 3.6.2 掺杂

上述已经叙述过，掺杂的含义。例如，如果此时我们想将纯 Al 中掺入少量 Ti 元素，那么直接在扩胞后的 POSCAR 的第 6-7 行修改为：

```

1      Al Ti
2      31 1

```

这代表，将一个 Al 置换为一个 Ti，掺杂浓度为  $1/32 = 0.031$ ，大约 3%，我们发现就算在  $2*2*2$  超胞中浓度也远远实际情况中的掺杂浓度。如果需要更低浓度的掺杂，模仿实际掺杂浓度就需要更大的超胞，这时候所需的算力就要更大。

### 3.6.3 掺杂后结构优化

我们进行掺杂后的初始体系并不是平衡体系，这时候就需要我们对此进行结构优化，在优化过程中，除需要将 POSCAR 替换为上述已掺杂的 POSCAR 外，要根据 2.2.2 中 INCAR 的描述将 ISIF=3，且要根据超胞大小适当改变离子步、电子步收敛精度和 KPONTS 取点个数，权衡计算时间与计算精度，在合适的时间中得到有效的合理平衡体系的信息。同时注意：因为我们将 POSCAR 中的原子种类由 Al 修改为 Al Ti 这要求我们同样要构建一个按同样 Al Ti 顺序的 POTCAR，才能进行结构优化。

### 3.6.4 空位缺陷

所谓空位缺陷是缺陷中的一种，上述我们涉及到了掺杂也就是置换缺陷，在掺杂中我们将体系中的某一原子换成另一种原子，而空位缺陷也即是在 POSCAR 中的删掉一个原子，以上述为例，POSCAR 第 6-7 行变为：

1	Al
2	31

注意：POSCAR 第 9-40 行描述原子位置的数据也要同样删掉一行，保持 31 行，与原子数匹配。

存在空位的超胞结构的结构优化与 3.6.3 相同，只需注意 POSCAR 与 POTCAR 原子种类保持一致。

## 3.7 切面

我们在解释 bulk 与 slab 区别的时候提到过，对于原胞来说，是表示周期性循环的结构，并无任何原子裸漏在外，而切面就是将 bulk 转变为 slab 的过程，使得一个方向的晶面的原子暴露在外，这样就能给研究表面、界面性质提供了初步的模型。

### 3.7.1 切表面

- 1、同样如同扩胞利用 MS 完成此项工作，我们将拟合好的原胞以 CIF 格式导入 MS。
- 2、点击上方操作栏中的 Build —> Surface —> Cleave Surface，跳出的对话框中，我们看到三个模块选项，分别为 Surface Box、Surface Mash 以及 Options。
- 3、我们主要关注 Surface Box，第一项为 Cleave plane(hkl) 即为我们切的晶面指数，至于晶面指数的含义具体见《材料科学基础》，我们可以浅显的理解为要将哪一方向的晶面暴露在外。
- 4、Surface Box 的第二项 Position 中有 Thickness，即我们要切面形成模型的厚度大小，方便演示取 7 层，即 fractional 修改为 7，点击 Cleave。
- 5、此时我们得到了一个为 7 层的切面模型，但这尚不是一个完整的 slab，这需要我们在其晶面朝向位置加入真空层，即模拟晶面暴露的情况。
- 6、点击上方操作栏中的 Build —> Crystals —> Bulid Vacuum Slab...
- 7、在弹出的对话框中 Vacuum Slab 的第二项 Vacuum Thickness 中填入 10，单位为埃 (Å)。(真空层保证在周期性循环中相隔的原子之间没有相互作用即可。)，点击 Build，slab 模型构建完毕。
- 8、将 slab 模型导出，转化为文本文件格式，重命名为 POSCAR，即完成了表面模型的构建。

### 3.7.2 表面模型优化

与超胞优化不同，我们只需使得 ISIF=2，进行仅原子位置变化的优化即可。

### 3.7.3 构建界面模型

在对表面模型构建的过程中，我们选择的均匀同质的块体进行表面构建，若存在两个不同的块体，我们要想研究不同块体间在某一方向上的性质，这时候的模型如何构建？

我们把这种模型称作界面模型。

1、我们分别将两个已经拟合优化好的 Bulk 导入到同一 MS 软件的项目中，各自进行到在 3.7.1 中 4、中的步骤。

2、接下来就是对于两个已经切好的各自七层两个表面进行拼接，点击上方操作栏中的 Build ——> Build Layers。

3、在跳出的对话框第一项的 Define Layers 中 Layer1 与 Layer2 中各自选择 1、中切好的表面进行搭建，点击 Build。

4、这样以 Layer2 为基底的界面模型就搭建形成，但是注意此时并不是 Slab 模型，需要我们进一步添加真空层。

5、点击上方操作栏中的 Build ——> Crystals ——> Bulid Vacuum Slab...

6、在弹出的对话框中 Vacuum Slab 的第二项 Vacuum Thickness 中填入 10，单位为埃 (Å)。(真空层保证在周期性循环中相隔的原子之间没有相互作用即可。)，点击 Build，slab 模型构建完毕。

7、提取出模型转化为文本文件，重命名为 POSCAR，即可进行使用。

### 3.7.4 界面模型优化

与 3.7.2 相同，只需注意保持 POSCAR 与 POTCAR 中原子种类以及顺序一致。

### 3.7.5 原子固定

在进行表、界面实际计算中，与全弛豫 (原子位置全放开) 相较，选择对于某些原子层进行固定能够很好的模拟实现材料情况的前提下降低些许计算量，所谓固定原子是使得原子在某一个位置不移动，在 vasp 计算中，主要是根据原子间作用力来优化计算得出符合收敛阈值之内的合理原子位置。

具体固定原子的操作是对 POSCAR 进行修改实现的，以面心立方 Al 原胞为例：

```

1      A14
2
3      1.0
4      4.03    0.00    0.00
5      0.00    4.03    0.00
6      0.00    0.00    4.03
7
8      Al
9      4
10     direct
11
12     0.00    0.00    0.00 T T T
13     0.00    0.50    0.50 F F F
14     0.50    0.00    0.50 T T T
15     0.50    0.50    0.00 T T T

```

第 10 行，对比之前的原胞，在描述 xyz 原胞位置参数后增加了 F F F，表示对此原胞的第 2 号原子的 xyz 位置在优化时都不进行移动，相反若是 T T T 就是对原子进行放开。

此外，vaspkitd 的 402 功能提供了原子固定的功能。

## 4 进阶实例计算

### 4.1 态密度 (DOS)

态密度 ((Density of States) 是材料的电子或其他粒子在某一能量 (ev) 区间内，可占据的量子态的数量分布。与材料的导电性、光学吸收、热学性质有关系。

#### 4.1.1 DOS 计算流程

1、对初始模型进行结构优化。

2、对优化后模型结果 CONTCAR 进行复制重命名为 POSCAR，修改参数进行自洽计算：

```
1      IBRION = -1
2      NSW = 0
```

3、非自洽计算，对电子信息进行读取，在 DOSCAR 写入 DOS 信息，修改以及添加参数，注意要准备更高密度的 K 点，即增大 KPOINTS 中的值：

```
1      ISTART = 1
2      ICHARG = 11
3      ISMEAR = -5
4      NEDOS = 2001 #DOS 取点数目
5      LORBIT = 10 # 规定 DOS 投影方式
```

4、对 DOSCAR 文件使用脚本进行数据提取，这里主要使用网上可以获取的 splitdos 脚本文件即可，用以下命令在要提取分割的 DOSCAR 目录下：

```
1      bash split_dos
```

而后可以获取以原子为文件单位的文本文件，即 POSCAR 第 6-7 行规定有多少个原子就可以分割成几个文本文件，且序号按 POSCAR 第 6 行先后顺序排列。

5、作图，可将包含 DOS 的文本文件使用 origin 作图，也可以使用其他作图方式完成作图。

6、在计算 DOS 时 EDIFF 最好使用更精准的收敛标准，比如 1E-6、1E-7。

#### 4.1.2 以 mapbi3 钙钛矿为例

1、结构优化参数设置

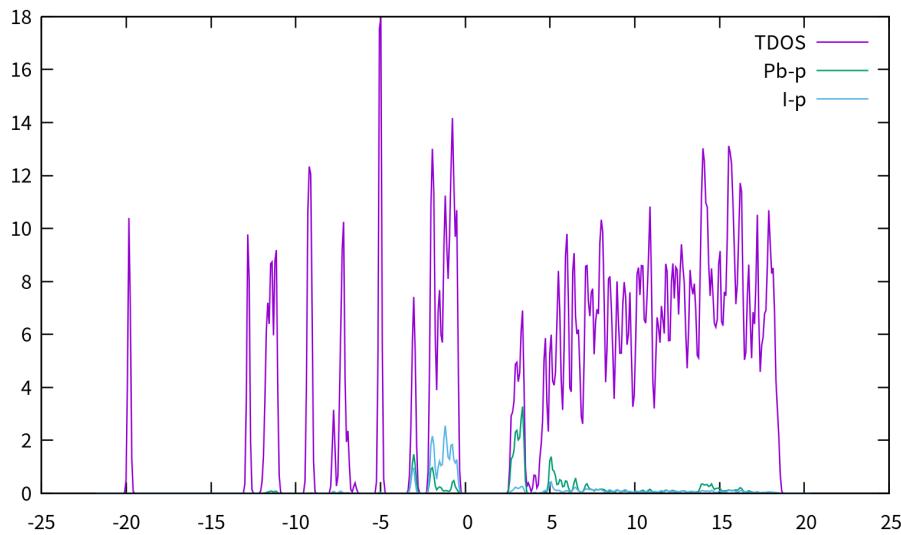
```
1      INCAR:
2          ISTART = 0
3          ICHARG = 2
4          ISPIN = 1
5          PREC = N
6
```

```

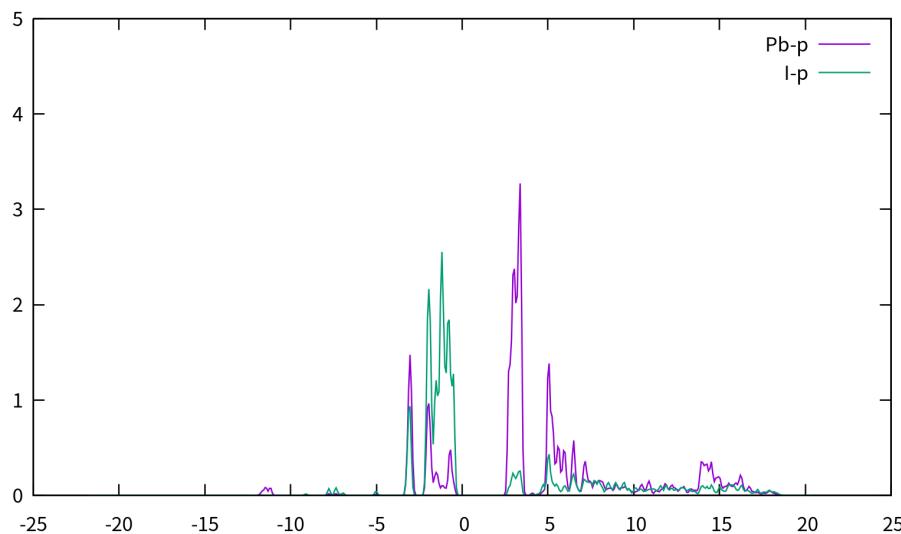
7      ALGO = N
8      NELM = 80
9      EDIFF = 1E-5
10     ENCUT = 400
11
12     IBRION = 2
13     NSW = 100
14     ISIF = 3 # 全放开优化
15     EDIFFG = -0.01
16
17     ISMEAR = 0
18     SIGMA = 0.1

```

2、其他过程循照 4.1.1 即可。3、DOS 数据经脚本处理后，有 DOS0-DOS12 共 13 个文本文件，DOS0 其中第一列为能量范围，第二列为总态密度 (TDOS) 数值，第三列为数值的积分值，各个分态密度 (PDOS) 其中第一列为能量范围，第二列到以后列分别为 s、p、d、f 轨道态密度，是否有此轨道数值依照原子具体情况信息而定，作图如下：



以下图 p 轨道为例子：



可以分析得到，大于 0 态密度不为零处称为价带，小于 0 态密度不为 0 处为导带，价带底与导带顶的信息是需要着重注意的地方。如图中，导带顶与价带底所夹态密度为零处称为禁带，一般存在于绝缘体或半导体材料中，其数值大小具有重要意义。于图中可知，I 元素的 p 轨道对导带贡献较大，Pb 元素的 p 轨道对价带贡献较大。注意：在使用 PBE 泛函时通常会低估禁带宽度。

## 4.2 电荷密度图

### 4.2.1 电荷密度计算流程

- 1、对初始结构进行优化。
- 2、将 CONTCAR 复制命名为 POSCAR，修改 INCAR

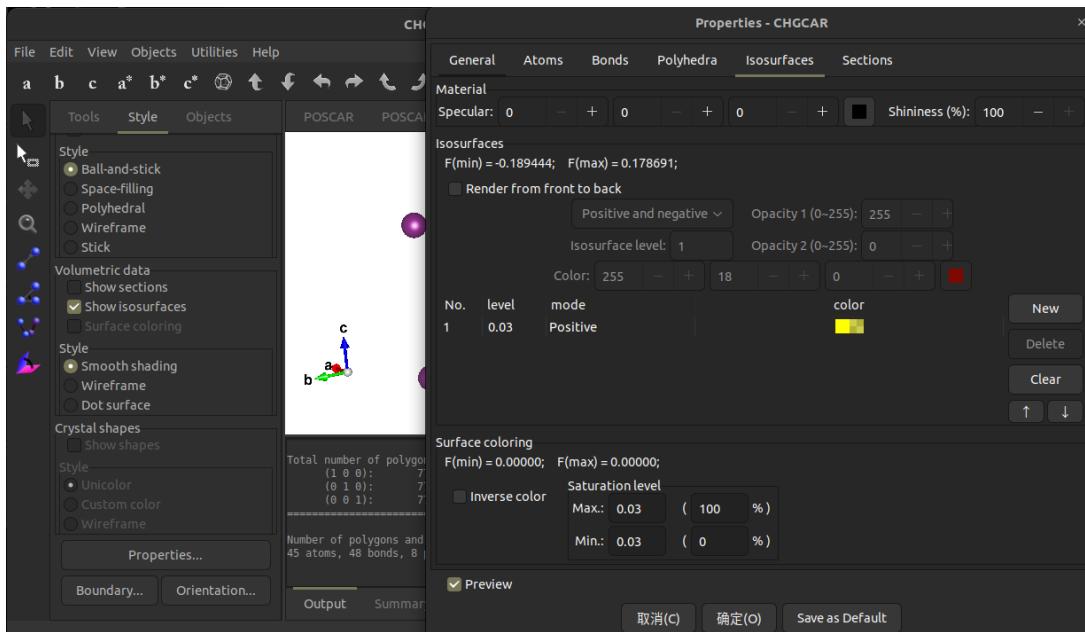
```

1      ISTART = 1
2      ICHARG = 1
3      IBRION = -1
4      NSW = 0

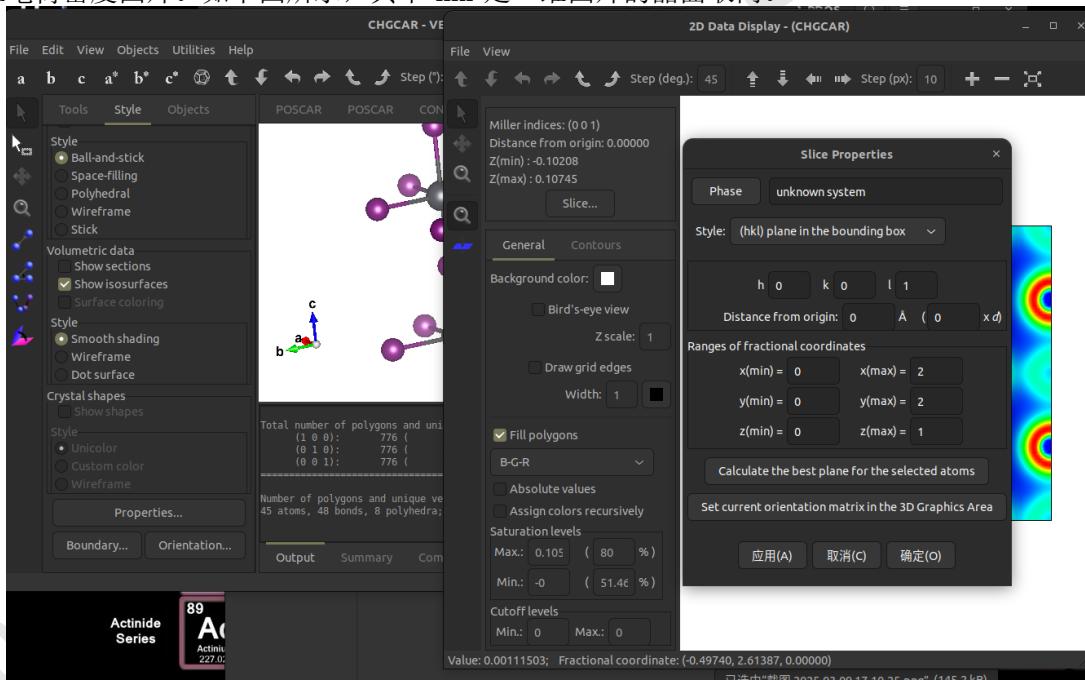
```

进行自洽计算。

- 3、将 CHARG 拖入 VESTA 中，在 Properties 中的 Isosurfaces 修改设置，提取三维电荷密度图片。如下图所示。



4、Utilities-2D Data Display-slice 进行设置；在窗口的 Saturation levels 进行范围调节，提取二维电荷密度图片。如下图所示，其中 hkl 是二维图片的晶面取向。



#### 4.2.2 以 CsPbI<sub>3</sub> 钙钛矿为例

##### 1、结构优化参数设置

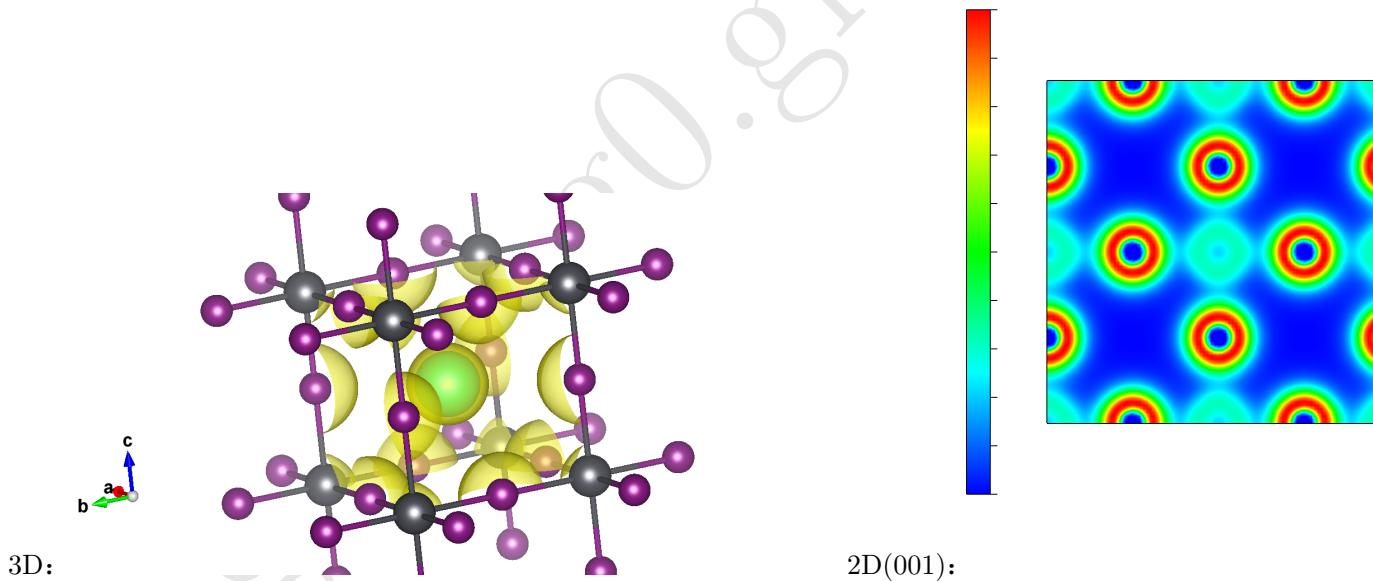
```

1 ISTART = 0
2 ICHARG = 2
3 ISPIN = 2 # 考虑自旋
4 PREC = N
5
6 ALGO = N

```

```
7      NELM = 80
8      EDIFF = 1E-5
9      ENCUT = 230
10
11     IBRION = 2
12     NSW = 100
13     ISIF = 3 # 全放开
14     EDIFF = -0.01
15
16     ISMEAR = 0
17     SIGMA = 0.1
```

2、进行 4.2.1 的操作流程，得到如下图片结果



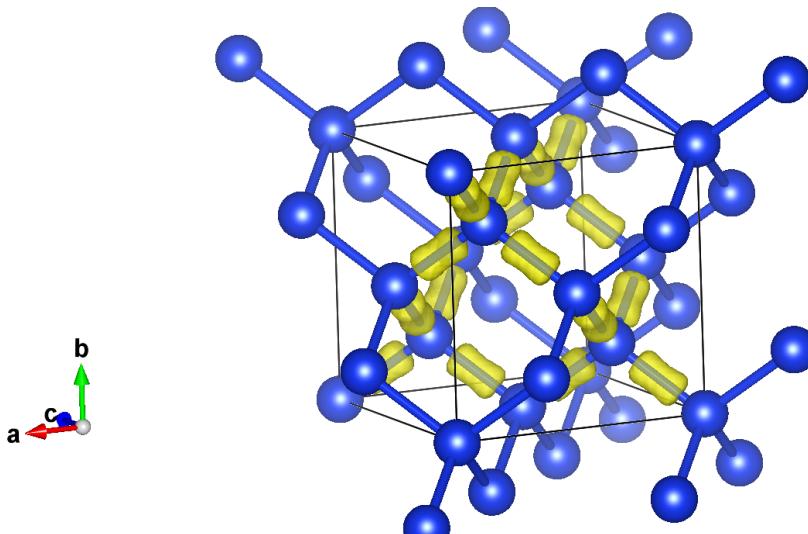
3D:

2D(001):

### 3、电荷密度的分析：

共价键：键间存在连续的电荷密度。离子键：电荷密度在原子间显著分离。金属键：电子云分布均匀，无明显的方向性。

如图分析得知键间不存在明显的电荷分布，而在原子间显著分离，故 Pb 与 I 是典型的离子键。例如，Si-Si 为典型的共价键，如下图所示：



### 4.3 差分电荷密度图 (电荷密度差)

#### 4.3.1 计算流程

- 1、将 A、B 两部分拼接好的模型进行优化。
- 2、将拼接好的模型进行自治计算输出电荷文件。
- 3、修改以上模型的 POSCAR 分别为 A 与 B 的两部分各自独立的 POSCAR，并分别自治计算输出这两部分的电荷文件。
- 4、使用 VESTA 进行差分电荷密度 (电荷密度差) 的可视化。

### 4.4 能带计算

能带计算与 DOS 计算的 INCAR 设置极其类似，主要是都要使用 ICHARGE = 11，且在计算之前都要进行结构优化，而后自治计算的流程。区别在于能带计算的时候需要高对称 K 点，DOS 计算则为平常的即可。ISMEAR 在 DOS 计算的时候设置为-5，能带则为 0 或-1。

#### 4.4.1 计算流程

- 1、结构优化以及自治计算。
- 2、用 vasppkit 3 功能生成高对称点，并将生成的 KPATH.in 改名为 KPOINTS。
- 3、准备能带计算的 INCAR，关键参数：

```

1      ICHARGE = 11
2      ISMEAR = 0 或 -1
3      NEDOS = 2001
4      LORBIT = 11 # 规定能带投影方式

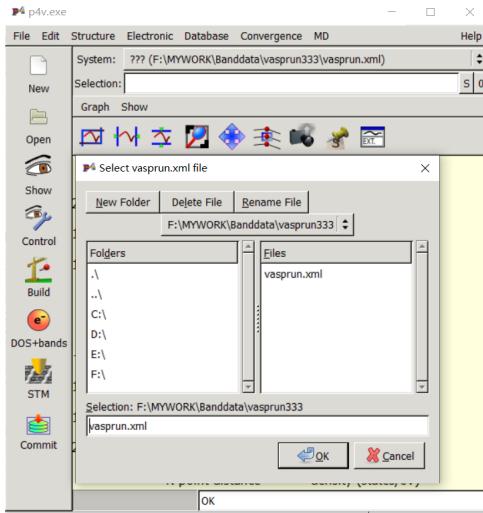
```

- 4、计算，完毕后能带信息保存到 EIGENVAL 中，将自治计算中的 DOSCAR 复制到计算能带的文件夹中，使用 vasppkit 211 功能导出能带信息到 BAND.dat 中。
- 5、将 BAND.dat 的数据拖入 Origin 即可绘制能带。

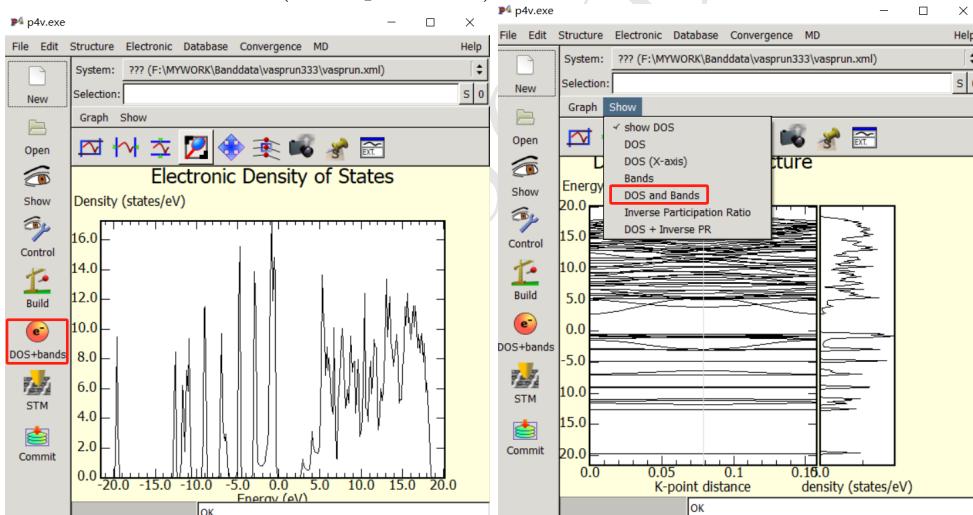
#### 4.4.2 用 p4vasp 查看能带与 DOS

1、默认体系完成优化-自洽计算-获取 DOS 信息/能带信息。

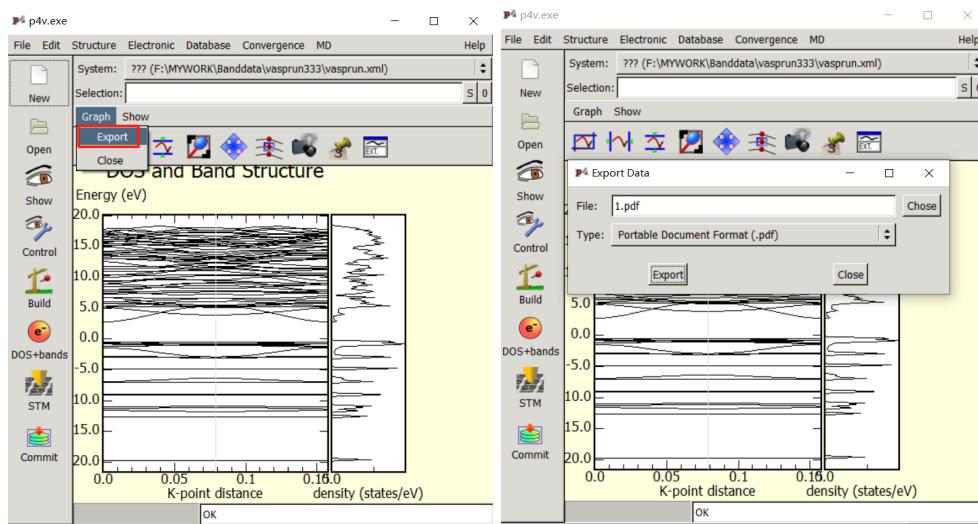
2、打开 p4vasp，点击 file-load system-找到 vasp 已经计算好的 vasprun.xml。



3、导入完成后点击 DOS+Bands，如图所示 (以 mapbi3 为例). 点击 show-DOS and Bands，获得能带与 DOS，如图所示 (以 mapbi3 为例)。



4、数据图片的导出 Graph-export，有多种格式记得添加后缀。



#### 4.4.3 利用 pymatgen 绘制能带与 DOS

事实上，pymatgen 是可以支持 vasp 计算强大的前后处理工具，不仅是能带及 DOS 图的绘制，pymatgen 几乎可以涉及到很多常用的前后处理的场景使用。由于 pymatgen 是基于 python 实现，需要有一定的 python 基础。

在上述章节中，已经叙述了能带以及 DOS 的计算流程，在完成能带以及 DOS 计算后会产生 Vasprun.xml 文件，其中包含了能带以及 DOS 的信息，pymatgen 中绘制能带以及 DOS 的模块就是对此文件的读取。

教程：<https://matgenb.materialsvirtuallab.org/>  
<https://snyk.io/advisor/python/pymatgen/example>

## 4.5 声子谱计算

利用声子谱可以分析晶格动力学性质以及热力学行为。通常采用两种方法计算声子谱：1、有限位移法，需要结合 phonopy 进行后处理。2、密度泛函微扰理论 (DFPT)，需要 VASP5.4.1 及以上版本。

### 4.5.1 有限位移法

- 1、对原胞进行高精度的优化。
- 2、对优化后的原胞进行扩胞，生成有限位移结构，以 2\*2\*2 为例：

```
1 phonopy -d --dim="2 2 2" -c POSCAR --amplitude=0.01
```

其中，dim="2 2 2" 为扩胞 abc 方向的倍数，POSCAR 为初始晶胞文件，amplitude 为原子位移，0.01 为默认值。执行完此命令后会生成 POSCAR-001 等

注：若为二维材料 y 轴扩胞为 1

```
1 phonopy -d --dim="2 2 1" -c POSCAR0 --amplitude=0.01
```

- 3、对 INCAR 进行修改

```

1   IBRION = -1
2   NSW = 0
3   #高精度
4   PREC = A
5   EDIFF = 1E-8

```

4、对生成的 POSCAR-001 等放进各自文件夹准备好 INCAR、POSCAR、KPOINTS 进行计算：

```

1   for i in POSCAR-*;
2     do
3       mkdir calc_$i && cp POSCAR-$i calc_$i/POSCAR
4       cp INCAR POTCAR KPOINTS calc_$i/
5       cd calc_$i && mpirun -np 72 vasp_std && cd ..
6     done

```

5、对已经计算完成的各个 vasprun.xml 进行收集：

```
1 phonopy -f 01/vasprun.xml 02/vasprun.xml ...
```

运行完毕后生成 FORCE\_SET 文件。

6、准备生成声子谱的输入文件，命名为 band.conf(随意命名)

```

1 ATOM_NAME = XX
2 DIM = 2 2 2
3 FORCE_SET = READ
4 EIGENVECTORS = .TRUE.
5 BAND_CONNECTION = .TRUE.
6 FORCE_CONSTANTS = WRITE
7 BAND_LABELS = X M G R
8 #高对称点选择需要根据初始晶胞选择合适的高对称点
9 BAND = 0.0 0.5 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.5

```

7、运行命令，输入 band.conf

```
1 phonopy -c POSCAR0 -p -s band.conf
```

生成 band.pdf 即声子谱，以及 FORCE\_CONSTANTS 力常数文件。

#### 4.5.2 有限位移法以单层 MoS<sub>2</sub> 为例

- 1、晶体学网站下载 POSCAR 文件。
- 2、对 POSCAR 进行优化获得优化后的原胞文件，命名为 POSCAR0。
- 3、进行扩胞，由于是二维材料所以不进行 z 方向的扩胞，输入：

```
1 phonopy -d --dim="2 2 1" -c POSCAR0 --amplitude=0.01
```

生成 SPOSCAR 以及 POSCAR-001、POSCAR-002、POSCAR-003 文件。

4、由于采用的是有限位移法，对各个 POSCAR-\* 新建文件夹分别命名为 01、02、03 进行自洽计算即可。其中，INCAR 需要进行修改的重要参数：

```
1      IBRION = -1  
2      NSW = 0  
3      ISIF = 2
```

5、对三个文件的 vasprun.xml 进行收集:

```
1 phonopy -f 01/vasprun.xml 02/vasprun.xml 03/vasprun.xml
```

生成 FORCE\_SET 文件。

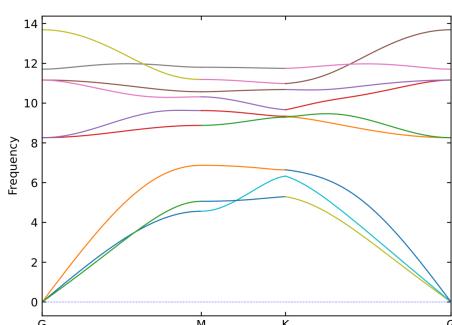
```
phonopy@phonopy:/home/phonopy$ phonopy -f 01/vasprun.xml 02/vasprun.xml 03/vasprun.xml  
.....  
.....[time 2025-07-01 15:57:16].....  
Compiled with OpenMP support (max 96 threads).  
Python version 3.9.12  
Spglib version 2.6.0  
  
Displacements were read from "phonopy_disp.yaml".  
Number of supercell files: 3  
Number of displacements in "phonopy_disp.yaml": 3  
counter (file index): 1 2 3  
"FORCE_SETS" has been created.  
.....[time 2025-07-01 15:57:16].....  
.....  
.....
```

6、生成声子谱以及力常数文件

```
1 phonopy -c POSCAR0 -p -s band.conf
```

```
Compiled with OpenMP support (max 96 threads).  
Python version 3.9.12  
Spglib version 2.6.0  
  
"band.conf" was read as phonopy configuration file.  
Crystal structure was read from "POSCAR0".  
Unit of length: angstrom  
Band structure mode  
Settings:  
    Supercell: [ 2 2 1 ]  
    Spacegroup: P-3m1 (167)  
Number of symmetry operations in supercell: 48  
Use -v option to watch primitive cell, unit cell, and supercell structures.  
  
Forces and displacements were read from "FORCE_SETS".  
Force constants solver of phonopy-traditional is used.  
Max drift of force constants: -0.079592 (yy) 0.000001 (yy)  
Force constants are written into "FORCE_CONSTANTS".  
Array shape of force constants is (3, 12, 3, 3).  
Use --full-fc option for full array of force constants.  
  
*****  
The "phonopy" command for running phonon calculations will be phased out in  
the future. It is recommended to use the "phonopy-load" command instead.  
For more details, please refer to this link.  
https://phonopy.github.io/phonopy/phonopy-load.html  
*****  
  
Reciprocal space paths in reduced coordinates:  
[ 0.000 0.000 0.000 ] -> [ 0.500 0.000 0.000 ]  
[ 0.500 0.000 0.000 ] -> [ 0.300 0.300 0.000 ]  
[ 0.300 0.300 0.000 ] -> [ 0.000 0.000 0.000 ]  
  
Summary of calculation was written in "phonopy.yaml".  
.....[time 2025-07-01 16:01:21].....  
.....  
.....
```

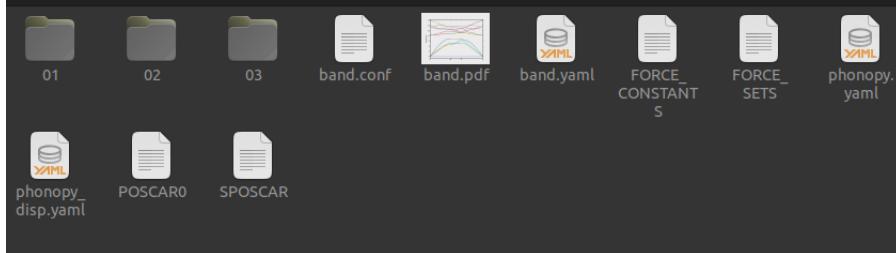
其中 band.pdf 即声子谱, 以及 FORCE\_CONSTANTS 力常数文件。MoS2 band.pdf:



注意：需要计算计算得出的声子谱不存在虚频，即在纵坐标 0 以下不存在数据。如若有虚频可尝试：

- (1) 将优化的原胞进行更高精度的优化增加收敛精度，增加优化的 K 点密度。
- (2) 对扩胞倍数进行减小或增大。
- (3) 对第 4 步进行自治计算的收敛精度进行增加。

最后，文件夹内文件：



#### 4.5.3 密度泛函微扰法

1、对原胞进行高精度的优化 2、对优化后的原胞进行扩胞，借助 phonopy，以 2\*2\*2 为例：

```
1 phonopy -d --dim="2 2 2" -c POSCAR --amplitude=0.01
```

各个参数代表意义在 4.5.1(2) 中有解释。执行完此命令生成 SPOSCAR 文件、POSCAR-\*、phonopy\_disp.yaml 文件。3、将 SPOSCAR 文件作为进行密度泛函微扰的 POSCAR 文件进行 IBRION = 8 的计算。

其中，INCAR 需要进行修改的重要参数：

```
1 EDIFF = 1E-8
2 IBRION = 8
3 NSW = 1
4 ISIF = 3
```

运行命令进行计算。

4、将计算中生成的文件中的 vasprun.xml 进行收集。

```
1 phonopy --fc 01/vasprun.xml
```

FORCE\_CONSTANTS 文件，即二阶力常数被创建。

5、准备 band.conf 文件

```
1 ATOM_NAME = ****
2 DIM = 2 2 2
3 BAND = 0.0 0.5 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.5 0.5 0.5
4 BAND_LABELS = X M G R
5 FORCE_CONSTANTS = READ # 注意此处为 READ 读取
6 FC_SYMMETRY = .TRUE.
```

6、执行命令，生成 band.yaml、phonopy.yaml、band.pdf 文件

```
1 phonopy -c POSCAR0 -p -s band.conf
```

## 4.6 晶体轨道重叠布局 (COOP)、晶体轨道哈密顿布局 (COHP)

### 4.6.1 LOBSTER 程序之 lobsterin

lobsterin 是控制 LOBSTER 计算内容的输入文件，以文本文件形式，命名为 lobsterin 放置于计算文件下即可，下面是 lobsterin 控制参数，在使用 LOBSTER 时候选择性的输入到 lobsterin 中即可。

1、 basiset 局域化轨道基组: pbevaspfit2015、 koga、 bunge

```
1 basiset pbevaspfit2015
2 basiset koga
3 basiset bunge
```

注: vasp 中常用 pbevaspfit2015。

2、 includeorbitals 原子轨道基组: spd

```
1 includeorbitals spd
2 includeorbitals spdf
```

注: 如果不进行指定则按 POTCAR 中存在的轨道进行计算。

3、 basisfunctions 指定每种元素所用的原子轨道基组

```
1 basisfunctions Al 3s 3p
2 basisfunctions Sc 3s 3p 3d 4s
```

注: 如果不进行指定则按 POTCAR 中存在的轨道进行计算

4、 cohpbetween 指定原子对

```
1 cohpbetween atom 1 and atom 2# 总体相互作用
2 cohpbetween atom 1 and atom 2 orbitalwise# 原子轨道间相互作用
```

注: 原子与原子间相互作用的重要描述参数。

5、 cohpgenerator 指定距离或元素

```
1 cohpgenerator form 1.5 to 2.5 type Al type Sc
2 cohpgenerator type Al type Sc
```

6、 cohptestartenergy 最小、最大能量 (ev)

```
1 cohptestartenergy -10
2 cohptestartenergy 10
```

7、 saveprojectiontofile 保存原子轨道基组信息到文件，从文件读取原子轨道基组信息，可以加速计算。

8、 skipMadelungEnergy 由于在计算 Madelung 能量时会使用大量时间，所以对于计算无要求的情况下设置此参数用于跳过 Madelung 能量的计算。

### 4.6.2 计算流程

1、结构优化

2、静态计算需要注意的参数:

```

1 ISYM = 0 或 -1
2 ISMEAR 设置为 1 或 0 时 lobsterin 要增加 gaussianSmearingWidth 参数。
3 NBANDS 尽量大
4 LORBIT 推荐用 12

```

### 3、COHP 运行

设置好 4.5.1 的参数存入 txt 文件中，重命名为 lobsterin 运行读取静态计算中的能量文件，运行以下命令：

```
1 lobster-4.1.0
```

注意：以上命令是本机命令，自己机器上关于 lobster 运行的命令需要进一步确定。

4、运行后的结果检查所有运行屏幕显示的信息会存入 lobsterin 文件中，其中：

```

1 saving projection to projectionData.lobster...
2 abs. total spilling: 14.64%
3 abs. charge spilling: 1.54%

```

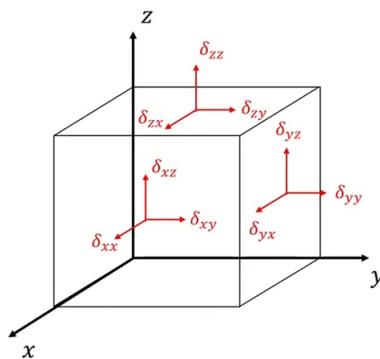
charge spilling (电子溢出)：必须要小于 5%，才可保证计算精准度。

## 4.7 力学性质

关于弹性常数的计算常用两种方法：应力-应变法和能量-应变法。

### 4.7.1 弹性常数定义

应力-应变法源自胡克定律  $F = -kx$ ，F 为应力，x 为应变，k 为弹性系数，广义上来说所谓的弹性常数就是应力与应变的比值。



在固体力学中，对于一个材料而言，应力与应变是二阶张量，其中的弹性常数是应力对应变的偏导数，得到一个四阶张量。这里涉及到的张量的概念可以浅显的认为是符号角标的个数，例如  $\sigma_{ij}$  就是一个二阶张量，根据应力应变关系： $\sigma_{ij} = C_{ijkl}\varepsilon_{kl}$  其中的  $ijkl$  分别遍历 1 2 3，将上述式子展

开即可得到应力应变的最原始的样子：

$$\left\{ \begin{array}{l} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{21} \\ \sigma_{32} \\ \sigma_{31} \end{array} \right\} = \left[ \begin{array}{l} C_{1111} C_{1122} C_{1133} C_{1112} C_{1123} C_{1113} C_{1121} C_{1132} C_{1131} \\ C_{2211} C_{2222} C_{2233} C_{2212} C_{2223} C_{2213} C_{2221} C_{2232} C_{2231} \\ C_{3311} C_{3322} C_{3333} C_{3312} C_{3323} C_{3313} C_{3321} C_{3332} C_{3331} \\ C_{1211} C_{1222} C_{1233}; C_{1212} C_{1223} C_{1213} C_{1221} C_{1232} C_{1231} \\ C_{2311} C_{2322} C_{2333} C_{2312} C_{2323} C_{2313} C_{2321} C_{2332} C_{2331} \\ C_{1311} C_{1322} C_{1333} C_{1312} C_{1323} C_{1313} C_{1321} C_{1332} C_{1331} \\ C_{2111}; C_{2122}; C_{2133} C_{2112} C_{2123} C_{2113} C_{2121} C_{2132} C_{2131} \\ C_{3211} C_{3222} C_{3233} C_{3212} C_{3223} C_{3213} C_{3221} C_{3232} C_{3231} \\ C_{3111} C_{3122} C_{3133} C_{3112} C_{3123} C_{3113} C_{3121} C_{3132} C_{3131} \end{array} \right] \left\{ \begin{array}{l} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \varepsilon_{12} \\ \varepsilon_{23} \\ \varepsilon_{13} \\ \varepsilon_{21} \\ \varepsilon_{32} \\ \varepsilon_{31} \end{array} \right\} \quad (13)$$

如果每一次表征应力应变关系都是这种公式，那将是很麻烦的事情，根据对称性关系，将上式进行操作：

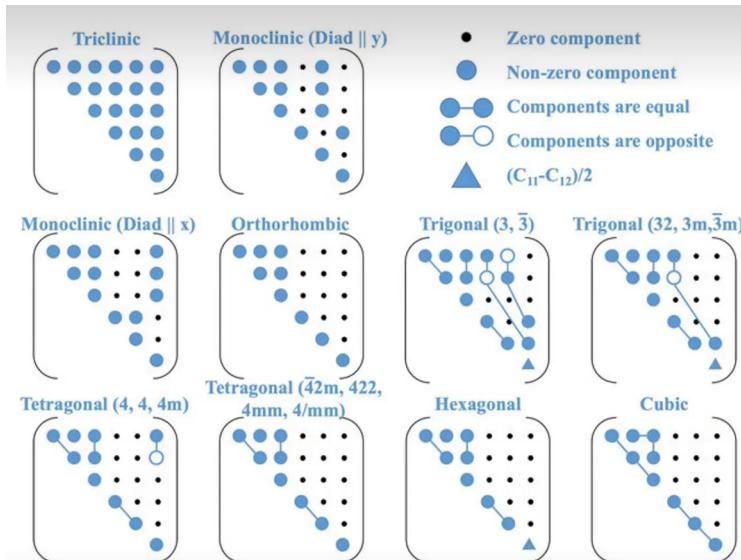
$$\left\{ \begin{array}{l} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{13} \end{array} \right\} = \left[ \begin{array}{l} C_{1111} C_{1122} C_{1133} C_{1112} C_{1123} C_{1113} \\ C_{2211} C_{2222} C_{2233} C_{2212} C_{2223} C_{2213} \\ C_{3311} C_{3322} C_{3333} C_{3312} C_{3323} C_{3313} \\ C_{1211} C_{1222} C_{1233} C_{1212} C_{1223} C_{1213} \\ C_{2311} C_{2322} C_{2333} C_{2312} C_{2323} C_{2313} \\ C_{1311} C_{1322} C_{1333} C_{1312} C_{1323} C_{1313} \end{array} \right] \left\{ \begin{array}{l} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \end{array} \right\} \quad (14)$$

这种记法称为 Voigt 标记，再对上式子根据爱因斯坦求和约定进一步进行简写：

$$\left\{ \begin{array}{l} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{array} \right\} = \left[ \begin{array}{l} C_{11} C_{12} C_{13} C_{14} C_{15} C_{16} \\ C_{21} C_{22} C_{23} C_{24} C_{25} C_{26} \\ C_{31} C_{32} C_{33} C_{34} C_{35} C_{36} \\ C_{41} C_{42} C_{43} C_{44} C_{45} C_{46} \\ C_{51} C_{52} C_{53} C_{54} C_{55} C_{56} \\ C_{61} C_{62} C_{63} C_{64} C_{65} C_{66} \end{array} \right] \left\{ \begin{array}{l} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{array} \right\} \quad (15)$$

根据晶体实际情况的空间群对称性将变量缩减为最多 21 个 (如图)，三斜晶系对称性最差需要考虑 21 个变量，立方晶系对称性最好仅需要考虑 3 个变量。在要进行的弹性常数的计算中立方晶系是最常见的。

上述提到应力对应变的偏导数为弹性常数，在具体计算中只需求出应力对应变的变化率即可。



另一种方法为能量-应变法，对平衡晶体施加微小应变得到如下关系式：

$$\Delta E(V, \{\varepsilon_i\}) = E(V, \{\varepsilon_i\}) - E(V_0, 0) = \frac{V_0}{2} \sum_{i,j=1}^6 C_{ij} \varepsilon_j \varepsilon_i,$$

$E(V, \{\varepsilon_i\})$  和  $E(V_0, 0)$  分别是体系微小应变下与不施加应变状态下的总能。vaspkit 手册中提到，由于应力应变法在相同计算结果上需要更大的计算精准度，一般推荐使用能量应变法，对于立方晶系将上式展开：

$$\Delta E(V, \{\varepsilon_i\}) = \frac{V_0}{2} \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \varepsilon_3 & \varepsilon_4 & \varepsilon_5 & \varepsilon_6 \end{bmatrix} \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{bmatrix}$$

进而 (1) 施加三轴切应力  $\varepsilon = (0, 0, 0, \delta, \delta, \delta)$ ，得到  $\Delta E = \frac{V_0}{2} (C_{44}\varepsilon_4\varepsilon_4 + C_{44}\varepsilon_5\varepsilon_5 + C_{44}\varepsilon_6\varepsilon_6)$  由于  $\varepsilon_4 = \varepsilon_5 = \varepsilon_6 = \delta$  所以  $\frac{\Delta E}{V_0} = \frac{3}{2}C_{44}\delta^2$ ，得出  $C_{44}$  的结果。

(2) 施加应力  $\varepsilon = (\delta, \delta, 0, 0, 0, 0)$  得到  $\Delta E = \frac{V_0}{2} (C_{11}\varepsilon_1\varepsilon_1 + C_{11}\varepsilon_2\varepsilon_2 + C_{12}\varepsilon_1\varepsilon_2 + C_{12}\varepsilon_2\varepsilon_1)$  得到  $\frac{\Delta E}{V_0} = (C_{11} + C_{12})\delta^2$ 。

(3) 施加应力  $\varepsilon = (\delta, \delta, \delta, 0, 0, 0)$  得到  $\frac{\Delta E}{V_0} = \frac{3}{2}(C_{11} + 2C_{12})\delta^2$ 。

(4) 将 (2),(3) 联立得到  $C_{11}$  与  $C_{12}$ 。

晶格矩阵与应变的关系式子为：

$$\begin{pmatrix} a' \\ b' \\ c' \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \cdot (\mathbf{I} + \boldsymbol{\varepsilon})$$

其中  $\mathbf{I}$  为单位矩阵，

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 & \frac{\varepsilon_6}{2} & \frac{\varepsilon_5}{2} \\ \frac{\varepsilon_6}{2} & \varepsilon_2 & \frac{\varepsilon_4}{2} \\ \frac{\varepsilon_5}{2} & \frac{\varepsilon_4}{2} & \varepsilon_3 \end{pmatrix}$$

例如，晶格参数为：

$$A = \begin{pmatrix} 4.0402 & 0 & 0 \\ 0 & 4.0402 & 0 \\ 0 & 0 & 4.0402 \end{pmatrix}$$

$$\boldsymbol{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

施加 1% 应力  $\boldsymbol{\varepsilon} = (0.01, 0.01, 0.01, 0, 0, 0,)$

即

$$\boldsymbol{\varepsilon} = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{pmatrix}$$

$$\boldsymbol{A}' = \boldsymbol{A}(\boldsymbol{I} + \boldsymbol{\varepsilon})$$

$\boldsymbol{A}'$  即为应变后晶格参数。

注：以上施加应变的方式会引起晶格体积的变化，这在一定程度上会影响计算结果的准确性，为避免这种影响。

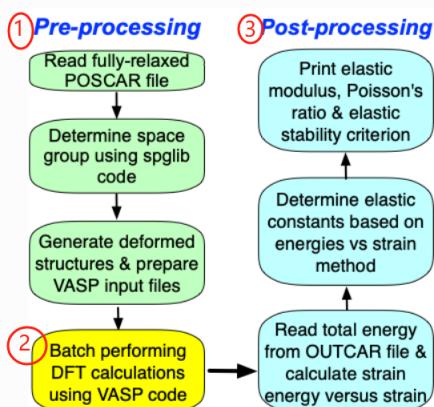
(1) 施加应力  $\boldsymbol{\varepsilon} = (0, 0, \delta^2/(1 - \delta), 0, 0, \delta)$ ，可在体积不变的情况下，得到  $\frac{\Delta E}{V_0} = 2C_{66}\delta^2$ 。

(2) 施加应力  $\boldsymbol{\varepsilon} = (\delta, \delta, \delta^2/(1 - \delta), 0, 0, 0)$ ，得到  $\frac{\Delta E}{V_0} = (C_{11} - C_{22})\delta^2$ 。

这即为脚本中 1、2 文件夹施加应力的来源。

#### 4.7.2 弹性常数：利用 vaspy 计算

vaspy 中 200 与 201 功能分别是应力应变法与能量应变法计算力学性能。



(1) 对于晶胞进行结构优化。(vaspy 提到要使用 vaspy603 功能生成对称晶胞，认为对于一般立方晶系可以不进行这一步。) (2) 准备 INCAR 文件，注意 ISIF=2，vaspy 给的 INCAR 参考如下：

```

1      Global Parameters
2          ISTART = 0
3          LREAL  = F
4          PREC   = High
5          LWAVE  = F

```

```

6      LCHARG = F
7      ADDGRID= .TRUE.
8      Electronic Relaxation
9      ISMEAR = 0
10     SIGMA = 0.05
11     NELM = 40
12     NELMIN = 4
13     EDIFF = 1E-08
14     Ionic Relaxation
15     NELMIN = 6
16     NSW = 100
17     IBRION = 2
18     ISIF = 2      (Must be 2)
19     EDIFFG = -1E-02

```

(2) 在文件夹下新建一个名为 INPUT.in 文件，用来创建各自应变下的文件。参考，3、4 行应变取点个数可以适量增加：

```

1      ! 1 for pre-processing; 2 for post-processing
2 3D      ! 2D for two-dimentional, 3D for bulk
3 7      ! number of strain case
4 -0.015 -0.010 -0.005 0.000 0.005 0.010 0.015 ! Strain range

```

第一行：1 为前处理，2 为后处理，第二行 3D 为三维材料，2D 为二维材料，第三行拉伸取点个数，注意与第四行数量一致，第四行为应变比例，注意正负对称一般 -0.02 到 0.02 之间即可。

准备好文件 INCAR、POTCAR、KPOINTS、POSCAR、INPUT.in 后，终端输入：

```
1 vaspkit -task 201 或 200
```

进行前处理。

(3) 对各个文件进行 ISIF=2 的计算参考以下：

```

1      #!/bin/bash
2      for i in C*
3          do
4              cd $i
5                  for j in *
6                      do
7                          cd $j
8                              mpirun -np 72 vasp_std
9                          cd ..
10                         done
11                     cd ..
12                 done

```

注意：在输入代码时，最好用 vim，即 vim 文件名称，按 i 切换输入字符模式，在非输入字符模式下，按:wq 退出保存。

(4) 对数据进行后处理将 INPUT.in 文件中第一行改为 2，然后输入：

```
1      vasakit -task 201或200
```

结果输出在屏幕上，弹性常数保存在 ELASTIC\_TENSOR 文件中。

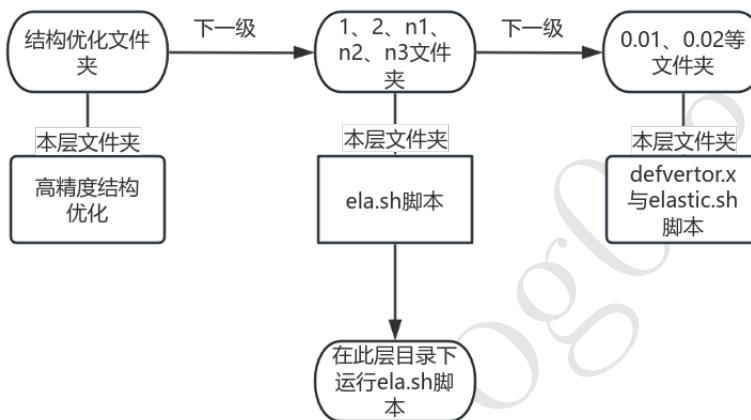
或输入：

```
1      echo -e "201或200" |vaskit >> result
```

结果保存在 result 中，弹性常数保存在 ELASTIC\_TENSOR 文件中。

结束！

#### 4.7.3 弹性常数：脚本计算



**defvector.x:** 将无形变的晶胞进行微小形变，形变方法如 4.7.1 所示。实现了未加形变晶胞的数据读取，施加形变后的晶胞的输出。

**elastic.sh:** 对同一变形下不同形变量的文件夹进行创建，并进行一次 ISIF=2 的结构优化与一次静态计算。并将每个微小形变下的能量、体积、计算时间写入 SUMMARY 文件中。

**ela.sh:** 遍历分别属于不同形变下的文件夹 1、2、nc1、nc2、nc3，并对已经过高精度优化后晶胞文件进行重命名为 OLDPOS，并将其第六行数据删除，第一次替换为第六行的元素符号以及元素种类数量（数量必须相符）。进行 elastic.sh 的调用完成不同型变量的计算。

**P-R.sh:** 进行两次优化，一次静态计算，并将过程与结果记录。

**注意：** **ela.sh** 脚本中 `sed -i -e '需要修改为自己体系的元素名称与种类数量'`，每个不同形变文件要准备好 **defvector.f** 并将其中的 **strain** 改为相应的值。

计算流程：1、在高精度优化后的文件夹中创建 elastic 文件夹，将 ela.sh 脚本复制进此文件夹，并修改 ela.sh 脚本中：

```
1      sed -i -e '需要修改为自己体系的元素名称与种类数量'
2      例如 AlTi
3      sed -i -e 'AlTi 2' -e '6d' OLDPOS
```

在文件夹创建 1 2 nc1 nc2 nc3 五个文件夹并分别将五个文件夹中复制进去 defvector.f 文件、elastic.sh 脚本，修改 defvector.f 其中 strain 参数。

文件夹	nc1	nc2	nc3	1	2
strain(1)	detal	detal	0.0	detal	0.0
strain(2)	detal	detal	0.0	-detal	0.0
strain(3)	detal	0.0	0.0	detal**2/(1-delta**2)	detal**2/(4-delta**2)
strain(4)	0.0	0.0	detal	0.0	0.0
strain(5)	0.0	0.0	detal	0.0	0.0
strain(6)	0.0	0.0	detal	0.0	detal

其中 1、2 为体积不变的能量-应变拉伸，nc1、nc2、nc3 都会引起体积变化。

2、运行 ela.sh

```
1 bash ela.sh
```

#### 4.7.4 脚本文件注释

ela.sh

```
1 #!/bin/sh
2
3 #遍历不同应变文件夹.修改OLDPOS的第一行为元素名称+种类个数, 删除第六行。
4 #并准备计算所需的文件, 将.x文件还原为.f, 已被修改不同应变参数。
5 #调用脚本elastic.sh
6         for i in 1 2 nc1 nc2 nc3
7             do cd $i/
8                 cp ../../CONTCAR OLDPOS
9                 sed -i -e '1cXXX 3' -e '6d' OLDPOS
10                cp ../../KPOINTS .
11                cp ../../POTCAR .
12                ifort -o defvector.x defvctor.f
13                sh elastic.sh
14                cd ../
15                done
```

elastic.sh

```
1 #!/bin/sh
2
3 #遍历创建文件夹并调用施加应变的脚本defvector.x
4 for i in -0.02 -0.01 0.00 0.01 0.02
5 do
6 echo $i | ./defvector.x
7 cat tempPOS
8
9 #创建计算需要的各个应变后的文件并进行一次ISIF=2的优化, 以及一次静态计算
10 #并将应变比例(文件夹名称)、能量、体积、时间输入到SUMMARY文件中
11 mkdir strain.$i
12 cp POTCAR KPOINTS defvector.x strain.$i ???
```

```

13 cd strain.$i
14 cp ..../tempPOS POSCAR
15 cat >INCAR <<!
16 省略
17 !
18 echo "deta = $i " ;mpirun -n 72 vasp_std |tee -a out_r.log
19 E1 = $(grep "E0" OSZICAR | tail -1 |awk '{printf "%12.6f \n" , $5}')
20 V1 = $(grep "volume" OUTCAR | tail -1 |awk '{printf "%12.6f \n" , $5}')
21 T1 = $(grep "Elapsed" OUTCAR | tail -1 |awk '{printf "%12.6f \n" , $4}')
22 echo "The relax results :" $i $V1 $E1 $T1 >>SUMMARY
23 cp CONTCAR POSCAR
24 cat >INCAR<<!
25 省略
26 !
27 echo "deta = $i " ;mpirun -n 72 vasp_std |tee -a out_r.log
28 E1 = $(grep "E0" OSZICAR | tail -1 |awk '{printf "%12.6f \n" , $5}')
29 V1 = $(grep "volume" OUTCAR | tail -1 |awk '{printf "%12.6f \n" , $5}')
30 T1 = $(grep "Elapsed" OUTCAR | tail -1 |awk '{printf "%12.6f \n" , $4}')
31 echo "The static results :" $i $V1 $E1 $T1 >>SUMMARY
32 rm WAVECAR CHG*
33 cd ..
34 echo $V1 $E1 $T1 >>SUMMARY
35 done

```

defvertor.f

```

1 program defvector
2 #定义变量：
3 real*8 privect,strvect,delta,strten,strain,pos,alat
4 dimension privect,strvect,strten,strain
5 dimension pos(100,3)
6 character*10 bravlat, natomi, nn
7 dimension natomi(10)
8
9 #######读取文件
10 open(7,file='OLDPOS')
11 read(7,*) title ,ntype
12 read(7,*) alat
13 do i=1,3
14     read(7,*) (privect(i,j),j=1,3)
15     write(*,*) (privect(i,j),j=1,3)
16     enddo
17     read(7,*) (natomi(i),i=1,ntype)
18     nn=0
19     do i = 1,ntype
20         nn = nn +natomi(i)

```

```
21      enddo
22      read(7,*) direct
23      do i= 1,nn
24          read(7,*) (pos(i,j),j=1,3)
25      enddo
26
27 ##### 施加应变， 理论见 4.7.3
28      read(*,*) detal
29
30      strain(1) =
31      strain(2) =
32      strain(3) =
33      strain(4) =
34      strain(5) =
35      strain(6) =
36
37      strten(1,1) = strain(1)+1.0
38      strten(1,2) =0.5*strain(6)
39      strten(1,3) =0.5*strain(5)
40      strten(2,1) =0.5*strain(6)
41      strten(2,2) = strain(2)+1.0
42      strten(2,3) =0.5*strain(4)
43      strten(3,1) =0.5*strain(5)
44      strten(3,2) =0.5*strain(4)
45      strten(3,3) = strain(3)+1.0
46
47 ## 写入应变后的数据
48      do i= 1,3
49          write(*,100)(strvect(i,j),j=1,3)
50      enddo
51 100 format(3f20.15)
52      open(3,FILE="tempPOS")
53      write(3,'(A10)') title
54      write(3,'(f15.10)') alat
55      do i=1,3
56          write(3,100)(strvect(i,j),j=1,3)
57      enddo
58      write(3,'(10I4)') (natomi(i), i=1,nype)
59      write(3,'(A6)') direct
60      do i=1, nn
61          write(3,100)(pos(i,j),j=1,3)
62      enddo
63
64      end
```

66

```
#####
#####
```

## 4.8 热学性质

在 4.5 声子谱计算中已经得到 FORCE\_CONSTANTS 力常数文件,准备输入文件,命名 mesh.conf:

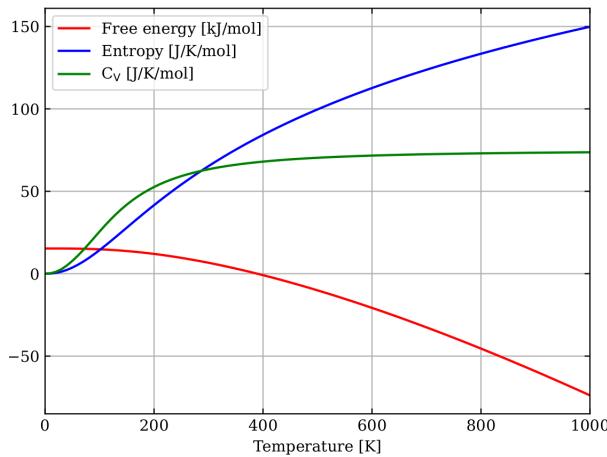
```
1 DIM = 2 2 1 # 扩胞倍数
2 MP = 8 8 1 # 取点密度此处为 slab 或二维材料取值
3 FORCE_CONSTANTS = READ # 读取力常数文件
4 #DIM 与 MP 的具体取值要依照具体体系进行修改。
```

输入运行:

```
1 phonopy -c POSCAR0 -p -t -s mesh.conf
```

即可得到保存为 thermal\_properties.pdf 的热学性质曲线,以及为 thermal\_properties.yaml 的数据。注意: 其中热学性质为 unit cell 即以 POSCAR0 为一个单元的数据。

MoS2 thermal\_properties.pdf:



## 4.9 热电性质

### 4.9.1 基本参数介绍

热电优值:

$$zT = \frac{\alpha^2 \sigma T}{\kappa}$$

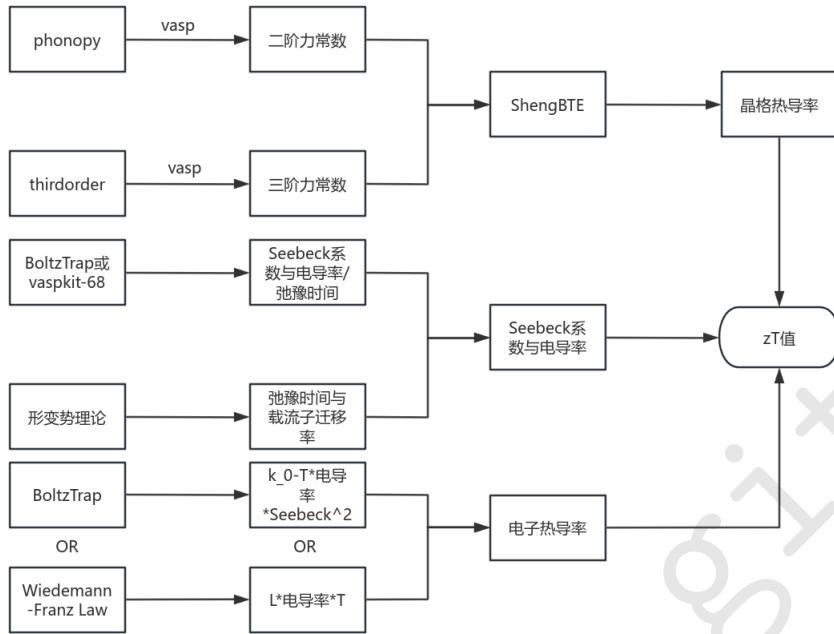
其中  $\alpha$  为 Seebeck 系数,  $\sigma$  为电导率,  $T$  为温度,  $\kappa$  为热导率。注意: 在计算中热导率分为电子热导率以及晶格热导率,即:

$$\kappa = \kappa_e + \kappa_l$$

### 4.9.2 基本流程介绍

Seebeck 系数、电导率: BoltzTrap 或 vasptkit-68、形变势理论

晶格热导率: phonopy、Thirdorder、ShengBTE



#### 4.9.3 Thirdorder

计算三阶力常数与二阶力常数类似, 二阶力常数流程见 4.5 声子谱计算给出, 将高精度优化完成的结构文件放入一个新建文件夹中, 输入以下代码运行 python 文件包即可生成 3RD.POSCAR.\*。

```
1 thirddorder_vasp.py sow 2 2 1 -10
```

其中 2 2 1 为扩胞倍数, 扩胞倍数要求与计算二阶力常数倍数一致, -10 为考虑相邻的 10 个原子, 原则上考虑原子数越多精度越高计算量越大, 一般需要进行测试。

将生成的各个 POSCAR 结构文件分别放入各自的文件夹中进行自洽计算, 其中 INCAR 重要参数:

```
1 EDIFF = 1E-8
2 IBRION = -1
3 NSW = 0
```

最后, 输入读取各自的计算好的 vasprun.xml 文件, 生成 FORCE\_CONSTANTS\_3RD:

```
1 find {001..140} -name vasprun.xml | sort -n |
2 thirddorder_vasp.py reap 2 2 1 -10
```

其中 find 001..140 -name vasprun.xml 为寻找文件夹分别为 001 到 140 中名为 vasprun.xml 的文件。thirddorder\_vasp.py reap 2 2 1 -10 为运行 python 文件包扩胞倍数为 2 2 1 且考虑相邻 10 个原子。

此时, 我们有二阶力常数和三阶力常数, 可以导入 ShengBTE 进行晶格热导率的计算。

#### 4.9.4 ShengBTE

将计算出来的二阶力常数文件重命名为 FORCE\_CONSTANTS\_2RD, 与 FORCE\_CONSTANTS\_3RD 一起复制进去, 并准备 CONTROL。

```

1 #CONTROL for ShengBTE
2 &allocations
3 nelements=2 # 原子种类
4 natoms=3 # 原子个数
5 ngrid(:)=21 21 1 # 取点密度，需要测试
6 &end
7 &crystal
8 lfactor=0.1
9 # 转换系数，因为vasp和ShengBTE单位不一样，取0.1
10 lattvec(:,1)=1.59 -2.75 0.00
11 lattvec(:,2)=1.59 2.75 0.00
12 lattvec(:,3)=0.00 0.00 17.33
13 elements= "Mo" "S"
14 types= 1 1 1
15 positions(:,1)=0.00 0.00 0.00
16 positions(:,2)=0.33 0.66 0.09
17 positions(:,3)=0.33 0.66 0.909
18 scell(:)=2 2 1 # 扩胞倍数
19 &end
20 &parameters
21 T=300 # 温度
22 scalebroad=0.1 # 重要的参数
23 maxiter=2000 # 最大循环步数
24 &end
25 &flags
26 nonanalytic=.TRUE.
27 nanowires=.FALSE.
28 convergence=.TRUE. # 算法相关
29 &end

```

10-12 为晶胞晶格参数

15-17 为原子位置

输入命令运行：

1

ShengBTE

等待运算收敛完成，获得各种热学性质的文件。

#### 4.9.5 BoltzTrap

BoltzTrap 能计算 Seebeck 系数以及电导率除以弛豫时间的值，前置准备文件将在高精度优化的基础上，新建名为 case 的文件夹，将优化后的文件夹的所有内容复制到 case 中，借助 vasppkit，运行 vasppkit781 命令：

1

vasppkit -task 781

将会读取 DOSCAR、INCAR、EIGENVAL、POSCAR 文件生成 case.energy、case.intrans、case.struct 文件。将 case.energy、case.struct 文件第一行修改为 case，执行：

```
1 x_trans BoltzTrap
```

运算完成！注意：大小写。

其中输出文件中 case.trans 为最重要的文件，其中每列数据的值的含义参考：

The traces of the conductivity tensors are written as a function of  $\mu$  and T in the file called case.trace. You must write your own program/script to extract the information you need. The

column	1	2	3	4	5	6	7	8	9	10
quantity	$\mu$	T	N	$n(\mu)$	S	$\sigma/\tau$	$R_H$	$\kappa^0$	c	$\chi$
unit	Ry	K	e/uc	e/uc	V/K	1/( $\Omega$ m s)	$m^3/C$	W/(m K s)	J/(mol K)	$m^3/mol$

Table 1.2: Output in case.trace

#### 4.9.6 vaspykit-68

vaspykit 此功能与 BoltzTrap 有相同的算法，作为替代，重要文件为 INCAR、KPOINTS、INPUT.in。

注意：在进行此计算前 POSCAR 为已经高精度优化后的结构。

其中 INCAR 要修改为：

```
1 LORBIT = 11
2 NEDOS = 2001
3 EDIFF = 1E-8
```

KPOINTS 为 vaspykit 681 生成的。

由此进行一次 vaspy 计算。

新建 INPUT.in 文件：

```
1
2 # 1 for gradient method with constant relaxation time
3 approximation
4 3D
5 # 2D for slab, 3D for bulk
6 -2.0 2.0
7 # Minimum and maximum values of chemical potential
8 (float type, in units of eV), referenced to the Fermi energy
9 1000
10 # Number of intervals between the minimum and maximum
11 chemical potential values
12 300.0
13 # Temperature (float type, in units of K)
14 1.0
15 # Relaxation time (float type, in units of s),
16 Set 1.0 if you don't know the exact relaxation time
17 0.0
18 # Desired gap value in scissor correction (float type,
19 in units of V), Zero means make no correction,
```

```

20 available only non-spin-polarization calculation
21 1
22 # Finite difference method (integer type)

```

调用 vasptkit682 计算:

```
1 vasptkit -task 682
```

#### 4.9.7 形变势理论

三维体系基本公式为:

$$\mu = \frac{2\sqrt{2\pi}e\hbar^4 C_{ii}}{3m_e^{\frac{5}{2}}(k_0T)^{\frac{3}{2}}E_1^2}$$

\*\*\*\*<https://doi.org/10.1103/PhysRev.80.72>\*\*\*\*

其中  $\mu$  为载流子迁移率,  $e$  为单电荷电量,  $\hbar$  为约化普朗克常数,  $C_{ii}$  为弹性常数,  $m_e$  为有效质量,  $k_0$  为玻尔兹曼常数,  $T$  为温度,  $E_1$  为形变势能。

载流子迁移率与弛豫时间的关系式为:

$$\mu = \frac{e\tau}{m_e}$$

即:

$$\tau = \frac{2\sqrt{2\pi}\hbar^4 C_{ii}}{3m_e^{\frac{3}{2}}(k_0T)^{\frac{3}{2}}E_1^2}$$

对于二维材料:

$$\mu = \frac{2e\hbar^3 C_{ii}}{3m_e^2 k_0 T E_1^2}$$

$$\tau = \frac{2\hbar^3 C_{ii}}{3m_e k_0 T E_1^2}$$

其中, 约化普朗克常数  $\hbar = \frac{h}{2\pi} = 1.05457266(63) \times 10^{-34} J \cdot s$ , 玻尔兹曼常数  $k_0 = 1.380649 \times 10^{-23} J/K$ , 单电荷电量  $e = 1.60217733 \times 10^{-19} C$ , 只需将弹性常数  $C_{ii}$ 、形变势能  $E_1$ 、有效质量  $m_e$  求出即可。

对于弹性常数  $C_{ii}$  的计算:

$$C_{ii} = \frac{1}{V_0} \frac{\partial^2 E}{\partial(\Delta l/l_0)^2}$$

其中  $V_0$  为稳态下的晶胞体积, 在二维材料中为底面积。 $\Delta l$  为沿着某轴方向的形变大小,  $l_0$  为某轴方向的晶格参数,  $E$  为总能量。本式即为能量随拉伸比例的二阶偏导数变化量除以晶胞体积。称为求弹性常数的能量应变法。对于形变势能

$$E_1 = \frac{\partial E_{edge}}{\partial(\Delta l/l_0)}$$

其中  $E_{edge}$  与导带底  $CBM$  和价带顶  $VBM$  有关。本式即为随着形变方向上导带底或者价带顶的能量的偏移。其中  $CBM$  与  $VBM$  的值与真空能级  $VL$  有关, 真空能级即不受电子核束缚成为自由粒子的能级。有:

$$E_{VBM} = E'_{VBM} - E_{VL}$$

$$E_{CBM} = E'_{CBM} - E_{VL}$$

通过上式子计算得出的  $E_{VBM}$  及  $E_{CBM}$  为需要的值。

关于计算  $E_{VL}$ , 在 VASP 中, INCAR 中加入 LVHAR=.T., 自洽计算后, 即可利用 p4vasp 求得真空能级。注意三维材料用芯能级来修正。

注意  $VBM$  为空穴的有效质量,  $CBM$  为电子的有效质量。

有效质量:

$$\left(\frac{1}{m_e}\right)_{ij} = \frac{1}{\hbar^2} \frac{\partial^2 E(k)}{\partial k_i \partial k_j}, i, j = x, y, z$$

所以最后, 所求的弛豫时间与载流子迁移率分为空穴与电子两部分。

只考虑长声学波与载流子的散射机制, 只考虑单抛物线线型的能带结构的情况。利用电声耦合计算出来的结果, 相对形变势理论误差更小。

#### 4.9.8 电子热导率

4.9.5 中 case.trans 文件中序号 8 为  $\kappa_0$  为闭路下的电子热导率, 单位为  $W/(mKs)$ , 由于是闭路条件下的, 所以不可直接当做电子热导率用, 需要通过以下方法之一进行修正:

$$\kappa_e = \kappa_0 - T\sigma S^2 = L\sigma T$$

#### 4.9.9 Seebeck 系数与电导率

4.9.5 中 case.trans 文件中序号 6 为电导率与弛豫时间的比值, 单位为  $1/\Omega ms$ , 已根据形变势理论求得弛豫时间, 由此得出电导率。此外, 序号 5 为塞贝克系数。

#### 4.9.10 晶格热导率

4.9.4ShengBTE 中获得的各种热学性质的文件, 其中 BTE.KappaTensorVsT\_CONV 中可以得到在设定温度下各个方向上的晶格热导率。

#### 4.9.11 热电优值

将 4.9.8、4.9.9、4.9.10 的值代入 4.9.1 中, 即可得到热电优值 (ZT)。

### 4.10 高熵合金

高熵合金的主要的组成一般包含四种以上的元素, 高熵合金具有更高的强度、硬度、耐磨性、耐腐蚀、耐辐照等被广泛关注。

#### 4.10.1 ATAT-mcsqs 生成准随机结构

ATAT(Alloy Theoretic Automated Toolkit) 中的一个核心工具 mcsqs, 用于生成特殊的准随机模型 (SQS), 主要为第一性原理软件提供无序性较高的晶胞模型。

1、以 Fcc-Al 为例, 获取 Fcc-Al 的晶胞, 利用 vaspykit 414 功能生成 lat.in, 重命名为 rndstr.in, 并添加其中的原子位点中元素比例标示:

1	4.03 4.03 4.03 90.0 90.0 90.0
2	1.0 0.0 0.0
3	0.0 1.0 0.0

```

4      0.0 0.0 1.0
5
6      0.000 0.000 0.000 Al=0.2 Co=0.2 Cr=0.2 Fe=0.2 Ni=0.2
7      0.000 0.500 0.500 Al=0.2 Co=0.2 Cr=0.2 Fe=0.2 Ni=0.2
8      0.500 0.000 0.500 Al=0.2 Co=0.2 Cr=0.2 Fe=0.2 Ni=0.2
9      0.500 0.500 0.000 Al=0.2 Co=0.2 Cr=0.2 Fe=0.2 Ni=0.2

```

2、使用 corrdump 函数生成团簇关联函数:

```
1 corrdump -nop -noe -ro -l=rndstr.in -clus -2=4.5
```

其中, -2=9 代表原子对近邻距离为 4.5, 近邻距离越大越精准, 生成无序结构的计算量也会越大。所谓近邻距离也即是相互靠近最近的原子对的距离, 与选定原子靠近最近的第一个原子称为第一近邻距离, 第二个原子则称为第二近邻距离, 以此类推。上述中选定近邻距离为 4.5, 也就是计算考虑距离选定原子以 4.5 为半径的圆中所有的原子。

例如, 对于晶格参数为 1 的 FCC 晶胞, 第一近邻距离为 0.71, 第二近邻距离为 1。运行后立即生成 clusters.out 和 sym.out

3、生成超胞结构文件:

```
1 mcsqs -n=32
```

其中-n=32 表示生成的超胞无序结构有 32 个原子数量。此时对于初始模型(有 4 个 wyckoff 点位), 每个位点有 8 个原子, 每个元素有  $8 \times 0.2 = 1.6$  个原子, 此时原子不为整数, 必然会报错(Impossible to match point correlations due to incompatible supercell size)。修改为每个位点有 5 个原子, 位点每个元素为 1 个原子(整数), 即总原子数量为 20 个。使用并行计算加快无序计算速度:

```
1 mpirun -np 72 mcsqs -n=20
```

生成 bestcorr.out、bestsq.out、mcsqs.log、rndstrgrp.out、sqscell.out 文件, 其中 mcsqs.log 记录运行过程, Objective\_function 的值越负无序性越好, Correclations\_mismatch 值越接近零精度越好。注意: 运行不会主动停止, 可以运行一段时间后主动停止。

4、转换模型文件格式:

```
1 str2cif <bestsq.out> pos.cif
```

将 cif 文件导入 VESTA 中, 转换为适用于 VASP 用的 POSCAR 文件。

5、注意: 此时对 POSCAR 文件进行对称性分析, 会发现晶体结构或许是对称性最差的三斜晶系。

#### 4.10.2 生成指定晶系的无序结构

1、在 4.10.1(3) 中生成了 sqscell.out 文件, 其中给出许多可被寻找的晶型, 如果制定了 FCC 作为最后无序超胞的晶型, 则要将其中的所有内容删除, 并输入:

```

1
2
3      2 0 0
4      0 2 0
5      0 0 2

```

其中第一行定义了要寻找的晶型的种类，第三到六行定义了晶系的基矢和扩胞倍数，即上述为扩胞倍数  $2*2*2$ 。

2、运行命令：

```
1 mpirun -n 72 mcsqs -rc
```

后续与 4.10.1(4) 一致，注意：rndstr.in 文件要与需要指定生成的晶系一致，例如 rndstr.in 是使用角度为 60 60 60 的原胞生成的，则后面生成的无序晶体角度依旧为 60 60 60，晶系为三斜。

也即：若要生成立方晶系的无序晶体必须在满足 rndstr.in 也是由 90 90 90 角度的立方晶体生成的，而后在进行如上操作才可。

## 5 软件安装以及配置

### 5.1 VASP

#### 5.1.1 版本说明

操作系统版本：centos 7 以及 Ubuntu 20，Vasp 版本：vasp.5.4.4

#### 5.1.2 安装环境准备

编译器安装

```
1 # CentOS 7 中
2 sudo yum install gcc-c++
3 sudo yum install gcc
4 # Ubuntu 20 中
5 sudo apt update
6 sudo apt install gcc
7 sudo apt install g++
```

“”

#### 5.1.3 安装 OneAPI

安装包：l\_BaseKit\_p\_2023.0.0.25537\_offline.sh 和 l\_HPCKit\_p\_2023.0.0.25400\_offline.sh，用的是 2023 年版本，可以选择最新版本。

在安装包目录下，输入以下分别安装：

```
1 ./l_BaseKit_p_2023.0.0.25537_offline.sh
2 ./l_HPCKit_p_2023.0.0.25400_offline.sh
```

Basekit 可以只安装：Math Kernel Library

HPCKit 安装：MPI Library 和两个 Compiler Classic 节省空间。

安装完成后，在 oneapi 文件夹中有 setvars.sh，可用于调用环境：

```
1 source setvars.sh
```

因为需要每次进终端都要调用故：

```
1 gedit ~/.bashrc
```

在文件内输入:

```
1 source oneapi 编译器的地址/setvars.sh
```

保存后，重新打开终端会自动调用。

进入:

```
1 /intel/oneapi/mkl/2023.2.0/interfaces/fftw3xf
```

编译 libfftw3xf:

```
1 make libintel64
```

出现 libfftw3xf\_intel.a 文件，编译完成！

#### 5.1.4 安装 VASP

解压 Vasp 包:

```
1 tar -zxf vasp.5.4.4.tar.gz
```

进入解压出来包的 arch 文件夹中把 makefile.include.linux\_intel 复制到 vasp 文件夹中，并重新命名: makefile.include，打开此文件将 OFLAG = -O2 后面添加 -xhost 保存。

执行编译:

```
1 make makefile.include
```

编译完成后在 bin 文件中会有 vasp\_std、vasp\_ncl、vasp\_gam 三个执行文件。

获取此目录的地址:

```
1 pwd
```

复制此地址

输入:

```
1 gedit ~/.bashrc
```

把 vasp 可执行文件加入 PATH 环境变量:

```
1 export PATH=$PATH:刚才复制的地址
```

完毕！

#### 5.1.5 vasppkit 安装

```
1 form how_to_intall
2 1) cd vasppkit/
3 2) bash setup.sh
4 3) modify ~/.vasppkit file based on your machine environment
5 4) source ~/.bashrc
```

注：参考只作为参考，有些错误并未修改。

### 5.1.6 参考文献

- 1、[全网最全最简单的零基础 vasp 安装教程(良心之作)](<https://www.bilibili.com/video/BV15e4y1M7Do/>)
- 2、[VASP6.3.2 版本安装教程] (<https://blog.csdn.net/passsive/article/details/132982277>)

## 5.2 BoltzTrap

### 5.2.1 环境以及软件安装

下载网址:[boltztrap] (<https://www.tuwien.at/en/tch/tc/theoretical-materials-chemistry/boltztrap>)

软件用处: 计算热电材料 Seebeck 系数、电导率/弛豫时间

安装包解压:

```
1      tar -jxvf BoltzTrap.tar.bz2
2      # 注意此时安装包格式为 tar.bz, 当为 tar.gz 才使用 zxvf
```

根据自己机器的环境对以下路径的 makefile 文件进行修改

```
1      解压缩路径 /boltztrap-1.2.5/src
```

我修改为:

```
1 SHELL = /bin/sh
2
3 # gfortran
4 FC = gfortran
5 #FC = gfortran-mp-4.5
6
7 FOPT = -g -march=native -funroll-loops -O3 -ffast-math -fgcse-lm
8   -fgcse-sm -ffast-math -ftree-vectorize -fexternal-blas
9 #FOPT += -g -p -pg -Wall -fbounds-check -finit-integer=-666 -finit-real=nan
10 #LDFLAGS =
11 LIBS = -llapack -lblas
12
13 # ifort
14 FC = ifort #注意此处修改
15 FOPT = -FR -mp1 -w -prec_div -pc80 -pad -ip -DINTEL_VML
16
17 #LDFLAGS = -L/opt/intel/mkl/10.0.1.014/lib/32/ -Vaxlib -static-libcxa -pthread
18 #LIBS = -lmkl_lapack -lmkl_ia32 -lguide
19 # ifort bis: other mkl options
20 #LDFLAGS = $(FOPT) -L/opt/intel/mkl/10.0.1.014/lib/32 -pthread -i-static
21 #LIBS = -lmkl_lapack -lmkl -lguide
22 LDFLAGS = $(FOPT) -L/opt/intel/oneapi/mkl/2022.2.0/bin/intel64
23 -pthread -i-static #注意此处修改
24 LIBS = -lmkl_lapack95_lp64 -lmkl blas95_lp64
25 -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -qopenmp -lpthread #注意此处修改
```

然后在此目录下进行编译:

```
1     make
```

完毕!

### 5.2.2 结合 vasakit 尝试运行

使用 vasakit 中的 781 进行输入文件准备。

需要高精度自洽计算后的 EIGENVAL,OUTCAR,POSCAR,DOSCAR。

生成 intrans .energy .struct。

将 case.energy, case.struct 第一行修改为 case(文件夹名字即可)

输入运行:

```
1     x_trans BoltzTraP
```

完毕!

### 5.2.3 参考文献

1、[tar 命令 zcvf zxvf 和 jcvf jxvf]([https://blog.csdn.net/qq\\_25908839/article/details/102877287](https://blog.csdn.net/qq_25908839/article/details/102877287))

2、[BoltzTraP 下载及安装—给需要的人](<https://zhuanlan.zhihu.com/p/359080885>)

3、[vasp+BoltzTrap+vaskit 安装及使用](<https://www.bilibili.com/opus/497943493095599015>)

4、[第一性原理热点教程天玑算]([https://www.bilibili.com/video/BV1b64y147df/?vd\\_source=540d345d77e419a2](https://www.bilibili.com/video/BV1b64y147df/?vd_source=540d345d77e419a2))

## 5.3 ShengBTE

### 5.3.1 环境安装

Thirdorder: 用来计算三阶力常数

ShengBTE: 对三阶以及二阶力常数进行处理，计算热导率。

下载软件:[ShengBTE](<https://bitbucket.org/sousaw/shengbte/downloads/>)、[thirdorder](<https://bitbucket.org/spglib>)(<https://sourceforge.net/projects/spglib/>)

编译软件: intel OneAPI/

### 5.3.2 安装

spglib: 用于查找和处理晶体对称性

解压:

```
1     tar -zxvf spglib-1.9.7.tar.gz
```

进入解压后的文件包，创建安装文件夹

```
1     mkdir build
```

进入 build 目录下获取目录路径:

```
1     pwd
```

安装目录生成:

```
1 ./configure --prefix=刚才获取的目录
```

解压后的文件包中编译:

```
1 make
```

安装:

```
1 make install
```

ShengBTE:

解压: 由于上述下载链接下载的是 zip 压缩包, 直接提取即可。

进入解压后的文件包中, 存在 arch.make.example 文件, 将其复制到 Src 文件夹中, 重命名为 arch.make。并进行修改, 我修改为:

```
1 export FFLAGS=-traceback -debug -O2 -static_intel -qopenmp
2 export LDFLAGS=/home/panrk/software/spglib-1.9.7/build/lib/libsymspg.a
3 # 此处注意 spglib 中 libsymspg.a 的位置
4 export MPIFC=mpiifort # 此处注意 我使用 ifort 编译
5 MKL=/opt/intel/oneapi/mkl/2022.2.0/lib/intel64/libmkl_lapack95_lp64.a -Wl,--start-group
6 /opt/intel/oneapi/mkl/2022.2.0/lib/intel64/libmkl_intel_lp64.a
7 /opt/intel/oneapi/mkl/2022.2.0/lib/intel64/libmkl_sequential.a
8 /opt/intel/oneapi/mkl/2022.2.0/lib/intel64/libmkl_core.a -Wl,--end-group -lpthread -l
9 # 以上 MKL 对应自己编译器中的位置
10 export LAPACK=$(MKL)
11 export LIBS=$(LAPACK)
```

保存, 编译:

```
1 make
```

在 ShengBTE 文件夹中会存在测试文件夹, 若编译成功在 ShengBTE 中会出现以 ShengBTE 命名的可执行文件, 随意进一个测试文件夹, 如 Test-Vasp, 运行 ShengBTE:

```
1 mpirun -np 72 ./ShengBTE
```

输出:

```
1 Info: normal exit after xxxx seconds
```

计算完毕!

环境变量修改, 方便以后运行:

```
1 vi ~/.bashrc
```

按 i 键进行编辑, 最后一行添加:

```
1 export PATH = "你的 ShengBTE 可执行文件所在目录:$PATH"
```

输入, 退出保存:

```
1 :wq
```

下次再打开终端即可直接输入：

```
1 ShengBTE
```

执行。

thirdorder:

解压，同样为 zip 文件。将解压后文件夹中的 setup.py 进行修改，添加 INCLUDE\_DIRS 和 LIBRARY\_DIRS 的路径，并修改二、三行为 setuptools (Python 中的打包分发工具)：

```
1 import numpy
2 from setuptools import setup
3 from setuptools import Extension
4
5 # Add the location of the "spglib/spglib.h" to this list if necessary.
6 # Example: INCLUDE_DIRS=["/home/user/local/include"]
7 INCLUDE_DIRS = ["/home/panrk/software/spglib-1.9.7/build/include"]
8 # Add the location of the spglib shared library to this list if necessary.
9 # Example: LIBRARY_DIRS=["/home/user/local/lib"]
10 LIBRARY_DIRS = ["/home/panrk/software/spglib-1.9.7/build/lib"]
```

保存，退出，在文件夹中进行编译：

```
1 sh compile.sh
```

同样在添加环境变量方便调用，在/.bashrc 中添加：

```
1 export PATH="你的 thirdorder 文件夹路径 :$PATH"
```

完毕！

### 5.3.3 参考文献

[Linux ./configure --prefix 命令]

(<https://blog.csdn.net/xiyangfan/article/details/5321790>)

[记录 WSL2 下安装 Intel OneAPI, LAMMPS, VASP, ShengBTE 及 Thirdorder 的过程]

(<https://zhuanlan.zhihu.com/p/11864498005>)

[ShengBTE 报错“wrong number of force constants for the specified scell” 的解决方法]

(<https://zhuanlan.zhihu.com/p/574681290>)

[VASP 结合 vaspkit+ShengBTE 计算热电优值（二）]

(<https://zhuanlan.zhihu.com/p/693528740>)

[ShengBTE 简单介绍、安装与使用]

(<https://blog.csdn.net/icehoqion/article/details/125183366>)

## 5.4 MS

注意只支持 Centos 系统。

#### 5.4.1 安装（无网关版）

安装依赖

```
1 sudo yum install -y glibc* compat* libgcc* libsrdc* redhat* csh*
```

解压安装包

```
1 tar -zxvf MS 安装包名称
```

进入文件夹执行

```
1 ./install
```

一直回车最后输入：99

```
1 99) Finished with license configuration
```

找到文件包里面的 msi.lic 修改 License

```
1 /home/....BIOVIA/BIOVIA_LicensePack/Licenses
```

将第一行

```
1 SERVER ... ANY 27000
```

中的... 修改为本机名称

进入

```
1 /home/....BIOVIA/BIOVIA_LicensePack/linux/bin
```

执行安装

```
1 ./lp_install ~/msi.lic
```

完毕！

#### 5.4.2 文件上传

在 windows 下运行数据以 file 文件保存，复制到服务器硬盘中。

#### 5.4.3 模块运行（以 DMOL3 为例）

```
1 #!/bin/bash
2 MS_TNSTALL_PATH=/public/home/software/BIOVIA/MaterialsStudio20.1
3 #MS 文件位置 设置为安装的位置
4
5 cp /public/home/ABC #进入工作文件
6
7 DMOL3_SCRIPT=$MS_TNSTALL_PATH/etc/DMol3/bin/RunDMol3.sh
8 #运行脚本赋值给 DMOL3_SCRIPT 此脚本是在MS安装位置中存在的
9 #根据需求不一样进行修改 此处为DM013模块
10
```

```
11 $DMOL3_SCRIPT ABC #运行的文件名称  
12  
13 ##注：其实上面两行类似于RunnDmol3.sh（模块脚本）ABC0（运行文件）  
14 #进入要运行的文件内，运行模块。
```

#### 5.4.4 下载查看

在文件夹中 \*.outmol 即运行结果数据。

#### 5.4.5 参考文献

1、[十分钟搞定 MS 在 Linux 服务器上的安装 | Materials Studio Linux 版于集群、超算、高性能计算中心的安装 | MS 杨站长华算科技]

(<https://www.bilibili.com/video/BV1ym4y187gta>)

2、[Materials Studio for linux 库文件分享]

(<http://bbs.keinsci.com/thread-14926-1-1.html>)

3、[在 CentOS 上安装 MS8.0 的详细过程]

(<http://bbs.keinsci.com/thread-1405-1-1.html>)

### 5.5 关于大模型的使用以及本地部署—以 Deepseek 为例

诸如 DeepSeek、chatGPT、豆包等大模型工具日益成熟，拥有超越百度等搜索引擎的便利性与精准性。由于信息时效性与程序的局限性，推荐在使用过程中提供的知识作为参考，经验证后进行使用，不建议盲目使用。目前，可以通过网络登入相应网站进行使用，此外类似于 Deepseek 还可以进行不需要网络的本地部署。

#### 5.5.1 DeepSeek 本地部署

1、配置环境

系统：Linux 或 Windos 系统

配置：96 线程，192Gb 内存。4 线程，16Gb 内存。

2、网站

[ollama]([ollama.com](https://ollama.com))

3、流程

下载适用于系统的 [ollama](<https://ollama.com/download>) 安装包，并进行安装。在我下载的过程中因为文件过大下载缓慢，并且在 Linux 系统上出现过下载失败的错误，只要把/usr/local/lib/路径下的 ollama 的空文件夹删除掉，重新下载即可。在 Linux 系统上的下载命令为：

```
1 curl -fsSL https://ollama.com/install.sh | sh
```

curl -fsSL <https://ollama.com/install.sh> | sh

如 [ollama download for linux](<https://ollama.com/download/linux>) 所示。在 window 系统上只需将下载好的安装包进行安装即可，只不过安装位置无法进行手动更改仅能安装到 C 盘。打开 CMD(windos) 或终端 (Linux) 输入：

```
1 ollama -v
```

显示版本号即为安装成功，在安装过程中由于依赖关系还导致 GPU 并行程序 CUDA 没有安装，由于机器并没有强大的显卡作为支持，这里进行忽略，不知道会不会对使用产生影响。

接下来对大模型进行安装在 [ollama models](<https://ollama.com/search>) 中选择 DeepSeek 并根据硬件所能支持的程度安装相应版本的 DeepSeek。

当出现：

```
1 Seed a message (/? for help)
```

提示安装成功，可以输入文字进行使用。

输入命令：

```
1 ollama list
```

可显示大模型名称，可以查看 ollama 上面安装了哪些大模型。

输入命令：

```
1 ollama run 大模型名称
```

即可使用。

## 5.6 关于 pymatgen 中 BSDOSPlotter 模块的使用

在 4.4.3 中提到利用 pymatgen 可对能带以及 DOS 进行结果的后处理，绘制成图。其中包含此类后处理的模块为 BSPlotter、DOSPlotter、BSDOSPlotter 以及 Vasprun 与 matplotlib，以下就以一个例子介绍如何使用 BSDOSPlotter 进行能带以及 DOS 的绘制。

### 5.6.1 一个例子

此时，已将 DOS 以及能带分别计算完毕，然后准备如下程序命名为 plot.py:

```
1 # 导入模块
2 import matplotlib.pyplot as plt
3 from pymatgen.io.vasp.outputs import Vasprun
4 from pymatgen.electronic_structure.plotter import BSDOSPlotter
5
6 # 定义
7 dosrun = Vasprun("你的 DOS 文件夹 vaspxml")# 读取 DOS
8 dos_data = dosrun.complete_dos # 总态密度
9 spd_dos = dosrun.complete_dos.get_spd_dos() # 分态密度
10
11 # 读取能带
12 bs_run = Vasprun("你的能带文件夹 vaspxml", parse_projected_eigen=True)
13
14 # 读取高对称点结构
15 bs_data =
16 bs_run.get_band_structure("高对称点 KPOINTS", line_mode=1, efermi=dosrun.efermi)
```

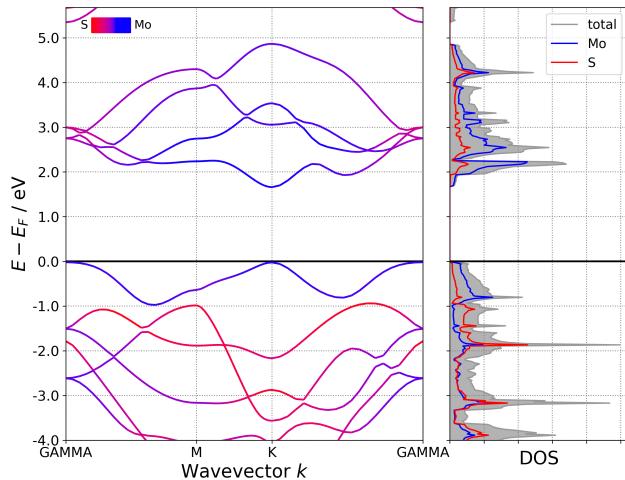
```
17  
18 #绘制  
19 plot = BSDOSPlotter(bs_projection = 'elements',dos_projection='elements')  
20 plot.get_plot(bs=bs_data,dos=dos_data)  
21 plt.savefig('plot.png',dpi = 300)  
22 plt.show()
```

运行：

```
1 python plot.py
```

### 5.6.2 结果输出

以  $MoS_2$  为例，如下图：



## 6 后记

知识就是力量。